ISTANBUL TECHNICAL UNIVERSITY ⋆ INSTITUTE OF SCIENCE

IMAGE COMPRESSION BASED ON CENTIPEDE MODEL

MASTER THESIS

Binnur KURT, B.Sc.

66517

Date Of Submission : 24 January 1997

Date Of Approval : 3 February 1997

Advisor : Assoc. Prof. Muhittin GÖKMEN

Member : Prof. Dr. Nadir YÜCEL

Member : Asst. Prof. A. Coşkun SÖNMEZ

YÜKSEKÖĞRETİM KURU
DOKÜMANTASYON

JANUARY 1997

İSTANBUL TEKNİK ÜNİVERSİTESİ ⋆ FEN BİLİMLERİ ENSTİTÜSÜ

KIRKAYAK MODELİNE DAYALI GÖRÜNTÜ SIKIŞTIRMA

YÜKSEK LİSANS TEZİ

Müh. Binnur KURT

Tezin Enstitüye Verildiği Tarih : 24 OCAK 1997

Tezin Savunulduğu Tarih : 3 ŞUBAT 1997

Tez Danışmanı : Doç. Dr. Muhittin GÖKMEN

Diğer Jüri Üyeleri : Prof. Dr. Nadir YÜCEL

: Yrd. Doç. Dr. A. Coşkun SÖNMEZ

OCAK 1997

PREFACE

I would like to thank my advisor, Associate Professor Muhittin Gökmen, not only for all his guidance, encouragement, and suggestions throughout my entire graduate study, but also for providing us a research environment.

I would also like to thank the research assistants Tahsin Demiral and İlker Ersoy for many useful discussions and conversations.

Special thanks go to Murat Vonalıoğlu who is very kind to share his computer which I have written the thesis.

I am grateful to my family for their care and encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

Image coding is a process which produces an image identical to the original one in the sense of quality and intelligibility, but occupies less space. The goal of this dissertation is to produce an algorithm, which we call image compression based on "centipede" model, for lossy-coding of an image in the way that edges, contrast and scale through edges are utilized to produce a powerful and sparse representation of the image. Edges detected by using generalized edge detector (GED) constitute very sparse information. It has been shown that edge brightness and contrast calculated through edge segments are not adequate to produce a reliable and precise representation. The proposed algorithm produces a contour code which consists of position, brightness, contrast and an estimate of the scale in the form of *width* for each edge element in the image. The segments obtained by tracing connected edge elements are sorted with respect to weighted sum of their length, mean contrast, deviation and curvature. The edges are then thresholded to eliminate some of which has less priority in this order. Coding of all these parameters is a bit-consuming operation. Since they change smoothly in a small neighborhood, they can be approximated by polynomials, then the coefficients of the polynomials are coded. Edge locations are coded by constructing differential chain code followed by Huffman coding and starting points are coded in the form of difference between lexicographically ordered points. A reliable approximation to the original image from the sparse information is obtained via solving the hybrid energy functional which spans $\lambda\tau$-space, where λ represents the smoothness of the image and τ represents the continuity of the image.

The proposed model and the algorithm has been tested on both real and synthetic images. Compression ratio is up to 180:1 for synthetic images and 10:1-100:1 for real images. Reconstructed images are evaluated both quantitatively with NMSE (normalized mean square error), SNR (signal-to-noise ratio) and PSNR (peak-to-peak SNR) and qualitatively with visual appearance of artifacts. We have experimentally shown that the proposed model preserves perceptually important features even at the high compression ratios.

ÖZET

KIRKAYAK MODELİNE DAYALI GÖRÜNTÜ SIKIŞTIRMA

Görüntü boyutlarının gün geçtikçe artması onların sıkıştırılmasını daha çok gerekli kılmaktadır. Görüntü sıkıştırmada amaçlanan, bir görüntünün aynı düzeyde kalite ve anlaşılabirlik seviyesine sahip ancak daha az yer kaplayan bir yaklaşığını elde etmeye çalışmaktır. Bazı uygulamalarda (örneğin, tıbbi uygulamalar) sıkıştırılmış görüntü ile gerçek görüntünün birbirinin aynı olması istenir. Bu durumda sıkıştırma yönteminin *kayıpsız* sıkıştırma yöntemi olması gerekir. Bazı uygulamalarda ise sıkıştırılmış görüntüde belirli oranlarda bozulmalara izin verilebilir. Bu tür sıkıştırma yöntemleri ise kayıplı sıkıştırma yöntemi adını alır.

Bu çalışmada, yeni bir ayrıt-tabanlı görüntü sıkıştırma yöntemi sunulmaktadır. Geliştirilen yöntem, bir kırkayak modeli yardımıyla ayrıtların yerlerini, kontrast bilgisini ve ayrıt ölçeği gibi bilgileri etkin bir biçimde kullanmakta ve bu gelişmiş model yardımıyla daha önceki ayrıt temelli yaklaşımlara kıyasla, özgün görüntüye daha yakın görüntü kurabilme olanağı sağlamaktadır. Bu yaklaşımda, önce görüntü ayrıtları, Canny ayrıt saptayıcıya kıyasla daha genel bir ayrıt saptayıcı olan ve değişik özelliklerde amaca uygun ayrıt üretebilen Genelleştirilmiş Ayrıt Saptayıcı (GAS) ile elde edilmiş ve daha sonra kırkayak modelini bu ayrıt parçaları üzerine yerleştirerek, görüntüdeki hızlı değişim bölgeleri kontrastları ve ölçekleri de göz önüne alınarak modellenmiştir. Bu model sayesinde, yalnız ayrıt yerleri ve yükseklikleri kullanıldığında ortaya çıkan bozulmaların yok edildiği gösterilmiştir. Çalışmada, GAS ile elde edilen ayrıtlar, uzunluk, çevrit boyunca ortalama benek değeri, ortalama kontrast ve ortalama eğrilik gibi özellikler göz önüne alınarak sıralanmış ve bunlardan belirli bir yüzdesinin seçilerek sıkıştırma oranının kontrol edilmesi sağlanmıştır. Ayrıt çevritleri üzerindeki benek değerleri, kontrast ve genişliklere sabit blok boyu ve değişken dereceli polinomlar uydurularak polinom katsayıları saklanmıştır.

Bir görüntüde, nesne sınırlarına karşılık gelen ve görüntüdeki bölütleri birbirinden ayıran ayrıtlar, o görüntüyü tanımlayan en önemli özelliklerden biridir. Bu ayrıtların hem sayıca az olmaları hem de görüntünün içeriği hakkında önemli bilgileri sağlamaları ayrıt temelli sıkıştırma algoritmalarının hareket noktasını oluşturmaktadır. Görüntüdeki seyrek ayrıt noktalarında benek değerleri hızlı değişmesine karşın, ayrıt dışı noktalarında değişim yavaş olmakta ve böylece ayrıt bilgilerinden görüntünün tümünü oluşturabilmek olası olmaktadır. İkinci kuşak sıkıştırma yöntemlerinin temel

özelliklerinden biri olan ayrıt temelli kodlamanın yararlarından biri, yüksek sıkıştırma oranlarında dahi görüntüye ilişkin önemli özelliklerin kaybolmaması, görüntünün hızla bulanıklaşmamasıdır. Ne var ki, bu yaklaşımda yalnızca ayrıtların konumu (ayrıt haritası) ve ayrıt boyunca kontrast bilgisi kullanıldığında, kurulan görüntülerde yapaylıklar oluşmaktadır. Bu çalışmada bu bilgilere ek olarak, ayrıtların ölçeklerinin de kullanılmasına olanak sağlayan bir model geliştirilmiş ve daha başarılı sonuçların elde edildiği gösterilmiştir.

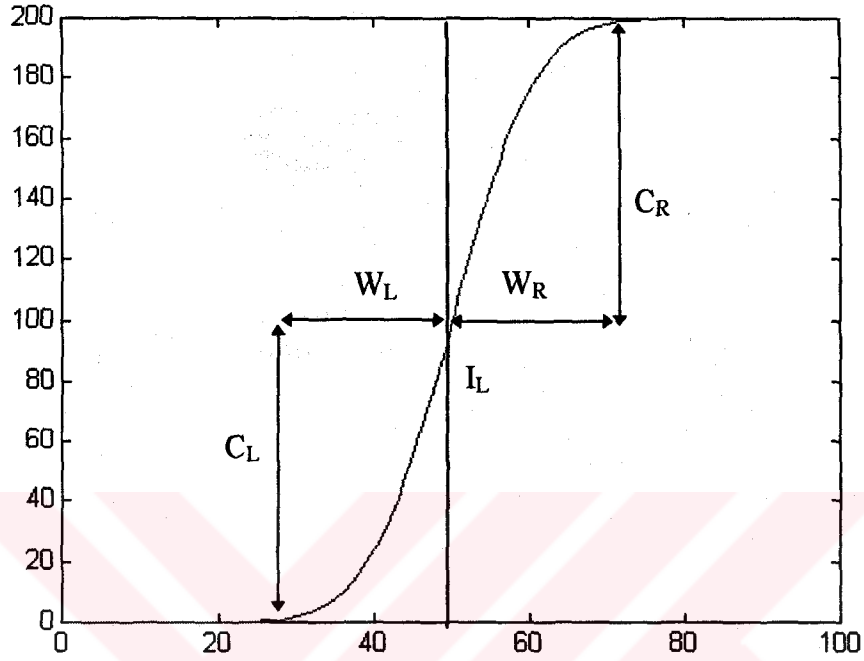
Geliştirilen yaklaşımda, önce ayrıtlar GAS ile elde edilmekte ve ayrıt parçaları, uzunlukları, ortalama benek değerleri, ortalama kontrast ve eğrilik değerleri göz önüne alınarak önem sırasına göre sıralanırlar. Daha sonra seçilen ayrıt parçaları çevresindeki değişimler, kırkayak modeli yardımıyla modellenmektedir. Bu modele ilişkin bilgilerin kodlanması farksal zincir kodu, Huffman kodlama ve polinomla eğri uydurma yöntemleri kullanılarak gerçekleştirilmiştir. Daha sonra modelden görüntülerin kurulması karma enerji fonksiyoneli yardımıyla gerçekleştirilmiştir.

Algoritmanın ilk aşamasında ayrıtlar GAS ile elde edilir. GAS, görüntüyü ölçek ve süreklilik özelliklerine bağlı olarak $\lambda\tau$-uzayında ifade etmeye olanak sağlayarak değişik amaçlar için istenen özellikte ayrıtlar üretir. Burada λ ölçek eksenini ve τ ise süreklilik eksenine karşı düşmektedir.

Elde edilen ayrıtlardan, izleme algoritması ile bağlı çevrit bölütleri elde edilir. İzleme algoritması sonlanma noktalarından itibaren aynı yönde izlemeye devam etmeye zorlanır. Çevritlerin izlenmesi ile elde edilen farksal zincir kodu, Huffman yöntemi ile kodlandığından bu şekil bir izleme, ayrıtların etkin bir şekilde kodlanmasını sağlar. Bu yaklaşımda ayrıtlar ortalama 1.8 bit ile kodlanmaktadır. Segmanların başlangıç noktaları ise sözdisimsel biçimde sıralanan başlangıç noktaları .aralarındaki fark şeklinde kodlanmaktadır. Başlangıç noktaları yaklaşık homogen bir biçimde dağıldıklarından bu şekil bir kodlama (x,y) koordinaları şeklindeki kodlamaya göre sıkıştırma kazancını arttırmaktadır.

Şekil 1'de bir ayrıt kesiti verilmiştir. Ayrıtın genişliği (W_R,W_L), ayrıtın o noktadaki normali doğrultusundaki kesiti için benek değerlerindeki değişimin küçük olmaya başladığı noktalara olan uzaklık olarak tanımlanmaktadır. Kontrast değeri (C_R,C_L) ise ayrıt üzerindeki benek değeri (I_L) ile ayrıtın genişliğinin belirlendiği noktadaki benek değeri arasındaki farktır. Kırkayak modeli ile her ayrıt noktası için (I_L ,W_R ,W_L, C_R, C_L) bilgileri belirlenir ve kodlanır. Her ayrıt noktasında model parametrelerinin, (W_R ,W_L, C_R, C_L), saklanması yerine bir blok pencere içindeki değerlerine değişken derecede polinomlar uydurulmuştur. Bu polinomların katsayıları eşiklendirilerek kodlanır.

Uzunluk, çevrit boyunca benek değerlerinin standart sapması, ortalama kontrast, ortalama eğrilik gibi özelliklerine göre sıralamaya sokulan ayrıtların eşiklenmesi ile önemli nitelikler korunarak sıkıştırma oranı arttırılabilir. Bu özelliklere verilen ağırlıklara göre sıralanan ayrıtların %50-85'inin kullanılması durumunda dahi yüksek kalitede görüntüler ve 10:1-80:1'lik sıkıştırma oranları elde edilmiştir.



(a)

(b)

Şekil-1 (a) Ayrıt Kesiti,
(b) Ev görüntüsü üzerine bindirilmiş model.

Seyrek ayrıt modeli parametrelerinden gerçek görüntünün bir yaklaşığı karma enerji fonksiyonelinin en aza indirgenmesi ile elde edilmiştir. Karma enerji fonksiyoneli üç bileşenden oluşmaktadır :

$$\mathbf{E} = \mathbf{D} + \mathbf{Z} + \mathbf{L}$$

$$\mathbf{D} = \sum_{i,j \in R} \sum \beta_{i,j} (f_{i,j} - d_{i,j})^2$$

$$\mathbf{Z} = \sum_{i,j \in R} \sum \lambda (1-\tau) \cdot [(f_{i,j} - f_{i-1,j})^2 + (f_{i,j} - f_{i,j-1})^2]$$

$$\mathbf{L} = \sum_{i,j \in R} \sum \lambda\tau \cdot [(f_{i+1,j} + f_{i-1,j} - 2 \cdot f_{i,j})^2 + (f_{i,j+1} + f_{i,j-1} - 2 \cdot f_{i,j})^2]$$

Burada **D** kurulan yüzey ile seyrek veri arasındaki farkın enerjisini, **Z** seyrek veri üzerinden geçirilen zarın enerjisini, **L** levhanın enerjisini, f kurulmak istenen yüzeyi, d seyrek veriyi göstermektedir. Bu fonksiyonel SOR (Successive Over-Relaxation) algoritması kullanılarak en aza indirilmiştir :

$$f_{i,j}^{(n+1)} = f_{i,j}^{(n)} - \frac{w}{T} \frac{\partial E(f)}{\partial f_{i,j}}$$

$$\begin{aligned} \frac{\partial E(f)}{\partial f_{i,j}} = & (\beta_{i,j} + 4\lambda(1+4\tau)) f_{i,j}^{(n)} - \lambda(1+7\tau)[f_{i-1,j}^{(n+1)} + f_{i,j-1}^{(n+1)} + f_{i+1,j}^{(n)} + f_{i,j+1}^{(n)}] \\ & + 2\lambda\tau[f_{i-1,j-1}^{(n+1)} + f_{i-1,j+1}^{(n+1)} + f_{i+1,j-1}^{(n)} + f_{i+1,j+1}^{(n)}] \\ & + \lambda\tau[f_{i-2,j}^{(n+1)} + f_{i,j-2}^{(n+1)} + f_{i+2,j}^{(n)} + f_{i,j+2}^{(n)}] \\ & - \beta_{i,j} d_{i,j} \end{aligned}$$

Bu iterasyonlar sırasında model parametreleri ile oluşturulan seyrek veri güncellenmemektedir. Iterasyonların sonlanma koşulu birbirini izleyen iki iterasyon çözümü arasındaki farkın önceden belirlenen bir değerden küçük olmasıdır.

İkinci bölümde sıkıştırma yöntemlerine ilişkin basit bir sınıflandırılma verilmekte ve mevcut sıkıştırma yöntemleri kısaca tanıtılmaktadır. Bu arada modle tabanlı sıkıştırma yöntemleri detaylı olarak incelenmektedir. Bu bölümün sonunda kırkayak modeli tanıtılmaktadır.

Üçünde bölümde, ayrıt-tabanlı görüntü kodlama yöntemlerinin önemli bir parçasını oluşturan ayrıt saptama, çevrit izleme, çevrit eleme ve kodlama yöntemleri için çözümler sunulmaktadır.

Dördüncü bölümde kırkayak model parametrelerinin nasıl çıkarıldığı ve bu parametrelere nasıl polinom uydurulduğu tanıtılmaktadır. Ayrıca polinom derecesi ve blok boyu ile kurulan görüntülerin kalitesinin nasıl etkilendiği incelenmiştir. Bölümün sonunda kırkayak modeline dayalı görüntü sıkıştırma yöntemi ile elde sonuçlar sunulmaktadır.

CHAPTER 1

INTRODUCTION

As images are getting larger in size, storage and transmission of them take much more storage space and transmission time on communication mediums, necessitating compression of images in many applications of computer vision ranging from medical to video phone. Image coding aims to reduce as much as possible the number of bits necessary to represent the image while preserving the quality and intelligibility required for the given application. The level quality and intelligibility required vary widely, depending on the application. In some applications like medical imaging, it is important that reconstructed image is the exact duplicate of the original. Such image-coding technique that preserves all information in the image and allows exact reconstruction of the original one is said to be information-preserving. Some applications such as video phone, where the exact reconstruction is not the primal goal, do not require information-preserving methods. They allow small amount of degradation on the reconstructed image. Such techniques are called information-lossy. For information-lossy techniques, it is important to describe the amount and the type of degradation at the reconstructed image [1]. Unfortunately, there is no subjective way of measuring the degradation introduced by the method. When the visual system is human, then it seems that best test measure is human brain. In other case, when the visual system is another visual machine system, the amount of degradation, deteriorating the task performed by machine vision system, is measured by reduction in performance and requires extensive examination on the system.

A typical image coding environment is shown in Figure 1.1. Coding has two parts as depicted in the figure. Purpose of the first part, called source coder, is to encode the digital data such that encoded data occupies less space. Second part, called channel coder, transforms the bit streams encoded by source coder into a form more suitable for transmission over a communication channel through appending some error detection/correction bits. At the receiver, transmitted image is reconstructed by image coder. In this study, we are interested in only source coding and decoding.

Image coding techniques can be classified into *three* groups according to what they actually code. In waveform image-coding techniques an image itself or some simple variation of that such as difference in intensity at consecutive pixels is coded. In transform-based image-coding techniques, an image is transformed into another domain such as frequency domain and transform coefficients are coded. In model-based image-coding techniques, an image is modeled and model parameters are coded. Waveform image coding techniques has used information theory end its results.

