NOTICE

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

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

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

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

AVIS

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

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

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

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

VECTOR QUANTIZATION APPLIED TO A FACET REPRESENTATION OF AN IMAGE

by
Steve Reed

A thesis
presented to the Ottawa-Carleton
Institute for Electrical Engineering
in fulfillment of the thesis
requirement for M.A.Sc in
Electrical Engineering

Ottawa, Ontario
August 30, 1987

CONTENTS

Abstract

A facet model for image data is introduced which uses a set of piecewise polynomial functions to represent an image. The image is partitioned in the spatial domain into equally sized connected blocks called facets. The correlation between pixels in images allow large block sizes to be used with only a small amount of error introduced. Different block sizes assure that the changing statistics within regions of an image are coded efficiently. Experiments are performed using 16 pixel blocks and 4 pixels blocks represented with a polynomial of degree one, resulting in rates between 1.5 and 6.0 bits per pixel. It is shown that most of the 16 pixel blocks can be represented well, while some have an unacceptable amount of error. Blocks with the largest error are split further resulting in an image with low error and excellent subjective quality. Vector quantization is applied to the vectors formed by the facet transformation resulting in rates between 0.3 and 1.5 bits per pixel. Experiments are performed on three test images to establish the following characteristics: 1. the correct percentage of 2 by 2 blocks required for coding within a given level of fidelity; 2. the optimum codebook size for efficient coding; 3. the need for a codebook for each resulting set of vector being quantized or for just a single codebook for both vector sets. It is shown that the size of codebook and percentage of smaller blocks required depends largely on the image statistics but that an optimum in terms of distortion can be established. It is also seen that only a single codebook is necessary. In all the test images, the r(D) were close whether a partitioned or a single codebook was used.

Acknowledgment

I wish to express my gratefulness and appreciation to my thesis advisor, Dr Morris Goldberg. I cannot thank him adequately enough for the guidance, encouragement and support which he has generously given me during the course of this research.

I would like to express my deep grief at the death of Dr. Andrew Smith, a thesis advisor and professor at the University of Ottawa. He was an invaluable assistance to the me in the initial stages of my work. It would be fair to say that everyone who knew him would like him. He was a man we all greatly respected and admired.

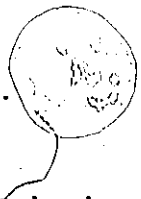
In addition I would like to thank my colleagues , Huifang Sun and Limin Wang, for their encouragement and help. Finally, I would like to acknowledge that this work was made possible by grants from NSERC.

LIST OF FIGURES

LIST OF SYMBOLS

$p(u)$	Probability density for a random sequence of images
$d(u,v)$	Distortion measure between image u and decoded approximation image v
$p(v/u)$	Probability transition distribution, probability of u given the received image v
$p(u)$	Probability density of image u
$P_{p(v/u)}$	Average distortion for transition probability p(v/u)
p_D	The set of all p(v/u) which achieves less than some distortion D
$I_p(v/u)$	Average mutual information for transition probability
$R(D)$	Rate distortion function
u_i	Any still life image fitting the class of images defined in section 4.1
$\hat{L}$	$N_x Y_y L$ dimensioned vector formed by blocking together source output $u_1, u_2, ...M$
$u_M^{\hat{L}}$	A codebook when m=1,2,3...M possible $\hat{L}$ dimension vectors
$d(u_m, u_m^L$	Distortion between input and u_m and u_m^L
u_{min}^L	Codeword than minimizes distortion
$\hat{D}$	Average distortion
$U(x,y)$	Test pattern used in detectability experiments
$w(x,y)$	The log of intensity values of u(x,y)
$z(x,y)$	A contrast variable defined by in terms of w(x,y)
$r(x,y)$	Result of filtering z(x,y) with bandpass filter
r_k	Real valued response variable of HVS
f	Spectral density of image u
g	Spectral density of reconstructed image v
x_i	Observed pixel value
$C(i)$	Predictor that minimizes $E[x_i - xi - 1]$
ρ	DPCM linear predictor
m	Mean grey level of block
$E[d[x,y]]$	Expected distortion
Z_x, Z_y	Row and column indexes of spatial domain of image u(x,y)
$B(x,y)$	Block of pixels centered around (x,y)
π_i	Partition of the spatial domain of Z_x, Z_y
α	Linear constraint in x dimension
β	Linear constraint in y direction of Z_x, Z_y
γ	Constant equivalent to image average
$\nu(x,y)$	Random noise having zero mean and covariance matrix proportional to some constant

1.0 Introduction



The information 'revolution' has created a demand for increased bandwidth to enable more users to benefit from communications systems. Because of the greater flexibility with data in discrete form, most of this information is in a digital format leading to even greater bandwidth requirements. The subject of this thesis is data compression for images in digital form.

