NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

APPLICATIONS OF IMAGE PROCESSING

Graduation Project
EE- 400

Student: Barış Önal (980176)

Supervisor: Assoc. Prof. Dr Adnan Khashman

Lefkoşa-2003

TABLE OF CONTENTS

ACKNOWLEDGMENT i

ABSTRACT ii

INTRODUCTION 1

1. IMAGE PROCESSING BACKGROUND 2

1.1 Overview 2

1.2 Definition 2

1.2.1 Signals and Systems 2

1.2.2 Analog Signals 2

1.2.3 Digital Signals 3

1.2.3.1 Real and Complex Valued Signals 4

1.2.3.2 Discrete Time Cosine Signal 4

1.2.3.3 Sinusoids 5

1.2.3.4 Unit Sample 5

1.2.4 Image Processing 5

1.2.5 Digital Image Analysis 7

1.3 History 8

1.4 Future 8

1.5 Importance 8

1.6 Summary 8

2. IMAGE PROCESSING TECHNIQUES 9

2.1 Overview 10

2.2 Image Compression 10

2.2.1 Why do we need compression 10

2.2.2 What are the principles behind compression? 11

2.2.3 What are the different classes of compresion? 12

2.2.4 What does a typical image coder look like? 12

2.2.4.1 Source Encoder 13

2.2.4.2 Quantizer 13

2.2.4.3 Entropy Encoder 13

2.2.5 JPEG: DTC- Based Image Coding Standard 14

2.2.6 Wavelets and Image Compression 16

2.2.6.1 What is a Wavelet Transform? 16

2.2.6.2 Why Wavelet-based Compression? 17

2.2.6.3 Subband Coding 18

2.2.6.4 From Subband to Wavelet Codind 19

2.3 Image Restoration 20

2.4 Image Recognition 22

2.4.1 Pattern recognition methods in image understanding 22

2.4.2 Contextual image classification 22

2.5 Edge Detection 29

2.5.1 Representing Lines 29

2.5.2 Extracting Edges from Images 31

2.5.3 Detecting Edge Points 31

2.5.3.1 Gradient based methods 31

2.5.3.2 Implementation 34

2.5.3.3 Second Order Methods 36

2.6 Summary 42

3. APPLICATIONS OF IMAGE PROCESSING 43

3.1 Overview 43

3.2 Image Processing For medical Applicaitons 43

3.3 Image Processing For Space Exploration 44

3.4 Geographical Information System Applications 45

3.5 Weather Graphics Applications 46

3.6 Weather Data Applications 46

3.7 Climate Analysis Tools 46

3.8 Miliraty 47

3.9 Summary 47

4. APPLICATION OF IMAGE PROCESSING TO MINE WARFARE SONAR 48

4.1 Overview 48

4.2 Introduction 48

4.3 Sector-scan Sonar 50

4.3.1 General Concepts 50

4.3.2 Enhancement 51

4.3.3 Segmentation 55

4.3.4 Computer Aided Detection 55

4.3.5 Computer Aided Classification 57

4.4 Side-Scan Sonar 57

4.4.1 General Concepts 58

4.4.2 Enhancement 58

4.4.3 Segmentation 59

4.4.4 Computer-Aided Detection 61

4.4.5 Computer-Aided Classification 62

4.5 The AMI Project 63

4.5.1 General Concepts 63

4.5.2 Enhancement 64

4.5.3 Segmentation 64

4.6 Summary 65

CONCLUSION 66

REFERENCES 67

ACKNOWLEDGMENTS

First I want to thank Assoc. Prof. Dr Adnan Khashman to be my supervisor and for his endless helps and courages to me. With his deep knowledge of Image Processing I manage to overcome the difficulties during my research.

I also want to thank to my friends at the faculty especially Kemal, Boran and Kamil for thier endless courages and helps. I want to thank to my friends in the company that I'm working. Especially Akar, Dinç,Melek and Erkan and finally to boss Mr. Mustafa and his wife Mrs.Göksel for their patience and helps to me.

Finally most impotant thing, my family. I would like to thank to my parents and my sisters for their endless helps and supports. Without them I wouldn't manage to do this project. Thank you. I wish you a healthy and good life now and ever. Dear my mother and father I proud of to being your son.

INTRODUCTION

Image Processing is becoming a widely acknowledged technology. Many everyday processes use automated vision systems, all of which rely upon image processing techniques. Image Processing, in general terms, refers to the manipulation, improvement and analysis of pictorial information. In this case, pictorial information means a two-dimensional visual image. Digital image processing is concerned with the improvement of quality of a picture that is digitally represented, as that represented in the digital computer.

From the 1960's through today, the evolution of the digital computer has certainly been largely responsible for enabling the proliferation of digital image processing applications. Costly mainframe computers are no longer a requirement of the digital image processing equation like they were in the 1960's. The advent of microprocessors, leading to the personal computer, has allowed stand-alone digital image processing applications to become viable.

In this project, applications of image processing is discussed. The aim of the project is to inform you about the real life applications of image processing and new techniques used in image processing.

The first chapter represents the background of image processing. This chapter includes history, basically description of image processing, importance and future of image processing.

Chapter two is assigned for the techniques that are used in image processing. These techniques can be sorted as;

Image compression, image restoration, image recognition and edge detection.

Chapter three represents the applications of image processing in real life.

Chapter four is assigned to applications of image processing to Mine Warfare Sonar Systems

CHAPTER ONE : IMAGE PROCESSING BACKGROUND

1.1 Overview

In this chapter the definition of image processing, history, future and importance in real life will be present.

1.2 Definition

1.2.1 Signals and Systems

Whether analog or digital, information is represented by the fundamental quantity in electrical engineering: the signal. Stated in mathematical terms, a signal is merely a function. Analog signals are continuous-valued; digital signals are discrete-valued. The independent variable of the signal could be time (speech, for example),space (images), or the integers.

1.2.2 Analog Sinals

Analog signals are usually signals defined over continuous independent variable(s). SPEECH is produced by your vocal cords exciting acoustic resonances in your vocal tract. The result is pressure waves propagating in the air, and the speech signal thus corresponds to a function having independent variables of space and time and a value corresponding to air pressure: sxt (Here we use vector notation x to denote spatial coordinates). When you record someone talking, you are evaluating the speech signal at a particular spatial location, x0 say. An example of the resulting waveform sx_0t is shown in Figure 1.1 [1].

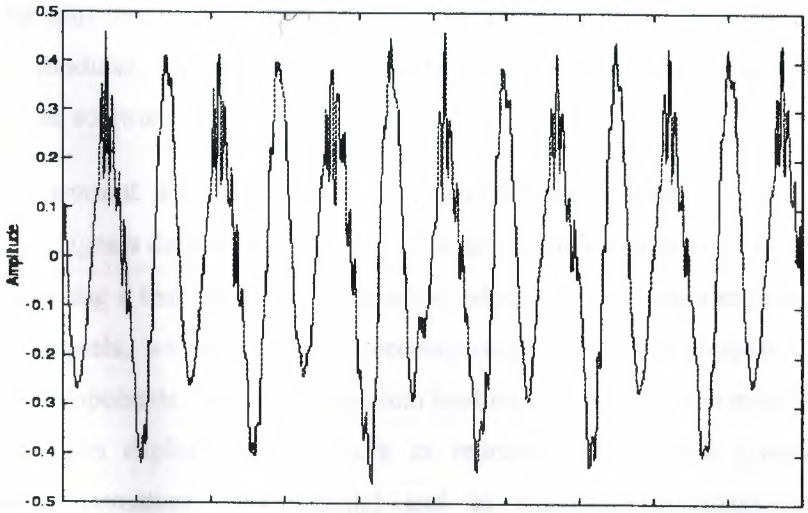


Figure 1.1 A speech signal's amplitude relates to tiny air pressure variations. Shown is a recording of the vowel "e" (as in "speech")

1.2.3 Digital Signals

The word "digital" means discrete-valued and implies the signal has an integer-valued independent variable. Digital information includes numbers and symbols (characters typed on the keyboard, for example). Computers rely on the digital representation of information to manipulate and transform information. Symbols do not have a numeric value, and each is represented by a unique number. The ASCII character code has the upper- and lowercase characters, the numbers, punctuation marks, and various other symbols represented by a seven-bit integer. For example, the ASCII code represents the letter a as the number 97 and the letter A as 65.

Signals can be represented by discrete quantities instead of as a function of a continuous variable. These discrete time signals do not necessarily have to take real number values. Many properties of continuous valued signals transfer almost directly to the discrete domain.

So far, we have treated what are known as **analog** signals and systems. Mathematically, analog signals are functions having continuous quantities as their independent variables, such as space and time. Discrete-time signals are functions defined on the integers; they are sequences. This result is important because discrete-time signals can be manipulated by systems instantiated as computer programs. Subsequent modules describe how virtually all analog signal processing can be performed with software.

As important as such results are, discrete-time signals are more general, encompassing signals derived from analog ones and signals that aren't. For example, the characters forming a text file form a sequence, which is also a discrete-time signal. As with analog signals, we seek ways of decomposing real-valued discrete-time signals into simpler components. With this approach leading to a better understanding of signal structure, we can exploit that structure to represent information (create ways of representing information with signals) and to extract information (retrieve the information thus represented). For symbolic-valued signals, the approach is different: We develop a common representation of all symbolic-valued signals so that we can embody the information they contain in a unified way. From an information representation perspective, the most important issue becomes, for both real-valued and symbolic-valued signals, efficiency.

1.2.3.1 Real and Complex Valued Signals

The discrete-time signals as stem plots to emphasize the fact they are functions defined only on the integers. We can delay a discrete-time signal by an integer just as with analog ones. A delayed unit sample has the expression δ□n-m, and equals one when n=m.

1.2.3.2 Discrete Time Cosine Signal

Figure 1.2 shows The Discrete Time Cosine Signal [1]

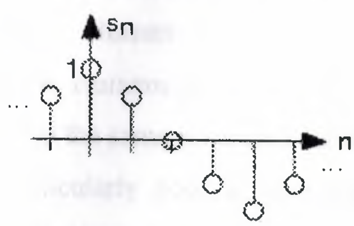


Figure 1.2 The Discrete Time Cosine Signal

1.2.3.3 Sinusoids

As opposed to analog complex exponentials and sinusoids that can have their frequencies be any real value, frequencies of their discrete-time counterparts yield unique waveforms only when f lies in the interval minus infinity to plus infinity.

1.2.3.4 Unit Sample

The second-most important discrete-time signal is the **unit sample**, which is defined in figure 1.3 [1]

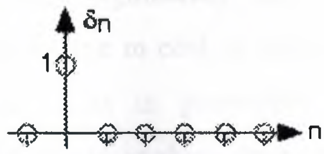


Figure 1.3 Unit Sample

1.2.4 Image Processing

Image processing and analysis is a combination of the visual enhancement of an image and the numerical evaluation of some aspect of the acquired image that would not be apparent in its analog form. The processing of a digital image refers to its actual refinement procedure. The analysis of a digital image encompasses the translation of the newly improved image's features into useful data. The necessity for this technology is due primarily to the difference between human visions and "computer vision." This contrast is most pronounced when comparing the type of information obtained from images and the methods used. Human vision is fundamentally qualitative and comparative, but not quantitative. Humans judge the relative size and shape of objects by mentally manipulating them to the same orientation and overlapping them to perform a comparison. Humans are particularly poor at assessing the color or brightness of features within images without direct comparison by positioning them adjacently. Progressive variations in brightness are typically dismissed as being representative of fluctuations in illumination, which the human visual system compensates automatically. In a gray scale image where the image is an array of intensities in a two-dimensional (2-D) space, the variation of intensity can provide you with a sense of texture, trend, edges, and anomalies on a surface. In a controlled environment, the variation of intensity is not merely changes in illumination, but an index of physical characteristics of a material.

Digital image analysis is a powerful method for gathering information. Relying on the convenience of computers, image processing and analysis methods have been used extensively in other disciplines for some time. However, their potential impact has started to be recognized in civil engineering only in the past few years. The advancement of technology and decline in cost of computers have provided numerous opportunities for significant advances in geotechnical and materials engineering. Researchers have applied digital image analysis in geotechnical engineering to study: cohesion less soil fabric, membrane penetration, mineral
phase percentages in granular rocks, pavement cracking, particular shape, and morphological analysis of geotextiles. This field of study continues to grow in its form of applications as equipment becomes cheaper and better.

Computer based digital image analysis may involve aspects of mathematical morphology, stereology, and image processing. After thirty years of development, mathematical morphology has become one of the major tools in 2-D image processing. Mathematical morphology is the use of systematic numerical algorithms that extract qualitative and quantitative information from digital spatial data. Mathematical morphology also discards excess information in a controlled way. The removal of irrelevant detail makes images easier for analysis. Stereological methods are precise tools for obtaining quantitative information about 3-D microscopic structures, based mainly on observations made on sections of a specimen. It has been demonstrated that without stereology, material science cannot evolve into a truly quantitative science. The information extracted using these methods may lead to a better understanding of an observed structure and related phenomena.

1.2.5 Digital Image Analysis

After acquisition and storage, the digital image can be subjected to a number of processes that require handling of the image matrix. A digital image is a two dimensional (2-D) matrix (or array, see figure 1.4 [8]) where its elements are called pixels (picture elements). The pixel values are a light intensity function f(x, y) where x and y are denote spatial coordinates and the function "f" is a measure of brightness (or gray level) or color of the image at that point. In a gray scale image the value of 0 denotes black (or lowest intensity) and the value of 255 denotes white (or highest intensity). A pixel usually denotes a dot on a computer display or monitor depending on the screen resolution.

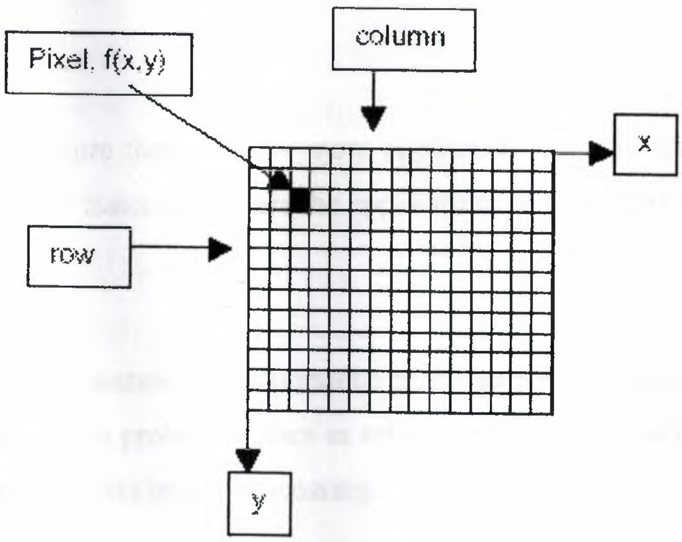


Figure 1.4 Matrix representation of an image

Image processing is a manipulation of matrices in the form of algorithms. Most processing functions can be implemented in a software application. The only reason to have specialized image-processing hardware is the need for speed in some applications. However, with high-speed desktop computers and storage devices becoming so accessible and affordable, specialized hardware is often not necessary. Today's image processing systems are a blend of off-the-shelf computers and specialized image processing accessories with overall operation being orchestrated by software running on the host computer.

1.3 History

In early 1960's pursuing lunar science program in NASA and a ranger program in NANA. In late 1960's image processing started to used by medical diagnostic imaging filed. Such as X-ray, computed tomography (CT), magnetic resonance imagery (MRI), positron emission tomography (PET) and ultrasound imaging.

In early 1970's Land sat (earth image) is used to analyse agricultural land-use and meteorological imagery.

In 1980's image processing is used for biological image, television broadcasting, military uses and to automate manufacturing processes.

1.4 Future

In think in near future there will be more application areas of image processing. With more powerful and faster computers the applications will be more wider.

1.5 Importance

Nowadays when understand the importance of image processing more. Such a way that we can send more probes to Mars or other planets, collect information and we can analyse the data with image processing.

1.6 Summary

In this chapter, definition of image processing, signals and systems, history, future and importance of image processing was discussed.

In the next chapter, image processing techniques will be discussed.

CHAPTER TWO: IMAGE PROCESSING TECHNIQUES

2.1 Overview

In this chapter we will discuss the image processing techniques. These techniques are;

Image Compression, image restoration, image enhancement, image recognition and edge detection.

2.2 Image Compression

Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications have not only sustained the need for more efficient ways to encode signals and images but have made compression of such signals central to storage and communication technology.

For still image compression, the `Joint Photographic Experts Group' or JPEG standard has been established by ISO (International Standards Organization) and IEC (International Electro-Technical Commission). The performance of these coders generally degrades at low bit-rates mainly because of the underlying block-based Discrete Cosine Transform (DCT) scheme. More recently, the wavelet transform has emerged as a cutting edge technology, within the field of image compression. Wavelet-based coding provides substantial improvements in picture quality at higher compression ratios. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression, as discussed later, have been developed and implemented. Because of the many advantages, the top contenders in the upcoming JPEG-2000 standard are all wavelet-based compression algorithms.

2.2.1 Why do we need compression?

The figures in Table 2.1 show the qualitative transition from simple text to full-motion video data and the disk space, transmission bandwidth, and transmission time needed to store and transmit such uncompressed data.

Table 2.1. Multimedia data types and uncompressed storage space, transmission bandwidth, and transmission time required. The prefix kilo- denotes a factor of 1000 rather than 1024.

Multimedia Data	**Size/Duration**	**Bits/Pixel or Bits/Sample**	**Uncompressed Size (B for bytes)**	**Transmission Bandwidth (b for bits)**	**Transmission Time (using a 28.8K Modem)**
A page of text	11" x 8.5"	Varying resolution	4-8 KB	32-64 Kb/page	1.1 - 2.2 sec
Telephone quality speech	10 sec	8 bps	80 KB	64 Kb/sec	22.2 sec
Grayscale Image	512 x 512	8 bpp	262 KB	2.1 Mb/image	1 min 13 sec
Color Image	512 x 512	24 bpp	786 KB	6.29 Mb/image	3 min 39 sec
Medical Image	2048 x 1680	12 bpp	5.16 MB	41.3 Mb/image	23 min 54 sec
SHD Image	2048 x 2048	24 bpp	12.58 MB	100 Mb/image	58 min 15 sec
Full-motion Video	640 x 480, 1 min (30 frames/sec)	24 bpp	1.66 GB	221 Mb/sec	5 days 8 hrs

The examples above clearly illustrate the need for sufficient storage space, large transmission bandwidth, and long transmission time for image, audio, and video data. At the present state of technology, the only solution is to compress multimedia data before its storage and transmission, and decompress it at the receiver for play back. For example, with a compression ratio of 32:1, the space, bandwidth, and transmission time requirements can be reduced by a factor of 32, with acceptable quality.

2.2.2 What are the principles behind compression?

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. **Redundancy reduction** aims at removing duplication from the signal source (image/video). **Irrelevancy reduction** omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS). In general, three types of redundancy can be identified:

- **Spatial Redundancy** or correlation between neighboring pixel values.
- **Spectral Redundancy** or correlation between different color planes or spectral bands.
- **Temporal Redundancy** or correlation between adjacent frames in a sequence of images (in video applications).

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Since we will focus only on still image compression, we will not worry about temporal redundancy.

2.2.3 What are the different classes of compression techniques?

Two ways of classifying compression techniques are mentioned here.

(a) Lossless vs. Lossy compression: In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Under normal viewing conditions, no visible loss is perceived (visually lossless).

(b) Predictive vs. Transform coding: In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding. Transform coding, on the other hand, first transforms the image from its spatial domain representation to a different type of representation using some well-known transform and then codes the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expense of greater computation.

2.2.4 What does a typical image coder look like?

A typical lossy image compression system is shown in Fig. 2.1. It consists of three closely connected components namely (a) Source Encoder (b) Quantizer, and (c) Entropy Encoder. Compression is accomplished by applying a linear transform to decorrelate the image data, quantizing the resulting transform coefficients, and entropy coding the quantized values.

