National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Progressive Image Transmission by Segmentation-Based Coding

by

Steven Caron, BASc.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

Ottawa-Carleton Institute of Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

ISBN 0-612-11542-9

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Acknowledgments

I would like to express my sincere appreciation to my thesis supervisor, Dr. Jean-François Rivest, for his support and guidance throughout this work. I am particularly thankful for the long hours we spent in our weekly discussions. I also wish to thank my co-supervisor Dr. S. Panchanathan. Special thanks to Dr. James Tam of the Communications Research Centre for his expert advice and his involvement in the design of the subjective experiment.

I would also like to thank all the support staff members at the Electrical Engineering department of the University of Ottawa, especially Ms. Lucette Lepage and Ms. Michèle Roy.

My special thanks to Grant Henderson and Jeff Brinskel, the administrators of the Multimedia Communications Laboratory, for their assistance and for letting me pick their brains. I would also like to thank Arif Obaid with whom I had many challenging discussions.

My special thanks to Renaud, Alex, James, Nicolas and all the other soccer players of the department for their support and friendship in the last two years.

ABSTRACT

Progressive image transmission, where an image builds up gradually, is gaining popularity in image database browsing applications. In these applications, a user might have to reject a large number of unwanted images before selecting the desired one. In such a case, the time required to identify the image contents becomes very important. In this thesis, we present a novel progressive image transmission technique based on a representation by segmentation. The technique preserves edges at low bit rates and is biased toward fast identification. An image is segmented into regions having constant intensity by applying a morphological operator: the watershed. The segmented image is gradually simplified using a graph. The simplifications are transmitted in the reverse order. At the decoder, the image is dynamically divided into an increasing number of regions as the transmission progresses. A subjective experiment was designed, and the recognition times of images transmitted with the proposed algorithm were compared with the recognition times of the same images transmitted with JPEG. The proposed method was found to result in faster recognition of image contents for almost all the images. Some work on the objective evaluation of coarse images is also presented.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Framework

The current level of processor performance, coupled with the recent progress in data storage technologies has contributed to the proliferation of digital images. The manipulation and the analysis of these images have opened new vistas in such fields as medicine, astronomy, remote sensing, geological prospection, and cinematographic production [Rab89]. In most cases, image compression and its standardization has played a key role in the achievement of cost-effective solutions.

On the other hand, the emergence of broadband communication technologies has been slow, and since these technologies are not yet widely available, there are still many occasions when the transmission of a digital image, even under compressed form, results in a significant delay. As a simple example, let us consider the transmission of a 512×512 pixel image at a resolution of 8 bits/pixel with a 14400 bps modem; even with a compression rate of 10:1 the transmission lasts over 14 seconds. Under these constraints, it becomes attractive to transmit first a coarse rendition and to improve it gradually. Such a process is termed *progressive image transmission* (PIT). PIT has

been the subject of extensive research [Tzo87] and is appropriate for applications involving image database browsing.

In instances where the bandwidth is even more limited, as it is often the case when a network is under congestion, even progressive image transmission may not be sufficient to eliminate all the delays. Browsing the World Wide Web (WWW) on the Internet is a common example. Because the application is interactive, a delay greater than a few seconds is often not acceptable and the best browsers use PIT to solve this problem. However, the techniques in use do not generate good results early enough in the transmission. As a result, PIT is used to entertain the end-user while the full image is being transmitted. An efficient PIT system should permit nearly instantaneous identification of visual information, while added details are made available for inspection after some delay. Let us now consider how this translates in the compression framework.

Two factors make image compression possible: redundancy and relevancy. Redundancy relates to the statistical properties of the image and can be either spatial or spectral. Spatial redundancy expresses the correlation between neighboring pixels while spectral redundancy expresses the correlation between spectral bands or color planes. Relevancy relates to the observer and depends on the limitation of the human visual system under the viewing conditions. An ideal compression mechanism removes redundancy and irrelevant information and efficiently encodes what remains. In practice, information that is neither redundant nor irrelevant is often discarded to obtain high compression results. Fig. 1.1 illustrates this general operation. We can distinguish the case of lossless compression, where all the information is considered relevant, as it is the case with medical images and images intended for machine vision processing, and the case of lossy compression where a quantizer is used to obtain higher compression factors.

Progressive image compression is generally achieved through reordering of the information; a coarse image can be formed as soon as enough information has been transmitted to cover the whole image plane. Hence an ideal PIT mechanism removes redundancy, orders the information according to its relevance, and efficiently encodes that information.

In this thesis, we attempted to reach this goal by gradually simplifying the image through the use of a representation by segmentation. Previous research has shown that such a technique performs well at very low bit rate [Kun 85].

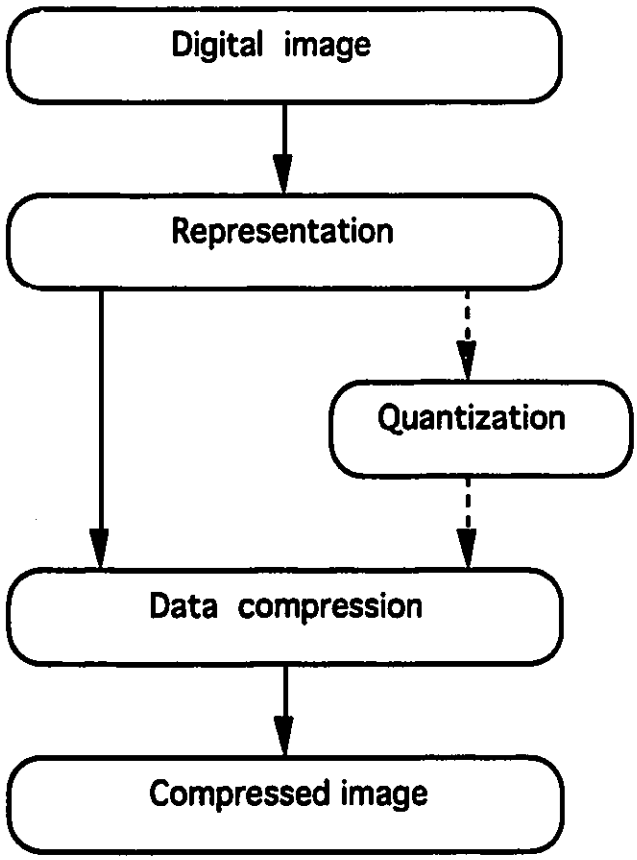


Fig. 1.1 Image compression framework.

In our implementation, the image is first decomposed into segments, or regions, where the pixel intensities are homogeneous. These regions can be described by their shape

and their content. The image is then gradually simplified by reducing the total number of regions needed to represent the complete image. This simplification produces a natural ordering that is used in the progressive transmission. The process is illustrated in Fig. 1.2. The contour information is transmitted first and used to predict the luminance content, which only accounts for a small part of the transmitted information.

1.2 Contributions

The main contributions of this thesis are :

- The introduction of a progressive image transmission scheme based on segmentation and biased towards fast identification.
- The design of a subjective test to rate progressive image transmission methods, its implementation, and its application on human subjects.
- The presentation of a novel objective measure of distortion for low bit rate images.
- The performance analysis of the new PIT scheme in comparison to the JPEG image standard using both subjective and objective measurement.

1.3 Thesis Organization

The thesis is organized as follows. The first chapter presents the framework which supports the research. The use of segmentation to represent the visual content of a

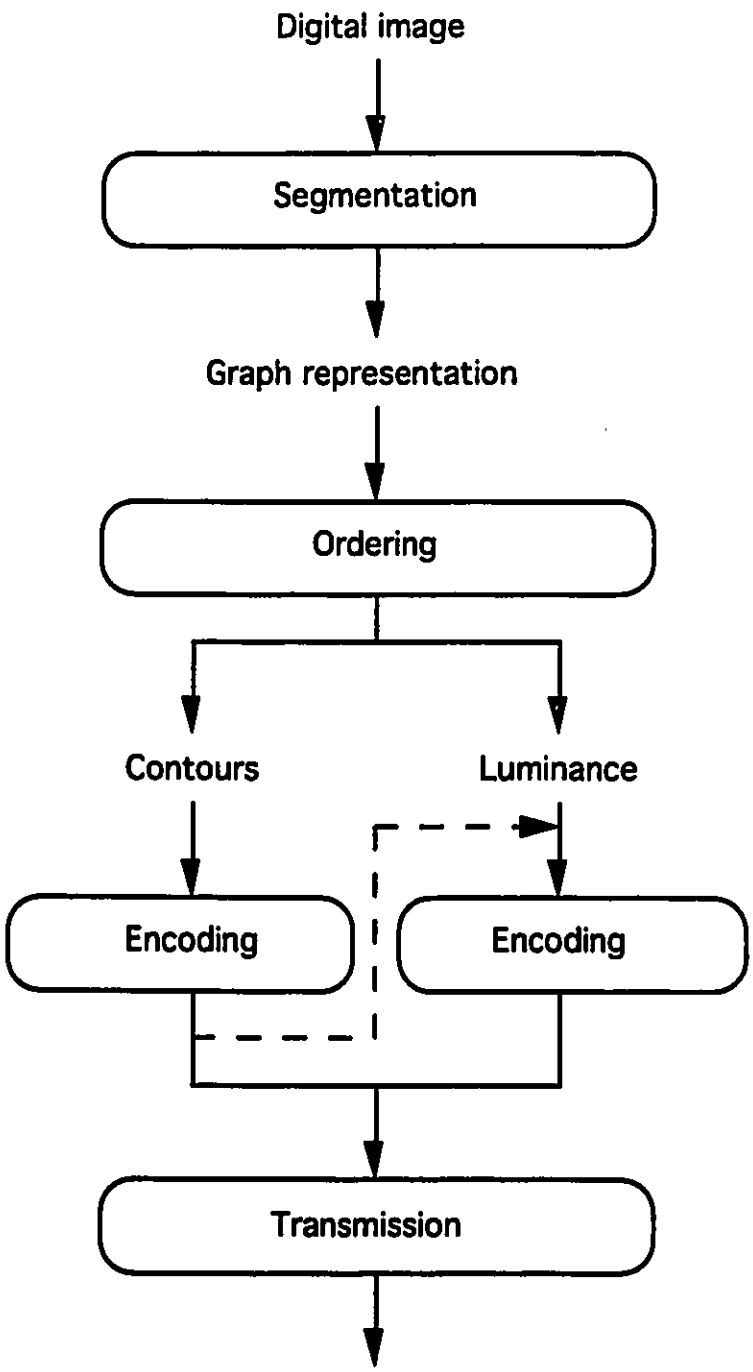


Fig. 1.2 PIT by Segmentation Coding.

simplified image is the basis of the new technique. The following three chapters are tutorial in nature.

Chapter 2 provides a review of progressive image transmission techniques. We first present the general principles behind information theory and data compression. We follow up with the principal symbol coding techniques: Huffman coding and arithmetic coding. We also stress the important role of data modeling in source compression. The remaining of the chapter is devoted to PIT, going from lossless schemes like bit-plane slicing and quadtree to lossy schemes, involving predictive coding, the Laplacian pyramid, vector quantization and transform coding, to more recent developments using the wavelet transform. We then describe JPEG, the standard for still image compression and transmission.

In Chapter 3, we review the tools of mathematical morphology, our favored methodology for image analysis. We start with the basic properties of set theory, the formalism on which mathematical morphology is based. We then introduce the basic operations of dilation and erosion and the composite operations; openings and closings. We also look at the geodesic operators that are used in the decoder, and we end the chapter with the presentation of the watershed transformation, the segmentation operator we used.

Chapter 4 is dedicated to the question of distortion measures. We begin the chapter with a description of the most widely used vision model. We then review the most used subjective measures: goodness and impairment scales, along with the accepted objective measures: mean square error and Signal to Noise Ratio (SNR). We close that chapter with the presentation of some proposed metrics that have not gained general acceptance.

In Chapter 5, we describe the proposed PIT methodology. First, we refine our image representation and explore the problem of segmentation with the watershed. Next, we show how the use of a proximity graph can simplify the problem of reordering the features to be transmitted. We then examine the specific problems of encoding

contour information and how redundancy can be eliminated. We end the chapter with the presentation of the decoder.

In chapter 6, we present the subjective testing and we analyze its results. We show how the parameters were chosen for the actual test and what the characteristics of the images were. Finally, we compare the performance of our algorithm with JPEG's.

In Chapter 7, we propose an objective distortion measure for highly compressed images and see how it correlates with the subjective results. We also compare the results with the mean square error measurement.

Finally, we present our conclusions in chapter 8.

Chapter 2

Progressive Image Transmission Review

In this chapter, we review the principal techniques for progressive image transmission. First, we introduce the general concept of entropy, used as a lower bound on the bit rate required for lossless encoding. Next, we see how source encoding can approach this limit. We then proceed with a brief description of PIT schemes, reviewing first simple methods used in lossless transmission, and finally, more evolved techniques using quantization to obtain higher compression rates. We close the chapter with a description of the JPEG standard and the provisions within it for progressive transmission.

2.1 Symbol Encoding

The problem of data compression is to produce a reversible mapping of a file, its content being text, image or any other type of data, into another file having a shorter length. The use of a short sequence of symbols to represent a longer sequence that

appearing frequently is the most intuitive solution to this problem. But what is the optimal solution?

2.1.1 Entropy

Let us consider the encoding of a source S having K possible symbols in its alphabet: $S(n) \in \{s_1, s_2, \ldots, s_K\}$ for all n. The special case of K=2 describes a binary source. We can view each symbol as an event having a probability $P\{S(n) = s_k\} = P(S_k) = P_k$. Shannon has shown [Sha48] that, for a memoryless source, i.e., if all the source samples are statistically independent of each other, the lower bound on the average length of a symbol of S is given by its average information content also called the entropy of S:

$$H(S) = -\sum_{k=1}^{N} P_k \cdot \log_2 P_k \text{ bits/symbol.} \tag{2.1}$$

H(S) is also the average of uncertainty about the source output before it is known. It can be shown that

$$0 \le H(S) \le \log_2 K. \tag{2.2}$$

The left side of the equality holds only if all probabilities are zero except one. This non zero probability has to be unity, implying a totally predictable source. The right of the equality holds, if and only if, all probabilities are equal: the source is totally unpredictable.

The redundancy of the source can be expressed as [Jay84]

$$R(S) = \log_2 K - H(S). \tag{2.3}$$

We can use Eq. 2.1 to calculate the entropy of an image by associating to each pixel value a probability of occurrence equal to its actual frequency of occurrence in that image. But, since this assumes that the pixels are identically and independently distributed, which is clearly not the case, the value obtained is of little value in determining a bound on the amount of information required to reproduce the image. The entropy of a source with memory, i.e., a source exhibiting statistical dependencies between successive samples, can be expressed in terms of block entropy. We can view M successive samples $S(n), S(n+1), \ldots, S(n+M-1)$ as a block or a source word. The probability of a specific source word W is $P(W)$, and the M-block entropy per symbol of the source is

$$H_M(S) = \frac{1}{M} E[-\log_2 P(m)] = \frac{-1}{M} \sum_{\text{all } m} P(m) \log_2 P(m) \text{ bits/symbol.} \qquad (2.4)$$

Let us consider an image source S generating images of size $R \times C$ pixels, with each pixel quantized to G gray levels. The probability of a particular image k out of $G^{R \times C}$ is given by $P\{S(n) = k\} = P(k)$ and the entropy of the image is given by

$$H(S) = \frac{-1}{R \times C} \sum_{\text{all } k} P(k) \log_2 P(k). \qquad (2.5)$$

Alternatively, the entropy of a source with memory can be expressed in terms of conditional probabilities. Let $H[S(n) | S(n-1), \ldots, S(n-M+1)]$ represent the conditional entropy of the source at time n, given the M-1 preceding symbols. It can be shown that

$$H[S(n) | S(n-1), \ldots, S(n-M+1)] \leq H_M(S), \qquad (2.6).$$

More generally, for two sources with identical alphabet and symbol probabilities,

$$H(S)|_{\text{source with memory}} < H(S)|_{\text{source without memory}}. \tag{2.7}$$

Hence the concept of entropy is useful only when used in conjunction with a model that correctly represents the source. In practice, it is often not possible to obtain a suitable source model for an image.

2.1.2 Huffman Coding

If the source statistics are known, it is possible to come arbitrary close to the entropy bound. For the encoding to be a reversible operation, it is necessary that a code be uniquely decodable. Huffman coding [Huf52] is an optimal mapping of each source symbol to an instantaneously decodable code word, i.e., knowledge of *S(n)* does not require knowledge of *S(n+1)*. Instantaneous codes are also known as prefix codes.
The construction of an Huffman code is straightforward and easy to implement. Given a set of possible events $\{s_1, s_2, \ldots, s_N\}$ arranged in order of decreasing probability, we replace the last two events with a new event with probability $P_{N-1} + P_N$, rearranging the probabilities of the new source in decreasing order. We iterate this procedure until only two events remains. Fig. 2.1(a) describes this source reduction process, sometimes called *Huffman contraction* [Pap91]. We assign the numbers 0 and 1 to the last digits of the code words of the events s_{N-1} and s_N. At each branch in the tree, we assign the numbers 0 and 1 to the left of the partially completed code word of each event on that branch. This process is illustrated in Fig. 2.1(b). The Huffman code so formed is the optimal prefix code. Since the assignment of 0's and 1's is arbitrary, as is the ordering of symbols with equal probabilities, an Huffman code is not unique.

However, placing newly formed symbols above older ones results in a code with minimum word length variance.

An Huffman code is truly optimal only if all the probabilities $\{P_1, P_2, \ldots P_N\}$ are integer powers of ½. In all other cases the average length of the Huffman code is greater than $H(S)$.

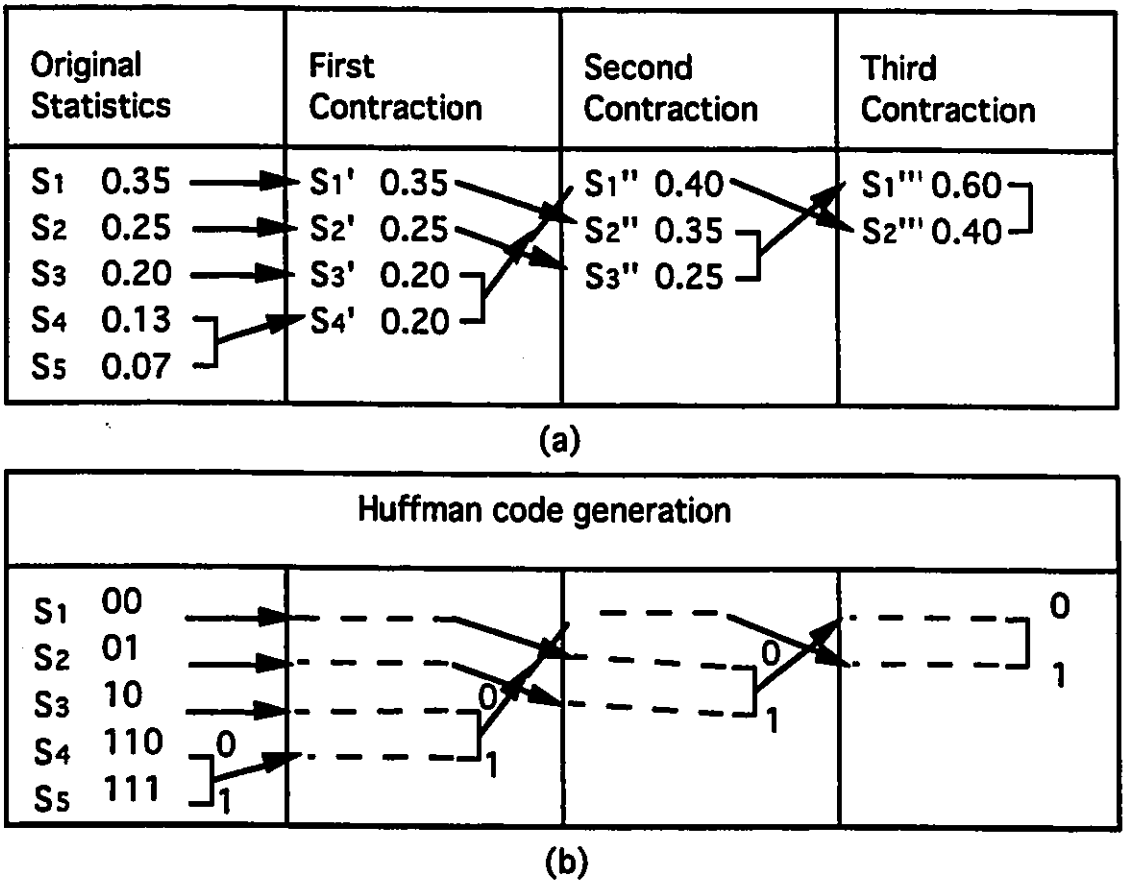


Fig. 2.1 a) Huffman contraction b) Huffman code generation

The efficiency of a prefix code can be calculated as follows:

$$Efficiency = \frac{H(S)}{R(S)} \times 100\%. \tag{2.8}$$

where $H(S)$ is the entropy of the source S and $R(S)$ is the average bit rate after encoding S, expressed in bits/sample [Pro89].

Nevertheless, an optimum solution can be approached by trading off on the instantaneous nature of the code: if we encode combinations of events instead of every single event, the resulting probabilities will be smaller, therefore, the difference between these probabilities and integer powers of ½ will also be smaller, implying a shorter average codeword length. Combining events in sequences of 2 symbols results in a second order Huffman code, combining events in sequences of 3 results in a third order Huffman code, and so on.

2.1.3 Arithmetic Coding

An arithmetic code can approach an optimal solution better than an n-order Huffman code [Lan84]. In fact, a pure arithmetic coder supplied with accurate probabilities provides an optimal compression and outputs at most $-\log_2 P_K + 2$ bits to encode a file with probability P_K. In Huffman coding, there is a one-to-one correspondence between a codeword and a source block, while, in theory, an arithmetic code assigns one codeword to an entire data set. The codewords consists of half-open subintervals of the half-open unit interval [0,1). They are expressed by specifying just enough bits to distinguish their corresponding subintervals from all other possible subintervals.

The basic algorithm for arithmetic coding is as follows: [Wit87]

1. Start with a current interval initialized to [0,1).
2. a) Subdivide the current interval into subintervals, one for each possible event. The size of an event's subinterval is chosen to be proportional to its probability.

b) Select the subinterval corresponding to the event that actually occurs and make it the current interval. Repeat 2 for all remaining events.

3. Output enough bits to distinguish the final output from all other intervals.

Fig. 2.2 shows an example of this procedure: the three sucessive events b1, a2, and b3 , with respective probabilities ⅓, ½, and ⅗ are represented by the binary string 11001.

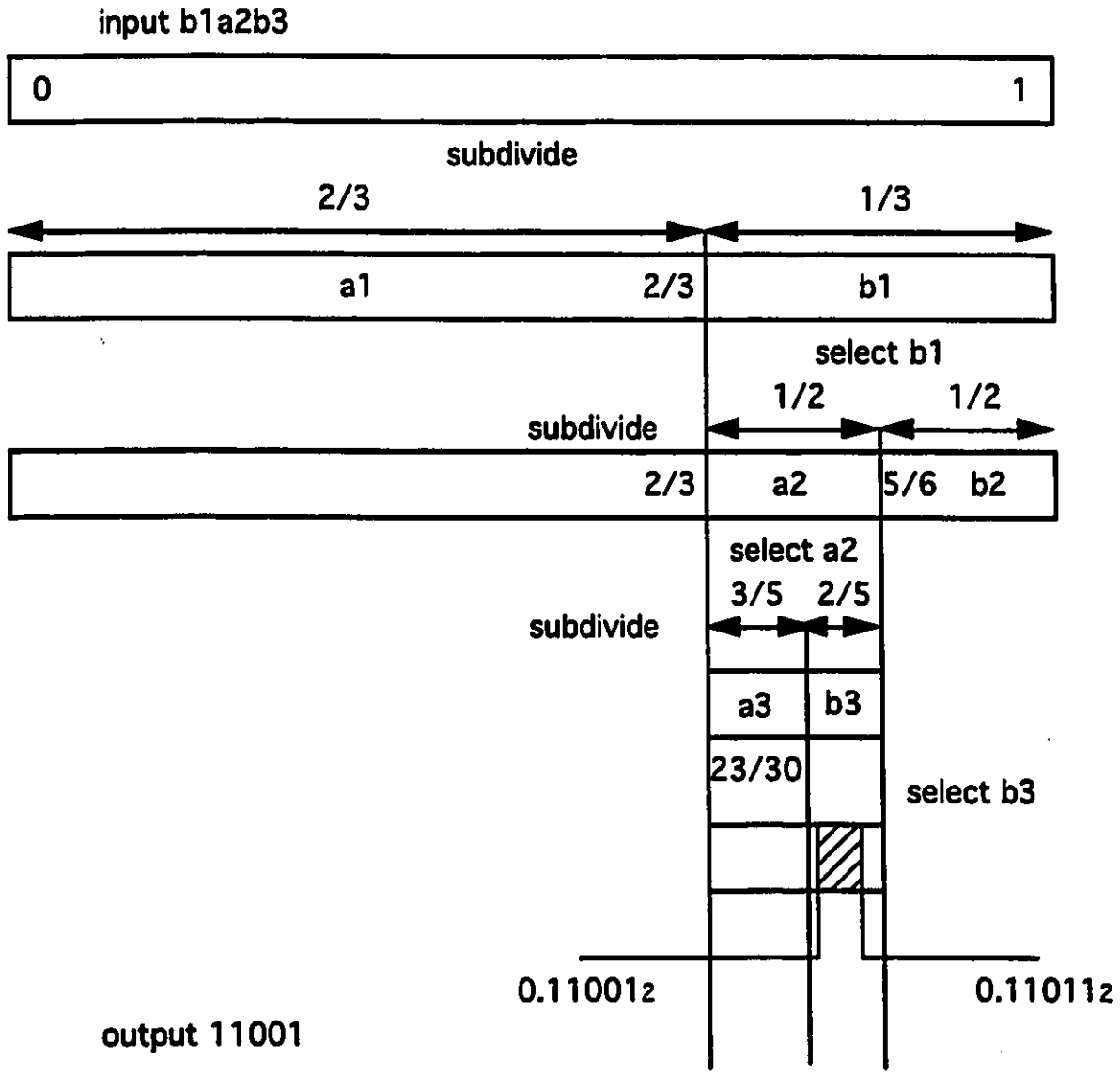


Fig. 2.2 Pure arithmetic coding.

In practice, the implementation encounters two major difficulties: the shrinking current interval requires the use of high-precision arithmetic, and no output is produced until all the file has been processed. The most straightforward solution to both these problems is to output each leading bit as soon as it is known, and then double the current interval so that it reflects only the unknown part of the final interval.

The use of floating point arithmetic for interval expansion leads to low speed implementations. Faster implementations use integer arithmetic and table-lookup approximation to give faster, yet less efficient conversions.

Finally, arithmetic coding is not instantaneous and either a word count or an end-of-sequence code is needed to ensure decodability. The main usefulness of arithmetic coding is in obtaining maximum compression in conjunction with an adaptive model or when the probability of one event is larger than ½ .

2.1.4 Statistical Modeling

The actual compression is optimal only with respect to the model used for the source. A source model is composed of both structural and probability components. The models can be *adaptive* (dynamically estimating the probability for each event based on preceding ones), *semi-adaptive* (using a preliminary pass to gather statistics), or *static* (using the same fixed probabilities for all encoding). Furthermore, the probability of an event can be based on its frequency within the data set, or based on a probability distribution whose parameters are estimated.

A static probability model entails the use of a fixed code, using either Huffman or arithmetic coding, which must be known by the decoder. An adaptive model uses a more complex data structure which is updated according to a method that the decoder can reproduce. Arithmetic coding is particularly well suited to this scenario, unlike

Huffman coding where the tables have to be regenerated at each update. Finally, a semi-adaptive model relies instead on transmitting the model data as side information. The actual performance of each type of model depends on the stationarity of the data, and how efficiently can model data be transmitted. In general, data is not stationary, and static models perform poorly [Ris79, JBI91].

Good compression requires models that go beyond global event counts and take into account the structure of the data. For lossless image compression, pixels are coded in a predetermined sequence, using linear prediction based on a fixed constellation of neighboring pixels. The prediction error is then encoded, using an exponential distribution with zero mean. For text, however, the best method involves changing structures. Typically, it is assumed that only some sequences will occur frequently. As encoding progress, we determine which sequences are most frequent. The Lempel-Ziv method [Ziv78] uses a fixed length codeword into which a variable number of source symbols is mapped. Lempel-Ziv has one advantage: speed. Better compression can be obtained using a Markov model of moderate order [Cle84].

A Markov source has finite memory. The probability of a given event depends upon the outcome of the last N events where N is the order of the Markov source. A binary first-order Markov source is depicted in Fig. 2.3.

In the special case of a source wich outputs a long repetition of a same symbol, the use of *run-length encoding* (RLE), in which N successive source codes are represented by a single code + a word count, can lead to substantial compression gain [Gol66]. RLE is routinely used in the transmission of facsimile, even though more evolved models now outperform it [Arp94].

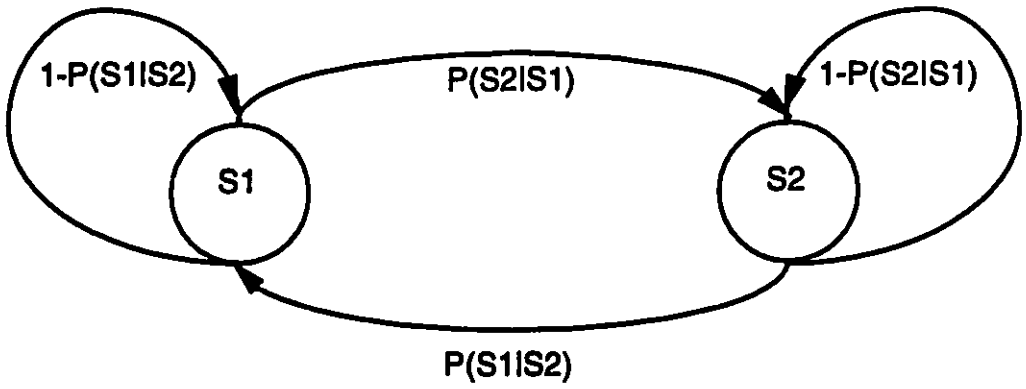


Fig. 2.3 A binary Markov source.

2.2 Lossless Progressive Image Transmission

Having surveyed the question of data compression, we can now turn our attention to progressive image transmission. Our interest here is limited to gray-scale images. Progressive transmission of bi-level images is a special case, and is supported by the *Joint Bi-level Image Group* (JBIG) standard [JBI91]. Progressive transmission of color images is somewhat more complicated and is beyond the scope of this research. We start by looking at simple techniques in which the image information is reordered but where no compression is performed. This corresponds to the left side of Fig. 1.1. We can clearly identify two approaches: in the first one, the pixel values are gradually refined, while in the second, intensity values are assigned to large image areas that are gradually sub-divided to eventually form the actual pixels.

2.2.1 Bit-Plane

The bit-plane method is both the simplest and the most intuitive form of progressive image transmission [Tzo87]. In this approach, an N gray-scale image is viewed as a series of N binary images, or bit-planes, stacked one on top of the others. This is illustrated in Fig. 2.4.

The most significant bit-plane is transmitted first and permits the reconstruction of a binary image at the receiver. The other planes are then used to refine the pixel intensities [Sch66]. Each value at the receiver, after the transmission of a particular plane, corresponds to an interval of the N input values. To minimize distortion, we represent each possible receiver value by the mean of all the input values falling within the corresponding interval. E.g., the first bit plane has two intervals [0,127] and [128,255] which are represented by the values 63 and 191 respectively. This is, in essence, the same as applying a uniform midrise quantizer [Jay84].

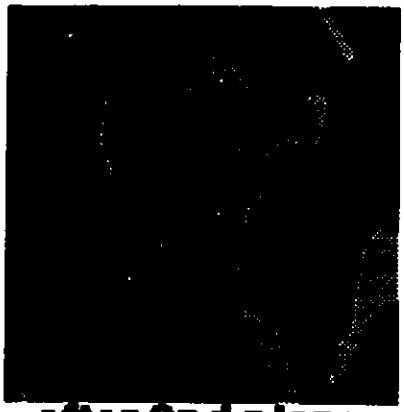
The performance of the method is very poor since the first image obtained is of poor quality and requires 1 bit/pixel. It is possible to apply lossless compression to each bit-plane, as is suggested by the JBIG workgroup [JBI91]. But since the correlation between neighboring pixel quickly fades as we move from one plane to the next, the overall compression ratio remains poor.

Still, the technique has very low computational requirements, which motivated a variant called dynamic binary thresholding in which the intervals and representation values are selected adaptively [Cha94]. With this method, the first approximation is a binary picture transmitted at less than 0.3 bits/pixels, while the refinements are produced by transmitting the positions of the pixels falling into the new intervals.

after 1st plane

191	191	191	191
191	191	191	191
191	191	191	63
63	63	63	63

top rightmost corner



after 2nd plane

159	159	159	159
159	159	159	159
159	159	95	95
95	95	31	31

after 8th plane

163	168	169	127
163	168	169	127
141	144	122	75
93	68	49	46

Fig. 2.4 The bit plane method.

2.2.2 Tree-Structured Pyramid

In a pyramid scheme, the image is gradually reduced in size. A set of neighboring pixels, called the sons, are reduced to a single pixel, called the father, on the higher level. A small image is first transmitted, followed by the incremental information necessary to double its size until the full size image is recreated. The pyramid data structures were proposed by Sloan and Tanimoto [Slo77], when they introduced the concept of progressive image transmission. There are many ways to implement a reduction rule to create the fathers from the sons: subsampling, averaging, median filtering, to cite a few.

Usually a quadtree structure is used, as depicted in Fig. 2.5, where each father is constructed from four sons [Str91]. Alternatively, a binary pyramid where the image is alternatively decimated in horizontal and vertical directions is sometimes preferred [Kno80].

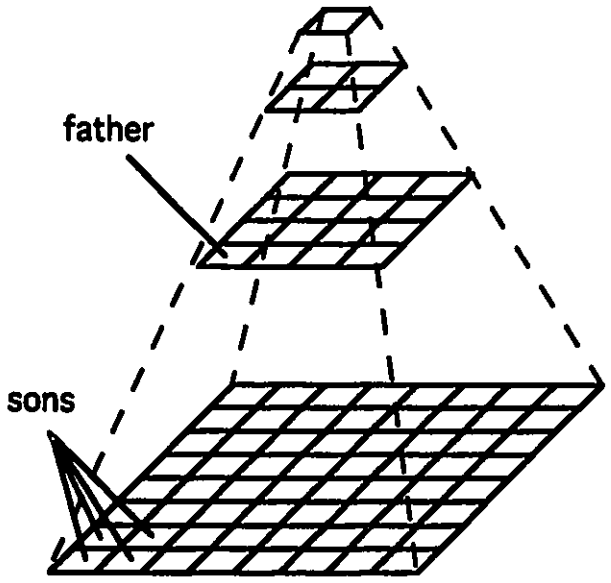


Fig. 2.5 Quadtree representation of an image.

Like the bit-plane method, the tree-structure pyramid alone does not achieve any compression. In fact, unless the difference between each successive level is transmitted and the information contained in the previous level is efficiently used, the pyramid will result in an expansion of the data set by as much as 33%. When implemented by transmitting differences between each level, the pyramid data structure results in good intelligibility at a bit rate of 0.125 bits/pixel [Tzo87].

Here again, it is possible to use entropy coding to achieve lossless compression. Since the spatial redundancy at various resolutions can be exploited, the image can be encoded at approximately 3 bit/pixels.

Wang and Goldberg have developed a method using vector quantization, a technique that we review in section 2.3.2, to represent the difference image at each level. The initial approximate image is constructed from sequences of mean value over 2×2 pixel blocks [Wan89].

2.3 Lossy Progressive Image Transmission

Where lossless compression offers only modest results, lossy compression trade some *possibly* visible degradation, in order to reach higher compression ratios. The term lossy reflects on the nature of the encoding, in all such schemes lossless compression is possible by transmitting the residual error image at the end. Performance is then non-optimal. At the heart of lossy compression is the mapping of many input values to a common single output value arbitrarily close to each input value. This is usually done through the use of a quantizer, such as the one of Fig. 2.6. Given $\{t_k, k = 1,..., L+1\}$, a set of increasing *decision levels*, if an input value x lies in the interval $[t_k, t_{k+1})$, then it is mapped to $Q(x) = r_k$, the *kth reconstuction level.*

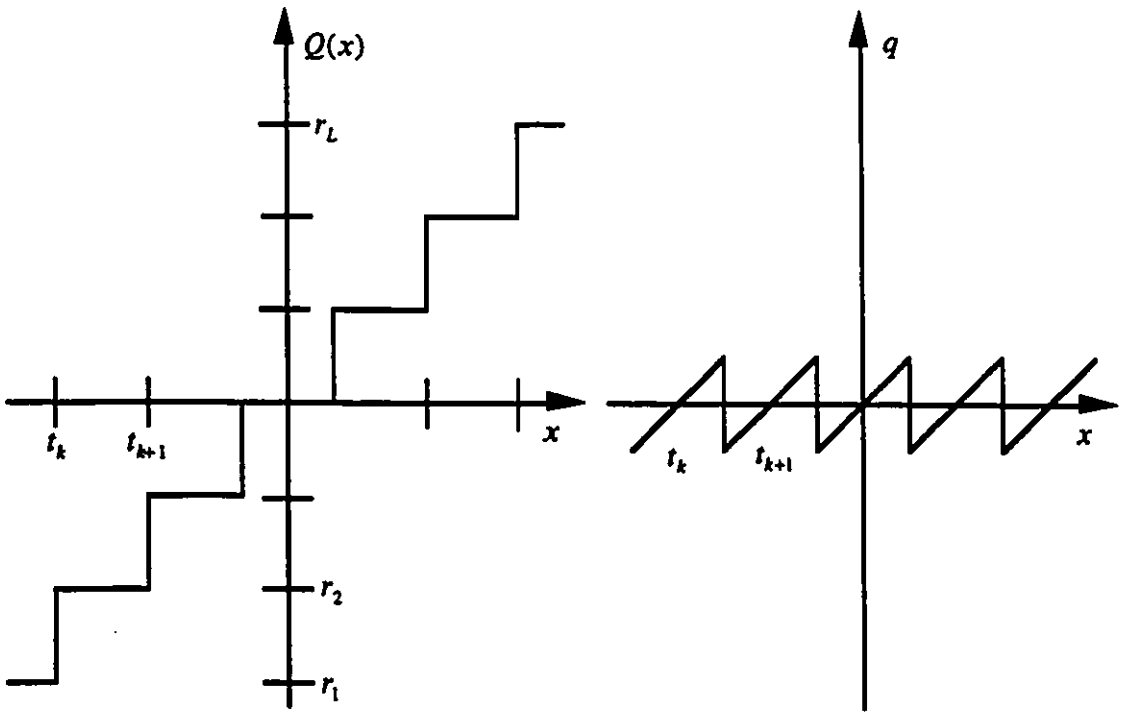


Fig. 2.6 A uniform midtread quantizer and its quantization error.