A digital image is obtained by quantizing a continuous image both spatially and in amplitude. Digitization of the spatial coordinates is called image sampling while amplitude digitization is referred to as grey-level quantization. In this thesis an image will be denoted by u(x,y), where the amplitude of u at the spatial coordinates (x,y) is the intensity or grey level value. The basic statistical property upon which image compression techniques rely is inter-pixel correlation. A second property that can often be exploited is that distortions can be introduced in the image where the human visual sensitivity is low with little subjective visual loss.

Chapter 2 reviews some fundamentals of rate distortion theory and the human visual system. The general rate distortion theory answers a fundamental question in communication theory; Given a prespecified source signal, what is the minimum channel capacity necessary to transmit the information from that source to the receiver within a prescribed degree of fidelity? The human visual response has been included because the permissible nature and amount of visual degradation produced when an image is coded at low bit rate is closely connected to the visual properties of the human observer. Chapter 3 reviews various image coding methods. Included in this is a description of various forms of vector quantization. Chapter 4 is entitled A Facet Model for Image Data. A facet model for image data is introduced which

uses a set of piecewise polynomial functions to represent an image. Experiments are performed using 16 pixel blocks and 4 pixel blocks which are represented with a polynomial of degree one. It is shown that most of the 16 pixel blocks can be represented well, while some have an unacceptable amount of error. Blocks with the largest error are split further resulting in an image with low error and excellent subjective quality.

Chapter 5 is entitled Vector Quantization Applied to Facet Representation. Using a facet representation for vector formation, followed by VQ, leads to coding rates between .3 and 1.5 bits/pixel. The method is compared to other methods. For a given bit rate there is a reduction in the blocking effect, superior edge reproduction, and better results in terms of the rate distortion.

Finally conclusions and further work are discussed in chapter 6.

2.0 Image Distortion and the Visual System

The objective of picture compression is to represent an image with as few bits as possible. For a given amount of average distortion, the minimum average number of bits per image (or rate) is given by Shannon's rate distortion theory. The character of the optimum encoding method is determined by the statistical nature of the source and the distortion measure chosen. Section 2.1 considers the optimum possible performance of a coding system as given by rate distortion theory. An examination of the achromatic characteristics of the human-visual-system (HVS) in section 2.2 reveals the nature of error permissible when an image is coded with low bit rate. Distortion measures do not generally quantitatively specify the degree of visual degradation in an image. Never-the-less an analytical distortion measure must be chosen and several are considered in section 2.3.

2.1 Rate Distortion Theory

Shannon's [1] coding theory provides the mathematical basis for finding the lowest achievable number of bits to represent an image, within the bounds of a fidelity requirement.

Assume the source to be coded generates a random sequence of images that are statistically independent characterized by a probability density p(u). A distortion measure, d(u,v), assigns a non-negative number to the error introduced when representing the image u(x,y) by a decoded approximation v(x,y). The transition distribution denoted by p(v/u) is the probability of u given the decoded image v. The density p(u) together with the transition distribution p(v/u), define a joint density distribution p(u,v), associated with an average distortion

$$D_{p(v/u)} = \sum_x \sum_y p(u,v)d(u,v) \tag{2.1}$$

$$= \sum_x \sum_y p(u)p(v/u)d(u,v)$$

where $D_{p(v/u)}$ is the average distortion for the transition probability p(v/u). Let us now define the set p_D as the set of distributions which achieves less than some allowable distortion D;

$$p_D = \{p(v/u): D_{p(v/u)} \leq D\} \tag{2.2}$$

The average mutual information for the transition probability is then;

$$I_{p(v/u)} = \sum_x \sum_y p(u)p(v/u)log\frac{p(v/u)}{p(v)} \tag{2.3}$$

and the corresponding rate distortion function is defined as

$$R(D) = min[I_{p(v/u)}p(v/u)\epsilon p_D] \tag{2.4}$$

It is important to observe that the rate distortion function is determined by the probability density of the source and the distortion measure d(u,v) imposed by the user.

Shannon's converse to the coding theorem states that given a source with rate distortion function R(D) and any rate R bits/pixel, there does not exist any coding-decoding method of transmitting the source output u at rate R with less than average distortion D.

$$R(D) = R \tag{2.5}$$

A coding method is suggested to attain this performance limit. A number of source outputs $u_1, u_2, u_3, \ldots u_L$ are blocked together to form an $\hat{L} = N_x N_y L$ dimensional vector $u^{\hat{L}}$. $N_x N_y$ are the row and column indices of a spatially contigous block. This $\hat{L}$ dimensional vector is quantized into one of M possible $\hat{L}$ dimensional vectors chosen from a codebook, $u_m^{\hat{L}}, m = 1, 2, \ldots, M$. It is now only necessary to transmit the index m. The decoder using the index reconstructs $u_m^{\hat{L}}$ as an approximation to the original u_m. The distortion between input, u_m, and the output, $u_m^{\hat{L}}$, is measured as follows,

$$d(u_m, u_m^{\hat{L}}) = (1/\hat{L}) \sum_{L=1}^{\hat{L}} d(u, v) \tag{2.6}$$

The encoder simply needs to select a codeword, $u_m^{\hat{L}}$, which minimizes the distortion $d(u_L, u_m^{\hat{L}})$.