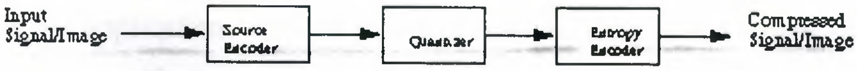


Figure 2.1 A Typical Lossy Signal/Image Encoder

2.2.4.1 Source Encoder (or Linear Transformer)

Over the years, a variety of linear transforms have been developed which include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and many more, each with its own advantages and disadvantages.

2.2.4.2 Quantizer

A quantizer simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it is a lossy process and is the main source of compression in an encoder. Quantization can be performed on each individual coefficient, which is known as Scalar Quantization (SQ). Quantization can also be performed on a group of coefficients together, and this is known as Vector Quantization (VQ). Both uniform and non-uniform quantizers can be used depending on the problem at hand. For an analysis on different quantization schemes.

2.2.4.3 Entropy Encoder

An entropy encoder further compresses the quantized values losslessly to give better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. The most commonly used entropy encoders are the Huffman encoder and the arithmetic encoder, although for applications requiring fast execution, simple run-length encoding (RLE) has proven very effective. It is important to note that a properly designed quantizer and entropy encoder are absolutely necessary along with optimum signal transformation to get the best possible compression.

2.2.5 JPEG : DCT-Based Image Coding Standard

The idea of compressing an image is not new. The discovery of DCT in 1974 is an important achievement for the research community working on image compression. The DCT can be regarded as a discrete-time version of the Fourier-Cosine series. It is a close relative of DFT, a technique for converting a signal into elementary frequency components. Thus DCT can be computed with a Fast Fourier Transform (FFT) like algorithm in *O(n log n)* operations. Unlike DFT, DCT is real-valued and provides a better approximation of a signal with fewer coefficients. The DCT of a discrete signal *x(n), n=0, 1, .. , N-1* is defined as:

$$X(u) = \sqrt{\frac{2}{N}} C(u) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(2n+1)u\pi}{2N}\right)$$

where, $C(u) = 0.707$ for $u = 0$ and $= 1$ otherwise.

In 1992, JPEG established the first international standard for still image compression where the encoders and decoders are DCT-based. The JPEG standard specifies three modes namely sequential, progressive, and hierarchical for lossy encoding, and one mode of lossless encoding. The `baseline JPEG coder' which is the sequential encoding in its simplest form, will be briefly discussed here. Fig. 2.2(a) and 2.2(b) show the key processing steps in such an encoder and decoder for grayscale images. Color image compression can be approximately regarded as compression of multiple grayscale images, which are either compressed entirely one at a time, or are compressed by alternately interleaving 8x8 sample blocks from each in turn.

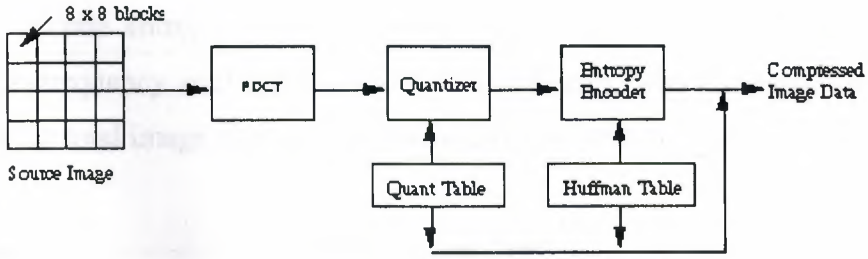


Figure 2.2(a) JPEG Encoder Block Diagram

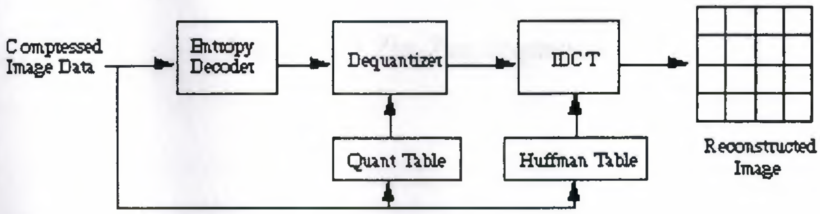


Figure 2.2(b) JPEG Decoder Block Diagram

The DCT-based encoder can be thought of as essentially compression of a stream of 8x8 blocks of image samples. Each 8x8 block makes its way through each processing step, and yields output in compressed form into the data stream. Because adjacent image pixels are highly correlated, the `forward' DCT (FDCT) processing step lays the foundation for achieving data compression by concentrating most of the signal in the lower spatial frequencies. For a typical 8x8 sample block from a typical source image, most of the spatial frequencies have zero or near-zero amplitude and need not be encoded. In principle, the DCT introduces no loss to the source image samples; it merely transforms them to a domain in which they can be more efficiently encoded.

After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a carefully designed 64-element Quantization Table (QT). At the decoder, the quantized values are multiplied by the corresponding QT elements to recover the original unquantized values. After quantization, all of the quantized coefficients are ordered into the "zig-zag" sequence as shown in Fig. 2.3. This ordering helps to facilitate entropy encoding by placing low-frequency non-zero coefficients before high-frequency coefficients. The DC coefficient, which contains a significant fraction of the total image energy, is differentially encoded.

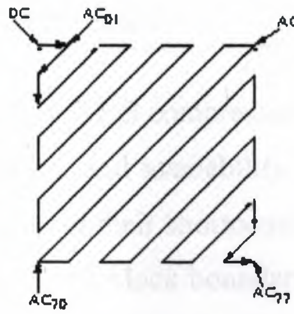


Figure 2.3 Zig-Zag sequence

Entropy Coding (EC) achieves additional compression losslessly by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. The JPEG proposal specifies both Huffman coding and arithmetic coding. The baseline sequential codec uses Huffman coding, but codecs with both methods are specified for all modes of operation. Arithmetic coding, though more complex, normally achieves 5-10% better compression than Huffman coding.

2.2.6 Wavelets and Image Compression

2.2.6.1 What is a Wavelet Transform ?

Wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). The

Discrete Wavelet Transform of a finite length signal $x(n)$ having N components, for example, is expressed by an $N \times N$ matrix.

2.2.6.2 Why Wavelet-based Compression?

Despite all the advantages of JPEG compression schemes based on DCT namely simplicity, satisfactory performance, and availability of special purpose hardware for implementation, these are not without their shortcomings. Since the input image needs to be ``blocked," correlation across the block boundaries is not eliminated. This results in noticeable and annoying ``blocking artifacts" particularly at low bit rates as shown in Fig. 2.4. Lapped Orthogonal Transforms (LOT) attempt to solve this problem by using smoothly overlapping blocks. Although blocking effects are reduced in LOT compressed images, increased computational complexity of such algorithms do not justify wide replacement of DCT by LOT.

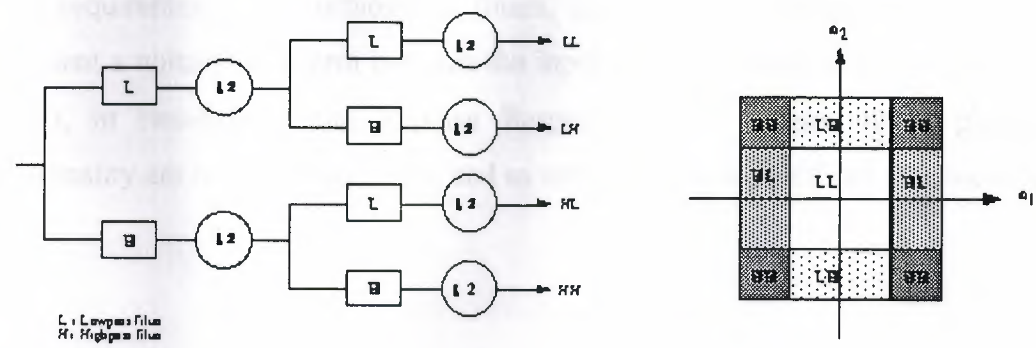
(a) (b)

Figure 2.4(a) Original Lena Image, and **(b)** Reconstructed Lena with DC component only, to show blocking artifacts

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications wavelet-based schemes (also referred as subband coding) outperform other coding schemes like the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. In addition, they are better matched to the HVS characteristics. Because of their inherent multiresolution nature, wavelet coding schemes are especially suitable for applications where *scalability* and *tolerable degradation* are important.

2.2.6.3 Subband Coding

The fundamental concept behind Subband Coding (SBC) is to split up the frequency band of a signal (image in our case) and then to code each subband using a coder and bit rate accurately matched to the statistics of the band. SBC has been used extensively first in speech coding and later in image coding because of its inherent advantages namely variable bit assignment among the subbands as well as coding error confinement within the subbands.



(a) (b)

Figure 2.5(a) Separable 4-subband Filterbank, and 2.**5(b)** Partition of the Frequency Domain

Woods and O'Neil used a separable combination of one-dimensional Quadrature Mirror Filterbanks (QMF) to perform a 4-band decomposition by the row-column approach as shown in Fig. 2.5(a). Corresponding division of the frequency spectrum is shown in Fig. 2.5(b). The process can be iterated to obtain higher band decomposition filter trees. At the decoder, the subband signals are decoded, upsampled and passed through a bank of synthesis filters and properly summed up to yield the reconstructed image.

2.2.6.4 From Subband to Wavelet Coding

Over the years, there have been many efforts leading to improved and efficient design of filterbanks and subband coding techniques. Since 1990, methods very similar and closely related to subband coding have been proposed by various researchers under the name of *Wavelet Coding* (WC) using filters specifically designed for this purpose. Such filters must meet additional and often conflicting requirements. These include short impulse response of the analysis filters to preserve the localization of image features as well as to have fast computation, short impulse response of the synthesis filters to prevent spreading of artifacts (ringing around edges) resulting from quantization errors, and linear phase of both types of filters since nonlinear phase introduce unpleasant waveform distortions around edges. Orthogonality is another useful requirement since orthogonal filters, in addition to preservation of energy, implement a unitary transform between the input and the subbands. But, as in the case of 1-D, in two-band Finite Impulse Response (FIR) systems linear phase and orthogonality are mutually exclusive, and so orthogonality is sacrificed to achieve linear phase.

2.3 Image Restoration

Digitized images typically suffer from a range of imperfections including geometric distortion, nonuniform contrast, and noise. These all introduce errors into $\rho(\mathbf{r}, t)$ unless steps are taken to restore the image to its ``ideal" state. Some geometric distortions are caused by defects in the microscope optics, but most are introduced in later stages of digitization. Video signals adhering to the RS-170 standard, for example, consist of rectangular pixels with a 4:3 aspect ratio. A circle imaged by a video camera appears uniaxially distorted into an ellipse when digitized and displayed by a computer, whose pixels are square. The analysis routines we describe below are most easily implemented for images consisting of square pixels. While many digitizing boards attempt to correct for uniaxial distortion, they often leave a residual anisotropy of a few percent. Both uniform and nonuniform geometric distortions can be measured by creating images of standard grids, identifying features in the images with features in the standards, and determining how far the image features are displaced from their ideal locations in an undistorted image. The algorithms we describe below for locating colloidal spheres also are useful for locating features in such calibration standards. Standard image processing texts describe algorithms for measuring apparent distortions in the calibration grid image and removing the distortion by spatial warping. Many image processing packages such as IDL include efficient implementations.

Contrast gradients can arise from nonuniform sensitivity among the camera's pixels. More significant variation often is due to uneven illumination. Long wavelength modulation of the background brightness complicates the design of criteria capable of locating spheres' images throughout an entire image. Subtracting off such a background is not difficult if the features of interest are relatively small and well separated as is frequently the case for colloidal images. Under these circumstances, the background is reasonably well modeled by a boxcar average over a region of extent $2w+1$, where w is an integer larger than a single sphere's apparent radius in pixels, but smaller than an intersphere separation:

$$A_w(x, y) = \frac{1}{(2w+1)^2} \sum_{i,j=-w}^{w} A(x+i, y+j). \tag{2}$$

While long-wavelength contrast variations waste the digital imaging system's dynamic range, noise actually destroys information. Coherent noise from radio frequency interference (RFI) can be removed with Fourier transform techniques but is best avoided with proper electrical shielding. Digitization noise in the CCD camera and the frame grabber, however, is unavoidable. Such noise tends to be purely random with a correlation length $\lambda_n \approx 1$ pixel. Convolving an image $A(x,y)$ with a Gaussian surface of revolution of half width λ_n strongly suppresses such noise without unduly blurring the image:

$$A_{\lambda_n}(x,y) = \frac{1}{B} \sum_{i,j=-w}^{w} A(x+i, y+j) \exp\left(-\frac{i^2+j^2}{4\lambda_n^2}\right), \quad (3)$$

with normalization $B = \left[\sum_{i=-w}^{w} \exp\left(-\frac{i^2}{4\lambda_n^2}\right)\right]^2$.

The difference between the noise-reduced and background images is an estimate of the ideal image. Since both eqn. (2) and eqn. (3) can be implemented as convolutions of the image $A(x,y)$ with simple kernels of support $2w + 1$, we can compute both in a single step with the convolution kernel

$$K(i,j) = \frac{1}{K_0}\left[\frac{1}{B}\exp\left(-\frac{i^2+j^2}{4\lambda_n^2}\right) - \frac{1}{(2w+1)^2}\right]. \quad (4)$$

The normalization constant

$K_0 = \frac{1}{B}\left[\sum_{i=-w}^{w} \exp\left(-\frac{i^2}{2\lambda_n^2}\right)\right]^2 - \frac{B}{(2w+1)^2}$ facilitates comparison among images filtered with different values of w. The correlation length of the noise generally is not used as input parameter, with λ_n instead being set to unity. The efficacy of the filter can be judged from the example in Fig. 1(b).

In practice, the image $A(x,y)$ must be cast from an array of bytes to a higher precision data format, such as a floating point array, before convolution. This scaling, together with the actual convolution operation can be implemented in hardware with an array processor such as the Data Translation DT-2878. Further speed enhancement is realized by decomposing the circularly symmetric two-dimensional convolution kernel $K(i,j)$ into four one-dimensional convolution kernels, so that filtering can be computed in $O(w)$ operations rather than. $O(\omega^2)$.

2.4 Image Recognition

2.4.1 Pattern recognition methods in image understanding

- Pattern recognition methods frequently appear in image understanding[8] .
- Classification-based segmentation of multispectral images (satellite images, magnetic resonance medical images, etc.) is a typical example.
- Supervised methods are used for classification, a priori knowledge is applied to form a training set.
- In the image understanding stage, feature vectors derived from local multispectral image values of image pixels are presented to the classifier which assigns a label to each pixel of the image.
- Image understanding is then achieved by pixel labeling.
- Thus the understanding process segments a multispectral image into regions of known labels.
- Training set construction, and therefore human interaction, is necessary for supervised classification methods, but if unsupervised classification is used, training set construction is avoided.
- As a result, the clusters and the pixel labels do not have a one-to-one correspondence with the class meaning.

- This implies the image is segmented, but labels are not available to support image understanding.
- Fortunately, a priori information can often be used to assign appropriate labels to the clusters without direct human interaction.
-

2.4.2 Contextual image classification

- The method presented above works well in non-noisy data, and if the spectral properties determine classes sufficiently well.
- If noise or substantial variations in in-class pixel properties are present, the resulting image segmentation may have many small (often one-pixel) regions, which are misclassified.
- Several standard approaches can be applied to avoid this misclassification, which is very common in classification-based labeling.
- All of them use contextual information to some extent
- Post-processing filter to a labeled image
    - Small or single-pixel regions then disappear as the most probable label from the local neighborhood is assigned to them.
    - This approach works well if the small regions are caused by noise.
    - Unfortunately, the small regions can result from true regions with different properties in the original multispectral image, and in this case such filtering would worsen labeling results.
    - Post-processing filters are widely used in remote sensing applications
- Post-processing classification improvement
    - Pixel labels resulting from pixel classification in a given neighborhood form a new feature vector for each pixel, and a second-stage classifier based on the new feature vectors assigns final pixel labels.
    - The contextual information is incorporated in the labeling process of the second-stage classifier learning.

- Context may also be introduced in earlier stages, merging pixels into homogeneous regions and classifying these regions.

- Another contextual pre-processing approach is based on acquiring pixel feature descriptions from a pixel neighborhood.
  - Mean values, variances, texture description, etc. may be added to (or may replace) original spectral data.
  - This approach is very common in textured image recognition.

- The most interesting option is to combine spectral and spatial information in the same stage of the classification process.
  - The label assigned to each image pixel depends not only on multispectral gray level properties of the particular pixel but also considers the context in the pixel neighborhood.

- The last approach is discussed in more detail.
- Contextual classification of image data is based on the Bayes minimum error classifier.
- For each pixel x_0, a vector consisting of (possibly multispectral) values $f(x_i)$ of pixels in a specified neighborhood $N(x_0)$ is used as a feature representation of the pixel x_0. Each pixel is represented by the vector

$$\boldsymbol{\xi} = (f(\mathbf{x}_0), f(\mathbf{x}_1), \ldots, f(\mathbf{x}_k))$$

$$\mathbf{x}_i \in N(\mathbf{x}_0); \qquad i = 0, \ldots, k$$

- Some more vectors are defined which will be used later.
- Let labels (classification) of pixels in the neighborhood $N(x_0)$ be represented by a vector

$$\eta = (\theta_0, \theta_1, \ldots, \theta_k)$$

$$\theta_i \in \{\omega_1, \omega_2, \ldots, \omega_R\}$$

- and omega$_s$ denotes the assigned class.
- Further, let the labels in the neighborhood excluding the pixel x_0 be represented by a vector

$$\tilde{\eta} = (\theta_1, \theta_2, \ldots, \theta_k)$$

- Theoretically, there may be no limitation on the neighborhood size, but the majority of contextual information is believed to be present in a small neighborhood of the pixel x_0.
- Therefore, a 3 x 3 neighborhood in 4-connectivity or in 8-connectivity is usually considered appropriate.
- Also, computational demands increase exponentially with growth of neighborhood size.

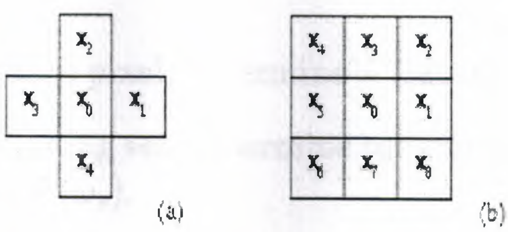


Figure 2.6 Pixel neighborhoods used in contextual image classification, pixel indexing scheme: (a) 4-neighborhood, (b) 8- neighborhood.

- A conventional minimum error classification method assigns a pixel x_0 to a class omega$_r$ if the probability of x_0 being from the class omega$_r$ is the highest of all possible classification probabilities

$$\theta_0 = \omega_r \quad \text{if} \quad P(\omega_r|f(\mathbf{x}_0)) = \max_{s=1,\ldots,R} P(\omega_s|f(\mathbf{x}_0))$$

- A contextual classification scheme uses the feature vector xi instead of x_0, and the decision rule remains similar

$$\theta_0 = \omega_r \quad \text{if} \quad P(\omega_r|\boldsymbol{\xi}) = \max_{s=1,\ldots,R} P(\omega_s|\boldsymbol{\xi})$$

- The a posteriori probability P(omega$_s$|xi) can be computed using the Bayes formula

$$P(\omega_s|\boldsymbol{\xi}) = \frac{p(\boldsymbol{\xi}|\omega_s)P(\omega_s)}{p(\boldsymbol{\xi})}$$

- Note that each image pixel is classified using a corresponding vector xi from its neighborhood, and so there are as many vectors xi as there are pixels in the image.
- The basic contextual classification algorithm can be summarized as

1. For each image pixel, determine a feature vector $\boldsymbol{\xi}$ (equation (8.15)).
2. From the training set, determine parameters of probability distributions $p(\boldsymbol{\xi}|\omega_s)$ and $P(\omega_s)$.
3. Compute maximum a posteriori probabilities $P(\omega_r|\boldsymbol{\xi})$ and label (classify) all pixels in the image according to Equation (8.19). An image classification results.