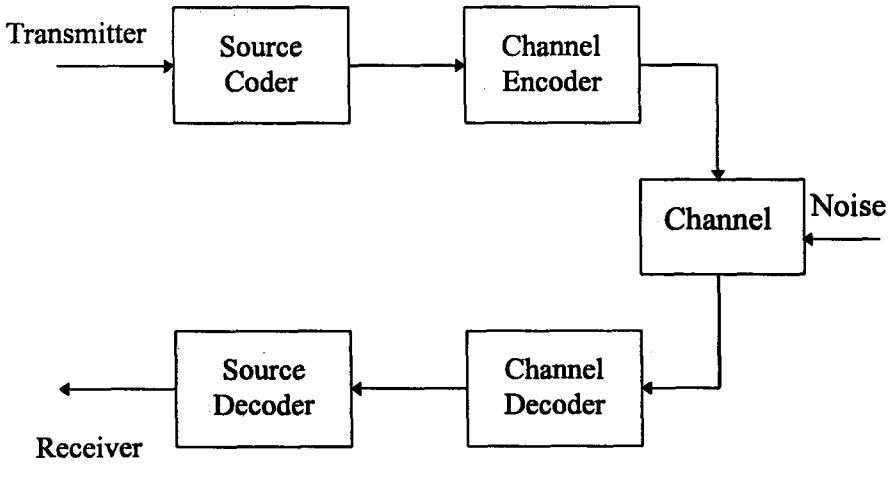


Figure 1.1 Typical environment for Image Coding

The entropy has been thought to be the upper bound of the compression ratio of any entropy coding methods such as Huffman coding and Arithmetic coding can succeed.

$$H(u)=\sum_{i=1}^{N} P(u=u_i)\cdot\log_2(\frac{1}{P(u=u_i)})$$

It is clear that images are not random and locally homogeneous. That means gray levels are similar over a small neighbor. First-order entropy is defined as

$$H(u_k|u_{k-1})=\sum_{i=1}^{N}\sum_{j=1}^{N} P(u=u_i,u=u_j)\cdot\log_2(\frac{1}{P(u=u_i|u=u_j)})$$

where $P(u=u_i|u=u_j)$ is the conditional probability. This is considered as the average information content of u_k in the case that u_{k-1} is known. Second and higher order entropy can be defined similarly. It can be shown that Huffman coding for higher order entropy approaches to the first-order entropy irregularly.

Waveform coding methods based on the decorrelation of images followed by quantization and entropy coding (Huffman Coding). Decorrelation is obtained by prediction (i.e. Differential PCM). Compression ratio is about 4:1 for waveform coding techniques.

In transform coding methods, an image is transformed into such a domain which is much more appropriate for coding. The transforms have the common property called *energy compaction* which states that transformation coefficients are distributed in a small fraction of the supporting domain. Compression ratio is about 20:1 for transform-based coding.

Actually, entropy of the image is unknown and depends mostly on the model used for the source. Third class of image-coding techniques are *model-based* approaches to the image coding problem. Model parameters are extracted and coded, at the receiver side reconstruction is obtained by using these model parameters. There are two groups of approaches in the class. One group of method makes use of local operators, convolving the image with impulse response of 2-D filters or filter banks. These filters or filter banks are designed so as to extract local feature. Then these local features are combined to obtain the messages to be coded. Nonstationary predictive coding is an example of this group of methods. Second approach uses completely different method. They utilize edges and textures to model the image. First step in this method is to detect the edges or segment the image into regions. Contour or region boundary locations, brightness at edges or regions and contrast are coded. Reconstruction of the original image is obtained by solving a *diffusion* (*heat*) equation. Compression ratio for model-based coding is dependent heavily on the model and model precision.

In this study, a model-based image coding method called "centipede model" is developed for compression of images using primitives based on edge location, intensity, contrast and scale. Edges, correspond to object boundaries where sharp changes occur due to some physical aspect of an image such as surface reflectance or illumination, are detected by using Generalized Edge Detector (GED). GED [2] introduces a $\lambda\tau$–space representation of images and consequently edges. On the $\lambda\tau$–plane, a point associate to two filters denoted by $R(\lambda,\tau)$ and its first-order derivative $G(\lambda,\tau)$ whose shapes are determined by τ and scales are determined by λ. One can obtain well-known edge filters by setting these parameters appropriately. Then edges are traced by following adjacent edge elements to produce distinct contours. Intensities on edges and contrasts are modeled by polynomials. We have shown that using these parameters and minimizing hybrid energy functional do not yield a powerful representation of the original image. It results in a blurry reconstructed image with many artifacts. We utilize edge scales in the form of widths to overcome

the problem. Width is defined as the distance at which the difference in intensity along the normal direction is lower than a given threshold.

Coding of all these parameters is a bit-consuming operation. Since they change smoothly in a small neighborhood, they can be approximated well by polynomials, instead of model parameters, the coefficients of the polynomials are coded. Edge locations are coded by constructing differential chain code followed by Huffman coding and end points are coded in the form of difference between lexicographically ordered points. A reliable approximation to the original image from the sparse information is obtained by solving the hybrid energy functional which constructs $\lambda\tau$-space, where λ represents smoothness of the image and τ represents the continuity of the image.

The thesis is organized as the following. In chapter two, a classification for image coding methods is presented based on what they code and particularly, existing model-based approaches are investigated. "Centipede Model" is explained at the end of this chapter. Chapter three develops edge detection, contour tracing, filtering and coding methods. Chapter four goes on the extraction of the centipede model (width and contrast). Since the parameters are modeled by polynomials, curve fitting with polynomials is introduced by minimizing MSE. Quantization and coding of polynomial coefficients are given at the end of the chapter. Experimental results are discussed and conclusions are drawn in chapter five.

CHAPTER 2

IMAGE CODING TECHNIQUES

A digital image could be viewed as a matrix of dimension $N \times M$,which has $N \times M$ - byte representation. Typical value for N is 256 or it may have higher spatial resolution up to 1024 for medical images. When color images are considered, the required storage is tripled. Well-known statement, "A picture is worth a thousand words" by Descartes, appears in many books on image processing and computer vision to express the observation that human being percept the images in a highly intelligent way, unfortunately images require much more storage space than it sounds. Image coding has focused on this problem. Though many image-coding methods for image compression have been proposed since late fifties, the main objective of all is to provide the best possible quality image for the minimum data rate. This chapter is devoted to review these efforts in image compression. Particularly, model-based image coding techniques will be investigated.

2.1 Introduction

Image coding techniques can be classified into *three* groups according to what they actually code. In waveform image-coding techniques an image itself or some simple variation of that such as difference in intensity at consecutive pixels is coded. In transform-based image-coding techniques, an image is transformed into another domain such as frequency domain and transform coefficients are coded. In model-based image-coding techniques, an image is modeled and model parameters are

based image-coding techniques, an image is modeled and model parameters are coded. Waveform image coding techniques has used information theory end its results.

2.2 Waveform Image Coding

Waveform image-coding systems consist of three elements. The first and most important element is the transformation of the image to the most suitable domain for quantization and code-word assignment. In waveform coding, we code the image intensity itself or some simple variation of image brightness such as the difference between consecutive pixel brightness. One major advantage of waveform coding is its simplicity. Since the waveform itself is coded, the coders are very simple both conceptually and computationally. Waveform coders do not generally perform as well as transform coders.

2.2.1 Pulse-Code Modulation (PCM)

The simplest waveform coding method is the basic pulse code modulation system, in which the image intensity is quantized by a uniform or a non-uniform quantizer. PCM systems introduce a noise to the image which can be modeled as additive random noise. A way of improving the performance of a PCM system is to remove the signal dependence of the quantization noise, which appears as false contours at low bit rates. Roberts`s psedonoise technique, also known as dithering, is a method that removes the signal dependence of the quantization noise. In this method, a known random noise is added to the original image before the quantization at the transmitter and then the same random noise is subtracted at the receiver. Since random noise is known at both the receiver and the transmitter prior to image transmission, it does not have to be transmitted. The compression ratio is about 3:1.

2.2.2 Delta Modulation (DM)

In the PCM system, the image intensity is coded bit scalar quantization, and the correlation among pixel intensities is not exploited. One way of exploiting some of the correlation is delta modulation. In the DM system, the difference between two consecutive pixel intensities is coded by a one-bit quantizer. Although the dynamic range of the difference signal is doubled as a result of differentiation, the variance of the difference signal is significantly reduced due to the strong correlation typically present in the intensities of two pixels that are spatially close.

An important design parameter in DM is the step size. In the region where the signal varies slowly, the reconstructed signal varies rapidly around the original signal. This is called *granular noise*. A large step size results in a correspondingly large amount of granular noise. When the signal increases or decreases rapidly, it may take many pixels to catch up original signal using small step size. The reconstructed signal will appear blurred in such regions. This is called *slope overload distortion*. The compression ratio is about 4:1.

2.2.3 Differential Pulse Code Modulation (DPCM)

Differential pulse code modulation (DPCM) can be viewed as a generalization of DM. In DM, the difference signal is quantized by a one-bit quantizer. In DPCM, more than one bit can be used in coding the error. Since a PCM is a component of a DPCM system, it is possible to use Robert's pseudonoise technique in a DPCM system. Since we reduce the number of bits available to encode each pixel, the quantization noise will be less if we use DPCM rather than PCM at the same rate. The compression ratio is between 2:1 and 3.5:1 depending on the image statistics.

This approach can easily be extended to the two dimensions. Two dimensional DPCM performs better than PCM and one dimensional DPCM by about 3.75:1.

2.2.4 Predictive Coding Techniques

There exists a statistical dependence between gray values at consecutive pixels. Previous transmitted signals convey some sort of information about oncoming signals. Prediction techniques are used to exploit the dependency and prediction sequence is defined as the error between estimated and actual signals. Now, prediction error signal is quantized instead of transmitted signal. Image compression ability depends on the prediction technique used and correlation exists among neighboring pixels. If all individual signals are mutually independent, then there is no advantage over PCM or DPCM. Prediction image coding assumes some amount of dependency and small variance of prediction error sequence.

2.3 Transform Coding

In transform image coding, an image is transformed to a domain significantly different from the image intensity domain, and the transform coefficients are then coded. In low bit rate applications, transform coding techniques with scalar quantization typically perform significantly better than waveform coding techniques with scalar quantization. However they are more expensive computationally.

Transform coding is significantly different from DPCM and achieves compression in transform domain. Transform coding techniques attempt to reduce the correlation that exists among image pixel intensities more fully than do waveform coding techniques. Transform coding techniques also exploits the observation that for typical images a large amount of energy is concentrated in a small fraction of the transform coefficients. This is called the *energy compaction* property. Because of this property, it is possible to code only a fraction of the transform coefficients without seriously affecting the image quality.

The statistically optimal linear block transform, in the sense that it minimizes the mean squared reconstruction error, for coding images is well known to be the

Karhunen-Loeve transformation (KLT) [1]. The KLT is related to principal component analysis, since the basis vectors are also the principal components of data. Because the KLT is an orthonormal transformation, its inverse is simply its transpose.

$$\bar{y} = W\bar{x}$$
$$\sum = E[\bar{x}\bar{x}^T]$$

A number of practical difficulties exist when trying to implement the KLT. The calculation of the covariance estimate, Σ, and its eigendecomposition is not practical even with today's computing resources. The algorithms used to do these computations are complex and therefore not suitable for hardware implementation. The calculation of the covariance estimate requires $O(N^2)$ calculations. Due to these difficulties, fixed basis transforms such as discrete cosine transform (DCT) [2], which can be computed in order $O(N \log N)$, are typically used.

2.3.1 Hybrid Coding

The term refers to techniques that combine transform coding which performs very well in low bit rate applications and waveform coding which is very simple to implement. In hybrid coding, a two-dimensional image or its row or column is transformed to obtain statistically independent sequence of transform coefficients accumulating in a narrow energy band known as energy compaction property and then this sequence is coded by waveform coders such as DPCM. Hybrid coding can achieve the compression ratio up to 8.0:1.

Hybrid coding of a single image is not practical since it does not decorrelate the image as much as 2D transform coding and implementation complexity of transform coding is not much more than hybrid coders. Hybrid coding is useful in interframe

image coding. Exploiting the temporal correlation as well as spatial correlation to code a sequence is called *interframe coding*. In intraframe hybrid coding, every frame is computed by 2D transform and then waveform coding is applied to 2D transform coefficients along temporal direction.

2.3.2 Adaptive Coding and Vector Quantization (VQ)

Transform coding can be made adaptive to the local characteristics within a block of particular size. For example, transform block size can be chosen small in regions containing edges. Adaptive coding significantly improves the performance of transform coding while adding little to its complexity.

Transform coefficients are generally coded with scalar quantizers. It is also possible to use vector quantizers in which designing goal is to obtain a quantizer consisting of N reproduction vectors, such that it minimizes the expected distortion rate.

Vector quantization is the joint quantization of the components of a vector. Unlike scalar quantization, it is often used to requantize signals that are already digital, for the purpose of compression. When vector quantization is used for image compression, the image is partitioned into blocks of $N = n \times n$ pixels, which form an N-dimensional vector. This vector is encoded by searching a codebook of representative quantization vectors. Let $x = [x_1,...,x_N]^T$ denote a vector that is formed of a block of size N of the image. In VQ, x is mapped into another N-dimensional vector $q = [q_1,...,q_N]^T$. The vector q is chosen from L possible quantization codewords in q way that minimizes an Euclidean distance measure. The quantization codewords are chosen so as to be optimal for a given distribution. The techniques for choosing the quantization symbols and the different search algorithms for the encoding stage have been reviewed and detailed in [4]. One of the most popular VQ algorithms makes use of the K-means algorithm and is known as the LBG algorithm [5]. Given the number of clusters to be formed, it iteratively refines the cluster

centers and boundaries. The iteration is stopped when the mean square error falls below a threshold or remains constant between iterations.

2.3.3 Two-Channel Coders

In a two-channel coder [6], an image, $f(n_1,n_2)$, is divided into two components: low and high frequency components. The lows component $f_L(n_1,n_2)$ consists of low-frequency components and represents the local luminance mean of $f(n_1,n_2)$. The high component $f_H(n_1,n_2)$ consists of high-frequency components and represents the local contrast of $f(n_1,n_2)$. Since low component is a low-pass filtered version of $f(n_1,n_2)$, it can be downsampled and interpolated by low-order polynomials or splines. High components can be coarsely quantized by a PCM system which may use Robert's pseudorandom noise technique.

Two-channel coding system is shown in Figure 2.1. The original image $f(n_1,n_2)$ is low-pass filtered by a FIR filter. Since lows component is smooth enough to be represented by a polynomial at coarser levels, the lows component $f_L(n_1,n_2)$ is subsampled and estimated by a polynomial. The high component is obtained by subtracting the $f_L(n_1,n_2)$ from $f(n_1,n_2)$ and quantized by a PCM.

A two-channel coding system can be viewed as a special case of a subband image coder. A coding method in which a signal is divided into many channels and each channel is then coded with own coder is called *subband image coding* [7].

2.3.4 Fractal Image Compression

A fractal is considered as a function, F, having the following properties:

1. F has detail at every scale,
2. F is self-similar,
3. The fractal dimension of F is greater than its topological dimension.
4. There is a simple algorithmic definition of F.

Fractal image coding [8] is based on the assumption that images are exactly or at least statistically self-similar. By using self-similarity, some regions called *domains* are transformed into such regions called *range* that means square error between ranges and transformed domain is minimum. Then the transformation coefficients are coded. Reconstruction is done by iteratively applying the transform onto itself. This is called *iterated function system*. Due to the property (1), iterations result in an image called *attractor* and final image, attractor, is not affected by initial image. Different transformations lead to different attractors. The only limitation on the selection of transformation is that the transformation must be contractive.In practice, the transformation is of the form

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

called *affine* transformation. Each affine transformation is defined by six numbers $a_i, b_i, c_i, d_i, e_i, f_i$.

Performance of fractal image compression is restricted with the self similarity of the image. Real images are not exactly self-similar which makes the compression ratio lowered.

2.3.5 Pyramid Coding

A pyramid is a pyramid-like grid structure where bottom of the pyramid corresponds to finest scale and top of the pyramid corresponds to coarsest grid. An image is described in the scale space on the pyramid from finer scales to coarser by convolving a scale-space filter like gaussian function or solving diffusion equation in time. The difference image between two consecutive levels is called *residual*. Since residuals are smooth signals, they can be represented finely at one coarser level by subsampling and interpolating. This process is iterated at every level of the pyramid.

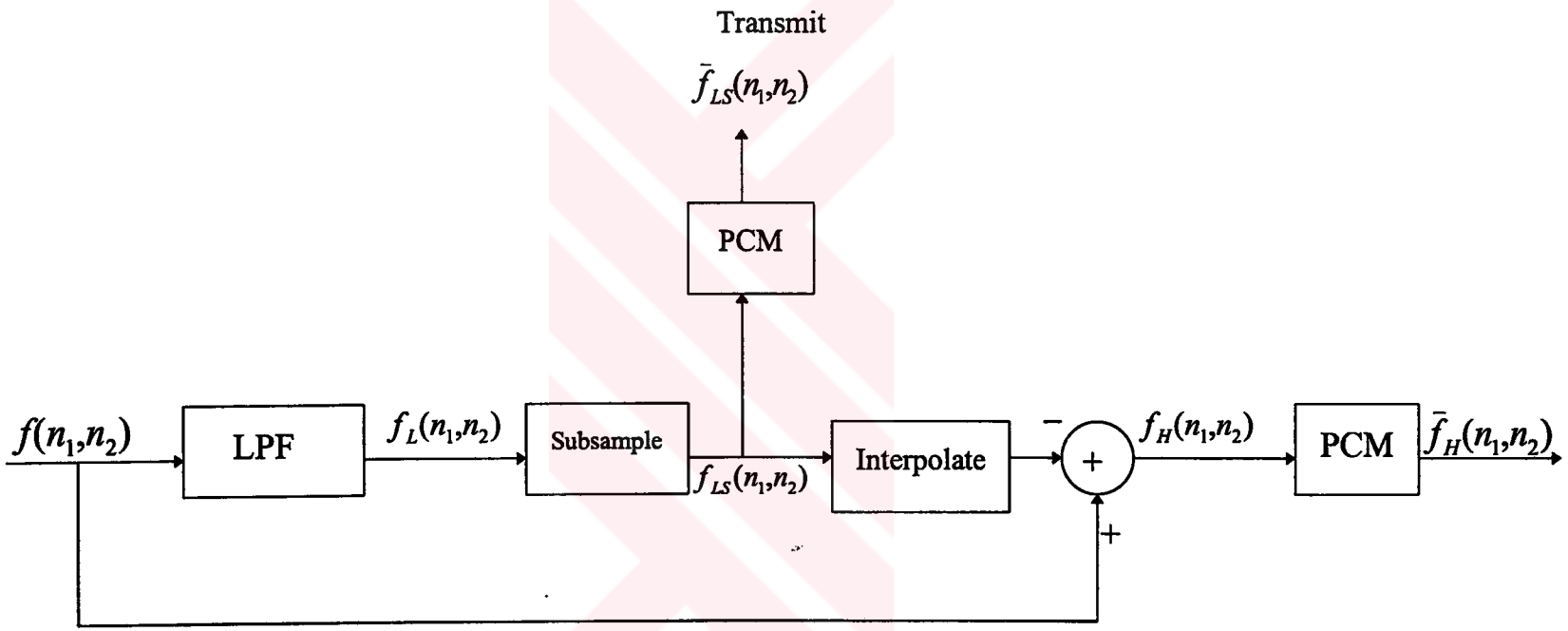


Figure 2.1 Two-channel image coding system

The image at the coarsest level is the blurry version the original. Each residual is coded by DPCM and transmitted. At the receiver, residuals from coarsest level to finest level are interpolated and added to the previous received residual. As the process is repeated for each level on the receiver, the reconstructed image will become higher spatial resolution image. One major advantage of the pyramid is its ability to progressive image transmission.