There are two ways to transmit progressively using a quantizer: multistage quantization and embedded quantization.

In multistage quantization, the residual error of a quantizer on one stage is fed to a finer quantizer on the next stage. The output of the coarser quantizer is used as an approximation which can be refined by adding the outputs of the other quantizers.

In embedded quantization, the decision levels of a coarse quantizer are aligned with those of finer quantizers. The reconstruction levels of a coarser quantizer are labeled as prefix of the corresponding reconstruction levels on the finer quantizers, as shown in Fig. 2.7. The recontruction level labels can therefore be transmitted progressively.

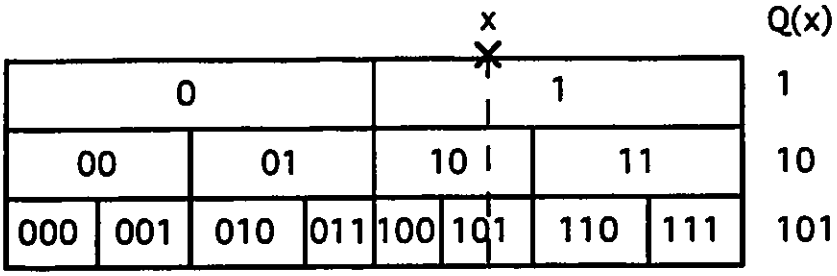


Fig. 2.7 An example of embedded quantization.

2.3.1 DPCM

Differential Pulse Code Modulation (DPCM) is the basic scheme used in predictive coding, a popular technique used extensively in lossy compression of one-dimensional data, specifically audio signals [One76]. The next pixel to be transmitted, x_n, is predicted using some of the pixels already transmitted, and the prediction error, $e_n = x_n - \hat{x}_n$, is quantized and encoded. A 3-bit quantizer is sufficient to reconstruct an image with good quality [Jai89]. Predictive coding is also the state of the art in lossless compression of images [Roo88]. The number of quantizer levels is increased so that the original digital image can be exactly reproduced. An entropy coder is then used to reduce the bit rate. Many variations are possible, depending on the constellation of the pixels used for prediction, the prediction technique, the quantizer used, and so on... [Jay84].

DPCM can be adapted to PIT using either the multistage approach, as outlined in Fig 2.8 or the embedded strategy. In the later case, the quantization levels can be transmitted a bit at a time very much like the actual pixel values in the bit-plane method.

Performance is also similar to the bit-plane method, since in either case the initial image requires 1 bit/pixel and as does each refinement. The initial image can be

represented with only 3 bits/pixel and the quality of the first image is higher than in the bit-plane method.

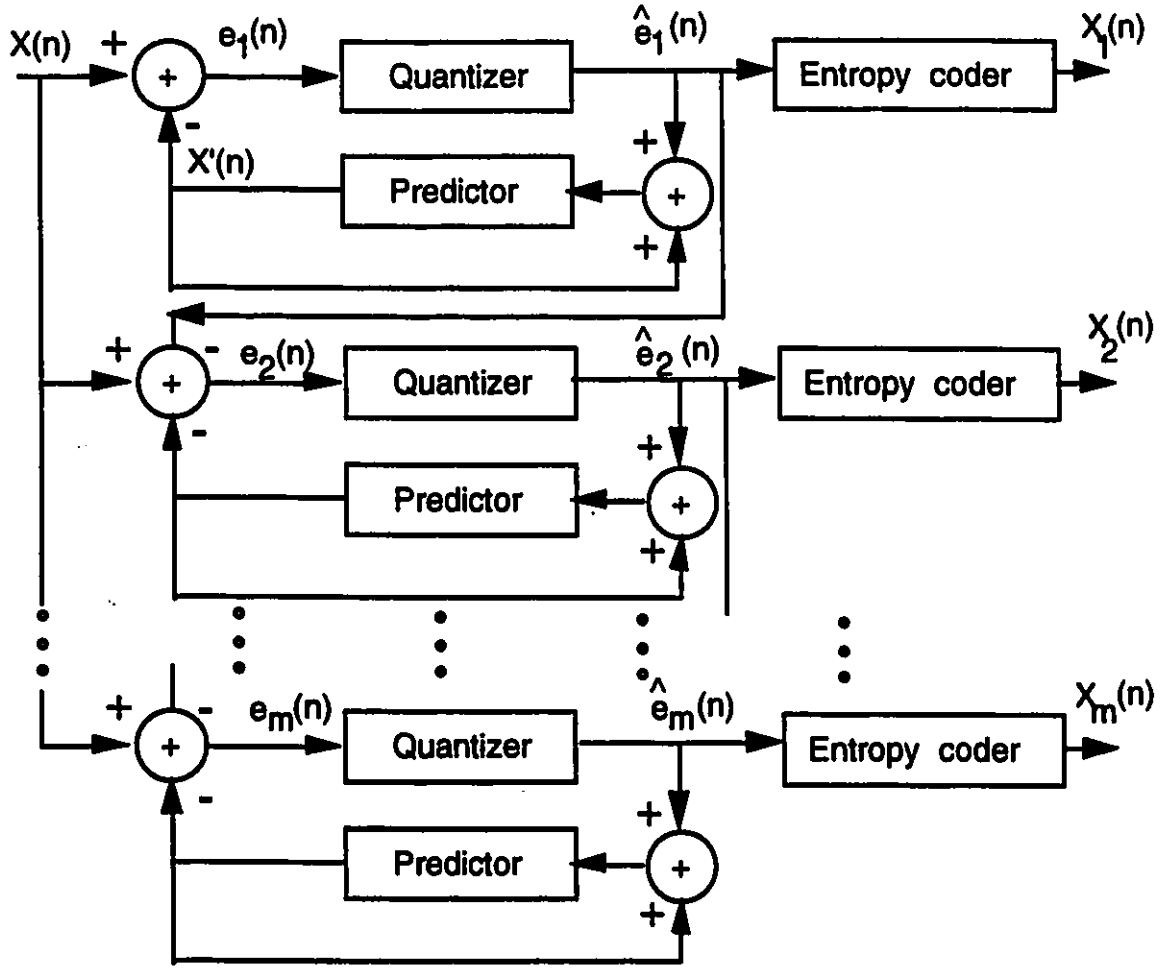


Fig. 2.8 Multistage DPCM.

2.3.2 Vector Quantization

Vector quantization (VQ) is a powerful way of going beyond the 1 bit/pixel representation.

In VQ, a block of adjacent pixels, called a vector, is represented by a label associated with a codeword. The set formed by all recognized codeword is called the codebook. Relatively high compression can be attained if the number of codewords is much smaller than all the possible combinations of pixel values in the vector, e.g., for a block of 4×4 pixels at 8 gray levels ($16\times8=128$ bits) if there is only 16 words in the codebook, then the image can be represented at $4/16=0.25$ bits/pixel.

The difficulty in VQ resides in the choice of the codebook to use. A universal codebook is a by definition non-optimal since it fails to exploit the statistics of all possible images. Adaptive VQ requires transmitting the codebook as side information, an additional step often costly in performance. In practice, however, it is often possible for a given application to use a near optimal codebook. The design of a codebook is usually based on a clustering technique. The LBG algorithm [Lin80] is the one most often used. Given an initial codebook C_0, containing N initialized codewords and a set of training vectors, each training vector is associated with the newest neighbor codebook. All the codewords are then modified so that each codeword is the centroid of the training vector set associated with it. The procedure is iterated until the difference between two successively generated codewords is sufficiently small.

Encoding is done by locating the best codeword for each input vector using generally a mean square criterion and an exhaustive search. Decoding is a simple look up operation of the label to find the codeword.

Here also, the method can be adapted to PIT using either a multistage or an embedded strategy. In the first case, the residual error left after encoding the image with a very small codebook is reencoded in the next stage. This process, called *residual error vector quantization* (REVQ) [Wan86], is demonstrated in Fig. 2.9.

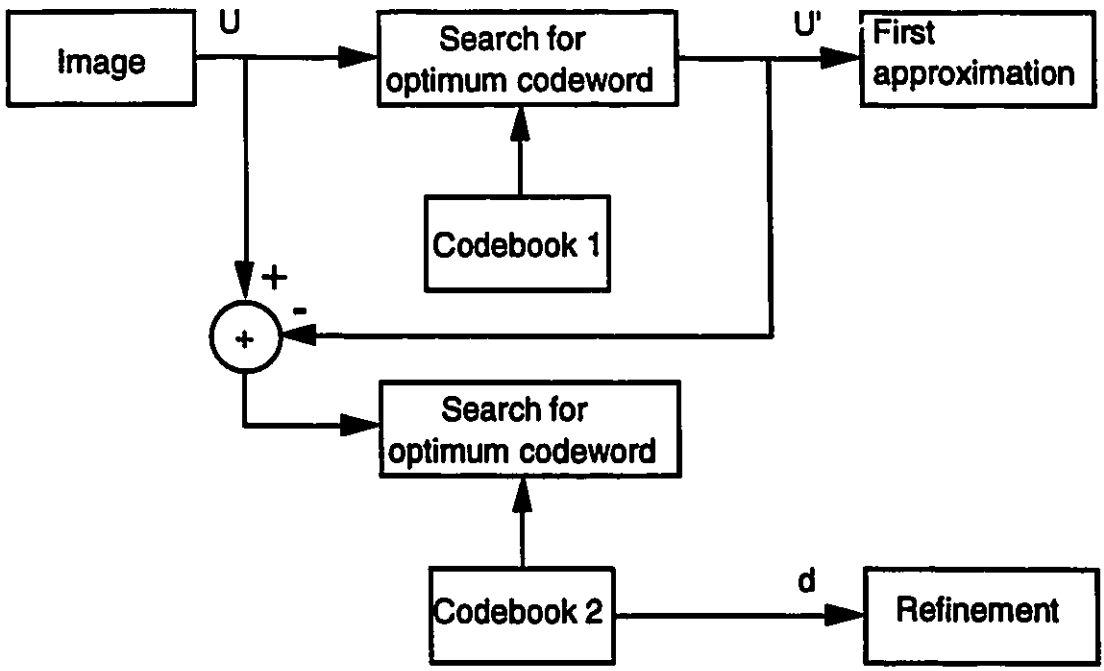


Fig. 2.9 Two-stage REVQ.

In the alternate approach using embedded quantization, the first bits of the label of a codeword at a given resolution form the label of a codeword at a lower resolution. This technique produces more distortion at the lower resolution and is called *tree-structured vector quantization* (TSVQ) [Nas88]. A simple codebook is illustrated in Fig. 2.10.

Performance of VQ is superior to predictive coding. The disadvantage of the method resides in its computational complexity.

2.3.3 Transform Coding

Transform coding involves the division of an image into a set of nonoverlapping square cells, usually 8×8 pixel blocks, linear transformation of each cell, and the quantization

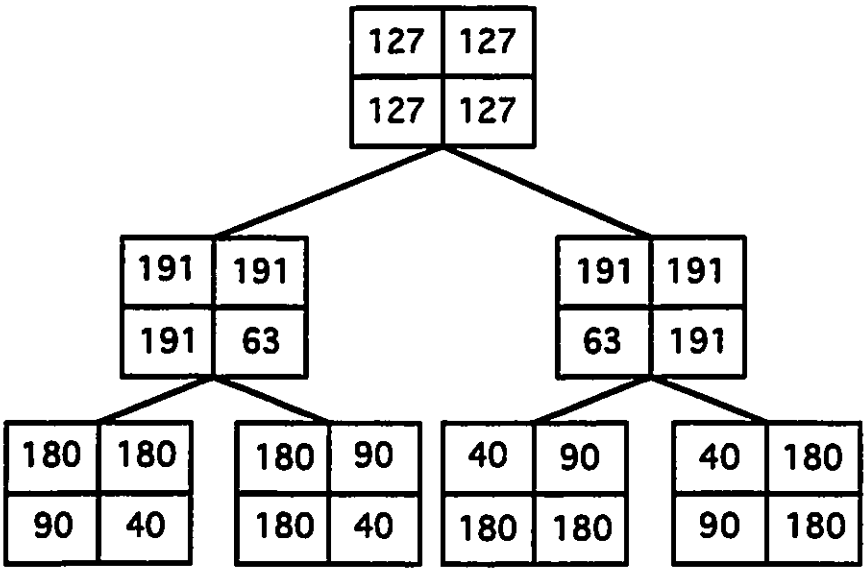


Fig. 2.10 Simple binary TSVQ codebook.

of the resulting coefficients, as shown in Fig. 2.11. The use of 2D-orthogonal transforms decorrelates the data in packing most of the signal energy into the low-frequency coefficients. As a result, the high frequency coefficients can be coarsely quantized without introducing highly visible distortion.

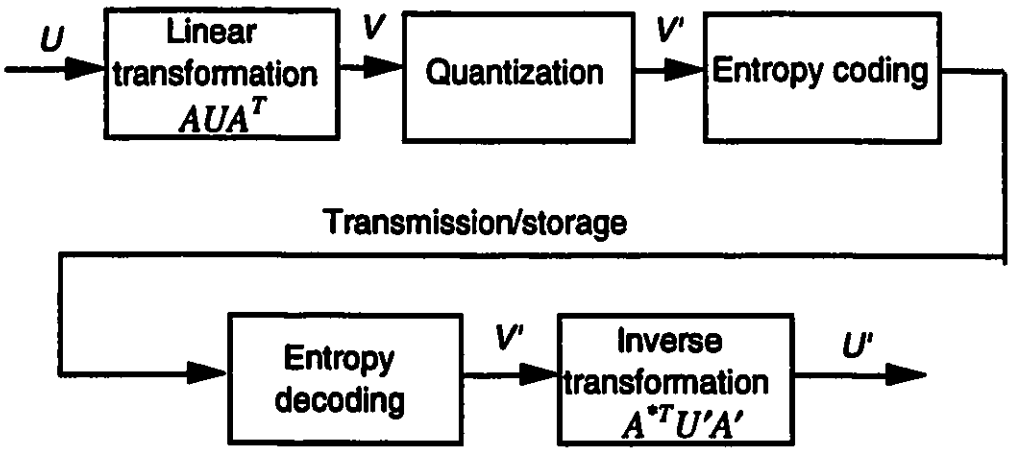


Fig. 2.11 Block diagram of transform coding.

Orthogonal transforms have the following properties:

- The average energy of the transform coefficients is equal to the average energy of the input image pixels.
- The average reconstruction error variance, σ_r^2 is equal to the average quantization error variance, σ_q^2 i.e.,

$$\sigma_r^2 = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \sigma_{r(i,j)}^2 = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \sigma_{q(i,j)}^2 = \sigma_q^2 \tag{2.9}$$

The optimal transform, i.e., the one that completely decorrelates the transform coefficients, the *Karhunen-Loeve Transform* (KLT), is not employed in practice because of its high computational complexity [Jai74]. The *Discrete Cosine Transform* (DCT) is generally preferred because it comes close to the performances of KLT and can be efficiently implemented. The two-dimensional DCT of a sequence $f(j,k)$ for $j,k=0,1....,N-1$ is defined by [Ahm74]:

$$F(u,v) = \frac{4C(u)C(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right], \tag{2.10}$$

for u,$v=0,1,...N-1$,

where $C(w) = \begin{cases} 1/\sqrt{2} & \text{for w=0} \\ 1 & \text{otherwise} \end{cases}$.

The inverse transform (IDCT) is given by:

$$f(j,k) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2j+1)v\pi}{2N}\right], \tag{2.11}$$

for $j,k=0,1,...,N-1$.

Bit allocation in transform coding is the key to high compression. It generally implies that the transform coefficients be quantized on different numbers of bits, allowing more to the ones where the image energy is concentrated. Let B_{ij} be the number of bits allocated to the (i,j)th transform coefficient, the optimum bit allocation, in virtue of the second property we mentioned, is given by [Jay84]:

$$B_{ij} = B + \frac{1}{2}\log_2 \frac{\sigma_{ij}^2 \varepsilon_{ij}^2}{\left[\prod_{k=0}^{N}\prod_{l=0}^{M} \varepsilon_{kl}^2 \sigma_{kl}^2\right]^{1/N \times M}}, \tag{2.12}$$

where B is the average bit allocation, and both the coefficient variance σ_{ij}^2 and the quantizer performance factor are proportional to the number of quantization levels $2^{B_{ij}}$. In addition, if a significant number of coefficients are quantized on a single bit, resulting in long runs of zeros and ones, a combination of RLE and Huffman coding can lead to high compression [Che84].

For progressive transmission, the transform coefficients must be transmitted in a number of passes, each one providing information on some of the coefficients in all the cells. The simplest strategy is to transmit the first N coefficients in the first pass, N more in the second pass, and so on. To do this, we need to order the coefficient so as to transmit the important ones first. A simple, yet near optimal way of achieving this is through the use of the zig-zag scan pattern shown in Fig. 2.12.

Better performance can be obtained by using multistage quantization on the transform coefficients. A coarse quantizer which allocates more bits to the lower frequency components is used on the first quantization and more uniform quantization is used thereafter on the residual coefficients.

Embedded quantization is, however, a better solution since is requires far less computation. In this case, the coefficients are transmitted using a bit-plane approach.

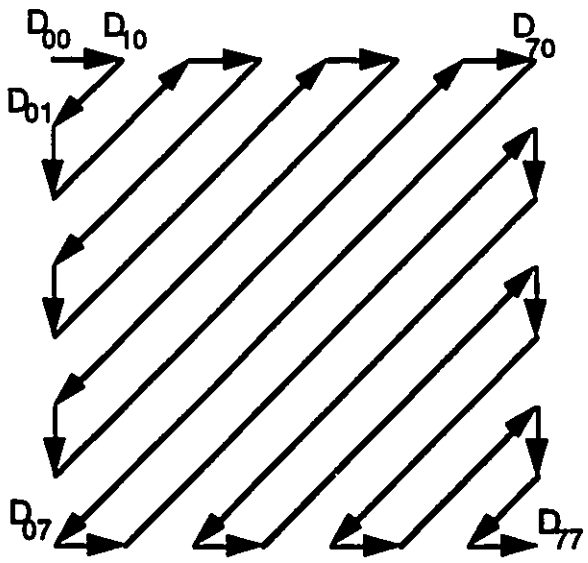


Fig. 2.12 Zig-zag scan pattern.

The same technique used to determinate the bit allocation is then used to determine in what order each bit are transmitted, as shown in Fig. 2.13.

As long as the optimal bit allocation is used for each variant, performance is related to the amount of overhead introduced in transmitting the coefficients in multiple passes. The scanning pattern method introduces the least amount of overhead, while the bit-plane method leads to less distortion at the lower bit rate reproductions. Multistage quantization is computationally more complex than the two other techniques, and gives intermediary results. We show how the JPEG standard deals with this issue in Section. 2.4.

Transform coding can represent an image without noticeable impairment at 1 bit/pixel and does not require as much computation as VQ encoding. Its performance at low bit-rate, however, suffers from blocking impairment, a condition arising from the visibility of the transform blocks when the coefficients are too coarsely quantized.

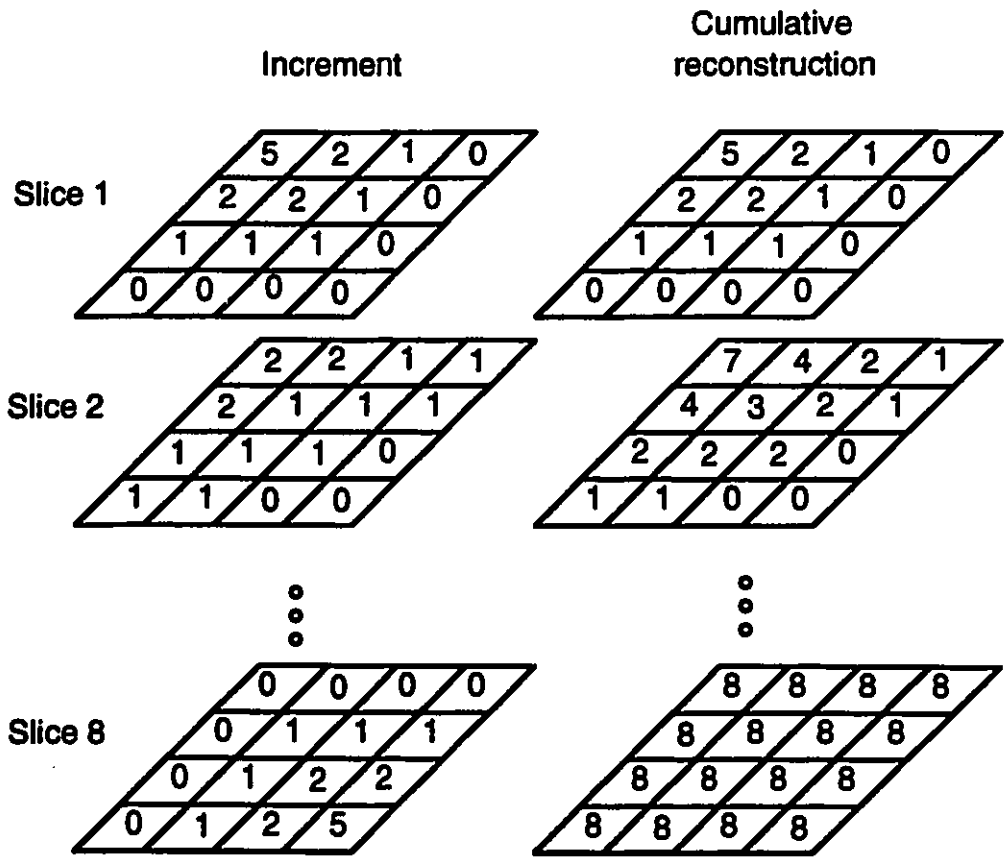
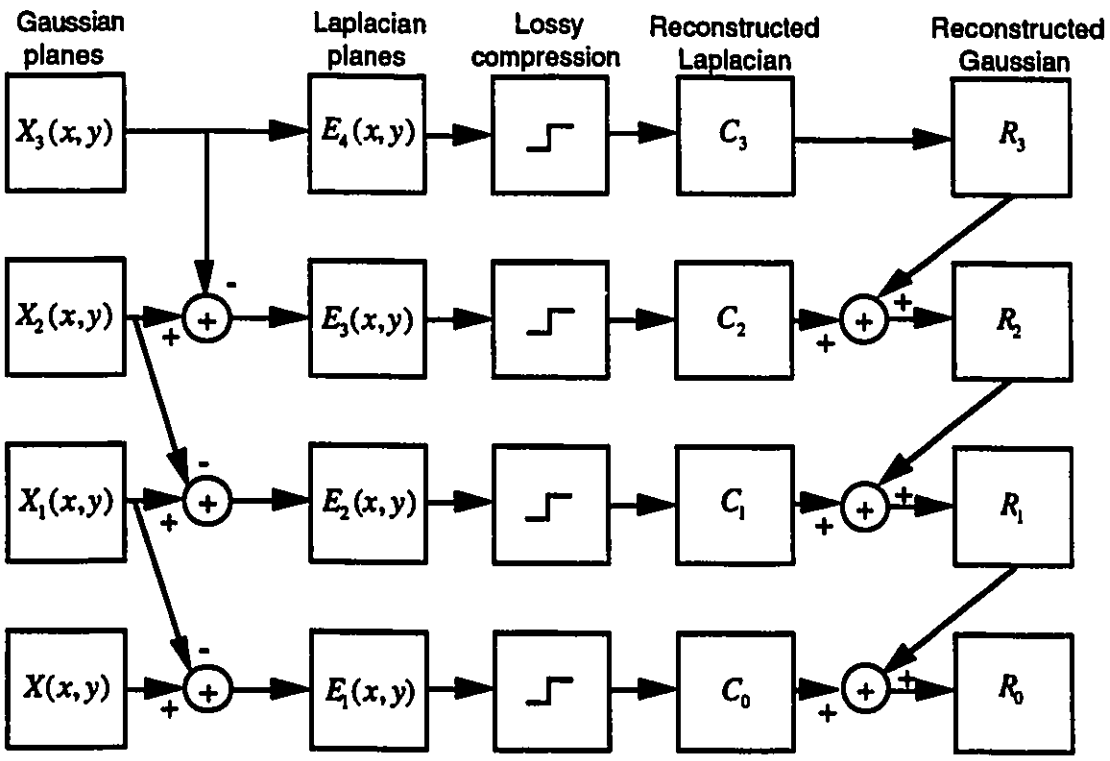


Fig. 2.13 Bit slicing.

2.3.4 Filtered Pyramid

Burt and Adelson have developed a more general way of constructing pyramids [Burt83]. To generate a higher level image, they used a low-pass filter followed by subsampling. The difference between an image and the expanded image on the higher level is transmitted to gradually reproduce the original image. They called "Gaussian pyramid" the set of subsampled image, and "Laplacian pyramid" the set of difference images. The encoding process is outlined in Fig 2.14.



2.14 Block diagram of the Gaussian pyramid technique.

Using 2D-Quadrature Filters (QMF), instead of low-pass filters, leads to better compression. QMF separates the image into high-band and low-band signals in both horizontal and vertical direction, producing 4 smaller images as shown in Fig. 2.15. Each of three sub-images in a given level of the pyramid has a different frequency content and can therefore be compressed using a model tuned to its statistics. Pyramid decomposition by QMF is a subset of the wavelet theory, which is more general in scope. The wavelet transform is obtained by dilating with a scale factor 2^j a two-dimensional function satisfying a number of mathematical constraints [Rio91]. Daubechies wavelets are the most commonly used [Dau88]. The advantage of the wavelet is that it allocates more bits to irregularities and less to encode regular features. This leads to better subjective quality at low bit rate.

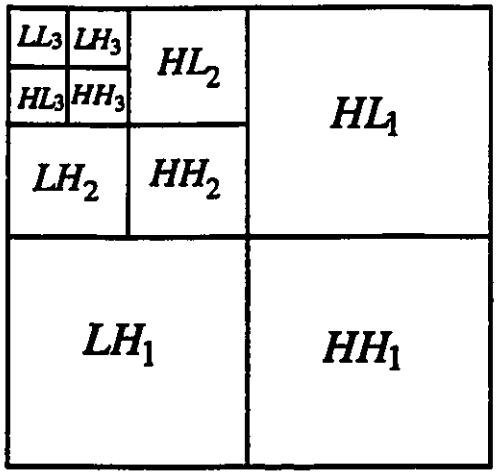


Fig. 2.15 Multiresolution decomposition with QMF.

The *discrete wavelet transform* DWT is defined by :

$$f(t) = \sum_{j,k=-\infty}^{\infty} a_{j,k} \Psi_{j,k}(t), \tag{2.13}$$

where a is a DWT coefficient, j represents the frequency and k the location.

The highest level Gaussian image is DPCM coded and the Laplacian images are coded using either RLE encoding, DPCM or DCT. By using very large filters, e.g. 16×16 or 32×32, this method can easily outperform DCT coding at the price of higher computational complexity. It can yield high quality at 1 bit/pixel. Images are still intelligible at 0.1 bit by pixel. Use of the wavelet can preserve intelligibility at less than 0.2 bit/pixel. This can be achieved by exploiting the redundancy between coefficients in the same location at different frequency using a quad-tree structure for predictive coding [Lew92] or VQ [Wan93].

2.4 The JPEG Standard

The JPEG standard for still images, named after the Joint Photographic Expert Group is a joint initiative of ISO (International Standard Organization), and the CCITT (Consultative Committee on International Telephony and Telegraphy), known today as the ITU-TS (International Telecommunicaton Union-Telecommunication Standardization). The objectives of JPEG were to develop a standard for the exchange of compressed images using technology near the state-of-the-art, yet simple enough to be implemented across a broad array of computing devices, including the average home computer.

The JPEG standard specifies four modes of operation:

1) Sequential encoding: each image is encoded in a single left-to-right, top-to-bottom scan;

2) Progressive encoding: the image is encoded to support progressive transmission;

3) Hierarchical encoding: the image is encoded at multiple resolution so that a low resolution version can be obtained without having to access any of the higher resolution information;

4) Lossless encoding: the reconstructed image is exactly the same as the original image.

While the lossless mode uses predictive coding, the three other modes are based on DCT coding.

2.4.1 DCT-Based Coding

Figs. 2.16 and 2.17 show the key processing steps of the DCT-based implementations. An important point is that most parameters are not fixed by the standard, which means that different implementations produce different output files and different reconstructed images, depending on the particular choice of the DCT approximation, the quantization tables, and the parameters used for entropy coding. The JPEG standard supports multi-component pictures by either coding each component one at a time, or by interleaving the 8×8 sample blocks.

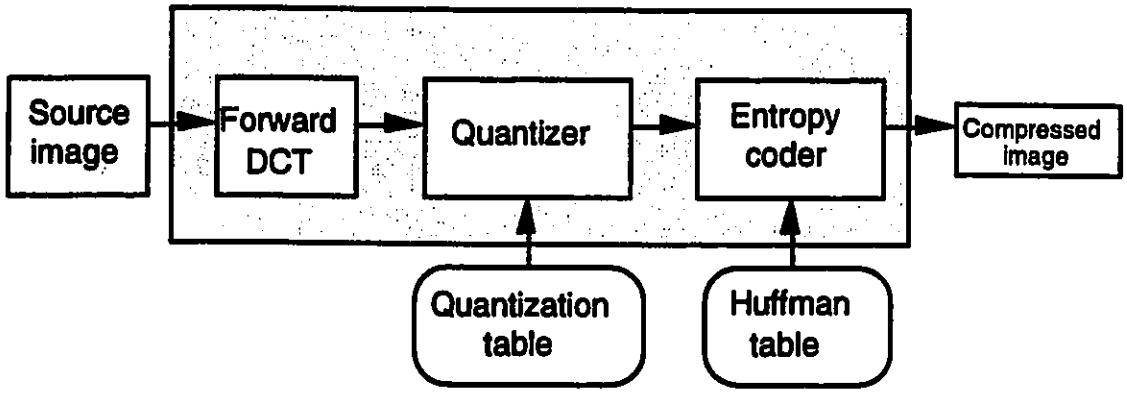


Fig. 2.16 Block diagram of DCT-based coder.

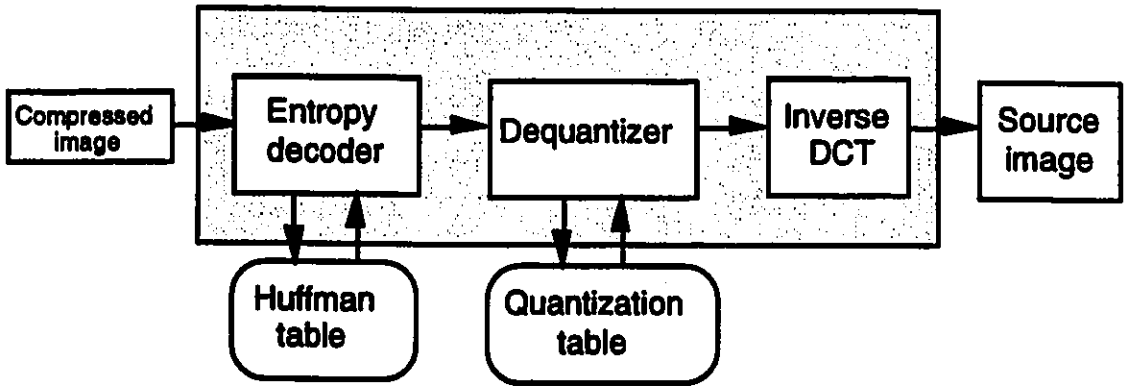


Fig. 2.17 Block diagram of DCT-based decoder.

The DCT coefficients are quantized according to a quantization table where the number of quantization levels attributed to each spatial frequency can vary from 1 to 255. The quantized value can be obtained by dividing the coefficient by its corresponding quantizer size followed by rounding to the closest integer. The output value is normalized by the quantizer size at the decoder. Some experiments to set each step size at the "just noticeable threshold" have resulted in a set of quantization tables (CCIR-601) which are made available, but are not part of the standard.

After quantization, the DCT coefficients are treated differently from the other 63 coefficients. Because the DC coefficient is in fact the average value of all the pixels in the block, there is usually a strong correlation between the DC coefficients of adjacent blocks. To take advantage of this, the DC coefficients are encoded as the difference between successive DC coefficients in the scan order. This is illustrated in Fig. 2.18. The other coefficients are ordered according to the zig-zag pattern as shown in Fig. 2.12. This ordering places the coefficients which are most likely to be nonzero first in the scan order. The coefficients are then entropy coded in two steps. The first step is to convert the zig-zag scanned coefficients into a set of intermediate symbols. The second is to convert these symbols into a bitstream in which the symbols are compactly represented. JPEG specifies no final entropy coding techniques, but makes available a set of Huffman tables, a few probability symbols for adaptive Huffman coding, and a method for adaptive arithmetic coding.

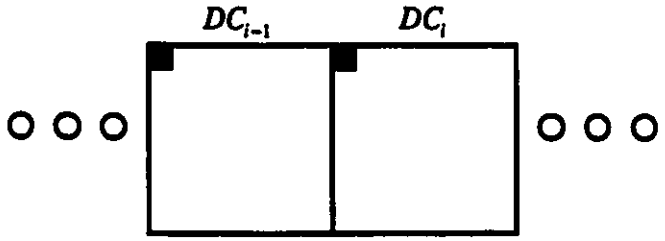


Fig. 2.18 Encoding of DC coefficients.

The form and definition of the intermediate symbols are dependant of both the mode of operation and the entropy coding method. We consider the Baseline method as an example of intermediate symbol structure. The Baseline method, the most widely implemented JPEG method, is a subset of the sequential mode for images with 8-bit samples and uses Huffman coding. In the intermediate symbol sequence each nonzero coefficient is represented by a combination of two symbols. The first one is a pair formed by the number of zero-valued coefficients preceding it and the number of bits used to encode its amplitude as can be seen in Table 2.1. The second is the actual amplitude of the nonzero coefficients . The DC coefficients are encoded similarly, but the first symbol represents only the size.

Size	Range
1	-1,1
2	-3,-2,2,3
3	-7,...,-4,4,...,7
4	-15,...,-8,8,...,15
5	-31,...-16,16,..31
6	-63,...,-32,32,...,63
7	-127,...,-64,64,...127
8	-255,...,-128,128,...255
9	-511,...,-256,256,...,511
10	-1023,...,-512,512,...,1023

Table 2.1 Baseline Entropy Coding Symbol Structure.

Variable length coding is used only to encode the first symbol; the second is simply represented using the specified number of bits. The entropy coding method used must

be specified externally, but the representation of the coefficients using a variable number of bits is set by the standard.

2.4.2 Progressive Mode

The DCT progressive mode of the JPEG standard uses the same transformation and quantization steps as the sequential mode. The key difference is that each image is encoded in multiple scans using either the scanning pattern approach of section 2.3.3, called here *spectral selection* or the embedded quantization approach refered to as *successive approximation*, or a combination of the two. Both modes are shown in Fig. 2.19

2.5 Interlaced GIF

Thanks mostly to the rapid expansion of the World Wide Web, progressive image transmission is readily available on the Internet in the form of interlaced GIF images. In a GIF image [GIF89], the pixel values are losslessly encoded sequencially using a variation of the Lempel-Ziv compression algorithm. In an interlaced GIF image, the rows are arranged according to a scheme similar to the one used in the bit-plane methodology:

Group 1: Every eighth row, starting with row 0.

Group 2: Every eighth row, starting with row 4.

Group 3: Every fourth row, starting with row 2.

Group 4: Every second row, starting with row 1.

The image is then encoded in four passes, each one corresponding to a group of rows. Since the average compression rate of a GIF image is approximately 3 to 1, the coarsest image achieves an effective compression ratio of about 24 to 1. This is far less than JPEG, which for the same gray scale image achieves a compression ratio of 512 to 1 on the lowest resolution image. In the light of its poor compression factor, the interlaced GIF format will not be discussed in more details. We must note, however, that it is a convenient format for image requiring lossless compression.

2.6 Summary

In this chapter, we saw how entropy can quantify the information content of a source and therefore offer a bound on the number of bits required to encode it.

We also showed the essential differences between Huffman and arithmetic coding and under which conditions one should be preferred to the other. Huffman coding is simpler and easier to implement, but is not efficient if one symbol has a probability much greater than ½ . Arithmetic coding is slightly more efficient than Huffman and is better suited to adaptive compression.

We followed with a review of the principal techniques used in progressive image transmission.

We first considered techniques in which the information is progressively transmitted without compression. Subsequent use of lossless compression leads to modest compression ratios, and techniques based on spatial improvement outperform schemes based on pixel refinement.

