Adaptive Bit Allocation for Spatiotemporal Subband Coding Using Vector Quantization

by

Xiang Chen

B.S., Electrical Engineering
Beijing Polytechnique University
Beijing, P.R.China
1982

SUBMITTED TO THE MEDIA ARTS AND SCIENCES SECTION
SCHOOL OF ARCHITECTURE AND PLANNING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTER OF SCIENCE

AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1990

Signature of the Author

....

Xiang Chen
Media Arts and Sciences Section
May 11, 1990

Certified by

....

Andrew B. Lippman
Lecturer, Associate Director, Media Laboratory
Thesis Supervisor

Accepted by

....

Stephen A. Benton
Chairman
Departmental Committee on Graduate Students

Adaptive Bit Allocation for Spatiotemporal Subband Coding Using Vector Quantization

by

Xiang Chen

Submitted to the Media Arts and Sciences Section on May 11, 1990 in partial fulfillment of the requirements of the degree of Master of Science at the Massachusetts Institute of Technology

Abstract

Subband coding of digital image is drawing more attention than ever before. The fundamental property of subband coding system is that the original signal is decomposed into several statistically dissimilar signals. On one hand, this decomposition constitutes the primary decorrelating mechanism, which affects the signal compression. However, it raises the problem of optimally allocating the available channel capacity among the signal components. This is more critical in coding image at low bit rate. A bit allocation algorithm based on the the statistics of source image data is developed in this thesis. The original image data is first transformed into a set of spatioltemporal subbands by using QMF filters. Then statistics are run through the subbands. According to the statistics, the bit allocation program allocates the predescribed total bits to different subbands in such a way that the total distortion in the reconstructed picture will be minimized. A bit rate controlled kd-tree based vector quantizer is used to conduct the quantization.

Thesis Supervisor: Andy Lippman
Title: Lecturer, Associate Director, Media Laboratory

Work reported herein is supported by a contract from the Movies of the Future consortium, including AT&T, Columbia Pictures Entertainment Inc., Eastman Kodak Co., Paramount Pictures Inc., Viacom International Inc., and Warner Brothers Inc.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital images represented at a low enough bandwidth for them to be useful in computing applications and new consumer products are more and more in demand, since both digital data communication and digital data storage are surprisingly increasing. We address image representations that allow high quality moving video sequences at near 1.5 Mb/sec.

For the past few years, the Movies of the Future Group at Media Lab, MIT has explored the use of subband decomposition of image sequences for this purpose.

Subband coding was introduced by Crochiere *et al.* [34] in 1976 originally for speech coding. However, considerable attention has been devoted to subband coding of images since then. The virtue of subband coding is to split up the signal frequency band into subbands and to encode each band with a coder and bit rate accurately matched to the statistics of that particular band. Also, the response of the viewer's eye to different frequencies can be taken into account, and quantization error can be increased in those bands for which it will be less visible.

Recent advances of subband coding shows that subbands allow a multi-scale image representation [5] [40] , and afford good performance in coding [23] [1] [22].

Subband coding can be viewed in two components. First the image sequence is analyzed by a series of linear filters into spatial and temporal regions, then these regions are each quantized. A variety of quantizers may be used, and further processing may also be performed on the subbands. We have experimented with DCTs and VQ at the Movies of the Future Group, Media Lab, MIT (refs, tech notes, etc)

In this thesis we examine the bit allocation applied to the subbands. and the preprocessing applied to the image before subband decomposition. This is important because the quantization is the source of the compression, and it inherently involves a loss of data and potential reduction of picture quality. We can view this process as adding noise selectively to the image, and the manner in which the noise is add and the amount of it will determine quality and efficiency. It is important to realize that the encoding process is a matter of throwing away information (data) in such a way that the reconstructed signal at the decoder output is corrupted the least. (At least by a visual criterion, if not by a mean squared error measure.)

We will describe the basic structure of the coder and its value in compression and application, some aspects of the psychophysics of image coding that are exploited in the subband coder, and characteristics of compressed images when noise is added to the subbands. Finally, we will describe a bit allocation method and a prefilter that can be applied to the images to allow high quality and low bit rate.

Before preceding, it is worth while to briefly review Shannon's source coding theorem.

1.1 Noiseless Source Coding

A data compression system is said to be ***noiseless*** if the source symbol sequence is reproduced exactly at the destination. Consider partitioning the source sequence into subsequences of length N. We define the ***information*** contained in a source symbol sequence of length N to be

$$i(x) = -\frac{1}{N}\log_2(p(x)) \quad \textit{bits/source symbol.} \tag{1.1}$$

The average information per source symbol, known as the source's *Nth order entropy* is

$$H_N(p) \triangleq \frac{1}{N}\sum_{x\in\chi^N} p_N(x)\log_2\left(\frac{1}{p_N(x)}\right) \quad \textit{bits / source symbol,} \tag{1.2}$$

where $p_N(x)$ is the *Nth* order source pmf (probobility mass function).For most discrete sources of interest, this quantity approaches a finite limit, called its *entropy*, as N approaches infinity:

$$H(p) \triangleq \lim_{N\to\infty} H_N(p) \quad \textit{bits /source symbol.} \tag{1.3}$$

The noiseless source coding theorem states:

Theorem 1 (*Shannon's Noiseless Source Coding Theorem* [38]) *For any $\epsilon > 0$, there exists a noiseless source encoder / decoder pair (codec) with average encoder output bit rate less than $H + \epsilon$, where H is the source entropy. Conversely, no noiseless source encoder and decoder exist for which the encoder output bit rate is less than H.*

The primary value of this theorem is that it provides a bound for the source coding if the source entropy is known. However, it does not give a method for constructing a source coding system.

1.2 Noise Source Coding

When the available channel capacity is less than the source entropy, the noiseless source coding theorem implies that we can not communicate the source sequence without error. In this case, we can, however, try to contract a codec which introduces relatively benign types of errors in the reconstructed sequence. For such an approach, we must be able to quantify distortion between the source sequence x and the reproduction sequence $\tilde{x}$ as a function of the two sequences in a meaningful way. Ideally, the distortion function should reflect the degree to which the reconstructed sequence differs from the original. So far, the most commonly employed distortion measure is the *mean square error* (mse), defined as

$$d(x,\tilde{x}) \triangleq \frac{1}{N}\sum_{i=0}^{N-1}(x(i)-\tilde{x}(i))^2, \tag{1.4}$$

for sequences of length N. Mean square error distortion is zero in the case of perfect reconstruction.

The source coding problem is to design a codec which achieve the minimum average distortion between the original and reconstructed source sequences, subject to the constraint that the average encoder output bit rate be less than a prescribed value. Alternately, we may want to minimize the average encoder output rate, subject to a constraint on the maximum allowable distortion. Here, we introduce a generalized entropy concept called *rate distortion function*, $R(\delta)$, which specifies the minimum channel capacity required to transmit the source sequence with distortion less than δ, as stated in *Shannon's Source Coding Theorem*,

Theorem 2 *(Shannon's Source Coding Theorem [38]) Fix $\delta \geq \delta_{min}$. For any $\delta' > \delta$ and $R' > R(\delta)$, for sufficiently large N there exists a source code C of length with M codewords, where:*

(a) $M \leq 2^{[NR']}$

(b) $d(C) < \delta'$

In this theorem, N is the number of source symbols grouped together into a block, $[\cdot]$ denotes the largest integer less than the argument, and C is a *source code* consisting of a set of M possible reconstruction vectors,

$$C = (x_0, x_1, \ldots, x_{M-1}), \quad x_j \in \chi^N. \tag{1.5}$$

If we assume that the encoding function, $C(\cdot)$, selects the reconstruction vector $\tilde{x} \in C$ that minimizes the mean square error to its argument, $x \in \chi^N$, then the average distortion of code C is defined as

$$d(C) \triangleq \sum_{x \in \chi^N} p(x) d(x,\ C(x)). \tag{1.6}$$

In general, R' approaches the rate distortion function only as N becomes large, implying that performance is proportional to the sequence length.

As we have seen, neither the noiseless nor the noisy source coding theorem provides an explicit method for constructing practical source coding systems. However, they do suggest methods for codec design in image coding.

Chapter 2

Subband Coding

A subband coder can be viewed as two basic components as shown in Figure 2.1. The first part of the chapter discusses the subband analysis/synthesis; the virtues of subbands, such as: energy concentration, localization of features in the images, and the division of images into varying and relatively stationary subbands.

The second part of the chapter explores the kd-tree based vector quantizer; its design issues; the selection of splitting dimension and the division point on the selected

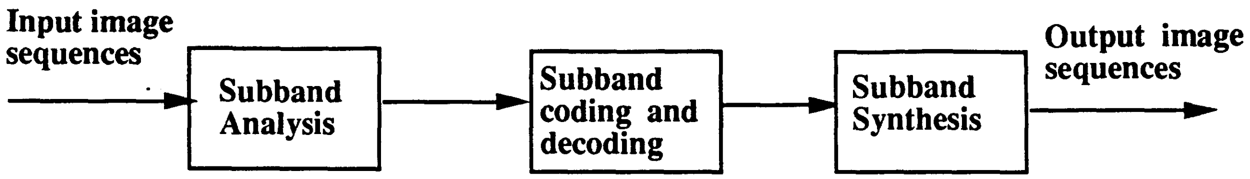

Figure 2.1: Subband Coder.

dimension; and the bit rate calculation and control in kd-tree based VQ process.

2.1 Subband principles

The moving image sequences are represented as a three-dimensional digital signal with two dimensions in space and the third over time. Therefore, the three dimensional subband method should be employed to remove the redundancy in each dimension. [44] [46] [5] [22].

In subband analysis, the input signal is broken up with a set of band-splitting filters that cover fully the frequency spectrum of the information in the original signal. Because the filters cover the frequency domain exactly and each resulting signal can be sub-sampled at its new Nyquist sampling rate, the number of data points does not increase. However, there is a chance of information loss, due to the leakage in the filters. Therefore, the desired characteristics of a subband are that its filters exhibit a minimal leakage and that the result will be a set of output subbands that have a definite hierarchy of importance. Here we need to solve two problems: finding a set of filters which preserve total information of original signal in the set of subband signals; splitting the subbands in such a way that the bandwidth of transmission channel can be intelligently allocated by more finely coding the important sub-bands and coarsely coding the unimportant ones. The *Quadrature Mirror Filter* (QMF) technique has been normally used as the solution for the first problem [12]. The solution for the second one is associated with issues in *Human Visual System* (HVS). The subband coding schemes for image coding in conjunction with a visual model is quite strong due to the HVS's nonuniform response across different portions of the spatialtemporal spectrum. Troxel et. al. and Schreiber [42], Glenn[13] [14] and Kelly[24] all discussed the reduced distortion sensitivity to high frequency spatial detail that humans exhibit.

Spatially, the human response is not easily modeled, but there are well known components of the early visual pathway that are surprisingly well approximated using simple filter banks. Butera[5] took advantage of this approach and employed a subband decomposition within an image coding system to model these components. He cites Wilson's work[45] which observes that groups of receptors in the human visual system. are organized into roughly one octave wide circularly symmetric band pass channels. Each of these receptive units therefore responds to detail at a given scale and is isotropic in nature. This suggests a spatial subband decomposition of an image into one octave wide regions is a suitable and practical means of modeling the human visual system. It also has been shown through experimentation that humans have lower acuity with respect to diagonal edge detail than more vertical or horizontal orientations. It is important to note that source modeling alone establishes the same fact. Even though there is no universal probability distribution function for images, most of the images do not have flat spectra. In fact, the amplitude of the spectral components decreases rapidly in proportion to spatial frequency. Octave subdivisions, thereafter, make sense in that they divide the spectra into pieces which have roughly equal amounts of information.

Temporally, there is less support for octave divisions. The studies of temporal subband division are not as satisfactory as the one of spatial subband. Romano [22] described a temporal uniform subdivision scheme which uses 6Hz as base frequency. This scheme is based on the fact that 24Hz, 30Hz, and 60Hz temporal rates are widely seen in current imaging systems in this country, and all of these frequencies are conveniently divisible by 6Hz. However, the computational complexity in the implementation is very high. Lippman and Butera [27] proposed a coarse temporal subband division method. In this method, there are only two temporal subbands, low and high. The low band is the average of a pair of pictures and the high band is the difference of the pair. The poor temporal resolution of the test sequences makes a further level of temporal filtering not necessary. As a matter of fact, these two temporal bands are actually composed by the shortest QMF filter. This method also gives us a perfect temporal reconstruction.

Since this thesis emphasizes the bit allocation among the subbands, choosing the coarse temporal subband division will not lose the generalization in the bit allocation problem. The bit allocation scheme developed on the two temporal subband system should be easily extended to multi-temporal subbands system.

Applying both spatial and temporal subband partitioning scheme examined above, we use a spatiotemporal system shown in Figure2.2 as the three dimensional subband model on which we will develop our adaptive bit allocation scheme. In the system, 1 dimensional filter is applied separately in vertical and horizontal direction to build up two dimensional spatial subband. The lowest spatial subbands of each level are further filtered to get the next level pyramid subbands. There are three levels of pyramid in all. In Figure2.2 and all of the figures throughout this thesis, we use the following notation:

- LL - Low pass filter applied both vertically and horizontally
- LH - Low pass filter applied vertically and high pass filter applied horizontally
- HL - High pass filter applied vertically and low pass filter applied horizontally
- HH - High pass filter applied both vertically and horizontally

Figure2.3 shows the corresponding diagram in frequency domain.

2.1.1 Subband Analysis & Synthesis

The problem of splitting a signal into subbands and then reconstructing it again is usually looked upon as a pure signal processing problem. The exact signal characteristics and the nature of application is not given any attention. Effectively this means that errors due to encoding, transmission and decoding are neglected. In that case the splitting and reconstruction stages become attached and the design is derected at perfect or nearly perfect reconstruction. For splitting a one dimensional signal into two subbands the QMF technique was introduced by Croisier, Esteban and Galand [7].

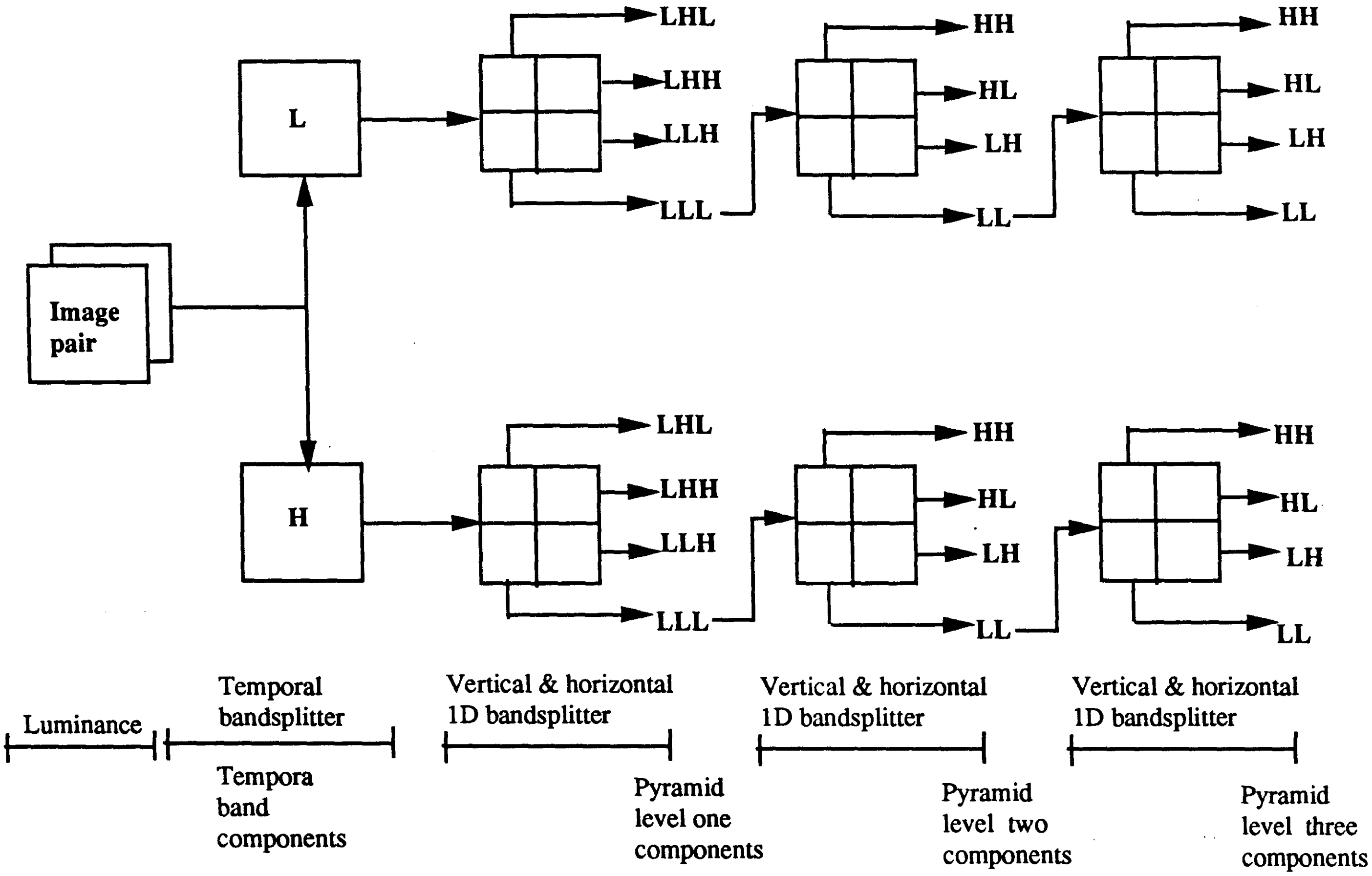
LHL
LHH
LLH
LLL
HH
HL
LH
LL
HH
HL
LH
LL
L
Image pair
H
LHL
LHH
LLH
LLL
HH
HL
LH
LL
HH
HL
LH
LL
Luminance
Temporal bandsplitter
Tempora band components
Vertical & horizontal 1D bandsplitter
Pyramid level one components
Vertical & horizontal 1D bandsplitter
Pyramid level two components
Vertical & horizontal 1D bandsplitter
Pyramid level three components

Figure 2.2: Partitioned Luminance Spectrum.

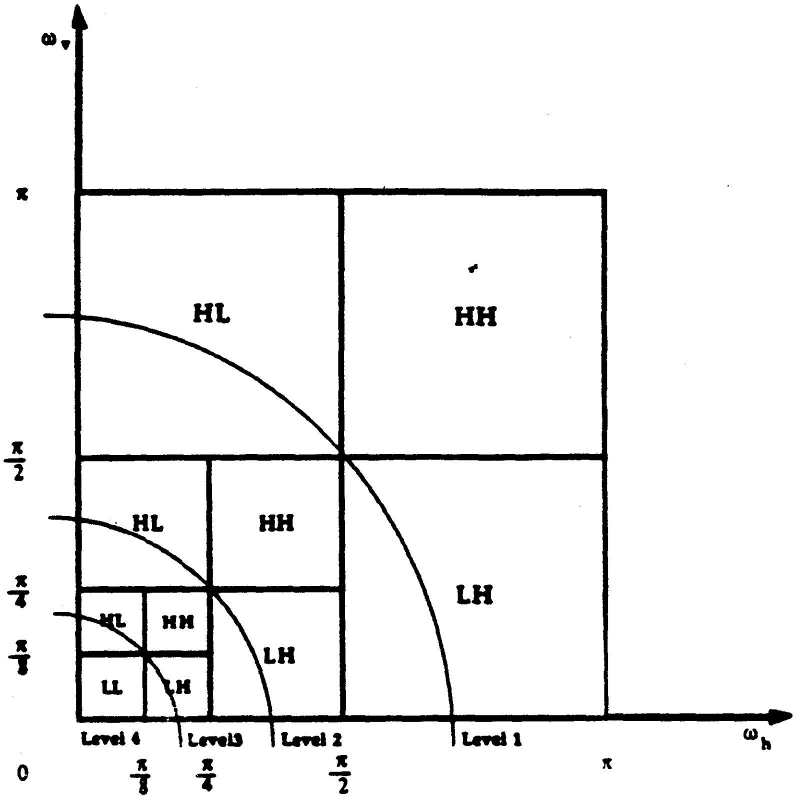

Figure 2.3: Frequency domain subdivision in a typical 3 level QMF pyramid.

A block diagram of an arbitrary M band system is shown in Figure2.4 The diagram implies that the M bands have equal extents, therefore, decimation rate is constant across subbands. This is not necessary. In this thesis, we set M = 2.

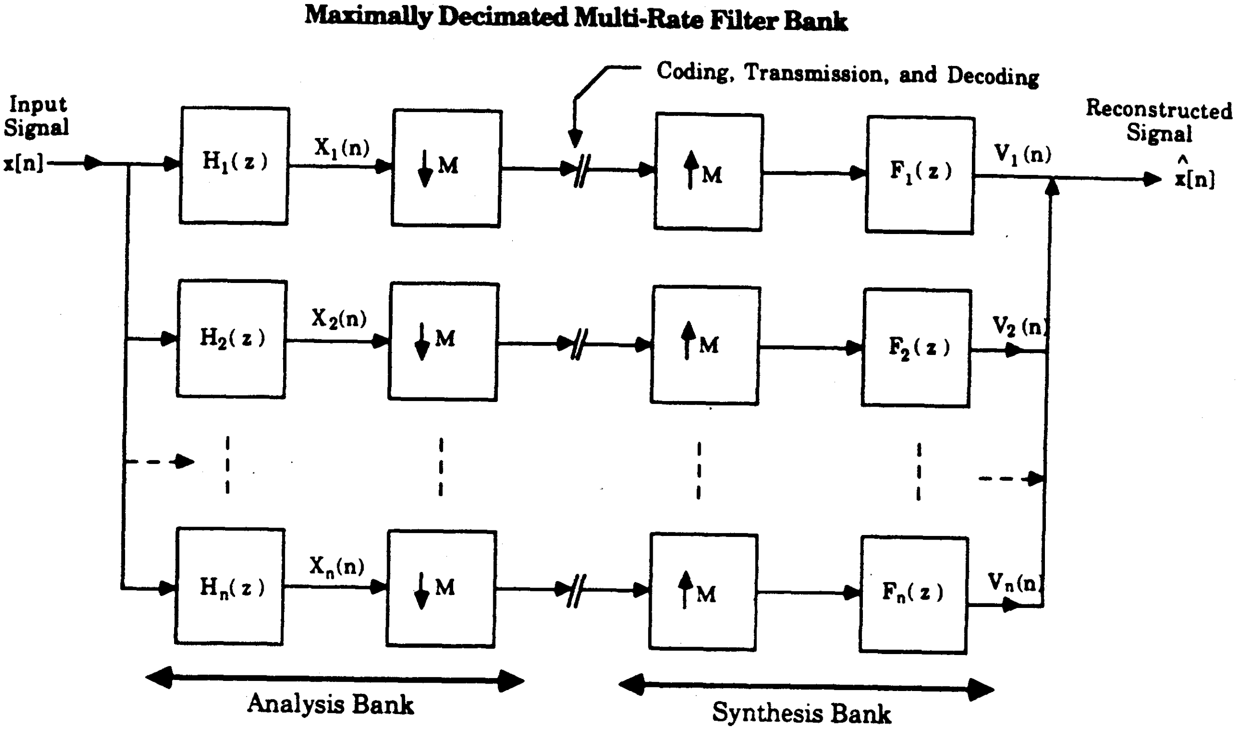

Figure 2.4: **Arbitrary M band analysis/synthesis system.**

In traditional filter design theory, an ideal band pass filter, which seems to be required by the perfect reconstruction filter bank, is not realizable. The filters in the real world are either separated to avoid overlap resulting in an overall notching of the spectrum or allowed to overlap resulting in aliasing. These situations are shown in Figure2.5 Neither choice satisfies the perfect reconstruction criterion.

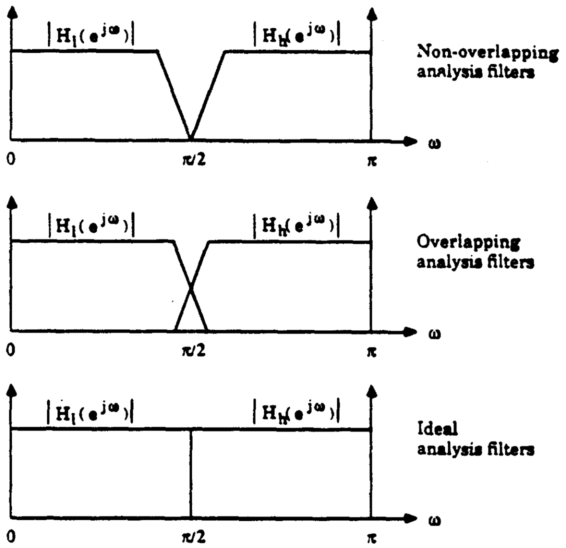

Figure 2.5: **Top:** Notched 2way band splitter. **Middle:** Overlapped 2way band splitter. **Bottom:** Ideal band splitter

Any notch in the analysis transfer function introduces an unrecoverable error, because it completely blocks a component from passing through the system. It is therefore reasonable to allow the analysis filters to overlap resulting in aliasing and to design the whole system in such a way that the overall system make a cancellation on any aliasing. The QMF filter, which has two band, is one implementation of this idea. Figure2.6 illustrate the aliasing as a result of decimating the pictured spectrum by a factor of 2. To understand how the aliasing might be canceled through out the overall system, let us begin by writing the expression out.

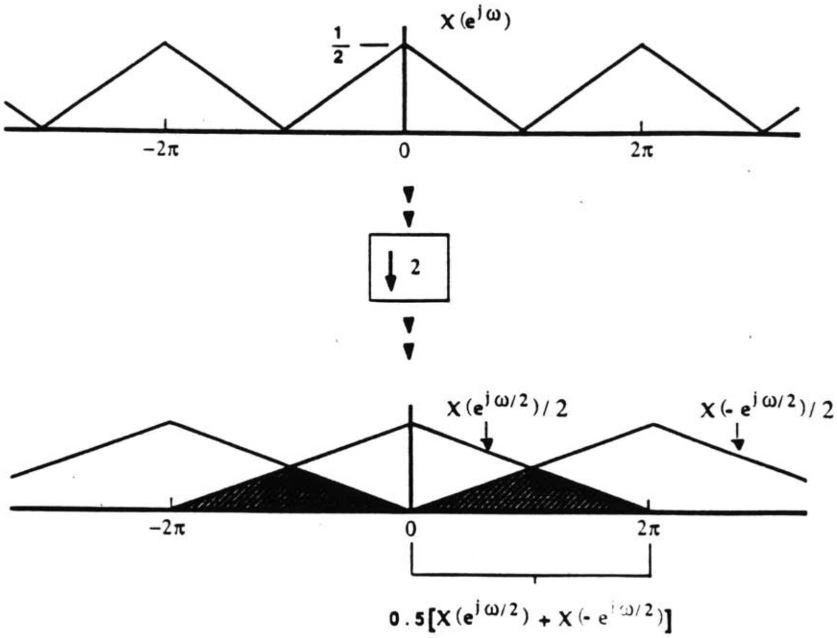

Figure 2.6: **Aliasing as a result of decimating the pictured spectrum (top) by a factor of 2**

Analysis filters $H_i\left(e^{j\omega}\right)$ are used to decompose input $x[n]$ into a low-frequency subband and a high-frequency subband which are in quadrature to one another. After coding, transmission, and decoding, synthesis filters $F_i\left(e^{j\omega}\right)$ are used to reconstruct the signal. If $h_i\left(e^{j\omega}\right)$ are ideal brickwall filters, the system will obviously work without failure. QMF's are such that aliasing caused by $H_i\left(e^{j\omega}\right)$ are canceled out by $F_i\left(e^{j\omega}\right)$.

The transfer function for the two-channel QMF bank of Figure2.4 is:

$$\begin{aligned} X'\left(e^{j\omega}\right) &= \frac{1}{2}\left[H_0\left(e^{j\omega}\right)F_0\left(e^{j\omega}\right) + H_1\left(e^{j\omega}\right)F_1\left(e^{j\omega}\right)\right]X\left(e^{j\omega}\right) + \\ &\quad \frac{1}{2}\left[H_0\left(-e^{j\omega}\right)F_0\left(e^{j\omega}\right) + H_1\left(-e^{j\omega}\right)F_1\left(e^{j\omega}\right)\right]X\left(-e^{j\omega}\right) \end{aligned} \tag{2.1}$$

Furthermore, choosing the synthesis filters to be

$$F_0\left(e^{j\omega}\right) = H_1\left(-e^{j\omega}\right) \tag{2.2}$$

$$F_1\left(e^{j\omega}\right) = -H_0\left(-e^{j\omega}\right) \tag{2.3}$$

yields the overall LTI transfer function

$$T\left(e^{j\omega}\right) = \frac{1}{2}\left[H_0\left(e^{j\omega}\right)H_1\left(-e^{j\omega}\right) - H_1\left(e^{j\omega}\right)H_0\left(-e^{j\omega}\right)\right] \tag{2.4}$$

Adding the further constraint of

$$H_1\left(e^{j\omega}\right) = e^{-j\omega(N-1)}H_0\left(-e^{-j\omega}\right) \tag{2.5}$$

Insures there is no aliasing, amplitude or phase distortion. For more detailed QMF design technique, see [39] or [33].