$$u_{min}^{\hat{L}} = min[u_m^{\hat{L}} \mid m = 1, 2, \ldots M. \tag{2.7}$$

2.2.1 Source-Coding Theorem

Given a source with rate distortion function R(.), an average distortion level $\hat{D}$, and small positive numbers E and G, there exists a block length L and a set of M

code words $u_1, u_2, \ldots, u_M$, with M the largest integer satisfying

$$M \leq 2^{L[R(\hat{D})+G]} \tag{2.8}$$

and the average encoding-decoding quantization distortion satisfying

$$D(u^L, v^L) \leq \hat{D} + E \tag{2.9}$$

Therefore M code vectors can be represented by L(R($\hat{D}$) + G) bits, and the rate per pixel, R+G, approaches arbitrarily close to the rate given in equation 2.5. Thus the rate distortion theorem shows that R(D) specifies the minimum possible rate required to achieve average distortion $\hat{D}$. The problem with extending the block lengths is that the complexity increases to unmanageable proportions.

Shannon's rate distortion theorem is the theoretical basis for improvement of coders using vector quantizers over the coders using scalar quantizers. The theorem is the motivation for pursuing vector quantization as a coding method.

2.2 The Human Visual System (HVS)

The final measure of performance in the image compression process is the human observer. Unfortunately, where the human observer is concerned, the determination of an appropriate measure of error has proven to be a stumbling block in attempts to apply theoretical analyses to the relation between minimum transmission rate and an allowable level of perceived image degradation. Distortion measures appropriate for image coding based on the achromatic properties of the HVS have been suggested [2][3], generally these measures are far too complex to be of practical use. However, by examining the HVS, insights into what aspects must be present in a more practically implementable distortion measure are gained.

2.2.1 Spatial and Amplitude Response of the eye

Experiments have been reported [4][5] on the threshold of detectability of a human observer to sinusoidal gratings. The stimuli were presented in the following form;

$$U(x,y) = U_o + kcos[2\pi f_o(xcos\theta - ysin\theta)] \tag{2.10}$$

These experiments showed that at photopic levels of illumination, and for fixed values of f_o and θ, the detectability depends only upon the contrast of the grating, k/U_o, and not independently upon the values of k or U_o. Sensitivity depends on f_o and θ and is inversely proportional to the contrast needed to detect it. The contrast sensitivity increases linearly from low spatial frequencies up to a maximum in the mid-frequency spatial frequency and then falls off rapidly with increasing frequency.

Maximum viewer sensitivity to spatial frequency in number of cycles in 1.0

degree of arc subtended by a viewer's field of vision was found to be between 3.0 and 4.5 cycles/degree [3]. The sensitivity also depends on θ reaching a maximum when the lines are horizontal and vertical and decreasing about 3 dB at an angle of 45 .

It has been observed that the level at which one frequency component is detected is unaffected by the presence of a second component indicating that the frequencies are separately detected by the observer. This observation suggests a model where the visual perceptive system is composed of a number of parallel detection mechanisms, termed spatial channels. Each channel is tuned to a different spatial frequency, separated by about an octave, and orientation angle.

Tests on sinusoidal gratings, narrow band random gratings, compound sinusoidal gratings, and compound random gratings [4][5] have indicated that for single component patterns, perceived contrast was a piecewise linear function of real contrast for contrast levels above a threshold. Near the threshold, the function of perceived contrast and actual contrast has a nonlinear form approximated by a sixth-power law out to some value and then a linear characteristic. This implies that for a narrow band stimuli, a single response variable r_k is itself an appropriate measure of perceived contrast.

Modelling the HVS

The observed properties of the achromatic visual system are modeled as follows;

A number of parallel processors produce real-valued response variables $r_k, k =$

1, 2...; whenever one of these outputs exceeds a threshold, the observer detects a stimulus. This model is shown in figure 2.1. Detectability depends upon a contrast variable z(x,y) defined as

$$z(x,y) = \hat{w}(x,y) - w(x,y) \tag{2.11}$$

where $w(x,y) = log[u(x,y)]$ and u(x,y) is the intensity values.

There is also a linear low-pass operation that precedes the logarithmic operation. However, usually this low-pass filtering drops off more slowly than the subsequent neural processing and can be neglected. The functions of the response variables can be generated from z(x,y) as follows. Let r(x,y) denote the result of filtering z(x,y) with a bandpass filter of center frequency f_k, radial bandwidth $\pm$ 0.5 f_k and angular bandwidth $\pm$ 10. Also let g(.) denote a pointwise nonlinearity, so the response variable is consistent with

$$r_k = \int\int g[r_k(x,y)]dxdy \tag{2.12}$$

In addition, an observer attempting to detect a deviation on a uniform background, as discussed thus far, differs from trying to detect an error in an image u(x,y). The ability to detect an error on an image $u(x,y) = U_o + rms$ is inversely proportional to the value $U_o + rms$. The reduction in sensitivity is greatest when the error is the same spatial frequency as the image. This observation demonstrates the well known effect of viewer sensitivity reduction when the image has sharp edges or large gradients. This can be expressed in the response variable r_k as follows;

$$r_k = \int\int \frac{a_k(x,y)r_k(x,y)}{(1+s(x,y))^n}dxdy \tag{2.13}$$

where s(x,y) is a spatially smoothed rms measure of local rms gradient.