2.3.6 Wavelet Image Coding

A wavelet transform is the decomposition of a signal into a set of basis functions consisting of contractions, expansions and translations of a mother function $\psi(t)$ called the wavelet [11,12]. It is equivalent to the analysis of a signal into several frequency bands having the same bandwidth on a logarithmic scale, each one representing a tradeoff between time and frequency resolution :

$$\text{Time - Bandwidth Product} = \Delta t \times \Delta f \geq \frac{1}{4\pi}$$

This is referred to as the uncertainty principle, or Heisenberg inequality. It means that one can only trade time resolution, or vice versa. The wavelet transform represents a different interpretation of subband coding. In this interpretation, other features of the filter banks than frequency selectivity, such as regularity and number of vanishing moments, are considered [13,14].

2.4 Model-based Image Coding Techniques

Common feature of all coding methods is to exploit the redundancies exist in images by decorrelating the image. Since techniques in first and second class explained above make use of information theory and its results, decorrelation is based on the entropy and entropy coding. Information theory suggests that upper level of coding gain is bounded by the entropy for a given signal and its distribution. Fortunately, the entropy of an image is unknown and depends mostly on the model used. Model-based image coding techniques [15,16] , contrast to previous class methods, make use of some features of images such as edges, segments, textures. This is based on the fact that an image can be described by its discontinuities where sharp changes occur in brightness and by regions where brightness is equally distributed. In fact, most of the computer vision high level processes are partially based on the edges, regions and texture information. Human brain also processes the visual information in the form of edges and regions.

The first study which inspired such methods classified as model-based techniques is Graham`s paper [17] on edge-based compression and actually goes back to late sixties. There are two reasons why edge-based techniques did not take so much interest of researchers as it deserves. First, from sixties to late seventies information theory was on the way to develop and, information theory tools were considered as the optimal way of coding. Second, there were no study on coding of arbitrary curves effectively. In fact, in middle of seventies Freeman [18] put forward a method called chain code for curve representation. Later, Kaneko and Okudaira [19] developed chain code representation by utilizing edge link concept. Schreiber [6] considered an image as being composed of low and high frequency parts, which are encoded separately. The high frequency part corresponds to contour and low frequency part corresponds to regions. Kunt et al. [15] use region growing to segment the image into regions and gray level intensity is modeled and encoded by polynomials. Region-based methods suffer from the artifacts that the transitions at the region borders are step edges. Kunt offered a solution that tries to model the edge profile by wavelets.

Carlson [20] presented a sketch-based coding scheme for gray level images. Carlson proposed to code the intensity on either side of each edge, and interpolate image intensity between contours by diffusing intensity estimates from these edges. The results were not good in perceptual quality. There are many reasons for the distortions on the reconstructed image. Distortion is mostly caused by the incomplete information on data. Recently, Elder and Zucker [21] have proposed an algorithm for reliable edge detection and blur estimation. Scale is estimated only on edges and approximated elsewhere by diffusing them. The reconstructed image is obtained by solving anisotropic heat equation with space-varying scale parameter. The results are perceptually very close to the original images. But they did not mention how to code the scale estimates at edges, intensities and contrast information.

Acar and Gençata [22,23] used a model called weak membrane. They formulated the edge detection and surface reconstruction problems as a regularization problem with a non-convex energy functional. They proposed to code the intensity on either side of each edge in the image modeled by weak membrane. Since the image is modeled by weak membrane, the resultant image is far from being perfect.

Gökmen and Ersoy [24] has recently developed a model-based coding method for fingerprint images. They use the regular structure of the fingerprint images and hybrid model. They proposed to detect ridges and valleys by adaptive thresholding and to code the intensities on ridges and valleys. Reconstruction was done by minimizing the hybrid energy functional. One advantage of the model is its ability of preserving the fingerprint structure in high compression ratios. Because, what they code is fingerprint structure itself. Another advantage is that one can process even on the compressed data without reconstruction for the recognition or classification purpose. The method suffers from two problems. First, coding and decoding processes require heavy computation load. Second, the reconstructed image is mostly blurry version of the original fingerprint.

J. Rabinson [26] presented a different model which uses ridges and valleys as perceptual image primitives instead of edges and regions. The primitive curves (ridges and valleys) are called "threads". Transmitted data are the thread locations, shapes and profiles. Reconstruction is done by interpolating the transmitted data. The image is first filtered with a valley/ridge detection operator such as a Gaussian-smoothed second derivative. He reported that LOG with Gaussian variance of 1.4 pixels had been used. The threads were obtained by a valley/ridge follower. Location of valley/ridge was coded by standard chain code. Intensities and profiles on valleys/ridges were compressed by fractal coding. The unknown pixels are interpolated by C^0 Natural Neighbor Interpolation [27]. Since the method is based on ridge/valley primitives, the approach preserves texture rather than edges. The model has a drawback that since the valleys/ridges are mostly discontinues, the interpolation causes the reconstructed image to blur at the discontinues.

U. Y. Desai et al. [28] has recently proposed an edge and mean-based compression method for color images. They used Sobel operator to detect edges. Contour intensity was coded by line-fitting in one dimension for both vertical and horizontal directions. In order to enhance the reconstructed image quality while keeping the bit rate down, mean of the image in a block of size 10x10 was also coded. In decoding stage, the mean in a block is used to modify the mean of the interpolated image. They reported acceptable quality images at 0.1 to 0.3 bpp for (256x256) color images.

Dijk and Martens [29] presented a method combining the transform coding and model-based coding. They introduced steered Hermite Transform described as a weighted sum of orthogonal polynomials. The term "steerable" is used to describe a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of basis filters. Hermite filters of order n form a steerable basis for every individual filter of order n. It had been shown that Hermite transform has the advantage that high energy compaction can be achieved by adaptively steering the transform. They used Gaussian edge model and expressed explicitly the local

edge parameters as a function of the Hermite Transform coefficients. Unfortunately, they did not deal with the problem of compressing these parameters.

Salembier et al. [30] has presented morphologic tools to compress images. They defined four morphologic operators rather than complete coding scheme. These operators are as the following :

- Connected Operators [31] : The operator solves the problem of image simplification while preserving the contour information.
- Region-growing version of the watershed [32] : The watershed transformation is the classical morphological tool for segmentation
- Geodesic skeleton [33] : In region-based image coding, there is a problem of assigning a contour to a region while the contour belongs to at least two different region.
- Morphological interpolation : Morphological interpolation can be used very efficiently to interpolate on irregular grids.

There are many studies on region-based image coding [34]. In [35], an image is partitioned into distinct regions by using a segmentation algorithm, the contents of the regions are then coded using polynomials. The effect of polynomial order was studied by comparing the segmentation and rate-distortion performance produced by different order approximations. In contrast to the previous studies on region-based coding, texture on each region was modeled.

2.5 Centipede Model

We have seen the following problems with the previous edge-based coding techniques

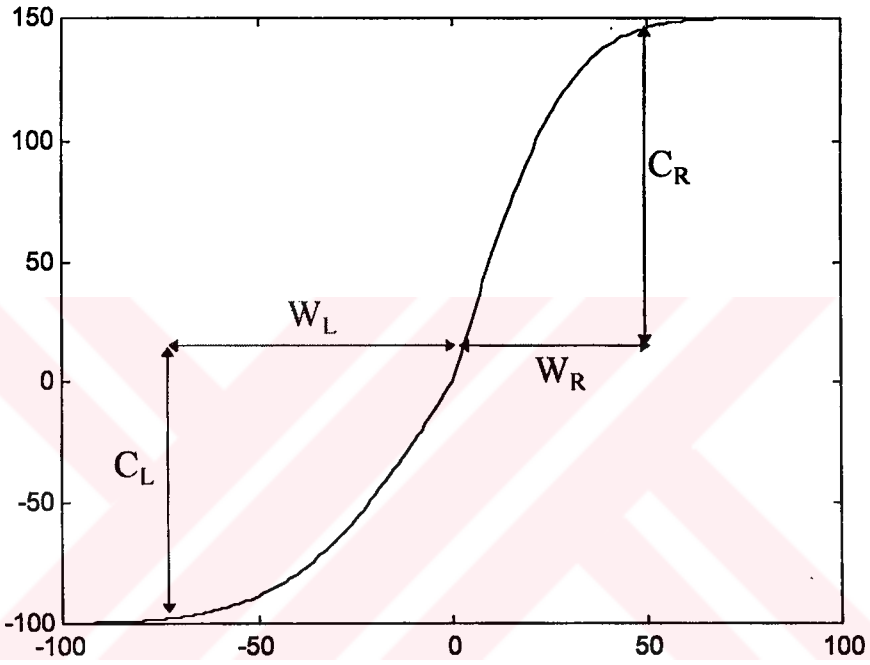
1) Edges are generally modeled by symmetric Gaussian (or similar) functions, causes that reconstructed image looking like artificial.

2) Since they do not attempt to control the selection of contours or the selection is done randomly, some contextually important features may disappear.

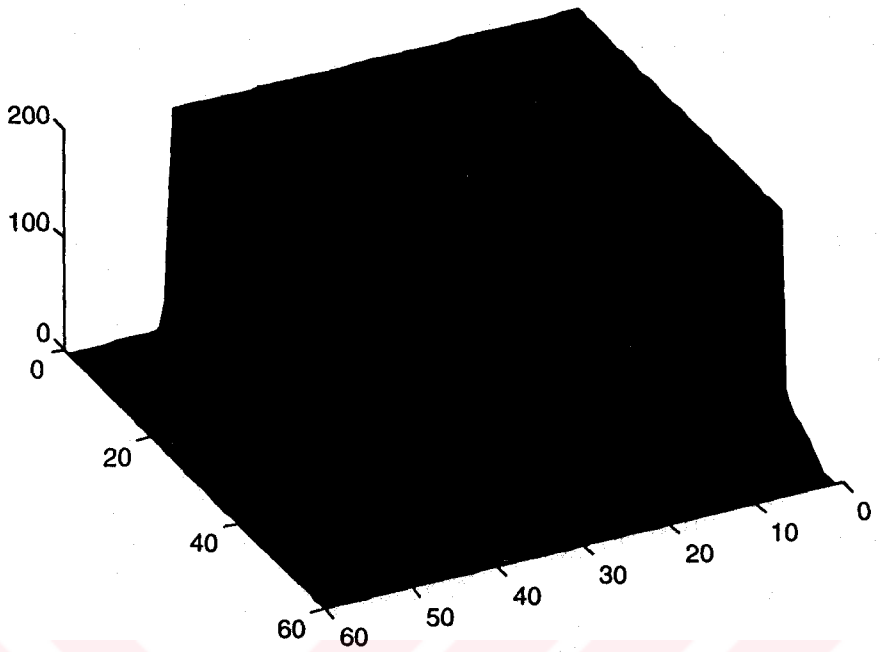
3) None of the techniques develops a unified approach for edge detection, contour selection and coding, edge representation, and reconstruction.

We have developed a model which we call "Centipede Model". Centipede model [25] consists of five parameters : Intensity on edge (I_L), Left and Right Contrast (C_L, C_R) and Left and Right Width (W_L, W_R). The model parameters are drawn on an edge profile given in Figure 2.2 (a). The model parameters are extracted on each edge element on the contour shown in Figure 2.2 (c). Two-dimensional edge profile is given in Figure 2.2 (b). Width is defined as the distance at which difference in consecutive pixel is lower than a given threshold. Threshold is determined from the SNR (dB)ratio for an image. Width is a rough estimate of edge scale. In chapter 4, it has been shown that the tuple (I_L,C_L, C_R, W_L, W_R) with the edge map constructs a powerful representation of the image (solve the problem (1)). How the parameters are extracted and coded are explained in chapter 4. We present an approximation scheme for the model parameters by polynomials with varying order and constant block length. We also investigate how the selection of polynomial order and block length effect the quality of the reconstructed image.

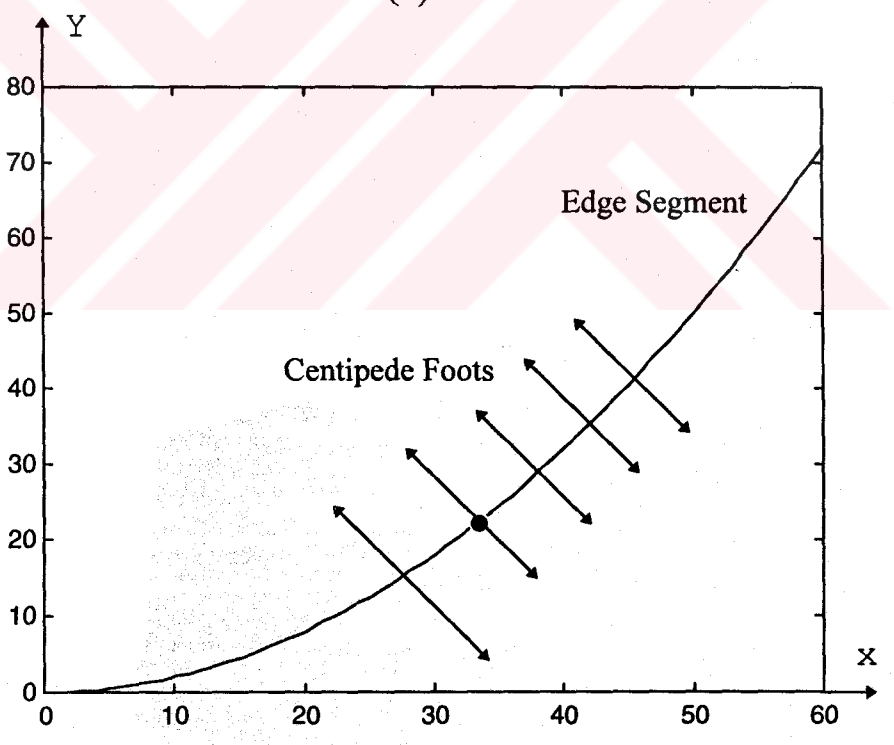
We propose a contour selection method based on the ordering of edge segments regarding to normalized feature set containing length, contrast and curvature in chapter 3. The method is experimentally tested and verified that for high compression ratios all perceptually most important features are still kept (solve the problem (2)).



(a)



(b)



(c)

Figure 2.2

(a) Centipede Model Parameters for the edge on the contour shown in (c),
(b) 2D Edge Profile,
(c) Contour.

CHAPTER 3

CONTOUR CODING

This chapter develops contour extraction, selection and coding methods used in contour-based image coding system that we have developed in this study. First, how edge information is extracted from an image via generalized edge detector is explained. Then, we proceed with the edge following algorithm which produces long and smooth segments as much as possible. In order to increase the compression ratio while preserving the quality of the image in terms of visual appearance to human viewers and allowing small amount of degradation, some of edge segments which have less significance on surface reconstruction process are eliminated for a given threshold. Edge segments are ordered according to normalized weighted sum of their length, mean contrast and mean curvature. We have coded these contours by using *differential chain code* followed by Huffman coding. End points are coded in the form of distance between lexographicly ordered end points.

3.1 Introduction

Contours correspond to object boundaries where sharp changes occur due to some physical aspect of an image such as surface reflectance or illumination. It is well known that an image which consists of only its edge is highly intelligible and structural. Studies on human brain and low-level retinal processes have showed that

brain also receives information in the form of edges and brightness. In fact, in computer vision, intermediate and high-level processes such as shape from shading, structure from motion are partly based on edge information and so extracting the edges highly accurately has primal importance for any algorithm in machine vision. On the other hand, for compression purpose, contours have another important property that they are very sparse.

An edge detector extracting and locating object boundaries in an image from intensity data is a crucial step of contour-based coding system. The goal of edge detection is to obtain powerful and complete description from an image by characterizing these intensity changes. By using this representation, it would be possible to reconstruct a good replica of the original. Since accuracy of model parameters (contrast, intensity and width)[1] are determined by the accuracy of detected edges, edge detection is the most important part of the algorithm. An edge detector is expected to have a good detection, good localization, one response to one edge, robustness, efficiency, applicability to sparse data properties.

We have used generalized edge detector which allows the description of images on a plane called $\lambda\tau$–space where τ controls the shape of the filter and λ controls the scale of the filter. One can obtain most of the well-known edge detectors by setting these two parameters appropriately.

3.2 Generalized Edge Detector (GED)

Generalized edge detector [2] is based on the regularization theory and convolution with filters. In regularization theory, the smoothness constraint is imposed to the solution in by means of energy functional containing derivatives of the solution. GED uses hybrid energy functional :

[1] Definition of these parameters and how they are extracted from an image will be explained in Chapter 4

$$E(f) = \iint (f(x,y) - d(x,y))^2 + \lambda\,[(1-\tau)(f_x^2 + f_y^2) + \tau(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)]\,dxdy \quad (3\text{-}1)$$

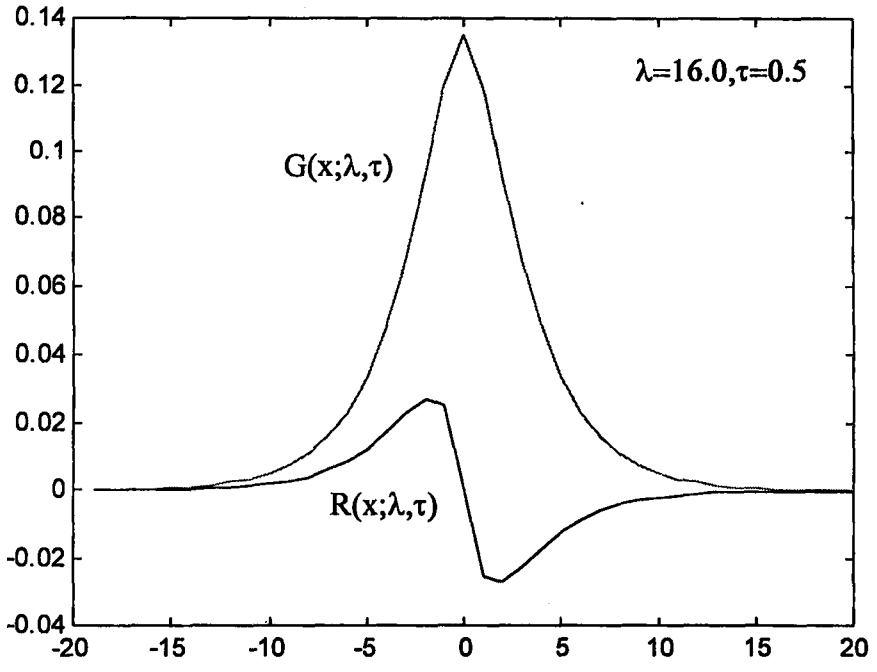
In the functional, first term on the right hand side is a measure of the closeness of the solution *f(x,y)* to the data *d(x,y)*, and the second and the third terms are stabilizers on the solution as the first and second order derivatives. The Euler-Lagrange equation associated with this hybrid functional is

$$\lambda\tau\ [f_{xxxx} + 2f_{xxyy} + f_{yyyy}] - \lambda(1-\tau)\ [f_{xx} + f_{yy}] + f = d \quad (3\text{-}2)$$

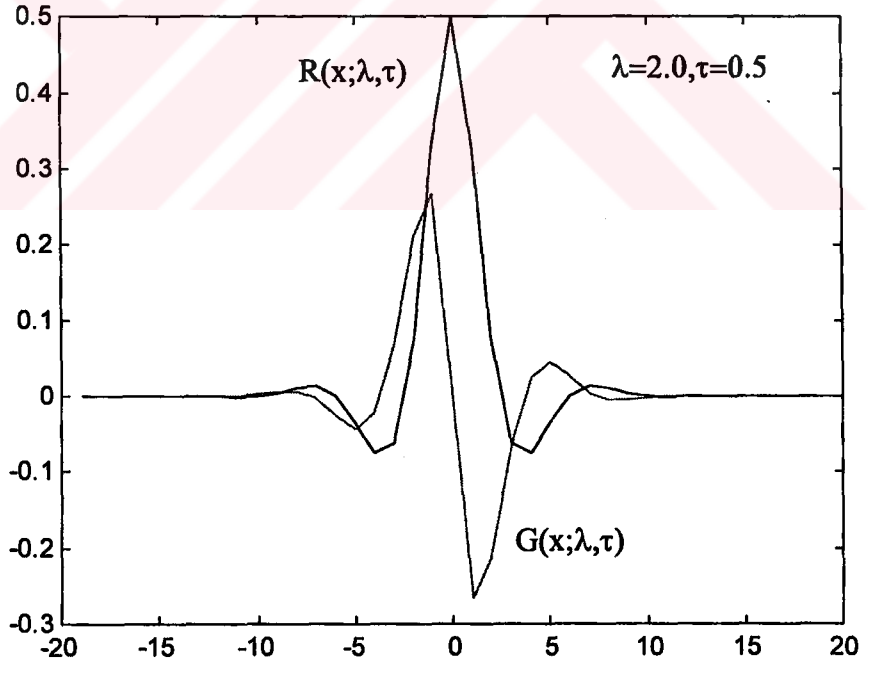
This is a forth-order partial differential equation. Solution of (1-2) is explained in details by using Green function in [2]. The solution gives five different form of filters denoted by G(x;λ,τ) and R(x;λ,τ) according to $\Delta = B^2 - 4A$, B and A where $B = \lambda\,(1-\tau)$ and $A = \lambda\,\tau$. These filters, G(x,λ,τ) and R(x;λ,τ), are drawn in Figure-3.1. Images are convolved with the filter G(x,λ,τ) and then edges are obtained from convolved signals by applying histerezis thresholding along tangent direction followed by thinning process. Images and edges detected for different λτ values are shown in Figure-3.2. G(x,λ,τ) and R(x;λ,τ) functions for five different cases are given in Table 3-1.