The QMF filter bank discussed above provides us the freedom in picking the optimal subdivision of the spatiotemporal spectrum to meet the requirements of the image coding system. The subdivision method used in this thesis is octave for above mentioned reasons. In this approach, each successive split cuts away half of the spectrum along a given axis as shown in Figure2.2. This spatial image decomposition is called QMF pyramid transform [1]. The QMF pyramid transform can be applied recursively to the lowest subband previous application, and the subbands from each QMF pyramid process compose one level of pyramid. Each level of pyramid contains one quarter of pixels of the above level pyramid.

The 9 tap QMF analysis/synthesis bank employed in this work(designed by [39]) to perform the spatial band splitting is pictured in Figure2.7

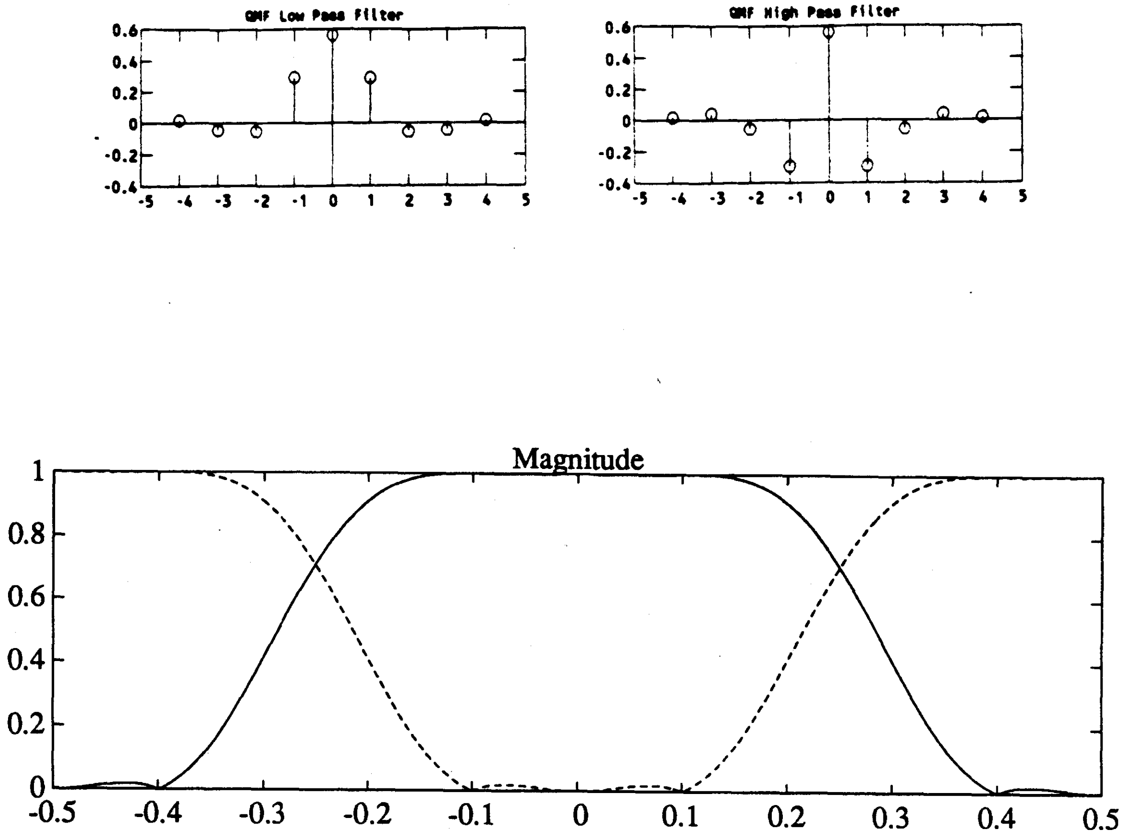

Figure 2.7: 9 tap QMF analysis/synthesis bank. Top left: Impulse response of the low pass filter. Top right: Impulse response of the high pass filter. Bottom: Magnitude response of low pass filter (solid line) and the high pass filter.

2.1.2 Subband Characteristics

The basic goal of subband coding is to isolate signal energy into a small number of bands and to separate information in band that are more or less critical to the human observer. An image is a mapping of three dimensional objects on a two dimensional plane. It consists of some amount of areas whose luminance is uniform or gradually varies. Consequently, the contiguous pixels in the image sequences are highly correlated with each other. In terms of statistics, subband decomposition is a method to utilize this redundancy. By dividing images into high and low frequency subbands, the probability density distribution of the pixels varies depending on the subband. As the high frequency subbands mainly consists of edges of the image, which occupies minor area of the whole image, the distribution of pixel values peaks at zero and only the edge pixels have non-zero value. Also the variance is highly reduced compared with that of the input distribution. A peaky distribution gives low entropy. The distribution characteristics of the lowest frequency subband is almost the same as the original image, that is, flat. However, by going down a pyramid level, the number of pixels decreases. Consequently, the total bits can be reduced.

Figure2.8 is the original picture (frame 6 of second 4 of alley sequence). and Figure2.9 illustrates the image characteristics of subbands in three level pyramid. The top left, left second from top, and the left third from top are HL bands corresponding to pyramid level 1, 2, 3, which contain the horizontal edges of the original image and some texture. The right one, two, three from bottom are LH bands corresponding to pyramid level 1, 2, 3, which contain the vertical edges of the original image and some texture. The one on the left corner is the LL band of pyramid level 3, which contains the basic shape of the original image, but blurred. The rest three subbands are HH bands from pyramid level 1, 2, 3, which contains the diagonal edges and some texture of the original image. All of the subbands here are spatial ones and no temporal ones are shown here. From this image pattern separation, the irrelevancy reduction is obtained.

Figure2.10 contains three histograms from original picture (from frame 6 of second 4 of alley sequence), LL band of pyramid level 1, and LL band of pyramid level 3, respectively. It is easy to see that all of these three histograms have almost the same shape. It proves that the LL band in each pyramid level has the same distribution characteristic as the original image, that is relatively flat. The three histogram in Figure2.11 are from LH, HL, and HH bands of the first level pyramid, and in which the frequency value higher than 200 has been omitted. We can easily see that the variance in these three bands is much reduced, especially, in the HH bands. Redundancy reduction has been achieved since the data with the same statistical characteristics

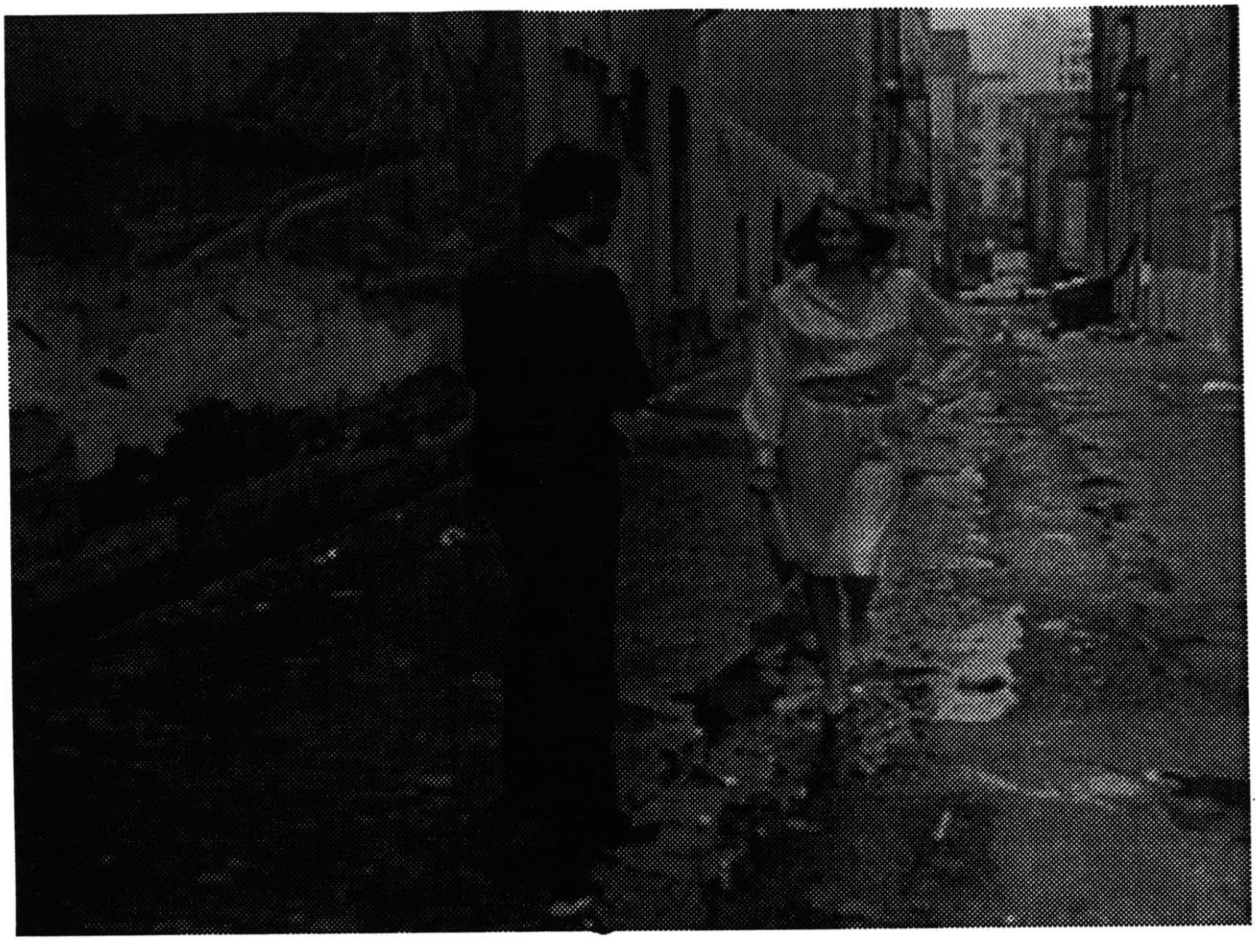

Figure 2.8: The original picture from frame 6 of second 4 of alley sequence.

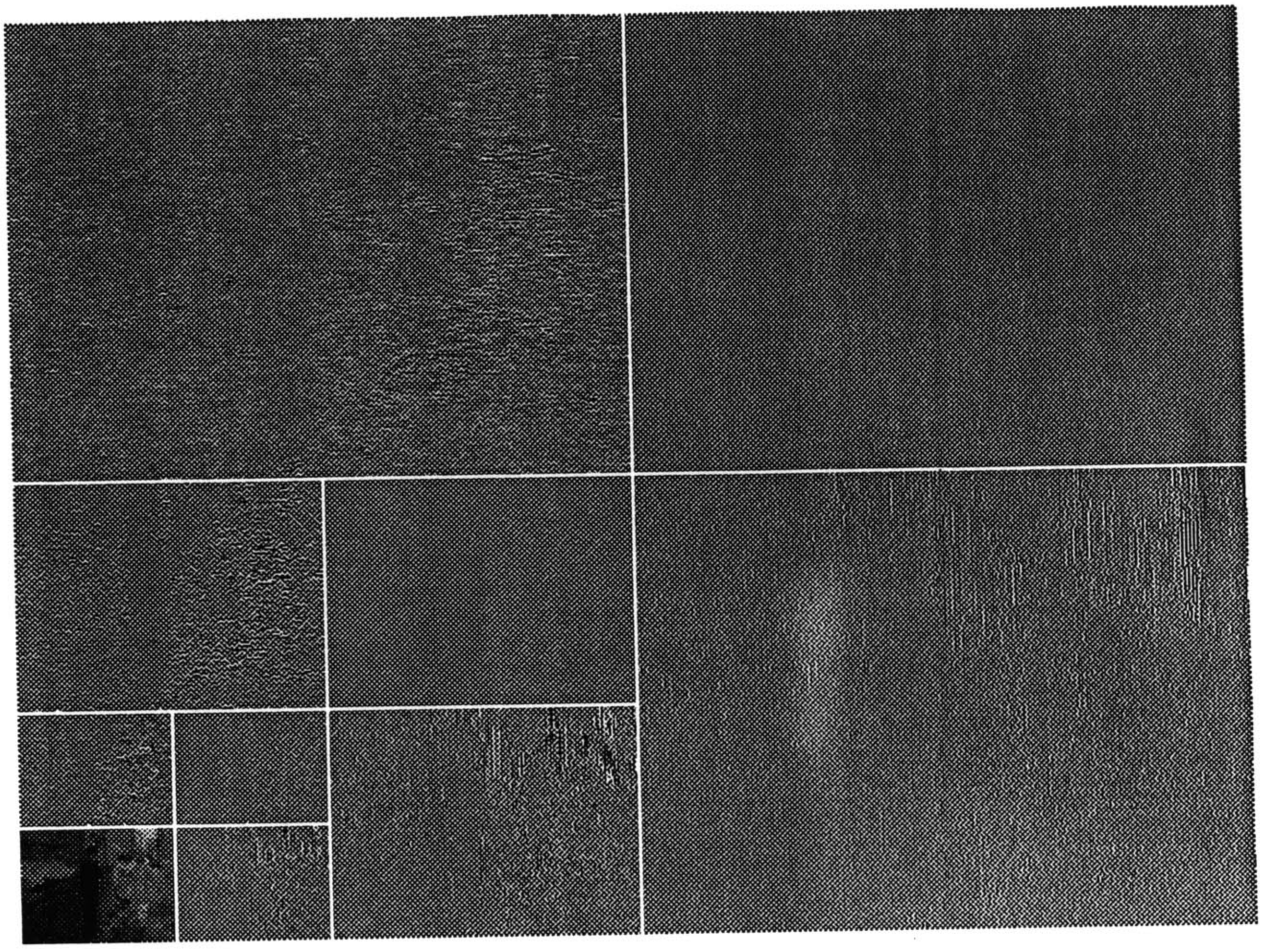

Figure 2.9: Pyramid decomposition of the original picture in Figure 2.8.

has been localized into the same subband.

The subband analysis yields a desirable separation of the data. However, it does not provide any data compression. The data compression can be realized by non-linear quantization of subband data. This will be discussed in next section.

2.2 Vector Quantization

The technique of subband analysis/synthesis dealt with in previous section offers us subbands with desired energy distribution and visual characterization. However, a proper quantizer is still needed to reach our goal: data compression. Vector Quantization(VQ) is a relatively new coding technique, but has many attractive features in image data applications, such as high compression ratios and very low decoding cost. Therefore, it has been widely used as the quantizer for the subband coding at low bit rate image communication.

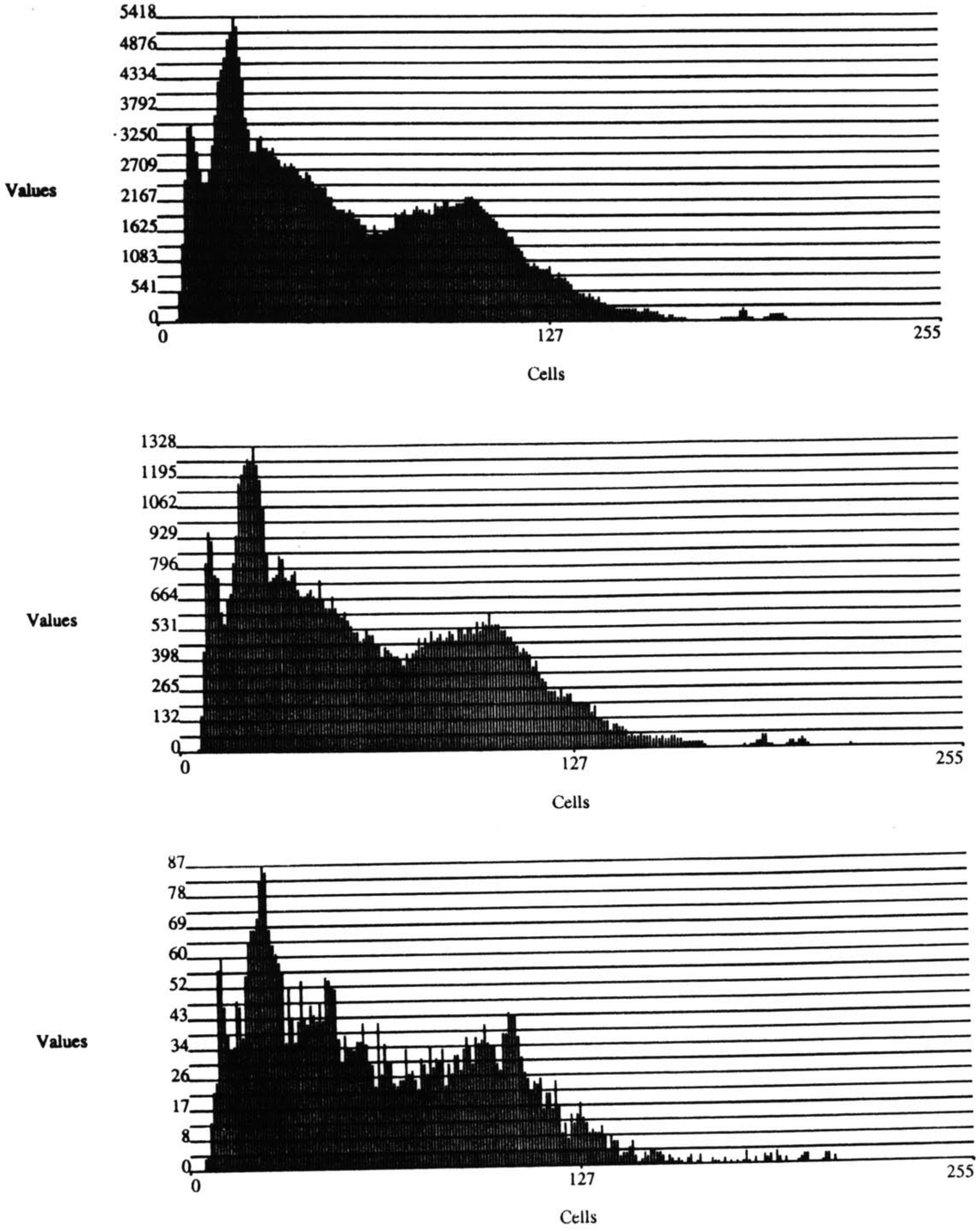

Figure 2.10: Histogram for original picture and LL bands.

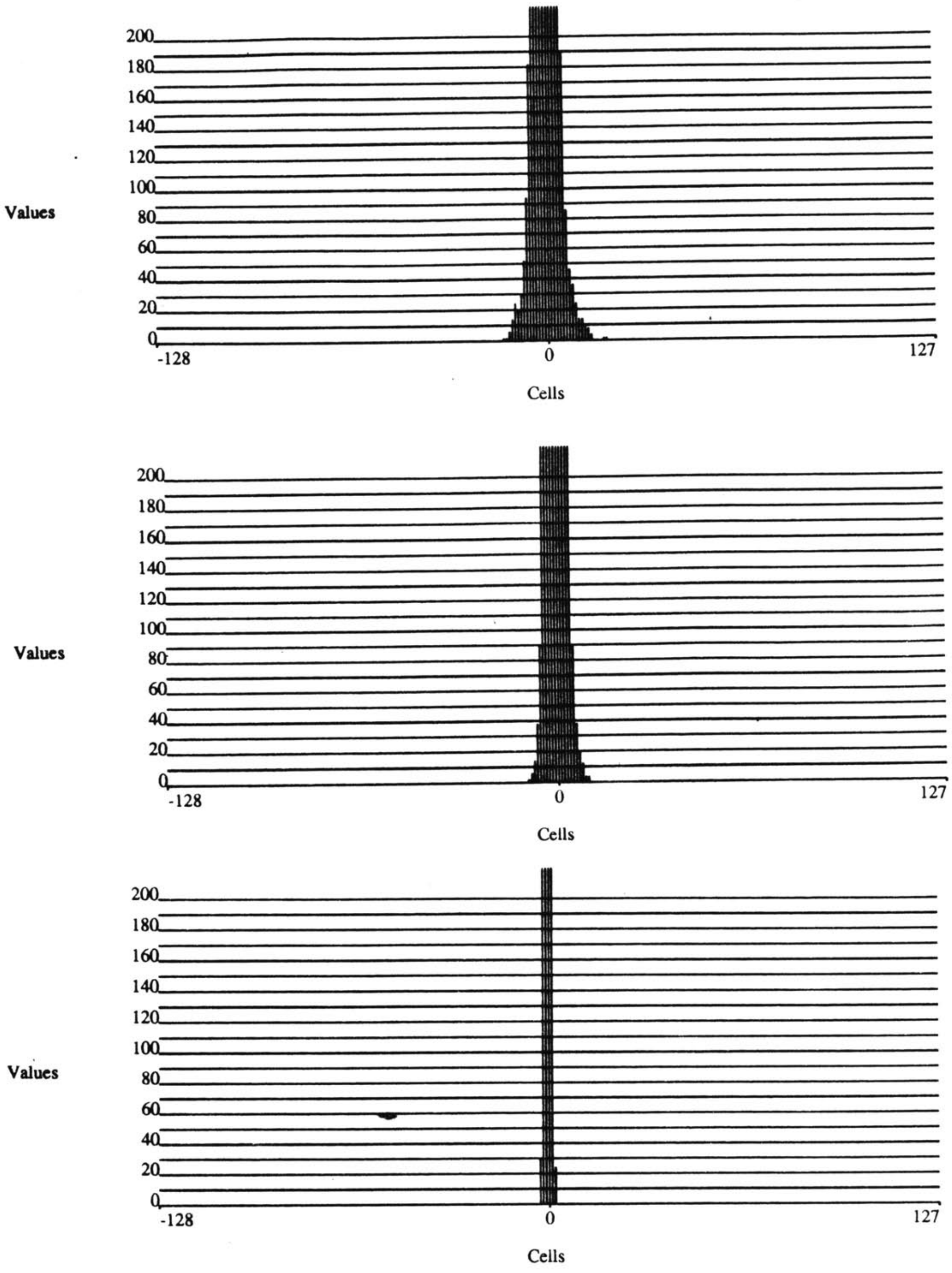

Figure 2.11: Histogram for LH, HL, and HH bands of pyramid level one.

2.2.1 An Overview on VQ

Although VQ had been applied to speech already in the 1950's by Dudley [9], the use of VQ was stimulated mainly by the work of Buzo and Linde *et al.* in 1980 [26], who proposed a practical method for vector quantizer design based on learning, using a training set. Gray [16] discusses a large number of variations that are possible on the basic vector quantization scheme.

A quantizer is defined to be a system that maps a domain to a range in a many to one fashion. In terms of coordinate systems, a quantizer divides the domain space into regions and maps all points residing in each to a single representative value. In general cases, the term quantizer refers to the type of operation described above applied in one dimension (known as a scalar quantizer). Vector Quantization, also known as block quantization or pattern-matching quantization, is a direct extension from one-dimensional (or scalar) quantization to the quantization of multi-dimensional signals.

The basic VQ block diagram is shown in Figure2.12. The input X to the encoder is a vector of dimension k. In image coding application, X is a block of pixels from an image. The image is partitioned into contiguous, nonoverlapping, blocks. We chose square blocks for convenience, although nonrectangular blocks have been found

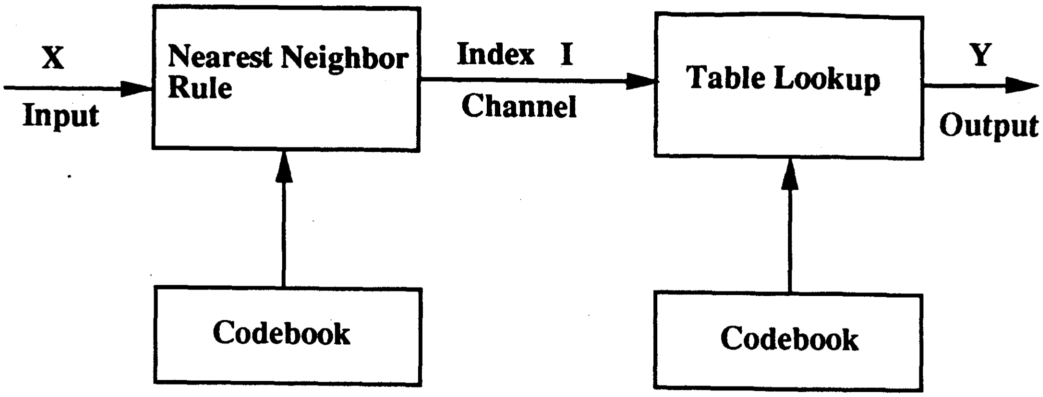

Figure 2.12: Vector quantization

to give slightly better performance. The encoder computes the distortion $d(X, Y_i)$ between the input X and each *code vector* Y_i, $i = 1, 2, \ldots, N$ from a *codebook* C. The optimum encoding rule [26] is the *nearest neighbor rule* 2.6,

$$Y_i \iff d(x, y_i) \leq d(x, y_j), \quad j = 1, \ldots, L. \tag{2.6}$$

in which the index I is transmitted to the decoder if code vector Y_i yields the least distortion. We need $\log_2 N$ bits to transmit the index if fixed length codewords are employed. The quantity $R = (1/k)\log_2 N$ is the rate of the coder in bits per pixel (bpp). The decoder simply looks up the I_{th} code vector Y_i from a copy of the codebook C, and the output $\hat{X}$ is is Y_i. The performance of VQ is measured by the average distortion $D = E[d(X, \hat{X})]$, where the expectation is taken over the *k-dimensional probability density of X.*

An optimal vector quantizer is one which employs a codebook C^* that yields the least average distortion D^* among all possible codebooks. The design algorithm of an optimal codebook is not known in general. However, a clustering algorithm, referred to here as the LBG algorithm, is described in [26] which leads to a locally optimal codebook. LBG operates in the following manner. An initial guess for a representative (codebook) is made. The Euclidean distance between each input vector and each representative is calculated and the input vector is assigned to the representative that is nearest to it. After all input vectors are assigned to representatives, the aggregate error introduced by coding each input vector with the current codebook is calculated. Each vector in the codebook is then replaced by the centroid of the cluster of vectors that map to it and the procedure is repeated. The iteration will stop when the distortion reaches a minimum. However, it is impossible to bound the convergence time or to guarantee that the algorithm will converge at all. It is obvious that LBG is difficult to implement in real time.

2.2.2 Tree Based VQ Codebook Design

Kd-tree (K dimensional tree), introduced by Friedman and Bentley [11], makes it possible to characterize multi-dimensional data as a binary search tree at the expense of some constraints placed on the subdivision process. The application of kd-tree data structure in vector quantization is discussed in detail by Equitz [10]. In a normal binary tree, only a single binary decision may be made at any node. The kd-tree extends this in such a way that the dimension which is subdivided at each node may be different, even though there is still only one dimension chosen each time. This mechanism provides more freedom in codebook design, because a different splitting dimension may be chosen at each node according to different distortion specifications. (This will be discussed more in section 2.2.4). Since only local scalar partitions are used in kd-tree, the calculations are limited to only that node. This makes kd-tree based VQ codebook design much faster in comparison with LBG algorithm. and gives more potential for real time implementation.

Based on the above idea, Romano [22] put together the previous work done at Media Lab, MIT in his Master thesis to describe a tree structure based codebook design, which partition the whole vector space into a set of unequally divided subspaces. Each of the subspaces has its local representative (code vector), the one which gives the smallest distortion among the vectors in the subspace. All of the representatives form the codebook and the elements of the codebook are linked by a k-dimensional tree data structure. Figure 2.13 depicts an arbitrary kd-tree with $K = 2$. The basic consideration for the design of the kd-tree based codebook are: on which dimension

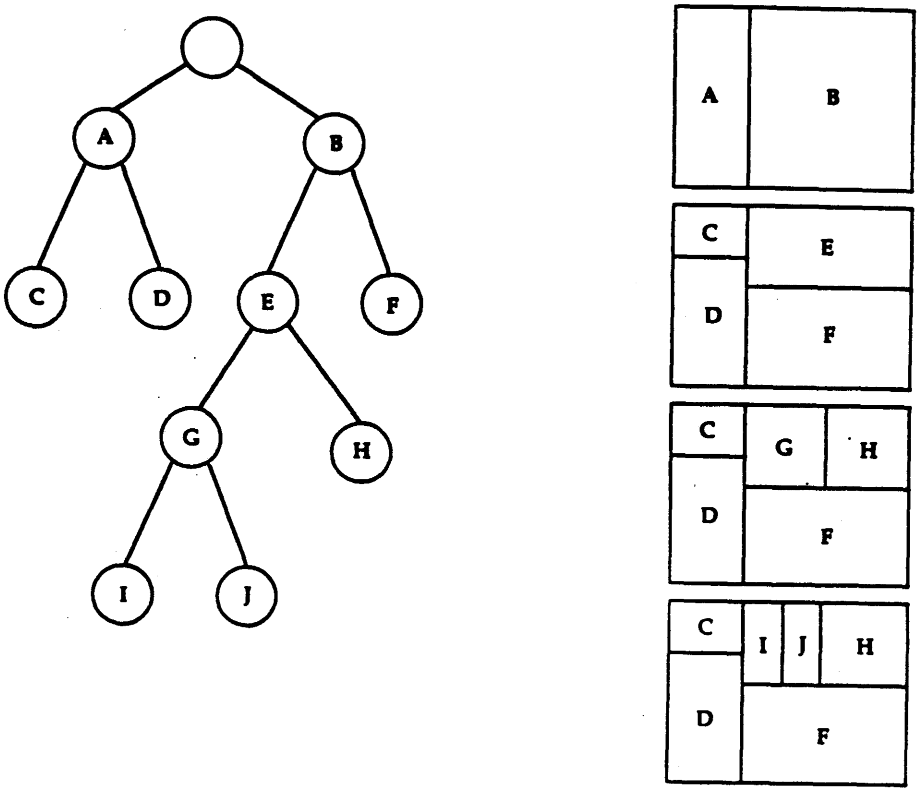

Figure 2.13: Kd-Tree: A typical kd-tree (K = 2) shown as both a binary tree (left) and a subdivided 2 dimension space (right)

we split the space; where on the selected dimension we split; and when we stop the subdivision.