- A substantial limitation in considering larger contextual neighborhoods is exponential growth of computational demands with increasing neighborhood size.
- A recursive contextual classification overcomes these difficulties.

- The main trick of this method is in propagating contextual information through the image although the computation is still kept in small neighborhoods.
- Spectral and neighborhood pixel labeling information are both used in classification.
- Context from a distant neighborhood can propagate to the labeling $theta_0$ of the pixel x_0

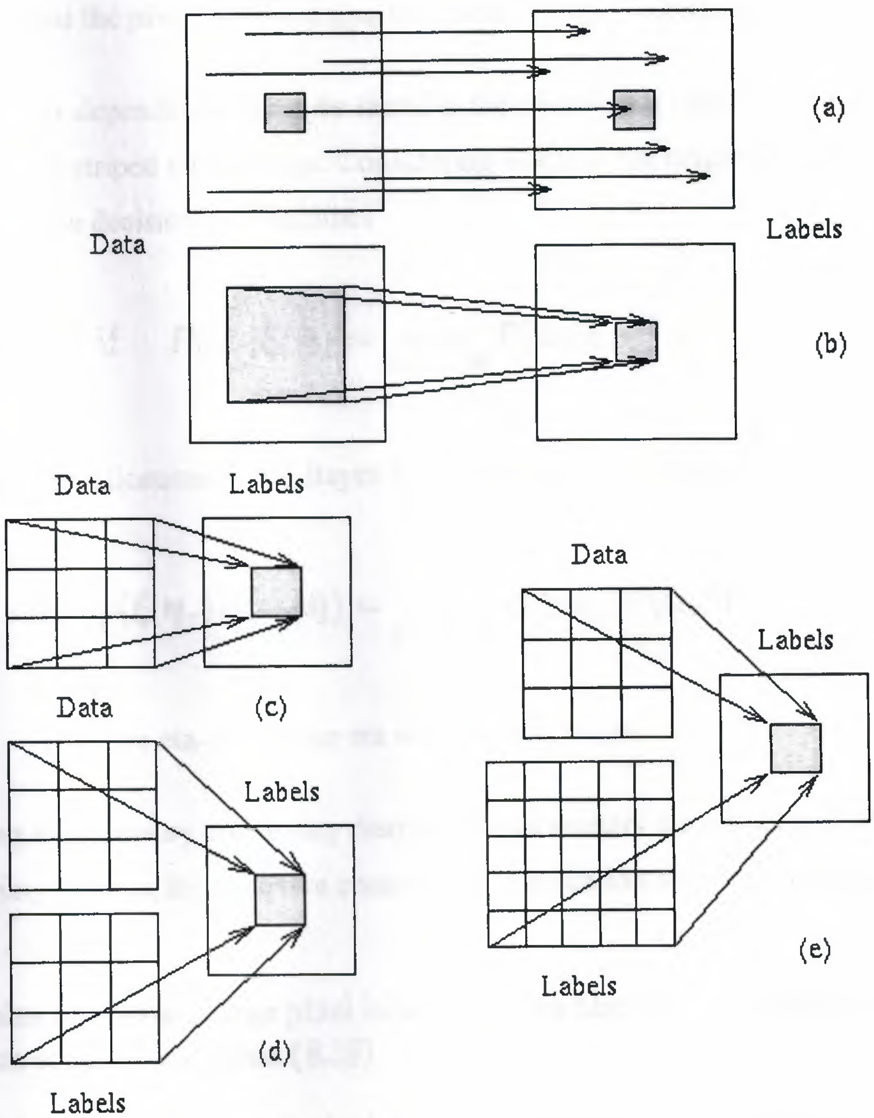


Figure 2.7 Principles of contextual classification: (a) Conventional non-contextual method (b) contextual method (c) recursive contextual method- step1 of previous algorithm (d) first application of step 2 (e) second application of step 2.

- The vector ~eta of labels in the neighborhood may further improve the contextual representation.
- Clearly, if the information contained in the spectral data in the neighborhood is unreliable (e.g. based on spectral data, the pixel x_0 may be classified into a number of classes with similar probabilities) the information about labels in the neighborhood may increase confidence in one of those classes.
- If a majority of surrounding pixels are labeled as members of a class $omega_i$, the confidence that the pixel x_0 should also be labeled $omega_i$ increases.
- More complex dependencies may be found in the training set - for instance imagine a thin striped noisy image. Considering labels in the neighborhood of the pixel x_0, the decision rule becomes

$$\theta_0 = \omega_r \quad \text{if} \quad P(\omega_r|\boldsymbol{\xi}, \tilde{\boldsymbol{\eta}}) = \max_{s=1,\ldots,R} P(\omega_s|\boldsymbol{\xi}, \tilde{\boldsymbol{\eta}})$$

- After several applications of the Bayes formula the decision rule transforms into

$$\theta_0 = \omega_r \quad \text{if} \quad p(\boldsymbol{\xi}|\boldsymbol{\eta}_r)P(\omega_r|\tilde{\boldsymbol{\eta}}) = \max_{s=1,\ldots,R} p(\boldsymbol{\xi}|\boldsymbol{\eta}_s)P(\omega_s|\tilde{\boldsymbol{\eta}})$$

where eta_r is a vector eta with $theta_0$=$omega_r$.

- Assuming all necessary probability distribution parameters were determined in the learning process, the recursive contextual classification algorithm follows:

1. Determine an initial image pixel labelling using the non-contextual classification scheme, equation (8.18)
2. Update labels in each image pixel $\mathbf{x}_0$, applying the current label vectors $\boldsymbol{\eta}$, $\tilde{\boldsymbol{\eta}}$, and local spectral vector $\boldsymbol{\xi}$ to the decision rule Equation (8.22).
3. Terminate the algorithm if the labels of all pixels in the image are stable, repeat step (2) otherwise.

- There is a crucial idea incorporated in the algorithm of recursive contextual image classification that will be seen several times throughout this chapter; this is the idea of **information propagation from distant image locations** without the necessity for expensive consideration of context in large neighborhoods.

2.5 Edge Detection

Edges are very important to any vision system (biological or machine).

- They are fairly cheap to compute.
- They do provide strong visual clues that can help the recognition process.
- Edges are affected by noise present in an image though.

An edge may be regarded as a boundary between two dissimilar regions in an image.

These may be different surfaces of the object, or perhaps a boundary between light and shadow falling on a single surface.

In principle an edge is easy to find since differences in pixel values between regions are relatively easy to calculate by considering gradients.

2.5.1 Representing Lines

The representation usually used for a line in two dimensions is of the form

$$y = mx + c$$

where m is the gradient of the line and c is the intercept of the line with the y axis (Fig 2.8).

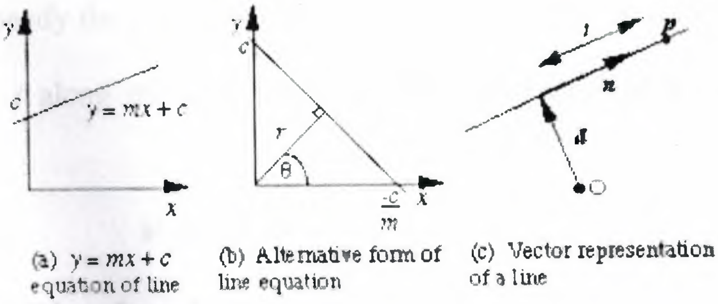


Figure 2.8 Line representation

An alternative representation of a line is

$$r = x\cos\theta + y\sin\theta$$

where r is the perpendicular distance from the line to the origin and is the angle the line makes with the x axis, as shown in Fig 20.

The latter form has the advantage that the gradient m, with a range $-\infty \leq m \leq +\infty$ has been replaced by the range of angles $0 \leq \theta \leq \pi$.

This is easier to deal with computationally.

(This will be important later -- see Hough Transforms).

Another alternative representation of an edge or line (again, see Fig 2.8) is by the vector pair $(\mathbf{n}, \mathbf{d})$, where $\mathbf{n}$ is a direction vector (usually normalised) along the edge and $\mathbf{d}$ is a vector from the origin to the closest point on the line.

Thus, the length of $\mathbf{d}$ is the perpendicular distance of the line from the origin.

This form of line representation is useful for both two- and three-dimensional lines, and indeed for three-dimensional lines this form is preferable.

Another advantage of this form of line representation is that the line can be parametrised.

Thus, we can specify the position of any point on the line, such as the end of an edge, by its distance t along the line. Therefore the coordinates of a point $\mathbf{p}(\mathbf{p}(x, y)$ or $\mathbf{p}(x, y, z))$ are

$$\mathbf{p} = \mathbf{u} + t\mathbf{n}.$$

2.5.2 Extracting Edges from Images

Many edge extraction techniques can be broken up into two distinct phases:

- Finding pixels in the image where edges are likely to occur by looking for discontinuities in gradients.

Candidate points for edges in the image are usually referred to as ***edge points***, ***edge pixels***, or ***edgels***.

- Linking these edge points in some way to produce descriptions of edges in terms of lines, curves *etc.*

2.5.3 Detecting Edge Points

2.5.3.1 Gradient based methods

An edge point can be regarded as a point in an image where a discontinuity (in gradient) occurs across some line. A discontinuity may be classified as one of three types (see Fig 2.9):

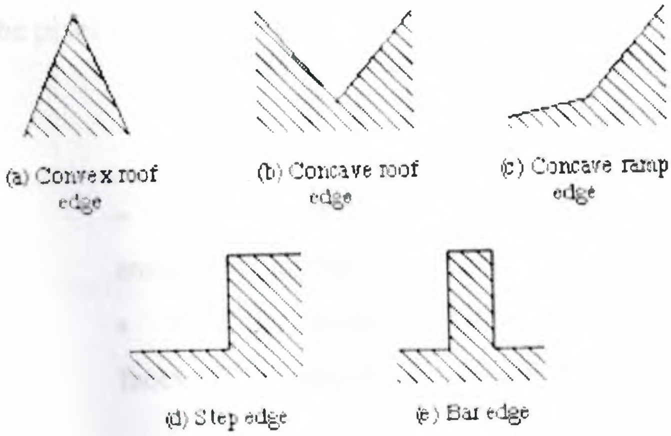


Figure 2.9 The C Compilation Model

A Gradient Discontinuity

-- where the gradient of the pixel values changes across a line. This type of discontinuity can be classed as

- *roof* edges
- *ramp* edges
- *convex* edges
- *concave* edges

by noting the sign of the component of the gradient perpendicular to the edge on either side of the edge.

Ramp edges have the same signs in the gradient components on either side of the discontinuity, while roof edges have opposite signs in the gradient components.

A Jump or Step Discontinuity

-- where pixel values themselves change suddenly across some line.

A Bar Discontinuity

-- where pixel values rapidly increase then decrease again (or ***vice versa***) across some line.

For example, if the pixel values are depth values,

- jump discontinuities occur where one object occludes another (or another part of itself).
- Gradient discontinuities usually occur between adjacent faces of the same object.

If the pixel values are intensities,

- a bar discontinuity would represent cases like a thin black line on a white piece of paper.
- Step edges may separate different objects, or may occur where a shadow falls across an object.

The gradient is a vector, whose components measure how rapidly pixel values are changing with distance in the x and y directions.

Thus, the components of the gradient may be found using the following approximation:

$$\frac{\partial f(x,y)}{\partial x} = \Delta_x = \frac{f(x+d_x,y)-f(x,y)}{d_x},$$
$$\frac{\partial f(x,y)}{\partial y} = \Delta_y = \frac{f(x,y+d_y)-f(x,y)}{d_y},$$

where d_x and d_y measure distance along the x and y directions respectively.

In (discrete) images we can consider d_x and d_y in terms of numbers of pixels between two points. Thus, when $d_x = d_y = 1$ (pixel spacing) and we are at the point whose pixel coordinates are (i,j) we have

$$\Delta_x = f(i+1,j)-f(i,j),$$
$$\Delta_y = f(i,j+1)-f(i,j).$$

In order to detect the presence of a gradient discontinuity we must calculate the ***change in gradient*** at (i,j). We can do this by finding the following ***gradient magnitude*** measure,

$$M = \sqrt{\Delta_x^2 + \Delta_y^2},$$

and the ***gradient direction***, , given by

$$\theta = \tan^{-1}\left[\frac{\Delta_y}{\Delta_x}\right].$$

2.5.3.2 Implementation:

The difference operators in Eqn. 44 correspond to convolving the image with the two masks in Fig. 2.10.

This is easy to compute:

- The top left-hand corner of the appropriate mask is superimposed over each pixel of the image in turn,
- A value is calculated for Δ_x or Δ_y by using the mask coefficients in a weighted sum of the value of pixel (i,j) and its neighbours.
- These masks are referred to as ***convolution masks*** or sometimes ***convolution kernels***.

-1	1
0	0

Δ_x

-1	0
1	0

Δ_y

Figure 2.10 Edge operator convolution masks

Instead of finding approximate gradient components along the x and y directions we can also approximate gradient components along directions at 45° and 135° to the axes

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

APPLICATIONS OF IMAGE PROCESSING

Graduation Project
EE- 400

Student: Barış Önal (980176)

Supervisor: Assoc. Prof. Dr Adnan Khashman

Lefkoşa-2003

TABLE OF CONTENTS

ACKNOWLEDGMENT i

ABSTRACT ii

INTRODUCTION 1

1. IMAGE PROCESSING BACKGROUND 2

1.1 Overview 2

1.2 Definition 2

1.2.1 Signals and Systems 2

1.2.2 Analog Signals 2

1.2.3 Digital Signals 3

1.2.3.1 Real and Complex Valued Signals 4

1.2.3.2 Discrete Time Cosine Signal 4

1.2.3.3 Sinusoids 5

1.2.3.4 Unit Sample 5

1.2.4 Image Processing 5

1.2.5 Digital Image Analysis 7

1.3 History 8

1.4 Future 8

1.5 Importance 8

1.6 Summary 8

2. IMAGE PROCESSING TECHNIQUES 9

2.1 Overview 10

2.2 Image Compression 10

2.2.1 Why do we need compression 10

2.2.2 What are the principles behind compression? 11

2.2.3 What are the different classes of compresion? 12

2.2.4 What does a typical image coder look like? 12

2.2.4.1 Source Encoder 13

2.2.4.2 Quantizer 13

2.2.4.3 Entropy Encoder 13

2.2.5 JPEG: DTC- Based Image Coding Standard 14

2.2.6 Wavelets and Image Compression 16

2.2.6.1 What is a Wavelet Transform? 16

2.2.6.2 Why Wavelet-based Compression? 17

2.2.6.3 Subband Coding 18

2.2.6.4 From Subband to Wavelet Codind 19

2.3 Image Restoration 20

2.4 Image Recognition 22

2.4.1 Pattern recognition methods in image understanding 22

2.4.2 Contextual image classification 22

2.5 Edge Detection 29

2.5.1 Representing Lines 29

2.5.2 Extracting Edges from Images 31

2.5.3 Detecting Edge Points 31

2.5.3.1 Gradient based methods 31

2.5.3.2 Implementation 34

2.5.3.3 Second Order Methods 36

2.6 Summary 42

3. APPLICATIONS OF IMAGE PROCESSING 43

3.1 Overview 43

3.2 Image Processing For medical Applicaitons 43

3.3 Image Processing For Space Exploration 44

3.4 Geographical Information System Applications 45

3.5 Weather Graphics Applications 46

3.6 Weather Data Applications 46

3.7 Climate Analysis Tools 46

3.8 Miliraty 47

3.9 Summary 47

4. APPLICATION OF IMAGE PROCESSING TO MINE WARFARE SONAR **48**

4.1 Overview 48

4.2 Introduction 48

4.3 Sector-scan Sonar 50

4.3.1 General Concepts 50

4.3.2 Enhancement 51

4.3.3 Segmentation 55

4.3.4 Computer Aided Detection 55

4.3.5 Computer Aided Classification 57

4.4 Side-Scan Sonar 57

4.4.1 General Concepts 58

4.4.2 Enhancement 58

4.4.3 Segmentation 59

4.4.4 Computer-Aided Detection 61

4.4.5 Computer-Aided Classification 62

4.5 The AMI Project 63

4.5.1 General Concepts 63

4.5.2 Enhancement 64

4.5.3 Segmentation 64

4.6 Summary 65

CONCLUSION 66

REFERENCES 67

ACKNOWLEDGMENTS

First I want to thank Assoc. Prof. Dr Adnan Khashman to be my supervisor and for his endless helps and courages to me. With his deep knowledge of Image Processing I manage to overcome the difficulties during my research.

I also want to thank to my friends at the faculty especially Kemal, Boran and Kamil for thier endless courages and helps. I want to thank to my friends in the company that I'm working. Especially Akar, Dinç,Melek and Erkan and finally to boss Mr. Mustafa and his wife Mrs.Göksel for their patience and helps to me.

Finally most impotant thing, my family. I would like to thank to my parents and my sisters for their endless helps and supports. Without them I wouldn't manage to do this project. Thank you. I wish you a healthy and good life now and ever. Dear my mother and father I proud of to being your son.

INTRODUCTION

Image Processing is becoming a widely acknowledged technology. Many everyday processes use automated vision systems, all of which rely upon image processing techniques. Image Processing, in general terms, refers to the manipulation, improvement and analysis of pictorial information. In this case, pictorial information means a two-dimensional visual image. Digital image processing is concerned with the improvement of quality of a picture that is digitally represented, as that represented in the digital computer.

From the 1960's through today, the evolution of the digital computer has certainly been largely responsible for enabling the proliferation of digital image processing applications. Costly mainframe computers are no longer a requirement of the digital image processing equation like they were in the 1960's. The advent of microprocessors, leading to the personal computer, has allowed stand-alone digital image processing applications to become viable.

In this project, applications of image processing is discussed. The aim of the project is to inform you about the real life applications of image processing and new techniques used in image processing.

The first chapter represents the background of image processing. This chapter includes history, basically description of image processing, importance and future of image processing.

Chapter two is assigned for the techniques that are used in image processing. These techniques can be sorted as;

Image compression, image restoration, image recognition and edge detection.

Chapter three represents the applications of image processing in real life.

Chapter four is assigned to applications of image processing to Mine Warfare Sonar Systems

CHAPTER ONE : IMAGE PROCESSING BACKGROUND

1.1 Overview

In this chapter the definition of image processing, history, future and importance in real life will be present.

1.2 Definition

1.2.1 Signals and Systems

Whether analog or digital, information is represented by the fundamental quantity in electrical engineering: the signal. Stated in mathematical terms, a signal is merely a function. Analog signals are continuous-valued; digital signals are discrete-valued. The independent variable of the signal could be time (speech, for example),space (images), or the integers.

1.2.2 Analog Sinals

Analog signals are usually signals defined over continuous independent variable(s). SPEECH is produced by your vocal cords exciting acoustic resonances in your vocal tract. The result is pressure waves propagating in the air, and the speech signal thus corresponds to a function having independent variables of space and time and a value corresponding to air pressure: sxt (Here we use vector notation x to denote spatial coordinates). When you record someone talking, you are evaluating the speech signal at a particular spatial location, x0 say. An example of the resulting waveform sx_0t is shown in Figure 1.1 [1].