Table 3.1 G(x,λ,τ) and R(x;λ,τ) filters

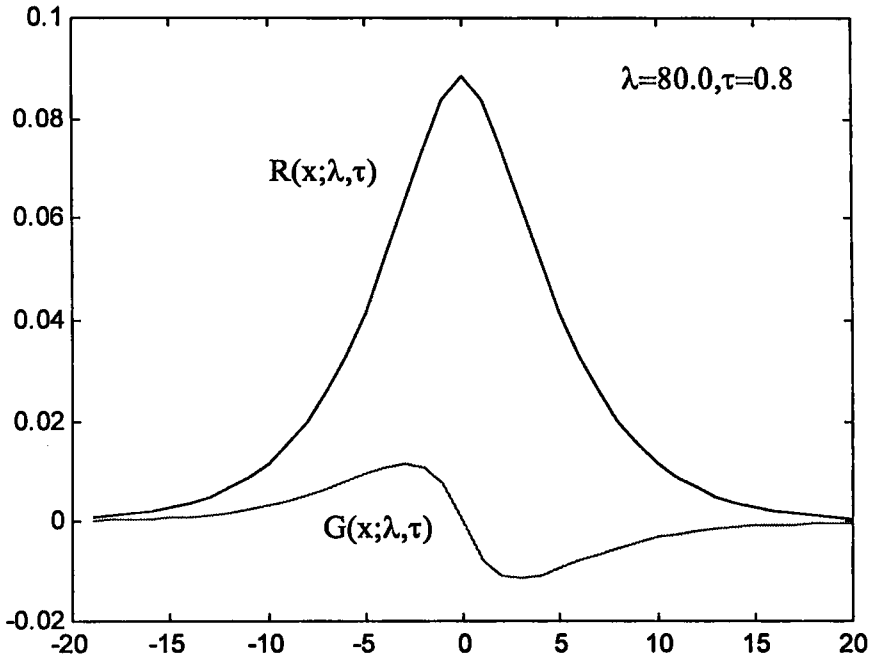
Case	G(x;λ,τ)	R(x;λ,τ)
1. Δ >0	$G(x;\lambda,\tau)=\frac{sign(x)}{2A(b^2-a^2)}(e^{-b\lvert x\rvert}-e^{-a\lvert x\rvert})$	$R(x;\lambda,\tau)=\frac{1}{2A(b^2-a^2)}\left[\frac{e^{-a\lvert x\rvert}}{a}-\frac{e^{-b\lvert x\rvert}}{b}\right]$
where $a=\sqrt{\frac{B+\sqrt{B^2-4A}}{2A}}$ $b=\sqrt{\frac{B-\sqrt{B^2-4A}}{2A}}$		
2. Δ <0	$G(x;\lambda,\tau)=\frac{-1}{4Aab}e^{-b\lvert x\rvert}\sin(ax)$	$R(x;\lambda,\tau)=\frac{e^{b\lvert x\rvert}}{4\sqrt{A}}\left[\frac{\cos(a\lvert x\rvert)}{b}+\frac{\sin(a\lvert x\rvert)}{a}\right]$
where $a=\frac{1}{A^{\frac{1}{4}}}\cos(\frac{1}{2}\arctan\sqrt{\frac{4A}{B^2}-1})$ $b=\frac{1}{A^{\frac{1}{4}}}\sin(\frac{1}{2}\arctan\sqrt{\frac{4A}{B^2}-1})$		
3. Δ =0	$G(x;\lambda,\tau)=\frac{-a}{2B}xe^{-a\lvert x\rvert}$	$R(x;\lambda,\tau)=\frac{e^{-a\lvert x\rvert}}{2B}\left[\frac{1}{a}+\lvert x\rvert\right]$
where $a=\sqrt{\frac{B}{2A}}$		
4. B=0	$G(x;\lambda,\tau)=\frac{-1}{2\sqrt{A}}e^{-a\lvert x\rvert}\sin a\lvert x\rvert$	$R(x;\lambda,\tau)=\frac{e^{-a\lvert x\rvert}}{4\sqrt{A}a}(\cos a\lvert x\rvert+\sin a\lvert x\rvert)$
where $a=\frac{-1}{\sqrt{2}A^{\frac{1}{4}}}$		
5. A=0	$G(x;\lambda,\tau)=\frac{-\mathrm{sgn}(x)}{B}e^{\frac{-\lvert x\rvert}{\sqrt{B}}}$	$R(x;\lambda,\tau)=\frac{e^{\frac{-\lvert x\rvert}{\sqrt{B}}}}{\sqrt{B}}$



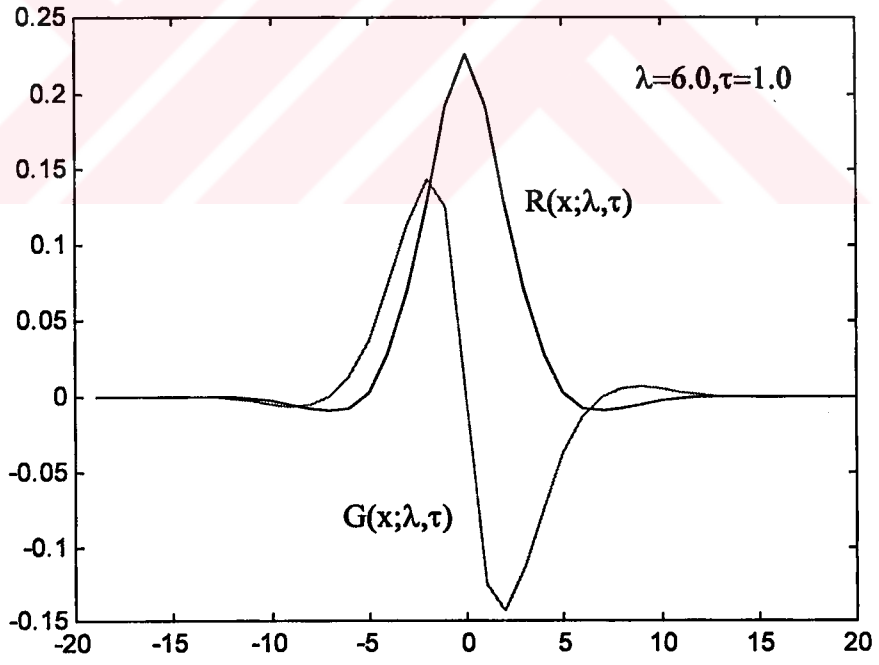
(a) Case 1, $\Delta > 0$



(b) Case 2, $\Delta < 0$



(c) Case 3, $\Delta = 0$



(c) Case 4, B=0

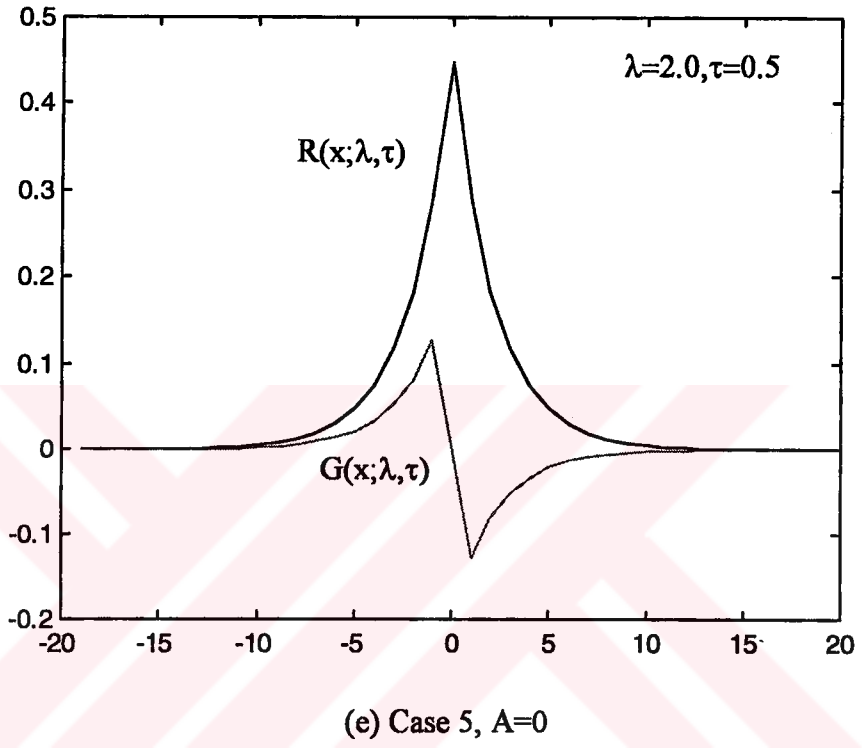
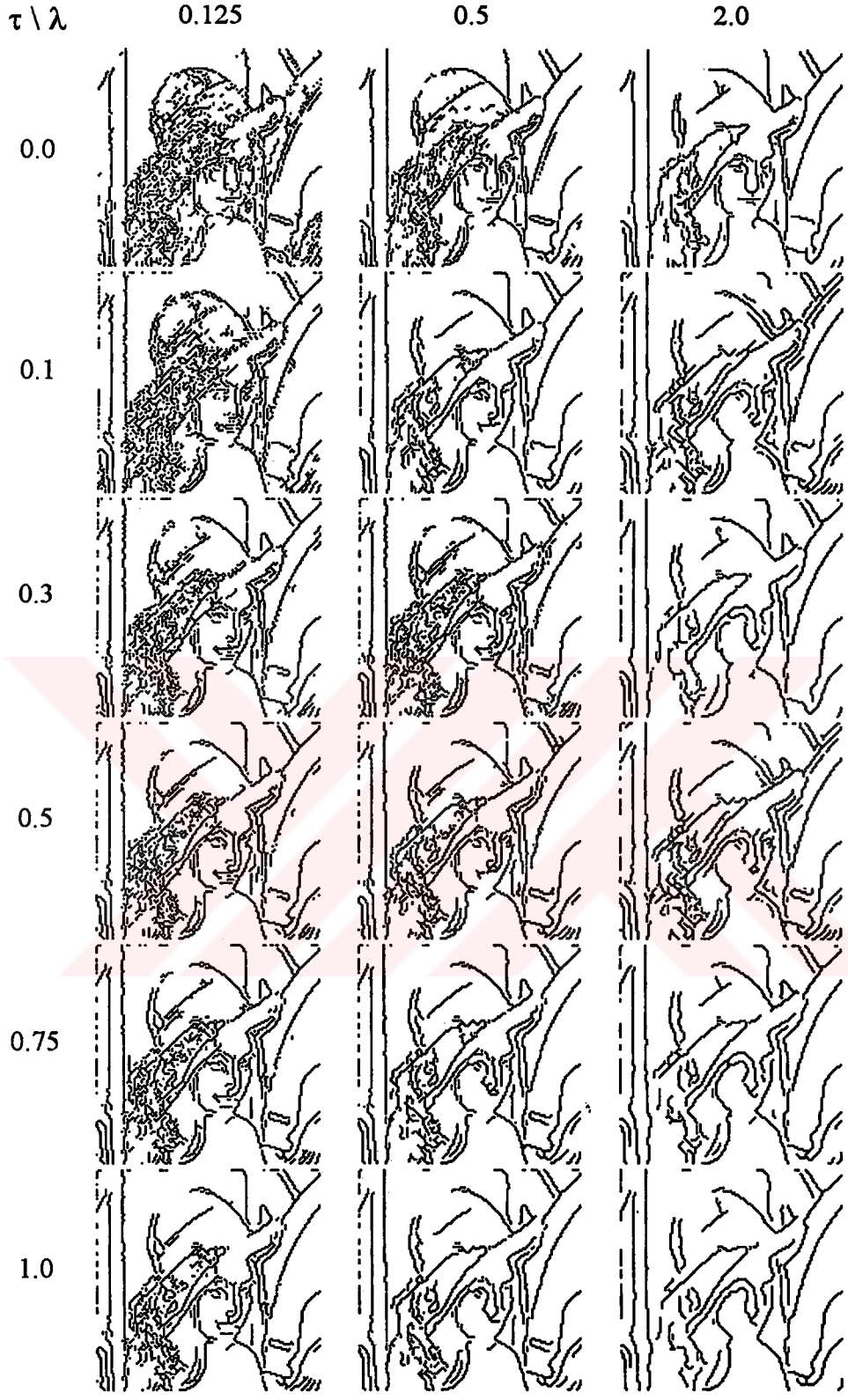


Figure-3.1 R(x;λ,τ) and G(x;λ,τ) filters



(a)

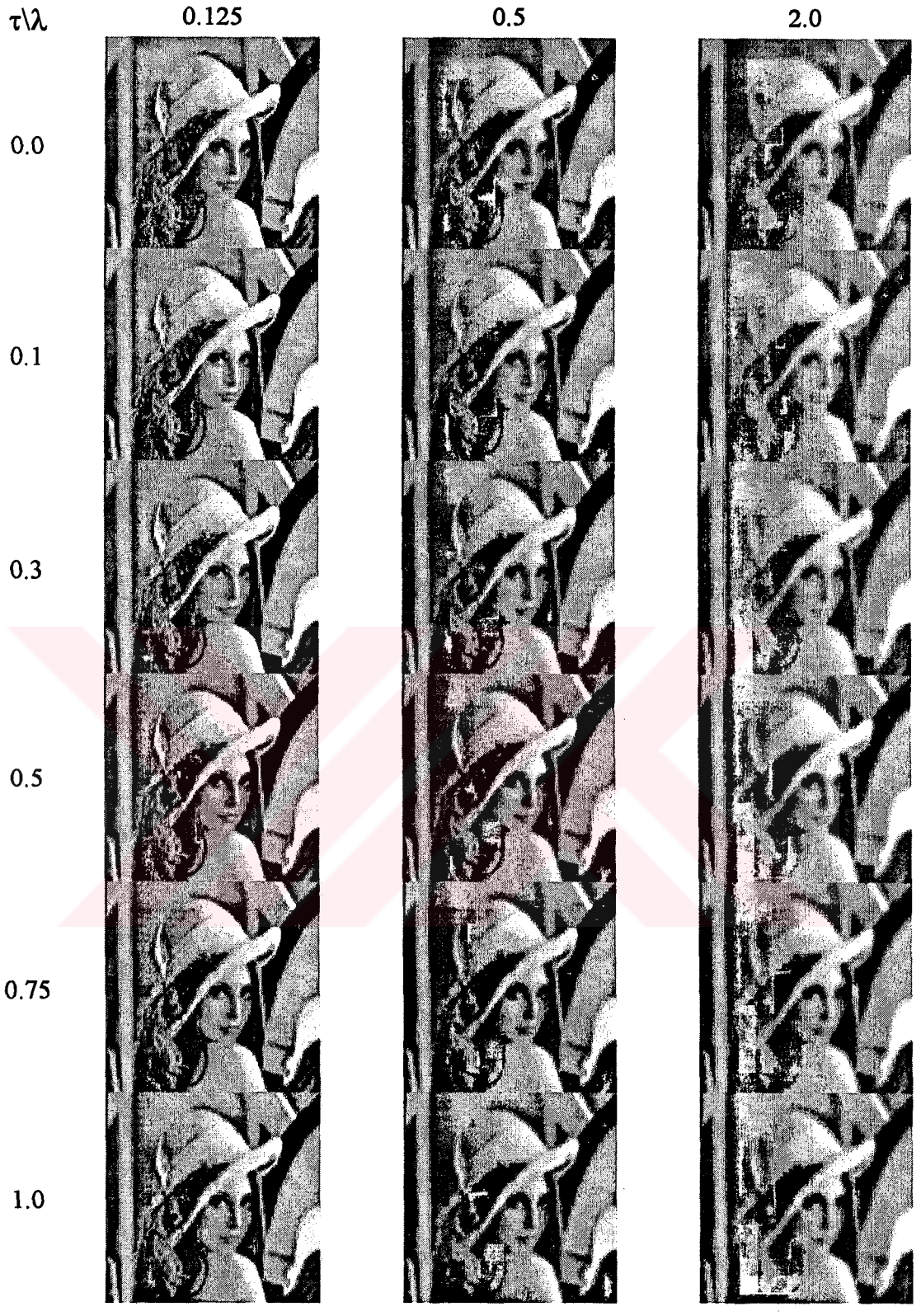


Figure 3.2 (a) Edges detected on λτ-space
(b) Images on λτ-space

3.3 Edge Tracing Algorithm

The output of the edge detection algorithm is a binary image. Binary images are coded effectively by differential chain code followed by Huffman coding. Chain code represents an arbitrary curve by assigning a codeword to each direction starting from the one end point of the curve. The purpose of edge tracing algorithm is to obtain the curves to code and to select with respect to their priority and eliminate some of which are less significant.