The following control parameters are used in Romano's work

- **Node/Dimension Selection:** This parameter dictates how leaves are prioritized for subdivision. Candidates for this include selecting the leaf (and dimension within that leaf) based on the variance, cell extent, mean squared error (with respect to the representative), maximum error (with respect to the representative) or maximum number of constituents. With each iteration the leaf with the highest selection value becomes the candidate for subdivision. Note that this value is calculated for each dimension in the cell and the highest one is retained as the nodes actual selection value. The dimension that resulted in this maximum score is set to be the split axis should the leaf be selected for subdivision.

- **Split Point Selection:** This parameter determines how the split point along the chosen split axis is calculated. Candidates for this are the mean, median, mode or midpoint of the split dimension.

- **Representative Calculation:** This parameter describes the method used in calculating the representative vector for a given cell. Candidate for this is the centroid of the cell's population in current work.

- **Bounding Condition:** This parameter sets the criterion which must be met before the subdivision process terminates. Candidates for this are the code book size, total distortion incurred by quantizing and Peak distortion incurred by quantizing. Note that the term "distortion" here can be replaced with any metric desired.

Even though there are 5 choices in dimension selection and 3 ways in split point selection, Romano did not explicitly explain how the combination of these parameters (up to 15 combinations) affect the output of the vector quantizer, or reconstructed images in his work [22].

Also, it should be noted that the three methods for split point selection: mean, median, middle, are identical in subband image coding practice, because all of the subbands except the lowest one have a histogram with Laplacian like distribution.

Some experiments are conducted in section 2.2.4 to examine how these parameters works.

Different distortion criteria were introduced as bounding condition in terminating the subdivisions. This shows a better cooperation with different distribution among subbands. However, the bounding condition here did not couple with any bit rate control. Since we are working on very low bit rate image coding, the bit allocation among the subbands becomes critical. We need to introduce a new bounding condition which can take the bit rate at each subdivision as the termination decision. This will make the explicit bit allocation in the tree based quantizer realizable.

2.2.3 Bit Rate Control in Tree Based Codebook Design

In Figure 2.12, the only information to be transmitted is the codebook index I, or *codeword*, if a global codebook is pre-stored in the decoder. However, some recent papers [2] [15] [17] [5] [22] suggest the advantages of codebooks which are tuned specifically for individual images or specific temporal neighborhoods. In other words, the codebooks must be incrementally updated to make the VQ useful for the low bit rate coding of images.

As shown in Figure 2.11, the peaky distribution characteristics of all of the subbands except the lowest one gives low entropy. In the review of noiseless source coding theory in chapter one, we know the entropy of the source data provides us a low bound for encoding. Therefore, an entropy coding scheme should be considered in coding the vector index. As a matter of fact, this is one of the advantage of subband coding we should thereafter take.

Many practical algorithms exist that allow one to get close to, or even fall below, the theoretical limit such as Huffman coding, run length encoding and arithmetic coding [35] [36] [31]. The reason it is possible to out perform this lower bound is that some entropy coders exploit the joint statistics between samples of a signal. The entropy calculation described in Equation 2.7 only utilizes first order statistics.

Entropy, here in this thesis, is calculated from following equation:

$$Entropy = \sum_{i=0}^{N-1} -p(x_i) \log_2 p(x_i) \tag{2.7}$$

where $p(x_i)$ is the probability of the sample value x_i. (This is $N = 1$ case of equation 1.2)

The total number of bits used for transmission of the coded the images is given:

$$Total\ Bits = Bits_{codebook} + Bits_{codeword} \tag{2.8}$$

Our goal here is to use this total bits to terminate the subdivision process in kd-tree based codebook design. We first find out how to calculate the bits used for codebook and codeword respectively, and then apply it to the kd-tree subdivision termination.

Let us use M to denote the number of pixels in the image to be coded, and L to denote the vector dimension. Hence the total bits for codeword is:

$$Bits_{codeword} = (M \div L) \times entropy \tag{2.9}$$

In kd-tree case, $p(x_i)$ in 2.7 may be expressed as:

$$p(x_i) = \frac{number\ of\ vectors\ in\ the\ i_{th}\ sub-space\ (leaf)}{Total\ number\ of\ vectors\ in\ the\ whole\ vector\ space}\ . \tag{2.10}$$

And N in 2.7 represents the number of subspace up to the current subdivision.

The total bits used for codebook may by calculated as:

$$Bits_{codebook} = L \times N \times entropy\ . \tag{2.11}$$

However, the entropy of the codebook is not as easily calculated as that of the codeword at each subdivision. In practice, the total bit rate of the codebook may be calculated in either of the following two ways which approximate the bit rate by entropy calculation:

$$Bits_{codebook} = \left(\sum_{i=1}^{L} ceiling\ [\log_2(max[i] + min[i] + 1)] \right) \times N \qquad (2.12)$$

$$Bits_{codebook} = \left(ceiling\ \left(\log_2 \left[\prod_{i=1}^{L} (max[i] - min[i] + 1) \right] \right) \right) \times N \qquad (2.13)$$

A test is conducted here to see the difference of bit rate calculation of codebook among the ways represented in 2.11,2.12, and 2.13.

Table 2.1 shows the kd-tree parameter's combination used in the tests, which is also referred throughout the rest of the chapter.

Table 2.2 illustrates the test result. The test picture is the one in Figure 2.15. All of the test are terminated by error limitation at $SNR_{peak} = 42db$, and the total bits used for codeword calculated by equation 2.9.

The numbers in the first column refer to the test described in 2.2; the numbers in

Test #	Select	Split
1	Peak Error	Mean
2	Peak Error	Median
3	Peak Error	Middle
4	Extent	Mean
5	Extent	Median
6	Extent	Middle
7	MSE	Mean
8	MSE	Median
9	MSE	Middle
10	MAX MSE	Mean
11	MAX MSE	Median
12	MAX MSE	Middle

Table 2.1: Kd-tree Parameter's Combination

Test #	Entropy	Bit-added	Bit-multi	Error-added	Error-multi
1	135398	154332	145758	0.1398	0.0765
2	118077	135468	127942	0.1473	0.0835
3	175474	200988	189822	0.1454	0.0821
4	142197	162252	153238	0.141	0.0776
5	128447	147168	138992	0.1457	0.0821
6	177597	203544	192236	0.1461	0.0824
7	210671	241740	228310	0.1475	0.0837
8	201333	231696	218824	0.1508	0.0869
9	244546	281952	266288	0.153	0.0889
10	193548	221796	209474	0.1459	0.0823
11	179678	205920	194480	0.1461	0.0824
12	216988	249336	235484	0.1491	0.0852

Table 2.2: Bits Calculation Methods Comparison

the second, third, and fourth column represent the total bits in which the bits for the codebook part are calculated in the ways shown in formula 2.11, formula 2.12, and formula 2.13 respectively. The fifth and sixth column shows the difference between the ways: formula 2.11 and formula2.12; formula 2.11 and formula 2.13. At the most, formula 2.12 requires about 15 percent more bits in total then formula 2.11 does, while formula 2.13 requires only about 8 percent comparatively. It is obvious that formula 2.13 offers a better approximation towards formula 2.11. Thereafter, formula 2.13 is employed in the thesis work.

So far we have cleared the total bit calculation in kd-tree based VQ codebook design. However, there is still one assumption we need to prove before we can go ahead to apply this bits calculation method to control the termination in kd-tree subdivision. The entropy in formula 2.7 has to be monotonically increasing while N increases. To prove it mathematically is beyond the scope of the thesis. A test was run to show this assumption is generally true. The data from the picture in 2.15 is used to run the test. The X axis in Figure 2.14 represents the N in formula 2.7 in log_2 scale, while Y axis shows the entropy correspondently. It seems true that entropy increases monotonically with N increasing.

Using the work derived above, we have the control of the total bits used up to the

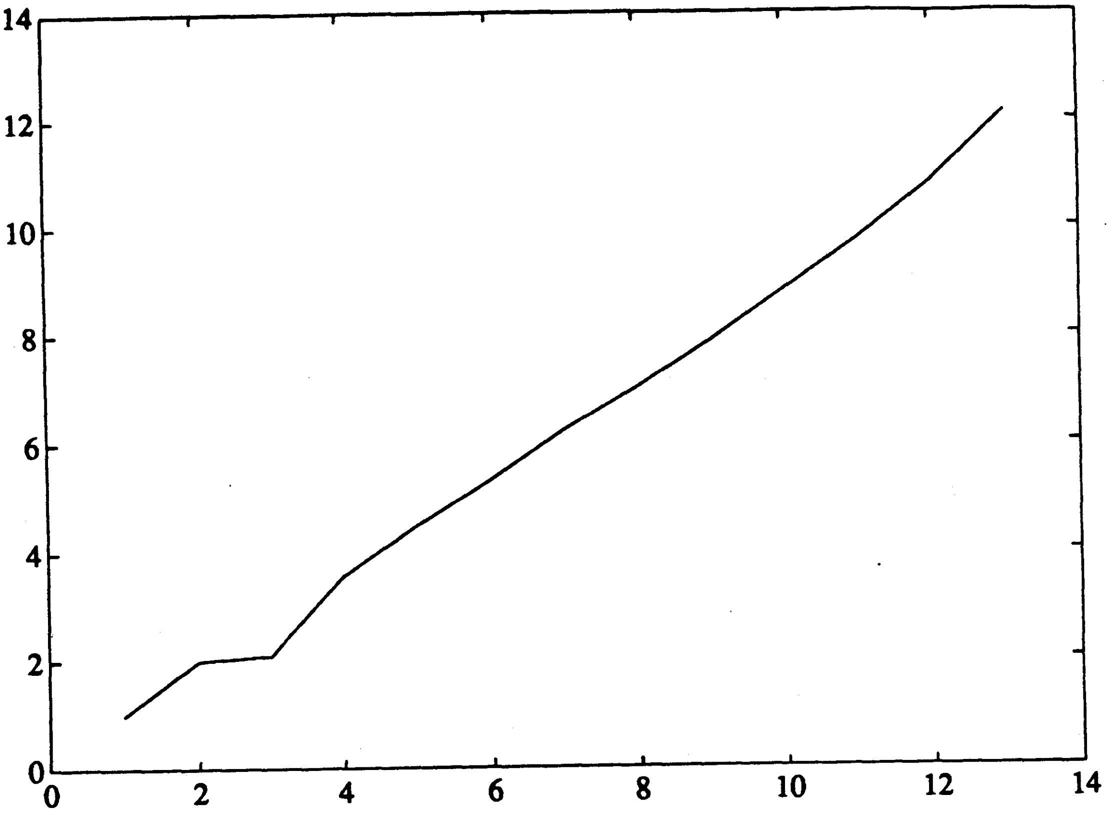

Figure 2.14: Entropy increasing monotonically with N in equation 2.7 increasing. The X axis represents N and Y axis represents entropy.

current subdivision in kd-tree data structure. This provide us a possibility to realize the bit allocation scheme which is going to be discussed in Chapter 4.

The total bit rate controlled kd-tree works in a very simple iteration. The tree initially begins as a single node whose constituency is the entire vector space. The selection value, split point and representative vector are also calculated. The following loop is then executed until pre-set total bit rate is met:

1. Find the leaf with the maximum selection value.
2. Make the leaf a node and create two new leaves.
3. Calculate the selection value/dimension, split point and representative vector for both new children.
4. Calculate the total bit rate according to formula 2.8, formula 2.11, and formula 2.13.
5. If pre-set total bit rate is not met, Goto step one.

2.2.4 Parameters in Tree Based Codebook Design

In kd-tree structure, the dimension and the point selected to do each iterative splitting are very important factors of the vector quantizer performance. A set of 12 test with different combinations of dimension selection and splitting point (Table 2.1) were run to evaluate how these combinations cooperate with total bit rate and image statistics.

Table 2.4 and Table 2.3 illustrate the experimental results from different kd-tree parameters selection combination using test picture 2.15 and 2.8 respectively. The tests use the entropy of the codeword to terminate the iterative kd-tree splitting and the entropy used here is 10.9 which is calculated by reconstructing the picture with SNR_{peak} = 42 db. The numbers in the first column are the tests with the kd-tree parameters combination listed in Table 2.1. Column two gives the codebook size. Column three shows the signal to noise ratio between the original picture and the reconstructed one. Column four and five describe the total bits used by codebook and codeword respectively. Numbers in the last column are the maximum pixel value in the error picture, which is the difference between the original picture and the reconstructed one. Carefully examining the two extreme cases, the test 1 and test 8, we can easily draw some conclusions. First, test 1 uses the codebook size much smaller than the test 8, even though both of them turned out reconstructed pictures

Test #	Codebook Size	SNR	Codebook Bits	Codeword Bits	Max MSE
1	3595	42.67	105080	207376	5
2	3835	41.12	111522	207362	8
3	4799	44.46	163562	207360	4
4	3971	43.25	161443	207360	8
5	4182	43.91	165556	207365	10
6	4848	44.57	177623	207364	5
7	5698	44.67	181840	207653	14
8	6117	43.71	139923	207379	24
9	6257	45.37	115795	207361	12
10	5193	44.41	121880	207363	11
11	5620	43.21	141366	207362	16
12	5553	45.08	165458	207361	8
13	5702	44.67	177945	207362	14
14	6105	43.71	181630	207371	24
15	6264	45.42	151038	207487	12

Table 2.3: Kd-tree Parameters Selection Tests for Picture 2.8

Test #	Codebook Size	SNR	Codebook Bits	Codeword Bits	Max Error
1	4287	39.12	135398	209280	60
2	3763	36.57	118077	209281	62
3	5583	40.86	175474	209757	60
4	4507	39.05	142197	209317	60
5	4088	37.27	128447	209303	63
6	5654	40.93	177597	209438	60
7	6715	41.09	210671	209287	58
8	6436	39.27	201333	209284	68
9	7832	41.88	244546	209288	58
10	6161	40.88	193548	209299	60
11	5720	38.89	179678	209283	64
12	6926	41.71	216988	209284	58
13	6716	41.09	210697	209282	64

Table 2.4: Kd-tree Parameters Selection Tests for Picture 2.15

with almost. same SNR value. Second, the maxmum pixel value in the error image of test 1 is smaller than that in the error image of test 8. Here a question is raised: what does the test 8 spend more code vectors for? To find the answer, it should be recalled that a variable-rate system is used here, and, the reproduction vectors are assumed to be coded into binary strings of varying lengths according to their entropy. Thereafter, the third conclusion is that test 8 devotes more bits to the areas of discontinuity and fewer bits to areas of slowly varying intensity, while test 1 uses more bits for the slowly varying intensity such as background. In other words, test 8 favors infrequent vectors while test 1 favors frequent vectors.

One application of the above conclusions may be the consideration of right parameters selection for the different subband with different energy distribution. For example, the parameters combination of test 1 may be good for the lowest subband because it has a relatively flat intensity distribution, while the parameters combination of test 8 may be preferred for the higher subband which usually has a peaky energy distribution.

Figure 2.15 to Figure 2.21 are the pictures supporting the above conclusion. Figure 2.16 and Figure 2.17 have the same SNR, but Figure 2.17 used much more code vectors to code. Figure 2.18 and Figure 2.19 show that the parameters combination of test 1 in

Table 2.1 perform better for the subband with a relatively flat intensity distribution, while the parameters combination of test 8 in Table 2.1 are more suitable for the subband with a peaky intensity distribution. Figure 2.20 and Figure 2.21 back up the same conclusion as the last pair of pictures do, but the image from LH subband of first level pyramid is used instead of LL band in the previous two.

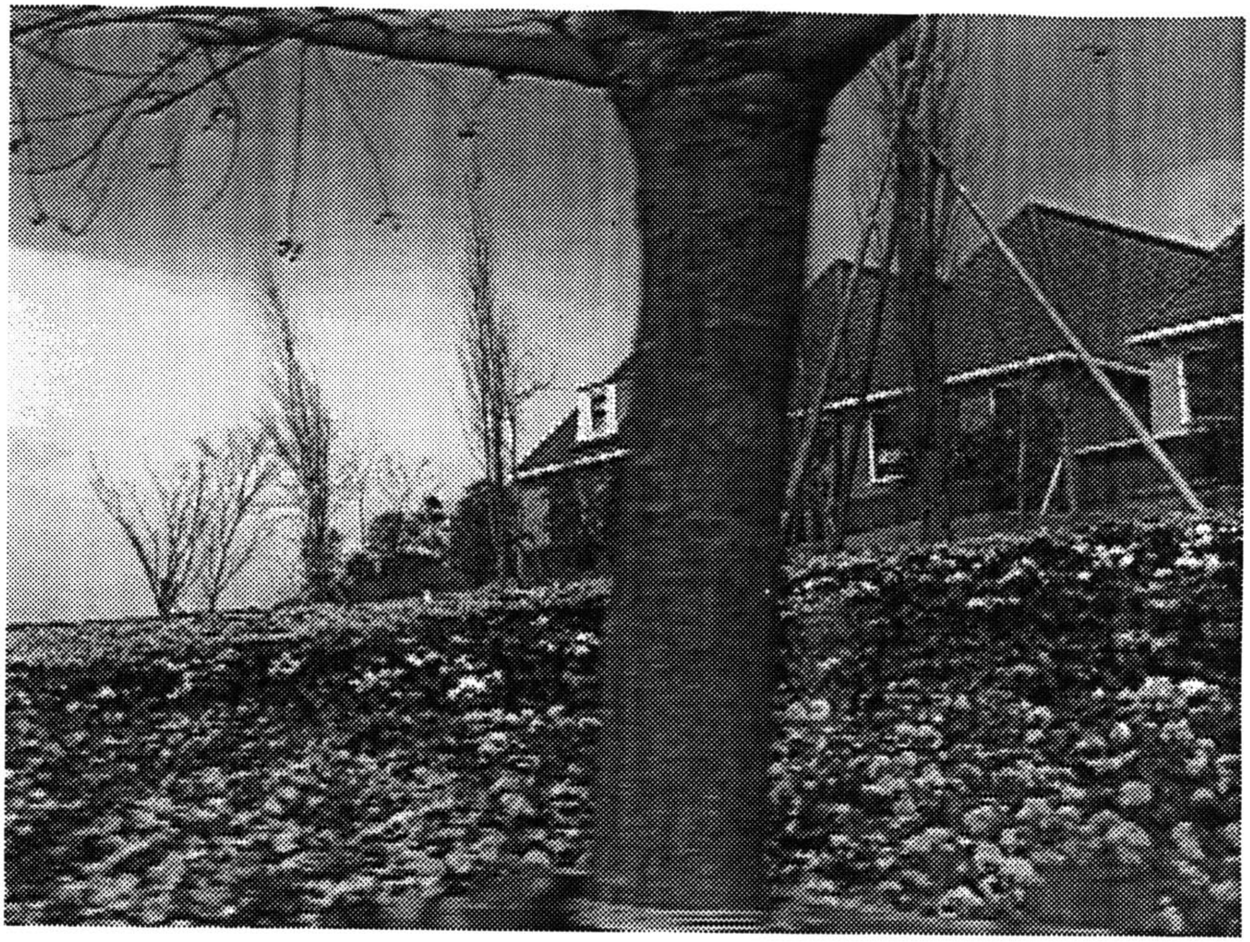

Figure 2.15: The Original Picture of Flower Garden

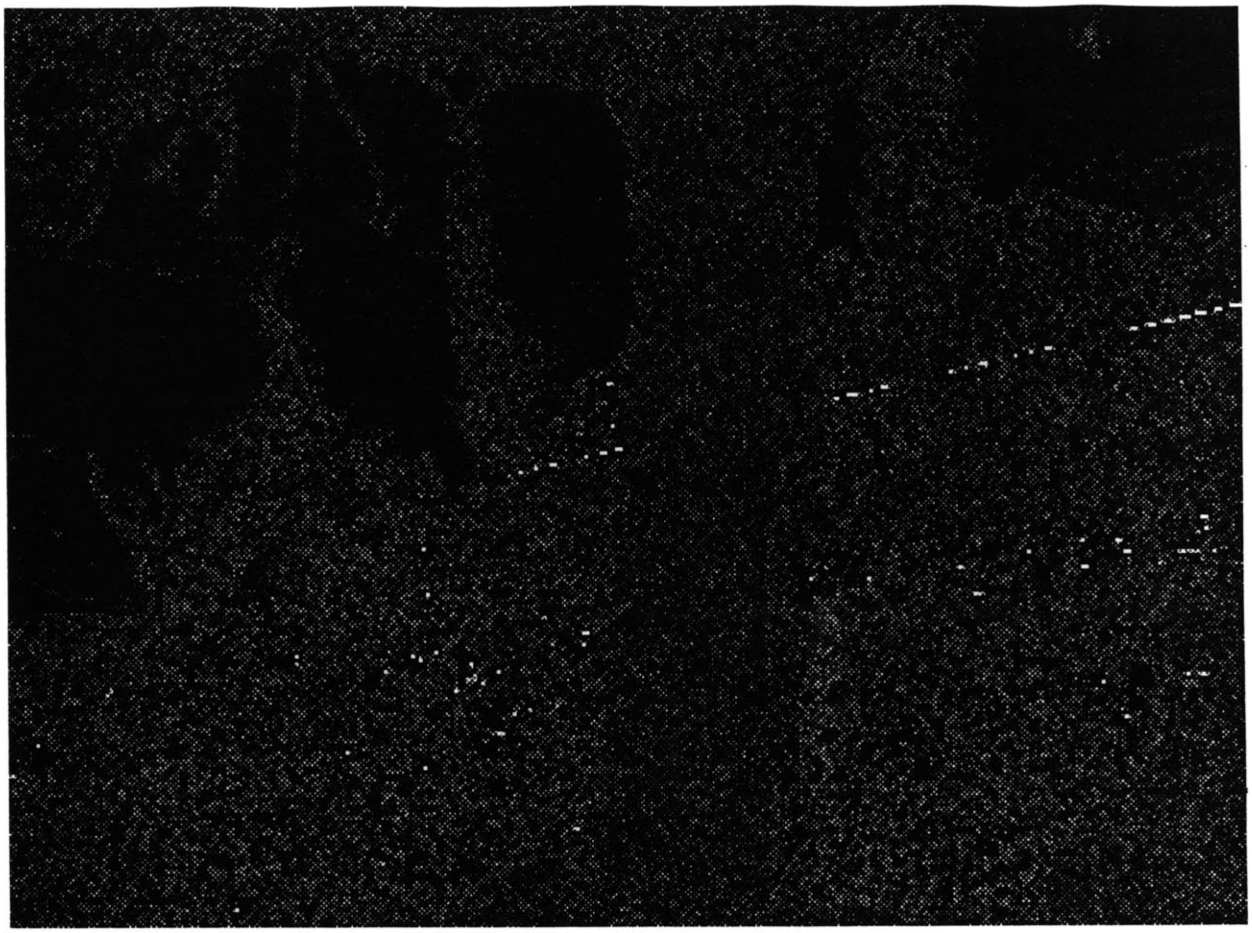

Figure 2.16: The picture shows the error pixels of the reconstructed image from test 1 in Table 2.4. (The error image has been gainbiased by 10.) Error pixels are easily seen to concentrate in detailed regions.

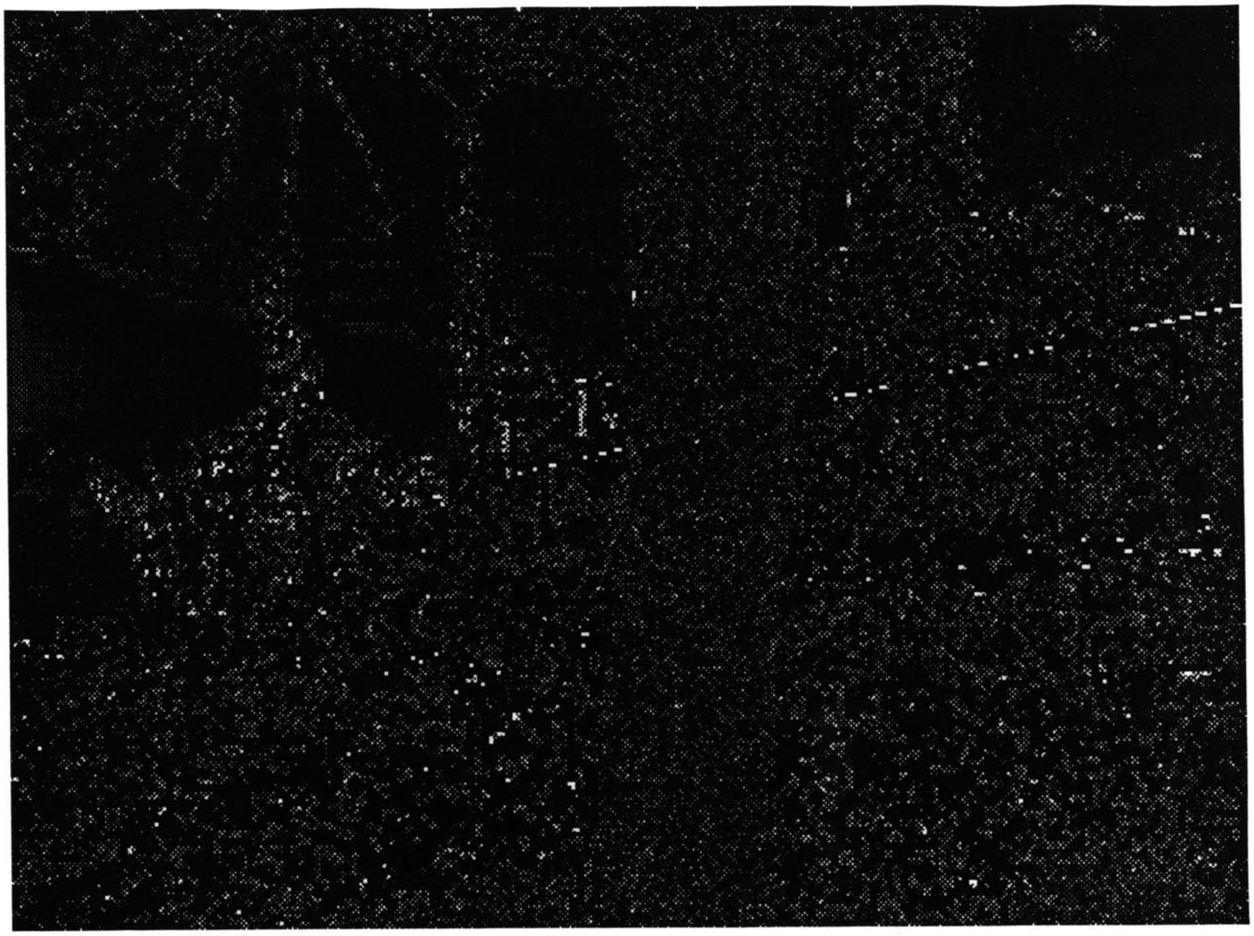

Figure 2.17: The picture shows the error pixels of the reconstructed image from test 8 in Table 2.4. (The error image has been garnbiased by 10.) Error pixels appear more in the slowly varying intensity areas, like the trunk of the tree and the sky. Meanwhile, there are more peaky error pixels than that in 2.16. However, there are less error pixels in the area of flowers.