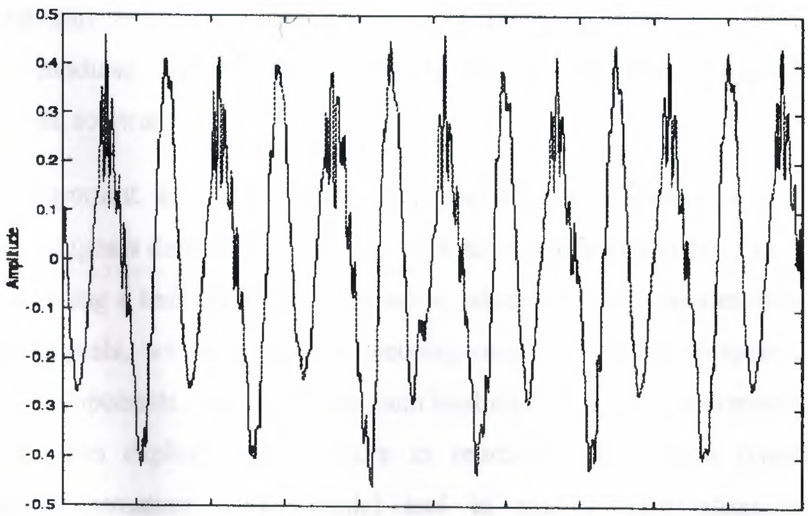


Figure 1.1 A speech signal's amplitude relates to tiny air pressure variations. Shown is a recording of the vowel "e" (as in "speech")

1.2.3 Digital Signals

The word "digital" means discrete-valued and implies the signal has an integer-valued independent variable. Digital information includes numbers and symbols (characters typed on the keyboard, for example). Computers rely on the digital representation of information to manipulate and transform information. Symbols do not have a numeric value, and each is represented by a unique number. The ASCII character code has the upper- and lowercase characters, the numbers, punctuation marks, and various other symbols represented by a seven-bit integer. For example, the ASCII code represents the letter a as the number 97 and the letter A as 65.

Signals can be represented by discrete quantities instead of as a function of a continuous variable. These discrete time signals do not necessarily have to take real number values. Many properties of continuous valued signals transfer almost directly to the discrete domain.

So far, we have treated what are known as **analog** signals and systems. Mathematically, analog signals are functions having continuous quantities as their independent variables, such as space and time. Discrete-time signals are functions defined on the integers; they are sequences. This result is important because discrete-time signals can be manipulated by systems instantiated as computer programs. Subsequent modules describe how virtually all analog signal processing can be performed with software.

As important as such results are, discrete-time signals are more general, encompassing signals derived from analog ones and signals that aren't. For example, the characters forming a text file form a sequence, which is also a discrete-time signal. As with analog signals, we seek ways of decomposing real-valued discrete-time signals into simpler components. With this approach leading to a better understanding of signal structure, we can exploit that structure to represent information (create ways of representing information with signals) and to extract information (retrieve the information thus represented). For symbolic-valued signals, the approach is different: We develop a common representation of all symbolic-valued signals so that we can embody the information they contain in a unified way. From an information representation perspective, the most important issue becomes, for both real-valued and symbolic-valued signals, efficiency.

1.2.3.1 Real and Complex Valued Signals

The discrete-time signals as stem plots to emphasize the fact they are functions defined only on the integers. We can delay a discrete-time signal by an integer just as with analog ones. A delayed unit sample has the expression δ□n-m, and equals one when n=m.

1.2.3.2 Discrete Time Cosine Signal

Figure 1.2 shows The Discrete Time Cosine Signal [1]

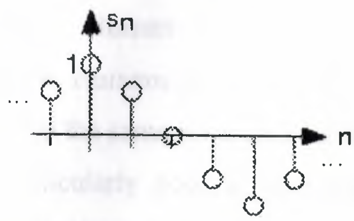


Figure 1.2 The Discrete Time Cosine Signal

1.2.3.3 Sinusoids

As opposed to analog complex exponentials and sinusoids that can have their frequencies be any real value, frequencies of their discrete-time counterparts yield unique waveforms only when f lies in the interval minus infinity to plus infinity.

1.2.3.4 Unit Sample

The second-most important discrete-time signal is the **unit sample**, which is defined in figure 1.3 [1]

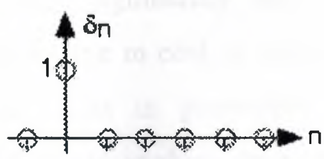


Figure 1.3 Unit Sample

1.2.4 Image Processing

Image processing and analysis is a combination of the visual enhancement of an image and the numerical evaluation of some aspect of the acquired image that would not be apparent in its analog form. The processing of a digital image refers to its actual refinement procedure. The analysis of a digital image encompasses the translation of the newly improved image's features into useful data. The necessity for this technology is due primarily to the difference between human visions and "computer vision." This contrast is most pronounced when comparing the type of information obtained from images and the methods used. Human vision is fundamentally qualitative and comparative, but not quantitative. Humans judge the relative size and shape of objects by mentally manipulating them to the same orientation and overlapping them to perform a comparison. Humans are particularly poor at assessing the color or brightness of features within images without direct comparison by positioning them adjacently. Progressive variations in brightness are typically dismissed as being representative of fluctuations in illumination, which the human visual system compensates automatically. In a gray scale image where the image is an array of intensities in a two-dimensional (2-D) space, the variation of intensity can provide you with a sense of texture, trend, edges, and anomalies on a surface. In a controlled environment, the variation of intensity is not merely changes in illumination, but an index of physical characteristics of a material.

Digital image analysis is a powerful method for gathering information. Relying on the convenience of computers, image processing and analysis methods have been used extensively in other disciplines for some time. However, their potential impact has started to be recognized in civil engineering only in the past few years. The advancement of technology and decline in cost of computers have provided numerous opportunities for significant advances in geotechnical and materials engineering. Researchers have applied digital image analysis in geotechnical engineering to study: cohesion less soil fabric, membrane penetration, mineral phase percentages in granular rocks, pavement cracking, particular shape, and morphological analysis of geotextiles. This field of study continues to grow in its form of applications as equipment becomes cheaper and better.

Computer based digital image analysis may involve aspects of mathematical morphology, stereology, and image processing. After thirty years of development, mathematical morphology has become one of the major tools in 2-D image processing. Mathematical morphology is the use of systematic numerical algorithms that extract qualitative and quantitative information from digital spatial data. Mathematical morphology also discards excess information in a controlled way. The removal of irrelevant detail makes images easier for analysis. Stereological methods are precise tools for obtaining quantitative information about 3-D microscopic structures, based mainly on observations made on sections of a specimen. It has been demonstrated that without stereology, material science cannot evolve into a truly quantitative science. The information extracted using these methods may lead to a better understanding of an observed structure and related phenomena.

1.2.5 Digital Image Analysis

After acquisition and storage, the digital image can be subjected to a number of processes that require handling of the image matrix. A digital image is a two dimensional (2-D) matrix (or array, see figure 1.4 [8]) where its elements are called pixels (picture elements). The pixel values are a light intensity function f(x, y) where x and y are denote spatial coordinates and the function "f" is a measure of brightness (or gray level) or color of the image at that point. In a gray scale image the value of 0 denotes black (or lowest intensity) and the value of 255 denotes white (or highest intensity). A pixel usually denotes a dot on a computer display or monitor depending on the screen resolution.

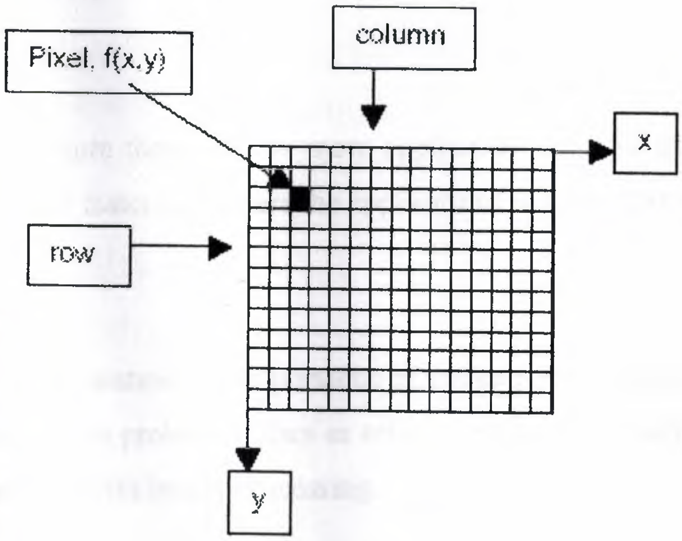


Figure 1.4 Matrix representation of an image

Image processing is a manipulation of matrices in the form of algorithms. Most processing functions can be implemented in a software application. The only reason to have specialized image-processing hardware is the need for speed in some applications. However, with high-speed desktop computers and storage devices becoming so accessible and affordable, specialized hardware is often not necessary. Today's image processing systems are a blend of off-the-shelf computers and specialized image processing accessories with overall operation being orchestrated by software running on the host computer.

1.3 History

In early 1960's pursuing lunar science program in NASA and a ranger program in NANA. In late 1960's image processing started to used by medical diagnostic imaging filed. Such as X-ray, computed tomography (CT), magnetic resonance imagery (MRI), positron emission tomography (PET) and ultrasound imaging.

In early 1970's Land sat (earth image) is used to analyse agricultural land-use and meteorological imagery.

In 1980's image processing is used for biological image, television broadcasting, military uses and to automate manufacturing processes.

1.4 Future

In think in near future there will be more application areas of image processing. With more powerful and faster computers the applications will be more wider.

1.5 Importance

Nowadays when understand the importance of image processing more. Such a way that we can send more probes to Mars or other planets, collect information and we can analyse the data with image processing.

1.6 Summary

In this chapter, definition of image processing, signals and systems, history, future and importance of image processing was discussed.

In the next chapter, image processing techniques will be discussed.

CHAPTER TWO: IMAGE PROCESSING TECHNIQUES

2.1 Overview

In this chapter we will discuss the image processing techniques. These techniques are;

Image Compression, image restoration, image enhancement, image recognition and edge detection.

2.2 Image Compression

Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications have not only sustained the need for more efficient ways to encode signals and images but have made compression of such signals central to storage and communication technology.

For still image compression, the `Joint Photographic Experts Group' or JPEG standard has been established by ISO (International Standards Organization) and IEC (International Electro-Technical Commission). The performance of these coders generally degrades at low bit-rates mainly because of the underlying block-based Discrete Cosine Transform (DCT) scheme. More recently, the wavelet transform has emerged as a cutting edge technology, within the field of image compression. Wavelet-based coding provides substantial improvements in picture quality at higher compression ratios. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression, as discussed later, have been developed and implemented. Because of the many advantages, the top contenders in the upcoming JPEG-2000 standard are all wavelet-based compression algorithms.

2.2.1 Why do we need compression?

The figures in Table 2.1 show the qualitative transition from simple text to full-motion video data and the disk space, transmission bandwidth, and transmission time needed to store and transmit such uncompressed data.

Table 2.1. Multimedia data types and uncompressed storage space, transmission bandwidth, and transmission time required. The prefix kilo- denotes a factor of 1000 rather than 1024.

Multimedia Data	**Size/Duration**	**Bits/Pixel or Bits/Sample**	**Uncompressed Size (B for bytes)**	**Transmission Bandwidth (b for bits)**	**Transmission Time (using a 28.8K Modem)**
A page of text	11" x 8.5"	Varying resolution	4-8 KB	32-64 Kb/page	1.1 - 2.2 sec
Telephone quality speech	10 sec	8 bps	80 KB	64 Kb/sec	22.2 sec
Grayscale Image	512 x 512	8 bpp	262 KB	2.1 Mb/image	1 min 13 sec
Color Image	512 x 512	24 bpp	786 KB	6.29 Mb/image	3 min 39 sec
Medical Image	2048 x 1680	12 bpp	5.16 MB	41.3 Mb/image	23 min 54 sec
SHD Image	2048 x 2048	24 bpp	12.58 MB	100 Mb/image	58 min 15 sec
Full-motion Video	640 x 480, 1 min (30 frames/sec)	24 bpp	1.66 GB	221 Mb/sec	5 days 8 hrs

The examples above clearly illustrate the need for sufficient storage space, large transmission bandwidth, and long transmission time for image, audio, and video data. At the present state of technology, the only solution is to compress multimedia data before its storage and transmission, and decompress it at the receiver for play back. For example, with a compression ratio of 32:1, the space, bandwidth, and transmission time requirements can be reduced by a factor of 32, with acceptable quality.

2.2.2 What are the principles behind compression?

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. **Redundancy reduction** aims at removing duplication from the signal source (image/video). **Irrelevancy reduction** omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS). In general, three types of redundancy can be identified:

- **Spatial Redundancy** or correlation between neighboring pixel values.
- **Spectral Redundancy** or correlation between different color planes or spectral bands.
- **Temporal Redundancy** or correlation between adjacent frames in a sequence of images (in video applications).

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Since we will focus only on still image compression, we will not worry about temporal redundancy.

2.2.3 What are the different classes of compression techniques?

Two ways of classifying compression techniques are mentioned here.

(a) Lossless vs. Lossy compression: In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Under normal viewing conditions, no visible loss is perceived (visually lossless).

(b) Predictive vs. Transform coding: In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding. Transform coding, on the other hand, first transforms the image from its spatial domain representation to a different type of representation using some well-known transform and then codes the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expense of greater computation.

2.2.4 What does a typical image coder look like?

A typical lossy image compression system is shown in Fig. 2.1. It consists of three closely connected components namely (a) Source Encoder (b) Quantizer, and (c) Entropy Encoder. Compression is accomplished by applying a linear transform to decorrelate the image data, quantizing the resulting transform coefficients, and entropy coding the quantized values.

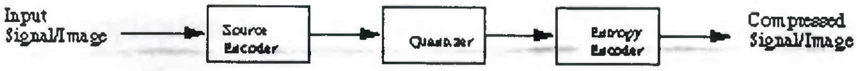


Figure 2.1 A Typical Lossy Signal/Image Encoder

2.2.4.1 Source Encoder (or Linear Transformer)

Over the years, a variety of linear transforms have been developed which include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and many more, each with its own advantages and disadvantages.

2.2.4.2 Quantizer

A quantizer simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it is a lossy process and is the main source of compression in an encoder. Quantization can be performed on each individual coefficient, which is known as Scalar Quantization (SQ). Quantization can also be performed on a group of coefficients together, and this is known as Vector Quantization (VQ). Both uniform and non-uniform quantizers can be used depending on the problem at hand. For an analysis on different quantization schemes.

2.2.4.3 Entropy Encoder

An entropy encoder further compresses the quantized values losslessly to give better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. The most commonly used entropy encoders are the Huffman encoder and the arithmetic encoder, although for applications requiring fast execution, simple run-length encoding (RLE) has proven very effective. It is important to note that a properly designed quantizer and entropy encoder are absolutely necessary along with optimum signal transformation to get the best possible compression.

2.2.5 JPEG : DCT-Based Image Coding Standard

The idea of compressing an image is not new. The discovery of DCT in 1974 is an important achievement for the research community working on image compression. The DCT can be regarded as a discrete-time version of the Fourier-Cosine series. It is a close relative of DFT, a technique for converting a signal into elementary frequency components. Thus DCT can be computed with a Fast Fourier Transform (FFT) like algorithm in *O(n log n)* operations. Unlike DFT, DCT is real-valued and provides a better approximation of a signal with fewer coefficients. The DCT of a discrete signal *x(n), n=0, 1, .. , N-1* is defined as:

$$X(u) = \sqrt{\frac{2}{N}}C(u)\sum_{n=0}^{N-1} x(n)\cos\left(\frac{(2n+1)u\pi}{2N}\right)$$

where, $C(u) = 0.707$ for $u = 0$ and $= 1$ otherwise.

In 1992, JPEG established the first international standard for still image compression where the encoders and decoders are DCT-based. The JPEG standard specifies three modes namely sequential, progressive, and hierarchical for lossy encoding, and one mode of lossless encoding. The `baseline JPEG coder' which is the sequential encoding in its simplest form, will be briefly discussed here. Fig. 2.2(a) and 2.2(b) show the key processing steps in such an encoder and decoder for grayscale images. Color image compression can be approximately regarded as compression of multiple grayscale images, which are either compressed entirely one at a time, or are compressed by alternately interleaving 8x8 sample blocks from each in turn.

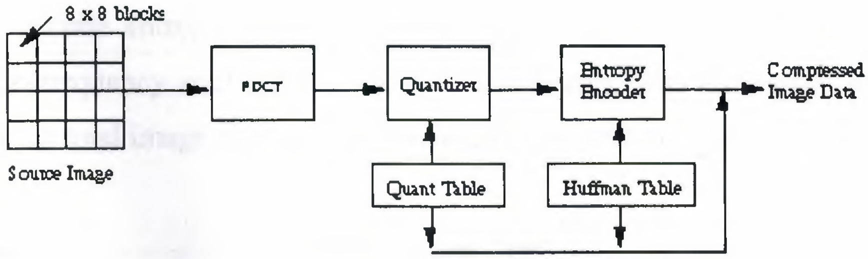


Figure 2.2(a) JPEG Encoder Block Diagram

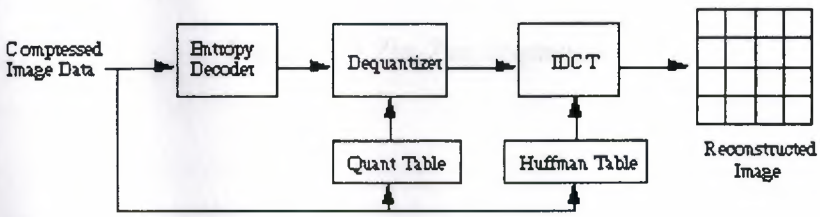


Figure 2.2(b) JPEG Decoder Block Diagram

The DCT-based encoder can be thought of as essentially compression of a stream of 8x8 blocks of image samples. Each 8x8 block makes its way through each processing step, and yields output in compressed form into the data stream. Because adjacent image pixels are highly correlated, the `forward' DCT (FDCT) processing step lays the foundation for achieving data compression by concentrating most of the signal in the lower spatial frequencies. For a typical 8x8 sample block from a typical source image, most of the spatial frequencies have zero or near-zero amplitude and need not be encoded. In principle, the DCT introduces no loss to the source image samples; it merely transforms them to a domain in which they can be more efficiently encoded.

After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a carefully designed 64-element Quantization Table (QT). At the decoder, the quantized values are multiplied by the corresponding QT elements to recover the original unquantized values. After quantization, all of the quantized coefficients are ordered into the "zig-zag" sequence as shown in Fig. 2.3. This ordering helps to facilitate entropy encoding by placing low-frequency non-zero coefficients before high-frequency coefficients. The DC coefficient, which contains a significant fraction of the total image energy, is differentially encoded.

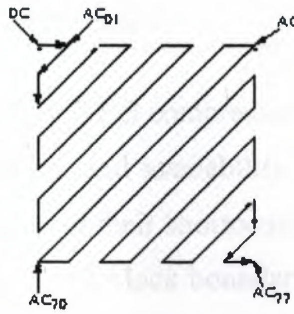


Figure 2.3 Zig-Zag sequence

Entropy Coding (EC) achieves additional compression losslessly by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. The JPEG proposal specifies both Huffman coding and arithmetic coding. The baseline sequential codec uses Huffman coding, but codecs with both methods are specified for all modes of operation. Arithmetic coding, though more complex, normally achieves 5-10% better compression than Huffman coding.

2.2.6 Wavelets and Image Compression

2.2.6.1 What is a Wavelet Transform ?

Wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). The

Discrete Wavelet Transform of a finite length signal *x(n)* having *N* components, for example, is expressed by an *N* x *N* matrix.

2.2.6.2 Why Wavelet-based Compression?

Despite all the advantages of JPEG compression schemes based on DCT namely simplicity, satisfactory performance, and availability of special purpose hardware for implementation, these are not without their shortcomings. Since the input image needs to be ``blocked," correlation across the block boundaries is not eliminated. This results in noticeable and annoying ``blocking artifacts" particularly at low bit rates as shown in Fig. 2.4. Lapped Orthogonal Transforms (LOT) attempt to solve this problem by using smoothly overlapping blocks. Although blocking effects are reduced in LOT compressed images, increased computational complexity of such algorithms do not justify wide replacement of DCT by LOT.