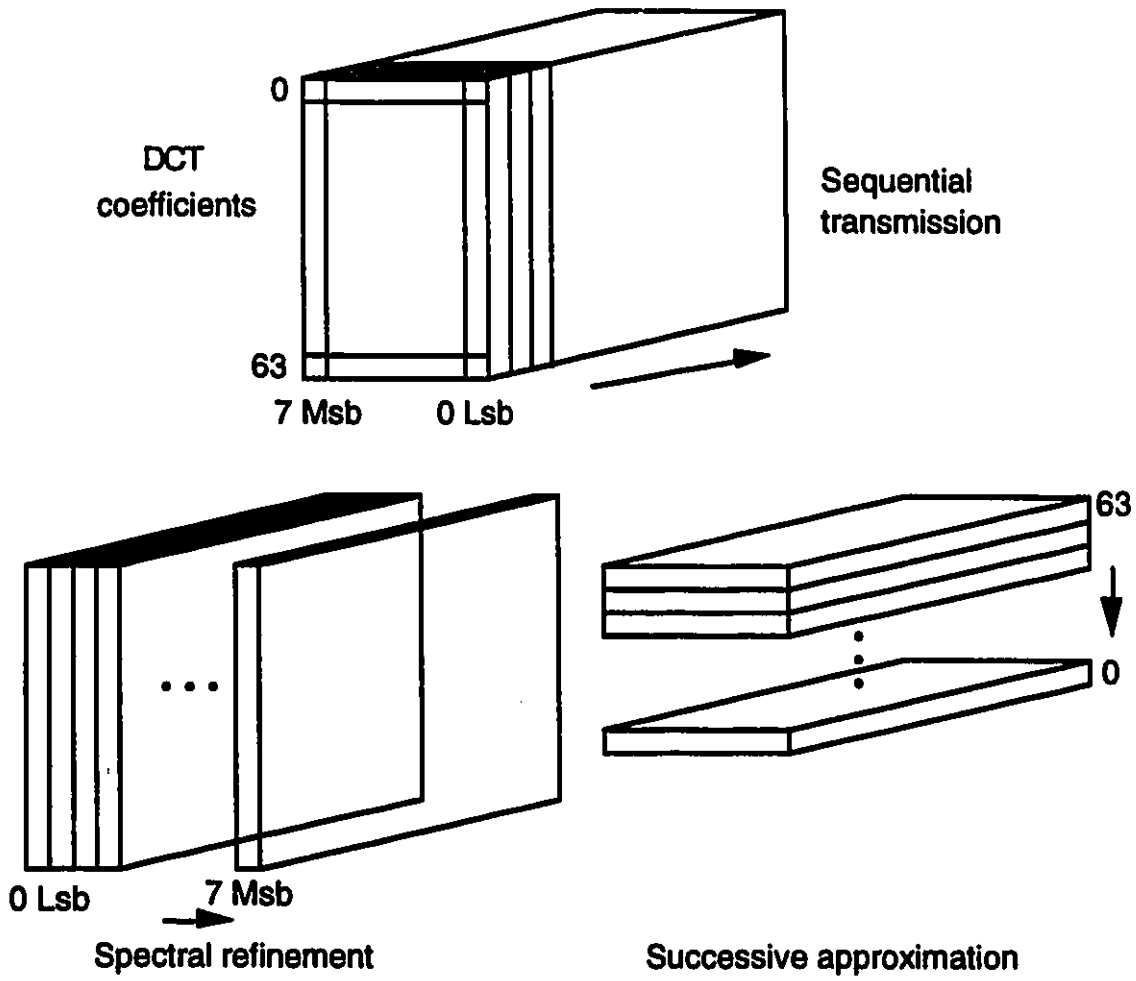


Fig. 2.19 JPEG Progressive modes.

We then reviewed lossy techniques where ordering is performed on a compressed image using either multistage or embedded quantization. Hierarchical decomposition based on wavelets delivers the best performance. In particular, block quantization by DCT does not perform well in the early stages due to blocking impairment. Vector quantization can deliver high performance but requires more computation.

Finally, we described the implementation of the JPEG standard in its sequential and progressive modes.

Chapter 3

Mathematical Morphology

Mathematical morphology is an emerging theory in the field of image analysis. It has successfully been applied to medical imaging, machine vision, remote sensing, character recognition, and many other imaging fields. Its formalism is based on set theory.

Mathematical morphology complements the other image processing theories;

1) Classic signal processing, which includes linear signal processing, convolutions, linear transformations, and wavelets;
2) Statistical methods, which involves parameter extraction rather than image transformation;
3) Syntactic methods, including neighborhood operators [Har93], which extract primitives and use grammatical rules.

Mathematical morphology is a vast subject and is covered by numerous conferences, publications, and industrial applications. The goal of this chapter is to present its basic elements, the ones which we use in subsequent chapters. First, we review

some set operators and properties. We then present the morphological transformations pertinent to our research. Finally, we introduce the watershed transformation as a segmentation operator.

3.1 Basic Set Theory

Let $A, B, C, \ldots$ designate sets and $a, b, c, \ldots$ designate elements. $\varnothing$ designates the empty set.

3.1.1 Set Operators

Union. The union of two sets is the set formed with all the elements which belongs to one or the other, i. e. , $X \cup Y = \{p | p \in X \text{ or } p \in Y\}$. The union of two sets is commutative and associative.

Intersection. The intersection of two sets is a set formed with all the elements that belongs to both: $X \cap Y = \{p | p \in X \text{ and } p \in Y\}$. The intersection is commutative and associative.

Complementarity. Given a subset X of a set E being used as a reference, the complement of X in E, X^c is given by $X^c = \{p | p \in E \text{ and } p \notin X\}$. De Morgan Laws apply to complementarity as to logical operations: $(X \cap Y)^c = X^c \cup Y^c$ and so on...

Translation. The translation of a set A by an element x is denoted A_x, and is defined by $A_x = \{a + x | a \in A\}$

Transposition. The point symmetry of a set B around the origin, called the transposed of B, is denoted $\check{B}$, and is defined as $\check{B} = \{-b | b \in B\}$.

Minkowski Addition. Given two sets X and Y in a space E, to each $x \in X$ and $y \in Y$ corresponds the algebraic sum $x+y$. Selecting all the x and y, we can form a new set called Minkowski addition and noted $X \oplus Y$. $X \oplus Y = \bigcup_{x \in X} Y_x$. Minkowski addition is commutative.

3.1.2 Properties

Sets can have the following properties:

Set equality. Two sets are equal if they have the same elements; $X = Y \Leftrightarrow (p \in X \Rightarrow p \in Y$ and $p \in Y \Rightarrow p \in X)$. Set equality is symmetric, reflexive, and transitive.

Inclusion. X is a subset of Y, or X is included in Y, if all the elements of X are elements of Y: $X \subseteq Y \Leftrightarrow (p \in X \Rightarrow p \in Y)$. Inclusion is reflexive, antisymmetric, and transitive.

In addition, some important properties are associated with set operators. We use ψ to designate an arbitrary operator. Consequently, $\psi(A)$ is the output set resulting from applying ψ to an input set A

Operators can have the following properties:

Increasing monotonicity. An operator is monotonically increasing if it preserves the inclusion property, i. e., $A \subseteq B \Rightarrow \psi(A) \subseteq \psi(B)$.

Idempotence. An operator is idempotent if applying this operator twice is equivalent to applying it once: $\psi(\psi(A)) = \psi(A)$.

Continuity. Given two sets X and Y, an operator is continuous if, when Y approaches X, the output of Y approaches the output of X: $Y \to X \Rightarrow \psi(Y) \to \psi(X)$. Formally, continuity is met if for all sets N close to $\psi(X)$, there is a set N' close to X such that $\forall Y \subseteq N'$; $\psi(Y) \subseteq N$. Clearly, continuity is related to neighborhood and connexity, two concepts that are be defined in Section 3.1.2.

Translation Invariance. Translation invariance is an important property of linear operations. The same property can be found in set operators: $\psi(X_p) = (\psi(X))_p$. The property implies that the origin can be selected arbitrarily.

Homogeneity. An operation can include a positive scale factor λ. Let λX be a set in $\Re^n$ and ψ_λ an operator with parameter λ. The operation is homogeneous if applying ψ_λ to X dilated by λ is the same as dilating the result of the transformation, i.e., $\psi_\lambda(\lambda X) = \lambda\psi(X)$ where $\psi = \psi_1$.

Extensivity. An operator is extensive if $\psi(A)$ always contains A: $A \subseteq \psi(A)$, $\forall A$, and it is said to be antiextensive if $\psi(A)$ is always a subset of A: $\psi(A) \subseteq A$, $\forall A$.

Duality. An operator is dual to another if applying this operator to a set is equivalent to applying its dual operator to the complemented set and complementing the output: if ψ^* denotes the dual operator of ψ, $\psi^*(A) = \psi(A^c)^c$.

3.1.3 Continuous Space and Discrete Space

We consider two types of sets or images: continuous and discrete. A continuous binary image is a subset of the 2-dimensional continuous space $\Re^2$. When introducing the basic operations and definitions, or when discussing fundamental properties, we refer to the continuous space. For digital applications, however, we consider an image to be a subset of the 2-dimensional Cartesian grid Z^2.

In a discrete grid, the 4-neighborhood of a point (i,j) is the set $N_4(i,j)$:

$$N_4(i,) = \{(i-1,j),(i,j-1),(i+1,j),(i,j+1)\}. \tag{3.1}$$

The 8-neighborhood is defined as:

$$N_8(i,j) = \{(i-1,j-1),(i,j-1),(i+1,j-1),(i-1,j),(i+1,j)(i-1,j+1),(i,j+1),(i+1,j+1)\} \tag{3.2}$$

In the same way, the contour of a set X is given by:

$$C_4(i,j) = \{(i,j) \in Z^2 | (i,j) \in X \text{ and } N_4(i,j) \cap X^c \neq \varnothing\} \tag{3.3}$$

and

$$C_8(i,j) = \{(i,j) \in Z^2 | (i,j) \in X \text{ and } N_8(i,j) \cap X^c \neq \varnothing\}. \tag{3.4}$$

An ambiguity arises since the complement of a 4-connected set is an 8-connected set. This problem motivates most morphologists into using an hexagonal grid which eliminate the confusion. In this thesis, however, we use solely square grids.

3.2 Basic Morphological Transformations

We first define the morphological transformations on sets or binary images. We then expand the definitions to functions or gray-scale images.

3.2.1 Structuring Element

All morphological operations are based on a fitting concept. An image is probed with a *structuring element* and a measure is taken to quantify the manner in which the structuring element fits within the image. This is seen as the result of set operations involving the image and the structuring element, usually denoted B. The result is dependent on both the size and the shape of the structuring element. The size plays a more important role than does the shape, and consequently, the structuring element is usually a small simple geometric figure such as a disc or a square .

3.2.2 Erosion

We can define the erosion of a set X by a structuring element B, as the locus of points p such that B is included in X when its origin is placed at p. Alternatively, the erosion is the set found by intersecting all translates of the input images by negatives of points in the structuring element:

$$\varepsilon_B(X) = \bigcap_{b \in B} X_{-b} = \left\{ p \in \Re^2 | B_p \subseteq X \right\}. \qquad (3.5)$$

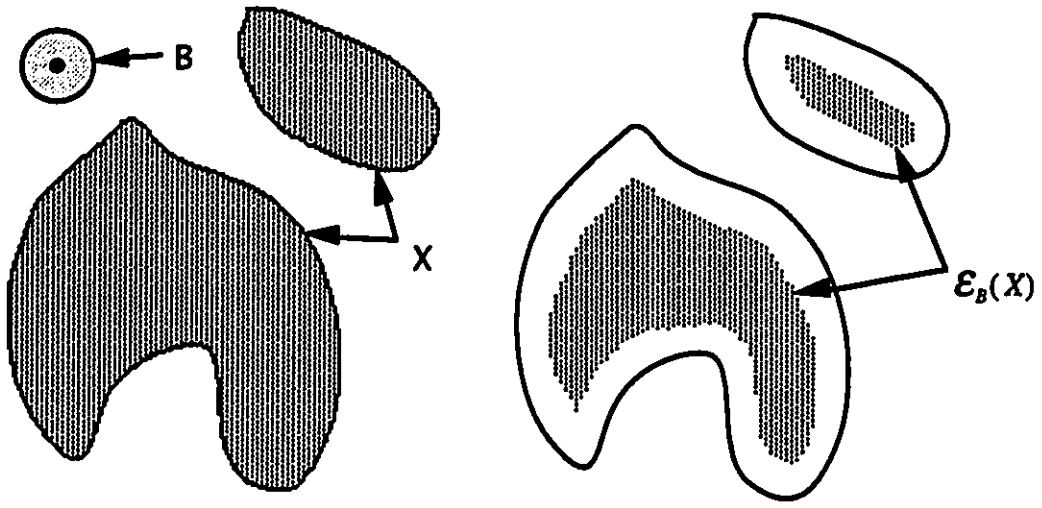


Fig. 3.1 Erosion of a set X by a disk B.

Note that $\mathcal{E}_B(X)=\varnothing$ if B is larger than X, because it is impossible to have B completely inside X. Fig. 3.1 shows the erosion of a set X by a disk B.

3.2.3 Dilation

The dilation of a set X by a structuring element B is denoted $\delta_B(X)$ and is defined as the locus points p such that B hits X when its origin coincides with p:

$$\delta_B(X)=\left\{p\middle|(B_p)\cap X\neq\varnothing\right\}. \tag{3.6}$$

Dilation can equivalently be defined as the union of all the translation of X by the vector $-b$ of B:

$$\delta_B(X)=\bigcup_{b\in B}X_{-b}. \tag{3.7}$$

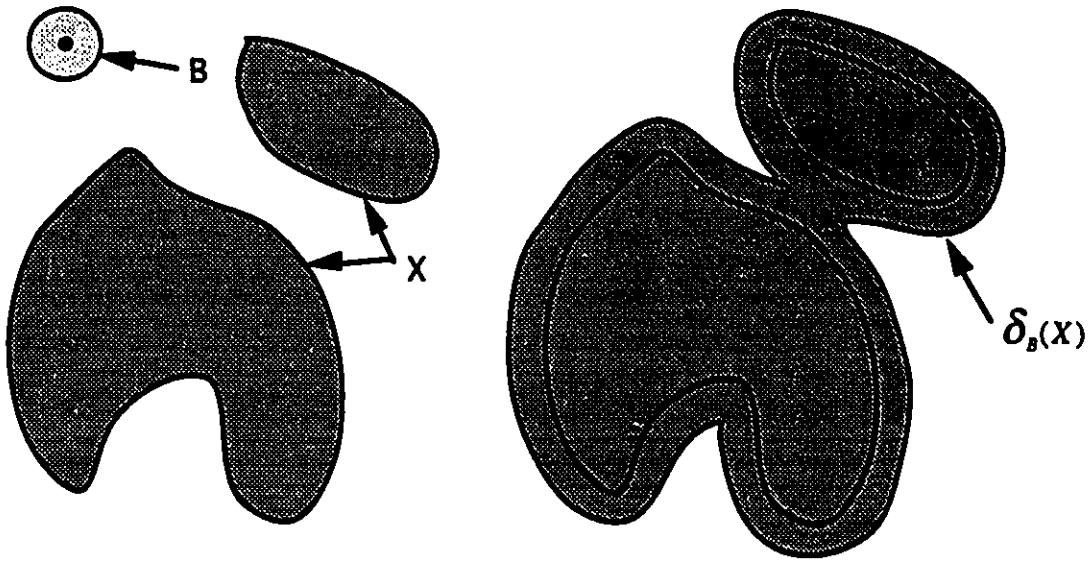


Fig. 3.2 Dilation of a set X by a disk B.

Fig. 3.2 shows the set X of Fig 3.1 dilated by the same disk B.

3.2.4 Properties of Erosion and Dilation

The erosion and dilation operations have the following properties:

- Erosion and dilation are translation invariant: $\mathcal{E}_B(X_p) = (\mathcal{E}_B(X))_p$ and, $\delta_B(X_p) = (\delta_B(X))_p$;
- Erosion and dilation are scale invariant: $\mathcal{E}_{\lambda \cdot B}(\lambda \cdot X) = \lambda \mathcal{E}_B(X)$ and, $\delta_{\lambda \cdot B}(\lambda \cdot X) = \lambda \delta_B(X)$;
- Erosion and dilation are semi-continuous. We refer to Serra [Ser82] for a rigorous demonstration. In practice, the use of a discrete space Z^2 removes most questions of continuity.

- They are increasing operators: if $X \subseteq Y$, then $\varepsilon_B(X) \subseteq \varepsilon_B(Y)$ and $\delta_B(X) \subseteq \delta_B(Y)$;
- If the origin is contained in the structuring element, the erosion is anti-extensive and the dilation is extensive, i.e., $\varepsilon_B(X) \subseteq X$ and $X \subseteq \delta_B(X)$;
- Erosion and dilation are dual operations: $\delta_B(X) = (\varepsilon_B(X^c))^c$ or $\delta_B(X) = (\varepsilon_B(X^c))^c$ and $\varepsilon_B(X) = (\delta_B(X^c))^c$ or $\varepsilon_B(X) = (\delta_B(X^c))^c$.

In addition to these properties, erosion and dilation have numerous algebraic properties. We only mention here a few properties which play a key role in reducing calculation times. An exhaustive listing of algebraic properties can be found in Haralick [Har93].

Dilation is commutative and associative.

$$\delta_B(X) = \delta_X(B) \tag{3.8}$$

$$\delta_C(\delta_B(X)) = \delta_{\delta_C(B)}(X) \tag{3.9}$$

In practice B is much smaller than X, since B is the structuring element and X is often the whole image. If we want to calculate $\delta_D(X)$ where $D = \delta_C(B)$ and B and C are squares of size N, the brutal application of two successive dilations requires N^2 operations, while the succession of $\delta_C(\delta_B(X))$ requires only 2N operations.

A similar property is used to speed up dilations by large structuring elements, namely, the chain rule:

$$\varepsilon_{\delta_C(B)}(X) = \varepsilon_C(\varepsilon_B(X)). \tag{3.10}$$

3.2.5 Morphological Opening

The morphological opening $\gamma_B(X)$ is obtained by cascading an erosion and a dilation with the structuring element B:

$$\gamma_B(X) = \delta_B(\varepsilon_B(X)). \tag{3.11}$$

Other types of opening exist in mathematical morphology. We do not consider them in this thesis. The filtering role of opening becomes more apparent when we use a different formulation:

$$\gamma_B(X) = \bigcup_{B_p \subseteq X} B_p. \tag{3.12}$$

The opening is the union of all translations of the structuring element which fits inside the input image. It results in the rejection of small variations on the contour of a binary object. This filtering effect can be seen in Fig. 3.3.

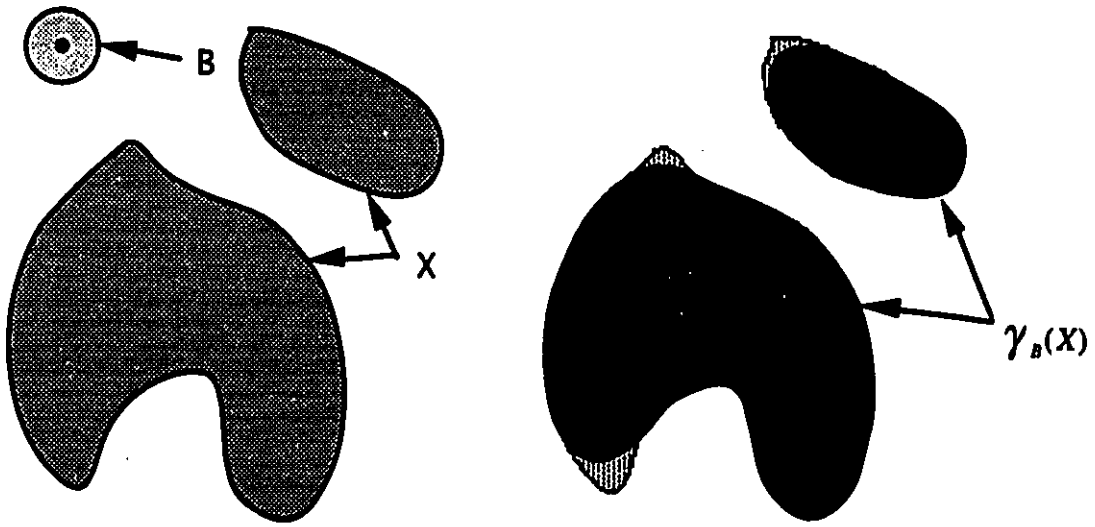


Fig. 3.3 Opening of a set X by a disk B.

We can see that the shape of the structuring element has a large effect on the result of the opening. Opening by a disc yields a "lowpass" effect.

3.2.6 Morphological Closing

By interverting the cascade of operations used to define the morphological opening, we obtain a new operatior called the morphological closing $\varphi_B(X)$:

$$\varphi_B(X) = \varepsilon_B(\delta_B(X)). \tag{3.13}$$

Morphological closing plays the same role in filtering as the does the morphological opening. Fig 3.4 shows that a closing is equivalent to an opening performed on the complement of the input.

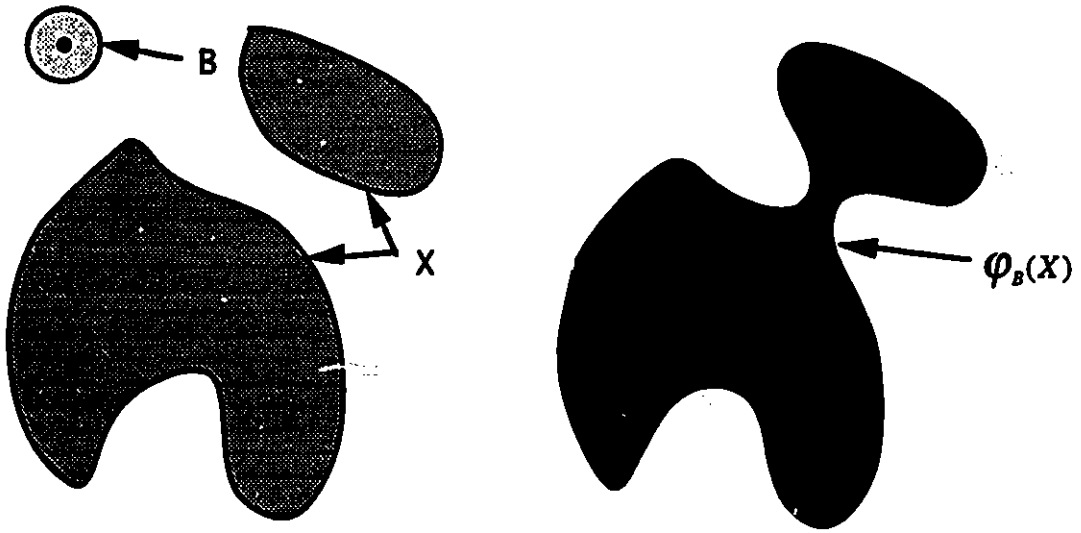


Fig. 3.4 Closing of a set X by a disk B.

3.2.7 Properties of closing and opening

Opening and closing retain all the properties of erosion and dilation:

- Opening and closing are homogeneous: $\gamma_{\lambda \cdot B}(\lambda \cdot X) = \lambda \gamma_B(X)$ and, $\varphi_{\lambda \cdot B}(\lambda \cdot X) = \lambda \varphi_B(X)$;
- Opening and closing are translation invariant: $\gamma_B(X_p) = \left(\gamma_B(X)\right)_p$ and, $\varphi_B(X_p) = \left(\varphi_B(X)\right)_p$;
- They are also semi-continuous;
- Opening and closing are dual operators: $\left[\gamma_B(X)\right]^c = \varphi_B(X^c)$ and $\left[\varphi_B(X)\right]^c = \gamma_B(X^c)$;
- Opening and closing are increasing: if $X \subseteq Y$, then $\varphi_B(X) \subseteq \varphi_B(Y)$ and $\gamma_B(X) \subseteq \gamma_B(Y)\ \varphi_B(X) \subseteq \varphi_B(Y)$.

They also have the following additional properties:

- Opening and closing are idempotent: $\gamma(\gamma(A)) = \gamma(A)$ and $\varphi(\varphi(A)) = \varphi(A)$;
- Opening is anti-extensive and closing is extensive: $\gamma_B(X) \subseteq X$ and $X \subseteq \varphi_B(X)$.

3.3 Geodesy and Reconstruction

To reconstitute a set Y from a subset X of its elements, we use a neighbor to neighbor propagation. This process, called reconstruction, belongs to the geodesic operator

family. Geodesic reconstruction plays an important part in our progressive transmission algorithm.

3.3.1 Geodesic dilation

Geodesic dilation involves two sets: the marker set X and the geodesic mask Y. The elementary geodesic dilation of X with reference to Y, $D_Y^{(1)}(X)$, is defined as the intersection between the dilation with a disc of radius 1 and Y:

$$\forall X \subseteq Y,\ D_Y^{(1)}(X) = (\delta_B(X) \cap Y), \tag{3.14}$$

where B is either in 4 or 8-connexity.

The geodesic dilation of size n of set X with reference to set Y, noted $D_Y^{(n)}(X)$, is defined as a succession of elementary geodesic dilations:

$$\forall X \subseteq Y,\ D_Y^{(n)}(X) = \underbrace{D_Y^{(1)}(D_Y^{(1)}(\ldots D_Y^{(1)}(X)))}_{n \text{ times}}. \tag{3.15}$$

Fig. 3.5 shows an example.

3.3.2 Morphological Reconstruction

Geodesic dilation, as such, is seldom used, but it is the building block of the morphological reconstruction operator.

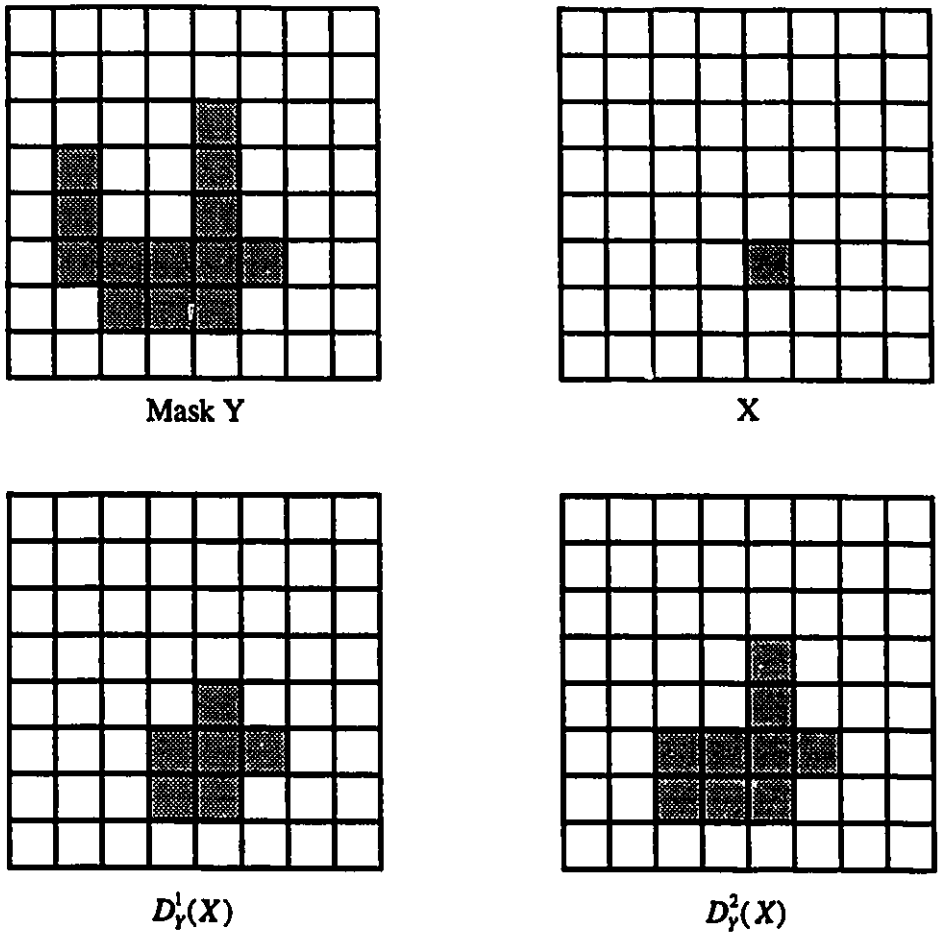


Fig. 3.5 An example of a geodesic dilation, B is in 8-connexity.

The reconstruction of set Y with markers set X, $R_Y(X)$, is the geodesic dilation of X until idempotence:

$$R_Y(X) = D_Y^{(i)}(X), \text{ where } D_Y^{(i+1)}(X) = D_Y^{(i)}, \tag{3.16}$$

or

$$R_Y(X) = D_Y^{\infty}(X). \tag{3.17}$$

As its name indicates, reconstruction can be used to reconstruct parts of a set that have been lost by other operations. Fig. 3.6 shows the three stages of a geodesic reconstruction.

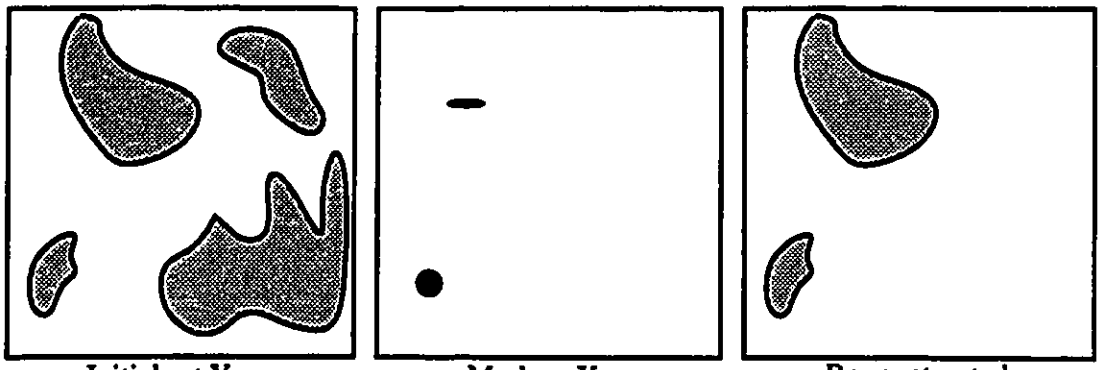


Fig. 3.6 Stages of geodesic reconstruction.

3.4 Gray-level morphology

All the morphological operators can be extended to gray-level images.

In order to accomplish this, we first generalize the basic set operators to functions. Then, we define the erosion and dilation on functions.

To facilitate the illustration and alleviate the notation we first consider 1-dimensional functions before generalizing to 2-dimensional space.

3.4.1 Mathematical Preliminaries

For gray-scale morphology, the set operations of union and intersection are replaced by supremum and infimum. In digital space, which is the space of interest in this thesis, the supremum and infimum are equivalent to maximum and minimum:

$$(F \wedge G)(x) = \max\{F(x), G(x)\}, \tag{3.18}$$

$$(F \vee G) = \min\{F(x), G(x)\}. \tag{3.19}$$

In the binary setting, the notion of subset provides an order relationship between images. An order relationship also exists for gray-scale images [Dou92]. We say that if $G(x)$ and $F(x)$ are signals with domain $D[G]$ and $D[F]$, respectively, G is *beneath* F, noted $G<<F$, if the domain of G is a subset of the domain of F and $G(x)$ is smaller or equal to $F(x)$ over its entire domain:

$$G(x) << F(x) \leftrightarrow D[G] \supseteq D[F] \text{ and } G(x) \leq F(x)\ \forall x \in D[G]. \tag{3.20}$$

Furthermore, we need to develop the notion of morphological translation. We call translation by x $F_x(z) = F(z-x)$, and offset by y $(F+y)(z) = F(z)+y$. Therefore a morphological translation includes both translation and offsetting

$$(F_x + y) = F(z-x)+y. \tag{3.21}$$

Finally, because of their key role in gray-scale morphology, we introduce a notation for flat structuring elements. By flat element we mean one that is constant over its domain. We can go further, and assume that the structuring element is zero over its

domain. If it not the case, we can simply offset the result, since all morphological operations are translation invariant. Hence, a flat structuring element can be described totally by its domain:

$$B(x)=0 \quad \forall x \in D[B(x)]. \tag{3.22}$$

3.4.2 Gray-Scale Erosion

The erosion of a signal by $F(x)$ by a structuring element $G(x)$ is defined by

$$\mathcal{E}_G[F(x)] = \min\left[F(z) - G_x(z) \middle| z \in D[G_x]\right]. \tag{3.23}$$

The erosion of an image $U(x,y)$ by a structuring element $B(x,y)$ is defined by

$$\mathcal{E}_B[U(x,y)] = \min\left[U(x,y) - B_{(z,w)}(x,y) \middle| (z,w) \in D\left[B_{(x,y)}\right]\right]. \tag{3.24}$$

In the case of flat structuring elements, which are usually favored for practical and physical reasons, the erosion of an image can be simplified to

$$\mathcal{E}_B[U(x,y)] = \min\left[U(z,w) \middle| (z,w) \in B_{(x,y)}\right], \tag{3.25}$$

Where B is the domain of the flat structuring element.

3.4.3 Gray-scale Dilation

The dilation of a signal $F(x)$ by a structuring element $G(x)$ is defined by

$$\delta_G[F(x)] = \max\left[F_z(x) + G(z) \middle| z \in D[G_x]\right]. \tag{3.26}$$

The dilation of an image $U(x,y)$ by a structuring element $B(x,y)$ is defined by

$$\delta_B[U(x,y)] = \max\left[U_{z,w}(x,y) + B(z,w) \middle| (z,w) \in D\left[B_{(x,y)}\right]\right]. \tag{3.27}$$

And can be simplified to:

$$\delta_B[U(x,y)] = \max\left[U(z,w) \middle| (z,w) \in B_{(x,y)}\right], \tag{3.28}$$

with a flat structuring element.

Fig. 3.7 shows an example of both a dilation and an erosion on a signal.

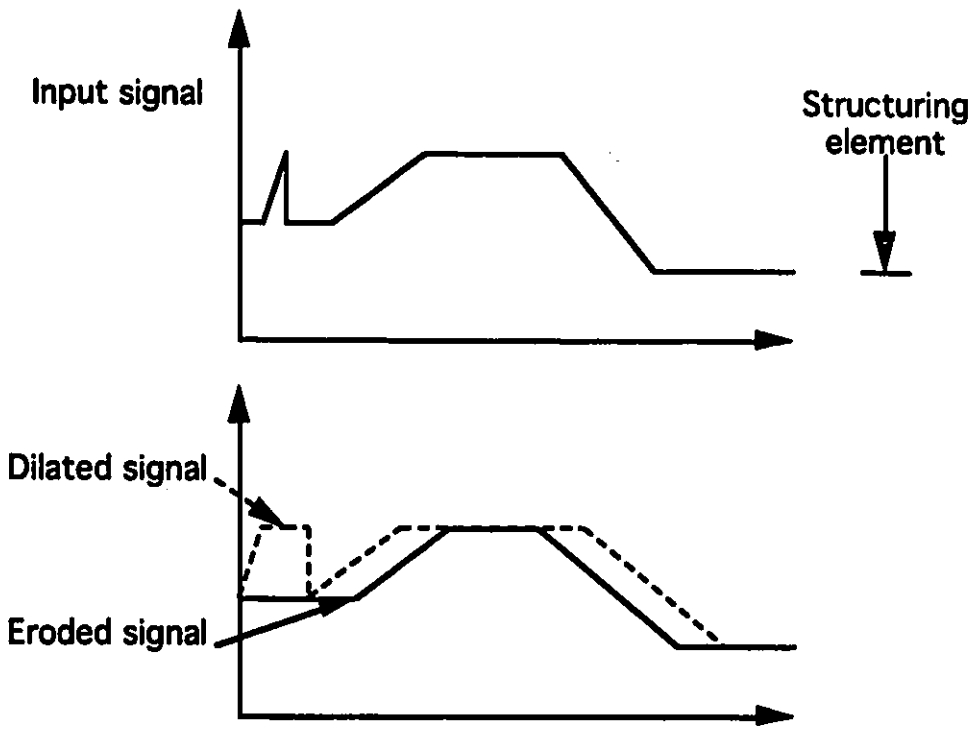


Fig. 3.7 Erosion and dilation on a gray-tone function.

3.5 Watershed Transformation and Segmentation

There are usually two approaches to the problem of segmentation of multilevel images into connected regions: region growing and contour detection [Jai89]. Segmentation using watersheds combines elements from both methods.

3.5.1 Watershed Analogy

A classical approach to the production of an edge image is to apply a threshold on a gradient image to produce a binary edge map. The problem with this method is the selection of a threshold value; too low and we obtain thick edges, too high and we fail to detect all the edges and those that are detected might be severely broken.

The gradient, viewed as a topographical surface, presents a number of basins separated by ridges. These ridges can be associated with ideal contours [Beu90].

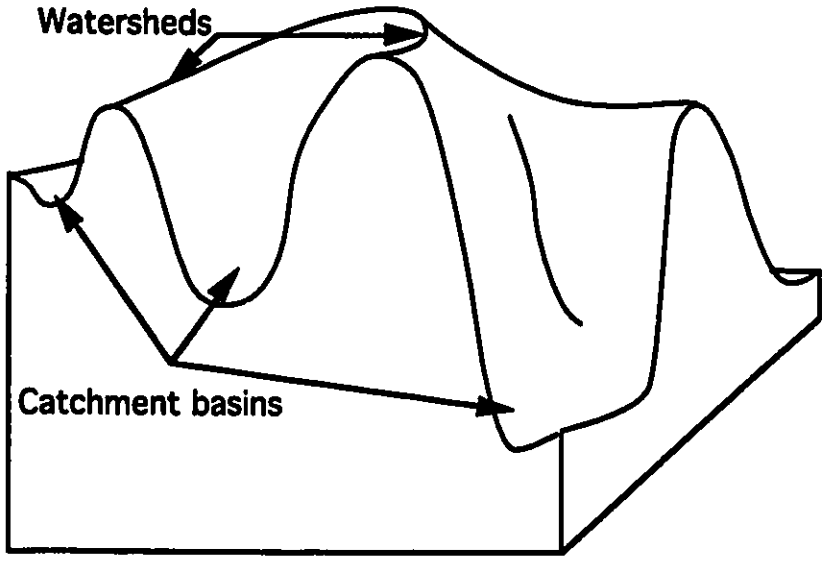


Fig. 3.8 Watersheds and catchment basins.

A drop a water falling in any point of the gradient would descend to a regional minimum. The set of all the points where the water converges into the same minimum forms a catchment basin. Two neighboring basins share a common border, or divide line, where the drop of water is equally likely to end up in either basin, as can be seen in Fig. 3.8. The union of all these borders is the watershed.

3.5.2 Implementation by Flooding

Rather than viewing water drops falling, we regard the procedure in terms of flooding. Holes are punched in the regional minima and the image is flooded from below by letting the water rise at a uniform rate across the entire image. When the rising water in distinct basins would merge, we erect dams to prevent the merging. At the end, these dams form the watershed. This process is illustrated in Fig. 3.9. We refer to Beucher for the mathematical developments [Beu90], and to Soille and Vincent [Soi90] for implementation details.

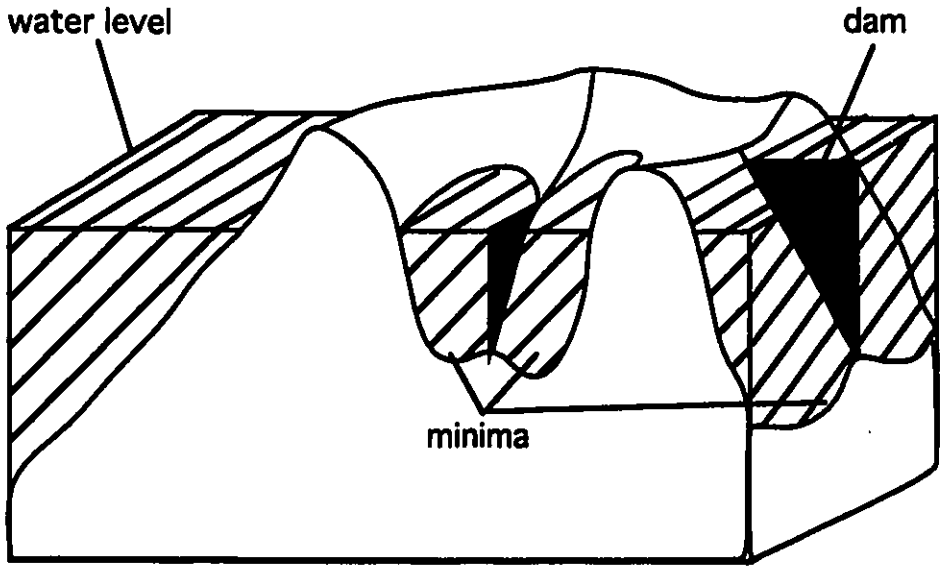


Fig. 3.9 Flooding simulation.