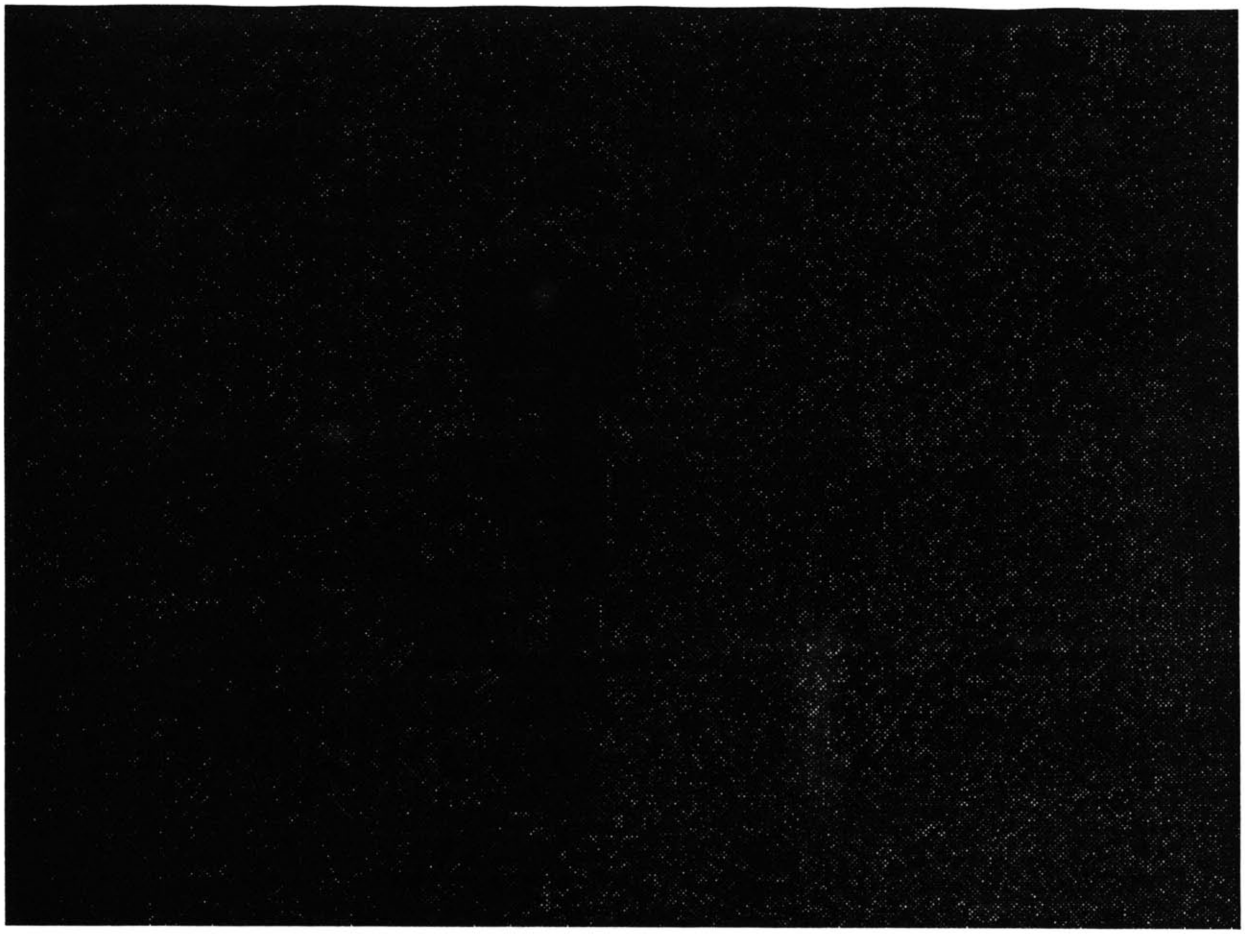

Figure 2.18: The picture shows the error pixels of the reconstructed image of lowest subband of the first level pyramid of the image 2.8. The parameters combination in test 1 of Table 2.1 is used with the codebook size = 3600. It shows much less error than next picture 2.19 which has the same codebook size but using the parameters combination in test 8 of Table 2.1. This proves that the parameters combination of test 1 is good for the subband with relatively flat intensity distribution.

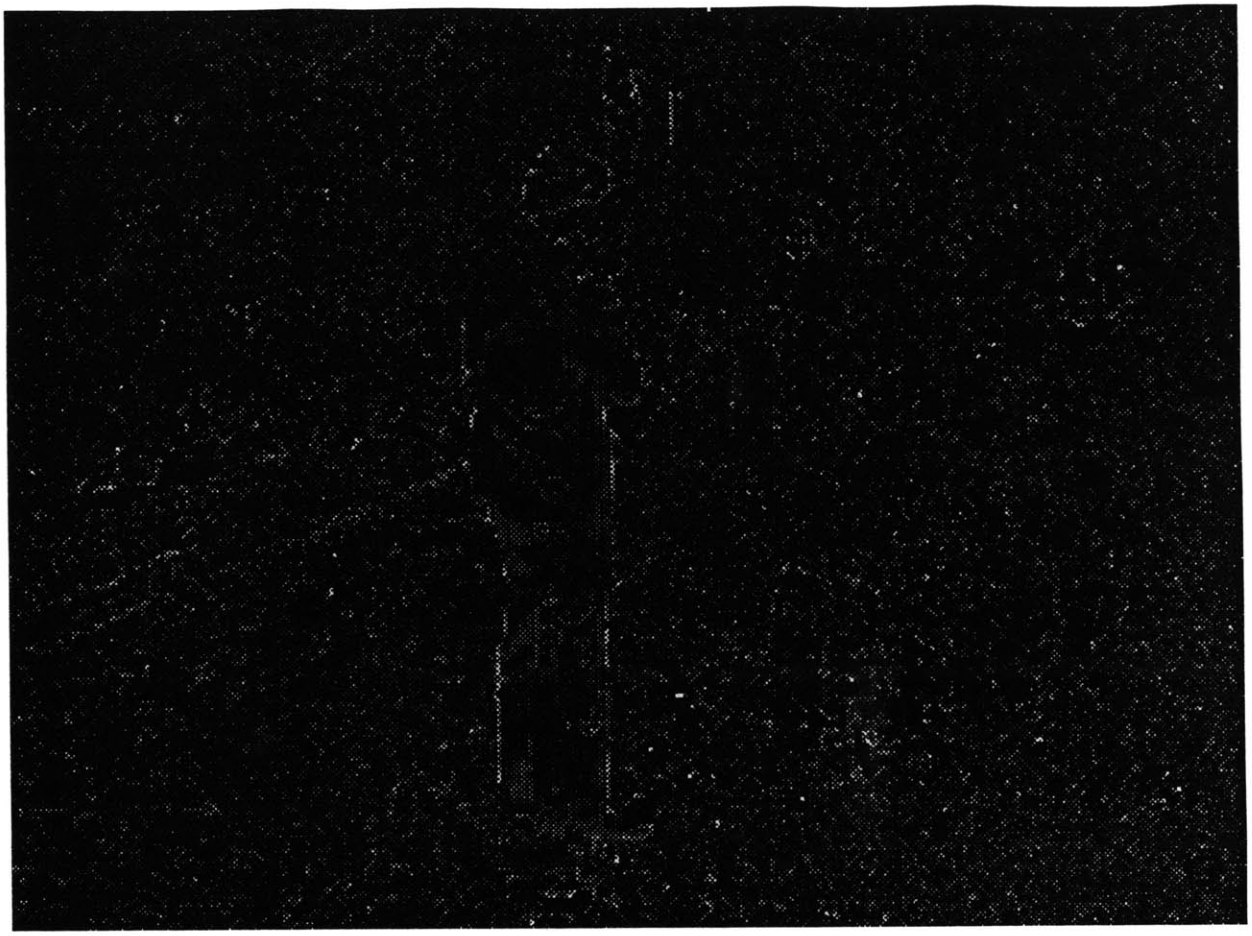

Figure 2.19: The picture shows the error pixels of the reconstructed image of lowest subband of the first level pyramid of the image 2.8. The parameters combination in test 8 of Table 2.1 is used with the codebook size = 3600. It shows much more error than previous picture 2.18 which has the same codebook size but using the parameters combination in test 1 of Table 2.1. This proves that the parameters combination of test 8 is not good for the subband with relatively flat intensity distribution.

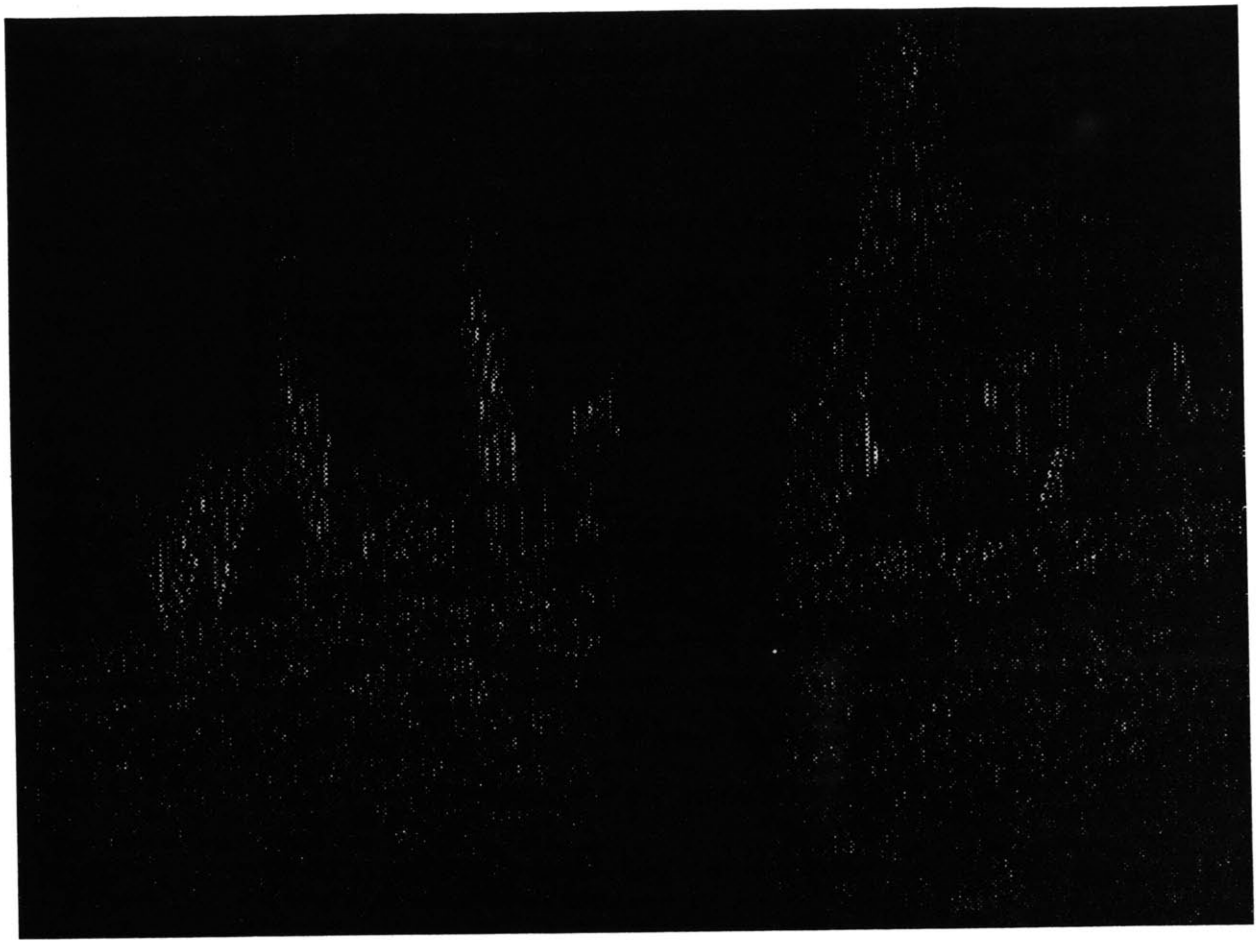

Figure 2.20: The picture shows the error pixels of the reconstructed image of LH subband of the first level pyramid of the image 2.15. The parameters combination in test 1 of Table 2.1 is used with the codebook size = 1450. It shows much more obvious errors than next picture 2.21 which has the same codebook size but using the parameters combination in test 8 of Table 2.1. This proves that the parameters combination of test 1 is not good for the subband with very peaky intensity distribution.

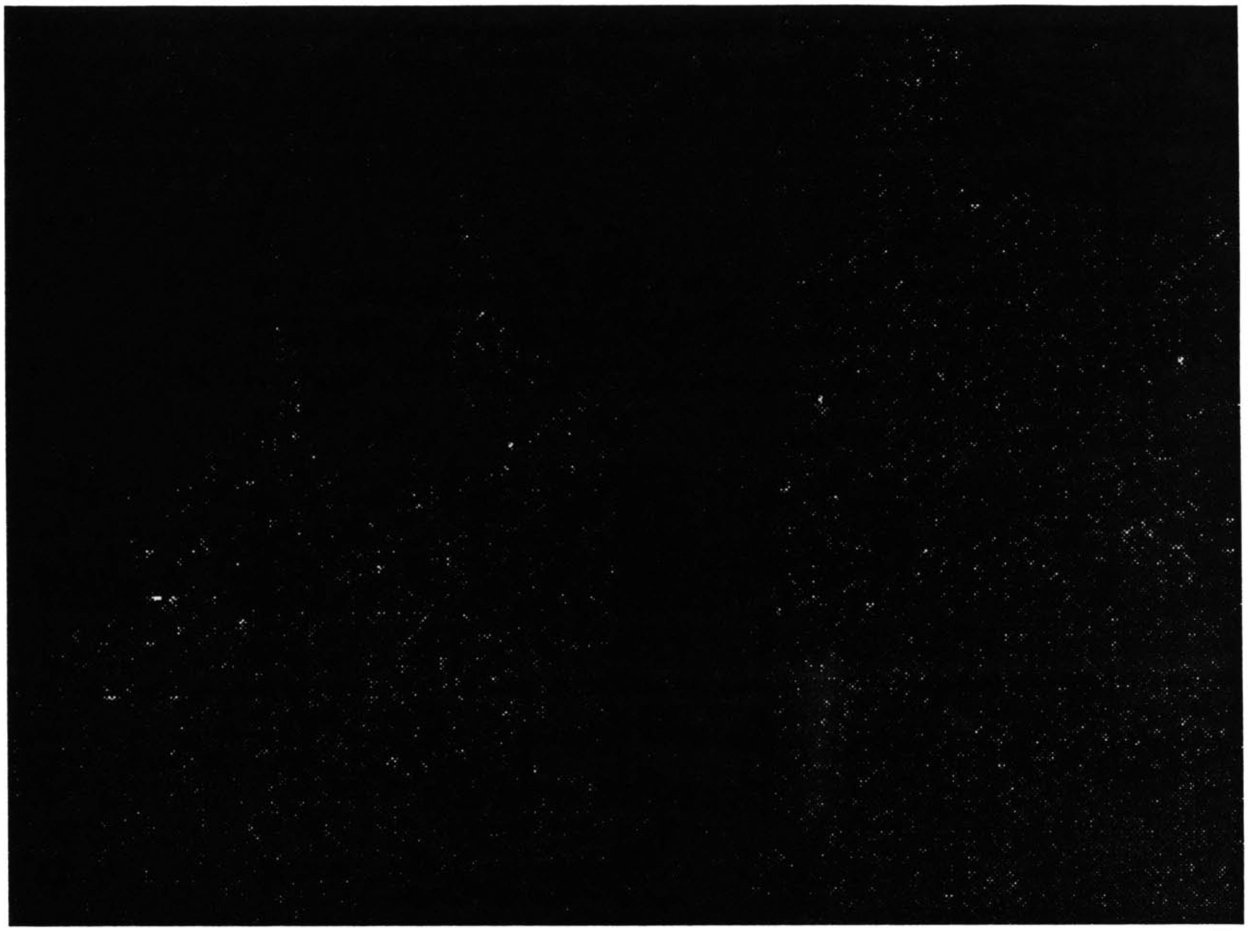

Figure 2.21: The picture shows the error pixels of the reconstructed image of LH subband of the first level pyramid of the image 2.15. The parameters combination in test 8 of Table 2.1 is used with the codebook size = 1450. It shows much less error than previous picture 2.20 which has the same codebook size but using the parameters combination in test 1 of Table 2.1. This proves that the parameters combination of test 8 is good for the subband with very peaky intensity distribution.

Chapter 3

Psychophysics and Application in Subband Coding

This chapter reviews the previous work in psychophysics about nonuniform sensitivity of the human vision system (HVS) and how these principles are applied in the subband coder design in terms of minimizing image distortion. The first section reviews the efforts towards modeling the response of the HVS. The second section uses the models to group the pictures in different subbands according to the visual sensitivity to the local energy distribution. Here, the physical meaning interpretation of subband image is employed to link the results of HVS with image data statistics among subbands.

3.1 Psychophysics in Image Coding

Figure 3.1 shows a general image processing system. Usually, the display is the final destination of processed images. Consequently, the system performance evaluations are mostly subjective. In the application of image coding, the objective criteria ,such as signal to noise ratio or mean square error (MSE), are usually employed as the measure of the reconstructed image quality. It is possible to encode an image with almost no regard for the psychophysics of the human observer. Sometimes, such a coder will produce a reasonably good image. For example, coders that operate to minimize mean square error alone often used in image processing. In general, mean square error is usually used only within a single algorithm to compare different versions of it. As mentioned in [21], mean square error provides a broad measure of picture quality. Even though it allows the tractable analysis for the image data, the limitations of the mean square error criterion as an indication of subjective image quality are well known [4] [21] [26] [8]. The advantage of mse is that is mathematically tractable, but it is not based on a visual mode However, the subjective image quality even plays a more important role in image processing system, because the displayed image still need to pass the human visual system (HVS) to target the final destination: human brains (Figure 3.1). One could argue that the human visual system is evolutionarily

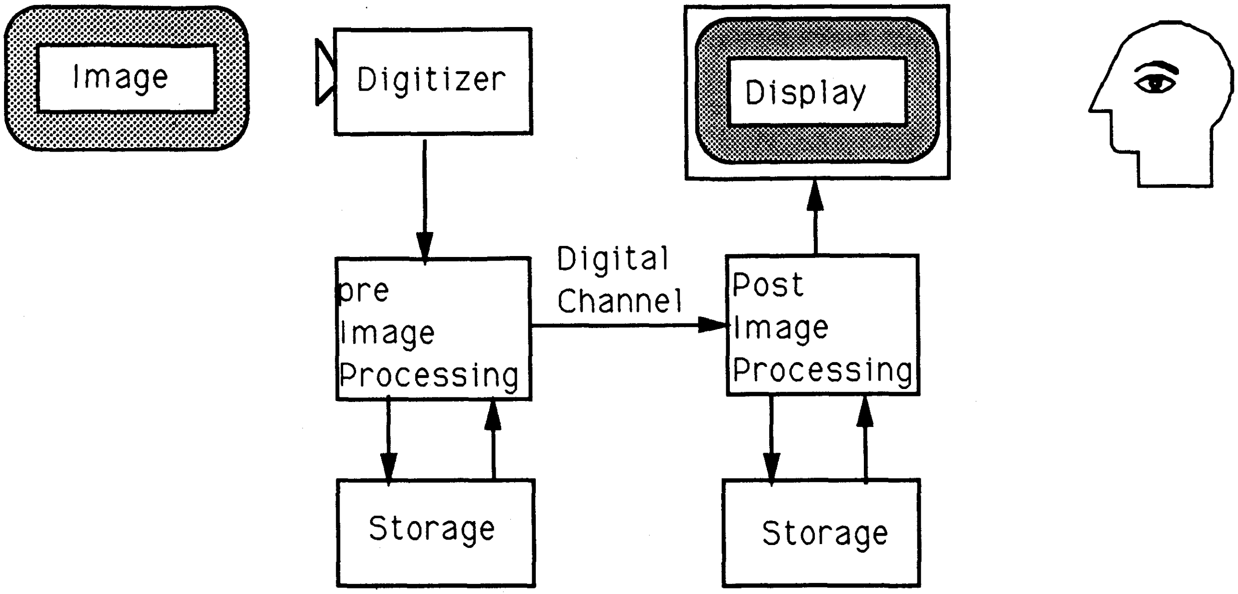

Figure 3.1: General Image Processing System

attuned to the types of images presented to it, and that the optic processing in the eye and visual cortex are in effect a coder that directly operates on scenes in the real world and analyzes them for our brains! Thus the HVS works well on natural scenes, it must follow the statistics of them, and our coder could do as well taking that into account only.

Detailed and comprehensive studies of HVS and subjective criteria of image processing system are outside the purview of this thesis. However, some study of the human vision system (HVS) will help us to understand what information in the visual input is necessary or most important in order to code,transmit and reconstruct image sequences. Especially, we will see how our subband coding structure is conveniently configured for exploiting the HVS sensitivity variations with spatiotemporal frequency.

We try to apply all of available knowledge about the psychophysics of human vision and to develop a model of the HVS and corresponding distortion measure that yields the closest possible agreement between calculated distortion and subjective picture quality. These models might attempt to account for:

- Variations in sensitivity with spatial frequency and direction.
- Non-linearity of the HVS.
- Threshold effects which render distortions below a certain intensity imperceptible.
- Masking of distortions around edges in an image resulting from the HVS preoccupation with the dominant edges in space and time
- The enhanced perceptibility of error signals which are highly structured in the spatial domain, such as the blocking artifacts of DCT codecs or contouring which results from low rate scalar quantization (see [19], Chapter 4, for example).

Such detailed modeling has been studied [4] [21] [20] [41] [3] [6], however, a definitive HVS model has not yet been determined.

Butera [5] summarized the recent studies on physiology and psychophysics [18] [29] [37] [30] [25] [24] [45], and made the following points:

- The HVS records chromatic stimuli from receptor with an inherently longer time constant and therefore at reduced temporal acuity relative to luminance.
- The retinal signals on which the HVS builds its visual representations appear to be localized in both space and spatial frequency.
- Early visual processing appears to take place in a number of separate channels which differ in their selectivity to spatial and temporal frequency components.
- The information in these channels is further processed, in part, by simple cortical cells which also appear to be orientation selective.

We are interested in how to use these points as a guide for matching the specifics of the subband coder described in chapter 2 to the performance of the HVS.

3.2 HVS Based Bit Allocation

One of the advantages which the subband transforms enjoy is the local characterization of image data. This is illustrated in a statistical point of view in Figure 2.10 and Figure 2.11 in section 2.1.2. Here we try to characterize the localized, oriented image features in subbands in a manner similar to that previously described for early visual processing in the HVS.

In section 2.1.1, we partitioned a pair of image into a set of 20 subbands which compose three level spatial pyramid (Figure 2.2. The image data in each level pyramid may be given physically meaningful interpretations. We take the 8 components of the first level pyramid in Figure 2.2 as an example to show what this means. The four temporally low passed subbands can be thought of as isolating energy from stationary objects. The four temporally high pass subbands hold the energy due to the oriented movement in the scene. These interpretations can be used to give the following intuitive meanings to the 3D subbands:

LLL	stationary ’blurs’
LLH	stationary vertical edges[a]
LHL	stationary horizontal edges
LHH	stationary diagonal edges
HLL	moving ’blurs’
HLH	moving vertical edges
HHL	moving horizontal edges
HHH	moving diagonal edges

[a] note that high pass filtering horizontally accents vertical edge detail

These physical interpretations offer us a chance to relate the human visual sensitivity to the energy distribution in the subbands. Thereafter, we may be able to find out the non-linear visibility of distortion in the individual subbands.

Based on previous work [32] [42] [43] [13] [14], Butera [5] proposed a relative hierarchy for the sensitivity to quantization distortion of the spatiotemporal subbands and ran a series of subjective tests to verify the results.

For the subbands resulting from spatial bandsplitting, the hierarchy is, in order of most sensitive to least sensitive (referring to Figure 2.2:

- *LL*
- *LH* and *HL*
- *HH*

For the spatiotemporal subbands, the order of most sensitive to least sensitive is (referring to Figure 2.2):

- *LLL*
- *LHL, LLH,* and *HLL*
- *HLH* and *HHL*
- *LHH* and *HHH*

The shortcoming of this hierarchies is that there is no information about the relative visual sensitivity crossing different pyramid levels. However, we may intuitively assume that the subbands with the same physical meaning in a lower level have a higher sensitivity than the ones in a higher level. The results above are by no means absolute or conclusive. However, we could use them as perceptual guidelines for the distribution of the quantization distortion among the components. And also, we are going to develop a bit allocation algorithm in next chapter, in which the MSE is used as distortion measurement. It was mentioned before that objective metrics do an incomplete job of predicting the subjective quality of an image. Therefore, we may want to compare the results from bit allocation algorithms, and then see if there is any motivation to modify the results by using the facts we get in this chapter.

Chapter 4

Bit Allocation & Pre-filtering

This chapter discusses two topics: bit allocation and prefiltering. Because of the interaction between input filtering and encoding, we choose to address bit allocation and image prefiltering jointly. The first part of the chapter deals with the development of bit allocation and its application to low bit rate image coding. The second part of the chapter explores the role the prefilter plays in the image coding system in terms of subjective and objective criteria.

4.1 Bit Allocation

4.1.1 A Bit Allocation Algorithm development

We begin to explore the problem of allocating bit rate among the component signals of subbands such that the mean square error in the reconstructed image is minimized for a prescribed total channel bit rate. A codec's performance is then characterized in terms of a channel bit rate versus mean square distortion characteristic. Even though there are shortcomings of mean square error as a distortion measure, this performance function proves to be instrumental in optimizing our system. At least, if not optimal, the resulting bit rate allocation is adaptive according to the input statistics.

The M subband encoder and decoder system in Figure 2.4 are referred to in the development of our bit rate allocation algorithm. We suppose that the transform, $\mathcal{T}$, splits an $L \times N$ pixels image into M components such that the total number of samples in all of the M component signals is $L \times N$. There may not be the same number of samples in each component signal, so we define s_k to be the number of samples in the kth component signal, expressed as a fraction of the total number of samples, $L \times N$. Then

$$\sum_{k=1}^{M} s_k = 1. \tag{4.1}$$

If the rate, in bits per component sample, allocated to the kth component signal is denoted r_k, then the total channel bit rate, R in channel bits per image pixel, is

$$R = \sum_{k=1}^{M} s_k r_k. \tag{4.2}$$

We may also relate the total mean square distortion in the reconstructed image to the distortion incurred by the individual component codecs using the *variance preserving* property, namely,

$$D = \sum_{k=1}^{M} d_k. \tag{4.3}$$

This relation between component codec distortion and total distortion may be generalized slightly by assigning relative weights, w_k, to each component distortion. A weighted distortion is potentially useful in attaining a better subjective image quality [28], since the human visual system is more sensitive to energy over a limited range of spatial frequencies (Chapter 3). The total weighted distortion is still a linear combination of the component distortions,

$$D = \sum_{k=1}^{M} w_k d_k. \tag{4.4}$$

Even though the implications of unequal weighting will not be explicitly expressed in the following mathematical derivation, we introduce the notation here, so that the more general allocation problem may be presented.

The final ingredient we need to formulate the allocation problem is the rate versus distortion characteristics of the component codecs. The component codec model in Figure 2.4 assumes a form similar to any other source coding problem that a encoder/decoder (codec) pair must be found for transmitting a random signal over a channel of finite capacity. In fact, we see that an M-component codec design contains within it M separate codec designs. The component codecs might be simple scale quantizers, or sophisticated spatial domain transform or subband codecs.. In our case is VQ. Not wishing to commit ourselves to a particular type of component codec at the moment, we assume only that we have component codecs which allow a choice of encoding rate and incur less mean square distortion at higher encoding rates. Then each component codec may be characterized by a *quantizer function*, $k_k(r_k)$, which expresses mean square distortion of the kth component signal as a non-increasing function of the rate allocated to the kth component codec.

Suppose now that we wish to design our component codec to achieve a desired total channel bit rate, while minimizing the total mean square distortion. This is *fixed rate bit allocation* problem which may be stated

$$min\ D = \sum_{k=1}^{M} d_k. \tag{4.5}$$

or equivalently,

$$min\ \sum_{k=1}^{M} k_k(r_k). \tag{4.6}$$

subject to

$$D = \sum_{k=1}^{M} s_k r_k \leq R \qquad r_k \geq 0, \quad \forall k. \tag{4.7}$$

where R denotes the desired total rate.

In this thesis, the *quantizer function* $k_k(r_k)$ is specialized to *negative exponential quantizer functions*, i.e., quantizer functions of the form ([19] page 491)

$$r_k = -\frac{1}{2}\log_2\left(\frac{\varepsilon^2\sigma_k^2}{d_k}\right), \tag{4.8}$$

and corresponding inverse quantizer functions

$$d_k = \varepsilon^2 \sigma_k^2 2^{-2r_k}, \tag{4.9}$$

where ε denotes the quantizer performance factor. We assume the quantizer performance factor is constant in our work.

Deriving a fixed rate allocation algorithm for the exponential quantizer functions is simply a matter of following the procedures outlined next. In this case, formula 4.6 could be rewritten as

$$min \sum_{k=1}^{M} \varepsilon^2 \sigma_k^2 2^{-2r_k}. \tag{4.10}$$

Given that the goal is to minimize the error in equation 4.10 under the constraint defined in equation 4.7, we use Lagrange multipliers to obtain:

$$\frac{\delta}{\delta r_k} \left[\varepsilon^2 2^{-2r_k} \sigma_k^2 \; - \; \lambda \left(R \sum_{k=1}^{M} s_k \; - \; \sum_{k=1}^{M} r_k s_k \right) \right] = 0$$

$$-2\varepsilon^2 2^{-2r_k} \ln 2 \sigma_k^2 \; + \; \lambda s_k = 0$$

$$2\varepsilon^2 \ln 2\sigma_k^2 = \lambda s_k 2^{2r_k}$$

$$r_k = \frac{1}{2}\log\left(\frac{2\varepsilon^2 \ln 2\sigma_k^2}{\lambda s_k}\right) \tag{4.11}$$

To find λ, we combine equation 4.1 and equation 4.2 to get

$$R\sum_{k=1}^{M} s_k = \sum_{k=1}^{M} r_k s_k. \tag{4.12}$$

Then we apply equation 4.12 to equation 4.11:

$$R\sum_{k=1}^{M} s_k = \sum_{k=1}^{M} s_k \frac{1}{2}\log\left(\frac{2\varepsilon^2 \ln 2\sigma_k^2}{\lambda s_k}\right)$$

$$2R\sum_{k=1}^{M} s_k = \sum_{k=1}^{M} \log\left[\left(\frac{2\varepsilon^2 \ln 2\sigma_k^2}{\lambda s_k}\right)^{s_k}\right]$$

$$2R\sum_{k=1}^{M} s_k = \sum_{k=1}^{M} \log\left[\left(\frac{2\varepsilon^2 \ln 2}{\lambda}\right)^{s_k}\right] + \sum_{j=1}^{M} \log\left[\left(\frac{\sigma_j^2}{s_j}\right)^{s_j}\right]$$

$$\log\left(2^{2R\sum_{k=1}^{M} s_k}\right) = \log\left(\frac{2\varepsilon^2 \ln 2}{\lambda}\right) + \log\prod_{j=1} M\left[\left(\frac{\sigma_j^2}{s_j}\right)^{s_j}\right]$$

$$\log\lambda = \log\left(2^{\frac{2\epsilon^2 \ln 2}{2R\sum_{k=1}^{M} s_k}} \prod_{j=1}^{M}\left[\left(\frac{\sigma_j^2}{s_j}\right)^{s_j}\right]\right)$$

$$\lambda = 2^{\frac{2\epsilon^2 \ln 2}{2R\sum_{k=1}^{M} s_k}} \prod_{j=1}^{M}\left[\left(\frac{\sigma_j^2}{s_j}\right)^{s_j}\right] \qquad (4.13)$$

Applying equation 4.13 to equation 4.11 then gives

$$r_k = R \; + \; \frac{1}{2}\log\left(\frac{2\varepsilon^2 \ln 2\sigma_k^2}{s_k}\frac{2^{2R\sum_{k=1}^{M} s_k}}{2\varepsilon^2 \ln 2\prod_{j=1}^{M}\left[\left(\frac{\sigma_j^2}{s_j}\right)^{s_j}\right]}\right)$$

Finally, we get

$$r_k = R \; + \; \frac{1}{2}\log\left(\frac{\frac{\sigma_k^2}{s_k}}{\prod_{j=1} M\left[\left(\frac{\sigma_j^2}{s_j}\right)^{s_j}\right]}\right) \qquad (4.14)$$

4.1.2 Experiments on Bit Allocation

Table 4.1 shows the encoding tests which use the bit allocation algorithm we developed above. The test pictures is the sixth and seventh frames of the second of the Alley sequence. The assigned bit rate in each individual subband r_k is used to control the kd-tree based VQ encoder to achieve the predescribed overall bit rate. Figure 4.1 is the plot of the test result.