(a)

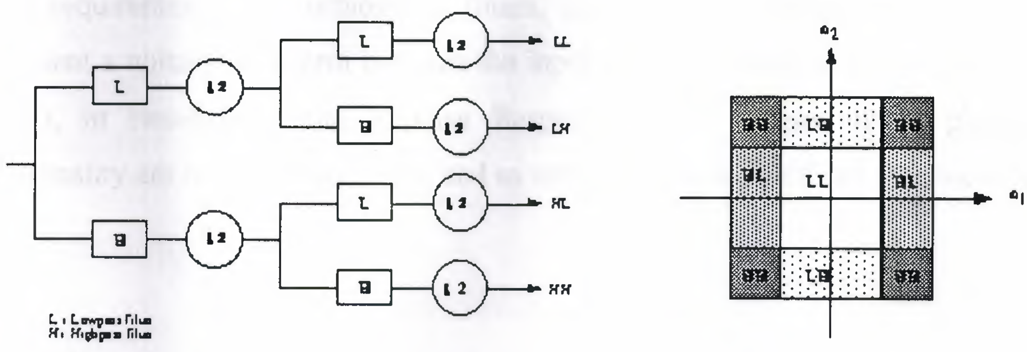
(b)

Figure 2.4(a) Original Lena Image, and **(b)** Reconstructed Lena with DC component only, to show blocking artifacts

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications wavelet-based schemes (also referred as subband coding) outperform other coding schemes like the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. In addition, they are better matched to the HVS characteristics. Because of their inherent multiresolution nature, wavelet coding schemes are especially suitable for applications where *scalability* and *tolerable degradation* are important.

2.2.6.3 Subband Coding

The fundamental concept behind Subband Coding (SBC) is to split up the frequency band of a signal (image in our case) and then to code each subband using a coder and bit rate accurately matched to the statistics of the band. SBC has been used extensively first in speech coding and later in image coding because of its inherent advantages namely variable bit assignment among the subbands as well as coding error confinement within the subbands.



(a) (b)

Figure 2.5(a) Separable 4-subband Filterbank, and 2.**5(b)** Partition of the Frequency Domain

Woods and O'Neil used a separable combination of one-dimensional Quadrature Mirror Filterbanks (QMF) to perform a 4-band decomposition by the row-column approach as shown in Fig. 2.5(a). Corresponding division of the frequency spectrum is shown in Fig. 2.5(b). The process can be iterated to obtain higher band decomposition filter trees. At the decoder, the subband signals are decoded, upsampled and passed through a bank of synthesis filters and properly summed up to yield the reconstructed image.

2.2.6.4 From Subband to Wavelet Coding

Over the years, there have been many efforts leading to improved and efficient design of filterbanks and subband coding techniques. Since 1990, methods very similar and closely related to subband coding have been proposed by various researchers under the name of *Wavelet Coding* (WC) using filters specifically designed for this purpose. Such filters must meet additional and often conflicting requirements. These include short impulse response of the analysis filters to preserve the localization of image features as well as to have fast computation, short impulse response of the synthesis filters to prevent spreading of artifacts (ringing around edges) resulting from quantization errors, and linear phase of both types of filters since nonlinear phase introduce unpleasant waveform distortions around edges. Orthogonality is another useful requirement since orthogonal filters, in addition to preservation of energy, implement a unitary transform between the input and the subbands. But, as in the case of 1-D, in two-band Finite Impulse Response (FIR) systems linear phase and orthogonality are mutually exclusive, and so orthogonality is sacrificed to achieve linear phase.

2.3 Image Restoration

Digitized images typically suffer from a range of imperfections including geometric distortion, nonuniform contrast, and noise. These all introduce errors into $\rho(\mathbf{r},t)$ unless steps are taken to restore the image to its ``ideal" state. Some geometric distortions are caused by defects in the microscope optics, but most are introduced in later stages of digitization. Video signals adhering to the RS-170 standard, for example, consist of rectangular pixels with a 4:3 aspect ratio. A circle imaged by a video camera appears uniaxially distorted into an ellipse when digitized and displayed by a computer, whose pixels are square. The analysis routines we describe below are most easily implemented for images consisting of square pixels. While many digitizing boards attempt to correct for uniaxial distortion, they often leave a residual anisotropy of a few percent. Both uniform and nonuniform geometric distortions can be measured by creating images of standard grids, identifying features in the images with features in the standards, and determining how far the image features are displaced from their ideal locations in an undistorted image. The algorithms we describe below for locating colloidal spheres also are useful for locating features in such calibration standards. Standard image processing texts describe algorithms for measuring apparent distortions in the calibration grid image and removing the distortion by spatial warping. Many image processing packages such as IDL include efficient implementations.

Contrast gradients can arise from nonuniform sensitivity among the camera's pixels. More significant variation often is due to uneven illumination. Long wavelength modulation of the background brightness complicates the design of criteria capable of locating spheres' images throughout an entire image. Subtracting off such a background is not difficult if the features of interest are relatively small and well separated as is frequently the case for colloidal images. Under these circumstances, the background is reasonably well modeled by a boxcar average over a region of extent $2w+1$, where w is an integer larger than a single sphere's apparent radius in pixels, but smaller than an intersphere separation:

$$A_w(x,y) = \frac{1}{(2w+1)^2} \sum_{i,j=-w}^{w} A(x+i, y+j). \qquad (2)$$

While long-wavelength contrast variations waste the digital imaging system's dynamic range, noise actually destroys information. Coherent noise from radio frequency interference (RFI) can be removed with Fourier transform techniques but is best avoided with proper electrical shielding. Digitization noise in the CCD camera and the frame grabber, however, is unavoidable. Such noise tends to be purely random with a correlation length $\lambda_n \approx 1$ pixel. Convolving an image $A(x,y)$ with a Gaussian surface of revolution of half width λ_n strongly suppresses such noise without unduly blurring the image:

$$A_{\lambda_n}(x,y) = \frac{1}{B} \sum_{i,j=-w}^{w} A(x+i, y+j) \exp\left(-\frac{i^2+j^2}{4\lambda_n^2}\right), \qquad (3)$$

with normalization $B = \left[\sum_{i=-w}^{w} \exp\left(-\frac{i^2}{4\lambda_n^2}\right)\right]^2$.

The difference between the noise-reduced and background images is an estimate of the ideal image. Since both eqn. (2) and eqn. (3) can be implemented as convolutions of the image $A(x,y)$ with simple kernels of support $2w + 1$, we can compute both in a single step with the convolution kernel

$$K(i,j) = \frac{1}{K_0}\left[\frac{1}{B}\exp\left(-\frac{i^2+j^2}{4\lambda_n^2}\right) - \frac{1}{(2w+1)^2}\right]. \qquad (4)$$

The normalization constant

$K_0 = \frac{1}{B}\left[\sum_{i=-w}^{w} \exp\left(-\frac{i^2}{2\lambda_n^2}\right)\right]^2 - \frac{B}{(2w+1)^2}$ facilitates comparison among images filtered with different values of w. The correlation length of the noise generally is not used as input parameter, with λ_n instead being set to unity. The efficacy of the filter can be judged from the example in Fig. 1(b).

In practice, the image $A(x,y)$ must be cast from an array of bytes to a higher precision data format, such as a floating point array, before convolution. This scaling, together with the actual convolution operation can be implemented in hardware with an array processor such as the Data Translation DT-2878. Further speed enhancement is realized by decomposing the circularly symmetric two-dimensional convolution kernel $K(i,j)$ into four one-dimensional convolution kernels, so that filtering can be computed in $O(w)$ operations rather than. $O(\omega^2)$.

2.4 Image Recognition

2.4.1 Pattern recognition methods in image understanding

- Pattern recognition methods frequently appear in image understanding[8] .
- Classification-based segmentation of multispectral images (satellite images, magnetic resonance medical images, etc.) is a typical example.
- Supervised methods are used for classification, a priori knowledge is applied to form a training set.
- In the image understanding stage, feature vectors derived from local multispectral image values of image pixels are presented to the classifier which assigns a label to each pixel of the image.
- Image understanding is then achieved by pixel labeling.
- Thus the understanding process segments a multispectral image into regions of known labels.
- Training set construction, and therefore human interaction, is necessary for supervised classification methods, but if unsupervised classification is used, training set construction is avoided.
- As a result, the clusters and the pixel labels do not have a one-to-one correspondence with the class meaning.

- This implies the image is segmented, but labels are not available to support image understanding.
- Fortunately, a priori information can often be used to assign appropriate labels to the clusters without direct human interaction.
-

2.4.2 Contextual image classification

- The method presented above works well in non-noisy data, and if the spectral properties determine classes sufficiently well.
- If noise or substantial variations in in-class pixel properties are present, the resulting image segmentation may have many small (often one-pixel) regions, which are misclassified.
- Several standard approaches can be applied to avoid this misclassification, which is very common in classification-based labeling.
- All of them use contextual information to some extent

- Post-processing filter to a labeled image
  - Small or single-pixel regions then disappear as the most probable label from the local neighborhood is assigned to them.
  - This approach works well if the small regions are caused by noise.
  - Unfortunately, the small regions can result from true regions with different properties in the original multispectral image, and in this case such filtering would worsen labeling results.
  - Post-processing filters are widely used in remote sensing applications

- Post-processing classification improvement
  - Pixel labels resulting from pixel classification in a given neighborhood form a new feature vector for each pixel, and a second-stage classifier based on the new feature vectors assigns final pixel labels.
  - The contextual information is incorporated in the labeling process of the second-stage classifier learning.

- Context may also be introduced in earlier stages, merging pixels into homogeneous regions and classifying these regions.

- Another contextual pre-processing approach is based on acquiring pixel feature descriptions from a pixel neighborhood.
    - Mean values, variances, texture description, etc. may be added to (or may replace) original spectral data.
    - This approach is very common in textured image recognition.

- The most interesting option is to combine spectral and spatial information in the same stage of the classification process.
    - The label assigned to each image pixel depends not only on multispectral gray level properties of the particular pixel but also considers the context in the pixel neighborhood.

- The last approach is discussed in more detail.
- Contextual classification of image data is based on the Bayes minimum error classifier.
- For each pixel x_0, a vector consisting of (possibly multispectral) values $f(x_i)$ of pixels in a specified neighborhood $N(x_0)$ is used as a feature representation of the pixel x_0. Each pixel is represented by the vector

$$\boldsymbol{\xi} = (f(\mathbf{x}_0), f(\mathbf{x}_1), \ldots, f(\mathbf{x}_k))$$

$$\mathbf{x}_i \in N(\mathbf{x}_0); \qquad i = 0, \ldots, k$$

- Some more vectors are defined which will be used later.
- Let labels (classification) of pixels in the neighborhood $N(x_0)$ be represented by a vector

$$\eta = (\theta_0, \theta_1, \ldots, \theta_k)$$

$$\theta_i \in \{\omega_1, \omega_2, \ldots, \omega_R\}$$

- and omega$_s$ denotes the assigned class.
- Further, let the labels in the neighborhood excluding the pixel x_0 be represented by a vector

$$\tilde{\eta} = (\theta_1, \theta_2, \ldots, \theta_k)$$

- Theoretically, there may be no limitation on the neighborhood size, but the majority of contextual information is believed to be present in a small neighborhood of the pixel x_0.
- Therefore, a 3 x 3 neighborhood in 4-connectivity or in 8-connectivity is usually considered appropriate.
- Also, computational demands increase exponentially with growth of neighborhood size.

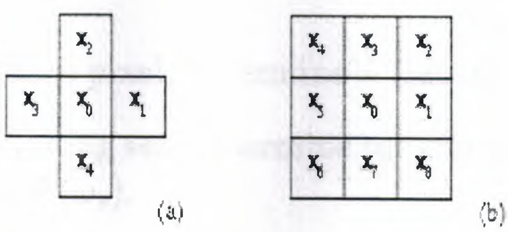


Figure 2.6 Pixel neighborhoods used in contextual image classification, pixel indexing scheme: (a) 4-neighborhood, (b) 8- neighborhood.

- A conventional minimum error classification method assigns a pixel x_0 to a class omega$_r$ if the probability of x_0 being from the class omega$_r$ is the highest of all possible classification probabilities

$$\theta_0 = \omega_r \quad \text{if} \quad P(\omega_r|f(\mathbf{x}_0)) = \max_{s=1,\dots,R} P(\omega_s|f(\mathbf{x}_0))$$

- A contextual classification scheme uses the feature vector xi instead of x_0, and the decision rule remains similar

$$\theta_0 = \omega_r \quad \text{if} \quad P(\omega_r|\boldsymbol{\xi}) = \max_{s=1,\dots,R} P(\omega_s|\boldsymbol{\xi})$$

- The a posteriori probability P(omega$_s$|xi) can be computed using the Bayes formula

$$P(\omega_s|\boldsymbol{\xi}) = \frac{p(\boldsymbol{\xi}|\omega_s)P(\omega_s)}{p(\boldsymbol{\xi})}$$

- Note that each image pixel is classified using a corresponding vector xi from its neighborhood, and so there are as many vectors xi as there are pixels in the image.
- The basic contextual classification algorithm can be summarized as

1. For each image pixel, determine a feature vector $\boldsymbol{\xi}$ (equation (8.15)).
2. From the training set, determine parameters of probability distributions $p(\boldsymbol{\xi}|\omega_s)$ and $P(\omega_s)$.
3. Compute maximum a posteriori probabilities $P(\omega_r|\boldsymbol{\xi})$ and label (classify) all pixels in the image according to Equation (8.19). An image classification results.

- A substantial limitation in considering larger contextual neighborhoods is exponential growth of computational demands with increasing neighborhood size.
- A recursive contextual classification overcomes these difficulties.

- The main trick of this method is in propagating contextual information through the image although the computation is still kept in small neighborhoods.
- Spectral and neighborhood pixel labeling information are both used in classification.
- Context from a distant neighborhood can propagate to the labeling $theta_0$ of the pixel x_0

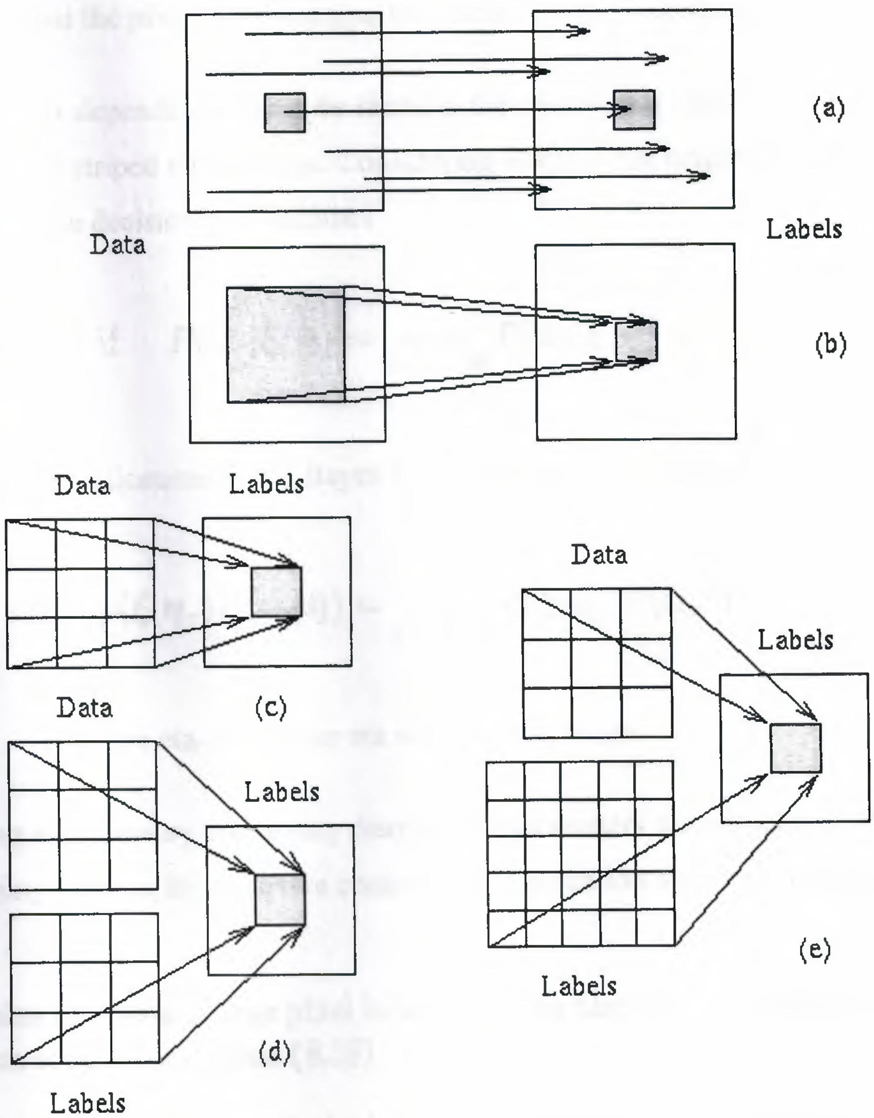


Figure 2.7 Principles of contextual classification: (a) Conventional non-contextual method (b) contextual method (c) recursive contextual method- step1 of previous algorithm (d) first application of step 2 (e) second application of step 2.

- The vector ~eta of labels in the neighborhood may further improve the contextual representation.
- Clearly, if the information contained in the spectral data in the neighborhood is unreliable (e.g. based on spectral data, the pixel x_0 may be classified into a number of classes with similar probabilities) the information about labels in the neighborhood may increase confidence in one of those classes.
- If a majority of surrounding pixels are labeled as members of a class $omega_i$, the confidence that the pixel x_0 should also be labeled $omega_i$ increases.
- More complex dependencies may be found in the training set - for instance imagine a thin striped noisy image. Considering labels in the neighborhood of the pixel x_0, the decision rule becomes

$$\theta_0 = \omega_r \quad \text{if} \quad P(\omega_r|\boldsymbol{\xi}, \tilde{\boldsymbol{\eta}}) = \max_{s=1,\ldots,R} P(\omega_s|\boldsymbol{\xi}, \tilde{\boldsymbol{\eta}})$$

- After several applications of the Bayes formula the decision rule transforms into

$$\theta_0 = \omega_r \quad \text{if} \quad p(\boldsymbol{\xi}|\boldsymbol{\eta}_r)P(\omega_r|\tilde{\boldsymbol{\eta}}) = \max_{s=1,\ldots,R} p(\boldsymbol{\xi}|\boldsymbol{\eta}_s)P(\omega_s|\tilde{\boldsymbol{\eta}})$$

where eta_r is a vector eta with $theta_0$=$omega_r$.

- Assuming all necessary probability distribution parameters were determined in the learning process, the recursive contextual classification algorithm follows:

1. Determine an initial image pixel labelling using the non-contextual classification scheme, equation (8.18)
2. Update labels in each image pixel $\mathbf{x}_0$, applying the current label vectors $\boldsymbol{\eta}$, $\tilde{\boldsymbol{\eta}}$, and local spectral vector $\boldsymbol{\xi}$ to the decision rule Equation (8.22).
3. Terminate the algorithm if the labels of all pixels in the image are stable, repeat step (2) otherwise.

- There is a crucial idea incorporated in the algorithm of recursive contextual image classification that will be seen several times throughout this chapter; this is the idea of **information propagation from distant image locations** without the necessity for expensive consideration of context in large neighborhoods.

2.5 Edge Detection

Edges are very important to any vision system (biological or machine).

- They are fairly cheap to compute.
- They do provide strong visual clues that can help the recognition process.
- Edges are affected by noise present in an image though.

An edge may be regarded as a boundary between two dissimilar regions in an image.

These may be different surfaces of the object, or perhaps a boundary between light and shadow falling on a single surface.

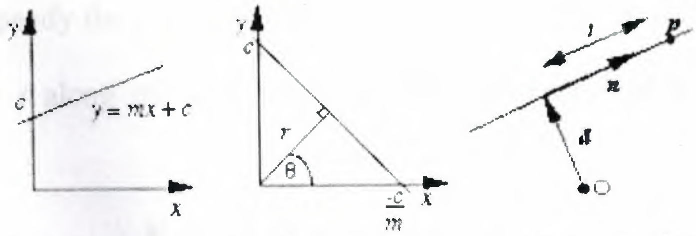
In principle an edge is easy to find since differences in pixel values between regions are relatively easy to calculate by considering gradients.