The observer will detect an error whenever any one of the r_k exceeds a threshold.

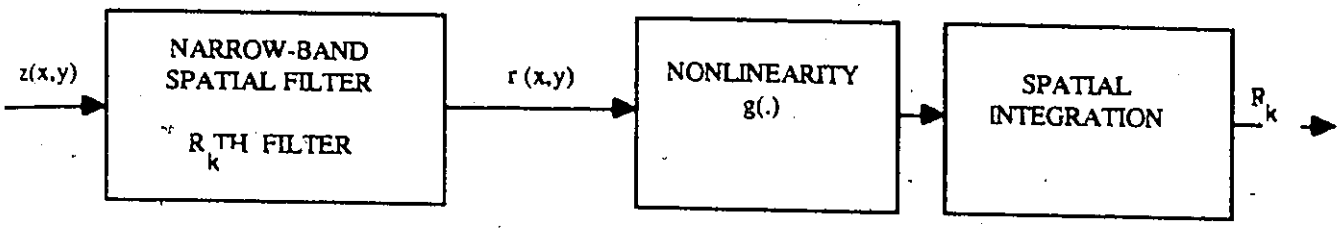


Figure 2.1 Block diagram of a simple model for a single-channel detector.

2.3 Distortion Measures

In this section, distortion measures are defined and briefly discussed [12][57]. Distortion measures pertaining to vector quantizers of images are emphasized.

A distortion measure is an assignment of a non-negative number to the pair of the original vector, u and reproduction vector, v of the compression system. The distortion measure d(u,v) must be non-negative, and if u=v, d(u,v) must be equal to 0.

For a distortion measure to be beneficial, it should possess certain attributes. First, it must correlate well with a viewer's subjective opinion, so that less and more distortion corresponds to better and worse images. Secondly, it should lend itself to mathematical analysis or be mathematically tractable. Finally, the actual distortions resulting in a real system should be efficiently computable [58].

The most common measures of distortion are difference distortions that use a L_p norm on the difference u-v. The most common choices for p are 1 or 2, yielding absolute and mean square. These are metrics or distances in that they satisfy a symmetry requirement d(u,v) = d(v,u) and a triangle inequality;

$$d(u,v) \leq d(u,w) + d(w,v) \tag{2.14}$$

Distortion measures may also depend on second order properties such as the sample auto-correlations or spectral models. Distortion measures that depend on the differences in log (usually applied to the spectral domains) are ratio distortion measures.

$$d(u,v) = d(1,v/u) = d(u/v,1) \tag{2.15}$$

If u is the original input and v is a reproduction set restricted to some finite or infinite set, but chosen to minimize the distortion measure d(u,v) over the set, two distortion measures will be *nearest neighbor equivalent* if they are minimized by the same v regardless of the reproduction set. From this equivalence, it is seen that one should always use the computationally simplest form of nearest-neighbor equivalent distortions in order to find the nearest neighbor.

2.3.1 Difference Distortion Measures

The most common distortion measure is the squared error (or error power or error energy). The measure is easily computed and mathematically tractable, however, it is not as subjectively meaningful. A poorer quality image does not necessarily follow from larger distortion using a squared error measure.

With the original and reproduction vectors in k-dimensional space, the context independent squared error distortion would be just the square of Euclidean distances between the two vectors.

$$d(u,v) = \| u - v \|^2 = \sum_{i=0}^{k-1} (u_i - v_i)^2 \tag{2.16}$$

A natural extension of the squared error is the context dependent rth power measure, defined as follows;

$$d(u,v) = \sum_{i=0}^{k-1} W_i (u_i - v_i)^r \tag{2.17}$$

where W_i is a function of the luminance values of the immediate neighbors, and r is an even integer.

2.3.2 Ratio Distortion Measures

Another class of distortion measures are the ratio distortion measures.

As a measure of performance for a system using squared error, one standard is the signal to quantization- noise ratio.

$$SNR = log_{10}\frac{[E \parallel u \parallel^2]}{E[d(u,v)]} \tag{2.18}$$

Since the squared error has little to do with subjectively meaningful images, the SNR is equally as suspect as a measure of subjectively good images.