Test #	bit/perpixel assigned	bit/perpixel coded	bit/persecond assigned	SNR
1	0.25	0.26	0.57Mbits	27.66
2	0.5	0.35	1.15Mbits	28.53
3	0.75	0.67	1.72Mbits	31.22
4	1	0.97	2.30Mbits	33.07
5	1.25	1.32	2.88Mbits	34.82
6	1.5	1.63	3.45Mbits	36.08

Table 4.1: The Bit Allocation Test

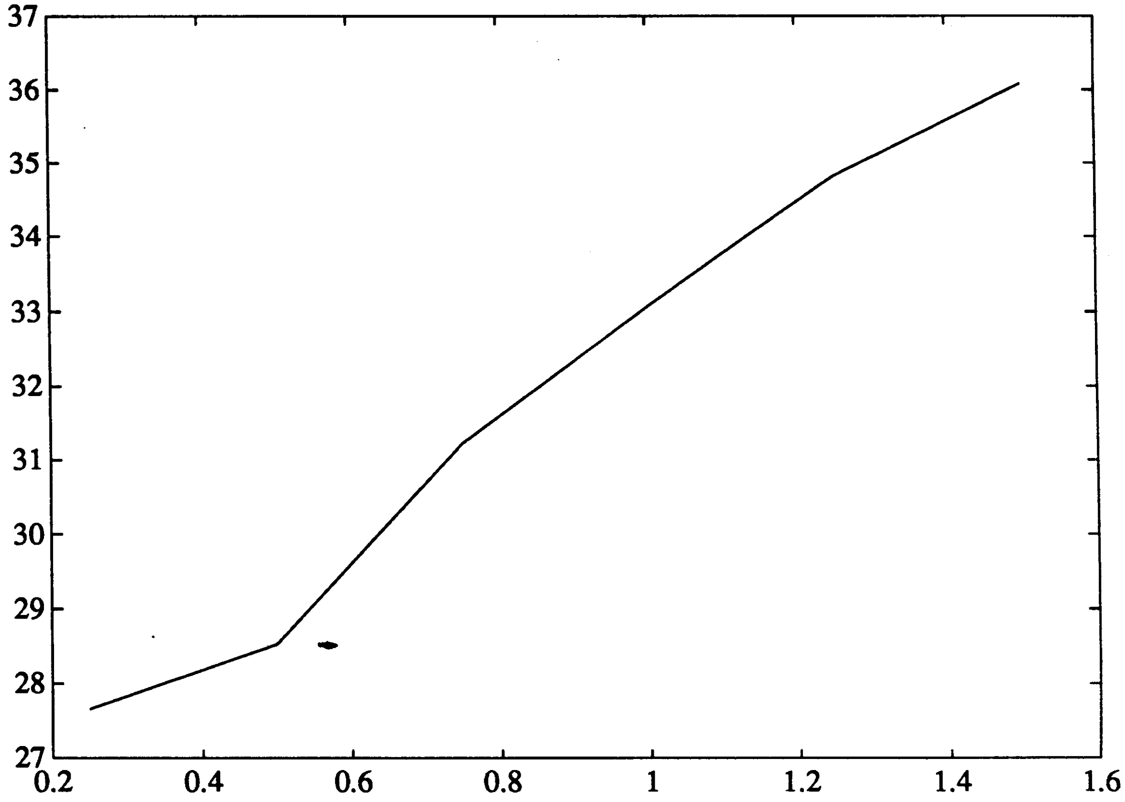

Figure 4.1: The Bit Allocation Test Result Plot

It is easily noticed in equation 4.14 that the bit rate in k's subband is related to the variance of that subband. Therefore, the overall bit allocation scheme should offer an adaptive allocation according to the input data statistics. And, the result of the adaptive allocation should also agree with the bit allocation based on HVS in chapter 2.2.4. The experiments which apply the bit allocation on pictures from different scenes show that this allocation algorithm, even though it is based on objective criterion, works pretty well with the subjective scene content. For example, we compare the bit allocation of two picture pairs, one from the sixth and seventh frame of fourth second of Alley sequence and the other from the tenth and eleventh frame of sixth second of Alley sequence. Subjectively, the visual content of the second picture pair has much more body movement. In other words, it contains more motion information. So, we wish the bit allocation algorithm could assign more bits towards both spatial and temporal high bands. It does. This is clearly illustrated in Table 4.3 and Table 4.4. The average bit rate R is set to 0.5 here. The pyramid subbands referring to Figure 2.2 have been labeled in Figure 4.2.

For the first pair of picture, 0 bit is assigned to most of the subbands in temporal high band since there is not significant much energy distributed among them. In contrast, more bits are assigned in those subbands in the second pair of pictures since there is much energy distributed there.

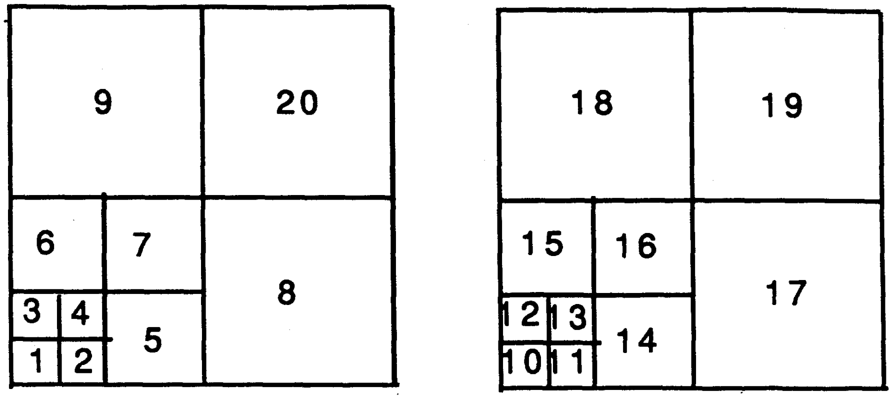

Figure 4.2: Labels of subbands in Figure 2.2.

subband #	bit/perpixel assigned	variance
1	5.99	1477.91
2	3.4	40.75
3	3.32	36.79
4	2.58	13.06
5	2.23	32.22
6	2.39	40.4
7	1.42	10.44
8	0.59	13.32
9	0.82	18.31
10	0.65	0.9
11	0.52	0.75
12	0.29	0.54
13	0.25	0.51
14	0	1.33
15	0	1.23
16	0	0.75
17	0	1.48
18	0	1.15
19	0	0.21
20	0	1.08

Figure 4.3: The Bit Allocation Test for the sixth and seventh Frame of fourth Second of Alley Sequence with $R = 0.5$

subband #	bit/perpixel assigned	variance
1	5.88	1442.57
2	2.95	24.75
3	3.26	37.94
4	2.02	6.86
5	1.26	9.55
6	1.92	23.95
7	0.59	3.79
8	0	4.07
9	0.2	8.78
10	3.3	40.42
11	2.95	24.84
12	2.72	18.03
13	2.34	10.65
14	1.87	22.32
15	1.74	18.59
16	0.92	5.97
17	0	3.31
18	0.11	7.79
19	0	0.38
20	0	0.37

Figure 4.4: The Bit Allocation Test for the tenth and eleventh Frame of sixth Second of Alley Sequence with $R = 0.5$

Figure 4,5 to Figure 4.10 shows the result pictures of bit allocation test in Table 4.1.

4.2 Pre-filter Issues in Image Coding

4.2.1 Prefiltering

In general, a coder degrades non-linearly when overloaded. That is to say the picture quality decreases rapidly as either bit rate goes down or input detail increases. Therefore, most image coding systems employ a filter to process the image (irreversible) before the actual encoding. For example, in the AT&T proposed MPEG (International Standards Organization / Moving Picture Experts Group) image sequence coding standard, an adaptive prefilter is used to reduce the bandwidth by a factor of 4, before the image data is sent to encode. The reconstructed image is then the filtered image with coding noise added. The essential idea to use a prefilter here is to remove certain amount of information (energy) from the image data, so that the job for coder may be easier. The prefiltered image will look softer since high frequency