2.5.1 Representing Lines

The representation usually used for a line in two dimensions is of the form

$$y = mx + c$$

where m is the gradient of the line and c is the intercept of the line with the y axis (Fig 2.8).



(a) $y = mx + c$ equation of line (b) Alternative form of line equation (c) Vector representation of a line

Figure 2.8 Line representation

An alternative representation of a line is

$$r = x\cos\theta + y\sin\theta$$

where *r* is the perpendicular distance from the line to the origin and is the angle the line makes with the *x* axis, as shown in Fig 20.

The latter form has the advantage that the gradient *m*, with a range $-\infty \leq m \leq +\infty$ has been replaced by the range of angles $0 \leq \theta \leq \pi$.

This is easier to deal with computationally.

(This will be important later -- see Hough Transforms).

Another alternative representation of an edge or line (again, see Fig 2.8) is by the vector pair $(\mathbf{n}, \mathbf{d})$, where $\mathbf{n}$ is a direction vector (usually normalised) along the edge and $\mathbf{d}$ is a vector from the origin to the closest point on the line.

Thus, the length of **d** is the perpendicular distance of the line from the origin.

This form of line representation is useful for both two- and three-dimensional lines, and indeed for three-dimensional lines this form is preferable.

Another advantage of this form of line representation is that the line can be parametrised.

Thus, we can specify the position of any point on the line, such as the end of an edge, by its distance t along the line. Therefore the coordinates of a point $\mathbf{p}(\mathbf{p}(x, y)$ or $\mathbf{p}(x, y, z))$ are

$$\mathbf{p} = \mathbf{u} + t\mathbf{n}.$$

2.5.2 Extracting Edges from Images

Many edge extraction techniques can be broken up into two distinct phases:

- Finding pixels in the image where edges are likely to occur by looking for discontinuities in gradients.

Candidate points for edges in the image are usually referred to as ***edge points***, ***edge pixels***, or ***edgels***.

- Linking these edge points in some way to produce descriptions of edges in terms of lines, curves ***etc.***

2.5.3 Detecting Edge Points

2.5.3.1 Gradient based methods

An edge point can be regarded as a point in an image where a discontinuity (in gradient) occurs across some line. A discontinuity may be classified as one of three types (see Fig 2.9):

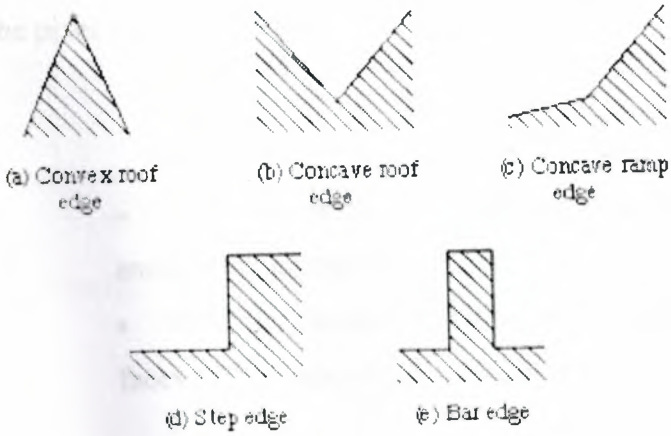


Figure 2.9 The C Compilation Model

A Gradient Discontinuity

-- where the gradient of the pixel values changes across a line. This type of discontinuity can be classed as

- *roof* edges
- *ramp* edges
- *convex* edges
- *concave* edges

by noting the sign of the component of the gradient perpendicular to the edge on either side of the edge.

Ramp edges have the same signs in the gradient components on either side of the discontinuity, while roof edges have opposite signs in the gradient components.

A Jump or Step Discontinuity

-- where pixel values themselves change suddenly across some line.

A Bar Discontinuity

-- where pixel values rapidly increase then decrease again (or ***vice versa***) across some line.

For example, if the pixel values are depth values,

- jump discontinuities occur where one object occludes another (or another part of itself).
- Gradient discontinuities usually occur between adjacent faces of the same object.

If the pixel values are intensities,

- a bar discontinuity would represent cases like a thin black line on a white piece of paper.
- Step edges may separate different objects, or may occur where a shadow falls across an object.

The gradient is a vector, whose components measure how rapidly pixel values are changing with distance in the *x* and *y* directions.

Thus, the components of the gradient may be found using the following approximation:

$$\frac{\partial f(x,y)}{\partial x} = \Delta_x = \frac{f(x+d_x, y) - f(x,y)}{d_x},$$
$$\frac{\partial f(x,y)}{\partial y} = \Delta_y = \frac{f(x, y+d_y) - f(x,y)}{d_y},$$

where d_x and d_y measure distance along the *x* and *y* directions respectively.

In (discrete) images we can consider d_x and d_y in terms of numbers of pixels between two points. Thus, when $d_x = d_y = 1$ (pixel spacing) and we are at the point whose pixel coordinates are (i,j) we have

$$\Delta_x = f(i+1, j) - f(i,j),$$
$$\Delta_y = f(i, j+1) - f(i,j).$$

In order to detect the presence of a gradient discontinuity we must calculate the ***change in gradient*** at (i,j). We can do this by finding the following ***gradient magnitude*** measure,

$$M = \sqrt{\Delta_x^2 + \Delta_y^2},$$

and the ***gradient direction***, , given by

$$\theta = \tan^{-1}\left[\frac{\Delta_y}{\Delta_x}\right].$$

2.5.3.2 Implementation:

The difference operators in Eqn. 44 correspond to convolving the image with the two masks in Fig. 2.10.

This is easy to compute:

- The top left-hand corner of the appropriate mask is superimposed over each pixel of the image in turn,
- A value is calculated for Δ_x or Δ_y by using the mask coefficients in a weighted sum of the value of pixel (i,j) and its neighbours.
- These masks are referred to as ***convolution masks*** or sometimes ***convolution kernels***.

-1	1
0	0

Δ_x

-1	0
1	0

Δ_y

Figure 2.10 Edge operator convolution masks

Instead of finding approximate gradient components along the x and y directions we can also approximate gradient components along directions at 45° and 135° to the axes respectively. In this case the following equations are used:

$$\Delta_1 = f(i+1, j+1) - f(i, j),$$
$$\Delta_2 = f(i, j+1) - f(i+1, j).$$

This form of operator is known as the ***Roberts edge operator*** and was one of the first operators used to detect edges in images. The corresponding convolution masks are given by;

0	1
-1	0

Δ_1

1	0
0	-1

Δ_2

Figure 2.11. The C Compilation Model

Many edge detectors have been designed using convolution mask techniques, often using 3×3 mask sizes or even larger.

An advantage of using a larger mask size is that errors due to the effects of noise are reduced by local averaging within the neighbourhood of the mask.

An advantage of using a mask of odd size is that the operators are ***centred*** and can therefore provide an estimate that is biased towards a centre pixel (i,j).

One important edge operator of this type is the ***Sobel edge operator***. The Sobel edge operator masks are given in Fig 2.12.

-1	0	1
-2	0	2
-1	0	1

Δ_x

1	2	1
0	0	0
-1	-2	-1

Δ_y

Figure 2.12 Sobel edge operator convolution masks

2.5.3.3 Second Order Methods

All of the previous edge detectors have approximated the first order derivatives of pixel values in an image.

It is also possible to use ***second order derivatives*** to detect edges.

A very popular second order operator is the ***Laplacian*** operator.

The Laplacian of a function $f(x,y)$, denoted by $\nabla^2 f(x,y)$, is defined by:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}.$$

Once more we can use discrete difference approximations to estimate the derivatives and represent the Laplacian operator with the 3×3 convolution mask shown in Fig 2.13.

0	1	0
1	-4	1
0	1	0

Figure 2.13 Laplacian operator convolution mask

However there are disadvantages to the use of second order derivatives.

- (We should note that first derivative operators exaggerate the effects of noise.) Second derivatives will exaggerated noise twice as much.
- No directional information about the edge is given.

The problems that the presence of noise causes when using edge detectors means we should try to reduce the noise in an image prior to or in conjunction with the edge detection process.

We have already discussed some methods of reducing or smoothing noise in the Image Processing Section.

Some of these methods may be of use here.

Another smoothing method is ***Gaussian smoothing***

- Gaussian smoothing is performed by convolving an image with a Gaussian operator which is defined below.
- By using Gaussian smoothing in conjunction with the Laplacian operator, or another Gaussian operator, it is possible to detect edges.

Lets look at the Gaussian smoothing process first.

The ***Gaussian distribution*** function in two variables, $g(x,y)$, is illustrated in Fig. 2.14 and is defined by

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

where σ is the standard deviation representing the width of the Gaussian distribution.

- The shape of the distribution and hence the amount of smoothing can be controlled by varying σ.
- In order to smooth an image $f(x,y)$, we convolve it with $g(x,y)$ to produce a smoothed image $s(x,y)$ ***i.e.*** $s(x,y) = f(x,y)*g(x,y)$.

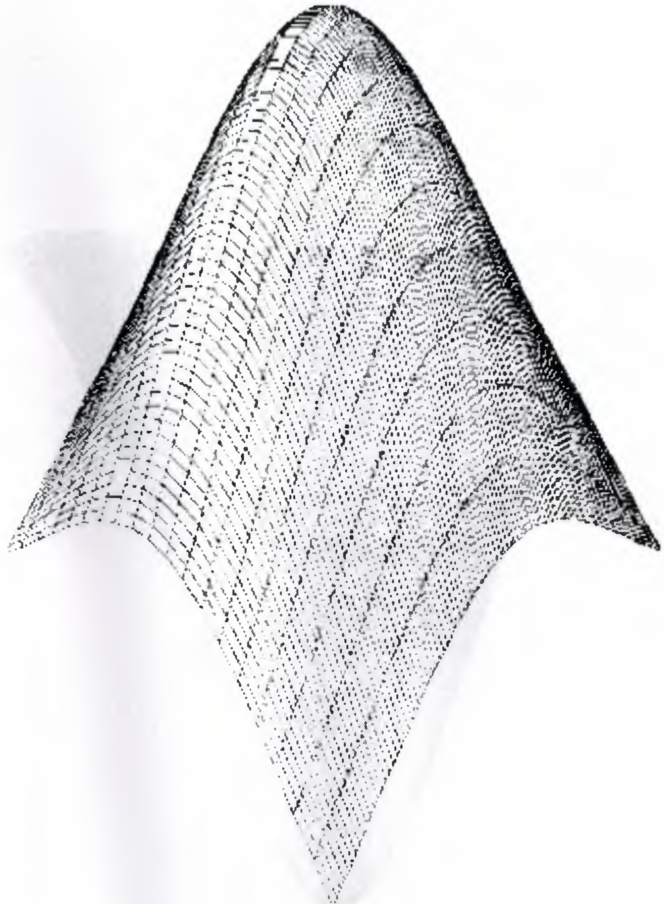


Figure 2.14 The Gaussian distribution in two variables

Having smoothed the image with a Gaussian operator we can now take the Laplacian of the smoothed image:

- Therefore the total operation of edge detection after smoothing on the original image is $\nabla^2(f(x,y) * g(x,y))$.
- It is simple to show that this operation can be reduced to convolving the original image *f(x,y)* with a ``Laplacian of a Gaussian'' (LOG) operator $\nabla^2 g(x,y)$, which is shown in Fig. 2.15.

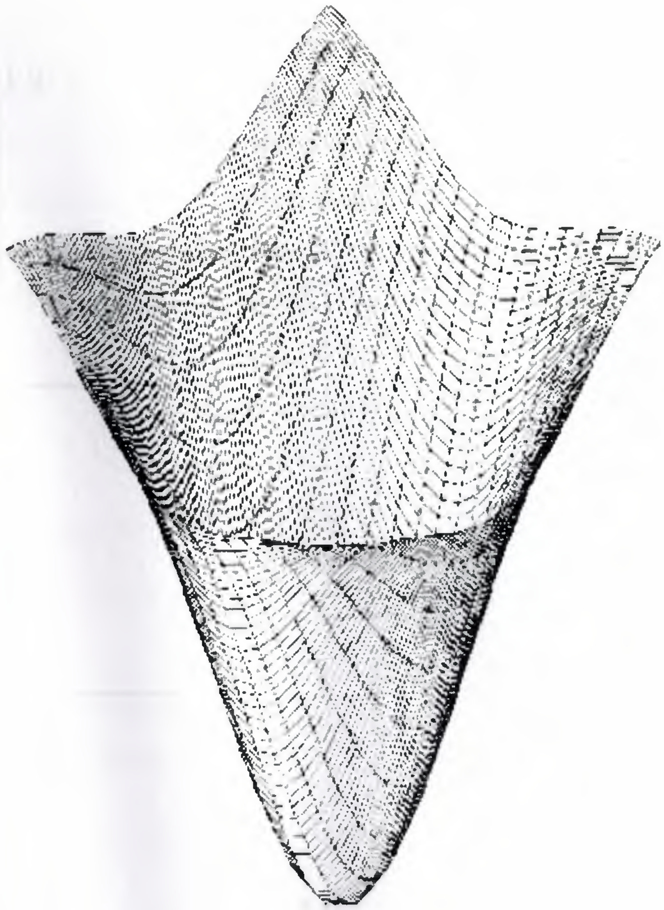


Figure 2.15 The LOG operator

Thus the edge pixels in an image are determined by a single convolution operation.

This method of edge detection was first proposed by Marr and Hildreth at MIT who introduced the principle of the ***zero-crossing*** method.

The basic principle of this method is to find the position in an image where the second derivatives become zero. These positions correspond to edge positions as shown in Fig. 2.16.

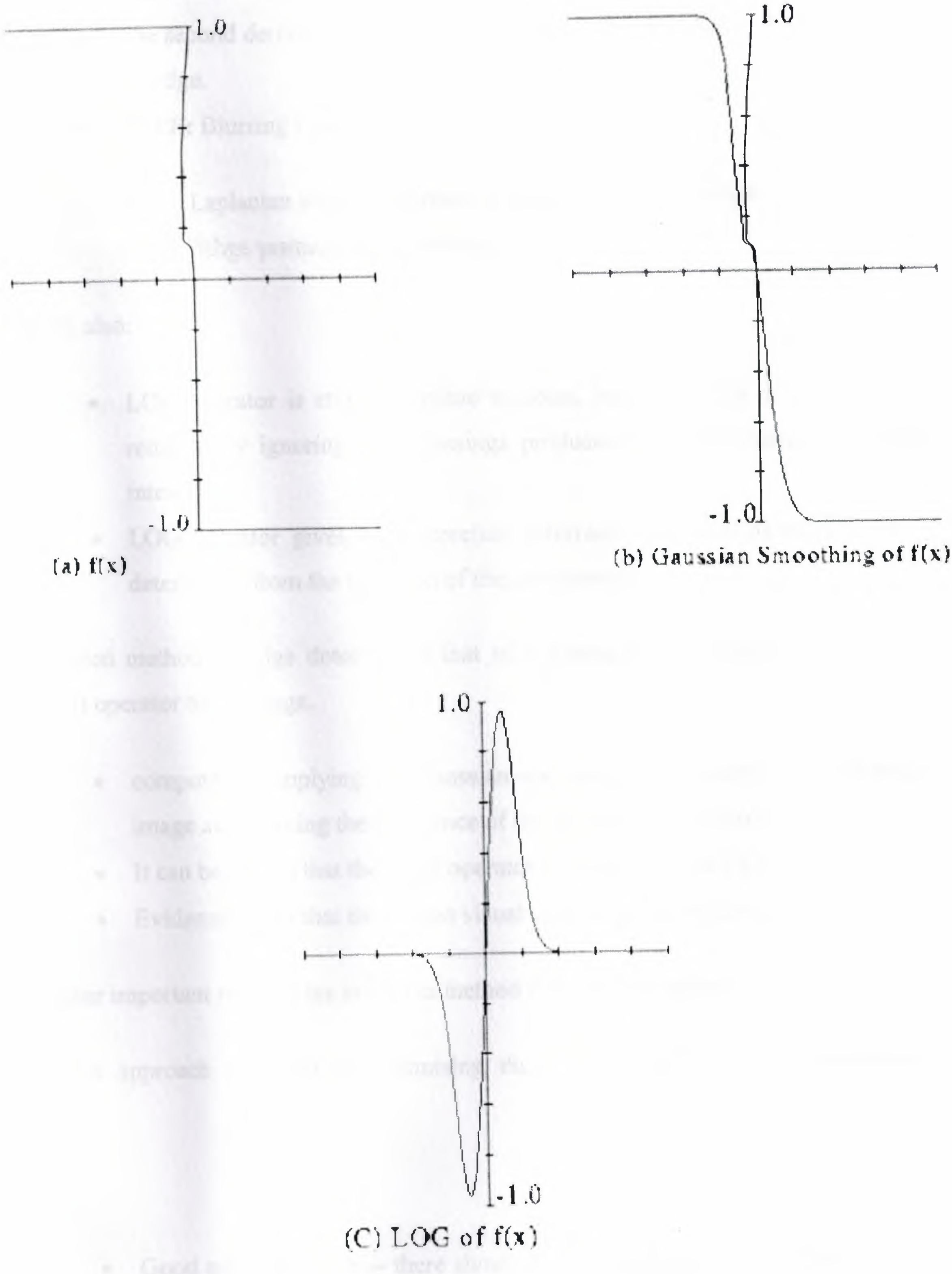


Figure 2.16 Steps of the LOG operator

- The Gaussian function firstly smooths or blurs any step edges.
- The second derivative of the blurred image is taken; it has a zero-crossing at the edge.
- **NOTE:** Blurring is advantageous here:

    1. Laplacian would be ***infinity*** at (unsmoothed) step edge.
    2. Edge position still preserved.

NOTE also:

- LOG operator is still susceptible to noise, but the effects of noise can be reduced by ignoring zero-crossings produced by small changes in image intensity.
- LOG operator gives edge direction information as well as edge points - determined from the direction of the zero-crossing.

A related method of edge detection is that of applying the **Difference of Gaussian** (DOG) operator to an image.

- computed by applying two Gaussian operators with different values of σ to an image and forming the difference of the resulting two smoothed images.
- It can be shown that the DOG operator approximates the LOG operator
- Evidence exists that the human visual system uses a similar method.

Another important recent edge detection method is the **Canny edge detector**.

Canny's approach is based on optimising the trade-off between two performance criteria:

- Good edge detection -- there should be low probabilities of failing to mark real edge points and marking false edge points.
- Good edge localisation -- the positions of edge points marked by the edge detector should be as close as possible to the real edge.

The optimisation can be formulated by maximising a function that is expressed in terms of

- The signal-to-noise ratio of the image,
- The localisation of the edges
- A probability that the edge detector only produces a single response to each actual edge in an image.

2.6 Summary

In this chapter, we discussed image processing techniques such as; Image compression, image restoration, image enhancement, image recognition and edge detection.

In the next chapter applications of image processing will be discussed.

CHAPTER THREE: APPLICATIONS OF IMAGE PROCESSING

3.1 Overview

In this chapter we will discuss real life applications of image processing techniques.

The applications can be sorted as;

- Medical Diagnostic Imaging
- Remote Sensing / Earth Resources
- Space Expolaration

3.2 Image Processing For Medical Applications

Image processing is very important in medical applications. In determining temperature field of various characteristics, one may choose between the following general methods, while their relative advantages and disadvantages must be decided in the light of the testsbeeing done:

- The selection of the temperature interval to be tested and, within that decision over the choice of the number and the widths of the isostrips.
- Determining the temperature at specified points of the surface under test
- A comparison between temperature distibutions along the horizontal and vertical lines
- Determining the temperature distribution and mean temperature in smaller specified areas of the syrface tested

- Pattern Classification
  - choosing therapy for peptic ulcers
  - diagnosing breast cancer
  - interpereting tissue sections and blood chemistry

- Image Analysis
  - interperetind radiograms
  - recognising macromolecules
  - monitoring diabetic retinopathies

3.3 Image Processing For Space Exploration

Image restoration techniques are normally used to increase the definiton of CCD image. Optical aberrations, seeing and tracking efficiency affect the images obtained with a CCD detector reducing its sharpness. The blurred image of a star, planet or galaxy can be significantly improves by deconvolving its Point Spread Function (PSF) in such a way that the end result is a sharper and more detailed image.