3.5.3 Segmentation with Markers

The watershed produces divide lines enclosing each minimum. Because the gradient image is highly noise sensitive, it contains a large number of minima. This leads invariably to over-segmentation since each regional minimum results in a different basin. To reduce the number of basins we need an additional source of information; the markers. We use markers, instead of the minima, to identify the points where the flooding originates. Alternatively, we can label the minima and only build dams to prevent the merging of flooding from differently labeled minima. Each marker roughly identifies one region of interest in the image. In general, finding markers is easier than finding the borders. Practical conditions dictate the best approach: multiresolution filtering, dome extraction, top-hat transformation, or even manual identification. Fig 3.10 illustrates the effects of markers.

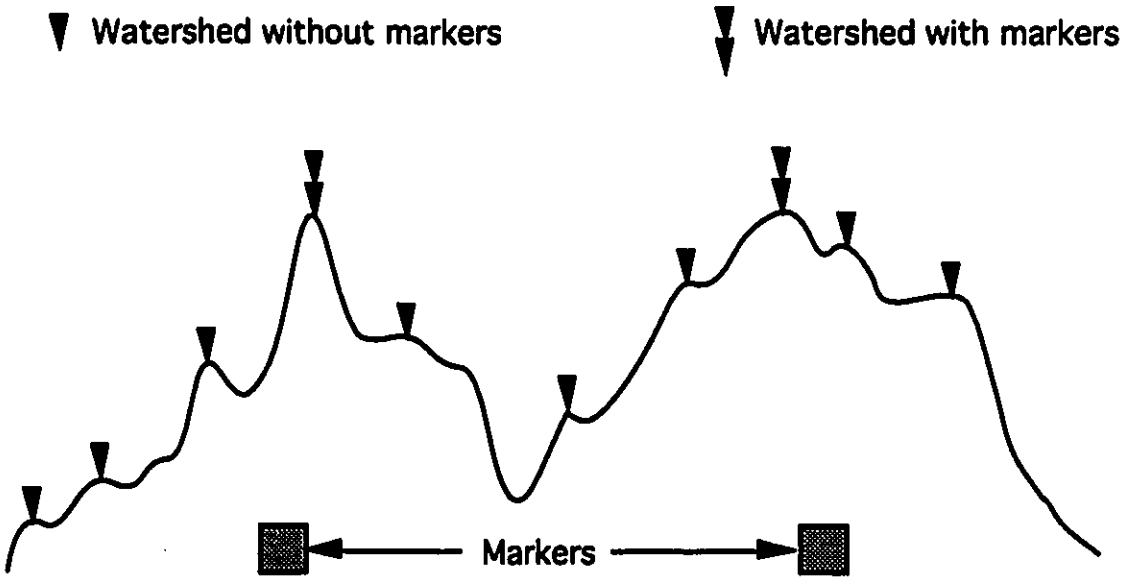


Fig. 3.10 Watershed with markers.

3.6 Summary

In this chapter we have presented the basic tools of mathematical morphology. We have defined erosion and dilation for binary and gray-tone images. We have showed that the elementary operations of erosion and dilation can be combined into closing and opening, which are more useful in filtering. We have also introduced the geodesic operations which play an important role in our decoding process. Finally, we have completed this chapter with the watershed transformation, a powerful segmentation operator which is the basis of our progressive image transmission scheme.

Chapter 4

Distortion Measures and the Human Visual System

In the initial image build up stages, PIT is very lossy. The performance of a lossy image coding algorithm must be assessed with reference to the quality of the images it produces. Furthermore, the rating given to an image must reflect the quality perceived by a human observer.

In this chapter, we review both the distortion measures used in image processing, and simple models of the *human visual system* (HVS). This material motivates the way in which we analyze the performance of our coding scheme.

4.1 The Human Visual System

We need a model of the human visual system to develop optimized compression and evaluation algorithms. There are two different ways of modeling the visual perception processes; from its performance, or from its inner workings.

4.1.1 The Psychophysical Model of Monochrome Vision

The easiest way to derive a model of the HVS, is to treat it like a black box and measure its characteristics from a limited number of psychophysical tests. What we get from this approach is a set of input-output relationships that can be easily incorporated in compression and assessment algorithms. The two most important phenomena we observe in this manner are Weber's Law and lateral inhibition. More detailed descriptions of the HVS have been made by Hall and Hall [Hal77], Sakrison [Sak77], and Kunt [Kun85].

Weber's Law. If the *intensity* or *luminance* of an object is clearly independent of its surrounding, the perceived luminance, called *brightness*, depends on the luminance of the background, i.e., two objects with different surroundings could have identical luminance but different brightness [Jai89].
Weber's law stipulates that if the luminance f_o of an object is just noticeably different from the luminance f_s of its background then

$$\frac{|f_s - f_o|}{f_o} = \text{constant}. \tag{4.1}$$

writing $f_o = f, f_s = f + \Delta f$ where Δf is small, we can rewrite (4.1) as:

$$\frac{\Delta f}{f} \approx (\log f) = \Delta c, \tag{4.2}$$

and contrast is defined as :

$$c = 50 \log_{10} f \quad 1 \le f \le 100. \tag{4.3}$$

Lateral inhibition. The spatial interaction of luminance from an object and its background is the cause of a phenomenon called *Mach bands*. Brightness is not a monotonic function of luminance. An object with constant luminance will appear non-uniform if enclosed by a brighter object on one side and a darker one on the opposing side. This can be experienced by looking at Fig. 4.1.

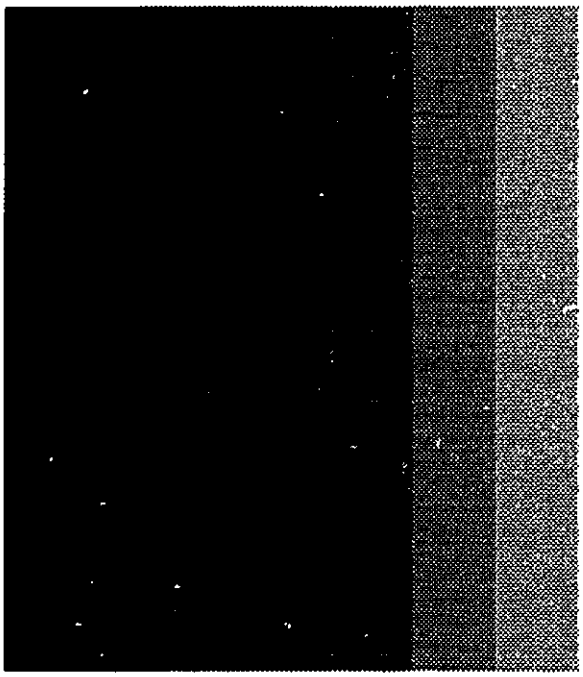


Fig. 4.1 Mach bands.

This effect shows the nature of the impulse response of the HVS, as shown in Fig. 4.2. The negative lobes are the result of lateral inhibition. The impulse response represents the spatial weighting of the contrast by the receptor of the eye. The negative lobes show that the neural signal produced by the receptors at one location is inhibited by the adjacent receptors. This process is commonly used in the construction of artificial neural networks [Zor94]. The overall effect is to enhance discontinuities in images. A well-known side effect, is that the eye is less sensible to contrast variation near an edge, [Net77]. From this, we can derive a *masking function* to

indicate where a bounded distortion would not be visible, and a *visibility function* to weigh the subjective visibility of random noise.

MTF. The same phenomenon can be represented in the spatial frequency domain. This can be done simply by considering a sinusoidal grating of varying contrast, and noting the threshold of visibility at various frequencies. The curve obtained is called the *Modulation Transfer Function* (MTF), and varies with the viewer and the view angle. A typical MTF is shown in Fig. 4.3.

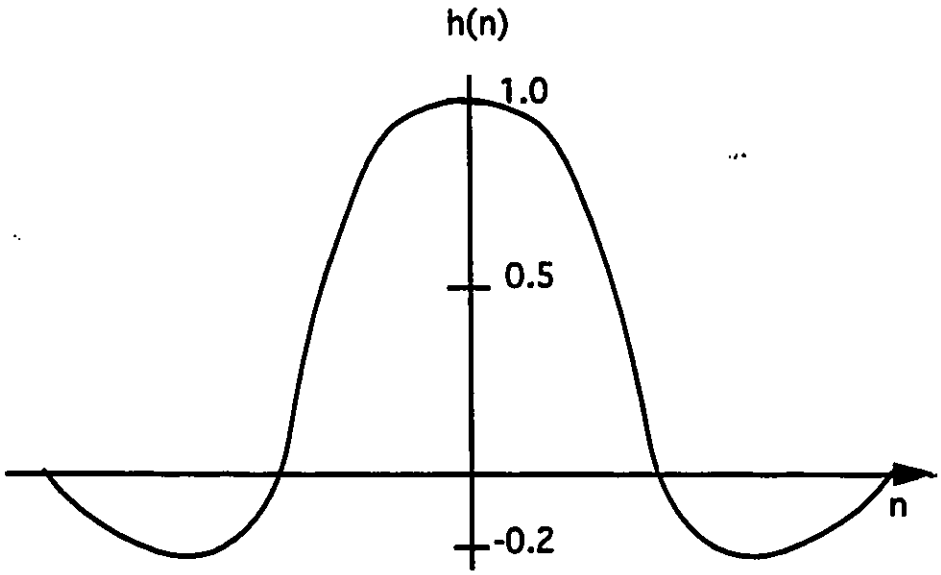


Fig. 4.2 Nature of the HVS impulse response.

The following formula, obtained with a curve-fitting procedure, describes the MTF of the HVS:

$$H_\rho(\rho) = A\left[\alpha + \left(\frac{\rho}{\rho_0}\right)\right]\exp\left[-\left(\frac{\rho}{\rho_0}\right)^\beta\right], \tag{4.4}$$

where A, α, β, and ρ_0 are constants.

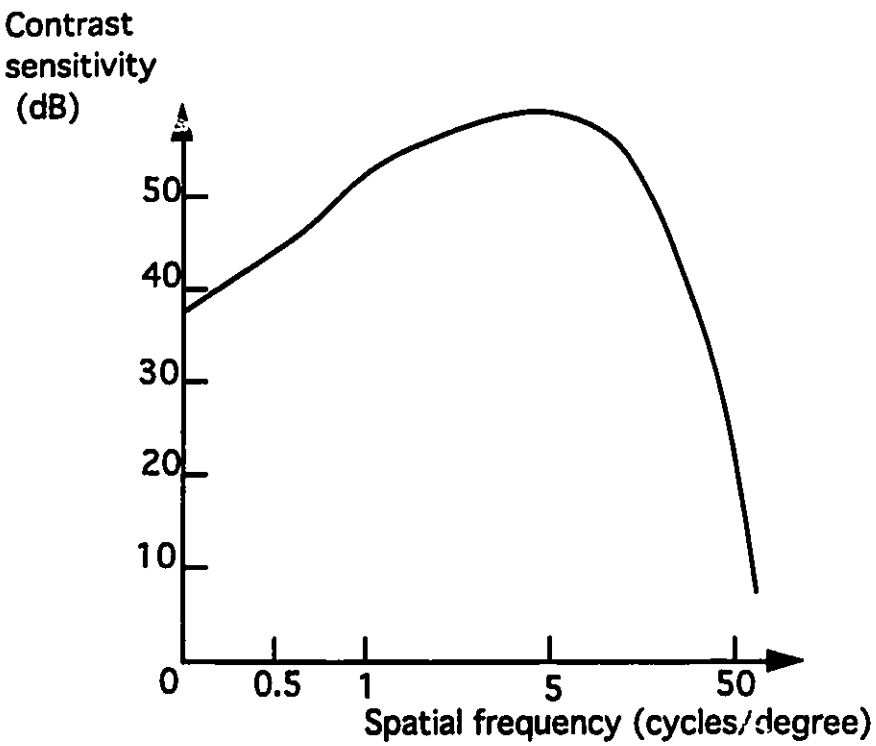


Fig. 4.3 Typical MTF function.

The joint effect of Weber's law and lateral inhibition are factored in a simplified model of the HVS presented in Fig 4.4.

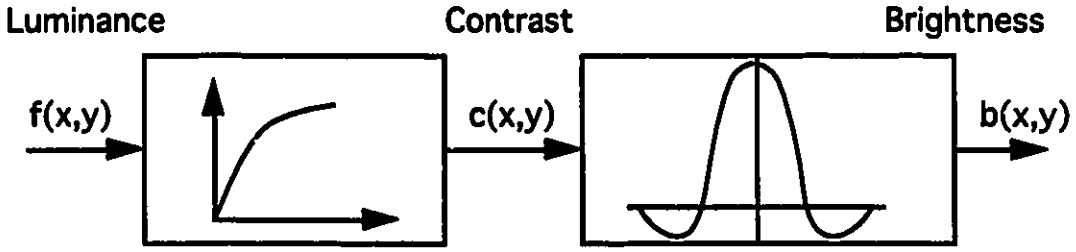


Fig. 4.4 Simplified model of the monochrome visual system.

4.1.2 The Physiological Model of Monochrome Vision

The other alternative in developing a model of the HVS is based on physiological analysis. Such a model is usually more elaborate and less tractable. It can produce more accurate prediction than the physiological model, whose results can usually only be applied under the same conditions as those under which the testing to create it was conducted. The physiological model, however, requires the knowledge of the viewing conditions: viewing angle, illumination, background luminance, focus point, and so on.... Fig. 4.5 shows a model that was proposed by Tzou and *al.* [Tzo84]. We note that it is very similar to the simplified model of Fig. 4.4. They are, after all, two models of the same system. The physiological model equations are nonetheless more complex.

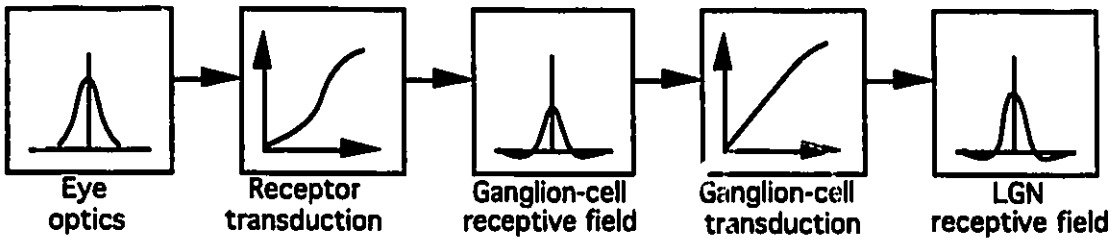


Fig. 4.5 Physiologically based human visual system model.

We note that this model stops before the cortex and the higher level visual processing. The cortex contains hierarchical layers of specialized cells, which are sensitive to discontinuity in a particular position or orientation. In essence, they act like the masks used by local edge operators [Kun85].

4.2 Subjective Measures

There are two types of quality assessment for images: those based on subjective testing, and those based on computable measurements. We first consider the subjective measures which use experimentation with a population of human subjects.

4.2.1 Goodness Scales

A goodness scale uses global or relative appreciation based either on personal judgment or on a comparison between images within a group. An example of each is presented in Table 4.1. The *Mean Objective Score* (MOS), which ranges from 0 to 5, is the most widely used goodness scale.

Overall Goodness scale		Group goodness scale	
Excellent	(5)	Best	(7)
Good	(4)	Well above average	(6)
Fair	(3)	Slightly above average	(5)
Poor	(2)	Average	(4)
Unsatisfactory	(1)	Slightly below average	(3)
		Well below average	(2)
		Worst	(1)

Table 4.1 Image goodness scales.

4.2.2 Impairment Scales

When the degradation within the image is clearly noticeable, the values do not vary sufficiently within the goodness scale. In such cases, the use of an impairment scale is a better way of rating the images. In the impairment scale of Table 4.2, the subjects are not asked to evaluate the quality of the image, but rather to judge how annoying the imperfections are.

Not noticeable	(1)
Just noticeable	(2)
Definitely noticeable but only slight impairment	(3)
Impairment not objectionable	(4)
Somewhat objectionable	(5)
Definitively objectionable	(6)
Extremely objectionable	(7)

Table 4.2 Impairment scales.

4.2.3 Task-Oriented Measures

In some cases, we are more interested in how does the coding noise affect the usefulness of the images. A good example of a rating system based on such criteria is the intelligibility test used in telephony, where the subject is asked to recognize one of several possible consonants in a close rhyme set (such as meat, heat, seat, beat, neat) [Jay84]. For images, the rating scheme is often simply to accept or reject the image, resulting in a binary decision task.

4.3 Objective Measures

Subjective assessment tests are widely used to evaluate the picture quality of images, but they are not without drawbacks: the results fluctuate depending on the viewing conditions, and the tests they require are both lengthy and costly.
Consequently, an appropriate objective measure, if one can be found, is preferable to a subjective one.

4.3.1 Mean Square Criterion

Among the quantitative measures, the class of criteria used the most is based on the mean square error. This method is favored because of its ease of calculation and its mathematical tractability.
For $M \times N$ images $u(m,n)$ and $u'(m,n)$ the least square error is given by:

$$\sigma_{ls}^2 \equiv \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|u(m,n)-u'(m,n)\right|^2, \tag{4.5}$$

and is usually used as an estimate of the mean square error:

$$\sigma_{ms}^2 \equiv E\left[\left|u(m,n)-u'(m,n)\right|^2\right], \tag{4.6}$$

where E represents the mathematical expectation.

Finally, another alternative is to define the expectation E with respect to the visibility function.

The resulting criterion is called the mean square subjective error and is given by:

$$\sigma^2_{msse} \equiv \int_{-\infty}^{\infty} |e|^2 v(e) de, \tag{4.7}$$

where e is the value of the error $u - u'$ *and* $v(e)$ is the visibility of that error. The visibility function represents the subjective visibility of unit noise in a scene. This function varies with the scene and must be measured through subjective testing.

4.3.2 Signal-to-Noise Ratio

In many applications, the mean square error is expressed in terms of a signal-to-noise ratio (SNR), which is defined, in decibels, as:

$$SNR = 10 \log_{10} \frac{\sigma^2}{\sigma^2_{ms\ or\ ls}} \text{ dB}. \tag{4.8}$$

This formulation is motivated to a great extend by work done on speech coding [Jay84].

Another definition of *SNR* commonly used in image coding is:

$$SNR' \equiv 10 \log_{10} \frac{(\text{peak - to - peak value of reference image})^2}{\sigma^2_{ms\ or\ ls}} \text{ dB}. \tag{4.9}$$

4.3.3 Other Proposed Criteria

The mean square error criterion is not a good global measure of image quality. Nor is it a good measure at all if the error is not random noise but a more structured kind of coding artifact. Not surprisingly, many researchers have proposed other metrics for the evaluation of distortion in images.

We can distinguish two methodologies used to develop a better distortion criterion as whether they are inspired by the physiological or the psychophysical model of the HVS. The first approach takes in consideration the viewing conditions and the characteristics of the image. This quickly leads to complex equations which are often not practical, since we have to specify the lighting conditions, the viewing distance, the angle of view, and so on. Furthermore, the usefulness of these metrics is limited since they can be used to determinate the visibility of a perturbation but not its relevancy, or the annoyance it causes.

Pearlman and *al.* proposed a distortion measure of the form:

$$d(u,u') = kI_0^{-2} \exp\left[-\left(k_t - \frac{B|u|}{I_0}\right)^2\right]|u-u'|^2, \qquad (4.10)$$

Where the d is the distortion over one spatial channel, k is a constant, B is a function of the display luminance ranging from 144 to 700, I_o is the mean intensity in the gray scale unit, and k_t is the threshold signal-to-noise ratio estimated at 2.6.

Their guiding assumption was that a change in intensity can be measured by a change in its probability of detection [Pea78].

The psychophysical option consists instead in trying to reproduce subjective results. Limb was one of the first to seriously consider this option [Lim79]. He experimented with various error summing algorithms, trying to find the best match to the MOS rating obtained from various types of distortions. He failed to come up with an error measure better than MSE.

More recently, Miyahara developed a more evolved procedure to obtain a match with a MOS rating. His approach takes into account the contour of the transmitted image and the autocorrelation of the error image [Miy88].

Following a different strategy, Saghri and Cheatnam [Sag89] suggested to consider the information content of the image. They defined it as being the weighted sum of the magnitude of the DCT coefficient after they have been normalized to the HVS sensibility. This produces a set of curves, each one corresponding to a particular resolution. A measure can be made by comparing the information content of an original image with the information content of a noisy image.

4.4 Summary

In this chapter, we described simple models of the HVS. We reviewed the subjective and objective distortion measures used in image compression, and we presented a survey of objective measures derived from the HVS.

The mean square error is the most common measure, but it is increasingly used in conjunction with a weighting based on the HVS. The SNR is a popular way of expressing the mean square error. MSE is a convenient image fidelity criterion, but not a good measure for all types of distortion.

In the case of PIT, where a low quality image is transmitted rapidly, even the standard subjective measures are not appropriate. The important question is no longer how are the distortions perceived, but how much of the image is perceived. In this light, the surveyed objective measures of the last section give us a model for the design of a more suitable metric.

Chapter 5

PIT by Segmentation Coding

In this chapter, we present the details of our progressive transmission scheme. First, we introduce an image representation derived from segmentation. We show how the watershed can be used to quickly convert an image into that representation. Then, we develop an ordering procedure that allows us to transmit progressively the most important features first. Next, we explain how we encode the ordered information into a bit stream. Finally, we show how the image is decoded.

5.1 Image Representation by a Graph of Regions

At the root of the image compression problem is the need to represent the pictorial content in a form that can be efficiently encoded. We motivate our choice of using segmentation to simplify the image content and introduce a graph representation for the segmented image.

5.1.1 Motivation

Our prime objective was to develop a progressive image transmission procedure tuned to the early stages of transmission. This is motivated by browsing applications; we can distinguish two tasks in image database browsing: identification and inspection [Mil85]. Identification requires less attention, and delays at this stage are not readily tolerated by the end-users.

In PIT, however, the quality of intermediary images is less than if the particular algorithm used had been tuned to that particular bit rate. This can be attributed to the fact that the image content should be simplified before it can be lossly encoded at a lower bit rate. As a simple example of this, we know that an image must be low-pass filtered before it can be sub-sampled. We also note that the PIT methods which perform best, like the hierarchical decomposition based on wavelets, compute a simplification of the image as part of the encoding process, and go beyond the multistage or embedded quantization step.

In the same way, we expect that the selection of an image representation adapted to a particular rate would lead to better performances than the tuning of a compression method based on a more general representation would.

Our development is based on the following assumptions:

1. The information is concentrated in the high frequencies, more specifically in the objects contours.
2. Some contours can be eliminated, but none should be added since we do not want to transmit false information, or mask the information we successfully transmitted.
3. The importance of a contour is proportional to the value of the gradient associated with it.

The last assumption is not always true. Other criteria, such as size and location, certainly play a role. We will see, however, that this assumption holds for most images.

We resort to segmentation to simplify the image, since there has been much evidence that a segmented image can be highly compressed and preserve much of the contour information [Kun87]. We must point out that the result of a segmentation operation should not be seen as an approximation of the input image, but rather as a description of it, which is a rather different paradigm than the classical one.

5.1.2 Description

Our strategy is to represent an image by a set of non-overlapping regions with homogeneous luminance as shown in Fig. 5.1.

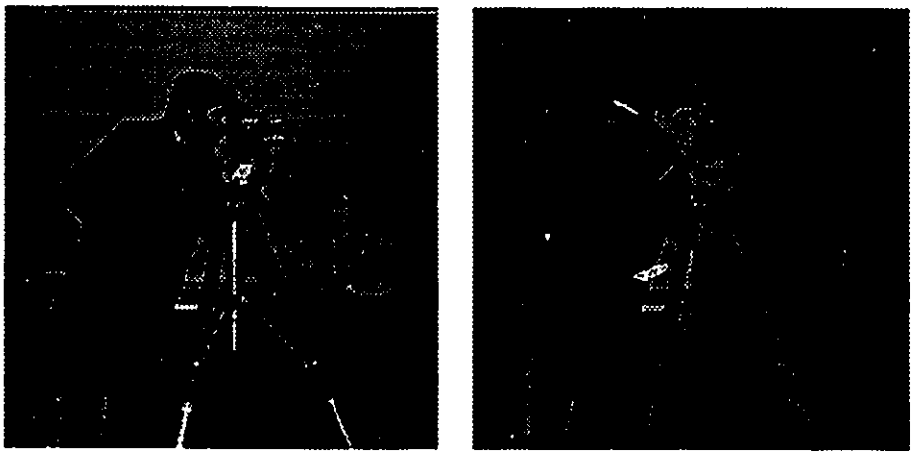


Fig. 5.1 a) Original image. b) Representation with 50 regions.

Like VQ, the representation is intrinsically lossy. Furthermore, the representation scheme is not as universal as transform coding or predictive coding; performance is

more image-specific. In particular, this representation does not work well with textured images, because there are very few homogenous regions in textured areas.

We can view the simplification as going from an image with N pixels to one with M regions: $M<N$. But, for the representation to be complete, we must be able to express the segmented image in terms of regions, not pixels.

If X is the original image and $\hat{X}$ is the segmented image divided in M regions $X_k (1 \le k \le M)$

$$\hat{X} = \bigcup_{k=1}^{M} X_k. \tag{5.1}$$

Furthermore, the following properties are observed:

- No two regions overlap:

$$\forall i, j \in \{1,\ldots,M\}: X_i \cap X_j = \emptyset. \tag{5.2}$$

- Each region form a connected component:

$$\forall x \in X_k | k \in M: R_{X_k}(x) = X_k. \tag{5.3}$$

But how can we describe the spatial organization of $\hat{X}$? The use of a graph solves the problem elegantly [Pav77]. A graph is a set assorted with a relation. The neighborhood relation that we use is the following:

$$\exists (x,y) \in X_i | N_4(x,y) \cap X_j \neq \emptyset,\ i \neq j, \tag{5.4}$$

two regions X_i and X_j, are neighbors if, and only if, there is one point in X_i whose 4-neighborhood, as defined in Eq. 3.1, includes one point of X_j.

The set S associated with the graph G is formed by taking all the nodes of the graph. Fig. 5.2 shows a simple image and its graph.

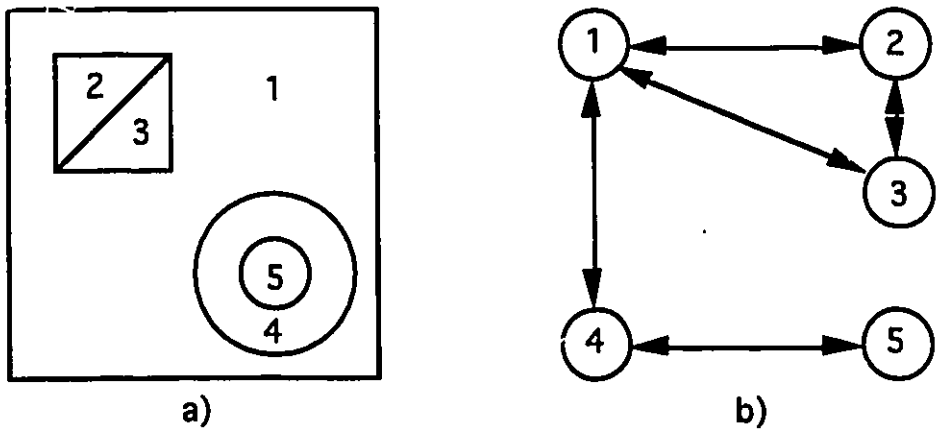


Fig. 5.2 a) A segmented image. b) Its graph.

5.2 Segmentation by Watersheds

We used the watershed operator described in section 3.5 to produce a segmented image. Since the procedure has already been covered, we need to add the following specifications.

5.2.1 The Gradient

The watershed is applied on a gradient image. We favored the morphological gradient $M_B(X)$:

$$M_B(X) = \delta_B(X) - \varepsilon_B(X), \tag{5.5}$$

where B is an unitary structuring element. We can write M_4 or M_8 to indicate that B is 4 or 8-connected, respectively.

Fig. 5.3 shows the two morphological gradients obtained on the image of Fig. 5.1.

a) b)

Fig. 5.3 a) M_4 b) M_8

The resulting gradient is thick. There are many edge detectors available, and we found that the selection of one in particular has little effect on the segmentation results.

5.2.2 Elimination of the Divide Lines

We have seen that the watershed segments an image by creating divide-lines. This is not consistent with our graph representation. Eq. 5.1 excludes such lines from the region set, since all the pixel must be part of one and only one region.

The pixels on the divide line have to be allocated to one region. In essence, there are three ways of dealing with this issue.

The first alternative is to label the pixel according to the first basin reaching it. The use of the morphological gradient, with its thick edges, helps in the process.

The second alternative is to apply a region growing criterion to assign each pixel on the divide line to the region in which it fits best. The presence of slanted borders often makes this an arduous task.

Finally, we chose instead to operate on a magnified version of the gradient image. Once this image has been segmented, the divide lines fall naturally on the pixels which are decimated when the labeled image is brought back to its original size.

5.2.3 The Markers

Another important factor in the segmentation by watershed is the selection of markers. The number of markers sets the number of regions and, therefore, the subjective quality of the segmented image. We used the regional minima of the gradient image, because we rely on the ordering stage to determine which features are important. Fig. 5.4 shows the results of the segmentation on the image of Fig. 5.1

5.3 Ordering the Regions

Our objective is to transmit first the information pertaining to the most contrasted contours. However, our representation is based on regions, not contours. We cannot transmit a contour unless it closes an existing region. Therefore, instead of trying to find the most important contours, we decided to successively eliminate the least important ones. Eliminating a contour preserves the integrity of the representation since it corresponds to the fusion of two nodes on the graph.

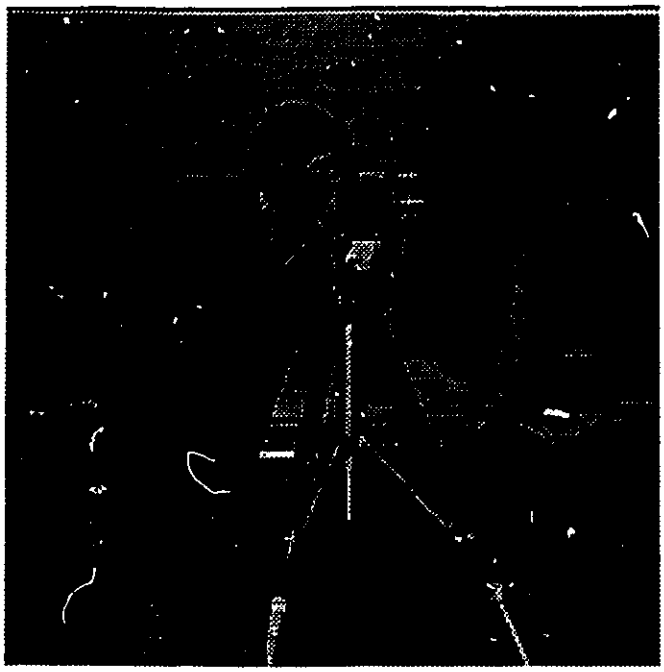


Fig. 5.4 A segmented image.

We implemented this simplification by merging the two regions with the least difference in their intensity values. This procedure is iterated until only one region occupying all the image frame remains.

Progressive image transmission can then be achieved by transmitting each region in the reverse order in which they were removed from the graph. Fig 5.5 illustrates the principle.

5.4 Describing the Regions

Having ordered the gradual decomposition of the image into an increasing number of regions, we must find an efficient way to encode that information. Each region can be described by its content, its shape, and its position.

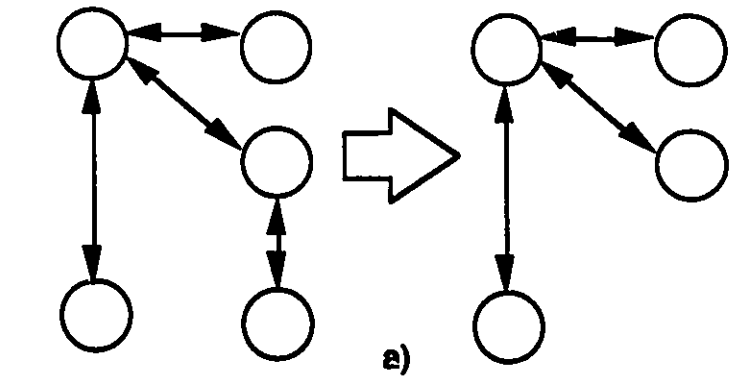


Fig. 5.5 a) Gradual simplification. b) Progressive transmission.

The shape of a region can be described by either the interior of the region or its contour. The first approach led to a representation of a region by an alphabet of overlapping elements of various shapes to cover the inside of the region [Dro94] as can be seen in Fig. 5.6.

Another interesting case is the representation by skeletons: a shape can be described by the center of the largest circles it can enclose and the size of these circles, as shown on Fig. 5.7. If we conserve only enough points on the skeleton to reconstruct the shape of the region, we obtain a variant of the previous method.

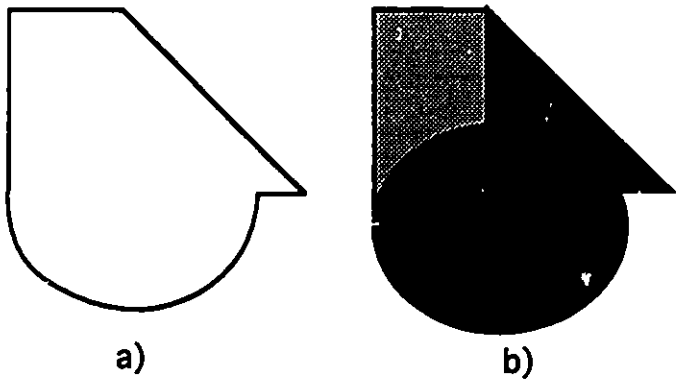


Fig. 5.6 a) A region. b) Its description by overlapping elements.

Fig. 5.7 A rectangular region and its skeleton.

Interesting results have been reported [Mar86], but they are applicable only to the description of regular shapes or filtered contours. By contrast, in our approach, we want to describe the region shape exactly, since the recognition of a shape might play an important role in the identification of the image content. Furthermore, when we split a region, there is some redundancy involved in describing even one region completely, since part of the shape was common with the known splitted region. A description of the contours seems therefore more appropriate.

We used chain coding, a method developed by Freeman [Fre61], to describe the contours. The principle is illustrated in Fig. 5.8. Some chain coding variations have been proposed to reduce the bit rate. In particular, Minami used line segments to approximate the contours [Min86], and PeNatale and *al.* limit the number of possible directions to 3 [DeN91]. Those variations involve lossy coding and cannot be used in our scheme.

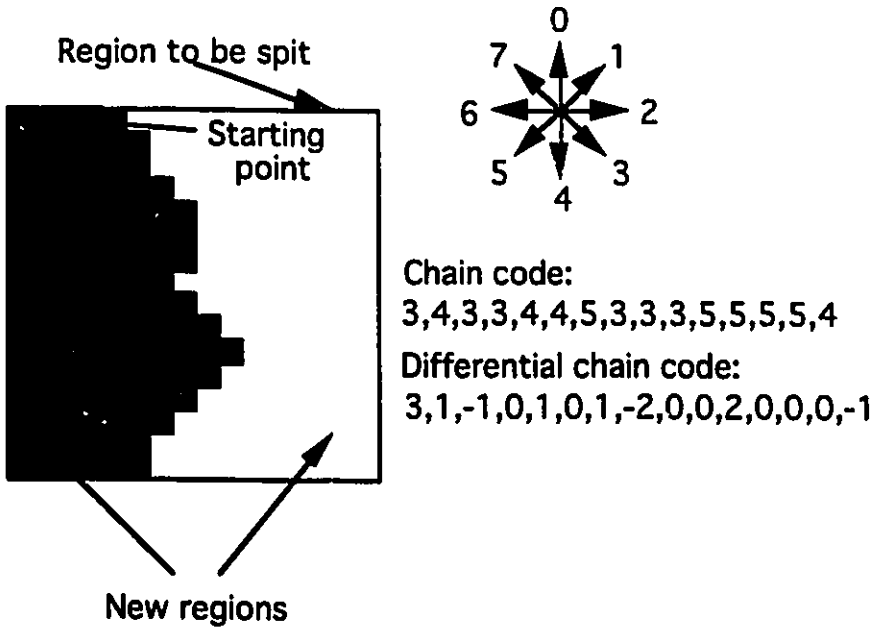


Fig. 5.8 Freeman chain codes.

5.5 Modelisation and Entropy Coding

A region can be fully described by its location, its shape, and its intensity. We encode each of these three components in a sequence of three codewords: SP, the starting point; CC, the chain code; DV, the intensity encoded as a difference value. We now detail the content of each codewords.

5.5.1 The starting points

We encoded the location of each region by a point on its contour. This is the starting point of the chain code description of the region contour. There is practically no correlation between two successive starting points, simply because there is no

correlation between the location of two successive regions in the transmission sequence. For this reason, the starting point information has not been entropy coded and requires 16 bits: SP is always a 16-bit codeword.

There are two ways of reducing the number of bits required to encode the starting points. The first one consists in permuting the order in which we transmit the regions. In practice, the image is not updated after reception of each new region. Therefore, when the image is updated on the display, the order in which the regions have been transmitted is of no importance. Thus, we can transmit the regions in a predetermined number of scans. We can then encode the offset between the starting point on each passes.

On the other hand, we expect that the content of the image can be identified from a dozen regions, to one or two hundred regions, at the most. The number of regions transmitted in each sequence should then increase geometrically, and the improvement would be minimum in the first stages of transmission: 1 or 2 bits saved on each starting point.

The second solution lies in the fact that most of the starting points are located on the contours of previously transmitted regions. It is not always the case since, some regions are totally enclosed by others. If we keep a list of transmitted contours at both ends, we can reduce the number of bits required to indicate the starting points: from 16 bits to address the whole image plan to 10 to address the first 1024 contour points.

Both methods add complexity to the scheme and bring very little improvement if we consider that the first regions are usually quite large and require the transmission of thousands of contour points.