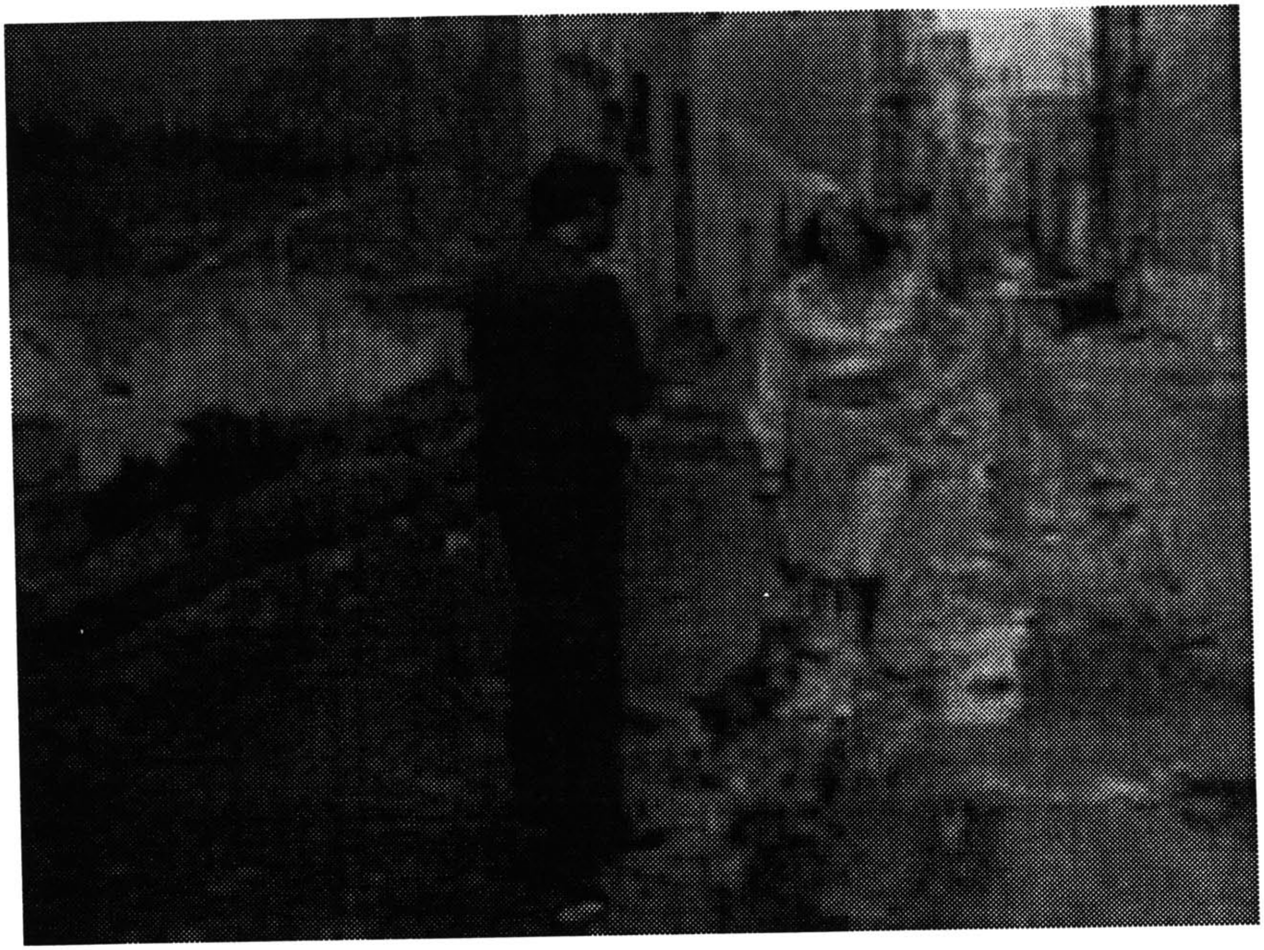

Figure 4.5: The Bit Allocation Test with $R = 0.25$ bits/per pixel

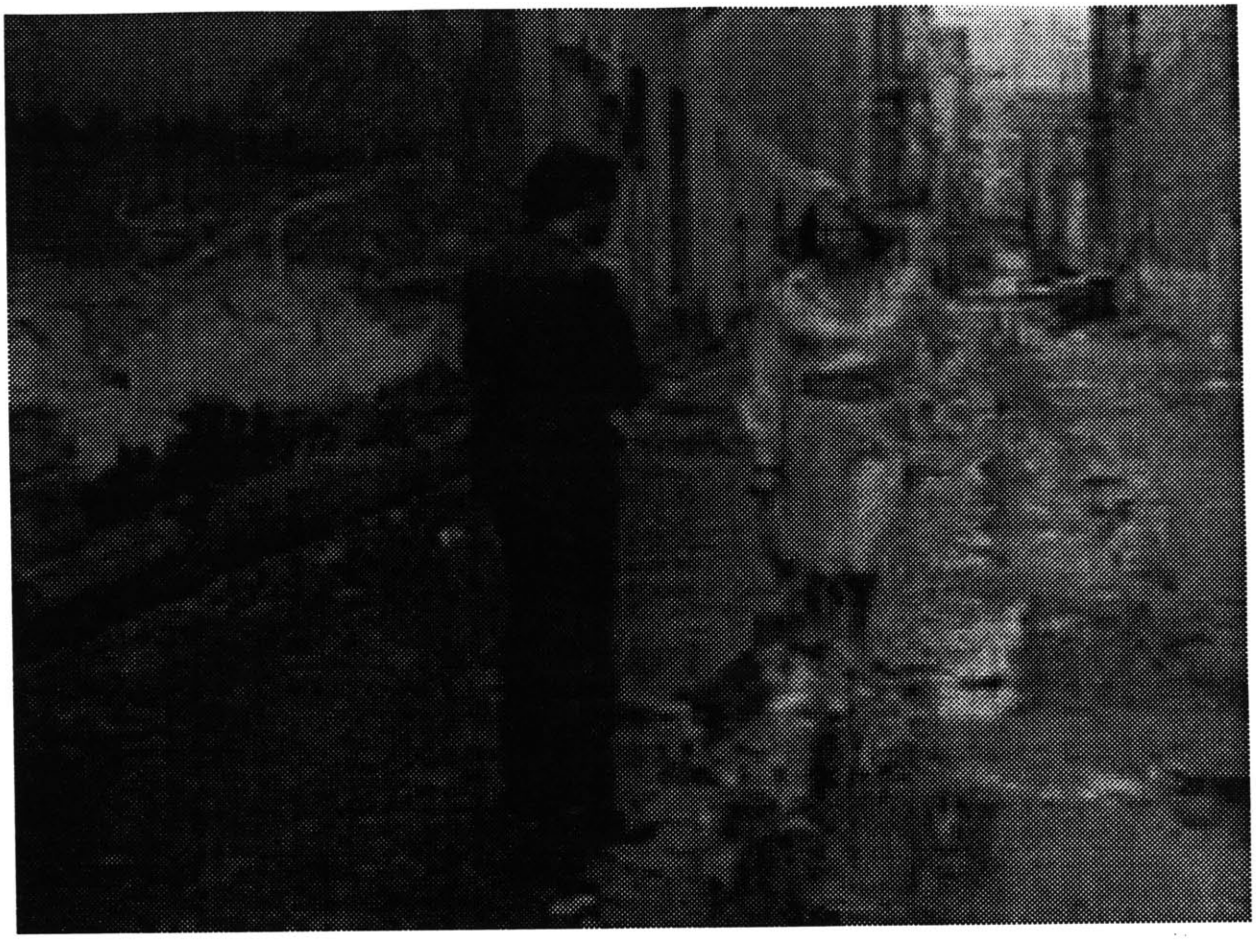

Figure 4.6: The Bit Allocation Test with $R = 0.5$ bits/per pixel

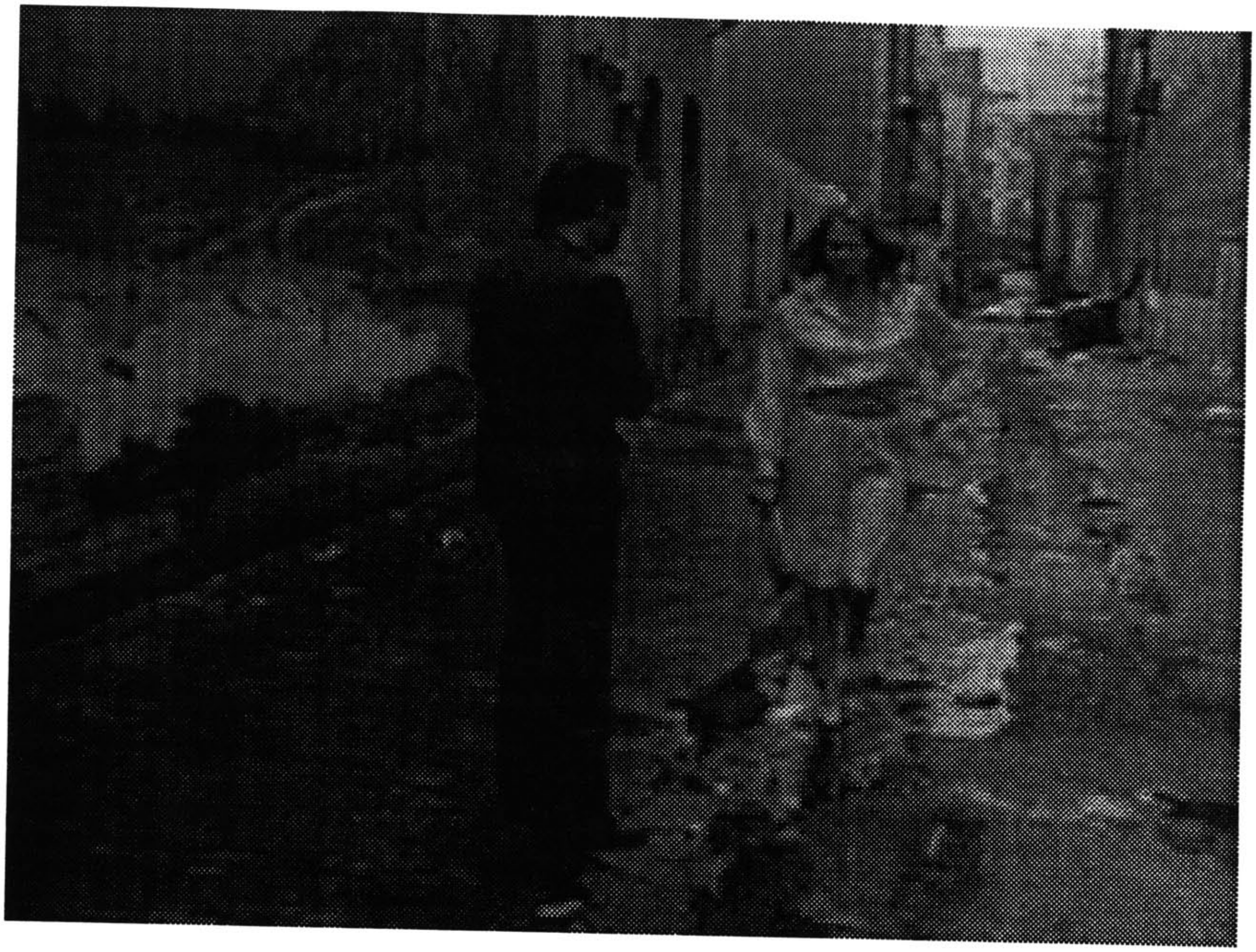

Figure 4.7: The Bit Allocation Test with $R = 0.75$ bits/per pixel

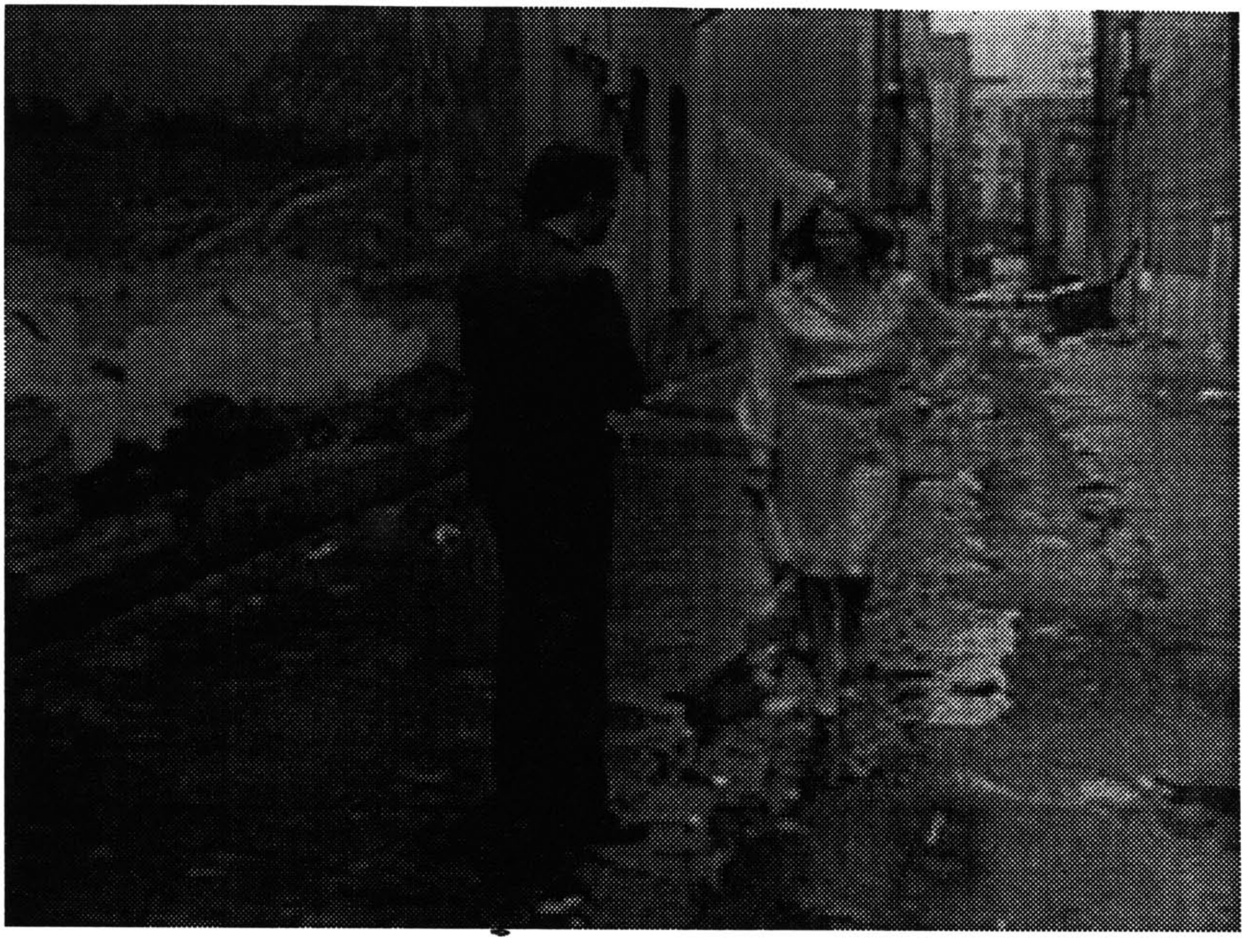

Figure 4.8: The Bit Allocation Test with $R = 1.0$ bits/per pixel

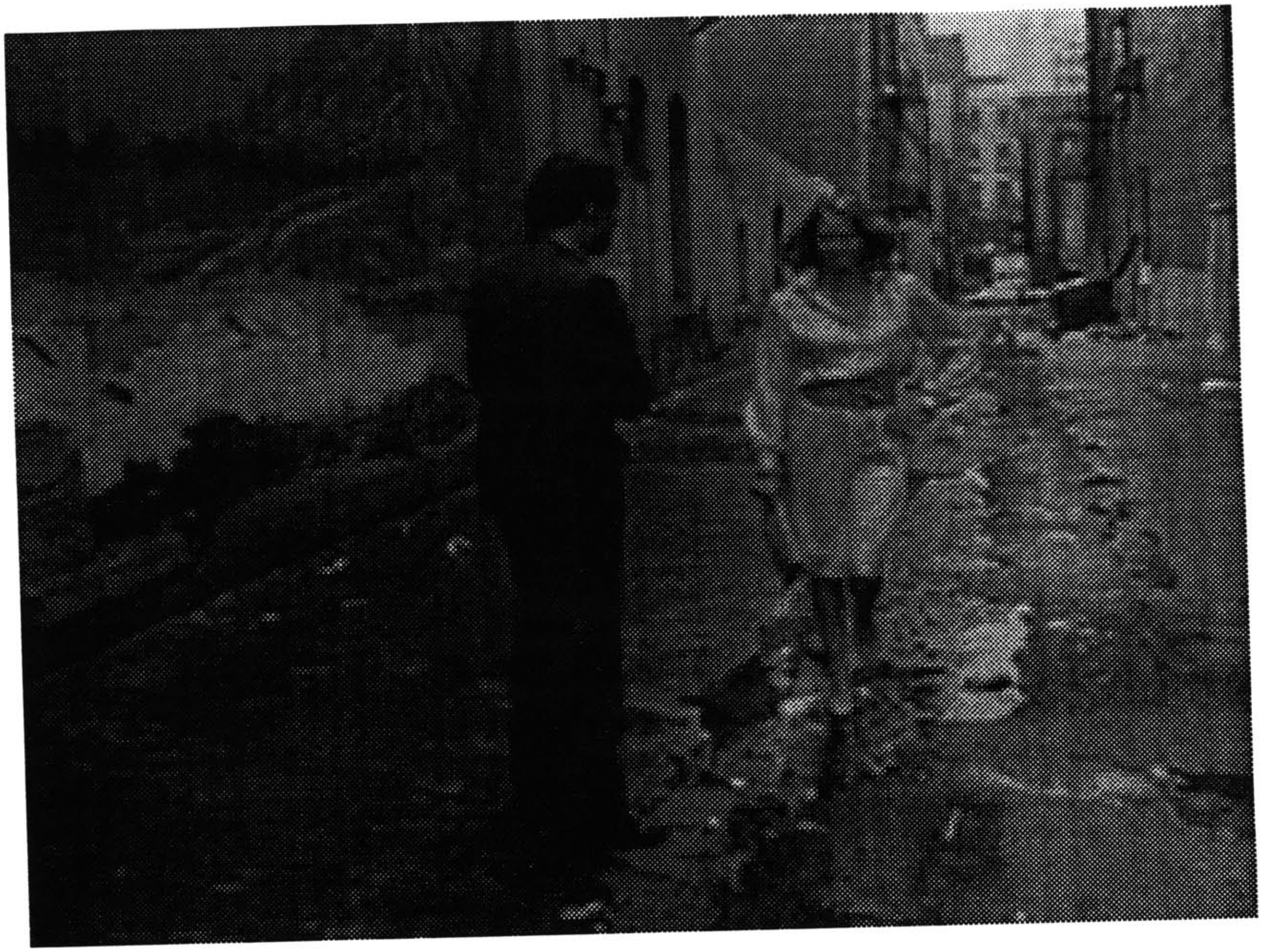

Figure 4.9: The Bit Allocation Test with $R = 1.25$ bits/per pixel

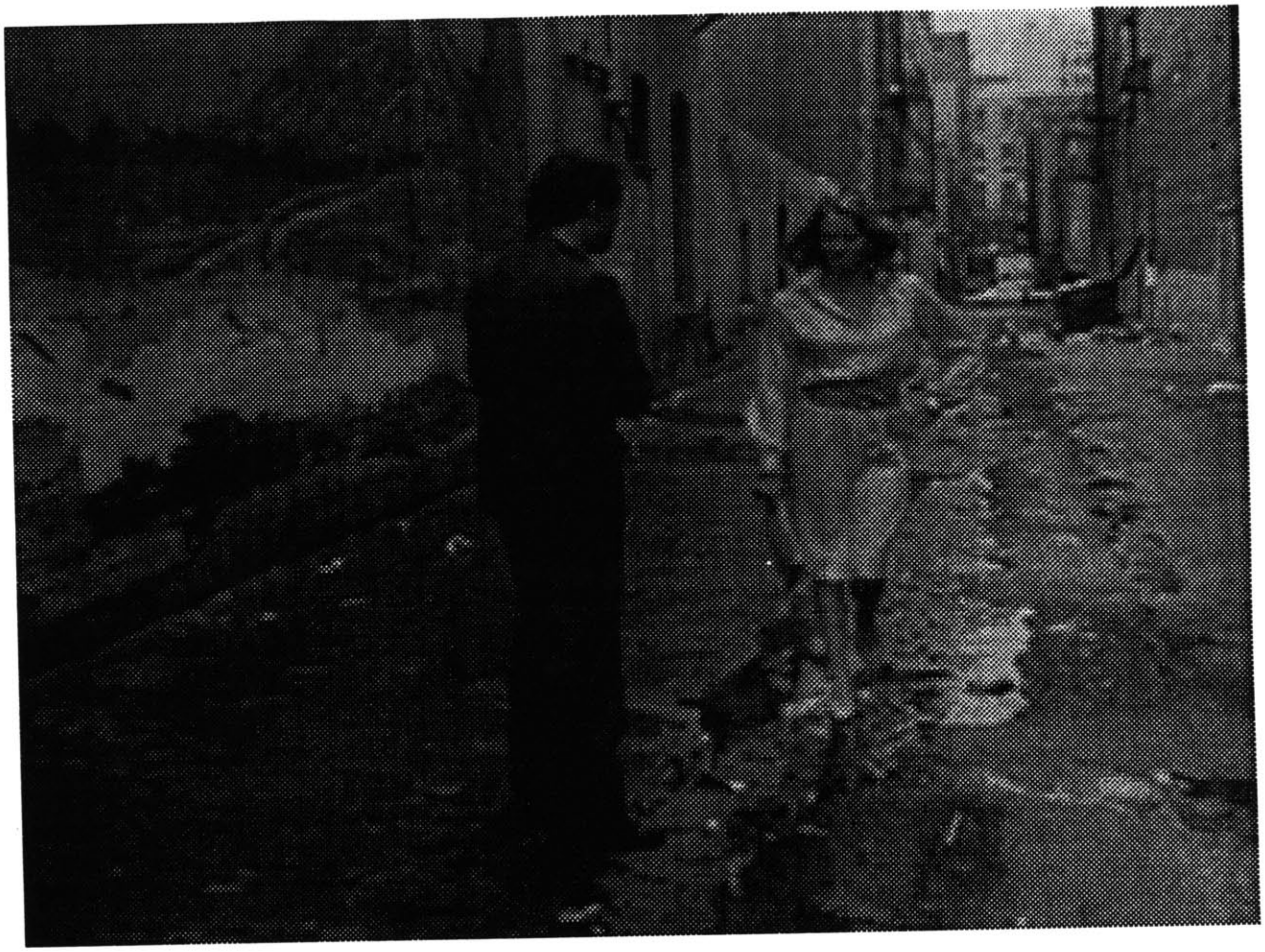

Figure 4.10: The Bit Allocation Test with R = 1.5 bits/per pixel

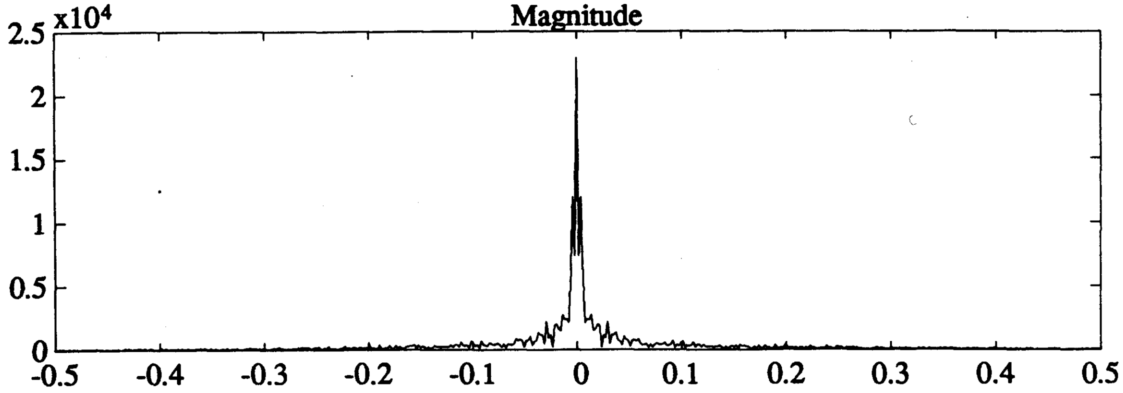

Figure 4.11: The Spectrum of a Horizontal Line of sixth frame of fourth Second from Alley Sequence.

detail is reduced. The trade-off here is between a soft picture and a noisy one. In many cases, this trade mitigates in favor of softness. We are going to discuss what character the prefilter should have and how they affect the statistical distribution of image data among subbands. Then, in next section we will conduct some experiments to see if the prefilter can make coder work better.

Even though there is no universal pdf (probability density function) for image data, Laplacian pdfs are mostly accepted as a basic model for image statistics [19]. Figure 4.11 shows the normalized frequency spectrum of a horizontal line from the sixth frame of fourth second of Alley sequence. It is easy to see that most of the energy has been concentrated at very low frequency range. This implies that the prefilter we want does not require a wide passband. As a matter of fact, it may have a narrow passband, but a very wide transition band. This will make the filter design easier.

Figure 4.12 shows the magnitude of four prefilters used in this thesis. Figure 4.13 draws the magnitude of MPEG interpolation filter which has been used as post filter in this work. We ran a test to see how these prefilters affect subband image data statistics. The original image (480 x 640) are first prefiltered by filter1, filter2, and filter3 in Figure 4.12. Then, the decimated images (240 x 320) are split into a 4 level spatial pyramid. Figure 4.14 illustrates the labeled subbands. We plot out the energy distribution of these 4 level subbands in Figure 4.15, in which the y axis is logarithmic. It can be easily noticed that the energy distribution in the lower two level pyramid is almost identical, no matter which prefilter is used to get the data. Meanwhile the energy distribution in the top two level pyramid is quite different. The energy decreases while the cut-off frequency of the prefilter gets narrow. Therefore, if we reconstruct the picture back without coding, picture prefiltered by filter3 will definitely look poorer or softer than the one prefiltered by filter1, because more information has been thrown away by filter3. Figure 4.16 shows this result.

The question raised here is if this is still true, when we code subbands and interpolate the image back to the original size.

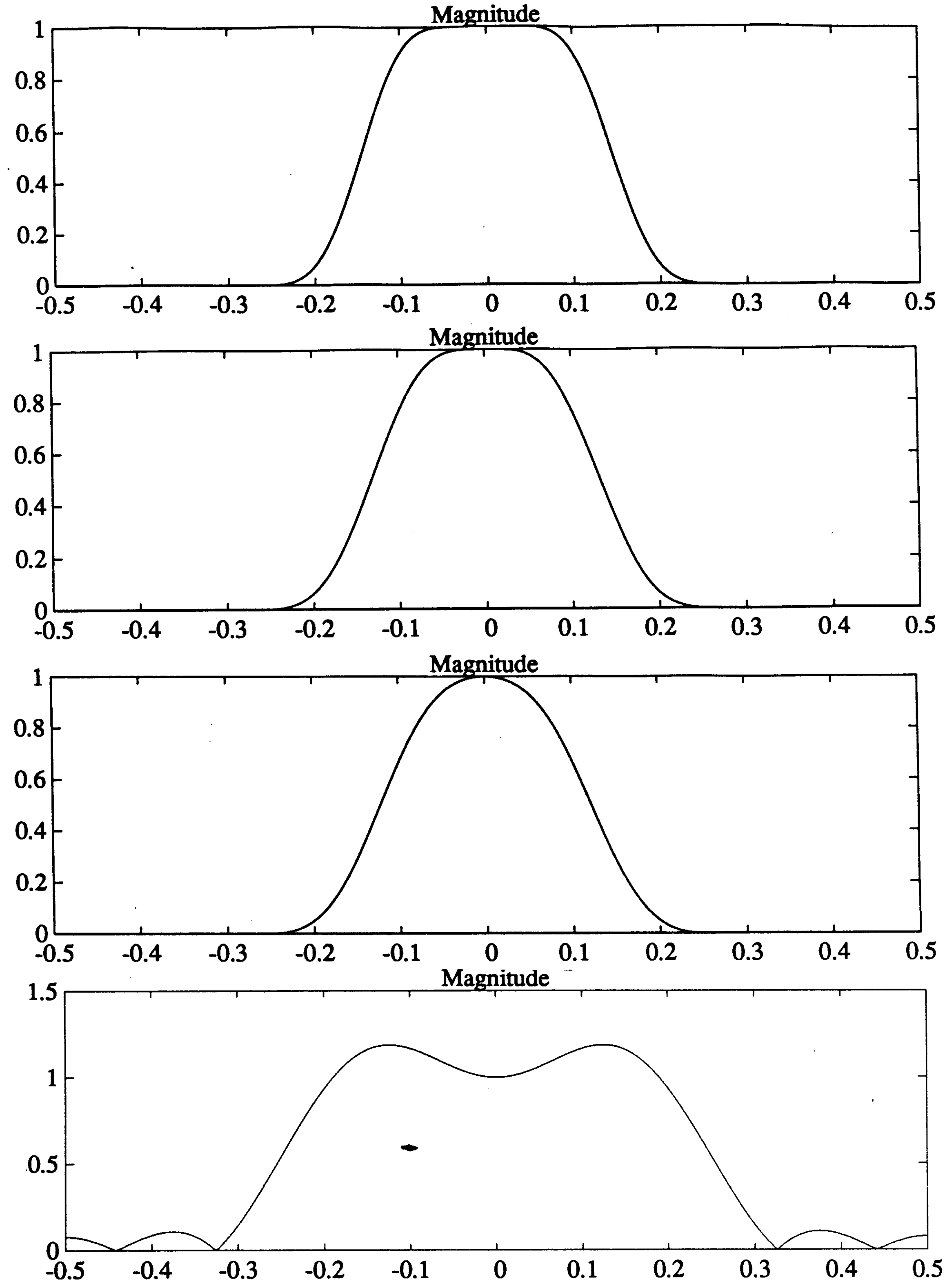

Figure 4.12: Prefilters: top: filter1, second from top: filter2, second from bottom: filter3, bottom: MPEG decimation filter.

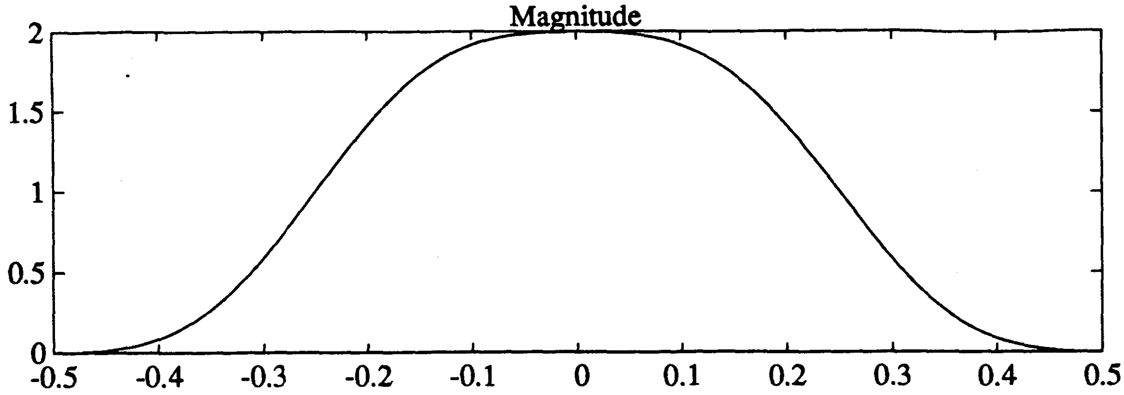

Figure 4.13: MPEG interpolation filter

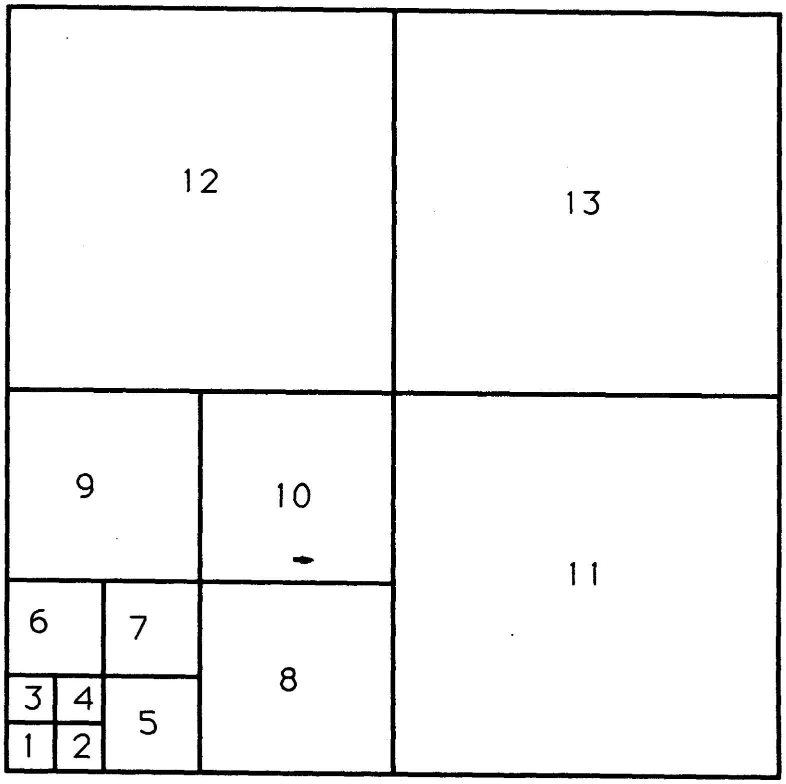

Figure 4.14: Labels of 4 level pyramid subband.

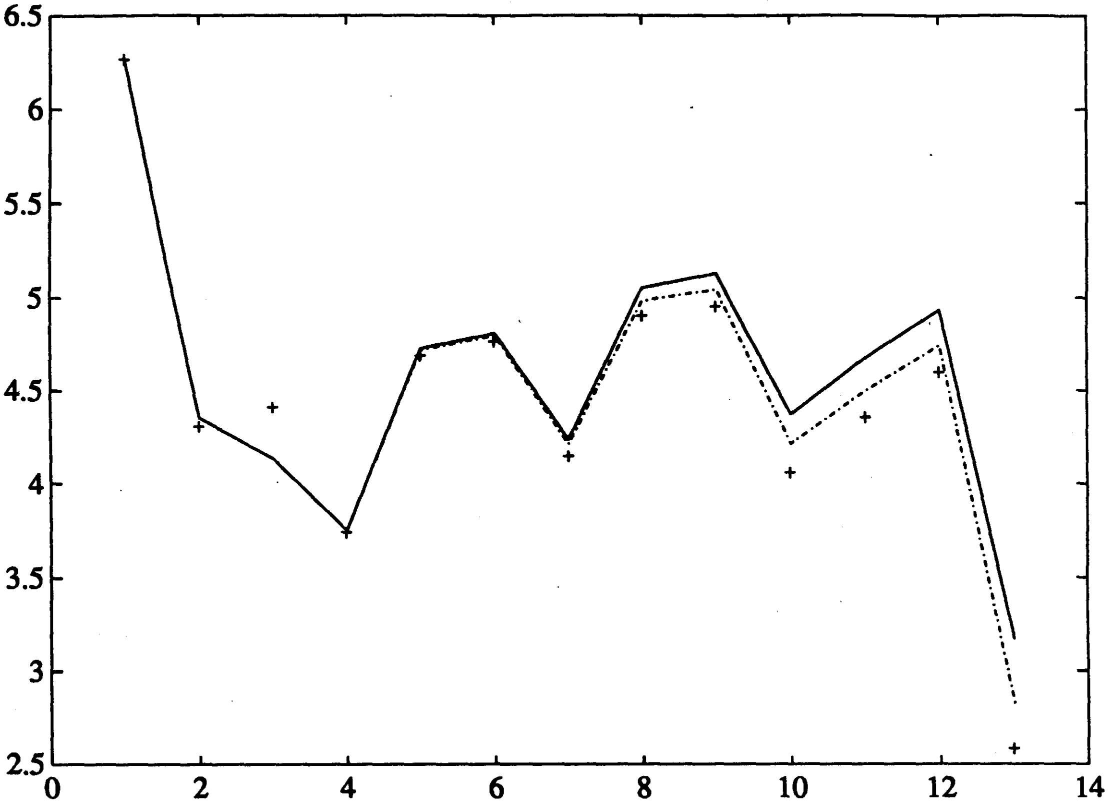

Figure 4.15: The plot-out of energy distribution among subbands. —, -.-., and +++ represent the image data prefiltered by filter1, filter2, and filter3, respectively.

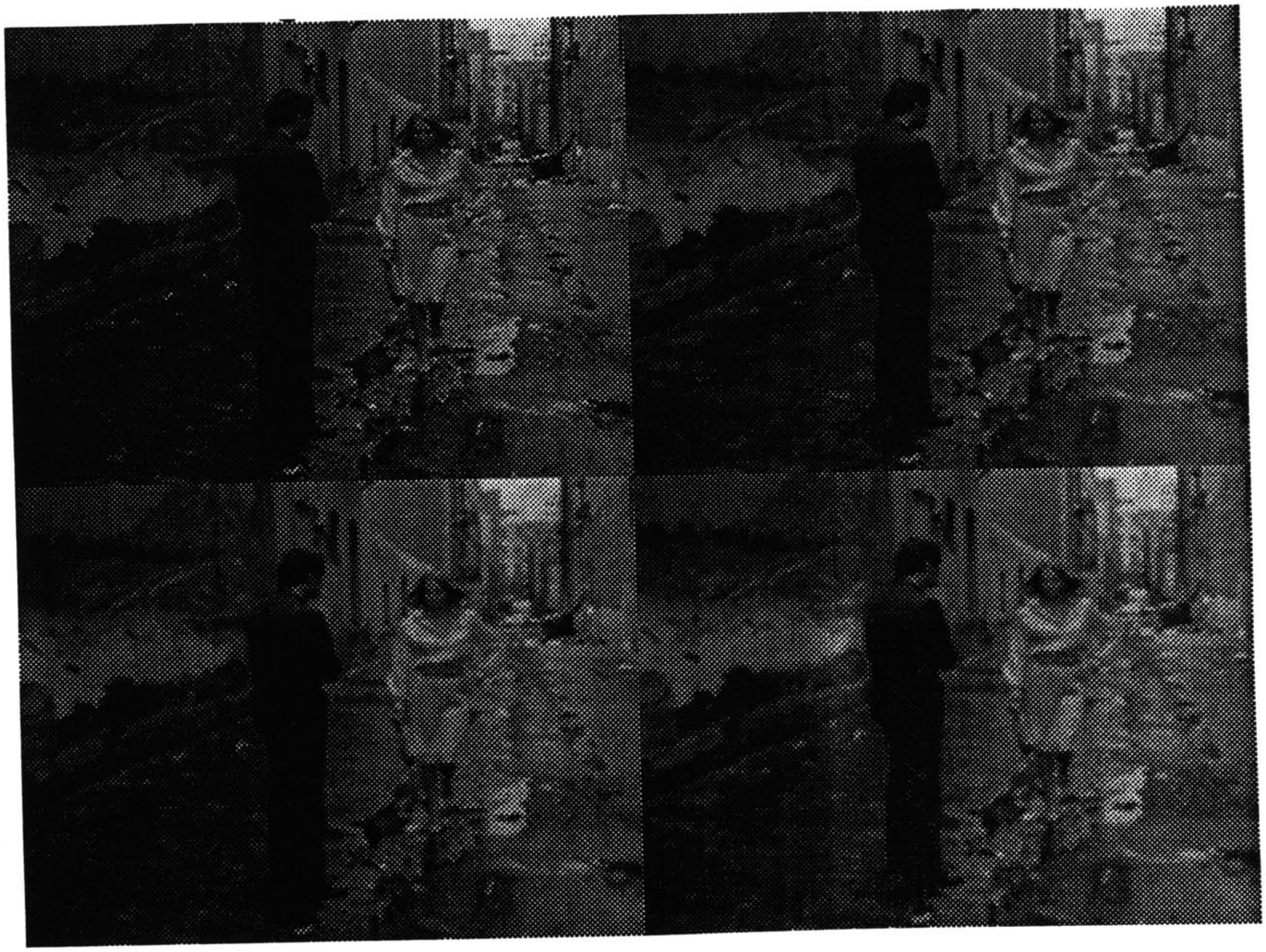

Figure 4.16: The pictures in the order of top left, top right, bottom left, bottom right represent the image prefiltered by MPEG decimation filter, filter1, filter2, filter3, respectively.

4.2.2 Picture Quality vs. Prefiltering

To verify if the different prefilter can make the coder work differently, we run though the three tests shown in Figure 4.17. The test picture pair are from sixth and seventh frame of fourth second of Alley sequence. The original picture is shown in Figure 2.8. The bit rata for both experiment SNR 2 and experiment SNR3 is set at $R =$ 0.5 bits/per pixel. (the actual coded bit rate is : $R_{MPEGdecimation} = 0.36$ bits/per pixel; $R_{f}ilter3 = 0.38$ bits/per pixel.) Let us have a look at the data in table 4.2. In the column of SNR 1, the images prefiltered by MPEG decimation filter have a higher SNR, because the filter favors higher frequency more than filter3 does and kept more spatial information. The SNR 2 column shows very interesting result. the reconstructed images prefiltered by filter3 have much higher SNR than images prefiltered by MPEG decimation filter this time, since the SNR here is mainly affected by our coder, instead of prefilter. Therefore, we can learn that some prefilters, like filter3, do make the coder work better. However, the pictures with higher SNR, at the output of the SNR 2 experiment, do not necessarily look better than the ones with lower SNR. This is because the input data of this SNR 2 experiment is prefiltered with different filters. In other words, the source images quality varies at the input of the SNR 2 experiment.

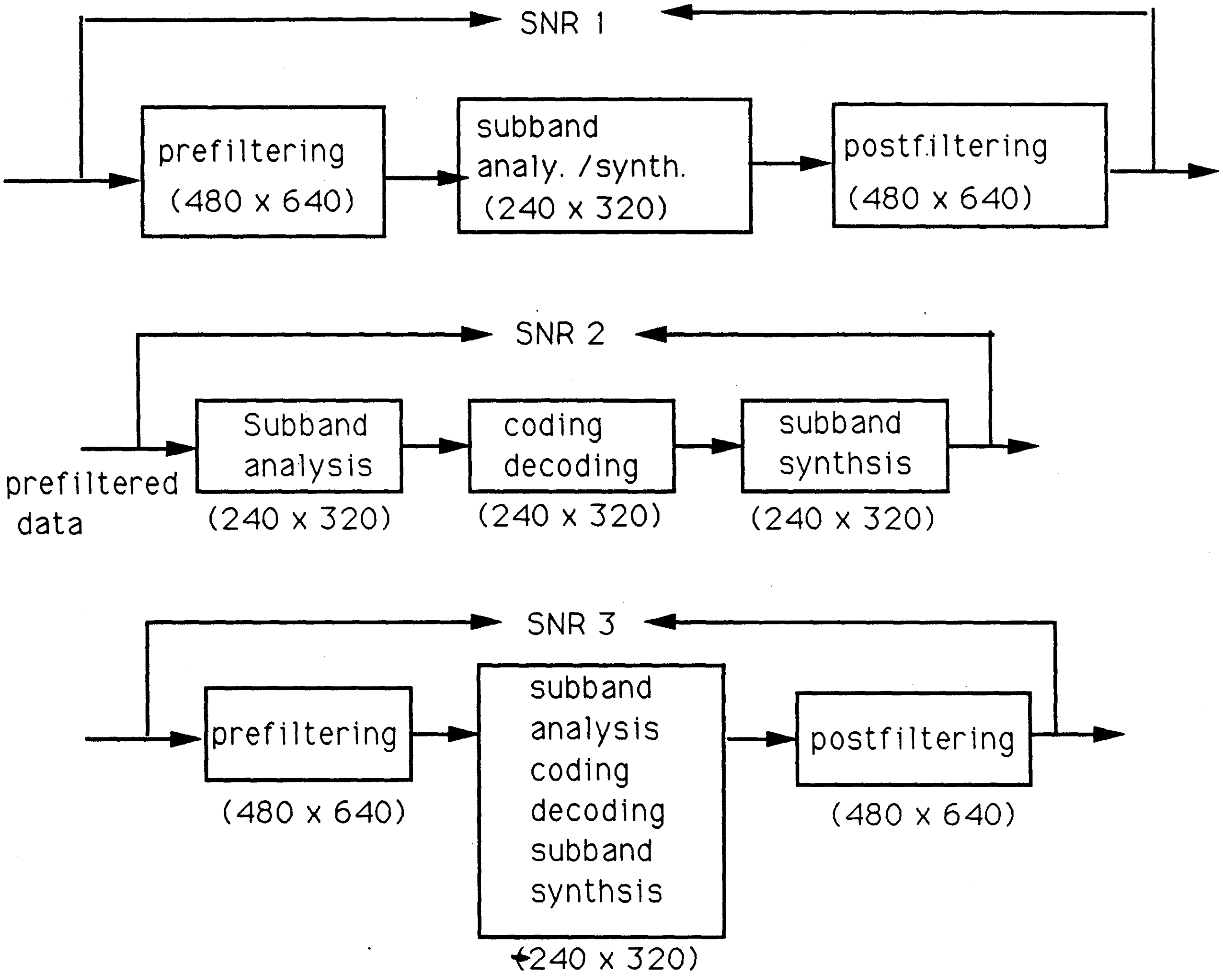

Figure 4.17: Diagram for Testing Prefilters

prefilter used	frame #	SNR 1	SNR 2	SNR 3
MPEGdeci.	1	38.12dB	28.53	29.45
MPEGdeci.	2	37.91dB	28.41	29.38
filter3	1	32.67dB	34.11	29.43
filter3	2	32.63dB	34.26	29.44

Table 4.2: Prefilter Test

The data in column SNR 3 reflects overall picture quality. The SNRs are all close to each other at this time. This is what we expected. The result here tells us that, from the system point of view, using a prefilter with wider passband can provide a sharper reconstructed image with more quantization noise, while using a prefilter with narrower passband can offer a softer reconstructed image with less quantization noise. The overall reconstructed image quality can be objectively identical, if proper prefilter and coding scheme are chosen. This conclusion may make more sense at very low bit rate coding scheme that it could do at high bit rate coding scheme. However, what prefilter should be chosen depends more on subjective preference, which is beyond the scope of this thesis.

The next 6 pictures illustrate the above conclusion. The original picture for these 6 pictures refers to Figure 2.8. Figure 4.18 and Figure 4.19 show the subjective effect of diffrent prefilter. Picture in Figure 4.18 looks sharper than picture in Figure 4.19 because the later one used a prefilter with narrow passband. Figure 4.20 and Figure

4.21 illustrate the results of experiment SNR 2 in Table 4.2 and Figure 4.17. Figure 4.22 and Figure 4.23 are from the result of experiment SNR 3 in Table 4.2 and Figure 4.17. From these two pictures, we can easily see that the two pictures subjectively look different, even though they have almost the same SNR value and they are from the same coding system with same bit rate. The only thing made the two pictures diffrent is the prefilter. So far, we have seen how the prefilter affect the image coding system at a low bit rate.