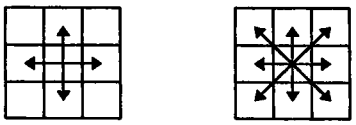


Figure 3.3 (a) 4-connected directions,
(b) 8-connected directions

An 8-connected directions are given in Figure 3.3-(b). Our studies have shown that if 4-connected directions given Figure 3.3-(a) are converted to 8-connected directions, then the extraction of model parameters becomes less sensitive to the noise and some false edges may be corrected. An example of 4-to-8 direction conversion is given in Figure 3.4 where the corners of a square is smoothed.

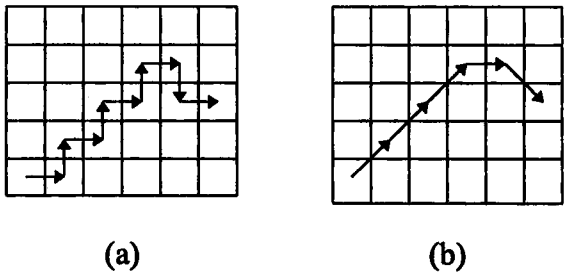


Figure 3.4 (a) Original edge map,
(b) Edge map after conversion

A block diagram of the tracing algorithm is shown in Figure-3.5. End points are obtained by scanning the binary image from left to right and top to bottom in the conventional manner. At the beginning of tracing, all edges are unmarked. When an edge is visited, then the visited edge point is marked as nonedge. If visited edge is on a cross- road point, edge is not marked. Since cross-road points correspond to such points where two or more edge segments intersect, marking a visited cross-road point results in breaking a continuous edge segment into two or more segments which increases the end points. At junction points, edge tracing algorithm is forced to follow the previous direction as much as possible so as to have high compression gain with the differential chain coding. If the visited point is not on a junction, then the next edge is the only edge that is in the neighborhood of the point. Otherwise the next edge is selected as a point which has the closest direction to the previous one.

3.4 Contour Filtering

Edge segments, $S = \{s_i | \ i = 1,...,N\}$, are obtained by the edge tracing algorithm explained above. These contours have to be filtered because of

- Edge detector gives some false contours,
- Since edge segments have different amount of contribution on reconstruction performance, some contextually unimportant features can be eliminated.

We have observed that edge segments can be classified and ordered with respect to their length, mean contrast calculated through normal direction of the segment and mean curvature directly derived from differential chain code. Because contextually unimportant information may consume much of the channel capacity, we offer a method controlling the selection of edge segments in a top-down manner. Through the ordering of edge segments, one can state that if leading edge segments in the order are used and others are omitted, since remaining edge segments are still enough

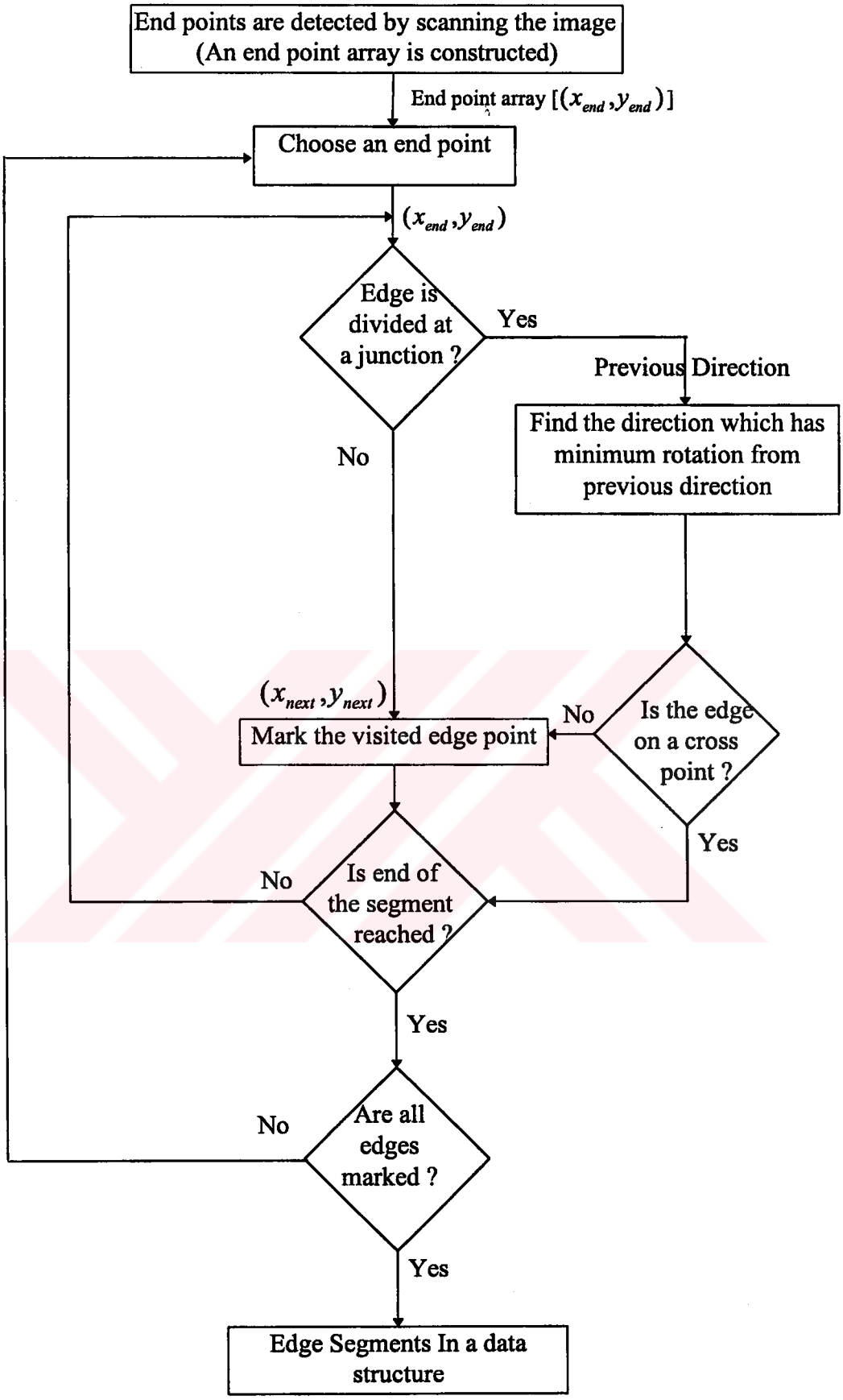


Figure 3.5 Edge Tracing Algorithm

to represent the image perceptually, the reconstructed image is still a good approximation of the original. In order to prove the validity of the selection algorithm, we have assigned a priority to each segments and eliminated some of them which has less priority for a given threshold. All segment features (length, contrast and curvature) are normalized.

$$\text{Priority}(s_i) = w_{\text{length}} \times Length(s_i) + w_{\text{contrast}} \times Contrast(s_i) + w_{\text{curvature}} \times Curvature(s_i) \quad (3.3)$$

The selection method is tested on three type of images having completely different characteristics. One type is a medical image (BRAIN.HIPS). In medical images, actually all features are perceptually important in some degree. Another type of image is a highly detailed one containing face (LENNA.HIPS). Other type of image contains broad regions and naturally very sparse and smooth edge data (HOUSE.HIPS). In Table 3.2-5, how SNR, NMSE, PSNR vary with the selection of edges is given. As contours are eliminated, NMSE increases, SNR and PSNR decreases expectedly, but reconstructed images (Figures 3.6-3.8, respectively) still contain most perceptual features.

Table 3.2 Selection and reconstruction results for BRAIN.HIPS
($w_{length} = -1.0,\ w_{contrast} = -1.0$)

Contour	Edge	%Threshold	NMSE	SNR(dB)	PSNR
339	4472	100	30.83	23.52	67.77
236	3998	75	31.36	23.19	67.43
164	3506	50	31.72	22.96	67.20
84	2855	25	33.01	22.16	66.40
36	1982	10	34.56	21.24	65.48

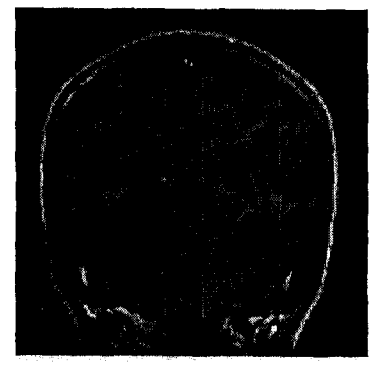
Table 3.3 Selection and reconstruction results for LENNA.HIPS
($w_{length} = -1.0,\ w_{contrast} = -1.0, w_{curvature} = 0.5$)

Contour	Edge	Threshold%	NMSE	SNR(dB)	PSNR
232	2754	100	20.73	31.46	63.47
165	2494	75	27.40	25.88	57.89
110	2199	50	28.98	24.76	56.88
56	1872	20	35.40	35.40	52.77

Table 3.4 Selection and reconstruction results for HOUSE.HIPS
($w_{length} = -1.0,\ w_{contrast} = -1.0$)

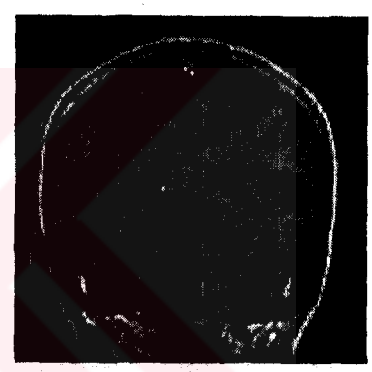
Contour	Edge	Threshold%	NMSE	SNR(dB)	PSNR
635	6626	100	12.46	41.60	72.08
475	6094	75	12.64	41.35	71.83
317	5336	50	13.37	40.23	70.70
160	4150	25	14.21	39.02	69.49
97	3466	20	15.39	37.41	67.89
64	3102	10	16.10	36.52	66.99
32	2663	5	18.66	33.56	64.04

(a)
Number of Contours 339
Number of Edges 4472



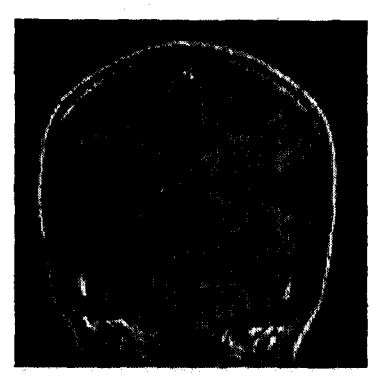
(b)
SNR = 23.52 dB

(c)
Number of Contours 236
Number of Edges 3998

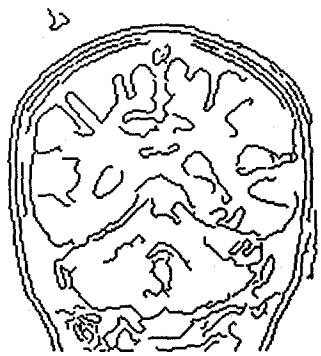


(d)
SNR = 23.19 dB

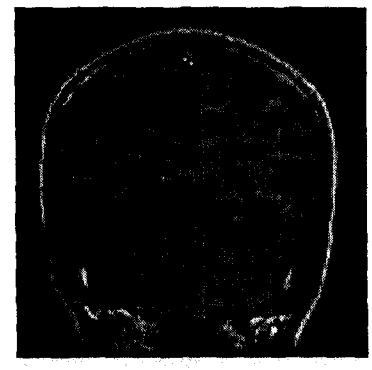
(e)
Number of Contours 164
Number of Edges 3506



(f)
SNR = 22.96 dB

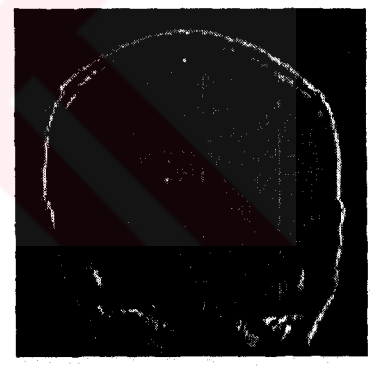


(g)
Number of Contours 84
Number of Edges 2855



(h)
SNR = 22.16 dB

(i)
Number of Contours 36,
Number of Edges 1982



(f) SNR = 21.24 dB

Figure 3.6 Reconstructed BRAIN Images

(a)
Number of Contours 232,
Number of Edges 2754

(b) SNR = 63.47 dB

(c)
Number of Contours 165,
Number of Edges 2494

(d) SNR = 57.89 dB

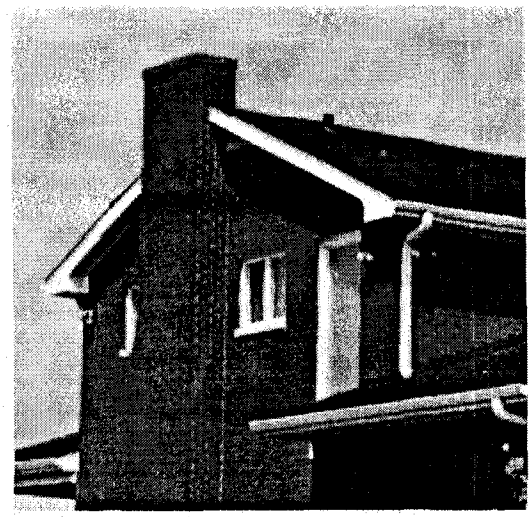
(e)
Number of Contours 110
Number of Edges 2199

(f) SNR = 56.88 dB

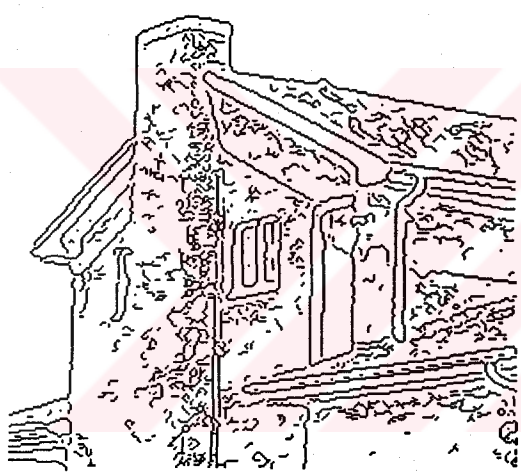
(g)
Number of Contours 56,
Number of Edges 1872

(h) SNR = 35.40 dB

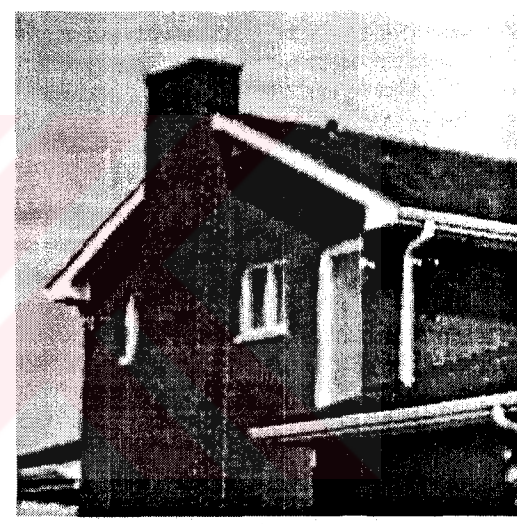
Figure 3.7 Reconstructed LENNA Images



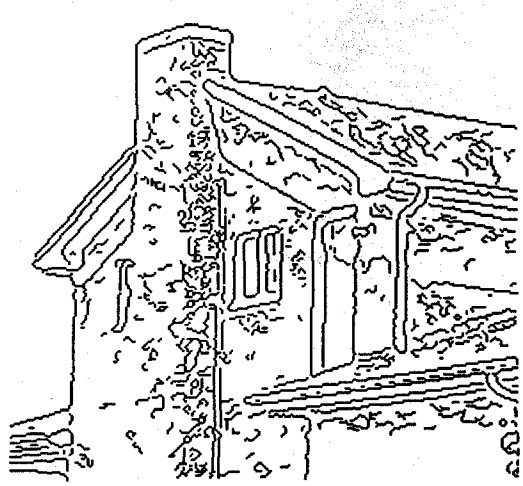
(a) Original House Image



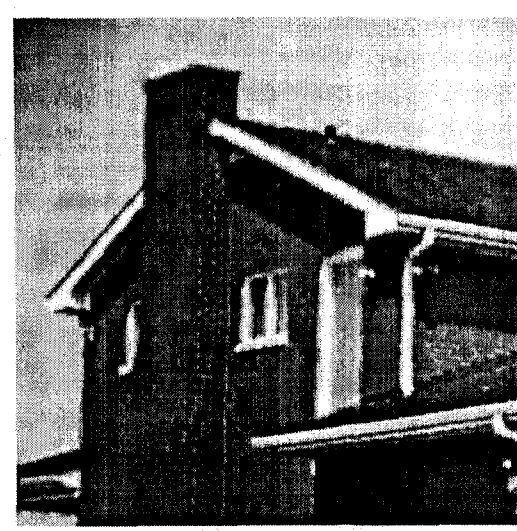
(b)
Number of Contours 635
Number of Edges 6626



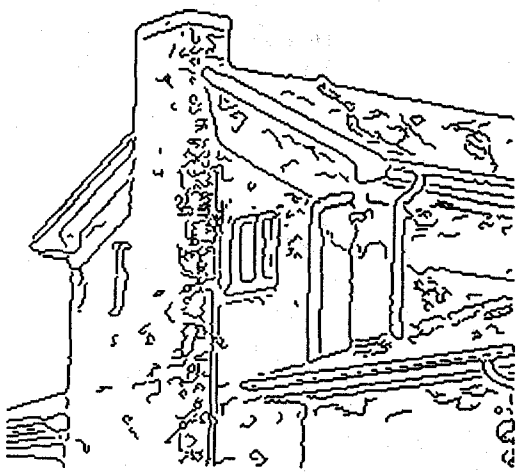
(c)
SNR = 41.60 dB



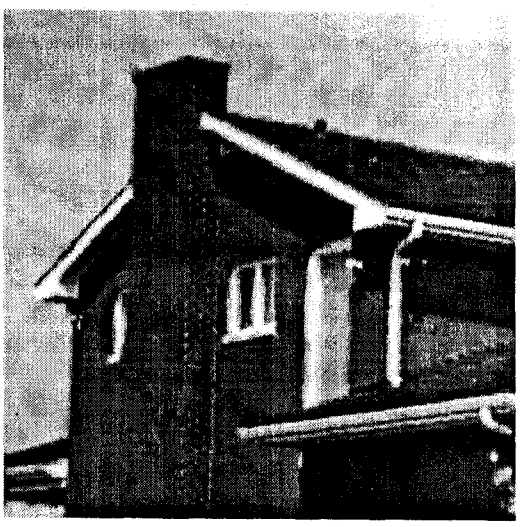
(d)
Number of Contours 475
Number of Edges 6094