5.5.2 The chain code

The contour is encoded from the starting point using a second order Markov model. Fig. 5.9 shows the structure of the chain code. A chain code component CC includes an end of code EC, most of the time preceded by one or possibly more segments, which are made up of at least one absolute direction code AD, which can be followed by a relative direction code RD and a sequence of second order relative direction codes SRD. If more then one segment precede the end of code symbol, they are separated by an end of segment ES symbol and a relative starting point RSP symbol.

$$[AD[RD[SRD...]]][ES,RSP[AD[RD[SRD...]]]]EC$$

"[]" Denotes a sequence appearing 1 or 0 times

"..." Denotes 0 or many repetitions or the preceding code

Fig. 5.9 Structure of the chain code.

Lu and Dunham have shown that good results can be obtained using a second order Markof model [Lu91]. Unlike them, we do not use a differential chain code, because, in our scheme, the probabilities are dependent on the absolute directions. There are three reasons for that: our starting points are always located on the leftmost upper point of the image and the contour scanning is always done in a clockwise motion, all the regions are 4-connected, and we transmit segments, not only closed contours. This reduces the probability of some orientation. Table 5.1 shows the frequency of occurrence of each direction in the full transmission of one image.

We assumed that the contour information is stationary. The statistics were gathered on one image with 10769 regions and 65 000 contour points. Similar statistics were obtained with other images, and the contour information does not appear to vary

significantly from one image to another. We attribute this to the high number of contours and regions in each image.

Previous	Next direction									
direction	0	1	2	3	4	5	6	7	ES	EC
0	0.255	0.165	0.145	0.06	0.01	0	0	0.140	0.025	0.200
1	0.227	0.076	0.265	0.086	0	0	0.038	0.184	0.049	0.184
2	0	0.058	0.265	0.060	0.053	0.074	0.081	0	0.006	0.403
3	0	0.037	0.205	0.074	0.091	0.088	0.110	0	0.005	0.388
4	0.014	0	0	0.117	0.209	0.080	0.128	0.046	0.011	0.395
5	0.008	0	0	0.095	0.228	0.091	0.102	0.040	0.012	0.424
6	0.038	0.008	0.002	0	0	0.208	0.409	0.067	0.014	0.254
7	0.117	0.044	0.073	0	0	0.122	0.263	0.049	0.024	0.307

Table 5.1 Empirical probability distribution for the first order Markov model.

We observed that the probability of an EC symbol is very high. This is a consequence of the high number of contours in the image and their small size. It would have been possible to encode the contour information using two of more sets of table adapted to a particular stage of the transmission since longer contours are more frequent in the early stages. However, since size is not a criterion in our simplification process, there is also a large number of short contours in the first stages of the transmission and the gain is not as important as it would first appear to be.

Previous direction	Next direction									
	0	1	2	3	4	5	6	7	ES	EC
6 0	0.258	0.100	0.106	0.017	0	0	0	0.312	0	0.206
6 1	0.322	0.191	0.113	0.052	0	0	0	0	0	0.122
6 2	0	0.317	0.488	0	0	0	0	0	0	0.195
6 3	0	0	0	0	0	0	0	0	0	0
6 4	0	0	0	0	0	0	0	0	0	0
6 5	0.039	0	0	0.105	0.223	0.165	0.252	0.057	0.007	0.151
6 6	0.038	0.012	0.012	0	0	0.131	0.643	0.700	0.005	0.089
6 7	0.077	0.030	0.033	0	0	0.122	0.458	0.156	0.002	0.122

Table 5.2 Partial probability distribution for the second order Markov model.

The statistics were used to produce a set of Huffman tables. Some of these tables are presented in Table 5.3 and Table 5.4.

Previous direction	Next direction									
	0	1	2	3	4	5	6	7	ES	EC
0	2	3	3	4	6	7	7	3	5	2
1	2	4	2	4	3	6	6	5	4	3
2	7	5	2	5	5	4	4	7	6	1
3	7	5	3	4	4	4	3	7	6	1
4	6	8	8	3	3	4	3	5	7	1
5	7	8	8	4	2	4	4	5	6	1
6	5	7	8	9	9	3	1	4	6	2
7	3	4	4	6	6	3	2	4	5	2

Table 5.3 Huffman codeword length for the first-order Markov model.

Previous direction	Next direction									
	0	1	2	3	4	5	6	7	ES	EC
6 0	2	4	3	5	7	7	7	2	7	2
6 1	2	3	3	4	6	6	6	2	6	3
6 2	6	2	1	6	6	6	5	6	6	3
6 3	3	4	3	4	3	3	3	4	3	4
6 4	3	4	3	4	3	3	3	4	4	3
6 5	5	7	7	3	2	3	2	4	6	3
6 6	4	6	5	8	8	3	1	3	7	3
6 7	4	6	5	8	8	3	1	3	7	3

Table 5.4 Partial Huffman codeword length for the second order Markof model.

Huffman coding was favored over arithmetic coding for simplicity, since there is no frequent symbol probability over 0.5. We also made no attempt at using an adaptive model.

The efficiency of the Huffman code, as per Eq. 2.8, is 92% .

5.5.3 The intensity values

The intensity can be predicted efficiently because we have access to all previous and past values. However, as it was the case with the starting point, the efficiency of the compression is not critical since most of the transmitted information is used to describe the contours. Since each region is represented by an average intensity, it is

sufficient to transmit the intensity of one region to describe the intensity of both, since the average intensity of the joint region is known.

If region X_A is to be divided in regions $X_{A'}$ and X_B with respectively N_A, $N_{A'}$ and N_B pixels, and average intensities I_A, $I_{A'}$ and I_B,

$$N_A I_A = N_{A'} I_{A'} + N_B I_B. \tag{5.6}$$

There is, however, little correlation between the intensity of two successively transmitted regions. For this reason, it is more convenient to consider the difference between two regions, since the regions are split according to this criterion. The intensity of both regions can be calculated from the difference value if we know which region has the highest intensity.

If $I_{A'} > I_B$ then:

$$I_{A'} = I_A + {}^{N_{A'}}\!/_{N_A} \cdot \Delta I, \tag{5.7}$$

$$I_B = I_A - {}^{N_B}\!/_{N_A} \cdot \Delta I, \tag{5.8}$$

where $\Delta I = I_{A'} - I_B$.

We used a sign bit to indicate which region had the highest intensity. The amplitude of the difference was then predicted.

Fig. 5.10 shows the empirical probability distribution of the prediction error obtained using a zeroth order predictor. These probabilities were used to create the Huffman table in Table 5.5. The efficiency of the Huffman code is 89%.

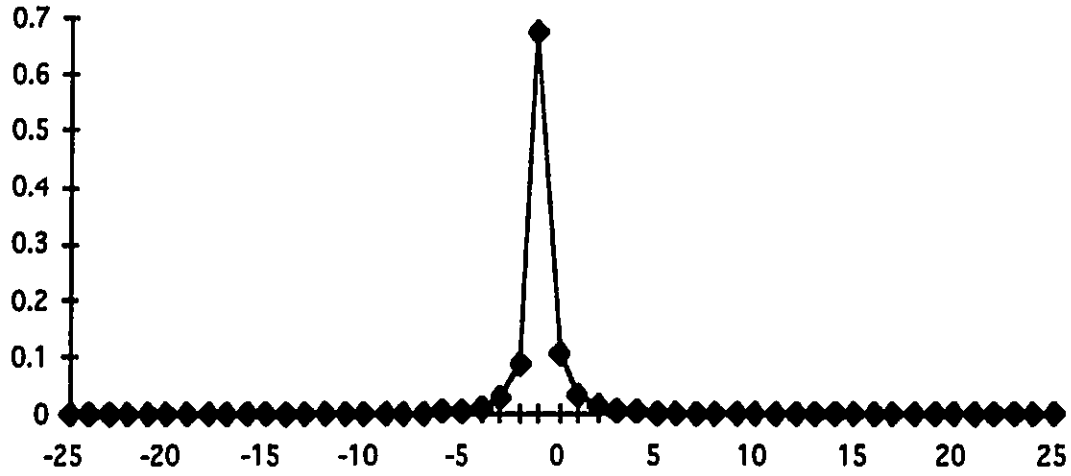


Fig. 5.10 Probability distribution of prediction error.

ΔI	Codeword length
-9	10
-8	9
-7	8
-6	8
-5	7
-4	6
-3	6
-2	5
-1	3
0	1
1	3
2	5
3	5
4	6
5	7
6	8
7	8
8	9
9	9

Table 5.5 Huffman coding table for the prediction error ΔI.

5.6 Decoding

The decoder is composed of an entropic decoder and a region grower, as shown in Fig 5.11. The graph representation is reproduced by the decoder as a label image. This is possible because the regions are connected, as we specified in Eq. 5.3.

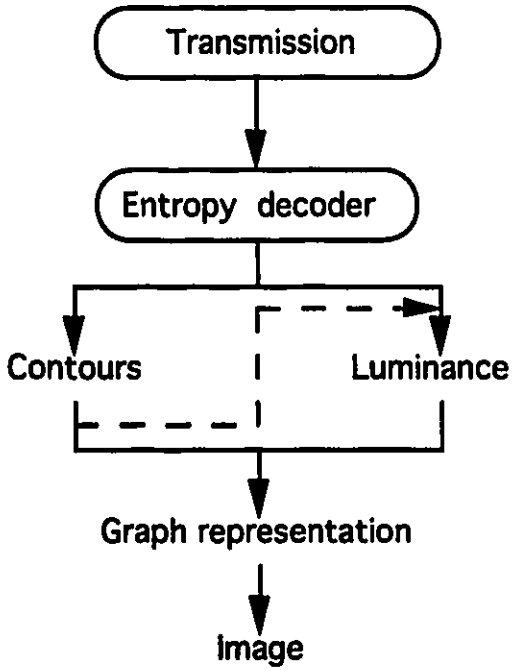


Fig. 5.11 Diagram of the segmentation based decoder.

To split a region labeled A into two regions labeled A and B, we transmit the inside contour of the new region which does not contain the first pixel x of the initial region A. We then reconstruct region B using the region A less the contour as the geodesic mask, with x as a marker:

$$B = R_{(A-C_4(A'))}x. \tag{5.9}$$

The contour is then erased and relabeled as part of A. The process is illustrated in Fig. 5.12.

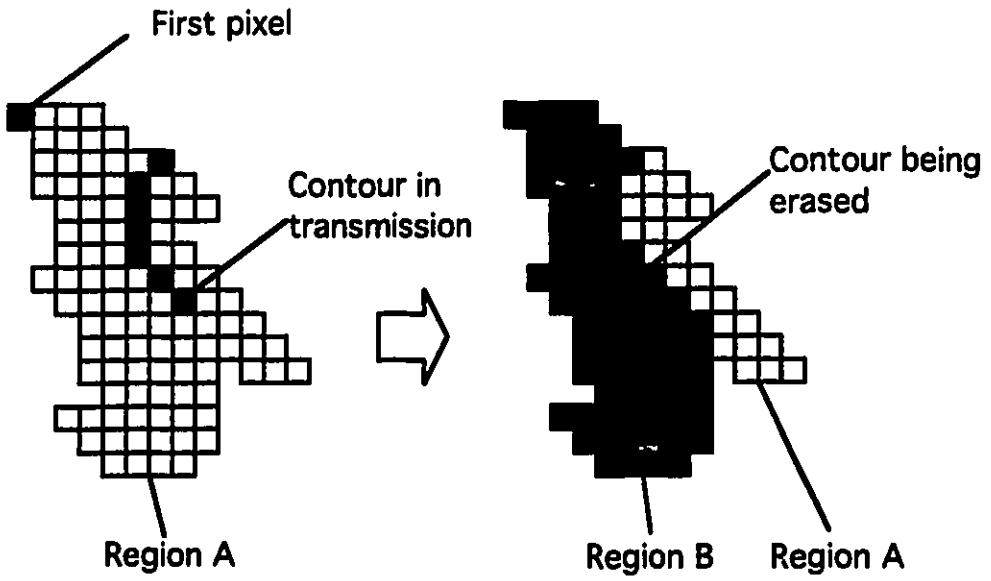


Fig. 5.12 Region reconstruction.

5.7 Summary

In this chapter we described the implementation of our segmentation-based PIT scheme. Our algorithm was designed to transmit the most contrasted contours first. We started by transposing the image on a graph of neighboring regions with constant luminance. This is achieved through the use of the watershed as a segmentation operator. We commented on the choice of the gradient, the markers, and on how to remove the divide lines.

The next step consisted in ordering the region by eliminating the weakest edges and merging the regions until only one remains. The regions were then transmitted in the reverse order into which they were merged.

We describe the shape of a region using Freeman chain code and a second order Markov model, and the intensity by using DPCM on the value of the difference between the two newly formed regions. This information is then encoded using a set of Huffman tables. Decoding is straightforward and involves reconstruction to reproduce the regions from the contours.

Chapter 6

Subjective Testing

This chapter describes the design, the implementation, and the results of a subjective experiment to determine the time required to identify objects using different progressive image transmission schemes.

6.1 Introduction

We saw in Chapter 4 that the mean square error and SNR are still the most widely used distortion measures. In the past, when one was dealing with an analog signal that was distorted by noise as it was transmitted over a network, these measures were quite appropriate. Today, however, when we are dealing with quality loss resulting from a digital compression process, these measures often do not correlate well with the quality as perceived by the end users [Wol91]. This can be simply attributed to the fact that digital compression has introduced many impairments, such as the blocking artifacts of transform coding, that are not easily quantified by a mean square measure.

Our approach here was to rely on subjective testing to evaluate the performance of our compression scheme. Furthermore, we used the results of our tests to validate objective quantitative measures in the context of progressive image compression. This is consistent with the methodology used to design objective measures as seen in Fig 6.1.

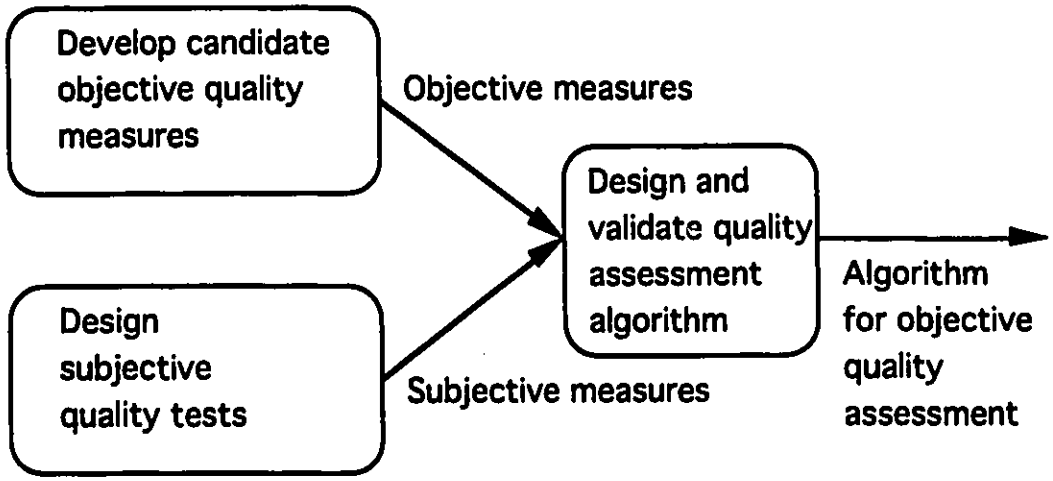


Fig. 6.1 Methodology for development of objective quality algorithm. [Wol91]

6.2 The Experiment Design

There are two important setbacks to the use of subjecting testing, [Miy88] and they must be both addressed early in the design of the tests. The first thing to consider is the time required to conduct the test. We assumed that the testing would require close supervision and should last less than one hour. To guarantee the feasibility of the testing with our limited resources, we assumed that 50 subjects would provide sufficient data.

The second factor to consider is the following: for subjective testing to yield useful information, the testing should mirror, as closely as possible, the actual condition

under which the service would be utilized [Wol91]. In particular, the selection of source material plays a critical role when dealing with lossy data compression. The higher the compression factor, the stronger the dependence between output quality and input source characteristic.

One of the main application for progressive image transmission is image database browsing. We selected this application as the service we would provide. Since we are interested in the identification phase, and not in the inspection phase of the browsing task, a measure based on performance seemed more appropriate than a measure based on a subjective quality ranking. Furthermore, we expect to encounter less variance in the results.

Our initial intent was to design a browsing task and compare the performance of subjects using our algorithm to those of subjects using other techniques. This approach faced one criticism: the results obtained using one browsing strategy could not readily be generalized to others. In fact, what we really wanted to know was, given a limited bandwidth, how fast can an image be recognized by a user. The next section will describe how we implemented a test to answer this question.

6.3 The Experiment

6.3.1 The Image Set

A set of 45 images was scanned from two merchandise catalogs. Each image is centered on an object which formed the main component of a scene. Some pictures also included some contextual information, for instance, part of a desk associated with an office chair. Others showed parts, or reduced reproduction of different models of the main object, as in the bike image in Fig. 6.2.

When possible, all the textual information was removed from each picture. All the images were then manipulated to improved the contrast of the objects. Some contrast loss can be attributed to the scanning process, the rest is due to the conversion of color images into gray-scale.

Some granular noise is visible on certain images. This noise was also visible on the printed original and seemed consistent with the subjective quality required to sell each object. For instance, large appliances need less image quality than furniture. Fig. 6.2 shows 4 of the 45 images used in the experiment. The complete data set is printed in Appendix A.

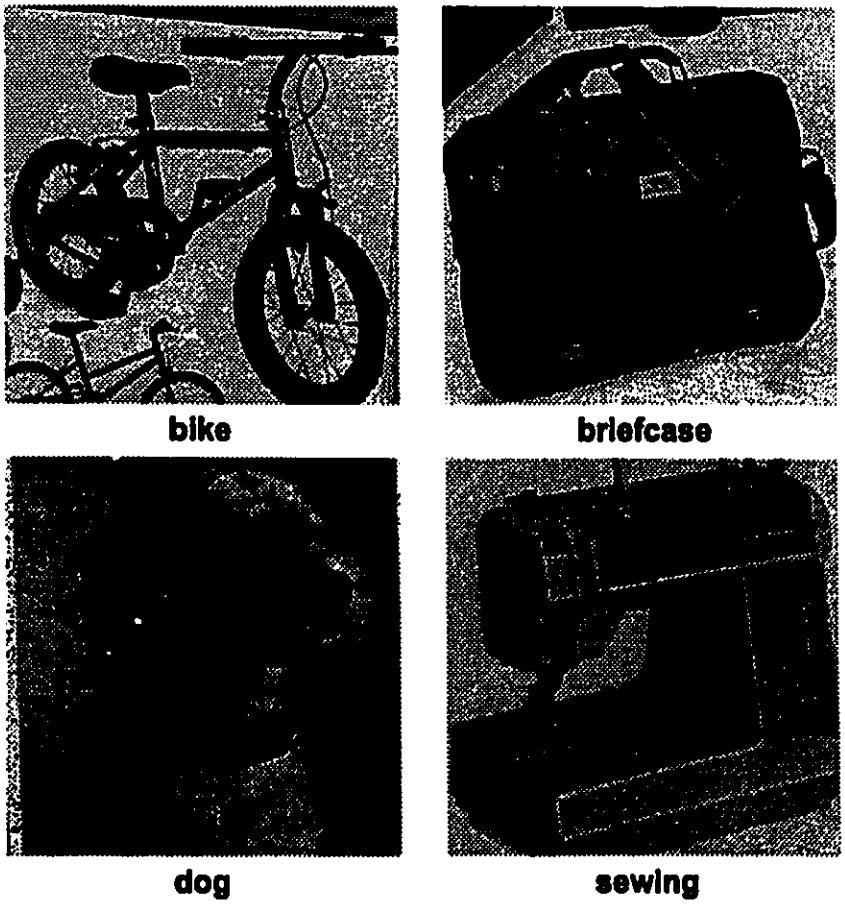


Fig. 6.2 Image set sample.

Each image was then encoded using both the JPEG standard in spectral mode and our method. For JPEG, we used the quantization table presented in Table 6.1. Each method was used to produce 14 frames at different points in the transmission. Each sequence of image was used to construct an animation simulating progressive transmission at 2400 bps. An example of a sequence is illustrated in Fig. 6.3. This particular bandwidth was selected so that the delay introduced was large enough to be easily measured. This should be consistent with a commercial service provided over an inexpensive link. We also note that lower transmission rates are common over the Internet.

16	11	12	14	12	10	16	14
13	14	18	17	16	19	24	40
26	24	22	22	24	49	35	37
29	40	58	51	61	60	57	51
56	55	64	72	92	78	64	68
87	69	55	56	80	109	81	87
95	98	103	104	103	62	77	113
121	112	100	120	92	101	103	99

Table 6.1 JPEG quantization table.

6.3.2 Subjects

Fifty observers were recruited for the experiment; all were students in the Faculty of Engineering at the University of Ottawa. Each participant was assigned randomly to one of three groups. Each group experienced one method of transmission; JPEG, PIT by segmentation-based coding, or simply the instantaneous transmission of the

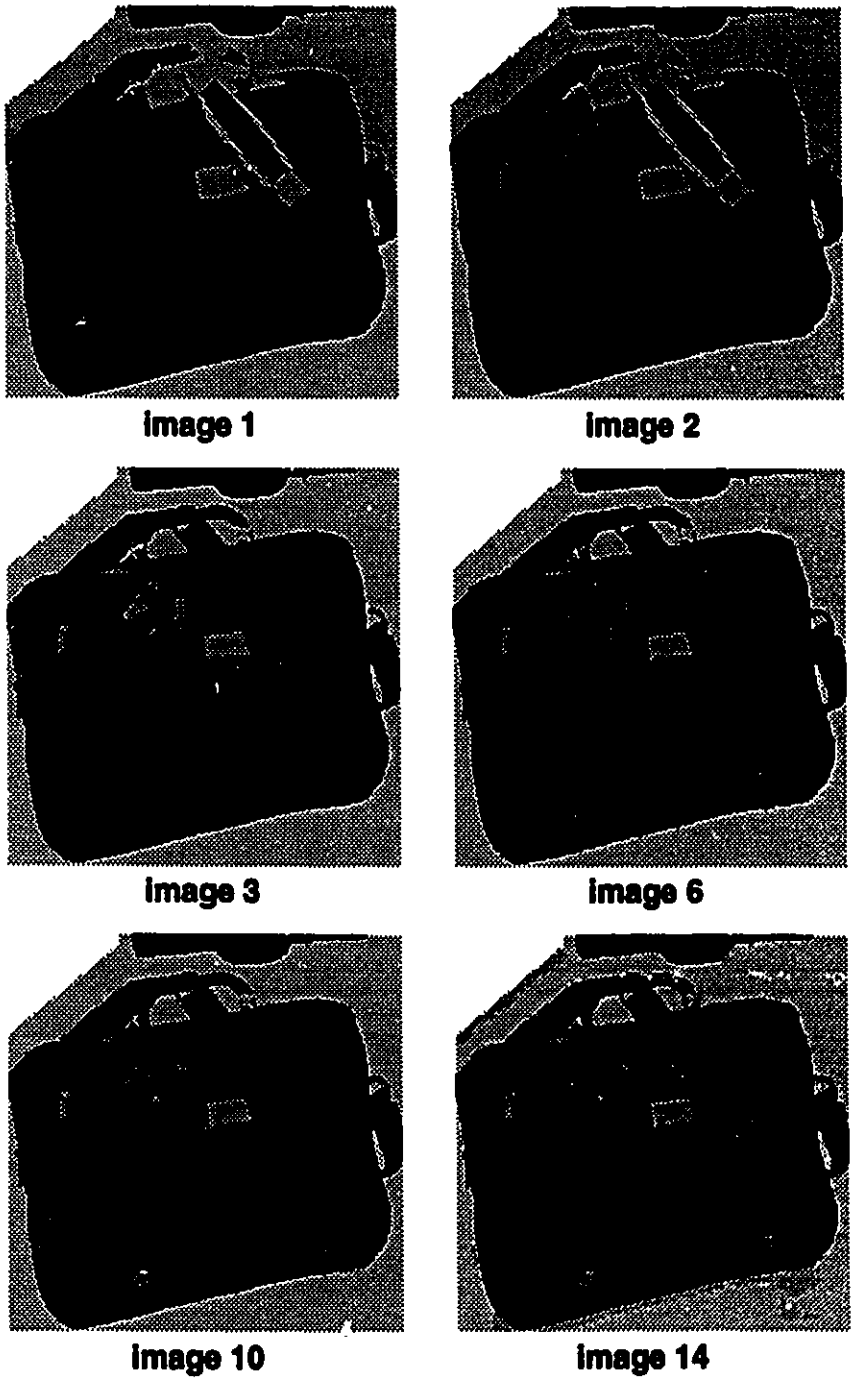


Fig. 6.3 Part of PIT sequence with segmentation-based coding.

original image, which constituted our control group. Seventeen subjects viewed the progressive transmissions, and sixteen the instantaneous presentation. In each

group, only the fifteen subjects with the smallest number of misidentifications were used. This procedure eliminated technical difficulties with the response time measurement.

6.3.3 Materials and Apparatus

A Sun Sparcstation 20 with a 20-inch screen and a headset microphone was used in the simulation. The display was set to a neutral gray level intensity corresponding to the luminance of the pixel value 128 on the test images. A black cross of 2 cm across was displayed in the center of the background to mark the location where the image appeared, as shown in Fig. 6.4.

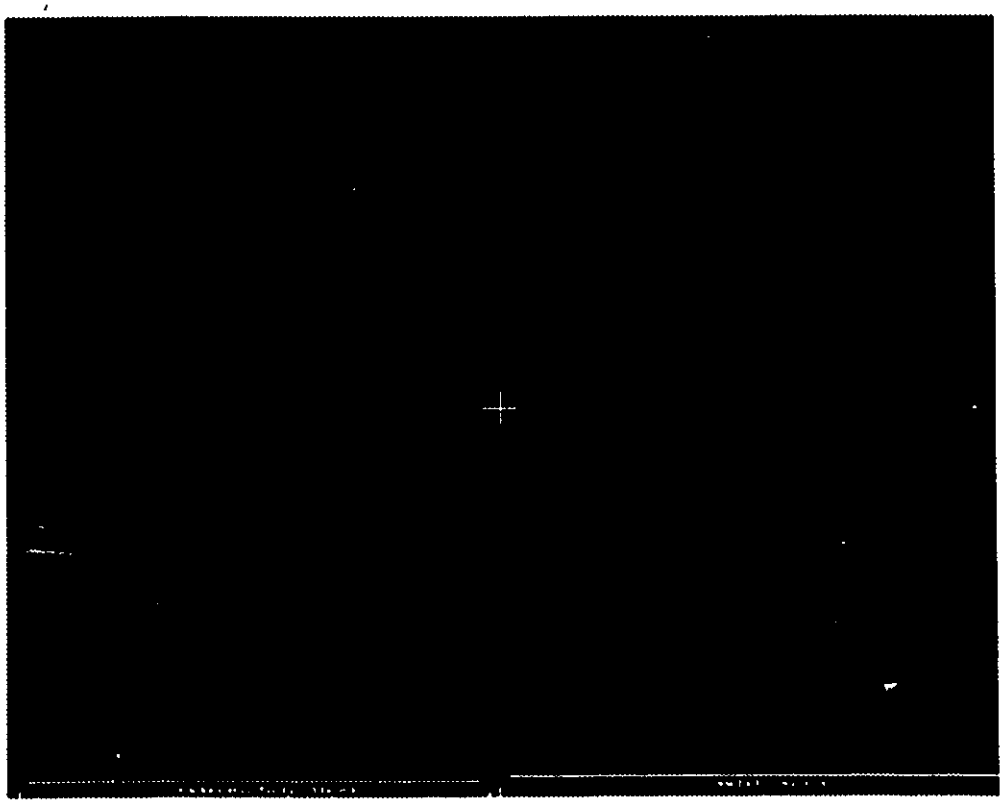


Fig. 6.4 Screen before image transmission.

The image sequences were displayed in a window with a thin border and the same gray-level intensity as the background, as shown in Fig 6.5.

Recognition times were measured by asking the subject to orally identify the image. The voice signal was acquired using a highly directional microphone mounted on a headset. The signal was digitized in the audio port of the workstation, and properly processed in order to detect the beginning of the utterance. Using this setup, we were able to measure the response time with a precision of 1 msec.

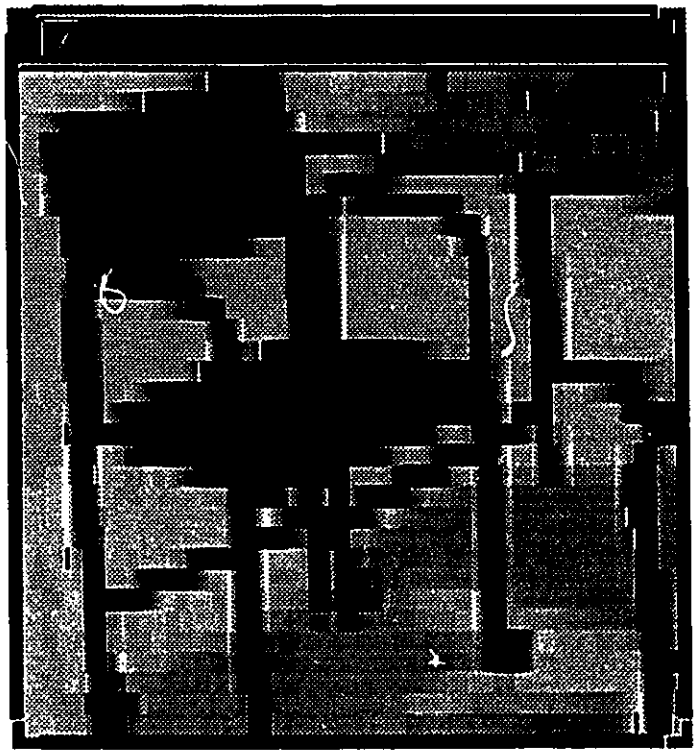


Fig. 6.5. Example of an image and the display window.

6.3.4 Procedure

Participants were provided with printed instructions indicating the nature of the experiment. They were then outfitted with the microphone and asked to name the

central object in the scene, as soon as they could identify it. Participants in the progressive image transmission simulations were informed that the image would build up gradually. No standard response form was used, e.g. "This is a ...", since some participants could have used the elocution time associated with the phrase to analyze the image further. The transmission was not interrupted by the participant's response, but was allowed to build up for 25 seconds so that the participants could see whenever they made a mistake. Each participant was tested individually. The participant was first presented with a training session of three images (case, bed, and oven). The participant was given control over the simulation and had to press a key to pass to the next image. The identification of each image was marked, based on a pass/fail criterion that indicated whether the participant had correctly identified the image or not.

6.4 Some Experimental Results

In this next sections of this chapter, we present the results of the testing procedure we just described. First, we look at the results for one particular image under one transmission scheme. Then, we extract a central value to characterize the performance of each scheme for each image. We do the same to generalize the results to the whole data set. We also look at the variations of the results. Finally we compare the performance of our segmentation-based technique with JPEG's.

We retained 3 sets of 15 subjects, one for each transmission scheme of chapter 6. Since each subject identified 42 images, this adds up to 1890 recognition times. There is obviously a need to describe the results more compactly. We can see histograms of the results involving the 4 images presented in Fig. 6.2, in Fig. 6.6–18. Most of the histograms are unimodal but some, like Fig. 6.10, are not. Furthermore, the mode is

not always at the centre of the distribution, as shown in Fig. 6.9. Finally, some extreme values are not likely to be representative of the recognition time, since they can be caused by a delay in finding a word, for example. For these reasons, the mean is not a good central value for these histograms. The median is a better choice.

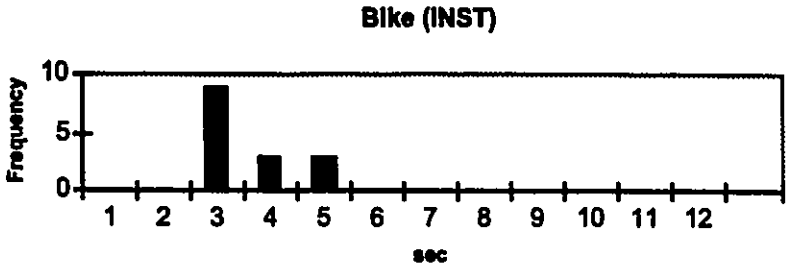


Fig. 6.6. Histogram of identification times for the image Bike transmitted instantaneously.

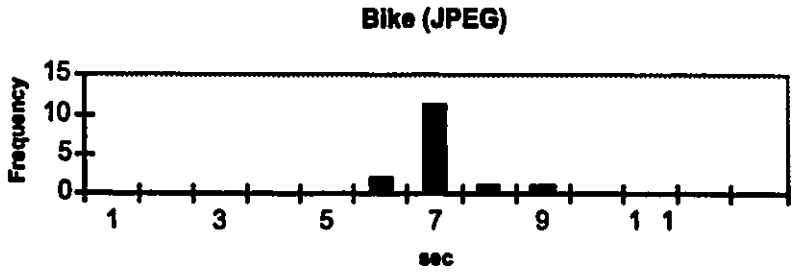


Fig. 6.7 Histogram of identification times for the image bike transmitted with JPEG.

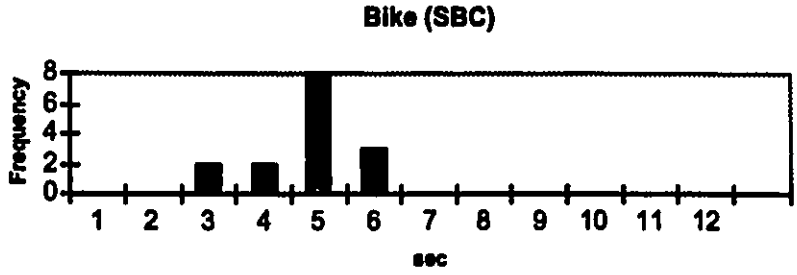


Fig. 6.8 Histogram of identification times for the image bike transmitted with SBC.

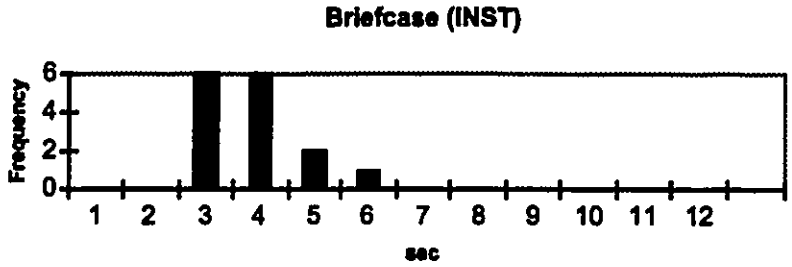


Fig. 6.9 Histogram of identification times for the image briefcase transmitted instantaneously.

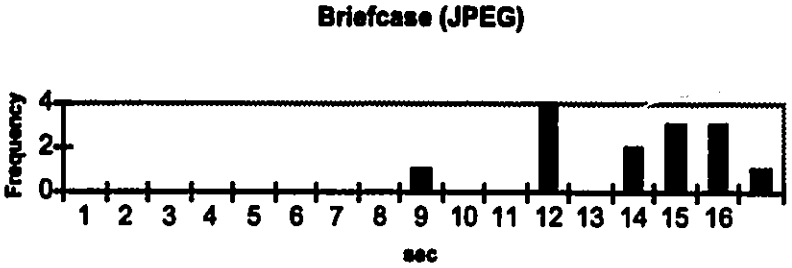


Fig. 6.10 Histogram of the identification time for the image Briefcase transmitted with JPEG.

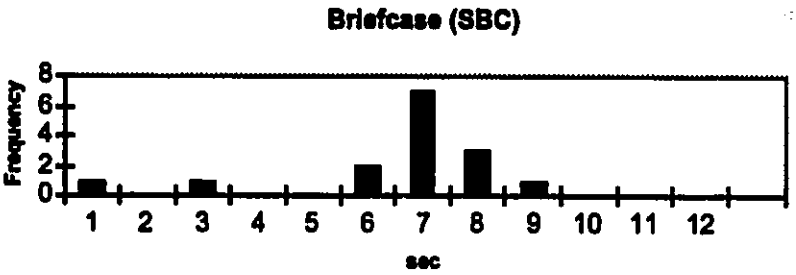


Fig. 6.11 Histogram of the identification times for the image briefcase transmitted with SBC.

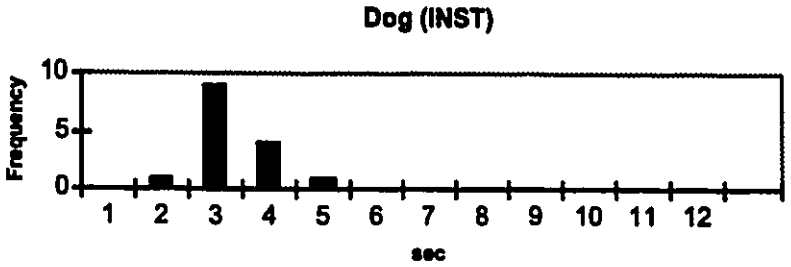


Fig. 6.12 Histogram of the identification times for the image dog transmitted instantaneously.

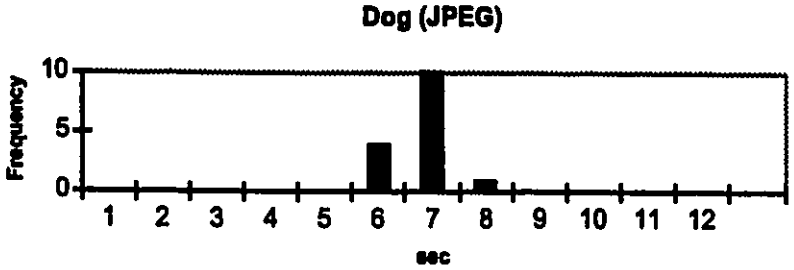


Fig. 6.13 Histogram of identification times for the image dog transmitted with JPEG.

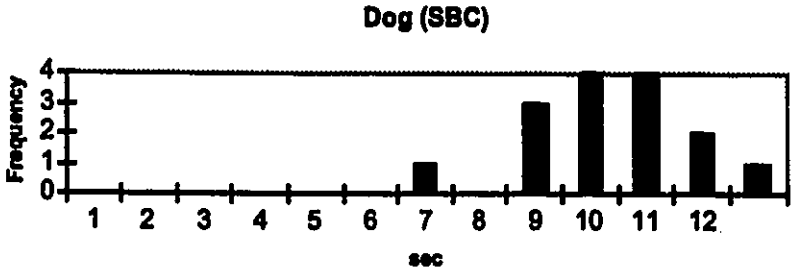


Fig. 6.14 Histogram of identification times for the image transmitted with SBC.