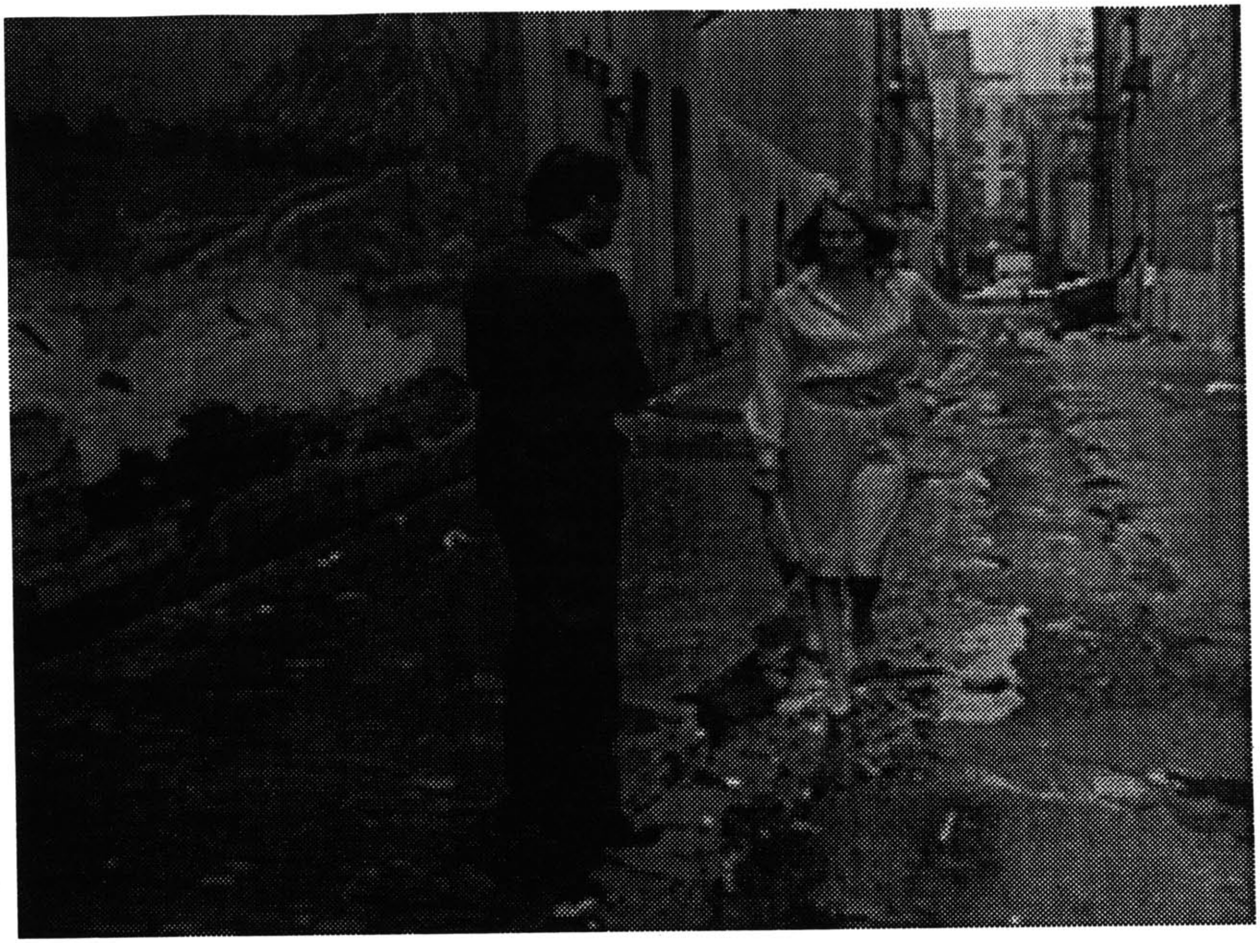

Figure 4.18: The picture is prefiltered by MPEG decimation filter and down sampled by 2. The test picture is sixth frame of fourth second of Alley sequence.

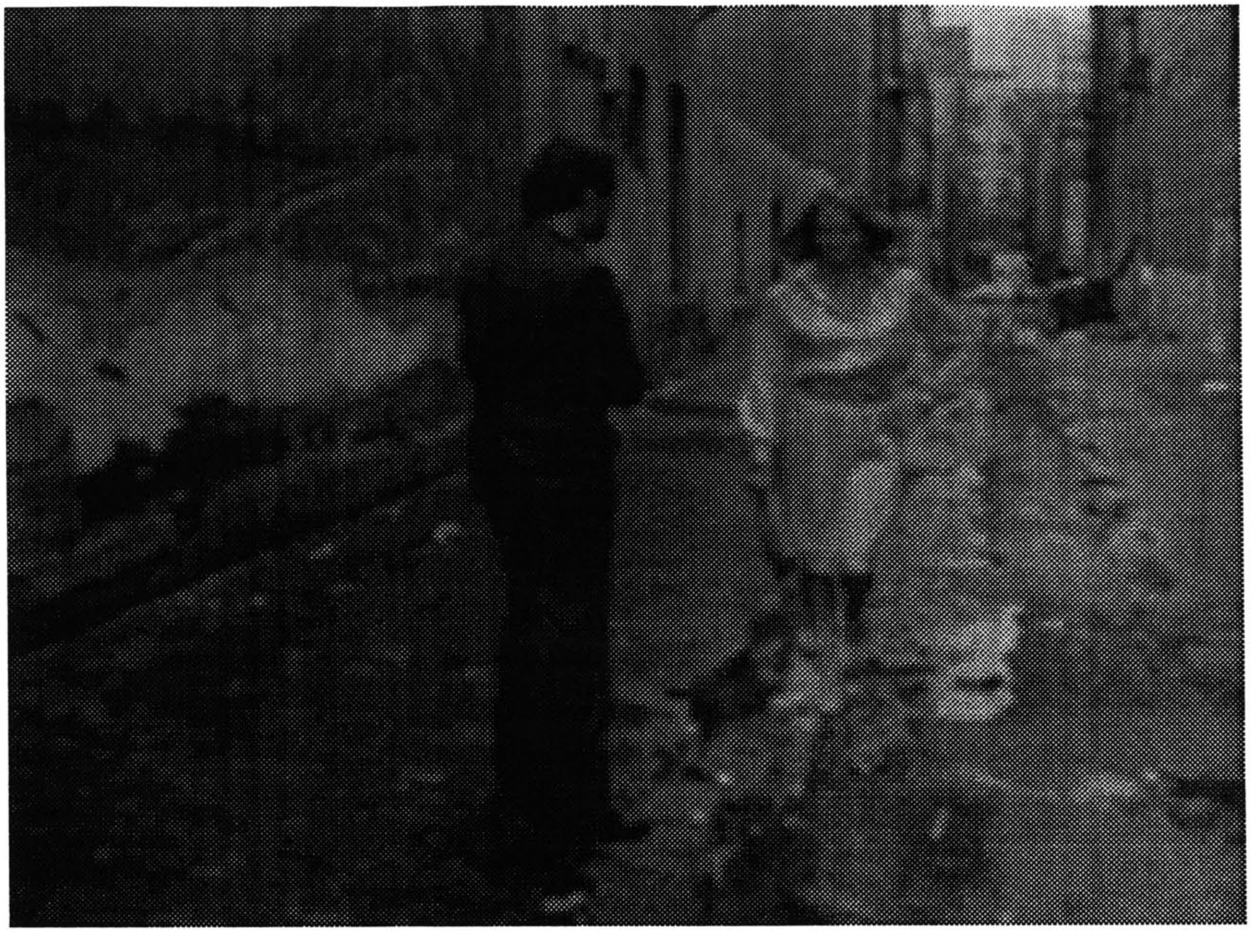

Figure 4.19: The picture is prefiltered by filter3 and down sampled by 2. The test picture is sixth frame of fourth second of Alley sequence.

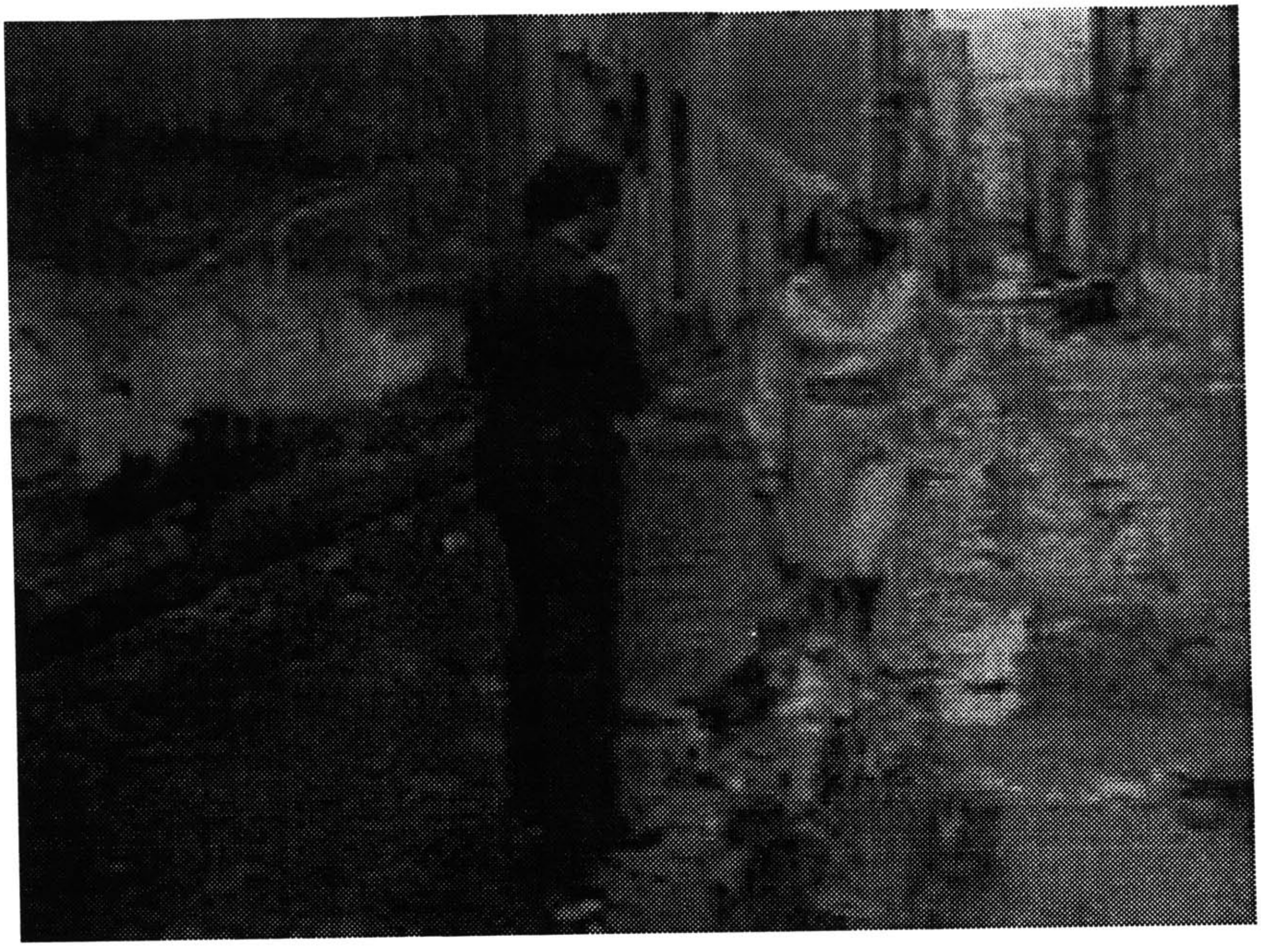

Figure 4.20: The picture is coded at 0.36 bits/per pixel. The source image data has been prefiltered by MPEG decimation filter before being sent to the encoder. The test picture is sixth frame of fourth second of Alley sequence.

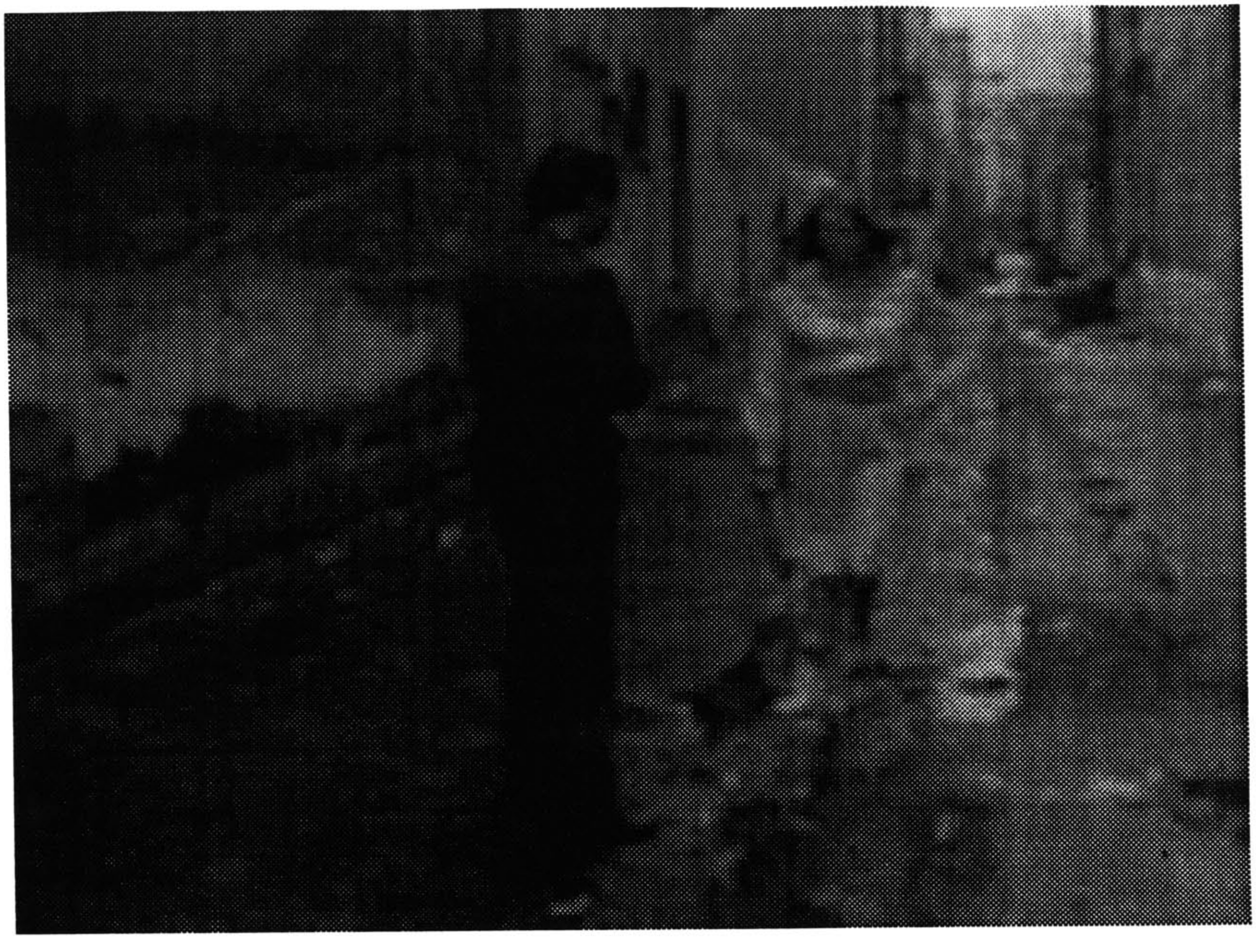

Figure 4.21: The picture is coded at 0.38 bits/per pixel. The source image data has been prefiltered by filter3 before being sent to the encoder. The test picture is sixth frame of fourth second of Alley sequence.

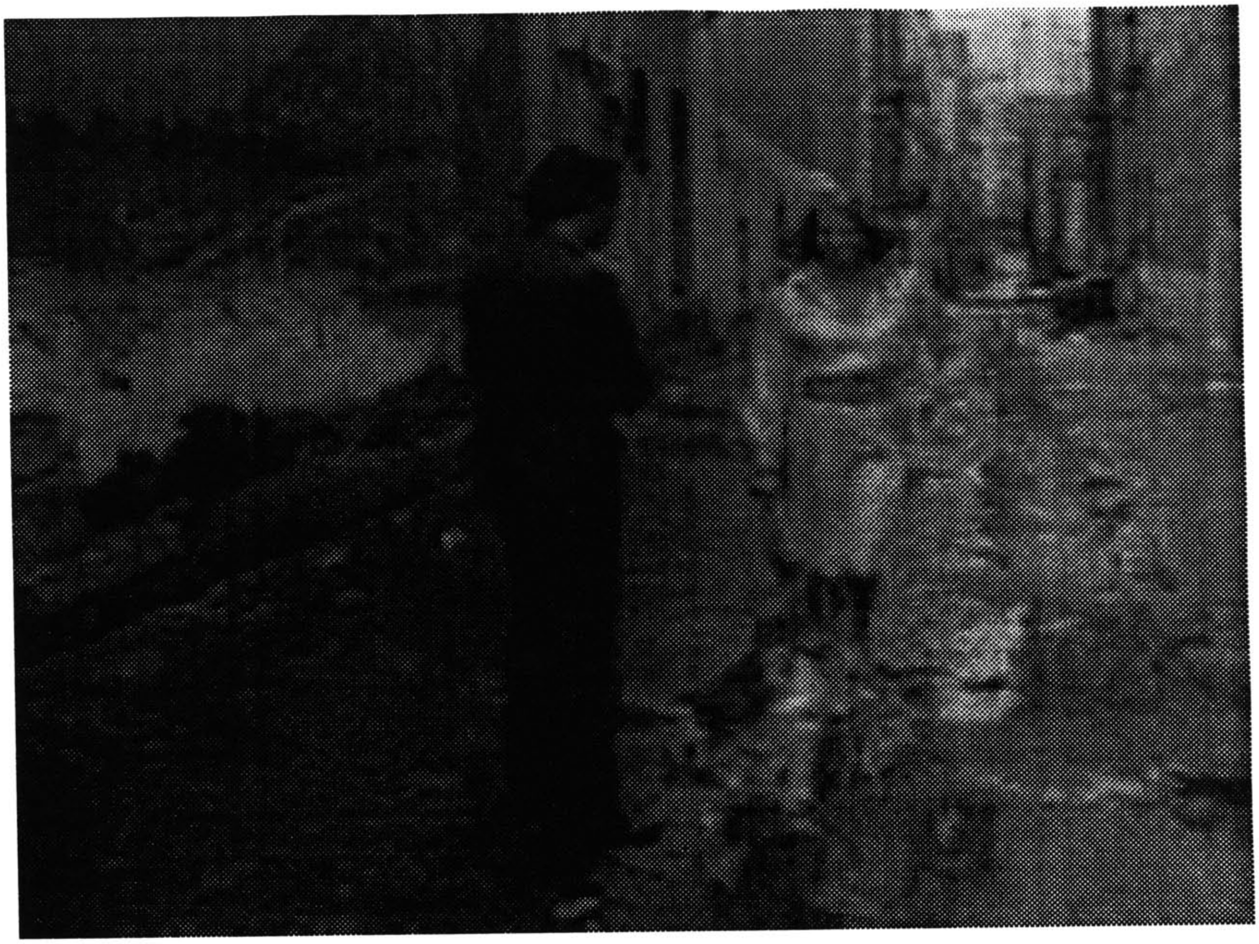

Figure 4.22: The picture is interpolated by 2 after being coded at 0.36 bits/per pixel. The source image data has been prefiltered by MPEG decimation filter before being sent to the encoder. The test picture is sixth frame of fourth second of Alley sequence.

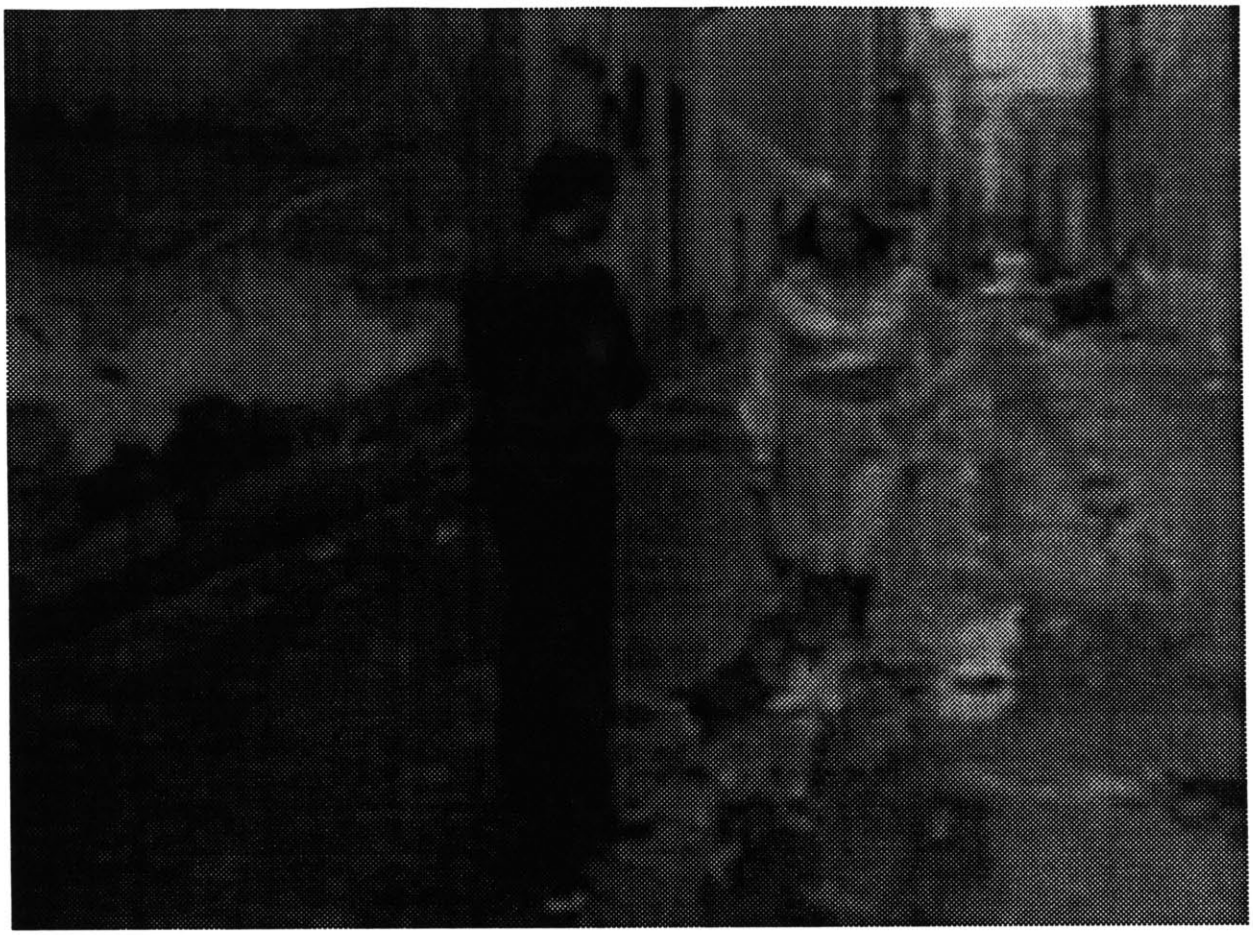

Figure 4.23: The picture is interpolated by 2 after being coded at 0.38 bits/per pixel. The source image data has been prefiltered by filter3 before being sent to the encoder. The test picture is sixth frame of fourth second of Alley sequence.

Chapter 5

Coding Experiments

5.1 A Coding System with Adaptive Bit Allocation — An Overview

The previous chapters have described the variety of elements involved in our image coding system. We can put them together here to build up our subband image coding system with adaptive bit allocation and flexible pre-processing (prefiltering).

The system is drawn in Figure 5.1 The input of the system is 480 x 640 pixels picture sequences. The system takes a pair of pictures at each time and does the prefiltering and downsampling work to take out the part of the energy which can potentially overload the coder in the later part of the system. A one dimensional filter with the required characteristics (described in chapter 4) is used as a prefilter and applied in both vertical and horizontal direction to reduce the bandwidth to 240 x 320. The subband decomposition is performed with respect to both space and time. The average and the difference of the pair of image are regarded as the temporal low and high bands. Spatially, one dimensional QMF filters are applied to generate three level subband pyramid. There are totally 20 subbands after subband analysis and they represent different energy concentrations and physical characteristics. Since our goal is to code image sequences at a low bit rate, the bit allocation algorithm developed in chapter 4 is employed here to allocate the available bits to meet the local statistics of the subbands. The prepared subband data was then sent to the encoder. Kd-tree based VQ is used as the encoder. The encoder is controlled by total bit rate so that the predescribed bit rate can be achieved. The encoded data may be either stored for later application or sent out through distributed networks. The docoder part of our codec system is very simple, just an index mapping. The 20 subbands have been individually active throughout our coding and decoding process. The next

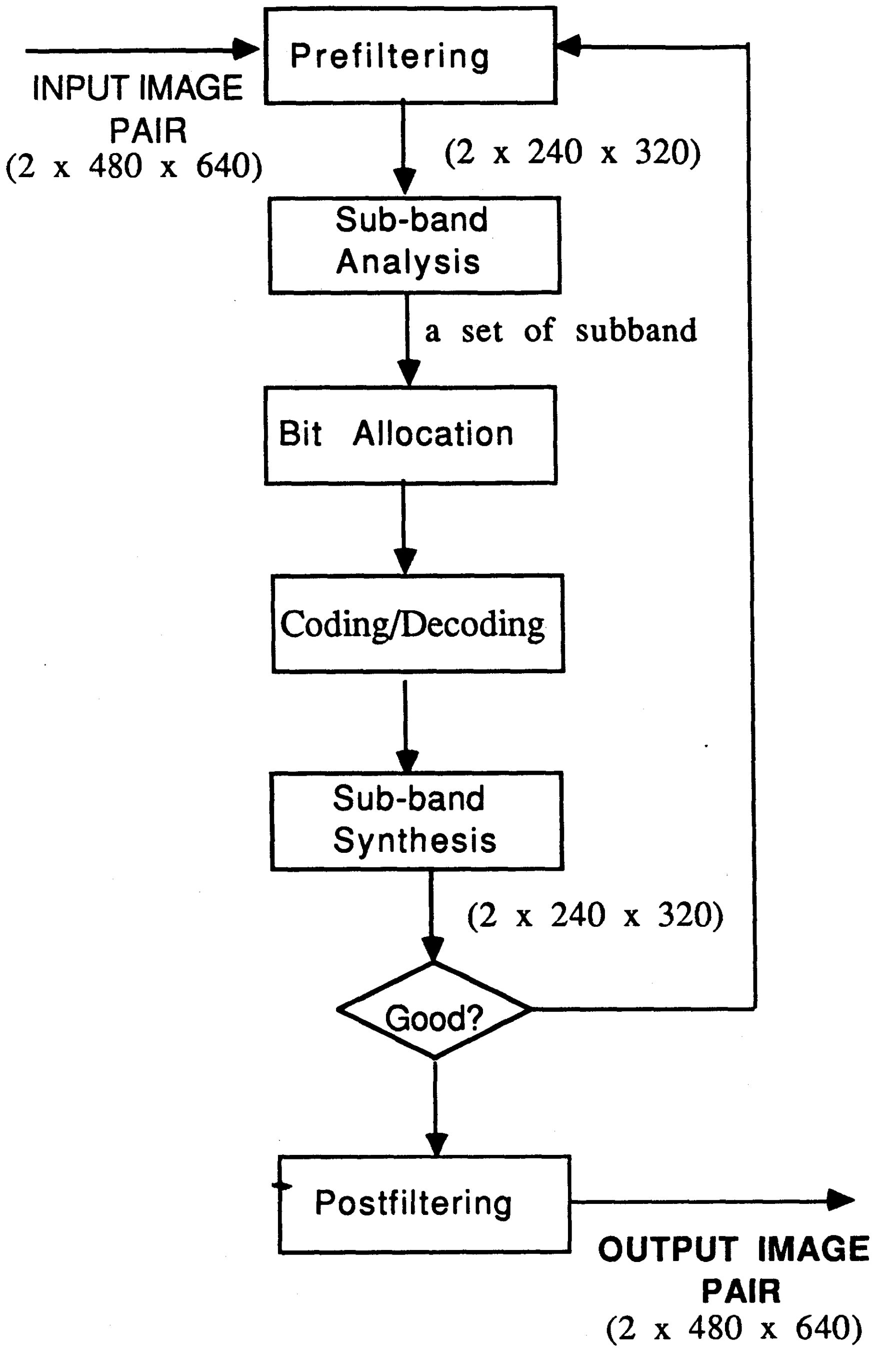

Figure 5.1: Image coding system with adaptive bit allocation

step is to put the subbands back through a subband synthesis system and to evaluate the output picture quality at this point. Our system provides a chance to go back to the prefilter stage if the output picture quality is not satisfactory. Different prefilters may be used to vary the statistics of subbands. The post filter is used to interpolate the image date back to its original size, before it is sent to any application destination.

5.2 Coding Examples

The following tables and figures demonstrate the results of coding a pair of image from the Alley sequence, by using 4 different prefilters. The four prefilters are described in chapter 4, Figure 4.12. The bit rates among the four experiment are between 3.6 bits/per pixel and 3.8 bits per pixel, which is approximately 820 Kbits/per second. SNR 1, SNR 2, and SNR 3 in Table 5.1 refer to the experiment described in Figure 4.17. In Table 5.1, we can see that the overall SNR's (reflected by column SNR 3) of these four experimental result are very close to each other. Carefully examining the data in Table 5.1, we find that the numbers in column SNR 1 decrease as passband of the applied prefilter becomes narrower; more and more information is getting lost. This implies that the picture processed by a prefilter with a narrow passband is

prefilter used	frame #	SNR 1	SNR 2	SNR 3
MPEGdeci.	1	38.12	28.53	29.45
MPEGdeci.	2	37.91	28.41	29.38
filter1	1	34.72	31.83	29.71
filter1	2	34.21	31.77	29.72
filter2	1	33.62	33.24	29.73
filter2	2	33.59	33.41	29.78
filter3	1	32.67	34.11	29.43
filter3	2	32.63	34.26	29.44

Table 5.1: Coding experimental results with different prefilters.

getting lossy. In contrast to this, the numbers in SNR 2 increase as the passband of the applied prefilter becomes narrower. This means the picture processed by a prefilter with a narrow passband is getting less and less quantization noise. To compromise the quantization error with informaiton loss, we get the result in the column of SNR 3.

From Table 5.2 to Table 5.5, we can tell that the bit allocation algorithm we ran allocates more bits towards the lower bands, when the passband of the applied prefilter becomes narrower and narrower.