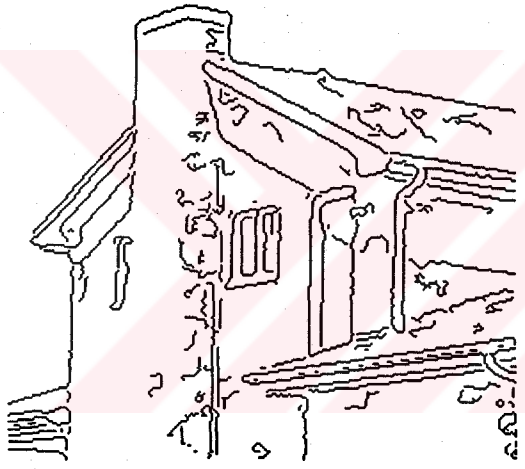
(e)
SNR = 41.35 dB



(f)
Number of Contours 317
Number of Edges 5336

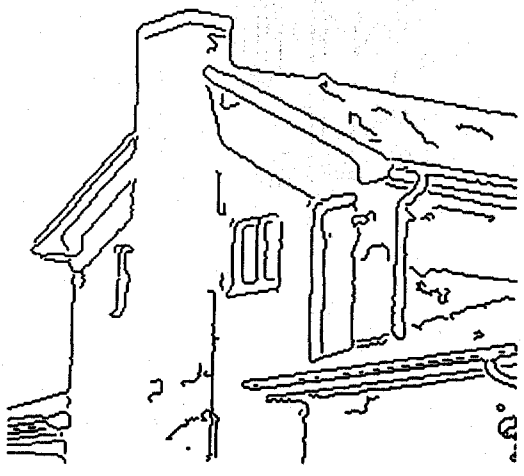


(g)
SNR = 40.23 dB

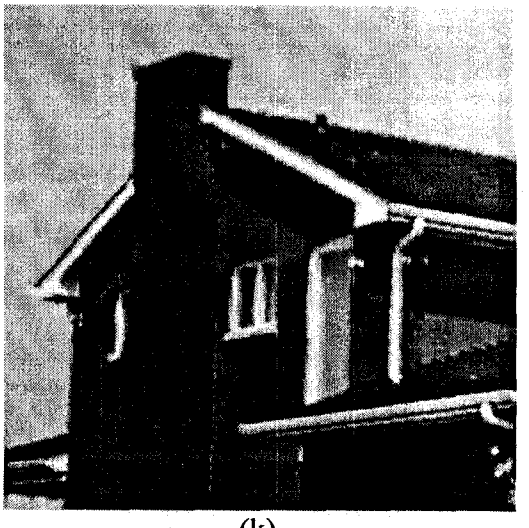


(h)
Number of Contours 160
Number of Edges 4150

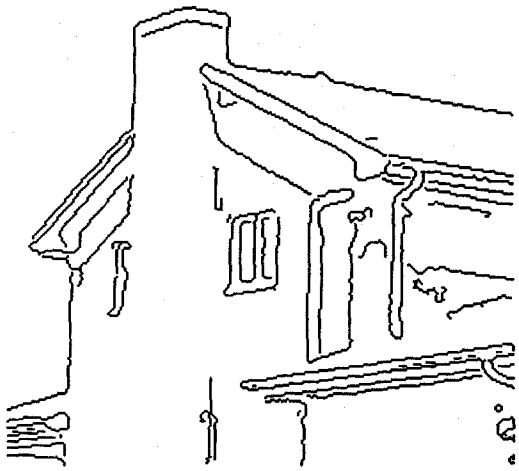
(i)
SNR = 39.02 dB



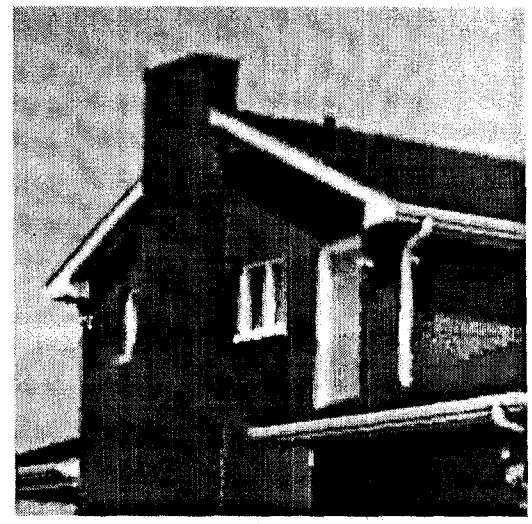
(j)
Number of Contours 97
Number of Edges 3466



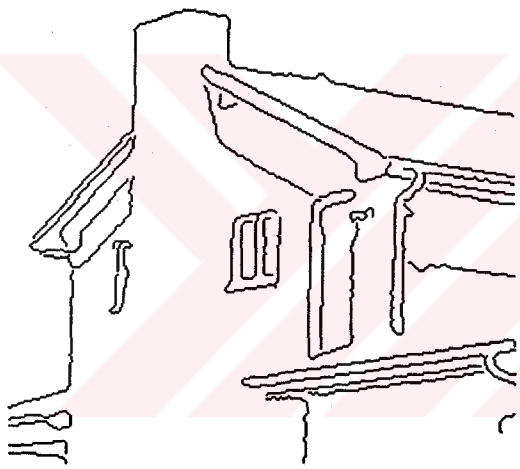
(k)
SNR = 37.41 dB



(l)
Number of Contours 64
Number of Edges 3102



(m)
SNR = 36.52 dB



(n)
Number of Contours 32
Number of Edges 2663

(o)
SNR = 33.56 dB

Figure 3.8 Reconstructed House Images

3.5 Differential Chain Coding

Chain code proposed by Freeman [18] in 1974, is an effective representation for arbitrary curves. Typically, this representation is based on the four or eight-directional encoding where the direction of each segment is coded using a numbering scheme given in Figure 3.9.

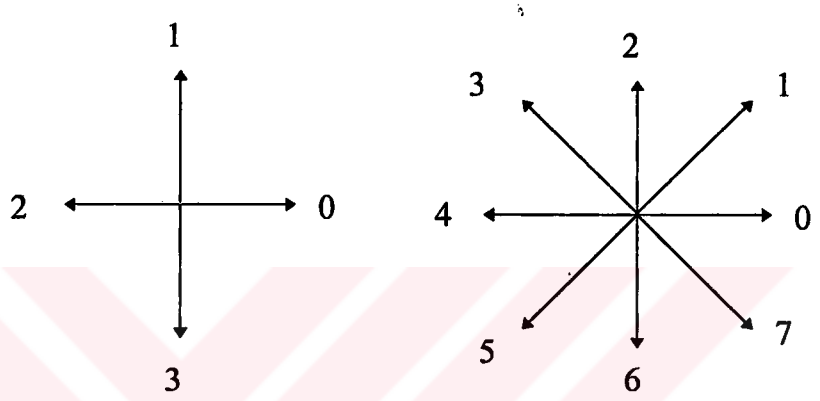


Figure 3.9 4 and 8-connected directions

The chain code is obtained by following a boundary in a direction explained in Section 3.3 -Edge Following, and assigning a direction to each segment.
A chain code is defined by

$$C = \{c_i \mid i = 1,\ldots,N\} \tag{3.4}$$

where c_i represents a direction according to the eight-directional chain code given in Figure 3.9 and N is the number of edge segments. Let $e_i = c_i - c_{i-1}$, for i=2,...,N be the difference in direction. Then the chain difference is defined by

$$d_i = \begin{cases} e_i + 8 & e_i < -3 \\ e_i - 8 & e_i > 4 \\ e_i & else \end{cases} \tag{3.5}$$

A differential chain code is defined by $D = \{d_i \mid i = 2,3,\ldots,N\}$. Smoothness property of curves suggests that the difference between two consecutive direction is small. The Huffman coding technique utilizes the statistics of the messages to make the most frequent symbols correspond to the shorter encoding and the rare symbols correspond to the longer encoding. Chain code does not only include the contours but also the starting point of them. Since the starting points in an edge map are distributed almost equally, they can be coded effectively in the form of distance between lexicographically ordered starting points. Another practical issue with the chain code is the need of a codeword for termination of contours. This problem is simply solved by defining the termination as a new direction. General view of edge map with contour code is given in Figure 3.10.

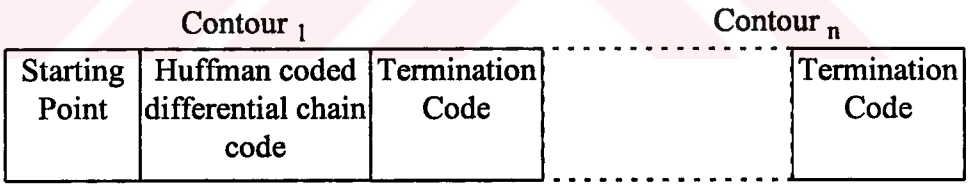


Figure 3.10 Huffman coded differential chain code format

Huffman coding makes two passes on the data stream. In the fist pass, frequency distribution of the data stream is obtained. Huffman coding scheme constructs a binary code book tree which is known to be uniquely decodable and optimal. In the second pass, each code is assigned to an appropriate codeword using the code book such that less probable messages are assigned to longer codewords and more probable messages are assigned to shorter codewords. Since the difference in

direction accommodate in the interval (-1..1), there is no need to send the codebook. It is implicitly assumed to be as in Figure 3.11. Experimental results given in Table (3.4,5,6) also prove the assumption.

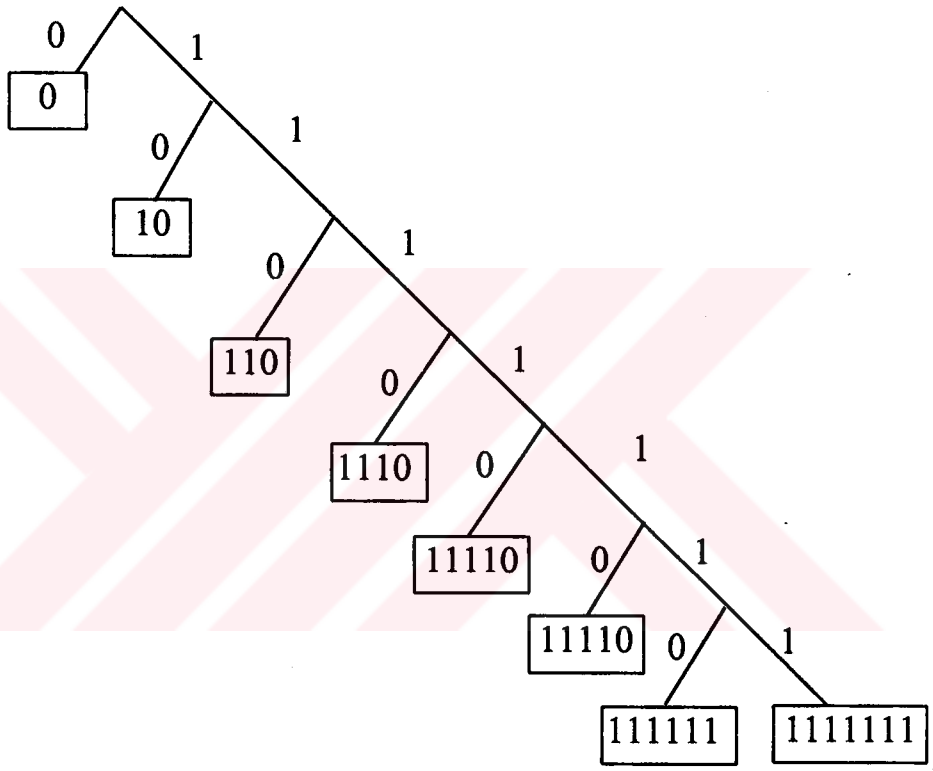


Figure 3.11 Code Book for Differential Chain

CR stands for Compression Ratio, ACL stands for Average Code Length, and ECL stands for End point Code Length in Table 3.5, Table 3.6, Table 3.7.

Table 3.5 Contour coding results for BRAIN.HIPS
($w_{length} = -1.0,\ w_{contrast} = -1.0$)

Contour	Edge	%Threshold	CR	ACL	ECL
339	4472	100	18.22:1	2.06	12
236	3998	75	21.7:1	2.00	13
164	3506	50	26.4:1	1.95	14
84	2855	25	36.5:1	1.88	15
36	1982	10	58.25:1	1.83	15

Table 3.6 Contour coding results for LENNA.HIPS
($w_{length} = -1.0,\ w_{contrast} = -1.0, w_{curvature} = 0.5$)

Contour	Edge	%Threshold	CR	ACL	ECL
232	2754	100	17.1:1	1.973	9
165	2494	75	20.0:1	1.926	10
110	2199	50	24.6:1	1.88	10
56	1872	20	35.5:1	1.81	11

Table 3.7 Contour coding results for HOUSE.HIPS
($w_{length} = -1.0,\ w_{contrast} = -1.0$)

Contour	Edge	%Threshold	CR	ACL	ECL
635	6626	100	23.6:1	2.01	13
475	6094	75	28.1:1	1.96	13
317	5336	50	35.1:1	1.89	14
160	4150	25	52.5:1	1.77	15
97	3466	20	71.5:1	1.49	13
64	3102	10	86.1:1	1.65	13
32	2663	5	109:1	1.61	14

CHAPTER 4

EXTRACTION OF MODEL PARAMETERS AND CODING

This chapter introduces the centipede model. Extraction and coding of the model parameters are investigated. Compression results are also shown at the end of the chapter.

4.1 Introduction to the Centipede Model

There are many ways to describe the intensity variations of pixels in an image. In edge-based image coding, previous studies has modeled the intensity variations along edge normal (edge profile) with Gaussian edge model such as

$$Edge(x;V_e,\Delta V,\sigma_b,d)=V_e+\frac{\Delta V}{2}erf\left(\frac{x-d}{\sigma_b}\right) \tag{4.1}$$

where $_e$ is the mean, ΔV is the contrast, and $_b$ is the blur parameter of the Gaussian. Such an edge profile model suffers from the following problems :

1) (4.1) does not work well for a non symmetric edge,

2) Edge profiles can not be modeled by a Gaussian-like function due to the interactions of near edges in scale space ([21,33,38]).

When deciding on a description, it must be proved at least experimentally that the model represents the original image enough to reproduce within a small error distance. We present a model which gives a powerful description of the image. The centipede model is given by a tuple (I_L,C_L,C_R,W_L,W_R) with the edge map where I_L represents the intensity on the edge, C_L (C_R) represents left (right) contrast along the normal direction of the contour, and W_L (W_R) represents left (right) width. Width is defined as the distance at which difference in consecutive pixel is lower than a given threshold. Threshold is determined from the SNR_{dB} ratio for an image. Width is proportional to edge scale.

The image wih the edge profile is obtained by minimizing the hybrid energy model. For 1D case, the model covers a vast range of edge profiles. Hybrid energy functional in 1D is given in (4.2).

$$E(f;\lambda,\tau)=\int[\beta(x)(f(x)-d(x))^2+\lambda(\tau f_x^2+(1-\tau)f_{xx}^2)]dx \tag{4.2}$$

Minimization of the energy functional is obtained by first discretization

$$E(f;\lambda,\tau)=\sum_{i\in I}\beta_i\,(f_i-d_i)^2+\lambda\,\tau\,(f_i-f_{i-1})^2+\lambda\,(1-\tau)(f_{i+1}+f_{i-1}-2f_i)^2 \tag{4.3}$$

followed by applying Successive Over Relaxation (SOR) [35] yields an iteration

$$f_i^{(n+1)}=f_i^{(n)}-\frac{w}{T}\frac{\partial E_i(f)}{\partial f_i} \tag{4.4}$$

$$\frac{1}{2}\frac{\partial E_i(f)}{\partial f_i} = -\beta_i d_i + (\beta_i + 2\lambda(3-2\tau))f_i^{(n)} - \lambda(4-3\tau)(f_{i-1}^{(n+1)} + f_{i+1}^{(n)}) + \lambda(1-\tau)(f_{i-2}^{(n+1)} + f_{i+2}^{(n)}) \quad (4.5)$$

The curve denoted by C_{org} in Figure 4.1 represents a nonsymmetric edge profile. The curve $C_{centipede}$ is obtained by the centipede model (I_L=0,C_L=100,C_R=150,W_L=75,W_R=82). The reconstructed edge profile is denoted by C_{cons} and obtained by solving the iteration (4.5) with parameters (w=1.0, λ=0.8, τ=0.5, iteration number 2000). Shape of the reconstructed curve is controlled by the pair (λ,τ). λ controls the smoothness of the profile and τ controls the continuity of the profile. This is called $\lambda\tau$–space representation of functions.

We have also tested the model on different type of edge profiles (e.g. step-edge, linear edge), the centipede model performs well for all of them.

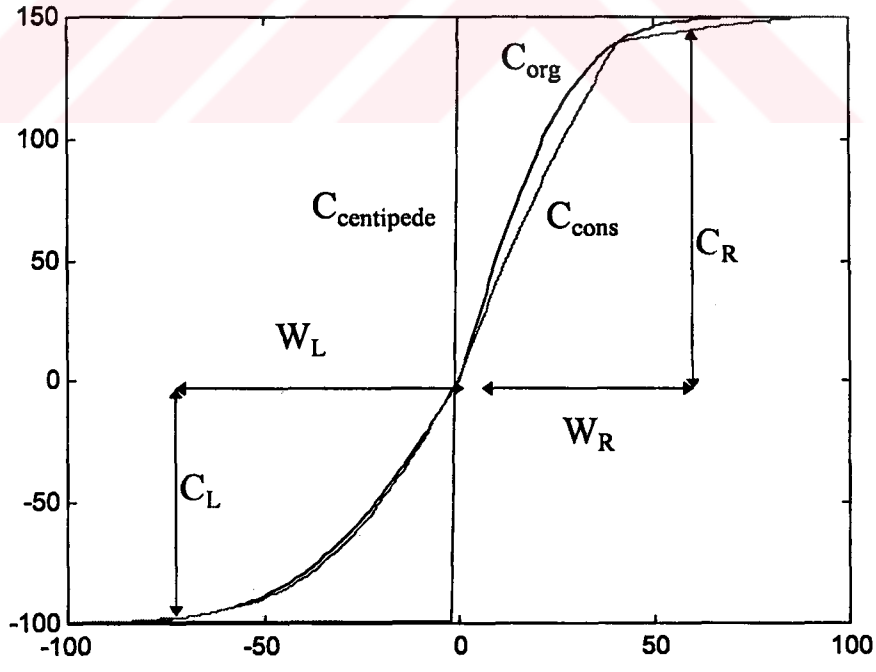


Figure 4.1 A non symmetric edge profile (C_{org}) and the reconstructed edge profile (C_{cons}) with the Centipede model ($C_{centipede}$).

4.2 Extraction of Model Parameters

In the previous section, the centipede model and its properties are explained. Performance of the model is dependent on how the parameters are extracted from given image and edge map obtained by GED. The problem is to extract numerical data for the Centipede Model. The extraction is done through the edge segments and along the normal direction to the contour for each edge element.

Let $P_n(c_i)$ be the edge profile along the direction perpendicular to the edge direction and $P_e(c_i)$ be the edge profile along the edge direction for the edge element $c_i = (x_i^{c_i}, y_i^{c_i})$. Since contours are obtained by following connected edge elements, $P_e(c_i)$ and consequently (I_L) is easily obtained. (C_L,C_R,W_L,W_R) is derived from the edge profile $P_n(c_i)$ captured by intensity along the normal direction at $(x_i^{c_i}, y_i^{c_i})$ (Figure 4.2).