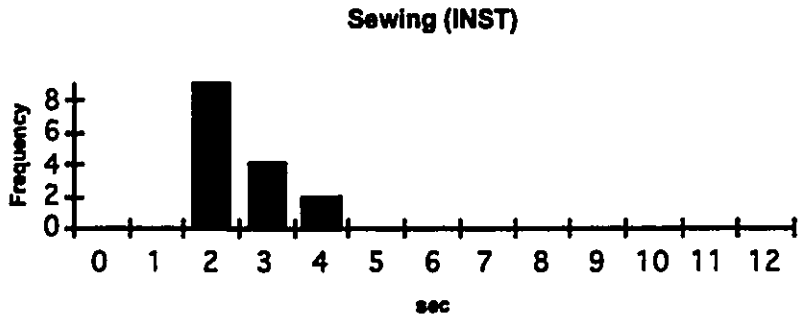


Fig. 6.15 Histogram of the identification times for the image sewing transmitted instantaneously.

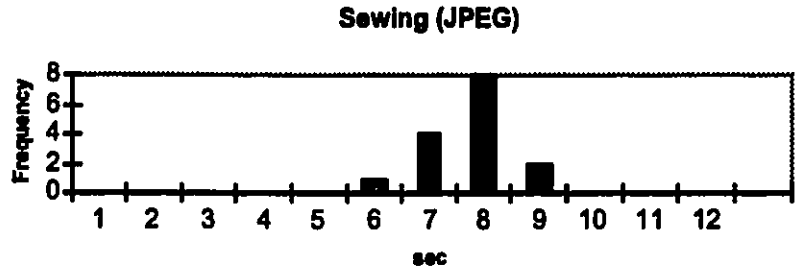


Fig. 6.16 Histogram of the identification times for the image sewing transmitted with JPEG.

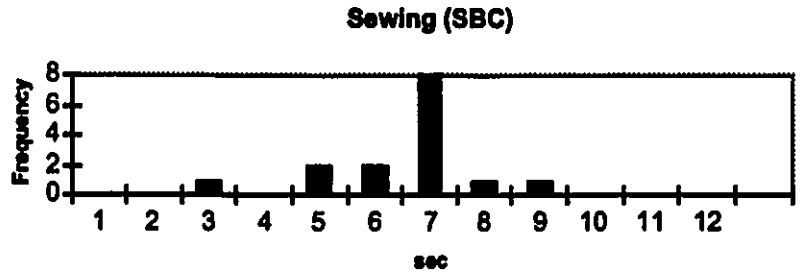


Fig. 6.17 Histogram of the identification times for the image sewing transmitted with SBC.

6.5 Median Values

Fig. 6.18 shows the average recognition time for the instantaneous transmission. We can see that there is not much variance between images.

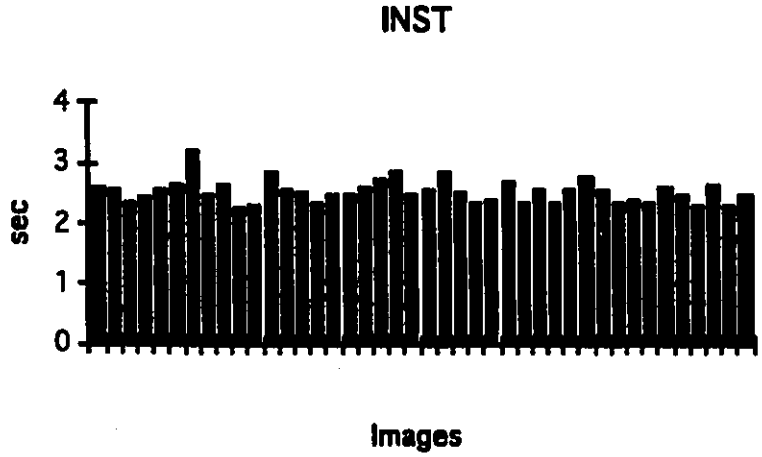


Fig. 6.18 Median recognition times for the instantaneous transmission.

Fig. 6.19 shows the median results for the progressive sequence using JPEG and SBC side by side. We can see that SBC outperforms JPEG for 38 of the 42 images of the data set. We do not attempt to quantify by how much at this stage, because there is still some variance in the data that relates to the subjects. It is interesting, however, to consider the four images for which JPEG gave better results.

The first is the dog image. This comes as no surprise since this image contains a lot of textural information and very few well-defined contours. In fact, there is virtually no constant valued region on this image, a feature our methodology relies heavily upon. The second is the boot image which is also very rich in texture and contains 3 sets of alpha-digital characters which were costly to encode using segementation, but were not helpful in identifying the image. The third image, phone, was not adverssly affected by the blocking effect of JPEG because the buttons of the phone where large enough.

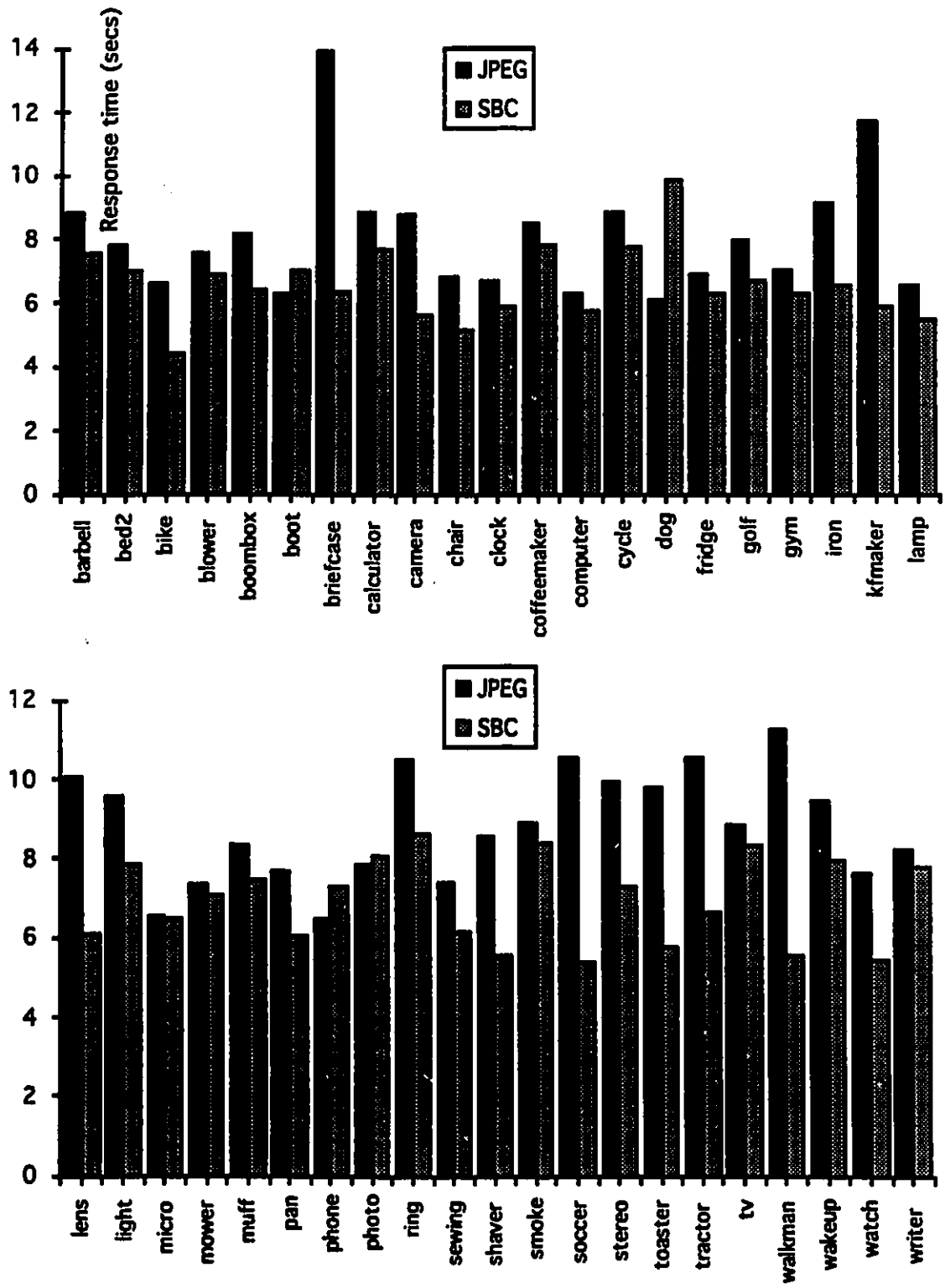


Fig. 6.19 Median recognition times for JPEG and SBC.

Finally, the photo image scored almost equally with both techniques: 7.87 seconds with JPEG and 8.09 seconds with our method.

6.6 Delay Measurements

We cannot compare directly the recognition times of different images. We cannot use the recognition times to quantify the performance of one method over the other either, because the recogition times are not caused only by the delays in the image transmission. To correct this, we subtracted the values obtained for the instantaneous transmission from the values obtained with both PIT methods. This allowed us to express the identification times in terms of identification delays. It was then possible to directly compare the identification delays associated with each method. Furthermore, by doing this, we reduce the delays that could be attributed to difficulty in identifying some of the pictures over others. Fig. 6.20 shows the median recognition delay for all the images, using JPEG. Fig. 6.21 shows the same for our scheme, using segmentation based coding.

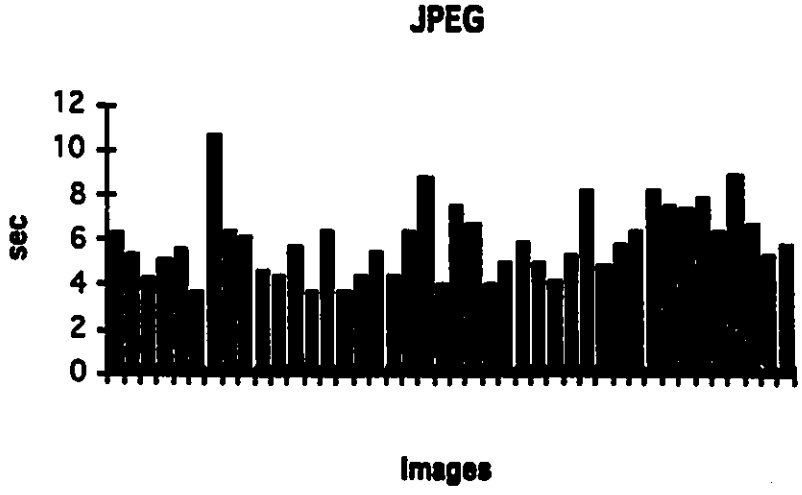


Fig. 6.20 Median recognition delay using JPEG.

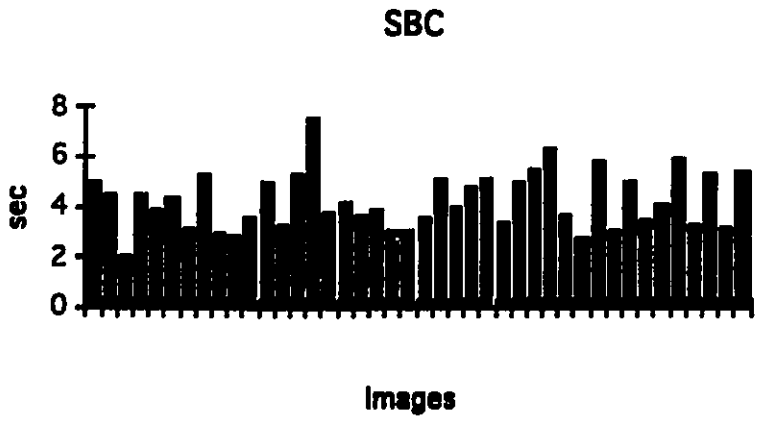


Fig. 6.21 Median recognition delay using SBC.

From these values we can compute the mean recognition delay for all the images of the set. We find the mean delay to be 5.91 s for JPEG and 4.22 s for SBC. This represents a 29% reduction in the delay.

6.7 Variation Analysis

We close this chapter by quantifying the variations in the results. We computed the inter-quartile difference given by the maximum of the values in the first quartile subtracted from the minimum of the values in the last quartile, for all the images in both progressive schemes. The results can be seen in Fig. 6.22 and Fig. 6.23.
Most of the differences are under 1.5 s in the segmentation-based method, while the average value is approximatively 2.5 s using JPEG. This suggest that identification is more dependent on the subject when JPEG is used.

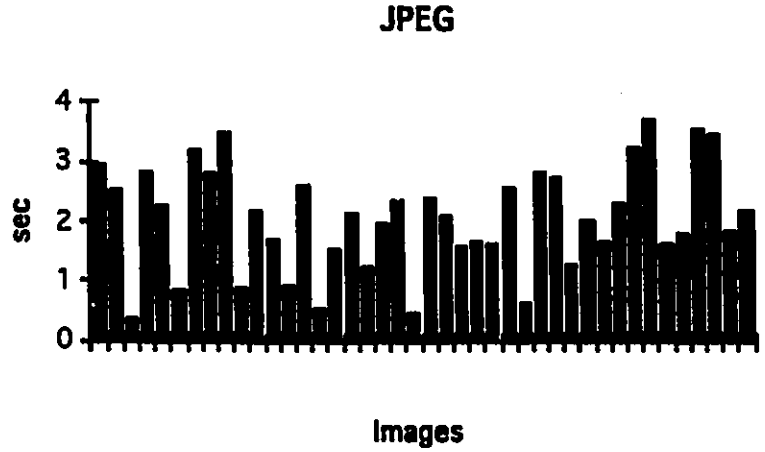


Fig. 6.22 Inter-quartile difference in identification times using JPEG.

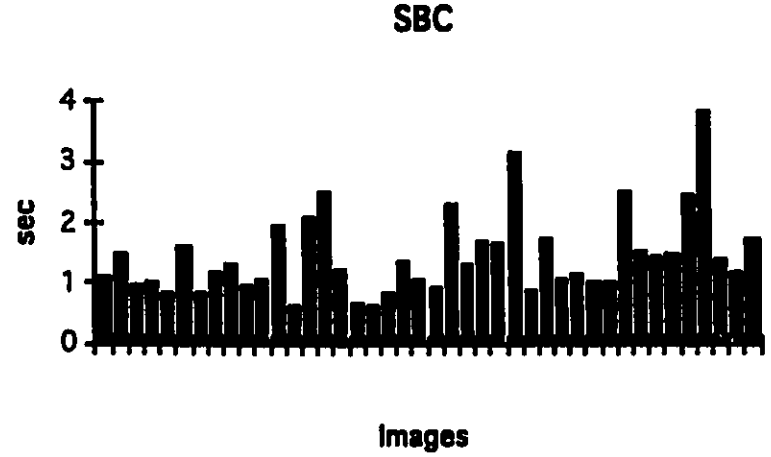


Fig. 6.23 Inter-quartile difference in identification times using SBC.

6.8 Summary

In this chapter, we described the subjective experiment we conducted. We presented our design constraints and the implementation details: material, procedure, subjects and image data set. We then saw that segmentation-based coding clearly outperforms JPEG on the test images used. The improvement using a bandwidth of 2400 bps is, in

average, a 29% reduction of the identification time. Furthermore, variations in the results, as measured by the inter-quartile difference, indicate that the results vary more with the observer if JPEG is used.

Chapter 7

Objective Measurements

In this chapter, we present some objective measurement based on edge rendition. We begin our discussion with an analysis of the mean square error on some of the image sequences to substantiate the claim we made in Chapter 4 that MSE is not suitable to evaluate a PIT scheme.

7.1 Mean Square Error

Fig. 7.1 shows the mean square error of Eq. 4.6, computed on the first 14 images of 6 JPEG sequences used in our experiment. There is a strong variation in the magnitude of the measurement from one picture to another. However, there is no correlation between the magnitude of the measurement on two different images and the recognition delays we associated with them as can be seen in Table 7.1 This holds for all the 42 images in the data set.

Fig 7.2 shows the MSE for two sequences of transmission involving the same image but the two different transmission schemes: JPEG and SBC. Mean square error measurements give better ratings for the images encoded with JPEG. We obtained similar results with all the other images.

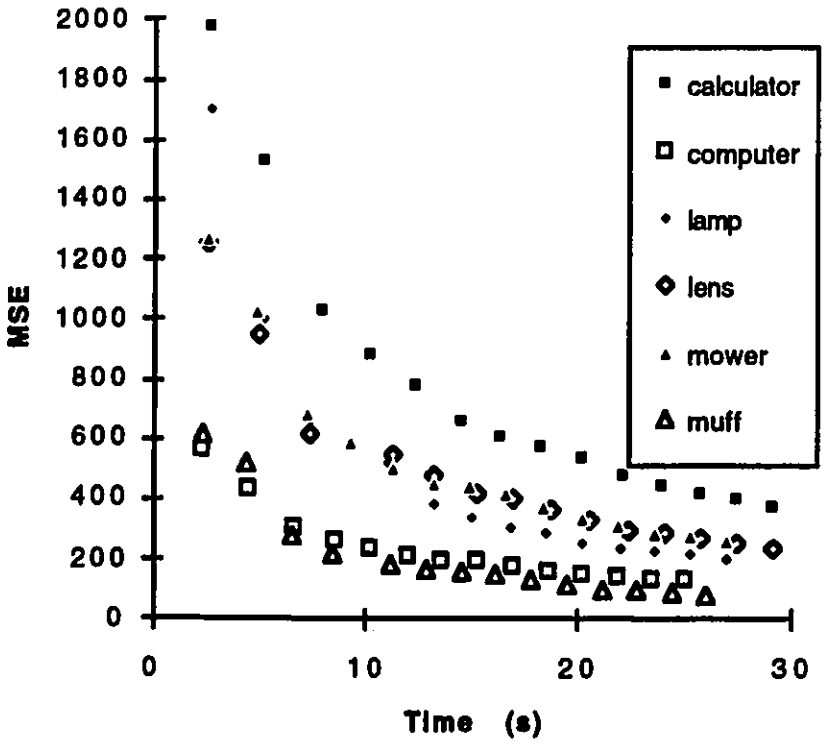


Fig. 7.1 Mean square error on six JPEG sequences.

Images	Recognition delay
calculator	6.377
computer	3.707
lamp	4.051
lens	7.484
mower	4.984
muff	5.953

Table 7.1 Recognition delays for six JPEG sequences.

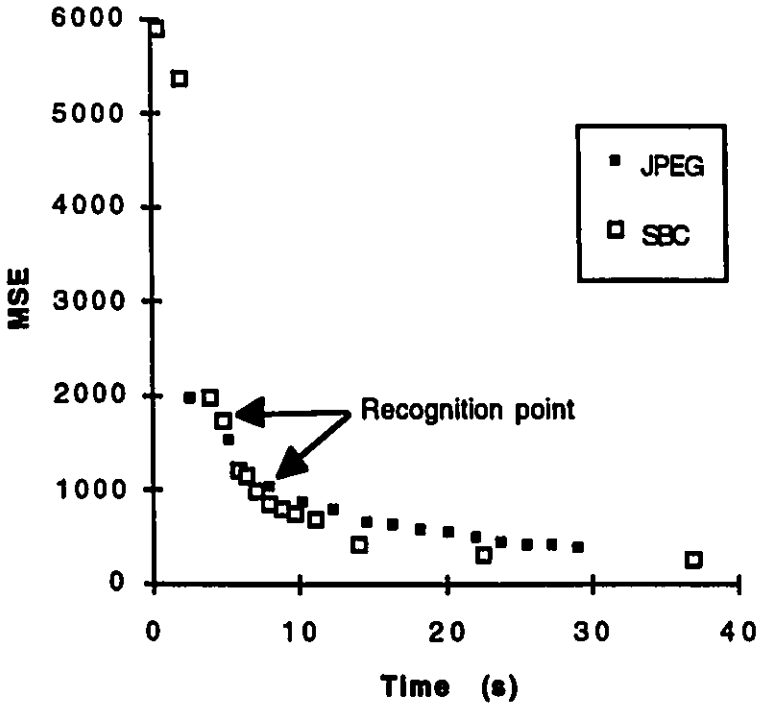


Fig. 7.2 Mean square error for Calculator sequences with JPEG and SBC.

The recognition points have been marked on each sequence. They correspond to the highest quality image of each sequence that can be transmitted during the recognition delay. Clearly, MSE gives us no indication as to where those points can be found, or what the recognition delay on a given sequence is.

These two examples illustrate the inadequacies of the mean square measure in dealing with different types of distortion. It also indicates that the MSE is dependent on the image characteristics.

As previously explained in Chapter 4, an objective measure is preferable to a subjective one. There are other distortion measures in the literature, but none is suitable for large impairments.

The next section introduces a novel approach to distortion measurements.

7.2 Edge Measurements

If we look at the question from a qualitative point of view, we can postulate that the information contained in an image is concentrated in the edges, which delimit the object contours. With this in mind, it seems possible to assess how much information has been lost by measuring how much the contours have changed.

This corresponds to adding the edge extraction mechanisms of the cortex to the HVS model of Fig 4.4.

7.2.1 Edges and the HVS

We saw that, according to Weber law (Eq. 4.3), luminance is not proportional to perceived luminance. However, the commonly used CRT displays are also non-linear and somewhat compensate for Weber law. In our experimental set-up, we verified that, the least noticeable distortion was constant over the gray-scale range and approximately equal to 5. Hence, the computations can be done on the gray-scale values.

Since our metrics are based on edge preservation, the preprocessing effect of the lateral inhibition can be discarded.

Finally, the cortex is a complex system that extracts edge segments according to their position and their orientation. A large system would be necessary to duplicate this functionality. However, since we are dealing with instantaneous identification, we can expect the eye to focus on the most visible features. Therefore we can simply compare the two edge maps obtained from the original image and its approximation.

An edge map is usually a binary image obtained from thresholding a gradient image. For our application, we set the threshold at the value corresponding to 75% of the cumulative distribution of the morphological gradient $M8$.

7.2.2 Edge Gain and Edge Loss

We measured independently the loss of an edge and the apparition of a spurious one. The loss of an edge corresponds to the loss of some information, while the apparition of an edge creates false information that masks the real scene contents. Furthermore, different processing can be done as part of each measure.

The edge gain is calculated by summing all the contour points on the edge map of the encoded image that are not contour points on the edge map of the original image. The summation is done with a weighting scheme that gives less importance to the points near the contours on the original image. Fig. 7.3 shows the edge gain, in arbitrary units, computed on the six sequences of Fig. 7.1.

The actual measure we propose, for edge gain, EG is

$$EG[U'(x,y)] = \sum_{U} (x,y) \Big| (x,y) \in Map(U') \text{ and } (x,y) \notin \delta_B[Map(U)], \qquad (7.1)$$

where U is the reference image, $Map(U)$ is the edge map of U, $Map(U')$ is the edge map of the distorted image $U'$and B is the elementary structuring element in 8-connexity.

We can see that the measurement depends less on the image than the MSE of Fig. 7.1.

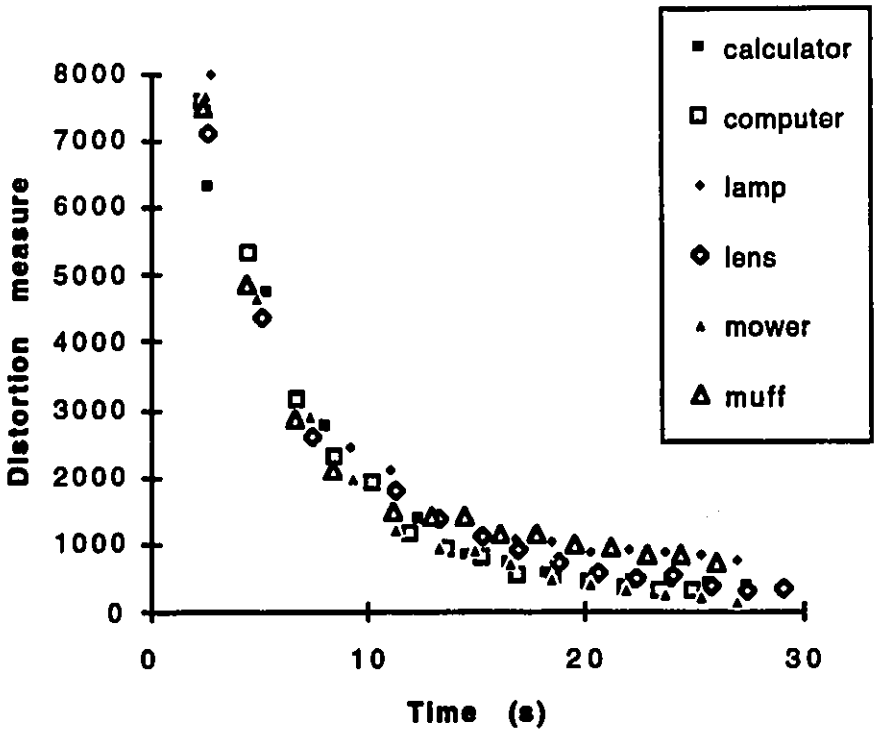


Fig. 7.3 Edge gain for six JPEG sequences.

A similar measure can be obtain for edge loss. In this case, the loss of part of an edge is less perturbing then the loss of a whole edge, and the thinning of an edge is less perturbing still. The edge loss on the JPEG sequences is not significant. The blocking impairment is dominant and it results in extra edges.

The situation is reversed with SBC. Fig 7.4 shows the edge loss in arbitrary units computed on the SBC sequences involving the same 6 images.

The measure we propose for edge loss, *EL*, is

$$EL[U'(x,y)] = \sum_{U} (x,y) \Big| (x,y) \in Map(U) \text{ and } (x,y) \notin \delta_B[Map(U')], \qquad (7.2)$$

The edge loss measurement shows more variance than the edge gain, but the nature of the coding makes it difficult to derive any conclusions from there. The subjective

quality for two intermediary images displayed at the same time varies largely, much more so than with JPEG.

These distortion measurements based on edge preservation are no more useful than MSE in determining at which point in the sequence an image can be recognized. This depends on the structure, the composition, the subject and the quality of the original images. These parameters cannot easily be factored into an objective distortion measure.

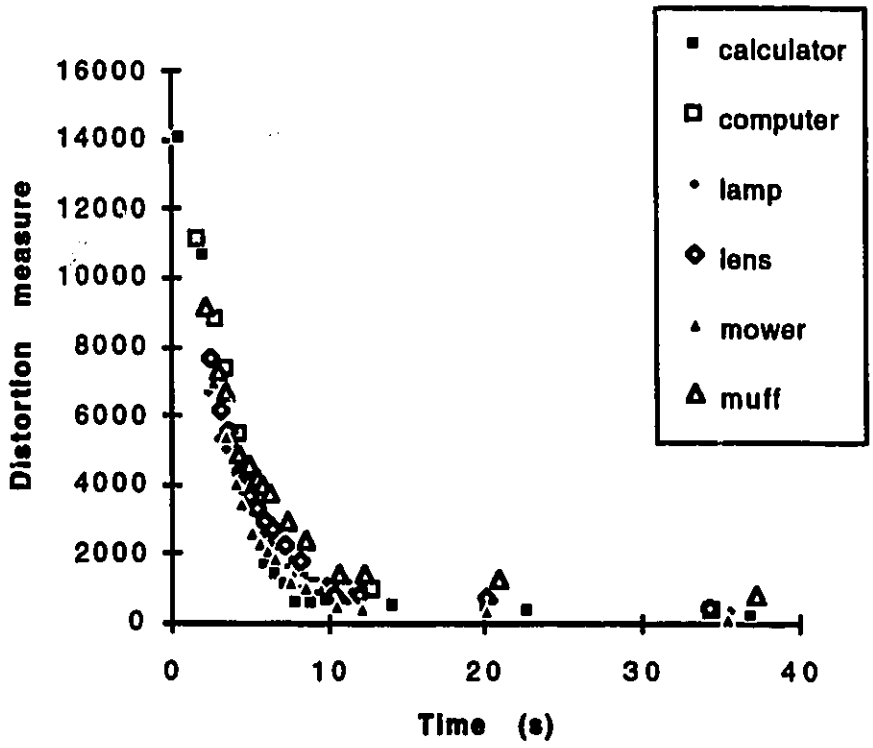


Fig. 7.4 Edge loss for 6 SBC sequences.

Furthermore, because there are few edges lost with JPEG and none added with SBC, we cannot use these measurements to compare the image sequences. It is not clear at this point how the two measurements can be summed, because they have a different physical signification.

7.3 Summary

In this chapter we have demonstrated that MSE is unsuitable to evaluate the quality of image with large impairments, such as the early images in a PIT sequence.

We also introduced two measures based on the edge contents. These measures depart from the classical methodology, but have been found to yield encouraging results. Different images presenting similar levels of distortion when viewed on the screen produced similar results, that was not always the case with MSE. Furthermore, different instances of the same image with more distortion always resulted in an increase in the measure. Unlike MSE, edge measurements take into account the structure of the reference image and its subjective quality.

Unfortunately, the measures we proposed are not appropriate to determine whether an image is intelligible or not. Consequently the results of Chapter 7 cannot be used to validate them. Predicting human performance in object identification might be a more difficult problem than object recognition.

Chapter 8

Conclusion

8.1 Summary

In the present thesis, we presented a methodology for progressive image transmission of gray scale images. The method is based on a representation by segmentation which is suitable for the encoding of moderately complex images at a very low bit rate.

The concept of using segmentation to progressively encode an image is not new. However, in the more classical approach, the partition is fixed and PIT can be achieved by refining the representation of each region. Consequently, the image is usually divided in a small number of large regions. Thus, the initial approximation one can obtain is composed of large regions whose contents are roughly approximated. The lack of detail makes early identification difficult.

In contrast, in our decomposition, the order of the representation of each region is fixed. It is the partitioning of the image that is refined. This gradual partitioning is done in such a way that the strongest edges in the image are transmitted first. As a result, the image content can be assessed much more quickly.

This was proven by conducting a series of tests in which the participants were asked to identify the content of images depicting consumers' goods. Our algorithm was compared with the JPEG standard. The design considerations of this testing, along with the implementation parameters, where outlined in Chapter 6.

The results were analyzed in Chapter 7, and showed that our methodology is indeed better suited for the identification of this type of images. We observed, in average, a reduction of 29% in the recognition delay when using our algorithm as compared to JPEG.

8.2 Future Research

PIT by segmentation coding could benefit from the addition of a preprocessing stage in which the visibility of a region would be used to modify the transmission order. In particular, regions with a very small surface, like one or two pixels, could be merged to other regions or simply be transmitted much later.

In the same manner, the textured components should be extracted and represented using a different method. We suspect that the shape of a textured area might convey more information than the nature of the texture itself.

We must also point out that the overall performance of our algorithm is not very good when it comes to transmitting the full image, since it was designed only for fast identification. Our algorithm needs considerable overhead to transmit the very small regions found in the late transmission stages. This could possibly be solved by combining our algorithm with another, better suited to the task.

Finally, the application of a similar solution to chromatic images poses a few additional problems: how to calculate the gradient for a color image? How to calculate

an average value for colored regions? How to add two colored regions? These questions do not have satisfactory answers at this point.

The distortion measures of Chapter 7 should be refined further. The weighting given to each gained/lost pixel edge should reflect the edge intensity and vary with the viewing distance. Furthermore, the adaptive threshold of the gradient that produces the edge maps should be replaced by a procedure that matches more closely the performance of the human eye.

It would be interesting to see how these measures correlate with subjective scales like those of Table 4.1 and 4.2. In particular, to find a mapping of the two edge measurements to a single distortion measurement. We cannot say at this point if our approach can be used to compare different types of distortion together. On the other hand, we suspect that given a particular type of distortion, a mapping of the edge measurement to a subjective rating is easily obtainable.

Bibliography

[Ahm74] N. Ahmed, T. Natarjan and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, January 1974.

[Ant92] M. Antonini, P. Mathiew and I. Daubechies, "Image coding using wavelet transforms," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, April 1992.

[Arp94] R. B. Arps and T. M. Truong, "Comparison of international standards for lossless still image compression," *Proc. IEEE*, vol. 82, no. 6, pp. 889–899, June 1994.

[Beu90] S. Beucher, "Segmentation d'images et morphologie mathématique,"Ph. D. thesis, École nationale supérieure des mines de Paris, June 1990.

[Bur83] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as compact image code," *IEEE Trans. commun.*, vol. COM-31, pp. 532–540, April 1983.

[Cha94] H. K.-C. Chang and S.-H. Chen, "Progressive image transmission by dynamic binary thresholding," *Opt. Eng.*, vol. 33, no. 2, pp. 586–595, February 1994.

[Che84] W.-H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, no. 3, pp. 235–232, March 1984.

[Cle84] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Commun.*, vol. COM-32, no. 4, pp. 396–402, April 1984.

[Dan88] I. Daubechies, "Orthogonal bases of compactly supported wavelets," *Comm. Pure Applied Math.*, vol. XLI, no. 7, pp. 909–996, July 1988.

[DeN91] F. G. B. DeNatale, G. S. Desoli and D. D. Giusto, "Segmentation-based hybrid-coding of color images," ICASSP'91, pp. 2757–2760.

[Dou92] E. Dougherty, *An Introduction to Morphological Image Processing*, vol. TT9 of Tutorial Texts Series, SPIE Optical Engineering Press, Bellingham, 1992.

[Dro94] M. V. Droogenbroeck, "Traitement d'images numériques au moyen d'algorithmes utilisant la morphologie mathématique et la notion d'object: application au codage," Ph. D. thesis, Université catholique de Louvain et Ecole nationale supérieure des mines de Paris, Mai 1994.

[Fre61] H. Freeman. "On the encoding of arbitrary geometic configurations," *IRE Trans. Electon. Comput.*, vol. 10, pp. 260–268, June 1961.

[GIF89] CompuServe Incorporated, *Graphics Interchange Format(sm)*, version 89a, Columbus, Ohio, 1990.

[Gol66] S. W. Golomb, "Run-length encoding," *IEEE Trans. Informat. Theory*, vol. IT-12, no. 4, pp. 399–401, July 1966.

[Hal77] C. L. Hall and E. L. Hall, "A non-linear model for the spatial characteristics of the human visual system," *IEEE Trans. Sys. Man Cybern.*, vol. SMC-7, no. 3, pp. 161–170, March 1977.

[Har93] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1, Addison-Wesley, 1993.

[How94] P.G. Howard and J. S. Vitter, "Arithmetic coding for data compression," *Proc. IEEE*, vol. 82, no. 6, pp. 857–865, June 1994

[Huf52] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, pp. 1088–1101, September 1952.

[JBI91] CCITT Draft recommendation T.82 ISO/IEC Commitee Draft 11544, "Coded Representation of Picture and Audio Information-Progressive Bi-level Image Compression," WG9-S1R4.1, September 1991.

[Jai74] A. K. Jain, " A fast Karhunen-Loeve transform for finite discrete images," *Proc. Nation. Electon. Conf.*, Chicago, Illinois, pp 322–328, October 1974.

[Jai89] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Eaglewood Cliffs, New Jersey.

[Jay84] N. S. Jayant and P. Noll, *Digital Coding of waveforms Principles and Applications to Speech and Video*, Prentice-Hall, New Jersey, 1984.

[Kno80] K. Knowlton, "Progressive transmission of grey-scale and binary pictures by simple, efficient, and lossless encoding schemes," *Proc. IEEE*, vol. 68, no. 7, July 1980.

[Kun85] M. Kunt, A. Ikonomopoulos and M. Kocher. "Second generation image coding techniques," *Proc. IEEE*, vol. 73, no. 4, April 1985.

[Kun87] M. Kunt, M. Bénard and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans.Circ.Syst.*, vol. 34, nol. 10, November 1987.

[Lew92] A. S. Lewis and G. Knowles, "Image compression using 2-D wavelet transforms," *IEEE Trans. Image Process.*, vol. 1, no. 2, April 1992.

[Lan84] G.G. Langdon, "An introduction to arithmetic coding," *IBM Journ. Res. Dev.*, vol. 28, pp. 135–149, March 1984.

[Lim79] J. O. Limb, "Distosion criterion of the human viewer," *IEEE Trans. Sys. Man Cybern.*, vol. SMC-9, no. 12, pp. 778–793, December 1979.

[Lin80] Y. Linde, A Baza, R. Gray, "An algoritm for vector quantizer design," *IEEE Trans. Commun.*, pp. 84-95, vol. COM-28, no. 1, January 1980.

[Lu91] C.-C. Lu and J. Dunham, "Highly efficient coding schemed for contour lines based on chain code representation," *IEEE Trans. Commun.*, vol. 39, no. 10, October 1991.

[Mar76] J. D. Markel and A. H. Gray Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.

[Mar86] P. Maragos and R Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoust.Speech Signal Process.*, vol 34, no. 5, pp. 1228–1244, October 1986.

[Mil85] M. I. Mills and P. J. Hearty, "Viewer evaluation of simulated teleshopping presentations: effects of image complexity and imagecompletion time," *Human Factors*, vol. 27, no. 4, pp. 447–483, August 1985.

[Min86] T. Minami and K. Shinohara, "Encoding of line drawings with a multiple grid chain code," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 8, no. 2, pp. 269–276, March 1986.

[Miy88] M. Miyahara, "Quality assessments for visual services," *IEEE Commun. Mag.* vol. 26, no 10, pp. 51-60, October 1988.

[Nas88] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Trans. Commun.*, vol COM-36, no. 8, pp. 957–971, August 1988.

[Net77] A. N. Netravali and B. Prasada, "Adaptive quantization of picture signals using spatial masking," *Proc. IEEE*, vol. 65, no. 4, pp. 536–548, April 1977.

[One76] J. B. O'neal, Jr., "Differential pulse-code modulation (DPCM) with entropy coding," *IEEE Trans. Informat. Theory*, vol. IT-21, no. 2, pp. 169–174, March 1976.

[Pap91] A. Papoulis, *Probability, Random Variables, and Stochastich Processes*, third edition, McGraw-Hill, New York, 1991.

[Pav77] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, 1977.

[Pea78] W. A. Pearlman, "A visual model and a new distorsion measure in the context of image processing," *J. Opt. Soc. Am.*, vol. 68, no. 3, pp. 374–386, March 1978.

[Pro89] J. G. Proatis, *Digital Communication*, second edition, McGraw-Hill, New York, 1989.

[Rab89] S. Rabani, *Image Coding and Compression,* SPIE Optical Engineering Press, vol. MS-48, Bellingham, 1989.

[Rio91] O. Rioul and. Vetterli, "Wavelets and signal processing," *IEEE Signal Process. Mag.*, vol. 8, no. 4, pp. 14–38, October 1991.

[Ris79] J. J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM J. Res. Develop.*, vol. 23, no. 2, pp. 149–169, 1979.

[Roo94] P. Roos, M. A. Viergever, M. C. A. Van Dijke and J. M. Peters, "Reversible Intraframe compression of Medical Images," IEEE Trans. on Med. Im., vol. 7, no. 4, pp. 328–336, December 1988.

[Sag89] J. A. Saghri, P. S. Cheatham and A. Habibi, "Image quality measure based on a human visual system model," *SPIE Opt. Eng.*, vol. 28, no.7, pp. 813–818, July 1989.