Figure 5.2 to Figure 5.5 show part of the experimental results in Table 5.1. The pictures for the rest of experimental results were shown in chapter 4 before.

subband #	bit/perpixel assigned	variance
1	5.99	1477.91
2	3.4	40.75
3	3.32	36.79
4	2.58	13.06
5	2.23	32.22
6	2.39	40.4
7	1.42	10.44
8	0.59	13.32
9	0.82	18.31
10	0.65	0.9
11	0.52	0.75
12	0.29	0.54
13	0.25	0.51
14	0	1.33
15	0	1.23
16	0	0.75
17	0	1.48
18	0	1.15
19	0	0.21
20	0	1.08

Table 5.2: Coding experimental results with MPEG decimation filter as prefilter

subband #	bit/perpixel assigned	variance
1	6.64	1474.36
2	3.98	36.71
3	3.91	32.84
4	3.11	10.83
5	2.62	22.22
6	2.75	26.58
7	1.54	4.98
8	0.29	3.51
9	0.52	4.86
10	1.31	0.91
11	1.09	0.67
12	0.83	0.46
13	0.71	0.38
14	0.22	0.79
15	0.21	0.77
16	0	0.32
17	0	0.27
18	0	0.23
19	0	0.01
20	0	0.11

Table 5.3: Coding experimental results with filterl as prefilter.

subband #	bit/perpixel assigned	variance
1	6.87	1474.73
2	4.21	36.09
3	4.11	31.97
4	3.28	10.09
5	2.72	18.78
6	2.83	21.78
7	1.51	3.47
8	0.21	2.28
9	0.43	3.13
10	1.51	0.86
11	1.29	0.64
12	1.01	0.43
13	0.91	0.37
14	0.27	0.62
15	0.25	0.61
16	0	0.21
17	0	0.16
18	0	0.13
19	0	0.01
20	0	0.04

Table 5.4: Coding experimental results with filter2 as prefilter.

subband #	bit/perpixel assigned	variance
1	7.07	1470.52
2	4.33	33.04
3	4.24	29.08
4	3.36	8.64
5	2.78	15.41
6	2.87	17.51
7	1.43	2.36
8	0.16	1.63
9	0.38	2.21
10	1.71	0.86
11	1.39	0.56
12	1.15	0.41
13	0.98	0.31
14	0.32	0.51
15	0.28	0.48
16	0	0.14
17	0	0.11
18	0	0.09
19	0	0
20	0	0.01

Table 5.5: Coding experimental results with filter3 as prefilters.

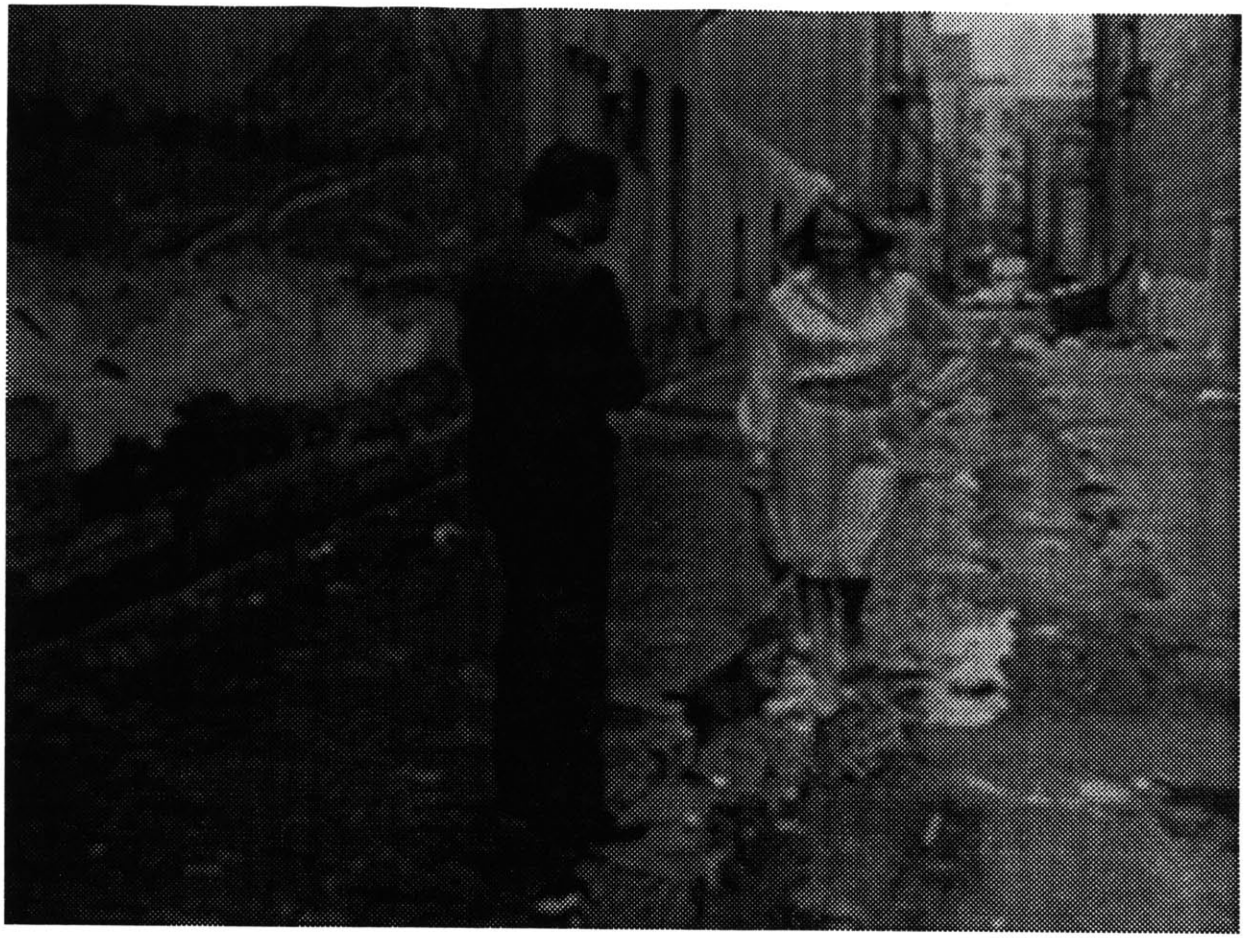

Figure 5.2: The picture is prefiltered by filter1 and downsampled by 2. The test picture is the sixth frame of the fourth second of the Alley sequence.

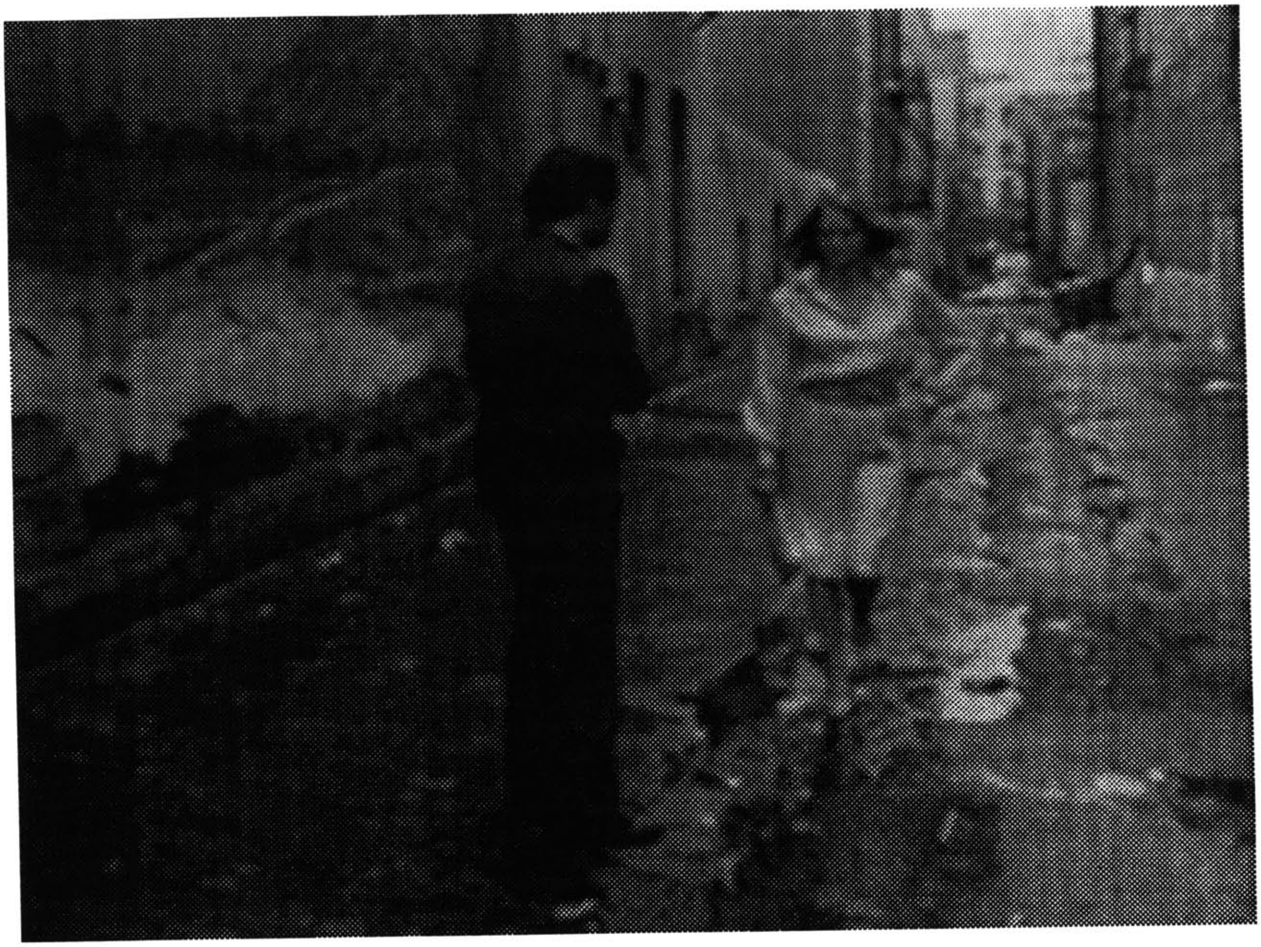

Figure 5.3: The picture is prefiltered by filter2 and down sampled by 2. The test picture is the sixth frame of the fourth second of the Alley sequence.

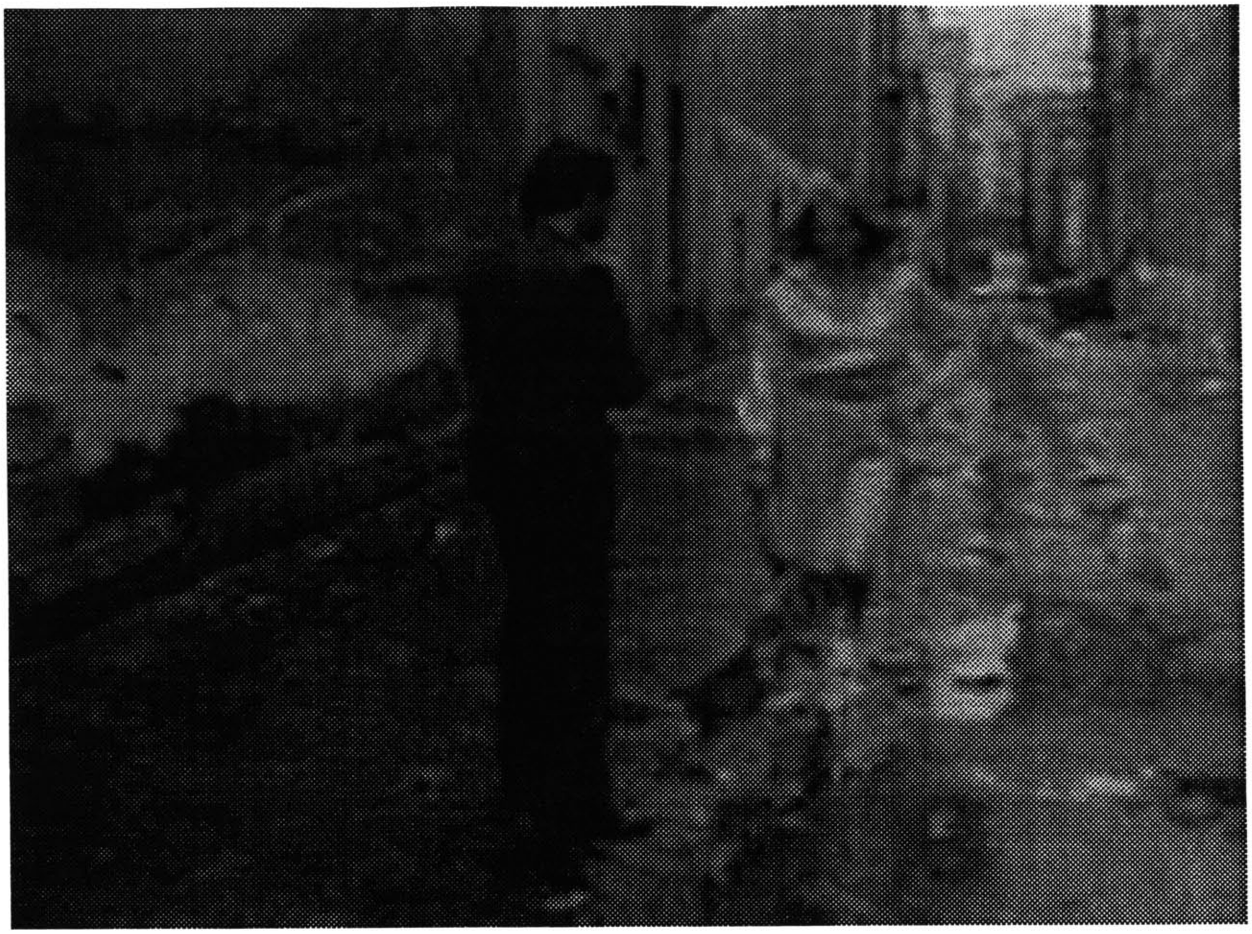

Figure 5.4: The picture is interpolated by 2 after being coded at 0.37 bits/per pixel. The source image data has been prefiltered by filter1 before being sent to encoder. The test picture is the sixth frame of the fourth second the of the Alley sequence.

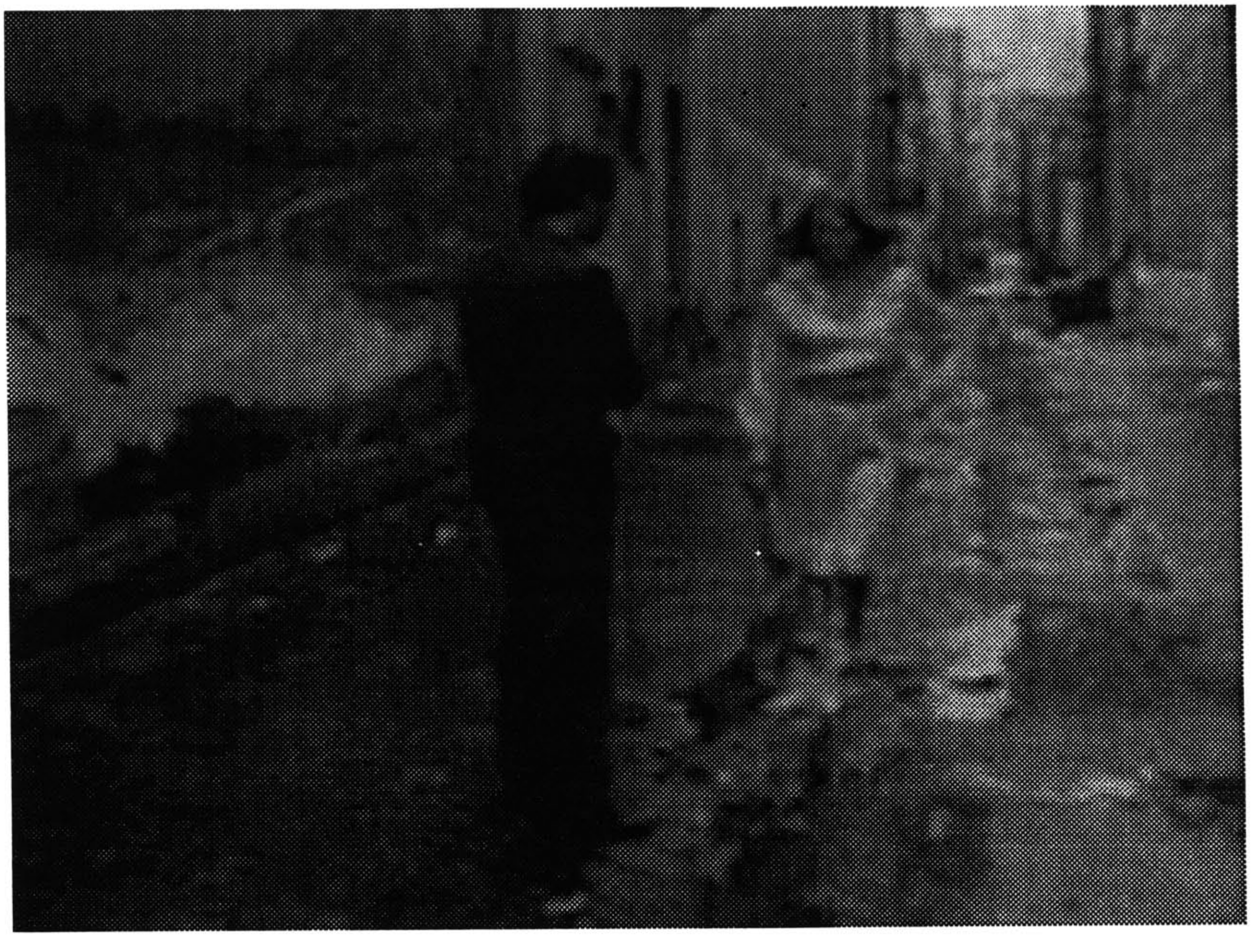

Figure 5.5: The picture is interpolated by 2 after being coded at 0.36 bits/per pixel. The source image data has been prefiltered by filter2 before being sent to encoder. The test picture is the sixth frame of the fourth second of the Alley sequence.

Chapter 6

Conclusions and Further Work

6.1 Suggestions for Further Work

This thesis brings up a couple of interesting topics and provides some experimental results. However, there is much more left for closer examination.

In chapter 2, we developed a total bit rate controlled kd-tree based vector quantizer, and pointed out that different kd-tree parameter's combination can lead to different error concentration under predescribed bit rate. We demonstrated that this is especially important when using variable length entropy coding. However, only few combinations have been tried. There is work leftto try to figure out how to relate the different combination of parameters to different individual subbands by either local statistics or physical features.

The bit allocation algorithm developed in chapter 4 shows some positive results. Even though it is not optimal, it does give good bit allocation according to the objective image data statistics and it is adaptive to the energy distribution of the subbands. But, there are two things we should explore in more detail in the future. The quantizer function we used is a one dimensional scalar quantizer function. The quantizer we really used is vector quantizer. One argument could be that a vector quantizer is just the extension of a scale quantizer in multiple dimensions. Whether or not this is true, we need more theoretical evidence. The other thing is the optimization. In equation 4.4, there is a weighting factor w_k which is supposed to be modified to meet the characteristics of the HVS. However, this factor has not been touched yet so far. This could be a good starting point to run on optimization.

We compared, in chapter 4, the pictures which have the same overall objective measure but different subjective character. The experimental results indicate that the prefilter in image processing can affect the subjective character of the reconstructed image. For example, the prefilter with narrow passband can make the picture look softer, while the prefilter with wide passband can make the picture look sharper but noisy if the pictures processed by either prefilter are coded at same bit rate. The experiments conducted in this thesis focus on the very low bit rate coding scheme. Under this circumstance, the prefilter does affect the overall picture quality very much. However, this may not be that obvious when the bit rate can afford to go up. Some more experiments could be conducted to see if bit rate goes up how the overall picture quality change with different prefilters. Meanwhile, some subjective test may help us to understand whether people prefer sharp but noisy pictures or like soft but smooth pictures.

6.2 Conclusion

An adaptive bit allocation algorithm was developed and tested here which provides the subband coding system with a better code bit distribution, even though it is not optimal. This is especial important for low bit rate image data transmission enabled by subband coding scheme. The original image data is first transformed into a set of spatialtemporal subbands by using QMF filter. Then statistics are run through the subbands. According to the statistical distribution, the bit allocation program allocates the predescribed total bits to different subbands in such a way that the total distortion in the reconstructed picture will be minimized. Since the prefiltering is often involved in the image processing, different prefilters have been tested in our system to check how they can affect overall image quality encoded by our system. The experiments conducted on the image coding system in this thesis demonstrate that the bit allocation algorithm is good and the prefiltering issues are worth to study, particularly their interaction with the other components of the image coding system.

Appendix A

Acknowledgments

I would thank the MIT Media Laboratory administration for keeping it an excellent place with marvelous faculty and one of the best research environments in media technology. I regard the Media Laboratory as the medium between the things I dream of and the real academic world. I enjoyed my two years of student life at MIT Media Laboratory more than I can verbally express.

I hope there are some words which can express my thanks to my advisor and friend, Andrew B. Lippman, for his encouragement, criticism, enthusiasm and wisdom. This thesis work could not have been done without his support. I admire his knowledge in every aspect of image processing.

I wish to thank Walter Bender for his valuable advice on image coding and software simulation, and his comments after reviewing the work.

I would give special thanks to Dr. V. Michael Bove Jr. for carefully reviewing the work and giving valuable corrections on some critical principles embedded in the work.

I wish to thank everybody in the Movies of the Future Group for being nice, helpful people to work with: Pat (who implemented many image coding tools for his thesis which have also been truly useful in this work), Foof (for his better understanding of the mathematical presentation of image coding language), Pizza "NOT related to any restaurant here" (for providing tools in vector quantization simulation), Mr. Alexander Benenson (for his patience when being bothered by tons of questions and bugs in my C programming progress for this thesis), Joel Wachman (for being my on-call computer trouble shooter, hardware and software, and always underpaid), Odin (for the late night help on VPR3 animation process), Jon Orwant (for his volunteering as a editor for my work), Paddy (for his enthusiastic help on the last minutes of the

thesis), Mark Sausville and Ahn (for help with a variety of computer system faults), Pascal and Wad (for keeping the Terminal Garden a great research environment), Gillian(for all the help she gave me in the last two years),...., and everyone else who worked together with me during my two year stay at MIT Media Laboratory.

Finally, I want to thank my family members, my parents, Yifan and Duanshu, (for having given me life and teaching me to value all that life offers), my wife, Jinglian, (for understanding me and loving me in her deepest heart), and my son Tony (for growing up so fast that he will be able to read this thesis before too long).

Bibliography

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, Nov/Dec 1984.

[2] R. Aravind and Allen Gersho. Image compression based on vector quantization with finite memory. *Optical Engineering*, 26(7):570–580, July 1987.

[3] C.J. Bartleson and E.J. Breneman. Brightness perception in complex field. *J.Opt. Soc. Amer.*, 57:953–957, 1957.

[4] Z.L. Budrikus. Visual fidelity criterion and modeling. *Proc. IEEE*, 60:771–779, 1972.

[5] William J. Butera. Multiscale coding of images. Master's thesis, Massachusetts Institute of Technology, 1988.

[6] G. Gambardella C. Bracinni and G. Sandini. A signal theory approach to the space and frequency variant filtering performed by the human visual system. *Signal Processing*, 3:231–240, 1981.

[7] A. Croisier, D. Esteban, and C Galand. Perfect channel splitting by use of interpolation/decimation tree decomposition techniques. In *International Conference on Information Science and Systems*, 1976.

[8] D.J.Sakrison. On the role of the observer and a distortion measure in image transmission. *IEEE Trans. COM*, 25:1251–1266, Nov. 1977.

[9] H. Dudley. Phonetic pattern recongnition vocoder for narrow-band speech transmission. *J. Acoustic Soc. Amer.*, 30:733–739, August 1958.

[10] William Equitz. Fast algorithms for vector quantization picture coding. *ICASSP*, 1987.

[11] J. J. Friedman, J. L. Bentley, and R. L. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematics Software*, pages 209–226, September 1977.

[12] H. Gharavi and A. Tabatabai. Application of quadrature mirror filtering to the coding of monochrome and color images. In *Proccedings ICASSP*, page 32.8.1 to 32.8.4, 1987.

[13] W. E. Glenn, Karen Glenn, R. L. Dhein, and I. C. Abrahams. Compatible transmission of high definition television using bandwidth reduction. In *Proceedings of the 37th Annual Broadcast Engineering Conference*, pages 341–349. National Association of Broadcasters, 1983.

[14] W. E. Glenn, Karen G. Glenn, Marcinka J., R. L. Dhein, and I. C. Abrahams. Reduced bandwidth requirements for compatible transmission of high definition television. In *Proceedings of the 38th Annual Broadcast Engineering Conference*, pages 297–305. National Association of Broadcasters, 1984.

[15] Morris Goldberg and Huifang Sun. Image sequence coding using vector quantization. *IEEE transactions on Communications*, COMM-34(7):703–710, July 1986.

[16] Robert M. Gray. Vector quantization. *IEEE ASSP Magazine*, April 1984.

[17] Paul Heckbert. Color image quantization for frame buffer display. *Computer Graphics*, 16(3):297–307, July 1982.

[18] Ellen C. Hildreth and John M. Hollerbach. A computational approach to vision and motor control. A.I. Memo 864, Massachusetts Institute of Technology, August 1985.

[19] N.S. Jayant and Peter Noll. *Digital Coding of Waveforms.* Prentice-Hall Signal Processing Series. Prentice-Hall, 1984.

[20] J.O.Limb. Visual perception applied to the encoding of pictures. *SPIE Advances in Image Transmission Techniques*, 87:80–87, 1976.

[21] J.O.Limb. Distortion criteria of the human viewer. *IEEE Trans. SMC*, 9:778–793, December 1979.

[22] Pasquale Romano Jr. Vector quantization for spatiotemporal subband coding. Master's thesis, Massachusetts Institute of Technology, 1990.

[23] Gunnar Karlsson and Martin Vetterli. Sub-band coding of video singals for packet-switched networks. In *Visual Communications and Image Processing II*, volume 845, pages 446–456, Bellingham, WA, October 1987. SPIE.

[24] D. H. Kelly. Motion and vision ii. stabilized spatio-temporal threshold surfaces. *Journal of the Optical Society of America*, 69(10):1340–1349, October 1979.

[25] D. H. Kelly. Spatiotemporal variation of chromatic and achromatic contrast thresholds. *Journal of the Optical Society of America*, 73(6), June 1983.

[26] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions On Communications*, 28:84–95, January 1980.

[27] Andrew Lippman and William Butera. Coding image sequences for interactive retrieval. *Communications of the ACM*, 32:852–860, July 1989.

[28] J.L. Mannos and D.H. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Trans.*, 20:525–536, July 1974.

[29] David Marr. *Vision.* W. H. Freeman and Company, 1982.

[30] John H. R. Maunsell and David C. Van Essen. Functional propoerties of neurons in middle temporal visual area of the macaque monkey. i. selectivity for stimulus direction, speed and orientation. *Journal of Neurophysiology*, 49(5):1127–1147, May 1983.

[31] R. W. McColl and G. R. Martin. Quad-tree modeling of colour image regions. In *Visual Communications and Image Processing '88*, volume 1001, pages 231–238. Proceedings of the SPIE, 1988.

[32] Arun N. Netravali and Birendra Prasada. Adaptive quantization of picture signals using spatial masking. *Proceedings of the IEEE*, 65(4):536–548, April 1977.

[33] Ashok C. Popat. Speech sub-band coder based on entropy-coded quantization. Master's thesis, Massachusetts Institute of Technology, 1989.

[34] S.A. Webber R.E. Crochiere and J.L. Flanagan. Digital coding of speech in subbands. *Bell Systems Technical Journal*, 55(8):1069–1085, October 1976.

[35] Jorma Rissanen and Glen G. Langdon. Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162, March 1979.

[36] Jorma Rissanen and Glen G. Langdon. Universal modeling and coding. *IEEE Transactions on Information Theory*, IT-27:12–23, January 1981.

[37] Peter H. Schiller, Barbara L. Finlay, and Susan F. Volman. Quantitative studies of single-cell properties in monkey striate cortex. i. spatiotemporal orgainzation of receptive fields. *Journal of Neurophysiology*, 39(6):1288–1319, November 1976.

[38] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, July 1948.

[39] Eero Peter Simoncelli. Orthogonal sub-band image transforms. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1988.

[40] Joseph B. Stampleman. Multiscale codebooks, 1989.

[41] T.G. Stockham. Image processing in the context of a visual model. *IEEE Proc.*, 60:828–847, 1972.

[42] D. E. Troxel, Schreiber W. F., R. Grass, G. Hoover, and R. Sharpe. Bandwidth compression of high quality images. In *International Conference on Communications*, pages 31.9.1 - 31.9.5, June 1980.

[43] Broder Wendland. Extended definition television with high picture quality. *SMPTE Journal*, pages 1028–1035, October 1983.

[44] Peter H. Westerink, Dick E. Boekee, Jan Biemond, and John W. Woods. Subband coding of images using vector quantization. *IEEE transactions on Communications*, 36(6):713–719, June 1988.

[45] Hugh R. Wilson and James R. Bergen. A four mechanism model for threshold spatial vision. *Vision Research*, 19:19–32, 1979.

[46] John W. Woods and Sean D. O'Neil. Subband coding of images. *IEEE Transactions on Acoutic, Speech, and Signal Processing*, ASSP-34(5):1278–1288, October 1986.