The distortion measure used for all the experiments in the NMSE.

$$NMSE = \frac{\sum_{i=0}^{k-1}[u_i - v_i]^2}{\sum_{i=0}^{k-1} u_i^2} \tag{2.19}$$

The major advantage to the normalized mean squared error is its intuitive appeal. Error in low pixel grey level values are given more weight than then errors in high pixel values where the eye is less sensitive. Several of the ratio distortion measures appear to be more subjectively meaningful than the squared error measure. However, even in the simplest form they are complicated, non-symmetric in input and output arguments, and are computationally more costly. They are widely used in speech applications although there are some sound theoretical reasons for using the measure in a nearest neighbor application [6].

These distortion measures have been applied as speech distortion measures to speech waveforms through the second- order properties, the sample auto correlations or spectral models. A spectral distortion measure assigns a non-negative number d(f,g) to the input-output pair f and g, where f is the input spectral density and g the reproduction spectral density.

Log spectral distortion measure [7] is formed by the difference L_p norm of the log spectra.

$$d_{lnp}(f,g) = \| ln(f) - ln(g) \|_p = \| ln(f/g) \|_p \tag{2.20}$$

The d_{lnp} can be called a pseudo-metric because it possesses symmetry and the triangle inequality properties.

Another ratio distortion measure proposed[8] is the **Itakura- Saito distortion** measure.

$$d_{IS}(f,g) = \| (f,g) - ln(f/g) - 1 \|_1 \tag{2.21}$$

For speech distortion measure, the Itakura-Saito distortion is a subjectively meaningful measure [9].

3.0 Image Coders

The basic goal of image coding is to reduce the number of bits required to represent an image and to reconstruct a faithful duplicate of the original picture. Image coding can be divided into two broad classes of techniques: information-lossless and information-lossy techniques. Lossless techniques are able to reconstruct the original image exactly, whereas, lossy techniques introduce distortions. Still or intra-frame image coding methods can also be divided into spatial domain encoding and transform domain encoding. Spatial domain methods operate on the pixel values in some appropriate way while transform domain methods operate on a set of transform coefficients derived from the pixel values. A further classification is often made between fixed or adaptive methods; the distinction depending on whether the parameters used are fixed or adapted in some way to the local statistics of the image.

3.1 Transform Methods

In transform coding[10] the image is first divided into spatially contiguous blocks of equal size, and then a transform is applied to each block to yield a set of coefficients. Transform coders attempt to transform one set of data into another set. A source block of pixels is transformed, for example, by some linear orthogonal transform, resulting in a set of transform coefficients. The coefficients are then ranked in order of importance and quantized. Less important coefficients are discarded and compression results. The inverse transformation recovers the original image minus any information discarded. The important parameters to consider when using a transform coding

system are; block size, coefficients to be selected, and bit assignment and quantization. Transform coding may also be adaptive by matching the parameters of the coder to the local statistics of the image[10]. Adaptation can be made with regards to the transform level, bit assignment or quantizer level assignment. Some examples of linear transformations that are unitary transformations are; Fourier, Hadamard, Haar, sine, cosine, slant and Karhunen-Loeve. The Karhunen-Loeve transform is noteworthy because it results in uncorrelated coefficients. The transform gives an upper bound with which other transformations can be compared.

3.1.2 Spatial Methods

Predictive coding techniques. Methods using prediction [11] produce data that is more uncorrelated that the original image data and tend to be easier to implement than transform methods.

Correlated image data allow a prediction of a given element in terms of other elements in the picture. A difference signal is formed by subtracting the prediction from the actual value and this is quantized into discrete levels and assigned a binary word. The signal is then stored or transmitted. The reconstruction is accomplished by using the same predictor and adding to it the difference signal. Error is introduced by the quantization of the difference signal. The method tends to be sensitive to variations in input image statistics and channel errors. Predictive coders can be decomposed into three processes; predictor, quantizer, and code assigner.

Predictor

Based on an observed set of pixels $\{x_{1-1} \mid i \epsilon I\}$, a prediction is made of the next pixel x_i, where I specifies the pixels to be used for prediction. A linear prediction of x_i based on an observed pixel x_{i-1} takes the following form

$$x_i = \sum_i C(i) x_{i-1} \tag{3.1}$$

The best mean square predictor is observed by choosing C(i) to minimize $E\{(x_i - x_i)^2\}$. Some predictors of interest include the following.

Differential pulse code modulation (DPCM). DPCM [12] results in uncorrelated difference signals and uses a linear estimator that results in the least mean square error, given by;

$$x_i = \rho x_{i-1} + (1 - \rho) m \tag{3.2}$$

where m is the mean grey level and ρ is the normalized correlation between the adjacent pixel elements. ρ is given by

$$\rho = \frac{E\{x_i x_{i-1}\}}{E\{x_i^2\}} \tag{3.3}$$

A block diagram of DPCM is given in figure 3.1.

Interpolative. Interpolative predictors [13] use zero-order and first order polynomials to predict the next pixel. Higher order polynomials can be used but the computational complexity does not justify the improvement.

Non-linear predictors. In one class of non-linear predictors [14] the direction of contours in the image is determined and the predictor is chosen accordingly.