[Sak77] D. J. Sakrison, "On the role of the observer and a distortion measure in image transmission," *IEEE Trans. Commun.*, vol. COM-25, no. 11, pp. 1251–1267, April 1977.

[Sch66] J. W. Schartz and R. C. Barker, "Bit-plane encoding: a technique for source encoding," *IEEE Trans. Aerosp. Electron. Syst. Mag.*, vol. AES-2, no. 4, pp. 385–392, July 1966.

[Ser82] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.

[Sha48] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Sys. Tech. J.*, pp. 379–423 (part I), pp. 623–656 (part II), 1948.

[Slo79] K.R. Sloan. Jr. and S. L. Tanimoto, "Progressive refinement of raster images," *IEEE Trans. Comput.*, vol. C-28, no. 11, pp. 871–374, November 1979.

[Soi90] P. Soille and L. Vincent, "Determining watersheds in digital pictures via flooding simulations," In *Visual Communications and Image Processing*, vol. 1360, pp. 240–250, SPIE, Lausanne, October 1990.

[Str91] P. Strobach, "Quadtree-structured recursive plane decomposition coding of images," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 39, no. 6, pp. 1380–1397, June 1991.

[Tzo84] K.-H. Tzou, T. R. Hsing and J. G. Dunham, "Applications of physiological human visual system model to image compression," *Applications of Digital Processing VII, Proc.*, vol. 504, pp. 419-424, 1984.

[Tzo87] K.-H. Tzou, "Progressive Image Transmission: a review and comparison of techniques," *Opt. Eng.*, vol. 26, no. 7, pp. 581–589, July 1987.

[Wal91] G. Wallace. "The JPEG Still Compression Standard, *Commun. ACM*, vol. 34, no. 4, April 1991.

[Wan86] L. Wang and M. Goldberg, "Lossless progressive image transmission by residual vector quantization," *Proc. 13th Biennnial Symposium on. Communications*, pp. c.1.9–c.1.12, Kingston, Ont., June 1986.

[Wan89] L. Wang and M. Goldberg, "Progressive image transmission using vector quantization on image in pyramid form," *IEEE Trans. Commun.*, vol. COM-37, no. 12, pp. 1339–1349, December 1989.

[Wit87] I. H. Witten, R. M. Ned, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Comm. ACM*, vol. 30, no. 6, pp. 520–540, June 1987.

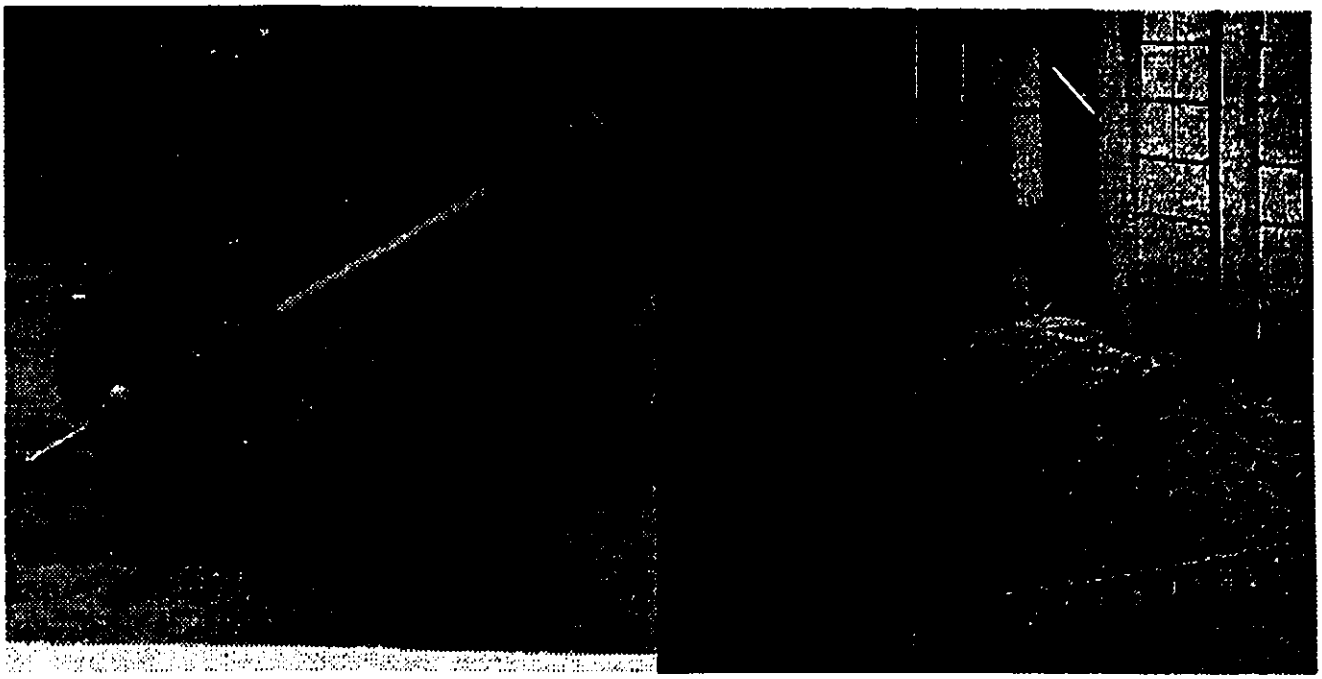
[Wol91] S. Wolf, C. A. Dvorak, R. F. Kubichek, C. R. South, R. A. Schaphorst and S. D. Voran, "How will we rate telecommunications system performance?" *IEEE Commun. Mag.*, no. 10, pp. 23–29, October 1991.

[Ziv78] J. Ziv and A. Lempel. "Compression of individual sequences via variable rate coding," *IEEE Trans. Informat. Theory*, vol. IT-24, no. 5, pp. 530–536, September 1978.

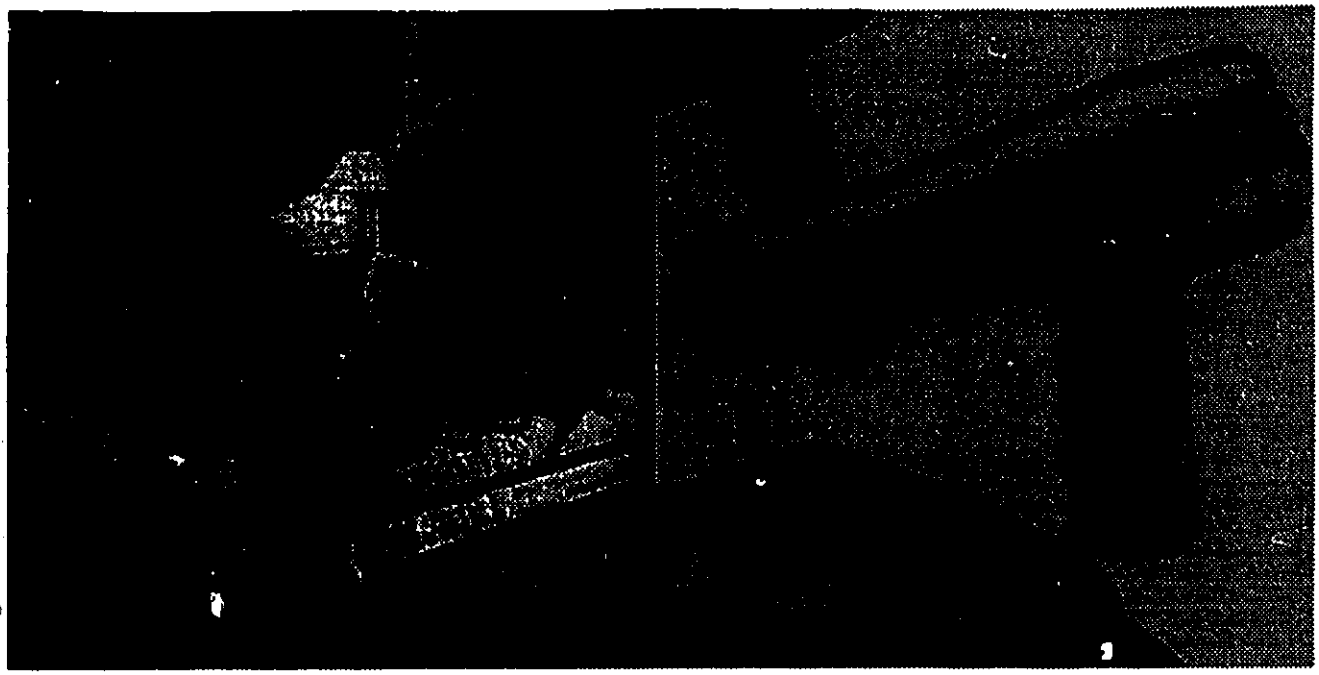
[Zor94] S. F. Zornetzer, An Introduction to Neural and Electronic Networks, 2nd edition, San Diego, Academic Press, 1994.