Several algorithms can be applied to the original image with impressive results. The best are the so-called inteactive techniques. The PSF of the image has to be determined before using any image restoration algorithm. This usually consists in isolatiıg a non-saturated star in the image to be treated and using this information as its PSF. The software works in an iterative way calculating several appriximations of the deconcolved image. Best examples of these algorithms are Maximum Entropy Deconvolution (MEM), Lucy-Richardson Deconvolution (LR) and Van-Cittert Deconvolution (VC). Direct algorithms can also be used with good results, such as the Weiner algorithm. There are however several drawbacks associated with the application of these algorithms (deconvolved images are usually noisy and they can not be used for photometry).

Image restoration techniques can improve the apparent sharpness of a CCD image by two to three times, meaning that medium size telescopes will perform like big telescopes.

We can use Image Processing for exploring the space as shown below.

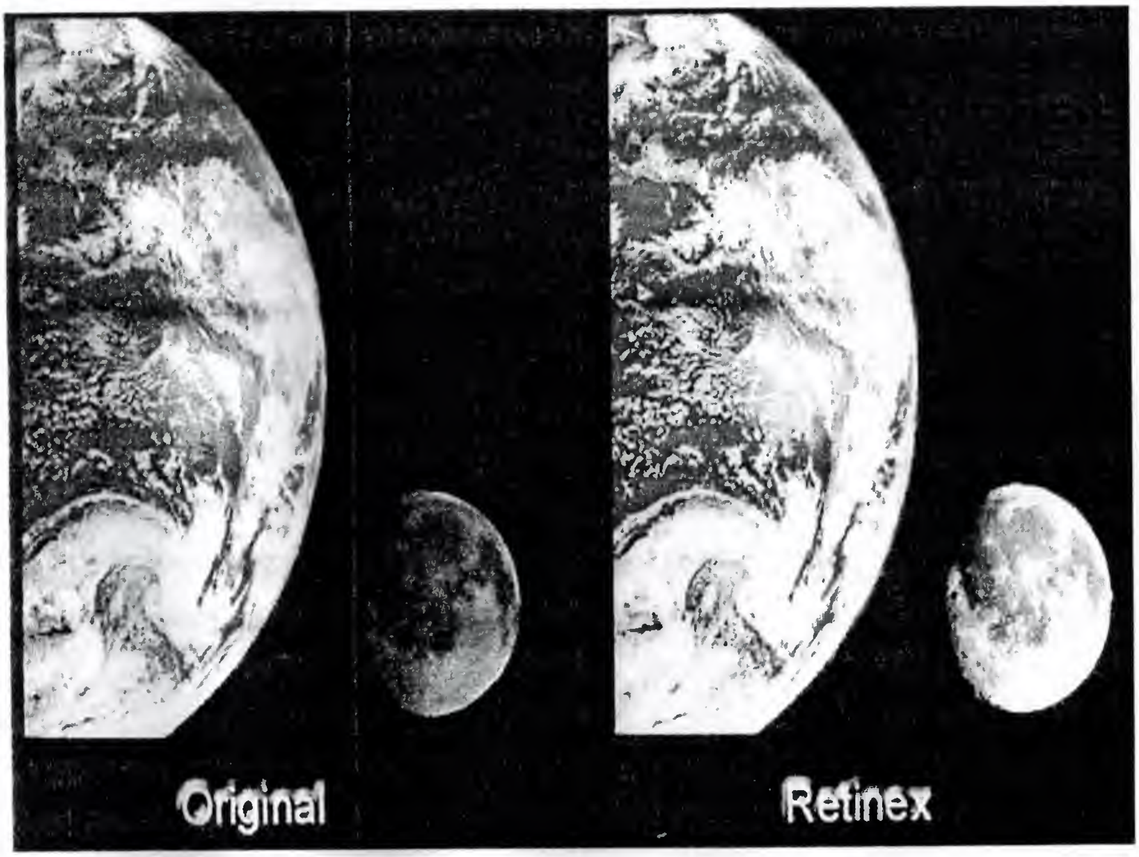


Figure 3.1 Space exploration using Image Processing

3.4 Geographical Information System Applications

- Major geopolotical boundaries: National, shoreline, major lakes and rivers, state, country
- Latitude and longitude grids
- User definable map bases
- Data combined into a single, common projection
- Map projections: Pseudo and true Mercantor, polar stereographic, Lambert Conformal
- Native sensor projectors: Geostationary, polar orbiting, radar.

3.5 Weather Graphics Applications

- Contours and plots of meteorological data
- Temporal and spatial meteorograms: Historical trace of surface and upper air parameters
- Thermodynamic diagrams: Skew T/log-P and Stuve
- Hodographs: Vertical wind soundings
- Cross-sections: Surface and full vertical isentropic
- Space/time graphs: Series of station model plots over time
- Weather watch and warning outlines

3.6 Weather Data Applications

- Observed parameters: Temperature, dew point, wind speed and direction, pressure, cloud information
- Derived Parameters: Equivalent potential temperature, mixing ratio, stability indices, potential temperature
- Future movement predictions

3.7 Climate Analysis Tools

- Dynamic analysis and additional processing of grindded data
- Time/ space analysis
- Relational or arithmetic operations
- Filtering with user-selected wave lengths

3.8 Military

- Mine Warfare Sonar detection systems
- Air detection systems
- Land mines detection systems

3.9 Summary

In this chapter we discussed the applications of image processing in some fields of life.

In the next chapter we will discuss the applications of image processing to Mine Warfare Sonar detction systems.

CHAPTER FOUR: APPLICATION OF IMAGE PROCESSING TO MINE WARFARE SONAR

4.1 Overview:

In this chapter we will discuss applications of image processing to Mine Warfare Sonar.

4.2 Introduction:

Sonar information collected while searching for, or identifying, underwater mines is often presented to the operator in the form of a two dimensional image. This is a result of the three dimensional nature of the search domain and the human use of vision as the primary source of sensory information. The heavy human reliance on visual information has made human beings highly skilled at the detection and classification of objects in images. Despite human expertise at comprehending visual information, sonar imagery still presents many challenges since it lies outside the normal scope of human visual experience. Signal or image processing can be applied to the sonar data to help human operators detect and classify mine-like objects in the operational environment. The processing of sonar data can be broken into two domains. The first domain is the use of signal processing (mostly one-dimensional) techniques to enhance the creation of sonar imagery. For example, the use of adaptive beam forming techniques to enhance the contrast of mine-like objects in sonar imagery lies in this first domain. The second domain is the use of image processing (two or higher dimensional) techniques on sonar imagery to aid or automate the detection and classification of mine-like objects. Both of these domains are vitally important to mine warfare sonar. This project examines image processing techniques and methodologies that have the potential to aid or automate the detection and classification of mine-like objects in sonar imagery. Such techniques may already have shown promise in the literature and are worth consideration and discussion, or may be potential new methods from other fields that might be applied to the field of mine warfare sonar. This project examines image-processing techniques tailored for three different types of sonar imagery. Sector-scan sonar images, side-scan sonar images and the three- dimensional images produced by the AMI project. The image processing techniques examined in this report can be grouped into four categories as follows:

- Enhancement techniques: Techniques that have the potential to enhance the contrast of mine-like objects in sonar images. Examples of this are the removal of noise and clutter, background normalisation, and the processing of sonar imagery to make best use of available knowledge of the human visual system.

- Segmentation techniques (low-level classification): Techniques that have the potential to classify individual pixels as belonging to background reverberation, clutter, highlights or shadows. This type of processing is usually not concerned with whether each pixel belongs to a mine-like object or not, but is often performed as a prelude to more advanced computer-aided detection and classification (CADCAC) techniques.

- Computer-aided detection (CAD): Techniques that may be useful to detect mine- like objects in sonar imagery. Confirmation of whether the object is actually a mine and its specific type are left to the human operator or subsequent processing methods.

- Computer-aided classification (CAC): Techniques which may be able to positively identify a mine-like object as a mine and determine the type and orientation of the mine involved.

The above grouping is only a rough guide to image processing techniques, as a great deal of overlap is often found, and some techniques defy being grouped in this way.

4.3 Sector-scan Sonar

This section will examine image processing techniques that may enhance the utility of sector scan sonar systems for mine warfare sonar. Sector-scan sonar imagery is produced by a sensor array that electronically scans a horizontally narrow beam to insonify an arc in a set direction. A two-dimensional image results which can be used to detect mine-like objects floating in the water column or resting on the seabed. During the formation of the image, any movement of the sensor array, or objects in the environment is assumed negligible. The images are formed fairly rapidly, generally within the order of a few seconds, and the human operators watch for objects in the images that persist for successive scans. Once an object is detected in a wide-angle view, the sonar settings may be changed to resolve a narrow field more highly. Under the right conditions, a mine-like object may be positively classified in this way.

4.3.1 General Concepts

An important general concept for processing sector-scan sonar imagery is the fact that temporal information is available. Human operators detect the presence of mine-like objects by watching for patterns in the image that persist for successive scans. In the same way, the image processing techniques to be applied to these images should make use of this temporal information whenever possible. The most successful techniques in the literature, as will be shown below, make use of multiple time frames to detect mine-like objects. Hence processing algorithms for sector-scan sonar should tend to be three-dimensional to make best use of the information available. A related concept is the matter of vessel movement. If information is known about the motion of the vessel, this should be incorporated into the CADCAC techniques to simplify and speed up processing. As will be shown below, use of vessel movement information can aid image enhancement and CADCAC techniques by removing the problem of tracking targets stationary relative to the seabed.

4.3.2 Enhancement

Enhancement techniques for sector-scan images divided into two categories. Techniques that do not make use of temporal information and techniques that do make use of temporal information. Usually sector scan images have a high degree of contrast, and hence most attempts at detecting and classifying mine-like objects in these images have used only simple enhancement techniques. For non-temporal techniques, median filtering is common. Median filtering was developed to handle image noise with so-called long tail statistics. This is due to the fact that the median filter is a maximum likelihood estimator of a signal in the presence of noise with a Laplacian distribution . Backscatter and sonar clutter are considered to fit in this category. For certain types of noise, median filtering can be a powerful noise removal technique. Figures 4.1 and 4.2 illustrate the point. Figure 4.1 shows a binary image corrupted by Salt and Pepper noise. In this case, 20% of pixels in the image have been set to 1 or 0 randomly. Figure 4.2 shows the result of applying a 3 by 3 median filter to the corrupted image. Note that the majority of noise pixels have been removed and the object in the image has become clearer.

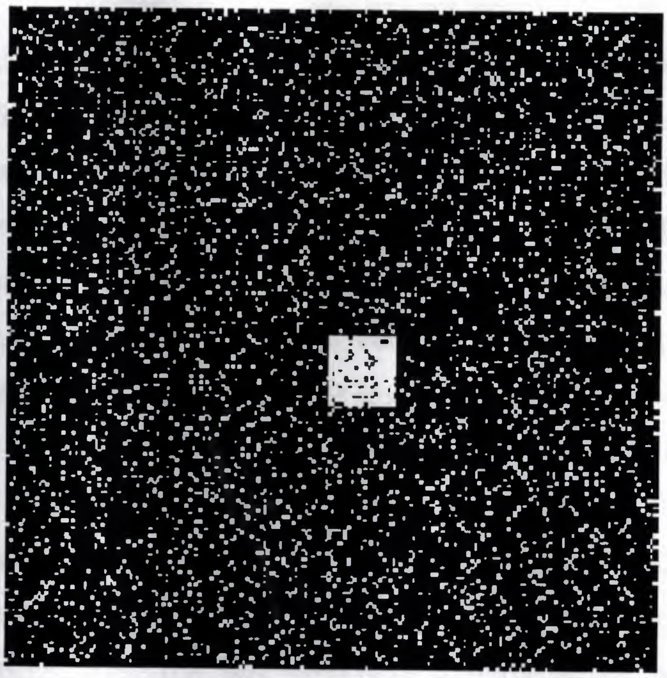


Figure 4.1 Image degraded by noise.

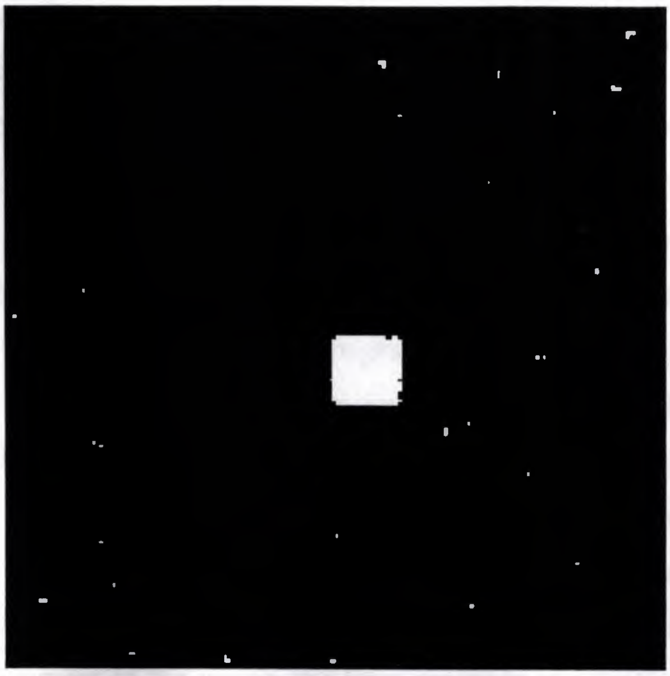


Figure 4.2 Image cleaned by median filter.

For more complicated images, or images degraded by noise with more complex distributions, plain median filters have a number of disadvantages. The most important disadvantage is the relatively high computational cost. Petillot *et al.* found that simple local averaging of sector-scan images could produce results similar to median filtering at a greatly reduced computational cost. This seems to be at odds with the assumption that noise in sonar imagery is non-Gaussian and can be assumed to work well only under certain conditions. Median filters also have the disadvantage of tending to blur edges in the image, although they are more capable of preserving edges as compared with linear filters. In the example of Figure 4.1, the median filter has degraded the corners of the object in the image. The symmetry and sharp edges of a mine are a major factor used to distinguish them from natural formations. The current use of median filters for sector-scan imagery appears to be at odds with this concept. Research has been done recently in the main stream image processing community detailing new forms of median filters (more generally, nonlinear filters) that have greater ability to remove noise while preserving edges without a great increase in computational load. These filters seem to have not been applied as yet to sonar imagery. Consideration should be made to examining the efficiency of median filters and whether advantages can be gained by using superior median filter variants or the wide variety of other non-linear edge preserving filters developed by the image processing community to handle noise distributions with long tail statistics.

The multi-resolution nature of the wavelet transform has many similarities to fractals and the way humans process images and is increasingly finding more applications in society. The wavelet transform divides the data into a number of different channels where each channel describes image information with a different spatial-frequency characteristic. Figures 4.3 and 4.4 show the way a correctly tailored wavelet transform can be used to extract an image neatly into a number of channels. Figure 4.3 shows a geometrical shape that contains lines of varying widths and directions. In the corners of the image, the lines are wide and diagonal. In the top and bottom middle of the image, the lines are predominantly horizontal, while in the left and right middle of the image, the lines are mostly vertical. In the centre of the image the lines are diagonal. Figure 4.4 shows a single level wavelet decomposition of Figure 4.3 using a bio orthogonal spline wavelet. Each quadrant of the image represents a different channel of the wavelet transform. Note how the various components of the image have been extracted to divide the vertical, horizontal and diagonal information present.

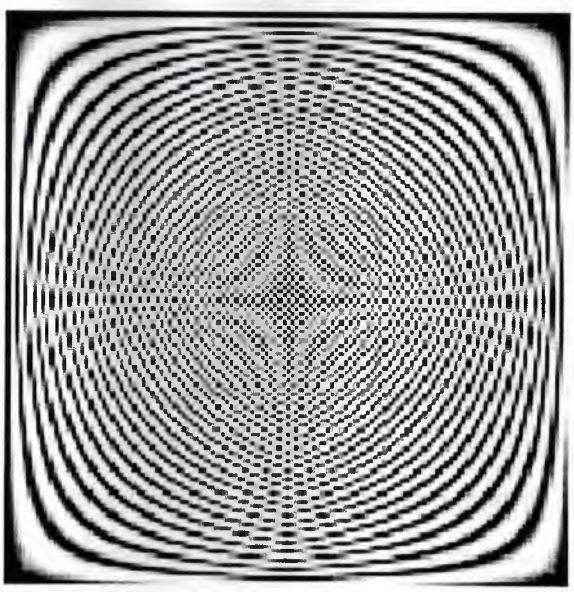


Figure 4.3 A geometrical form.

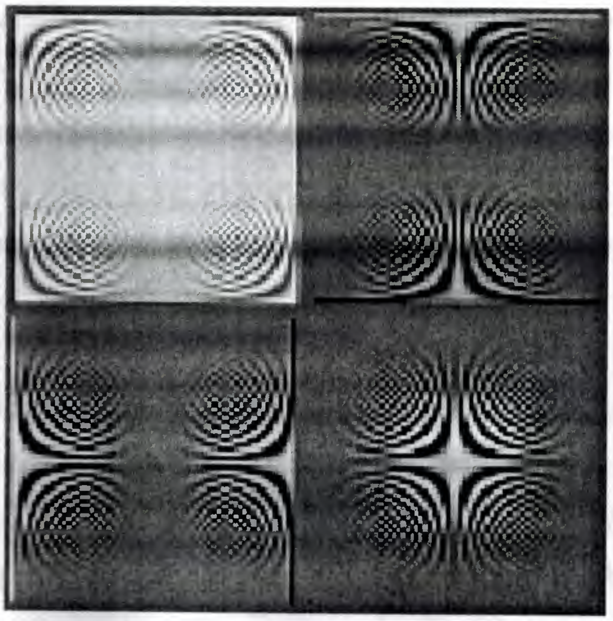


Figure 4.4 A Wavelet Transform of Figure 4.3

Tailoring the wavelet type and size to suit the dimensions of mine-like objects in the image might enable the mine-like object and clutter to appear in different sets of channels in the wavelet decomposition. This has the potential to enable effective removal of the clutter without degrading the image of the mine-like object. The size of the base wavelet used for the noise removal could also be scaled according to the current sonar range setting to provide consistent performance without operator input. Enhancement techniques that make use of temporal information have recently appeared in the sector-scan sonar literature. Most temporal enhancement techniques attempt to separate stationary objects from non-stationary objects and clutter. If the vessel motion is known, this motion can be corrected for within the image and this would enable mine-like objects be treated as stationary in the computational image domain. Temporally varying features, such as clutter, ambient biological noise, and fish could then be suppressed. Azimi-Sadjadi *et al.* compared a number of successive frames using a technique known as Recursive High Order Correlation., an extension of the standard concept of cross-correlation. Although computationally intensive, this method makes no assumptions about the objects being enhanced. Representing the sonar information in a three-dimensional format allows simple filtering operations to be performed in the temporal dimension of the data. The literature indicates that only 10 or so frames may be required to remove static objects from non- static ones. More complicated techniques involving linking spatial and temporal information are a worthwhile research direction, and the wavelet transform seems an excellent basis.

4.3.3 Segmentation

Sector-scan images generally do not show shadow effects to the same degree as side- scan imagery. For this reason, classifying pixels as either .highlight. or .background. usually performs basic segmentation. Chantler *et al.* chose an optimal threshold for each pixel using an iterative method. The resulting binary image can be operated on to close gaps or to remove incorrectly classified clutter. The high level of contrast of sector-scan imagery generally reduces the effectiveness of low-level segmentation techniques, rendering them for the most part unnecessary. However, some form of adaptive threshold where the threshold level is set according to the local statistical properties of the image, followed by clustering could be beneficial.