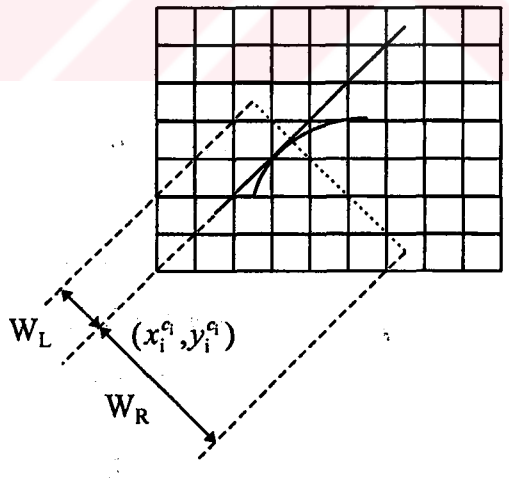


Figure 4.2 Edge profile extraction

Mathematical definitions for model parameters (C_L,C_R,W_L,W_R) are given in (4.6) and (4.7).

$$W_R(s) = \left\| V(s) - \inf_{V_R} \{ \|V - V_R\| ; \left\| \frac{\partial \mathrm{I}(x,y)}{\partial n} \right\|_{V=V_R} < T_R \} \right\|$$
$$W_L(s) = \left\| V(s) - \inf_{V_L} \{ \|V - V_L\| ; \left\| \frac{\partial \mathrm{I}(x,y)}{\partial n} \right\|_{V=V_L} < T_L \} \right\| \quad (4.6)$$

$$C_R = I(V_R(s)) - I(V(s)) \quad (4.7)$$
$$C_L = I(V_L(s)) - I(V(s))$$

where T_L and T_R represents the thresholds. Since difference operators are very sensitive to the noise, the definition of (W_L,W_R) is an ill-conditioned problem. The thresholds are used to make the extraction process robust to the noise. They are determined from the SNR_{dB} ratio for an image. As SNR_{dB} increases, T_L and T_R are lowered to zero.

Original house image and edge map obtained by GED with parameters (λ=2.0, τ=0.5) are given in Figure 4.3 (a)-(b). Extracted centipede model images are given in Figure 4.3 (c)-(d) for T_L =8 and T_R =12. These parameters are experimentally chosen. In Figure 4.3 (e), some of the lines whose length is left and wright widths (W_L,W_R) normal to the edge direction are drawn. The centipede model being overlaid on the original image is shown in Figure 4.3 (e). Also, Figure 4.3 (e) explains why the model is called "centipede". Results for two different types of images (brain and Lenna) are given in Figure (4.4) and (4.5).

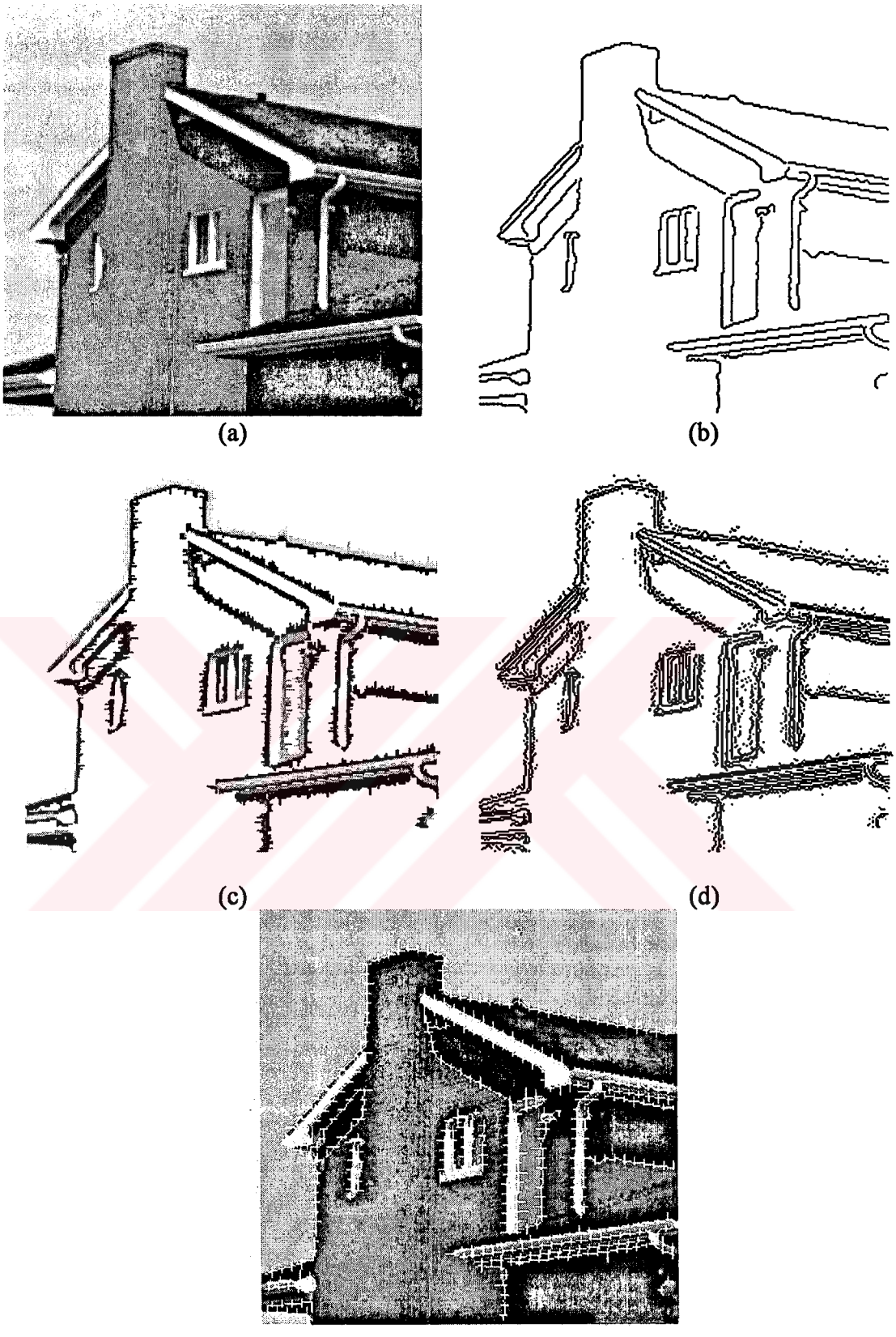


Figure 4.3 Extracted Centipede Model For House Image
(a) Original House Image
(b) Edges detected by GED for (λ=2.0,τ=0.5,Case II)
(c) Intensities at (d)
(d) Edge and Width Map
(e) The centipede model overlied on the original House image

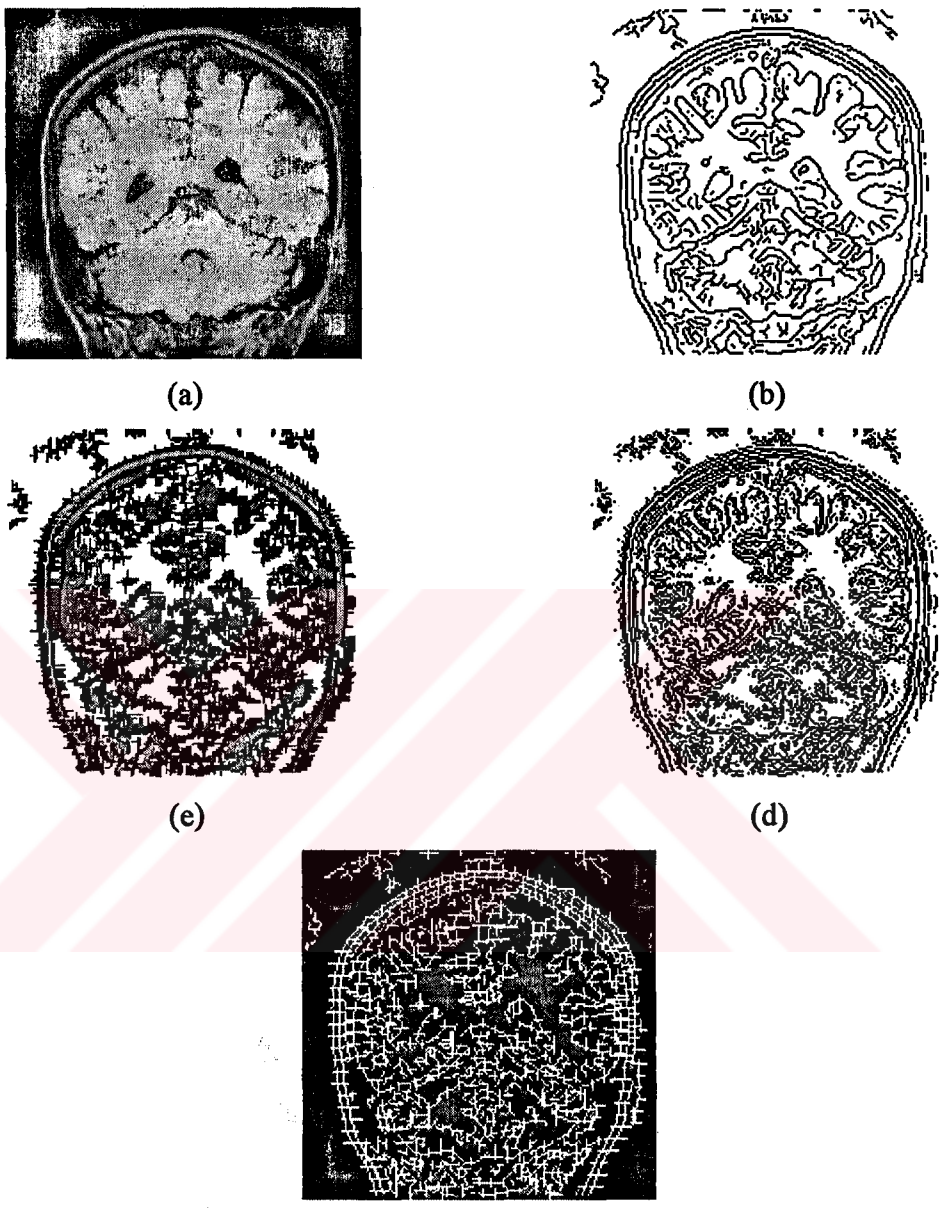


Figure 4.4 Extracted Centipede Model For Brain Image
(a) Original Brain Image
(b) Edges detected by GED for (λ=0.5,τ=0.5,Case II)
(c) Intensities at (d)
(d) Edge and Width Map
(e) The centipede model overlied on the original Brain image

Figure 4.5 Extracted Centipede Model For lenna Image
(a) Original Lenna Image
(b) Edges detected by GED for (λ=0.5,τ=0.5,Case II)
(c) Intensities at (d)
(d) Edge and Width Map
(e) The centipede model overlied on the original Lenna image

4.3 Coding Of The Model Parameters

In the previous section, a method is introduced to extract the model parameters (I_L,C_L,C_R,W_L,W_R) by capturing the edge profiles $P_n(c_i)$ and $P_c(c_i)$. The parameters are modeled by polynomials and polynomial coefficients are coded. There are many advantages of coding with polynomials :

- Since parameters extracted from neighboring edge points are very similar and exhibit high correlation, it is therefore reasonable to use polynomials for exploiting the redundancy. Also, coding of all model parameters is a bit-consuming operation.
- Fitting polynomials to the parameters by minimizing least mean square error results in smoothing of the parameters. This may also smooth the error due to the parameter extraction process.
- It gives the ability of changing the compression ratio while allowing small amount of degradation. Lower the order of the polynomials, higher the compression ratio is.
- It is computationally easy to fit a polynomial to a curve.

In this section, curve fitting with polynomials is explained, then we proceed with the image reconstruction from model parameters by hybrid energy functional.

4.3.1 Curve Fitting with Polynomials

In curve fitting, we are given n points with pairs (x_1,y_1), ... , (x_n,y_n) and we want to determine a function f(x) such that $f(x_j) \cong y_j$, j=1,...,n. The type of function may be suggested by the nature of the problem. We offer polynomials of order n due to the reasons explained above. A widely used procedure of curve fitting is the method of *least squares*.

Let $P_n(x)$ be the n[th] order polynomial with the coefficients $(c_0,c_1,...,c_n)$ as given in (4.7)

$$P_n(x)= c_0 + c_1x^1 + c_2x^2 +...+c_nx^n \tag{4.7}$$

M data pair is given in the form $(y_0,y_1,...,y_M)$. We want such a polynomial with the coefficients $(c_0,c_1,...,c_n)$ that it minimizes the quantity MSE denoted by Q :

$$Q = \sum_{i=1}^{M}[P(x_i)-y_i]^2 \tag{4.8}$$

A necessary condition for MSE to be minimum is

$$\frac{\partial Q}{\partial a_i} = 0,\ i=0,...,n \tag{4.9}$$

where $P(x_i)= \sum_{k=0}^{n} c_k x_i^k$. We define x_i as $x_i = i$ for simplicity. In order to have a matrix form, we will define the following matrices

$$S_k = \sum_{i=1}^{M} x_i^k = \sum_{i=1}^{M} i^k \tag{4.10}$$

$$M_k = \sum_{i=1}^{M} x_i^k y_i = \sum_{i=1}^{M} i^k y_i \tag{4.11}$$

Solution of equation (4.9) becomes the solution of the following linear matrix equation

$$[S][A]=[M] \tag{4.12}$$

where

$$[S]=\begin{bmatrix} S_0 & S_1 & \cdots & S_{n-1} \\ S_1 & S_2 & \cdots & S_n \\ \vdots & \vdots & \cdots & \vdots \\ S_{n-1} & S_n & \cdots & S_{2(n-1)} \end{bmatrix},$$

$$A=\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} \quad \text{and} \quad M=\begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \end{bmatrix}.$$

Note that this system is symmetrix. To solve (4.12) for unknowns $(c_0,c_1,...,c_n)$, first the matrix S is transformed into a tridiagonal form by Householder's method [36]. An algorithm for the curve fitting by polynomials is given in Table 4.1.

4.3.2 Image Reconstruction By Using The Hybrid Model

The following processes are applied to the image in given order at the transmitter

I. Edge map is obtained by generalized edge detector,

II. The edge map with the image is used to extract the model parameters,

III. The extracted parameters are modeled by polynomials,

IV. Polynomial coefficients are quantized and sent with the edge map coded by differential chain code followed by Huffman coding.

Receiver contructs two images which are

1. Edge map which is a binary image,
2. Intensity image is obtained from model parameters (I_L,C_L,C_R,W_L,W_R) which are constructed easily by evaluating the polnomial value at each edge point.

Since both images are sparse, we use hybrid energy functional to span a surface through these points. Hybrid energy functional tries to find such a function f(x,y) which minimizes

$$E(f) = \iint_{\Omega} (f(x,y) - d(x,y))^2 + \lambda \left[(1-\tau)(f_x^2 + f_y^2) + \tau(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)\right] dxdy \tag{4.13}$$

where λ controls the smoothness of the surface and τ controls the continuity of the surface. In the functional, first term on the right hand side is a measure of the closeness of the solution *f(x,y)* to the data *d(x,y)*, and the second and the third terms are stabilizers on the solution as the first and second order derivatives. The Euler-Lagrange equation associated with this hybrid functional is

$$\lambda\tau \left[f_{xxxx} + 2f_{xxyy} + f_{yyyy}\right] - \lambda(1-\tau) \left[f_{xx} + f_{yy}\right] + f = d \tag{4.14}$$

Properties of the hybrid model and $\lambda\tau$–space can be found in [2]. The minimization problem can be solved by discretizing the partial differential equation (4.14) or directly discretizing the hybrid energy functional by using finite difference approximation of derivatives.

$$E(u;\lambda,\tau) = \sum_i \sum_j \beta_{i,j}(u_{i,j} - d_{i,j})^2 + \sum_i \sum_j \lambda\tau((u_{i,j} - u_{i-1,j})^2 + (u_{i,j} - u_{i,j-1})^2) + \sum_i \sum_j \lambda(1-\tau)((u_{i+1,j} + u_{i-1,j} - 2u_{i,j})^2 + (u_{i,j+1} + u_{i,j-1} - 2u_{i,j})^2 + 2(u_{i+1,j+1} + u_{i,j} - u_{i+1,j} - u_{i,j+1})^2)$$

An iterative solution to (4.15) is obtained by the Successive-Over Relaxation iterations as

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} - \frac{w}{T}\frac{\partial E(u)}{\partial u_{i,j}} \tag{4.16}$$

$$\begin{aligned}\frac{\partial E(u)}{\partial u_{i,j}} = &-\beta_{i,j} \times d_{i,j} + [\beta_{i,j} + 4\lambda(1+4\tau)] \times u_{i,j}^{(n)} \\ &- \lambda(1+7\tau)[u_{i-1,j}^{(n+1)} + u_{i,j-1}^{(n+1)} + u_{i+1,j}^{(n)} + u_{i,j+1}^{(n)}] \\ &+ 2\lambda\tau[u_{i-1,j-1}^{(n+1)} + u_{i-1,j+1}^{(n+1)} + u_{i+1,j-1}^{(n)} + u_{i+1,j+1}^{(n)}] \\ &+ \lambda\tau[u_{i-2,j}^{(n+1)} + u_{i,j-2}^{(n+1)} + u_{i+2,j}^{(n)} + u_{i,j+2}^{(n)}]\end{aligned} \tag{4.17}$$

where $\beta_{i,j}$ handles the sparseness of data and it is equal to 1 if data is available at the point (i,j), and 0 otherwise. In Figure 4.3,4.4,4.5 (c) and (d) represents $\beta_{i,j}$ and $d_{i,j}$.The iterations terminate when the condition $\left\|u_{i,j}^{(n+1)} - u_{i,j}^{(n)}\right\|_\infty < \varepsilon$ is satisfied for a prespecified ε. The iteration given by (4.16) is an interpolation by iterative overrelaxation governed by the heat diffusion equation. Such iteraive solutions require long convergence time. Theoretically, the number of iterations for convergence is N^2 . One way to accelarate the convergence is to use multigrid technique [41]. Recently, Salembier et al. [30] has presented a new interpolant based on morphological operations such as dilation and erosions which do not require any multiplication. They have reported 13 iterations equivalent to 376 iterations of multigrid diffusion and 2980 iterations of linear diffusion.

We examine how the quality of reconstructed image by using the hybrid model is effected by the order of the polynomial and the block length in terms of nmse. It is

Table 4.1 Curve fitting by polynomials

- Construction of S and M matrices

```
for k:=1 to M
 for i:=0 to n
    for j:=0 to n
          begin
            S[i][j] = S[i][j] + power(k,i+j)     ;
            M[i]   = power(k,i) * y[i]       ;
         end
```

- Solution of linear matrix equation [S] [A] = [M]

Symmetric linear system [S] [A] = [M] is transformed into tridiagonal form by Householder's method. Then unknows can easily found as follows

```
for i:= 0 to n-2
for j:= i+1 to n-1
    begin
        pivot   := - S[i][j]/ S[i][i]        ;
      S[j][i]  :=  0                 ;
      for k:= j+1 to n-1
          S[j][k] = S[j][k] + pivot × S[i][k]   ;
     M[j]   = pivot × M[i]           ;
   end

A[n-1] := M[n-1] / S[n-1][n-1]          ;
for k:= n-2 to 0
   begin
      A[k] := M[k]                ;
  for i:= n-1 to k+1
      A[k] := A[k] - S[i] × A[i]          ;
end
```

evident that the nmse increases as the order decreases and block length increases. Beside that, we have also observed that there exist an order and a block length such that nmse of the reconstructed image starts to remain almost constant. Quality of the reconstructed image is effected by the order of the polynomial for intensity and contrast much more than the one for width.