Appendix A

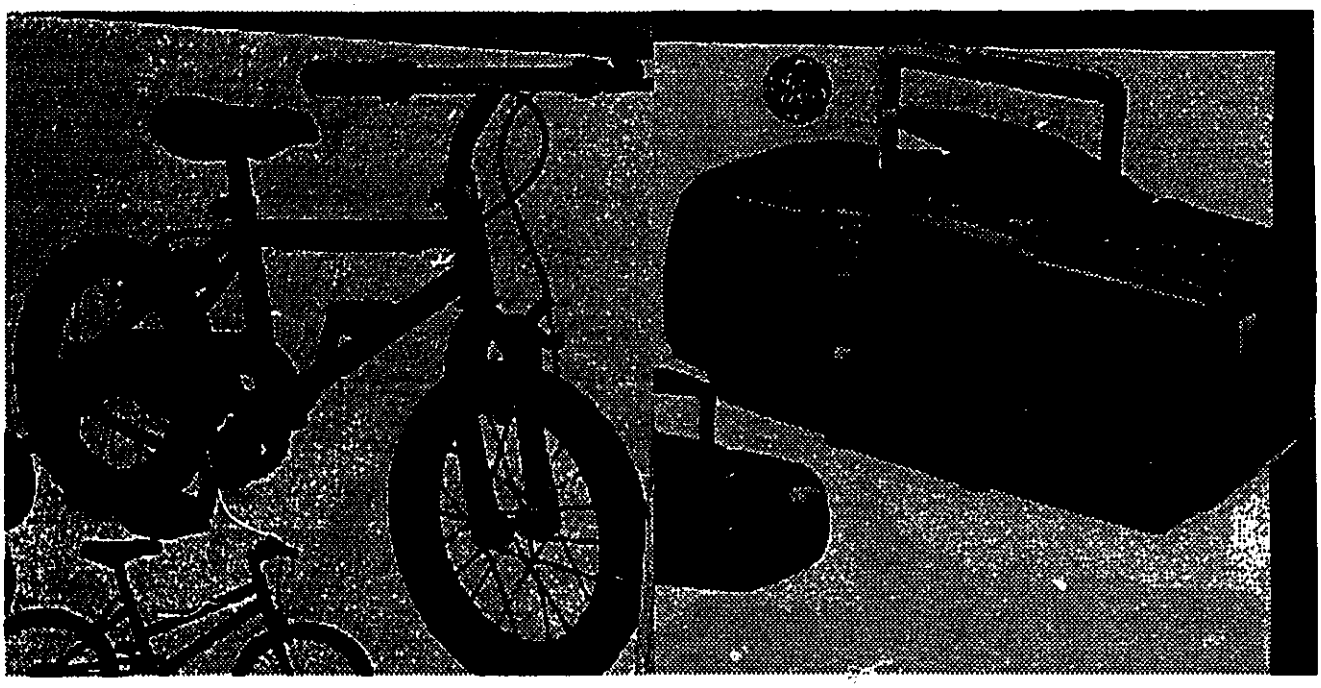
Image Data Set



barbell bed



bed2 blower



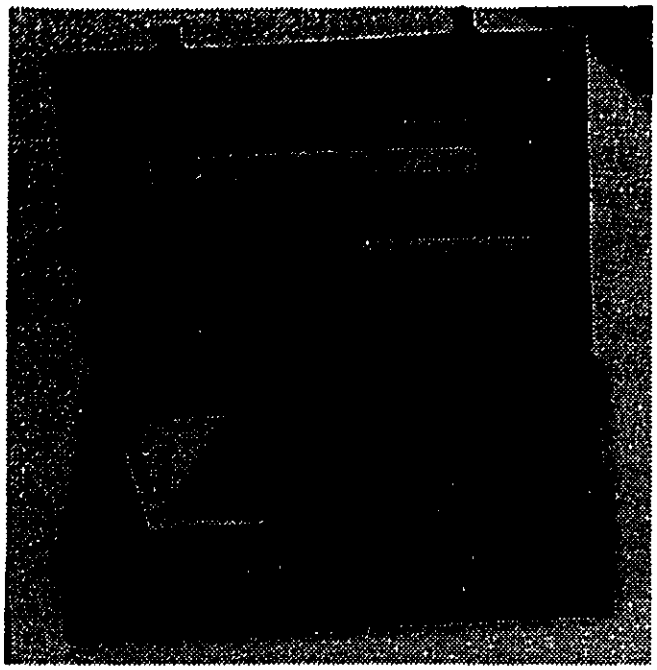
bike boombox

boot

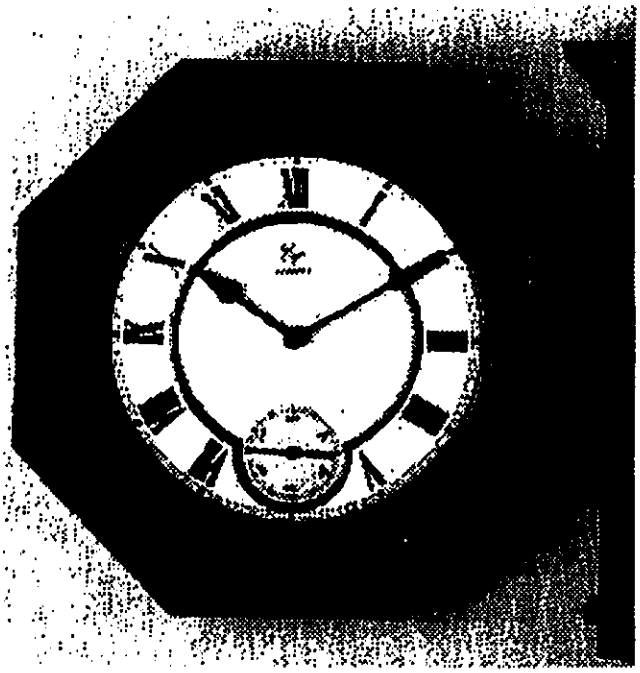
calculator

briefcase

camera

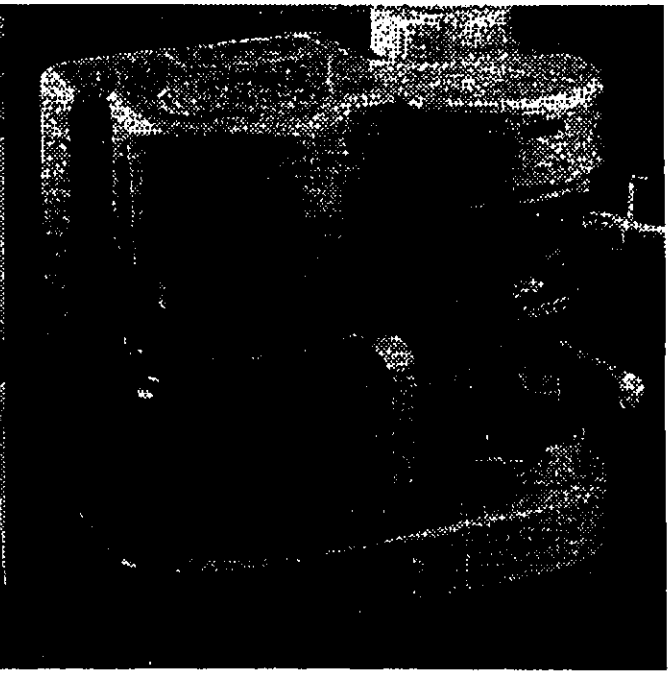


case

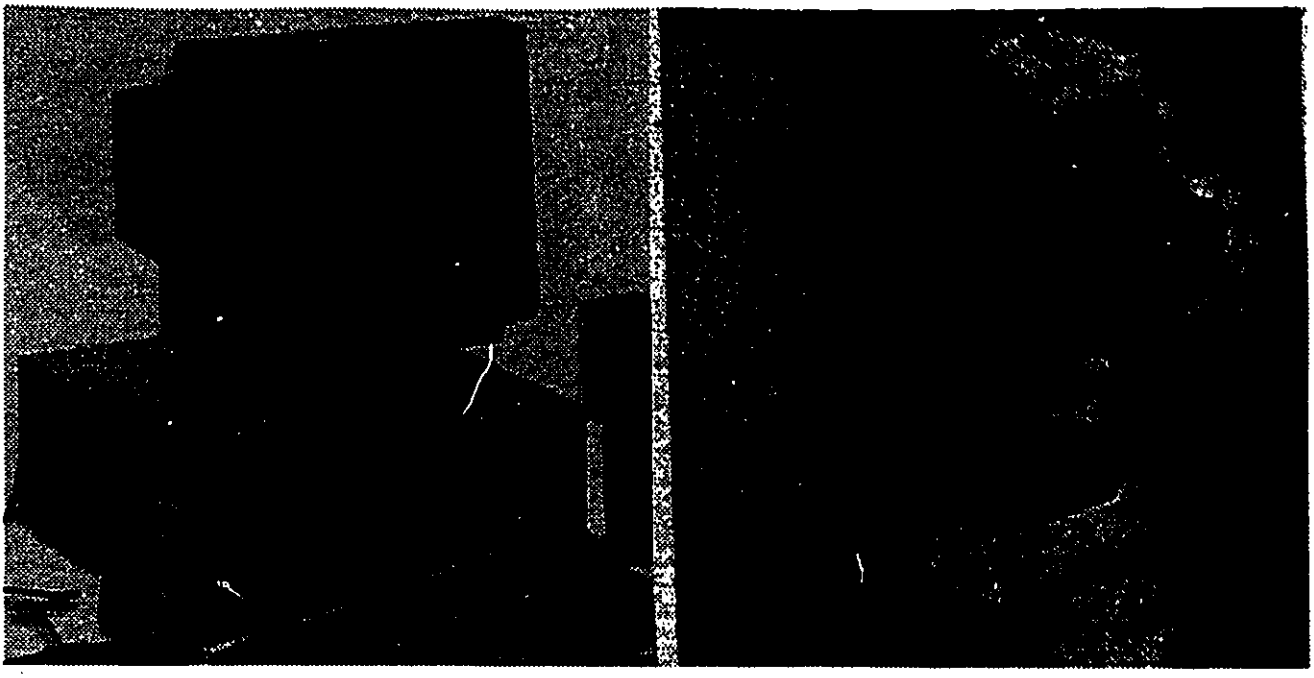


clock

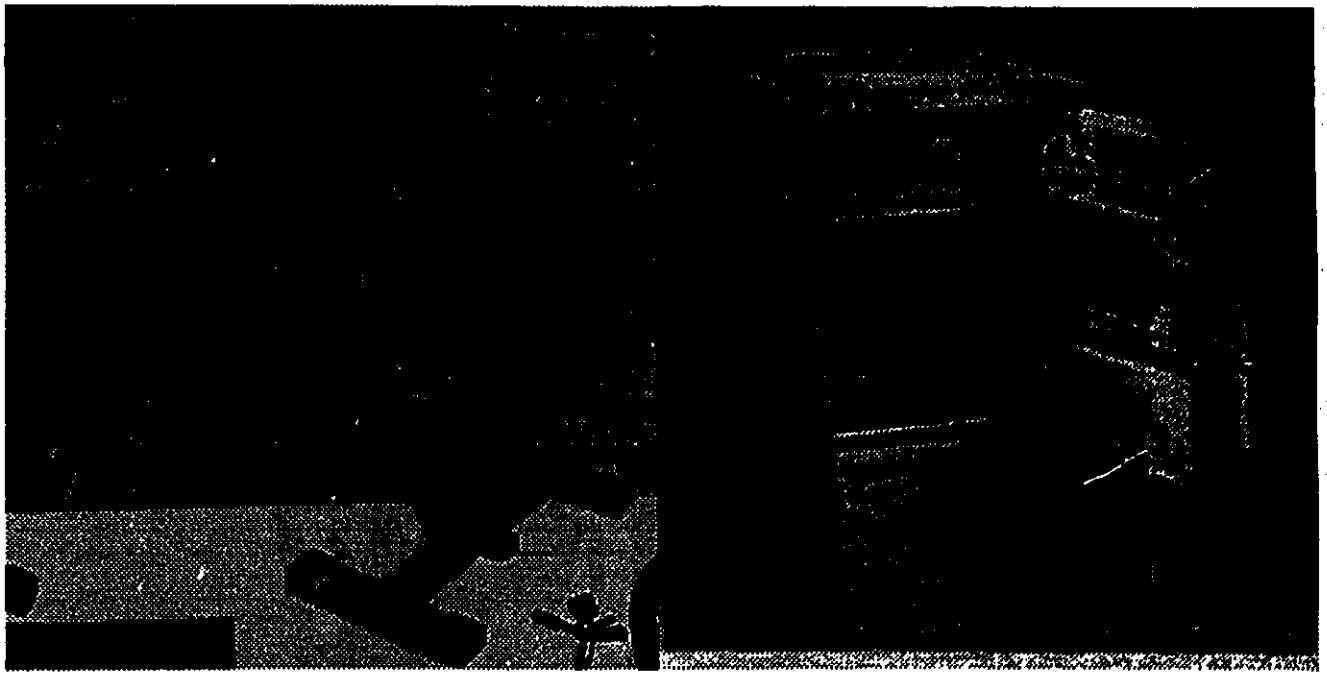
chair



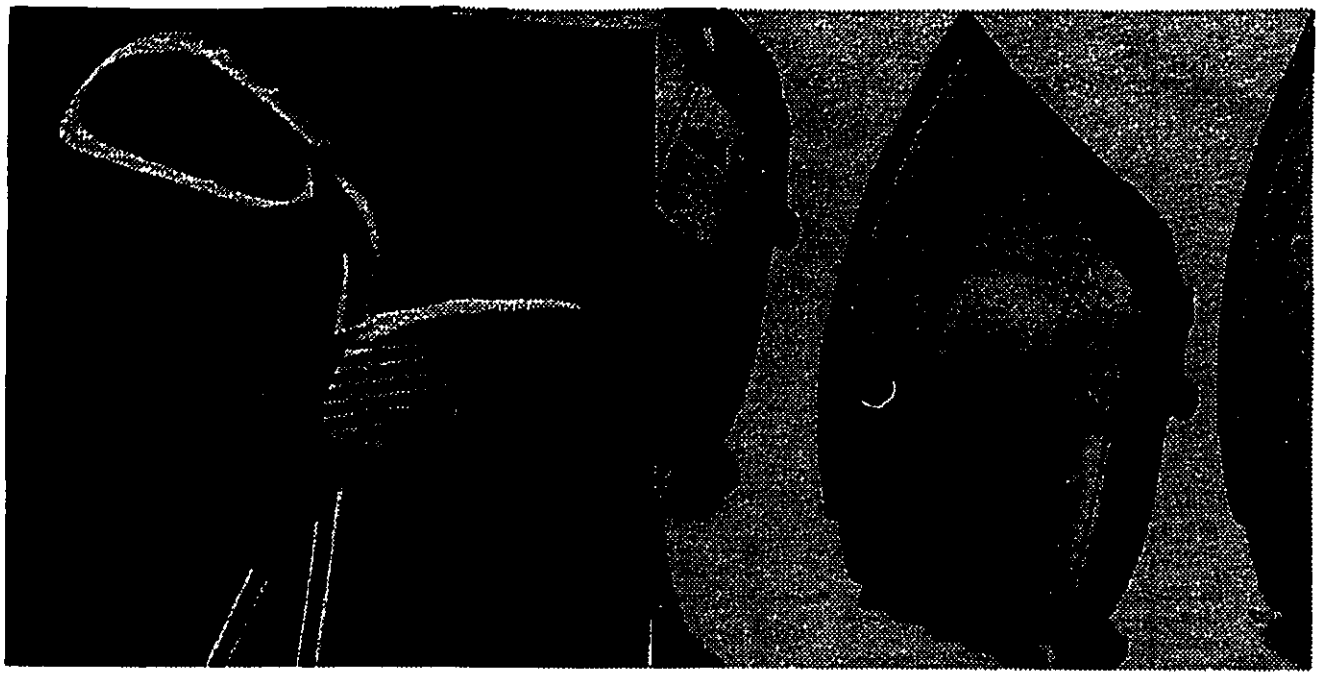
coffeemaker



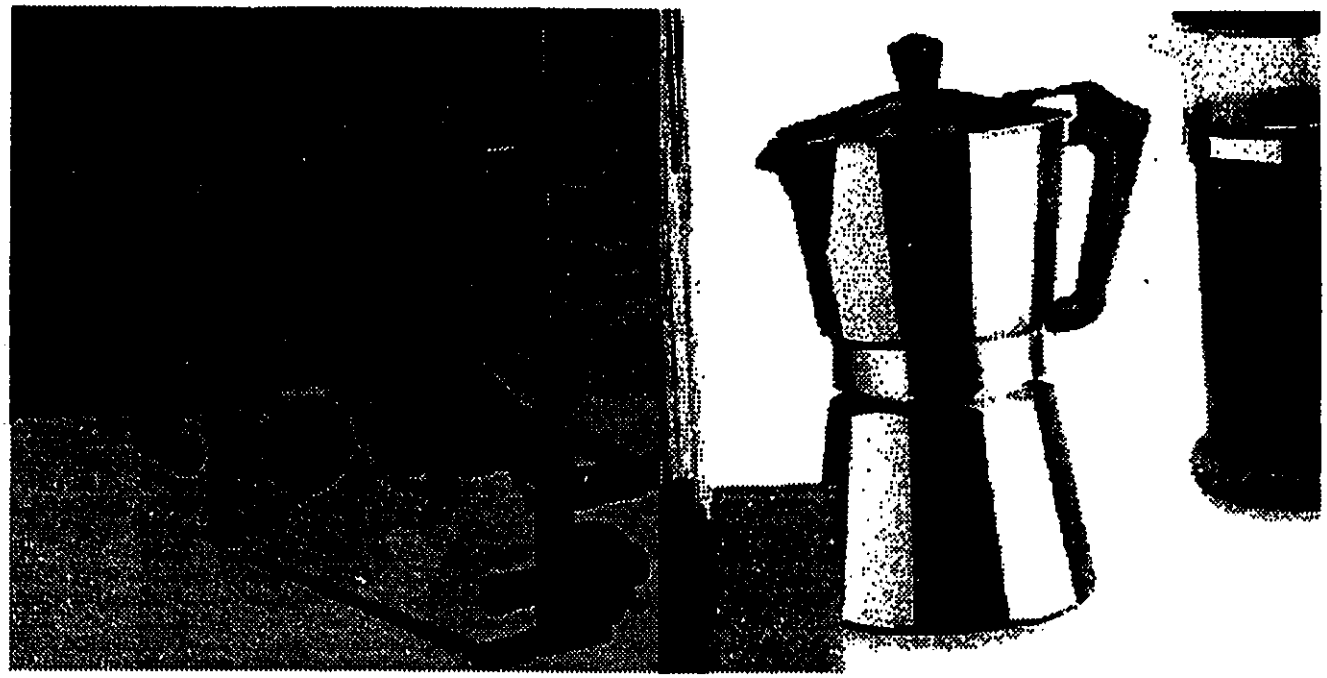
computer dog



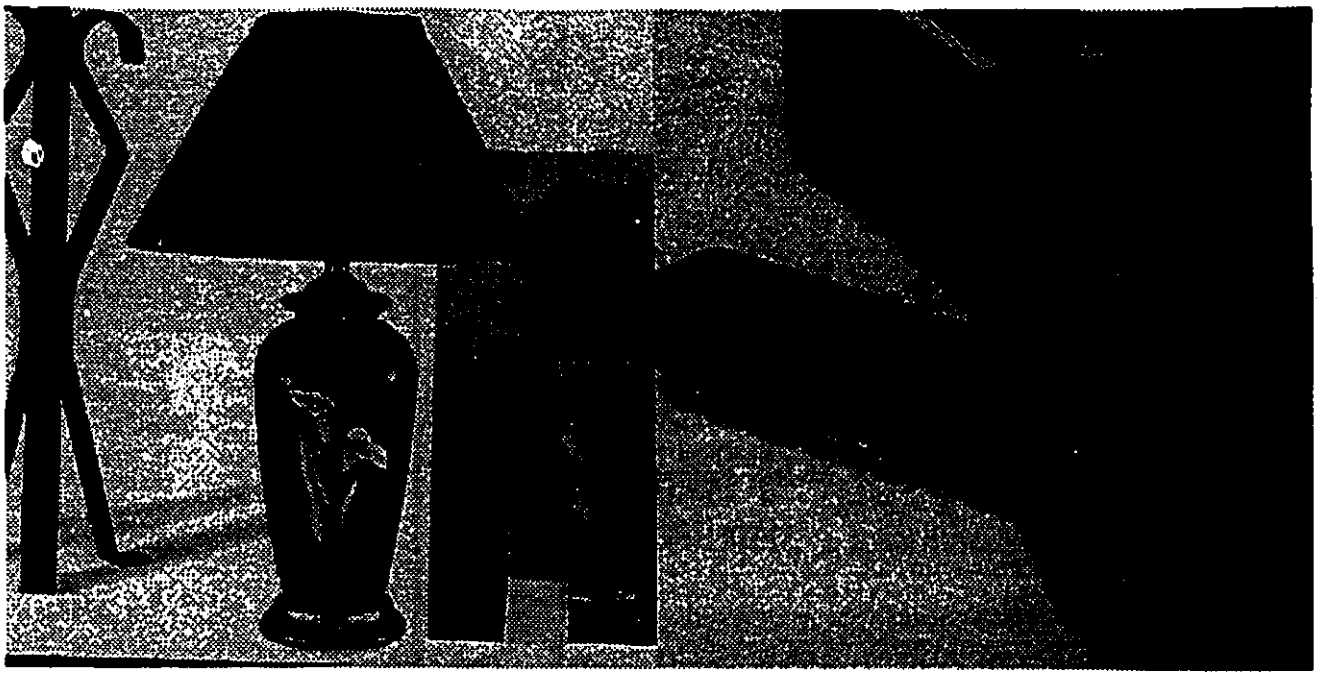
cycle fridge



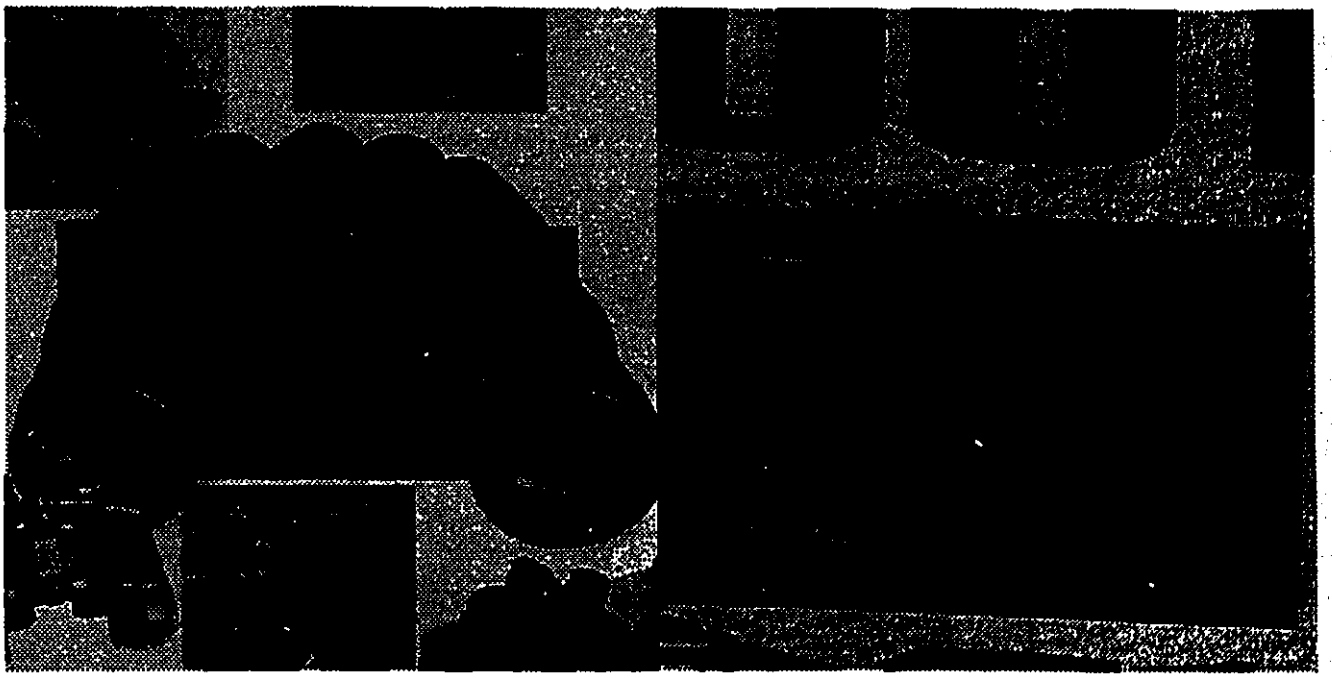
golf iron



gym kfmaker

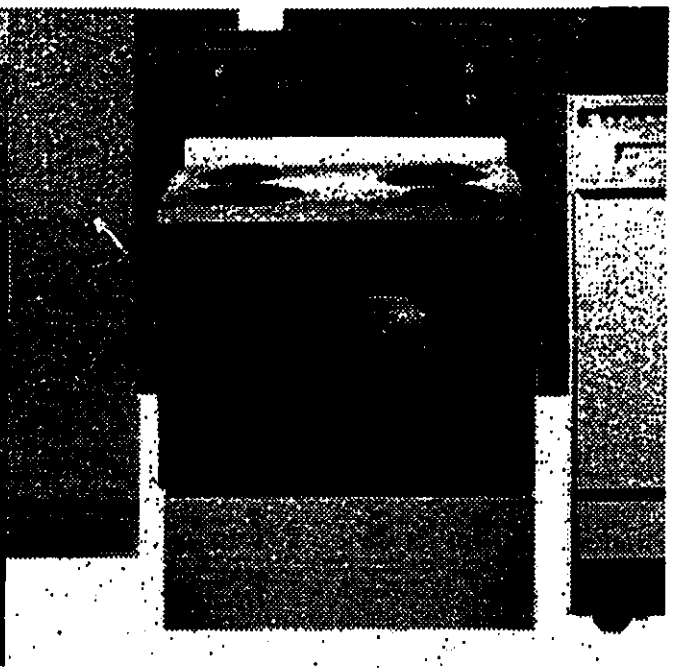


lamp light

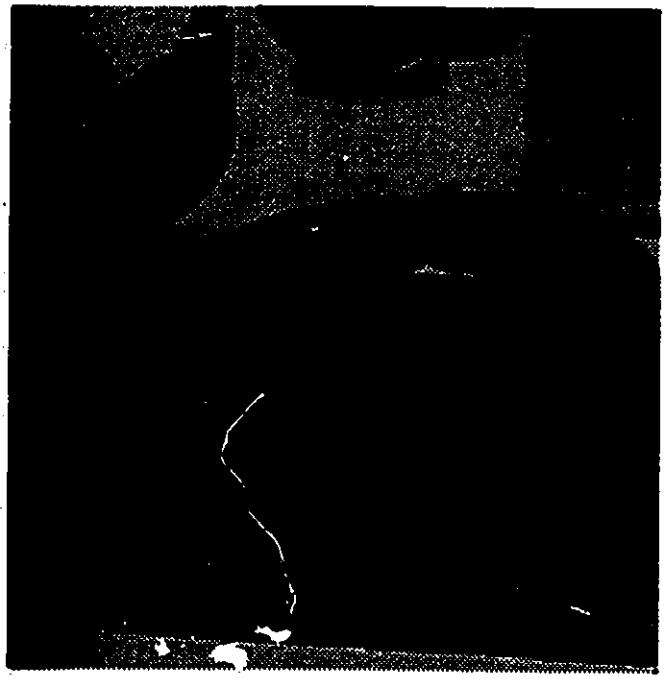


lens micro

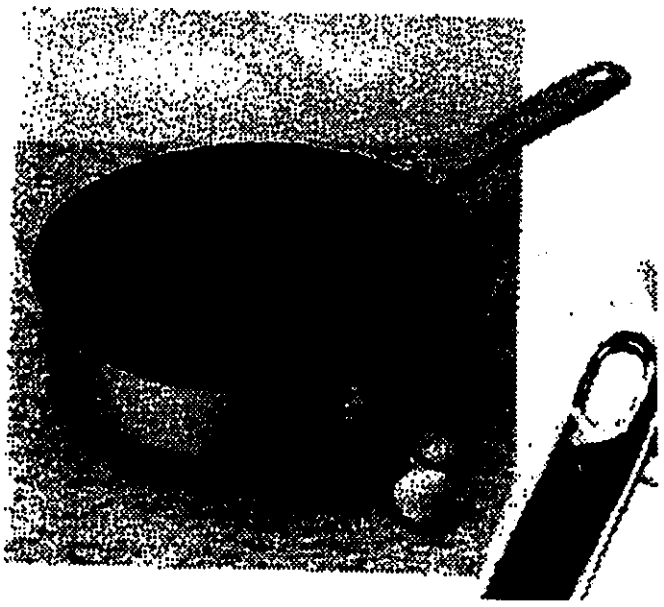
mower



oven

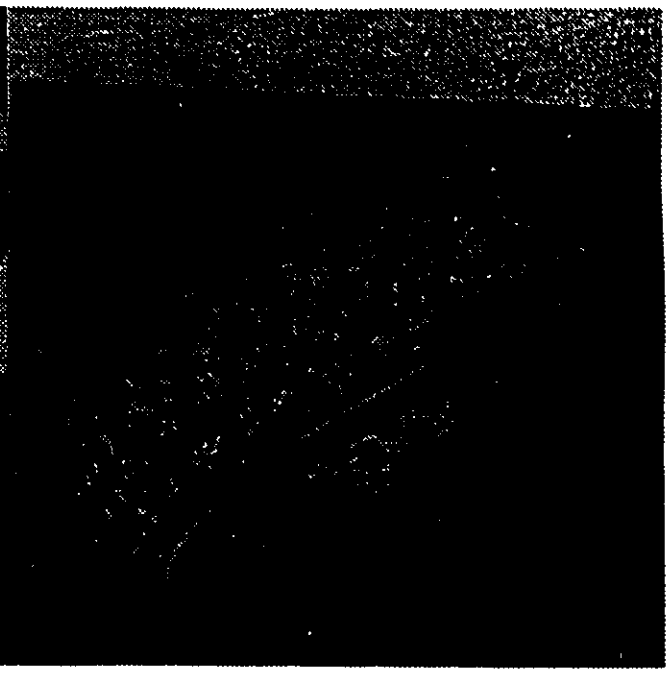


muff

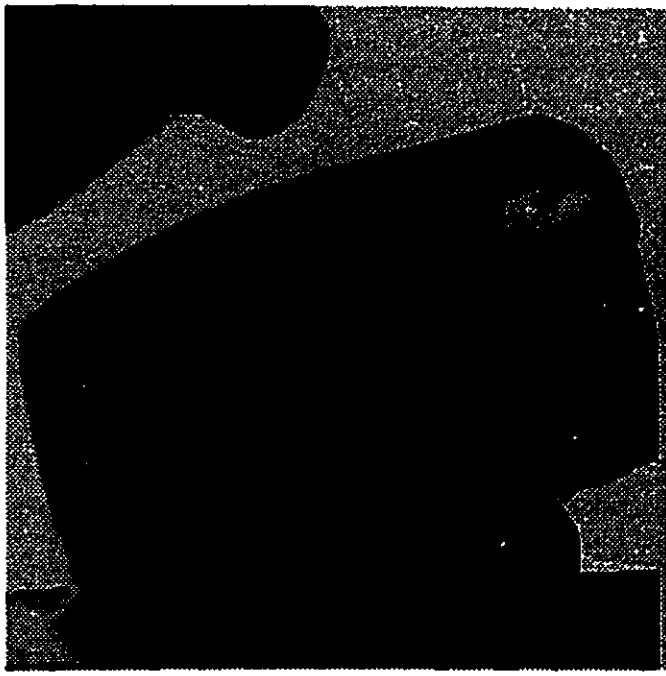


pan

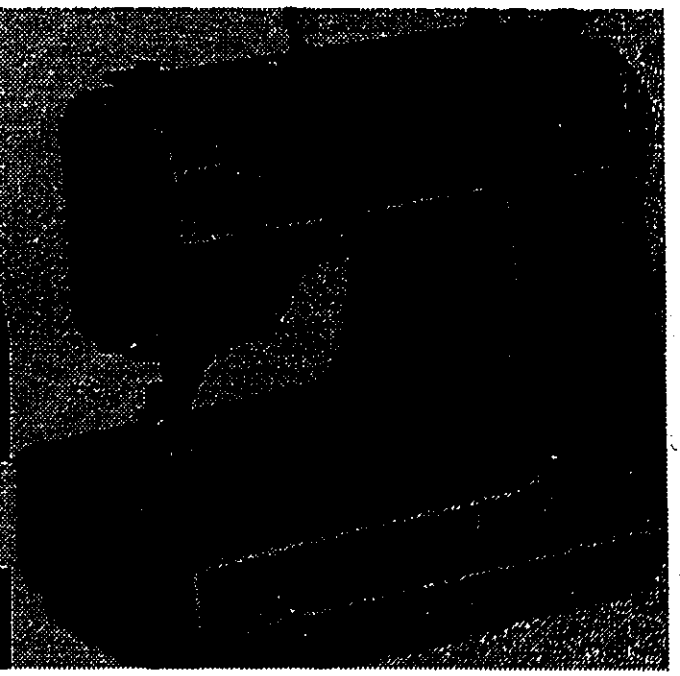
phone



ring

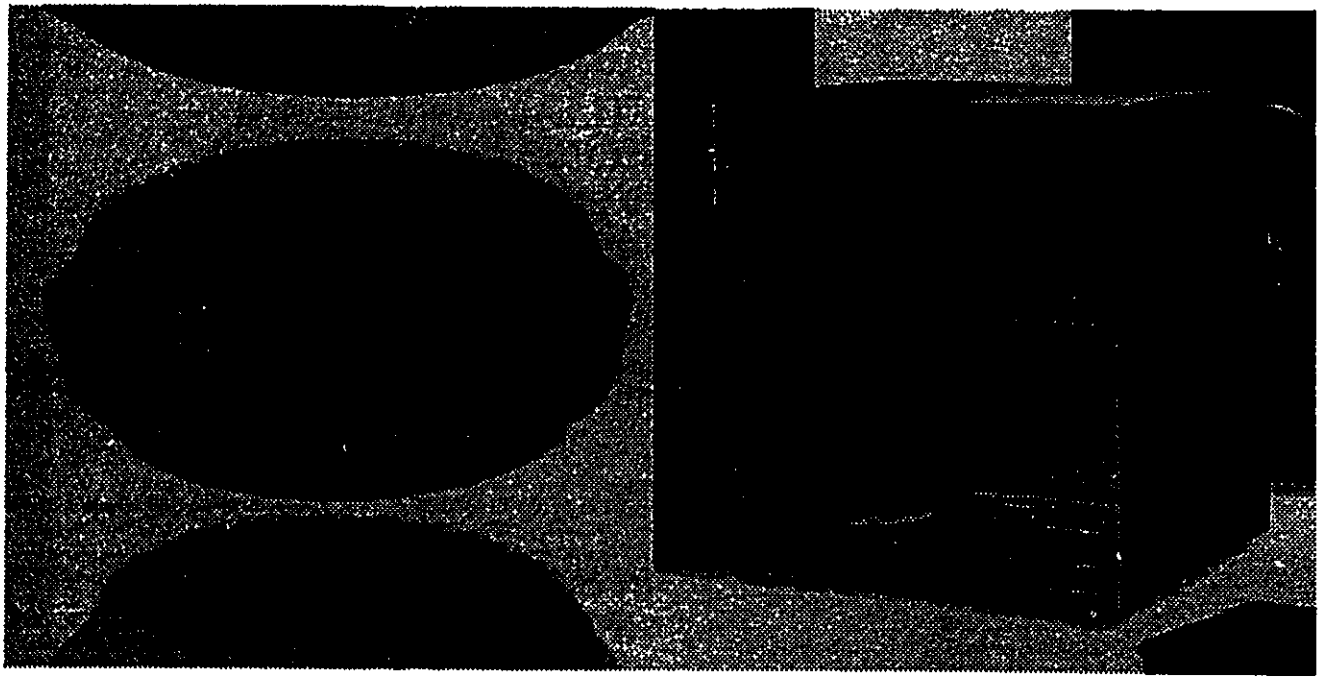


photo

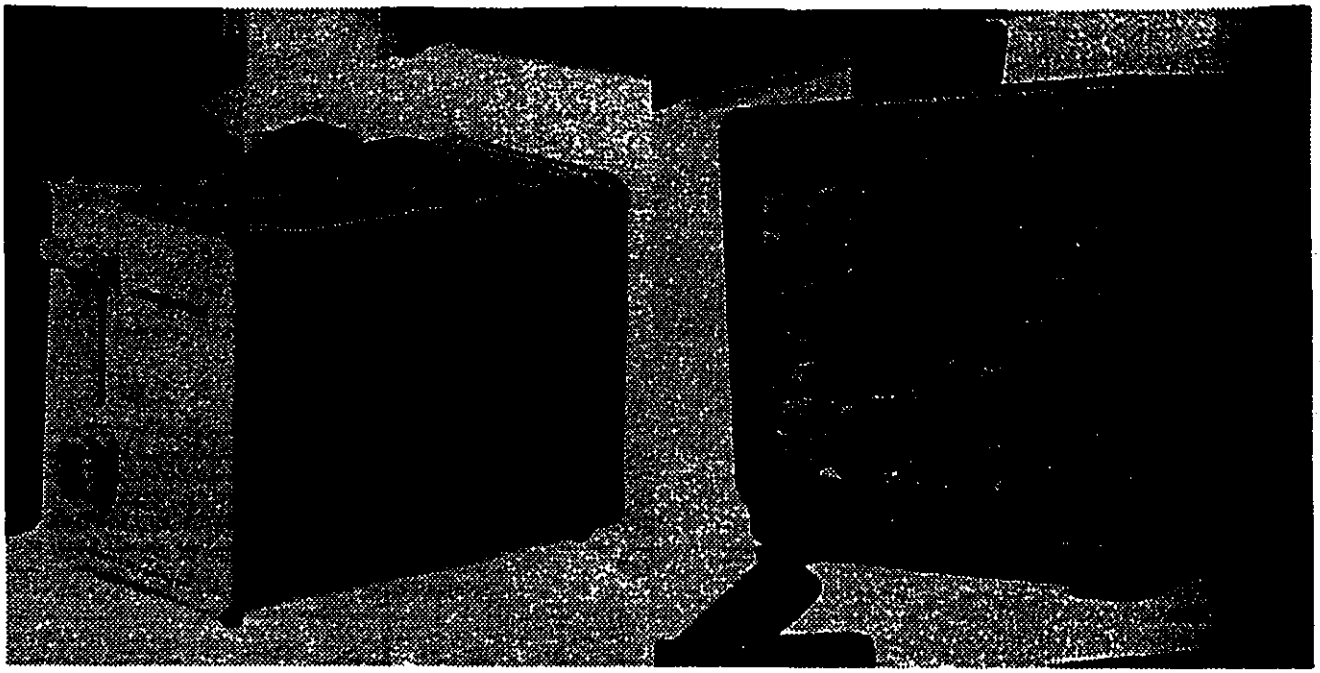


sewing

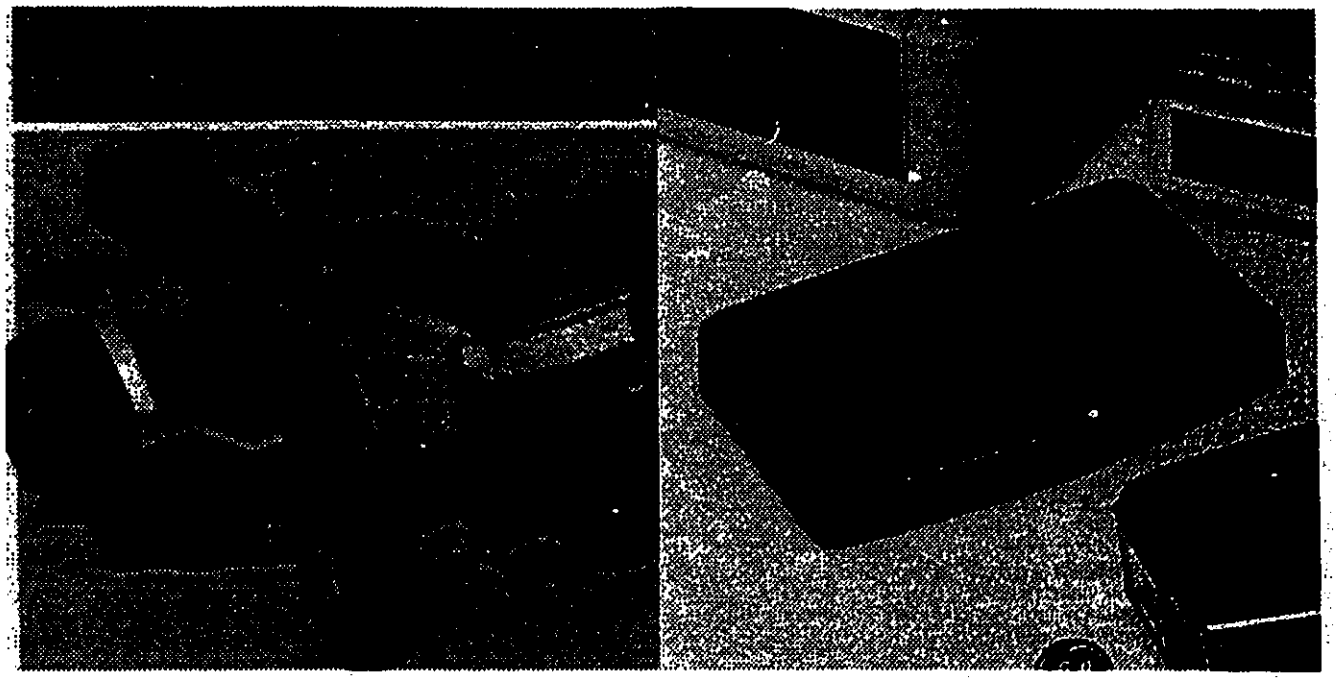
shaver soccer



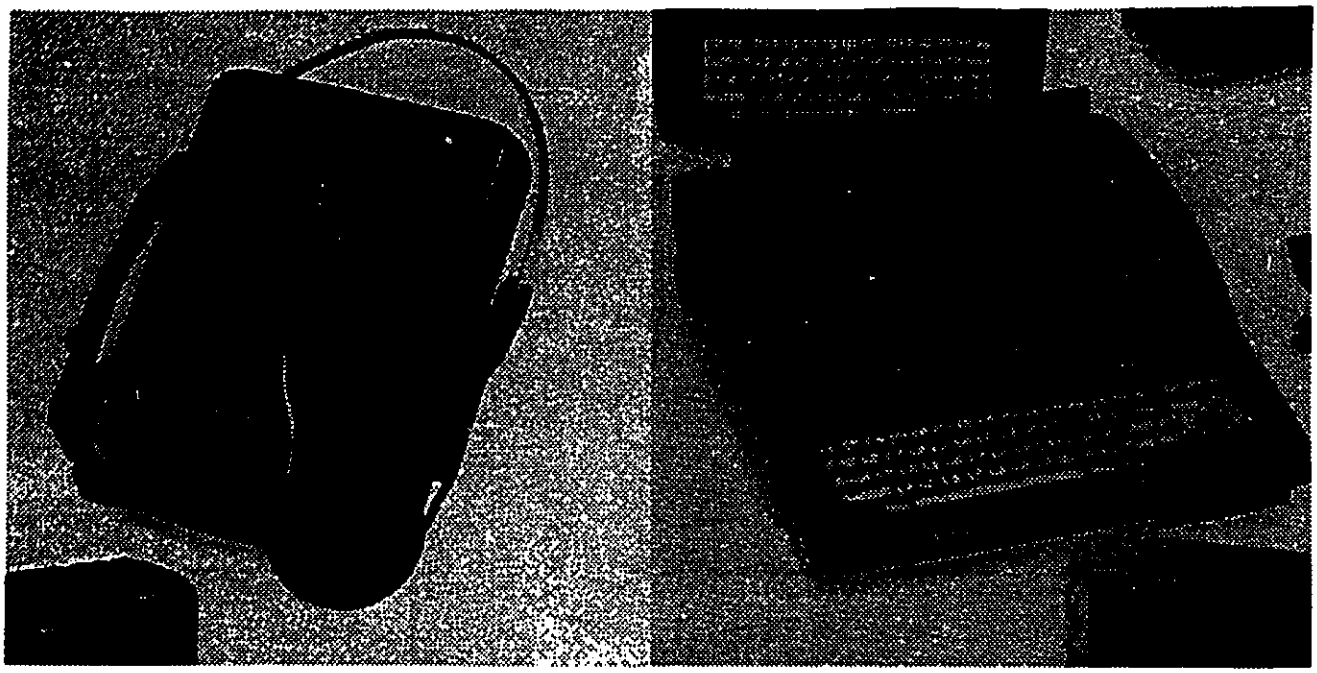
smoke stereo



toaster tv

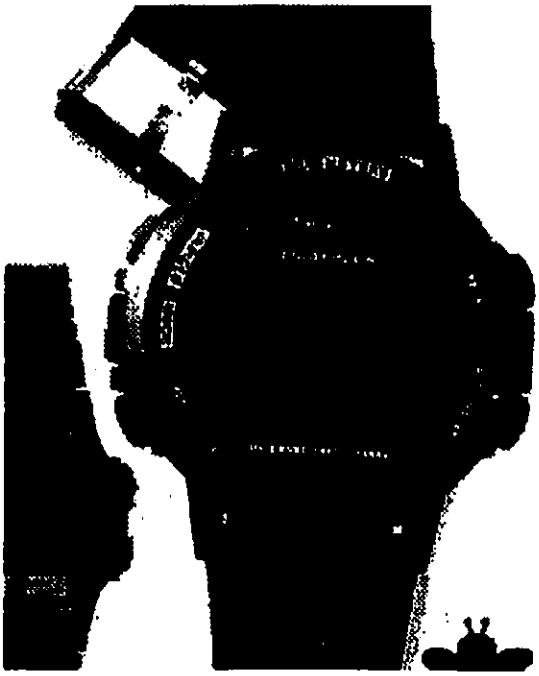


tractor wakeup



walkman

writer

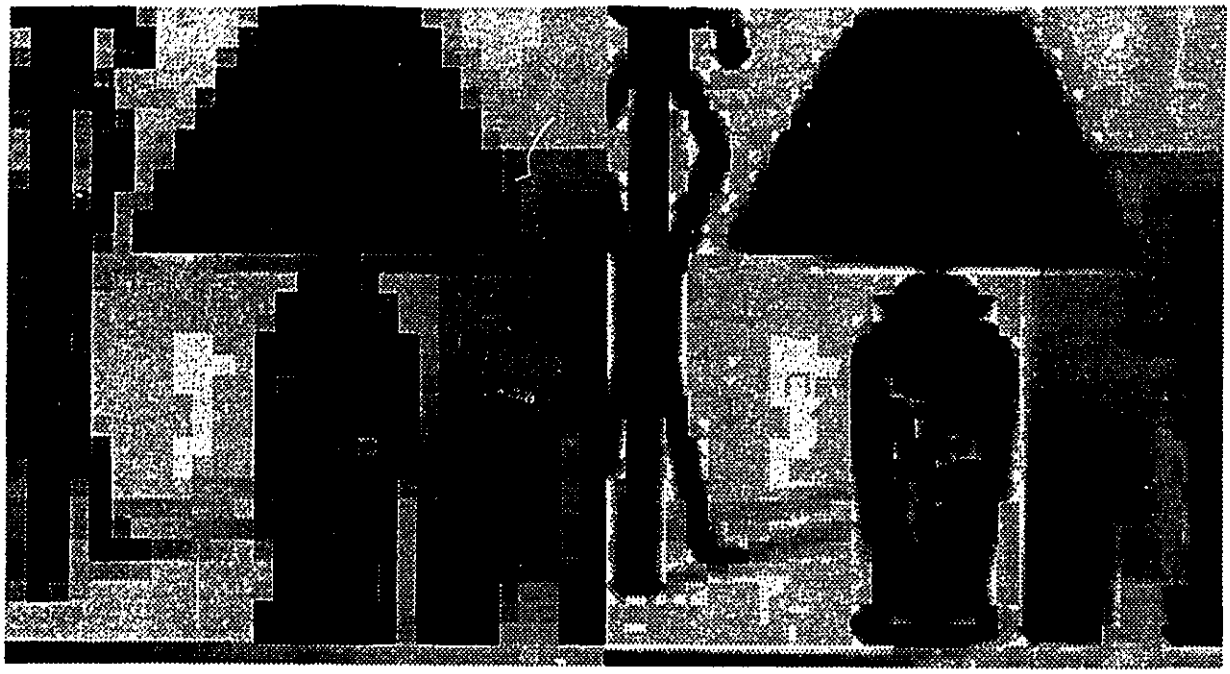


watch

Appendix B

Partial Image Sequences

This appendix presents 3 series of four images at different stages of transmission for both JPEG and segmentation based coding. This is intended to give some references to the reader concerning the perceived quality of the encoded image. The compression rates are expressed in bits per pixels.



0.099 bpp 0.271 bpp

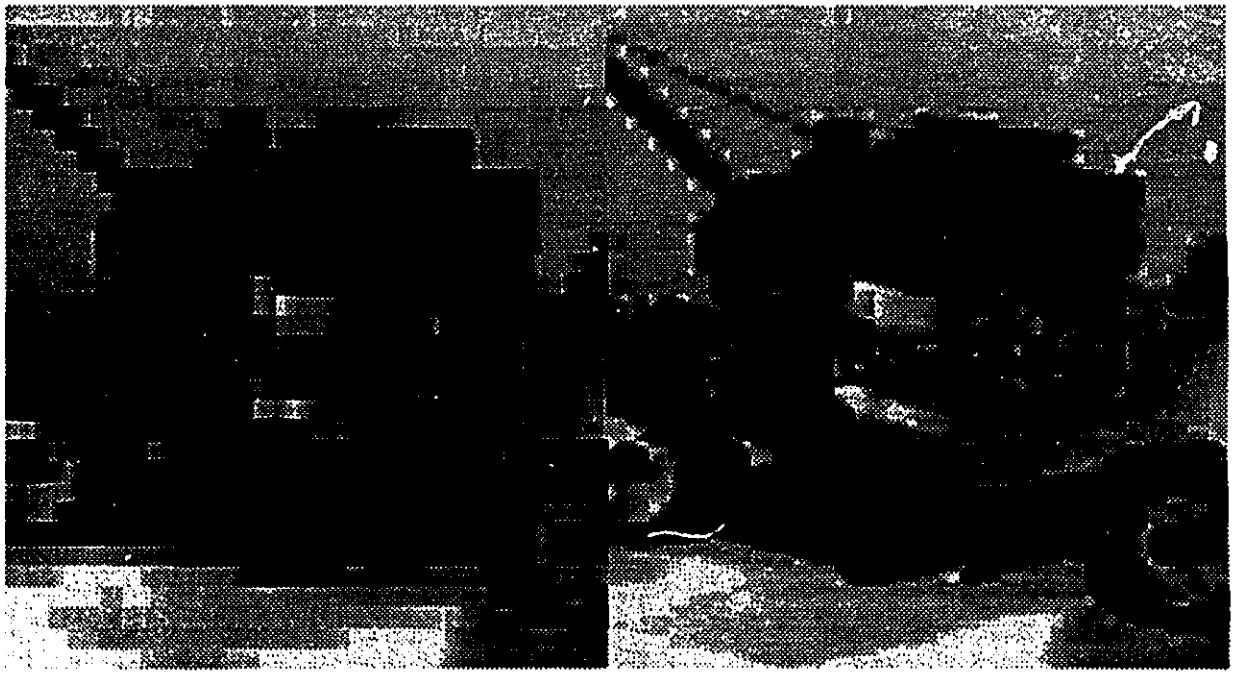
0.189 bpp 0.338 bpp

0.107 bpp 0.226 bpp

0.147 bpp 0.419 bpp

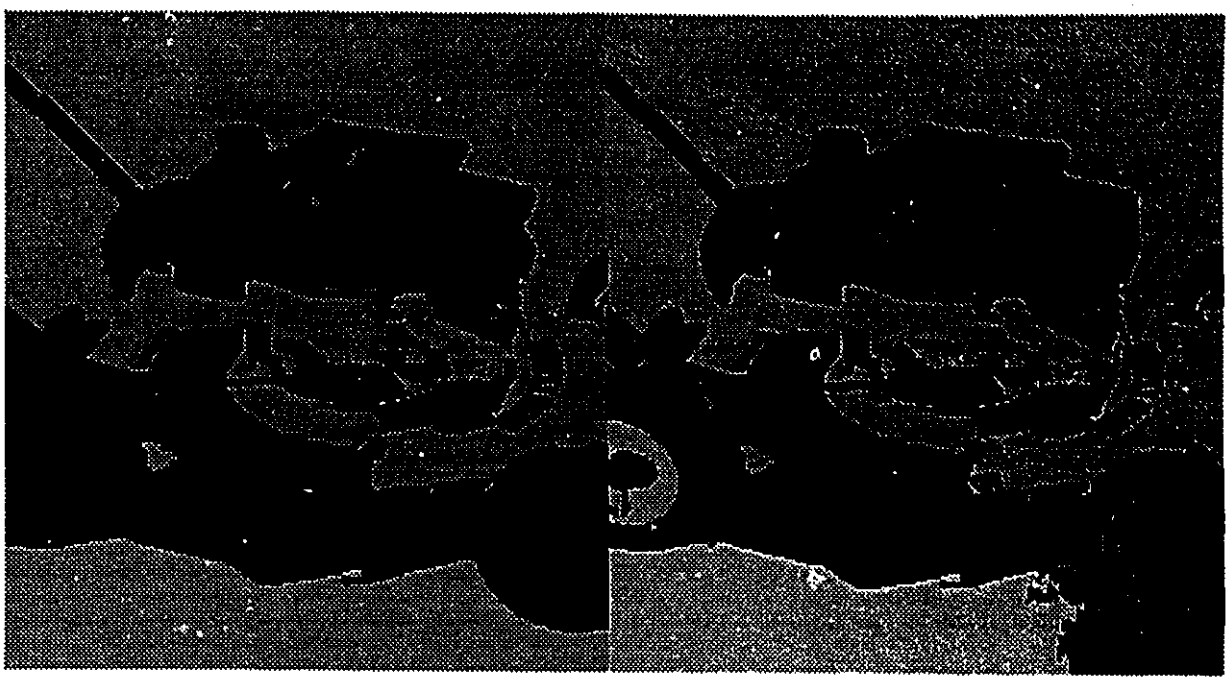
0.093 bpp

0.268 bpp

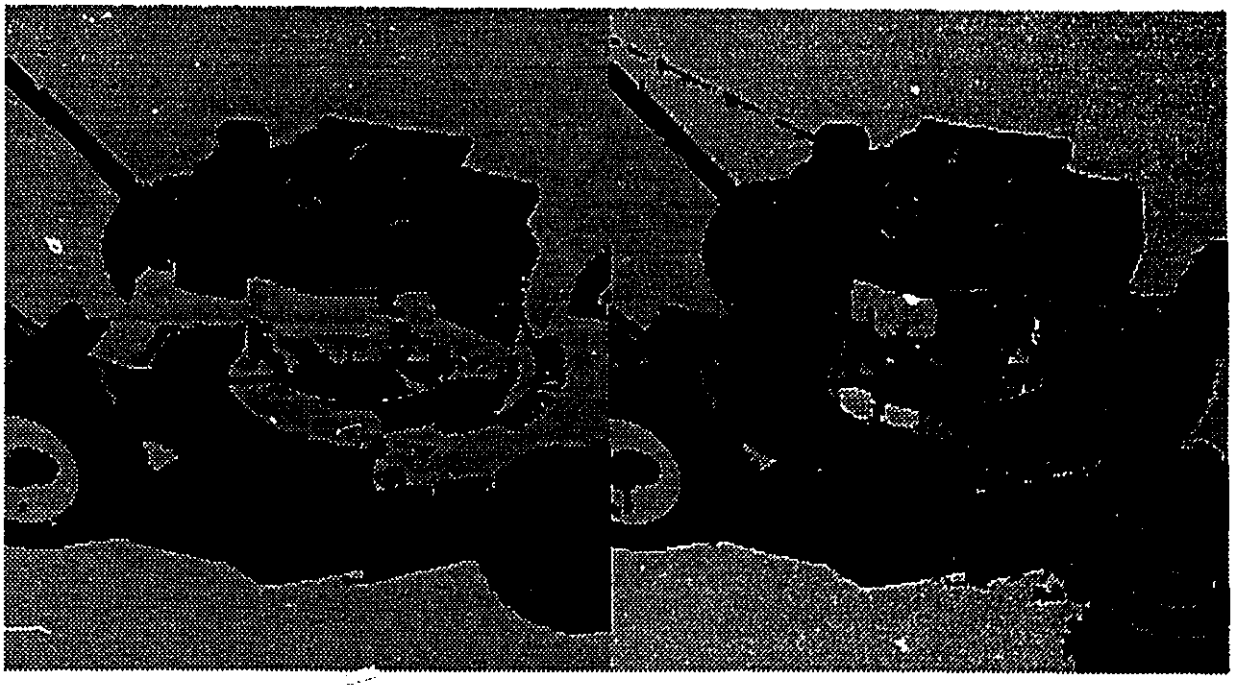


0.178 bpp

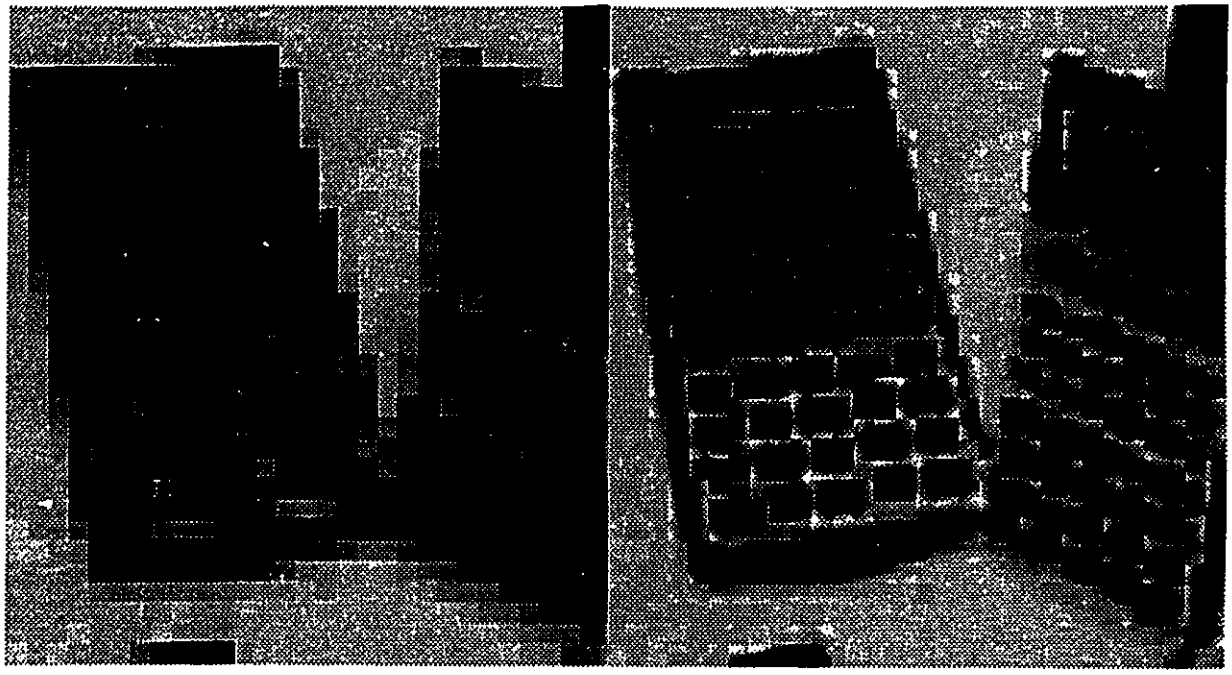
0.340 bpp



0.098 bpp 0.219 bpp

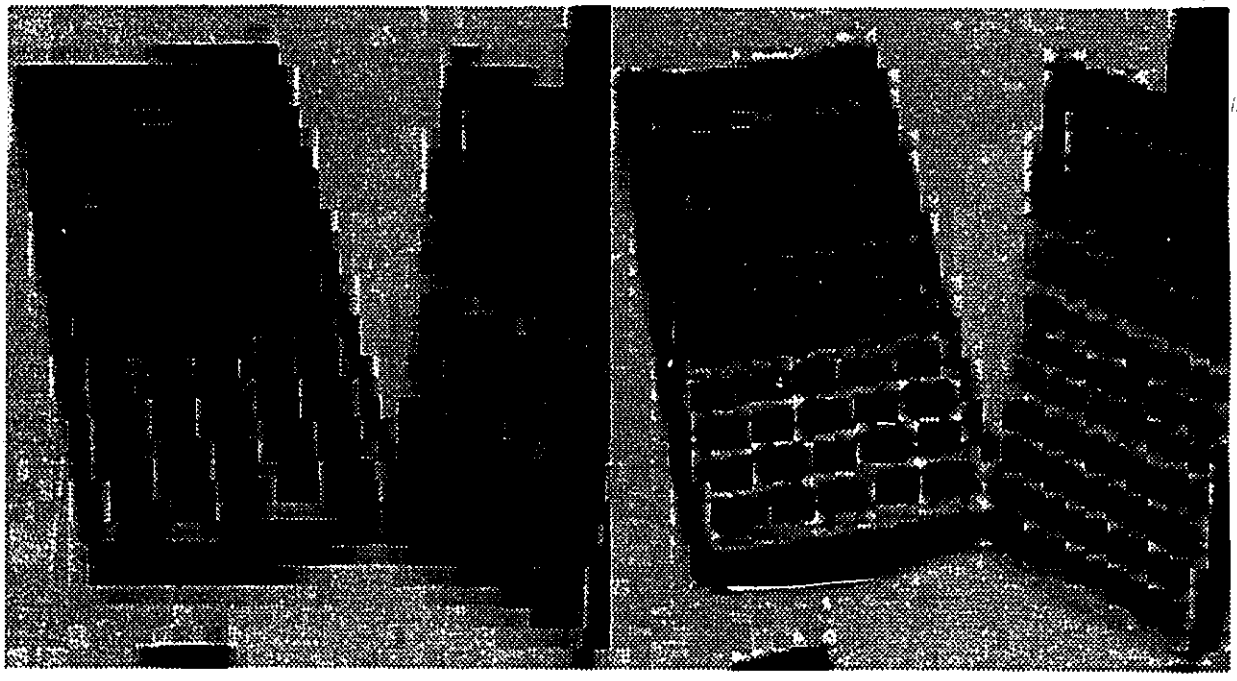


0.123 bpp 0.422 bpp



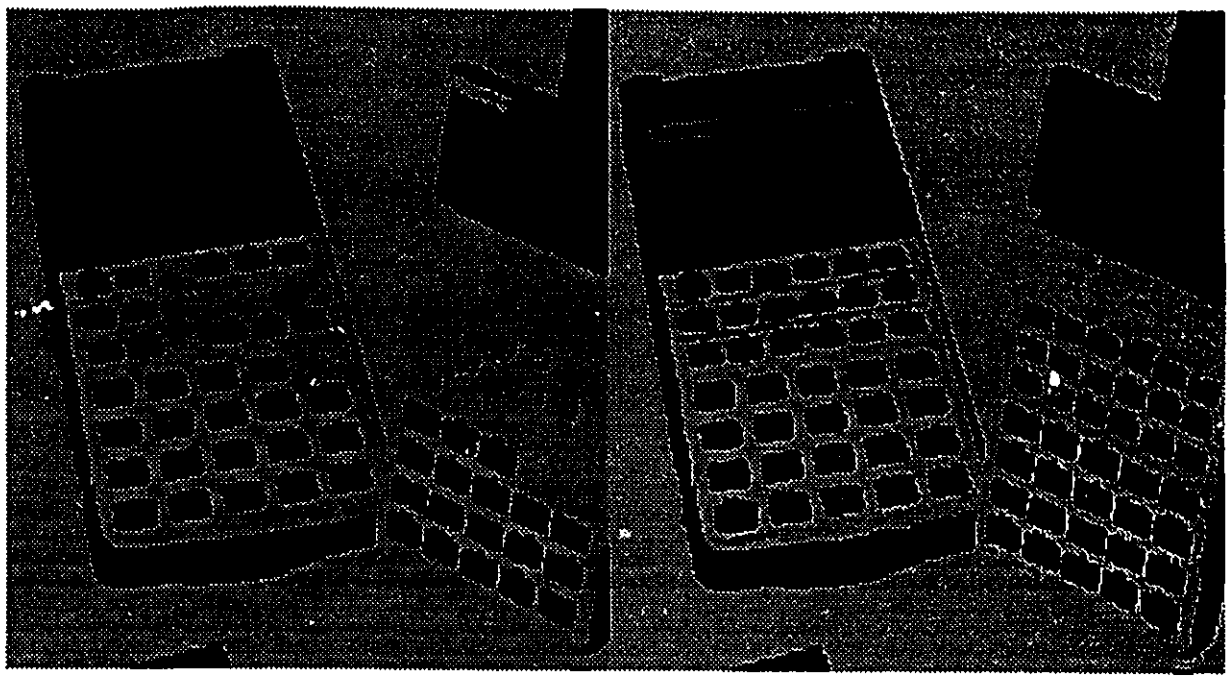
0.097 bpp

0.294 bpp

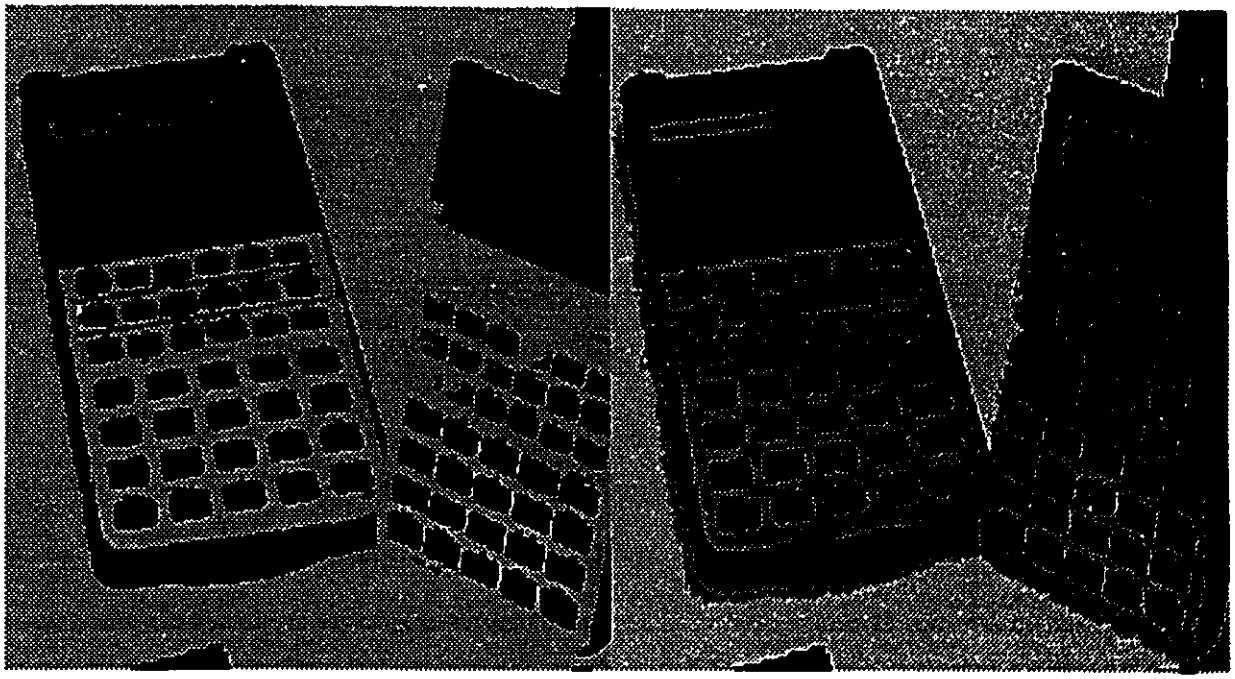


0.193 bpp

0.374 bpp



0.144 bpp 0.360 bpp



0.290 bpp 0.514 bpp