Quantization

With the quantizer in the encoder-decoder loop, the estimates are the quantized differences and the decoding process is no longer exact. One source of error is the so-called "slope overload." This occurs when the quantizer is set with closely spaced bins for slowly varying scan lines. On rapidly changing scan lines the decoder would not be able to follow the input because the largest difference is outside the quantizer range and error would result. This problem could be rectified by using wider quantization bins but wider bins would result in poor slowly varying regions creating an artifact known as granular noise.

Bit Assignment

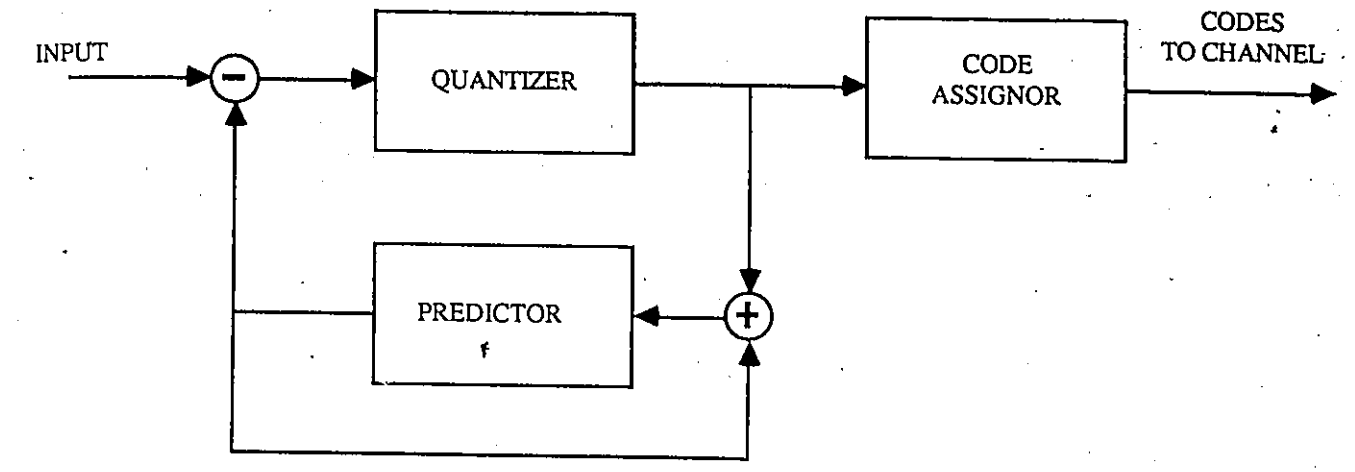
Each level in the quantizer is assigned a different codeword. Because of the non-uniform rate at which the quantizer outputs levels occur a variable length coding can be used. A Huffman code would give the minimum length code.

Hybrid Methods

Hybrid coding [15] methods refer to methods combining predictive coding and transform coding. One possibility is to use a one dimensional transform in one axial direction, (for example horizontally) followed by a predictive operation carried out between like members in the horizontal set. Generally the characteristics of such an approach lie between those of the constituent methods.

Delta Modulation

One special case of DPCM is when a 1-bit quantizer is used. This is called delta modulation[13].



Transmitter

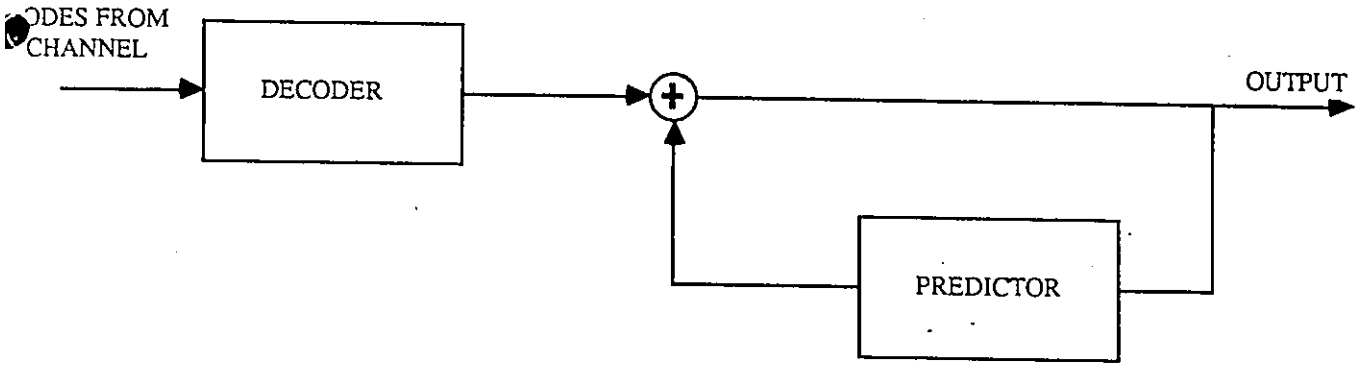


Figure 3.1 Block diagram of DPCM.