Reconstructed images by the centipede model with varying polynomial order is given in Figure 4.6. For Lenna image and its edge map given in Figure 4.6 (a) and (b) when the block length is equal to 10. Since nmse remains the same for higher order approximations at about 37, the polynomial order is ($Order_{contrast}$, $Order_{intensity}$,$Order_{width}$)=(3,4,1) enough to represent the model parameters.

Quality of the reconstructed images by the centipede model with varying block length is given in Table 4.2 for Lenna image. It has been observed that nmse for block length of 12 is still close to the nmse's for shorter block lengths.

(a)

(b)

(c)

(d)

(e) (f)

Figure 4.6 (a) Original Lenna Image
(b) Edges detected by GED for (λ=0.5,τ=0.5)
(c) Reconstructed Lenna Image for (1,1,1)
nmse=36.00, snr=20.42, psnr=52.11
(d) Reconstructed Lenna Image for (2,2,1)
nmse=34.11, snr=21.50, psnr=53.19
(e) Reconstructed Lenna Image for (3,4,1)
nmse=33.60, snr=21.80, psnr=53.49
(f) Reconstructed Lenna Image for (4,5,2)
nmse=33.38, snr=21.94, psnr=53.62
where ($Order_{contrast}$, $Order_{intensity}$,$Order_{width}$) stands for polynomial orders

Table 4.2 Quality of Image Recostructed by Centipede Model with respect to Block Length

Block Length	NMSE	SNR (dB)	PSNR
1	29.53	24.39	56.08
2	33.67	21.70	53.45
4	36.43	20.19	51.88
6	37.96	19.36	50.05
8	39.03	18.81	50.50
10	39.36	18.64	50.33
12	40.41	18.12	49.80
Contour Length	48.04	14.66	46.35

4.4 Experimental Results

We have used two methods to increase the quality of the reconstructed image while decreasing the compression ratio little:

- Adaptively varying the order of polynomial in a block with a constant length,
- Mean coding.

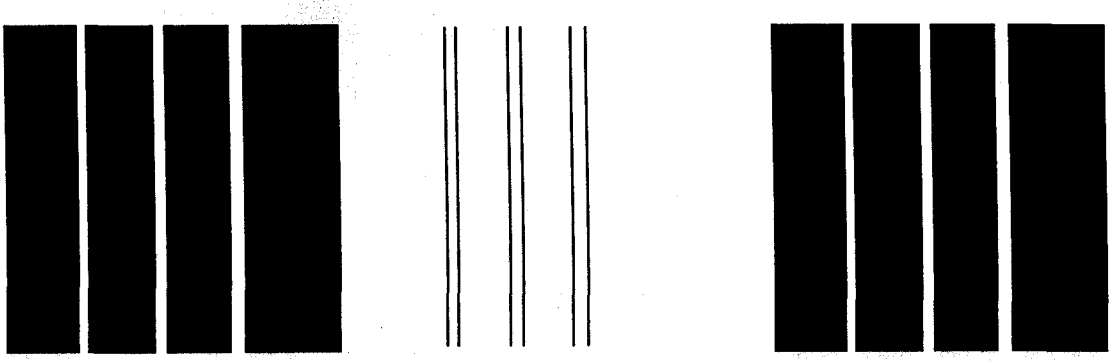
Different contours need different order of approximation for the same error measure. By adaptively changing the order of the polynomial, it is possible that the reconstructed image will have higher quality for the same compression ratio. The results from the previous section also suggests this conclusion.

Since the surface reconstruction does not guarantee that the mean of the image will be preserved. In mean coding, the mean of the image in a block of size W is coded and sent to the receiver. Receiver side uses the mean information in the reconstruction level in such a way that the mean of the reconstructed image is closer to the received mean in that block. Sending the mean of the image in a block does not only increase the quality of the reconstructed image but also speed the reconstruction process up.

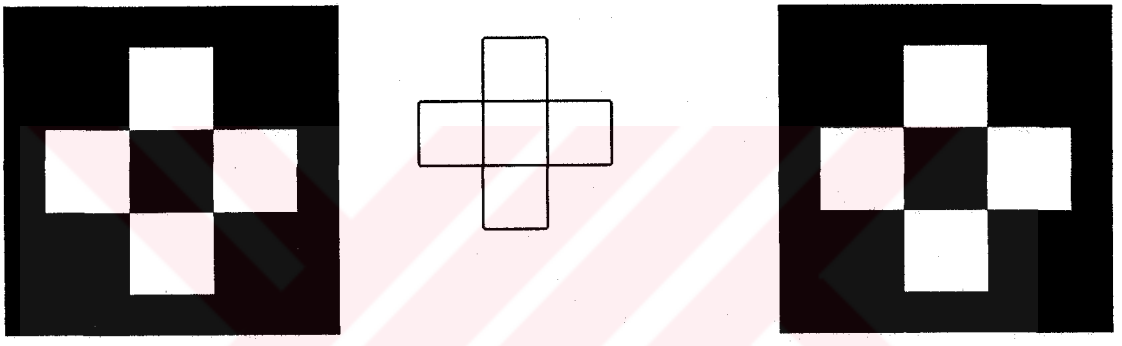
Compression results are given in Figure 4.7 with the improvements stated above. We have applied the centipede model on various type of images from artificial images to real images having completely different features. The results are summarized in Table 4.3.

Table 4.3 Compression Results for various type of images (CR stands for Compression Ratio)

Image	**Size** (bytes)	**CR Differential Chain Code**	**CR Polynomial Coefficients**	**CR**	**NMSE**	**SNR (dB)**
che.hips	16465	210:1	230:1	**157:1**	14.56	63.26
bars.hips	16465	160:1	606:1	**127:1**	10.28	65.73
mouse.hips	250149	186:1	251:1	**107:1**	16.25	36.33
house.hips	65106	86:1	107:1	**48:1**	38.56	19.05
camera.hips	262225	70:1	140:1	**43:1**	36.29	20.26
lena.hips	86561	48:1	123:1	**35:1**	35.95	20.45
brain.hips	30706	23:1	30:1	**13:1**	46.27	15.41



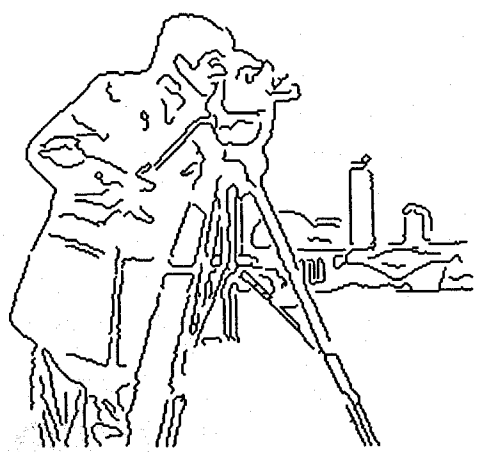
Compression Ratio=127:1



Compression Ratio=157:1

Compression Ratio=13:1

Figure 4.7 Compression Results For Different Types of Images
Original Image, Edges detected by GED, Reconstructed image by the centipede model are given from left to right

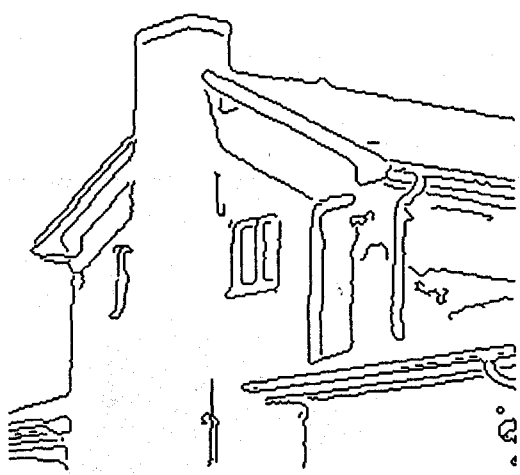


(C)

(D)

Figure 4.8 Compression Results For Cameraman
Original Image, Edges detected by GED, Reconstructed images by the centipede model are given,
(C) Compression Ratio=28:1
(D) Compression Ratio=43:1

Figure 4.9 Compression Result For Lena Image
Original Image, Edges detected by GED, Reconstructed image by the centipede model are given from left to right
Compression Ratio = 35:1 and SNR=20.45 dB



(C)

(D)

Figure 4.10 Compression Result For House Image
Original Image, Edges detected by GED, Reconstructed image by the centipede model are given,
(C) Compression Ratio = 48:1
(D) Compression Ratio = 74:1

CONCLUSIONS AND SUGGESTIONS

CONCLUSIONS

We have presented a model-based image compression method for loosy-coding of images. The model is based on the edge features detected by using GED, representing the image in a domain called $\lambda\tau$-space where λ represents the smoothness of the intensity and τ represents the continuity of the intensity information. The centipede model contains edge location, intensity on the edge, contrast of the edge and the widths. It has been shown that the model yields a powerful representation of the image such that the reconstructed image is perceptually closer to the original image than the existing edge-based methods can achieve for reasonable edge information.

We have also presented a contour selection algorithm in which edge segments are sorted with respect to their length, mean contrast through the segment along the normal direction, and mean curvature directly derived from differential chain code representation of the contour. One can select the most perceptual contours just by taking the leading contours in this order. We have experimentally shown that such a selection is meaningful and performs good selection in terms of NMSE and visual appearance of the reconstructed image.

The algorithm has two parts. In the first part, the edges are detected and processed by edge tracing algorithm to obtain distinct contours and contour selection algorithm to eliminate some of the contours which convey perceptually less information. In the second part, the parameters of the centipede are extracted by one dimensional profile

of the edge along the normal direction. The output of the first part is a binary image and coded by differential chain code followed by Huffman coding. Model parameters are fit to polynomials of some order. An approximation to the original image from the model parameters is obtained by minimizing the hybrid energy functional.

The performance of the centipede model is evaluated by both quantitatively and qualitatively. Compression ratio is up to 180:1 for synthetic images and 10:1-100:1 for real images. Reconstructed images are evaluated both quantitatively with NMSE (normalized mean square error), SNR (signal-to-noise ratio) and PSNR (peak-to-peak SNR) and qualitatively with visual appearance of artifacts. We have experimentally shown that the proposed model preserves perceptually important features even at the high compression ratios.

SUGGESTIONS

The width in the centipede model is a rough estimate of the edge scale. This causes some edge contours to being blurred. Since surface reconstruction is done on $\lambda\tau$-space, a good estimation can be obtained by determining appropriate (λ,τ) pair for each edge element. It is sure that this will enhance the reconstructed image requiring extra computation.

The proposed contour selection algorithm can be improved by utilizing the region segmentation information in such a way that a contour which does not belong to at least two distinct region boundaries can be eliminated.

Since the improvement in coding of binary images will also improve the compression performance, the block coding of the edge segment is required to go down the average bit rate per pixel below 1 bpp.

One of the most important advantages of edge-based coding techniques is the ability of processing even on the compressed image directly. Since the compressed image contains information on some aspect of the image such as edges and scales, the features can be used for further processing without decompressing.

We have currently been applying the principal component analysis directly on the edge information to face images for detection purpose. We try to prove that eigenedge decomposition of the face edges are less sensitive to the lighting conditions.

REFERENCES

[1] Ahmet M. Eskicioglu, P. S. Fisher, "Image Quality Measures and Their Performance," IEEE Transactions on Communications, Vol. 43, No. 12, pp. 2959-2965, December 1995.

[2] M. Gökmen, A. K. Jain, "λτ-Space representation of images and generalized edge detector," appear to IEEE Transactions on Pattern Analysis and Machine Intelligence.

[3] A. K. Jain, "A fast Karhunen-Loeve Transform for a class of stochastic processes," IEEE Transaction on Communications, vol. COM-24, pp. 1023-1029, September, 1976.

[4] Ahmed, N., Natarajan, T., and Rao, K. R., "Discrete cosine transform," IEEE Transactions on Computer, vol. C-23, pp. 90-93, January, 1974.

[5] N. M. Nasrabadi, R. A. King, "Image Coding using Vector Quantization : A Review," IEEE Transaction on Communications, Volume 36, number 8, pp. 957-971, August 1988.

[6] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications, vol. 28, pp. 84-95, January, 1980.

[8] W. F. Schreiber, R. R. Buckley, "A two-channel picture coding system : II - adaptive companding and color coding," IEEE Transactions on Communications, Vol. COM-29, pp. 1849-1858, December, 1981.

[9] John W. Woods, Subband Image Coding, Kluwer Academic Publishers, 1991.

[10] Yuval Fisher, "Fractal Image Compression : Theory and Application," Springer-Verlag, 1995.

[11] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Communications on Pure and Applied Mathematics, vol. XLI, pp. 990-996, 1988.

[12] A. Cohen, I. Daubechies, J.C. Feauveau, "Biorthogonal bases of compactly supported wavelets," Communications on Pure and Applied Mathematics, vol. XLV, pp. 485-560, 1992.

[13] O. Riol, "Simple regularity criteria for subdivision schemes," SIAM Journal of Mathematical Analysis, vol. 23, pp. 1544-1576, November 1992.

[14] I. Daubechies, "Orthonormal bases of compactly supported wavelets II. variations on theme," SIAM Journal of Mathematical Analysis, vol. 24, pp. 499-519, March, 1993.

[15] M. Kunt, A. Ikonomopoulos, M. Kocker, "Second-Generation Image-Coding Techniques," Proceedings of the IEEE, Vol. 73, No. 4, pp. 549-574, April, 1985.

[16] M. Kunt, M. Benard, R. Leonardi, "Recent Results in high-compression image coding," IEEE Transactions on Circuits and Systems, Vol. CAS-34, No. 11, pp. 1306-1336, November, 1987.

[17] D. Graham, "Image transmission by two-dimensional contour coding," Proceedings of IEEE, Vol. 55, No. 3, March, 1967.

[18] H. Freeman, "Computer processing of line drawing images," Computing Surveys, Vol.6, pp. 57-97, March, 1974.

[19] T. Kaneko, M. Okudaira, "Encoding of Arbitrary Curves Based on the Chain Code Representation," IEEE Transactions on Communications, Vol. COM-33, No. 7, pp. 697-706, July, 1985.

[20] S. Carlson, "Sketch based coding of gray level images," Signal Processing, Vol. 15, pp. 57-83, 1988.

[21] J. H. Elder, S. W. Zucker, "Scale Space Localization, Blur, and Contour-Based Image Coding," Proceedings of Int. Conf. on CVPR, pp. 27-34, 1996.

[22] T. Acar, M. Gökmen, "Image Coding Using Weak Membrane Model Of Images," In Visual Communications and Image Processing`94, pp. 1221-1230, Chicago, Illinois, 1994.

[23] A. Gençata, T. Acar, M. Gökmen, "Image Compression Using Weak Membrane Model," National Conference on Signal Processing and its Applications, pp. 117-121, Nevsehir, 1995.

[24] M. Gökmen, I. Ersoy, A. K. Jain, "Compression of fingerprint images using hybrid image model," Proceedings of Int. Conf. on IP, pp. 395-398, Lozan, 1996.

[25] B. Kurt, M. Gökmen, "Image Compression Based on Centipede Model," accepted to National Conference on Signal Processing and its Applications`97, Bodrum, Manisa.

[26] John A. Robinson, "Image Coding with Ridge and Valley Primitives," IEEE Transactions on Communications, Vol. 43, No. 6, pp. 2095-2102, June, 1995.

[27] R. Sibson, "A brief description of national neighbor interpolant," Interpreting Multivariate Data, Wiley Series in Probability and Mathematical Statistics, pp. 21-37, 1981.

[28] U. Y. Desai, M. M. Mizuki, I. Masaki and K. P. Horn, "Edge and Mean based Image Compression," Technical Report, Massachusettes Institute Of Technology, Artificial Intelligence Laboratory, Report No. 1584, November, 1996.

[29] Antoon M. van Dijk, Jean-Bernard Martens, "Feature-Based Image Compression With Steered Hermite Transforms," Proceedings of Int. Conf. on Image Processing, pp. 344-347, Lozan, 1996.

[30] Philippe Salembier, Patrick Brigger, J. R. Casas, M. Pardás, "Morphological Operators for Image and Video Compression," IEEE Transactions on Image Processing, Special Issue on Nonlinear Image Processing, Vol. 5, No. 6, pp. 881-898, June, 1996.

[31] P. Salembier, J. Serra, "Flat zones filtering, connected operators and filters by reconstruction," IEEE Transactions on Image Processing, Vol. 3, No. 8, pp. 1153-1160, August, 1995.

[32] P. Salembier, M. Pardás, "Hierarchical morphological segmentation for image sequence coding", IEEE Transactions on Image Processing, Vol. 3, No. 5, pp. 639-651, September, 1994.

[33] P.A. Maragos, R. W. Schafer, "Morphological skeleton representation and coding of binary images," IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. 34, No. 5, pp. 1228-1244, October, 1986.

[34] M.J. Biggar, O.J. Morris, A.G. Constantinides, "Segmented-image coding : performance comparison with the discrete cosine transform," IEE Proceedings, Vol. 135, No. 2, pp. 121-132, April, 1988.

[35] H. Sanderson, G. Crebbin, "Region-based image coding using polnomial intensity functions," IEE Proceedings on Vision and Image Processing, Vol. 143, No. 1, pp. 15-23, February, 1996.

[36] M. Gökmen, C. C. Li, "Multiscale Edge Detection Using First Order R-filter," Int. Conf. on Pattern Recognition, The Hague, Netherlands, pp. 307-310, 1992.

[37] Tony Lindeberg, "Scale-space theory : A basic tool for analysing structures at different scale," Journal of Applied Statistics, vol. 21, no. 2, pp. 225-270, 1994.

[38] Tony Lindeberg, "Edge and ridge detection with Automatic Scale Selection," Proceedings of International Conference on CVPR, pp. 465-470, 1996.

[39] C.C. Jay Kuo, Bernard C. Levy, "A two-level four-color SOR method," SIAM Journal of Numerical Analysis, Vol. 26, No. 1, pp. 129-151, February, 1989.

[40] Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley Publishing Co., Seventh Edition, 1993.

[41] W. Hackbush, Multigrid methods and applications, Springer Verlag, Berlin, 1985.

[42] J. F. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.8(6):679-698, 1986.

BIOGRAPHY

Binnur Kurt was born on July 11th1973 in Istanbul. He graduated from Üsküdar Halide Edip Adivar Lycee on 1989. He entered to Control and Computer Engineering Department of Istanbul Technical University and graduated on June 1995. His undergraduate thesis was edge detection, surface reconstruction and image restoration by using anisotropic diffusion equation. He has been a research assistant at Computer Science Department, Electrical and Electronics Faculty of Istanbul Technical University since 1995. His current research areas are image processing, computer vision. He is a student member of IEEE.

T.C. YÜKSEKÖĞRETİM KURULU