4.3.4 Computer-Aided Detection

Human operators use a number of visual cues to detect mine-like objects in sector-scan sonar imagery. Since the sonar images update every second or so, the operator looks for objects that remain present in many consecutive images and display a size and form consistent with mine-like objects. CAD techniques recently developed in the literature also make use of this temporal information. A very illustrative method is that of Chantler and Stoner. For objects discovered in the sonar image, a number of static features are computed that describe the shape and size properties of the object. Over consecutive scans, the feature measures for each object are computed. For any particular object, another set of temporal features is determined. These temporal features describe the changes in the static features over time. For example, a set of returns from a diver would be expected to display a lot of variation over time as the divers position shifts, hence the static features derived from the divers image would vary markedly from scan to scan. However a mine-like object would display little variation in its returns and hence the static features derived would remain relatively constant. This difference may not be very prominent using the static features from one scan, but by creating temporal features derived from the static features, the differences become easier to detect. Chantler and Stoner reported a marked detection improvement when temporal features were used.

In an operational environment, the mine-hunting vessel is generally in motion, which presents new opportunities or challenges. If the motion of the vessel is not known, the mine-like objects in the image must be matched with their occurrence in subsequent scans. This is a motion-tracking problem and has been looked at in the literature. Lane *et al.* and Chantler *et al.* reported that satisfactory results were achieved using concepts borrowed from the field of optical flow estimation. To estimate optical flow, a cost function is created which is optimised to estimate the motion of each pixel from one image to the next image. Certain assumptions are made regarding the brightness versus motion model used and the results typically contain some noise. By grouping and filtering the results, the motion of objects over many frames can be determined and those objects can be tracked. The objects that they sort to track displayed varying temporal returns, unlike mine-like objects, so the system they developed may be more complicated than is required for mine warfare sonar. If vessel motion is known to some degree the problem becomes much easier, since the computer knows the predicted location of the object in the next scan. Schweizer *et al.* used multiple detection algorithms to achieve an acceptable classification accuracy. Certain types of detectors may make better use of a priori information, or provide alternative feature discrimination. This increases the robustness of the detection results and improves accuracy.

Sector-scan CAD systems are still in their infancy and so a great deal of research can be done to improve them. More research needs to be done to determine the optimal temporal and static features to use for the detection of mine-like objects. Recently, certain powerful classification methods such as Residual Vector Quantisation have been applied to sector-scan mine warfare sonar with some success. It may be worthwhile to investigate whether features derived from the wavelet decomposition of a sonar image, specifically tailored for the dimensions of mine-like objects, can form good discriminators.

4.3.5 Computer-Aided Classification

Once the operator has detected the presence of a mine-like object, the system is switched to a high-resolution mode, imaging only the vicinity of the detection. The operator then looks for signs that the object is actually a mine. Symmetry, the presence of straight or curved edges, and regular formations may be used to determine the identity of the target. The target may not appear to have a recognisable form in any one scan, so the operator may be forced to observe the object for a period of time to identify it. Foresti et *al.* extracted information about edge orientations and their angular relations to other edges in the object and used this information to classify targets as man-made or natural. Foresti et *al.* developed a way to combine edge-detected information such that the effects of noise corrupting the edge orientation estimates can be greatly reduced and a stable estimate of the object shape can be obtained. In terms of mine warfare sonar, a similar concept could be used to determine the exact shape and orientation of the mine-like object and this boundary information could be encoded in a suitable way and compared to a database of encoded mine shapes. In this way, a classification of the mine type may be possible. There is considerable literature in the main stream image processing community describing robust, orientation and scale invariant methods for encoding boundary shapes, these techniques may be useful to help mine warfare sonar. Ongoing research in the image processing community into noise resistant edge detectors might also be useful to help the operator to classify the mine-like objects detected.

4.4 Side-Scan Sonar

This section will examine image processing techniques that may enhance the utility of side-scan sonar systems for mine warfare sonar applications. Side-scan sonar images are formed by a sensor array fixed on a moving platform. The sensor array forms a narrow image of a swath of the environment perpendicular to the motion of the imaging platform. As the platform moves, an image of the environment on either side of the platform is obtained. The image formation process ends when the imaging platform has left the zone of interest. Temporal information is used in the formation of a side-scan sonar image, but since only one image is obtained for each region within the *zone of interest, the use of temporal techniques such as those recommended for sector-scan imagery is not possible.*

4.4.1 General Concepts

Side-scan sonar images display many features similar to optical imagery from a purely image processing point of view. These images generally have a fairly even distribution of pixel values, and may often display a wide range of texture effects due to sea-floor characteristics. Mine-like objects in side-scan images can sometimes display low levels of contrast and often the acoustical shadow region appears larger and more prominent than the object itself. For this reason more research has been done to apply image processing techniques to side-scan images when compared to sector-scan images. The bulk of the research in the literature has concentrated on the presence of shadow effects produced by mine-like objects. The shadows produced by mine-like objects are often found to contain a great deal of information regarding the shape and size of the object, and it is generally considered crucial to use shadow information in any image processing-based CADCAC technique for side-scan imagery.

4.4.2 Enhancement

Most side-scan sonar image enhancement methods are designed to remove clutter and other forms of noise, without distorting or damaging the shape of the highlight and shadow regions associated with the mine-like object. Recently a promising technique has arisen which is based on concepts from the mature field of image restoration. In many works, clutter in side-scan images was suppressed using a concept known as Total Variation Minimisation. (TVM). This technique is based on altering the image to minimise a functional consisting of two terms. The first term in the functional assures image fidelity and the preservation of shadow and highlight information. The second term in the functional is designed to smooth the image and hence reduce noise. The optimisation of this functional forms a multi-dimensional minimisation problem almost identical to the image restoration problem. In all of these references, TVM was shown to be very efficient at suppressing clutter in the images while preserving edge information for use by later CADCAC techniques. Despite highly impressive results, the form of TVM techniques used so far has been fairly basic. In the field of image processing the TVM functional is related to the Constrained Least

Square Error. functional, and a great deal of research has been done into adaptive, intelligent algorithms and cost function variants to best solve this problem. The considerations involved are often identical; the removal of noise while preserving edge information. Many of these algorithms are designed to best preserve edge information in terms of human visual criteria. This is a vital consideration when human operators will examine the side-scan imagery. A worthwhile research direction would be to look at whether adaptive algorithms could be used to better enhance the TVM technique for side-scan images. Aridgides *et al.* used an adaptive linear finite impulse response (FIR) filter to suppress clutter. This is a well-known technique in both the signal and image processing communities but requires a model of the mine-like object to be enhanced. Such filters can adapt to the local clutter statistics in the image to achieve good results. Fernandez and Aridgides extended this concept to form an adaptive order-statistic filter, which like the median filter is tailored for noise with long tail statistics. Huynh *et al.* examined the wavelet transform as a way to remove noise from side-scan sonar imagery. It was shown that wavelet and wavelet packet de-noising techniques could improve the quality of side-scan images and the optimal wavelet type and size was investigated. They found the best performance was obtained when the wavelets were tailored to the size of expected mine-like objects. This success should encourage the consideration of other wavelet-based noise cleaning techniques from the optical imagery field as suitable methods for the enhancement of side-scan images.

4.4.3 Segmentation

For side-scan sonar images, segmentation is often used to separately classify pixels as belonging to highlights, background, or shadow regions before higher level CADCAC techniques are used to search for mine-like objects. After each pixel has been classified into one of the three choices, the pixels are often clustered together with their neighbours to remove incorrectly classified pixels. There exists a large variety of image processing techniques for segmentation and many of these have been applied to this problem.

Hoelscher-Hoebing and Kraus used Expectation Maximisation. This one of many iterative approaches where pixels are classified based on how well their local gray-level statistics match statistical models of the intended classes. Relating the image to a Markov random field model is then used to perform clustering. This method produced interesting results, yet has the disadvantage of requiring estimates of the probability density functions (PDFs) of the three classes involved. There are many related Bayesian techniques for image segmentation that may be used in similar ways. Guillaudeux et *al.* segmented side-scan images using a fuzzy version of k-means classification. The use of a fuzzy technique provides a consistent framework for measuring how well each pixel matches each class, which can be used in an iterative manner to improve results. The concept of fuzzy sets is intended to mimic human decision making processes and has been shown to be highly effective in some cases. Nagao filtering was then used to group pixels and remove deviations. The Nagao filter uses the fuzzy set membership information to cluster pixels, while still preserving edge information. This direction of research may prove very useful to the problems of mine warfare sonar, and provides a framework for understanding and improving upon the way human operators examine such imagery. Szymczak *et al.* examined the adaptation of the TVM approach from image enhancement to an image segmentation model for side-scan sonar images. This approach is called .Mumford-Shah. segmentation and is based heavily on the TVM functional (see Section 3.2). Szymczak *et al.* achieved satisfactory segmentation results using this methodology. This technique may be worth examining because it provides a way to link the enhancement and segmentation methodologies into a unified approach. In this way, research into techniques to improve the performance of TVM-related functionals in the image processing community may be applied to the problem of segmentation directly.

When clustering the segmented pixels, the clustering procedure could make use of boundary and edge orientation information to determine whether an uncertain pixel should be added to a shadow or not. For example, if a shadow has an otherwise straight boundary except for a few pixels of uncertain class, an intelligent algorithm may assign a greater than normal probability to these pixels belonging to the shadow. Such an algorithm would then favour the correct segmentation of man-made objects.

4.4.4 Computer-Aided Detection

A wide variety of techniques have been used in the literature to attempt to detect mine- like objects in side-scan sonar imagery. Dobeck *et al* used a two-dimensional non- linear matched filter. The matched filter was basically a model of a mine-like object and the results of the matched filtering were fed into a k-nearest neighbour-based neural network classifier and an optimal discriminatory classifier. The decisions of these two classifiers were then combined to produce a final decision. The use of a non- linear matched filter appears problematic due to the many possible orientations of mine-like objects and the various different conditions in the operational environment, however an important concept from that investigation was the use of multiple classifiers to produce a more robust decision.

Guo and Szymczak used the wavelet transform to decompose a side-scan image into a number of different channels. The image of an object in each channel then forms features for a neural network classifier. The neural network classifier uses a set of sub-networks, each examining a different wavelet channel. This forms an interesting multi-resolution neural network which detects mine-like objects based on features at various different resolutions. This concept is probably an important component of human visual detection and classification, and may be a useful research direction. Nelson and Tuovila used information about pixel groupings within clutter to create a clutter detector. For each object detected in the image, a set of features was sent to a fractal-based classifier. The classifier was designed to detect clutter and hence could be used to remove false positives from the candidate objects. Calder *et al.* used a Bayesian classifier to detect objects against textured background in side-scan sonar images. The technique requires models of the various textures present and the objects to be detected, and hence is of limited utility if such models do not exist. In conditions where accurate statistical models do exist, Bayesian classifiers perform well. As per the discussion in the previous section, mine-like objects will have a high probability of displaying symmetry or straight edges in their shadows. None of the above detectors of objects in side-scan images makes use of this fact. A simple method to incorporate this information would be to create a feature for each potential object that is weighted to indicate how many local edge orientations match others. In this way the detector could be weighted to detect objects with straight boundaries, and hence a high probability of being man-made. There are of course a variety of ways to include geometrical boundary properties as features and many such approaches can be found in the image processing literature.

4.4.5 Computer-Aided Classification

The difficult problem of classifying mine type and orientation has not been greatly researched. Mignotte *et al.* used a genetic optimisation technique to search through a template space. The technique described was not applied to mine classification in particular, but is instead a general method. In this template space, the shadow shapes from every possible mine type are stored. There are a number of permissible transformations that may occur to the basic template to reflect the orientation and range of the mine. These transformations are used as genes. in a genetic optimization technique. Such techniques attempt to simulate the principles of natural selection and evolve. the solution to a cost function. Such a technique may be a useful research direction, but genetic optimisation methods have often been criticised as being computationally expensive. Galerne *et al.* extracted and encoded shadow boundaries using Fourier descriptors. Fourier descriptors are a result of research in the image processing community into the encoding of boundary shapes in such a way that is invariant to scale, rotation and translation. Given a well-segmented mine-like object, the Fourier descriptors extracted could be compared against those present in a database to identify the mine type.

Galerne *et al* used this method to classify objects as man-made or natural only, but such a technique could be extended to classify mine types. Research continues in the image processing community into the best representation of boundary shapes for comparison. Recent research has found that using cubic b-spline curves or the wavelet transform to represent object contour shapes has many desirable properties and can outperform Fourier descriptors.

4.5. The AMI Project

This section will examine image processing techniques that may enhance the utility of the AMI project. The AMI (Acoustic Mine Imaging) project involves the use of a large acoustic array to create a high resolution image at close range of a suspected mine for the purposes of positive identification. The images obtained by this method are three dimensional in nature, and suffer from poor contrast and the presence of side-lobe distortion. Image processing concepts can be used to enhance and possibly segment these images. In this section, computer-aided detection and classification will not be discussed since the detection of a mine-like object is assumed to be already performed, and the classification problem is assumed to be handled by a human operator. The objective here will therefore be the representation of the object in the best form for the human operator. At the time this report was written, the author had not had the chance to carefully examine the nature of the images produced by the AMI project. Image processing techniques considered would greatly depend on the nature of the data; in particular, the effective dynamic range of pixel information will have important consequences on the range of techniques available. A low dynamic range restricts the options to binary morphology techniques, such as dilation and erosion.

4.5.1 General Concepts

Special compensation can be made for the differences in the data relationships along certain dimensions due to the position of the sensor array, however the fundamental three-dimensional nature of the data should be used whenever possible. With three-dimensional data, the volume of data to process rapidly increases with image size and computational time also increases time taken to perform an operation can easily get out of hand.

4.5.2 Enhancement

At the current time, enhancement and background correction are performed using a split-window normalise along each dimension separately. This is not a three- dimensional approach and could probably be improved. For noise removal, it may be worthwhile to examine whether three-dimensional median filter variants, or wavelet transform approaches can improve image quality. If sufficient dynamic range exists, some methods of adaptive histogram specification may be appropriate. Adaptive histogram specification involves analysing the histogram details in the neighbourhood of each pixel and computing a pixel value transformation for that region which best emphasises desired details. This may improve the contrast of the imaged object. The visual representation of the information in this type of image may also benefit from the use of colour to represent pixel distance from the observer in the third dimension. Another approach that may prove useful is to weight each pixel value dependent upon the statistics and density characteristics of its local region. Pixels in regions whose characteristics do not suggest the presence of an object can have their contrast reduced, while pixels in regions whose statistical/density characteristics do suggest the presence of an object can have their contrast enhanced. For images with low dynamic range, noise can sometimes be suppressed by adaptively thresholding the object followed by binary operations such as erosion followed by dilation.

4.5.3 Segmentation

A simple technique currently used is a neighbour association algorithm applied to the data. Each pixel has its local neighbourhood examined to determine whether other large valued pixels lie in the immediate vicinity. If none are present, the pixel being examined is assumed to be due to noise and removed. This seems a good approach to remain as part of a segmentation methodology. This technique has a result that is very similar to the Erode and Dilate noise removal technique described in the previous section. The problem of segmentation is in effect actually a problem of classification (although it may not always be directly treated as such) and hence a classification approach will probably produce the optimal results. For each pixel a set of features should be extracted detailing local statistics and density aspects in the region surrounding the current pixel. Research into the correct features to discriminate the object in these images is a logical first step to any segmentation method. These features can then be used to segment the image based on a set of heuristic rules or some other type of classifier. Using some heuristic rules to perform the segmentation is most probably preferable due to changing operational conditions and the speed at which processing needs to be done. Following segmentation, the pixels will need to be clustered to remove noise and incorrectly classified pixels. The method used will depend on the available processing power. Complicated methods of clustering based on various models of the data (such as Markov random field) can be used if the necessary

processing power is available. If processing power is at a premium, simple methods to close gaps and remove noise can be used.

4.6 Summary

In this section we discussed the Mine Warfare Sonar Detection System.

CONCLUSION

There are many application techniques and fields of image processing. There are many applications in different fields. Such as space exploration, photography, medical application field, military and defence, remote sensing, law forensics, factory automation systems, even in transportation.

This report has examined various image processing techniques which have the potential to aid the detection and classification of mine-like objects in sonar imagery. Three types of sonar imagery were looked at in particular:

- Sector-scan (forward looking) sonar.
- Side-scan sonar.
- The AMI project.

Within each of these sonar-imaging applications, each of the four components of any Computer-Aided Detection and Classification (CADCAC) system was examined. These components are:

- Enhancement.
- Segmentation.
- Computer-Aided Detection.
- Computer-Aided Classification.

For each of these components, image processing techniques with the potential to improve the performance of mine warfare sonar systems were discussed.

I think in future with the developments in technology application fields will be wider.

REFERENCES

1. Gonzalez, R. C. and Woods, R. E., Digital Image Processing, Addison-Wesley, 1992.

2. H. C. Andrews and B. R. Hunt, *Digital Image restoration*. Englewood Cliffs, NJ: Prentice-
Hall, 1977.

3. Photogrammetry II- DIGITAL PHOTOGRAMMETRY (1) M.ORHAN ALTAN Page1-1

4. http://www.students.ncl.ac.uk/k.a.o'keeffe/ADVICE.htm

5. http://www.cla.sc.edu/geog/rslab/

6. http://www.cla.sc.edu/geog/rslab/551/index.html

7. http://users.ece.gatech.edu/njayant/mmc11/sld003.htm

8. http://www.icaen.uiowa.edu/~dip/LECTURE/Understanding4.html

9. http://attrasoft.com/image.htm

10. http://burks.brighton.ac.uk/burks/foldoc/31/56.htm

11. Geman, S. & Geman, D. (1984) *IEEE Trans. Pattern Anal. Machine Intelligence* **6,** 721–741.

12. Wang, J. Y. A. & Adelson, E. H. (1994) *IEEE Trans. Image Proc.* **3,** 625–638.

13. Shi, J. & Malik, J. (1997) in *Proceedings of the IEEE Conference: Computer Vision and Pattern Recognition '97* (IEEE Press, Los Alamitos, CA), pp. 731–737.

14. Sirovich, L. & Kirby, M. (1987) *J. Opt. Soc. Am. A* **4,** 519–524.

15. Vetter, T. & Poggio, T. (1997) *IEEE Trans. Pattern Anal. Machine Intelligence* **19,** 733–742.

16. Burl, M., Weber, M. & Perona, P. (1998) in *Proceedings of the 5th European Conference on Computer Vision '98*, eds. Burkhardt, H.&Neumann, B. (Springer, New York), LNCS 1407, Vol. II, pp. 628–641.

17. Zhu, S. C. & Mumford, D. (1997) *IEEE Tr. Pattern Anal. Machine Intelligence* **19,** 1236–1250.

18. Vapnik, V. N. (1995) *The Nature of Statistical Learning Theory* (Springer, New York).

19. Hofmann, T., Puzicha, J. & Buhmann, J. (1998) *IEEE Trans. Pattern Anal. Machine Intelligence* **20,** 803–818.

20. S. Belongie, C. Carson, H. Greenspan, and J. Malik, *Color- and Texture-Based Image Segmentation Using EM and Its Application to Content-Based Image Retrieval*, Proceedings of the Sixth International Conference on Computer Vision, 1998.

21. J. Canny, *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (1986), no. 6.

22. I. J. Cox, M. Miller, S. Omohundro, and P. Yianilos, *The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments*, IEEE Transactions on Image Processing 9 (2000), no. 1.

23. E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, Academic Press, 1990.

24. O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1993.

25. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, *Query by Image and Video Content: The QBIC System*, IEEE Computer (1995).

26. R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Addison Wesley, Reading, 1992.

27. R. M. Haralick, K. Shanmugam, and I. Dinstein, *Textural Features for Image Classification*, IEEE Transactions On Systems, Man, and Cybernetics SMC-3 (1973), no. 6.