3.1 Vector Quantization

A vector quantizer, or more exactly a memoryless vector quantizer, consists of an encoder/decoder pair[58]. The encoder maps each input vector

$$X = (X_0, X_1, X_2 \cdots X_{k-1}) \qquad (3.4)$$

into a channel symbol (or index) A(X) chosen from some channel symbol set M. The decoder performs a second mapping, assigning a symbol from a reproduction alphabet A for each index m producing a reproduction sequence. Vector quantization with a minimum distortion encoder is Shannon's model for a block source code subject to a fidelity criterion. Vector quantization can be characterized by four processes.(figure 3.2)

1. The decomposition of the image into a set of vectors.

2. The choice of a training set from these vectors.

3. Generation of a codebook from the training set.

4. Quantization of the input vector into the closest code word in the codebook and assignment of a label.

3.1.1 Vector Formation

Vector formation is the process used to decompose the image into a set of vectors. The approaches to vector formation can be broadly classed into two categories: direct spatial/temporal formation and feature extraction formation. A di-

rect spatial/temporal approach forms the vectors from the intensity values of spatially/temporally contiguous blocks of pixels in an image. Vector formation by image feature extraction uses distinguishing characteristics to derive vectors. Various methods of feature extraction have been proposed and include: original grey levels, block average, transformed coefficients of the block [19,20] and the adaptive linear predictive coding coefficients for a block [20]. Of specific interest are the following;

Transform coefficients [19][20] The transform is first applied to blocks of pixels, some higher order coefficients containing little information are discarded. The selected coefficients are then used to form a vector. Two main benefits are realized by forming vectors in this way. First, because the dimension of the formed vector is less then the original spatial block, a saving in computation is realized. The second is that decorrelating the vectors leads indirectly to reduced blocking effect [21]. In chapter 4 a non-unitary transform will be introduced that produces three coefficients from any block size. In chapter 5 these vectors are vector quantized realizing the expected advantages and some additional benefits.

Multistep VQ. In this technique[26] the error vector resulting from the first vector quantization is fed to a second vector quantization. The process is repeated by feeding the second step error to a third, and so on. The memory size for storing the codebook and the computational complexity is reduced with this method.

3.1.2 Training Sequence Generation

The task of training set generation is to accurately represent the input vector sequence. It is necessary to use a training set because the statistics are generally

unknown. If the input sequences have large variances a large training sequence is required. A random selection of input vector can be used as a training sequence for vector quantization. However, as the codebook is optimized to the given training sequence and therefore it must be representative of the statistics of the input source. One approach to training set generation is to use a selection of input vectors by dividing the source into sets of similar statistics. Other methods that have been suggested are adaptive methods [23] and partitioned codebook methods [16]. Some of these methods are now discussed below.

Local training sequence generation. Adaptive methods [23] improve the overall performance by changing the codebook to the local statistics of the image. The image is divided into smaller sets of sub-images. For each of these sub-images a separate codebook is generated. Each sub-image is taken as a training set and is used to generate its corresponding codebook. The main difficulty in this approach is the overhead needed to update (adapt) the codebooks.

Separating edge and non-edge vectors. Using spatially contiguous blocks of pixels introduces a problem in the reproduction of edges. One proposed improvement [16] is to distinguish the edge vectors from the non-edge vectors and to use separate quantizers for each class of vectors. This requires two codebooks at the decoder. This generally results in an improved subjective image but may result in an increase in NMSE.

Gain/shape VQ. Gain/shape VQ [22] uses two codebooks to quantize the 'shape' and 'gain' of the image blocks. The 'shape' is defined as the original input vector normalized by the removal of a 'gain' term. The 'gain' can be the block mean or block energy.

Finite state VQ. This method [27] incorporates memory about previously encoded blocks into the encoding of each successive input block. The technique uses a finite number of states which summarizes key information about previous encoded vectors to select one of a number of codebooks with which to encode the input vector. In other words, for each state there is a corresponding codebook.

3.1.3 Codebook generation

Central to obtaining good results in vector quantization is the development of a codebook. Optimal codewords are the mean value of the partitioned region in Euclidian space (Voronoi regions) under the MSE measurement criterion. The optimal codebook, using the mean squared error (MSE) criterion, must satisfy two necessary conditions[24];

1. The input vector source, V, is partitioned into N closed sets or Voronoi regions, $\{R_i: 1 \leq i \leq N\}$ determined by the minimum distance rule:

$$R_i = \{V: |v - w_i| \leq |v - w_j|\} \ for\ all\ j \neq i \tag{3.5}$$

2. The corresponding codewords are then defined by

$$w_i = E\{v|v \in R_i\}, i = 1, 2, \cdots, N \tag{3.6}$$

The codebook generation can be performed by an iterative algorithm developed by Linde, Buzo and Grey [25]. If certain conditions are met, the algorithm produces a locally optimum quantizer. The steps in the algorithm are as follows:

1. Starting with the source vectors or a training sequence of the source vectors X, and a convex distortion measure d(x,y), an initial guess of the codebook is made $y_1^0, y_2^0, \ldots, y_n^0$.

2. At the kth iteration the nearest neighbor code vector for each training vector is found. That is, the training sequence is partitioned into sets by using the

nearest distance rule (smallest distortion measure) i.e.

$$S_i^k = \{x: d(x, y_i) \leq d(x, y_j)\}, i = 1, 2, \cdots N_c \tag{3.7}$$

3. The kth reproduction vectors, y_i^k, are found as follows;

$$y_i^k = E\{y_i: x \epsilon S_i^k\}, \quad i = 1, 2, \ldots N_c \tag{3.8}$$

In other words, this minimizes the expected distortion.

4. Convergence for the process is tested for by evaluating the fractional change in the total distortion. If the change is above some threshold, steps 2 and 3 are repeated.

A number of variations for codebook generation have been proposed.

Multistep VQ. In this technique, the error vector resulting from the first vector quantization is fed to a second vector quantization. The process can be repeated by feeding the second step error vector into a third step vector quantization, and so on. If N_i is the codebook size at the ith step for a m-step vector quantization then the number of combined representative vectors is $N = N_1 + N_2 + \ldots + N_m$. The computation time and memory requirements are reduced with this method.

Initial codebook. The codebook is generally generated by a clustering algorithm which requires an initial guess at the codewords. Two approaches have been suggested to obtain the initial codebook. In the first, a codebook is constructed from only two codewords, the codebook is generated by recursively splitting the codewords until the required number of codewords is found. This technique is known as binary splitting [59]. The second approach is to start with the required number of codewords, e.g. by uniformly distributing the values[16][23].

3.1.4 Quantization

The process of quantization involves finding the closest codeword for each input vector. From Shannon's model for a block code, the VQ is optimum if it minimizes an average distortion E(d(x,y)). Two necessary conditions for optimum PCM with a squared error distortion measure to vector quantization were developed by Lloyd [30]. These two necessary conditions are:

> Property 1; To minimize the average distortion for a given decoder B and thus for a given codebook, the best the encoder can do is to assign the codeword v in M which will yield the minimum possible distortion at the output. More succinctly, the best possible encoder for a given codebook is a minimum distortion or nearest neighbor mapping.
>
> Property 2; A decoder can do no better than to assign to each channel symbol v the generalized centroid of all source vectors encoded into v. The centroid is the vector yielding the minimum average conditional distortion given that the input vector was mapped into v. For a squared error distortion measure, this would be the Euclidean centroid, or the vector average of all input vectors encoded in the given channel symbol.

A quantizer can search over the entire codebook in which case the search is exhaustive or it can search by a hierarchical partition known as a tree search quantization.

Tree searched quantization [31]. At each layer of the tree a decision is made between one of two sets of codewords. At the last layer a representative codeword

is found for the input vector. Tree-search is suboptimal for the given codebook and requires more memory.

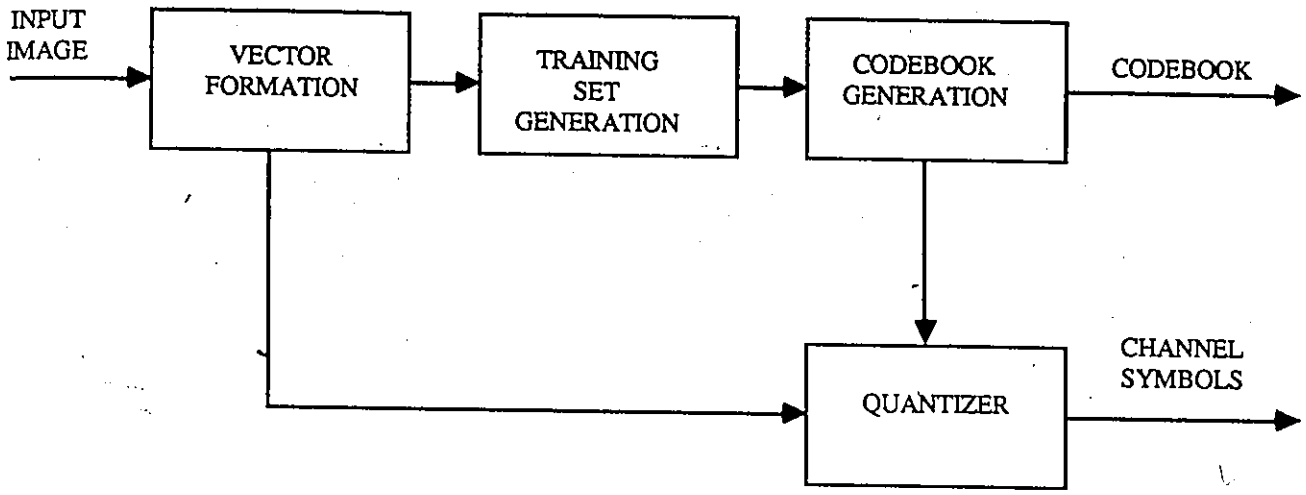


Figure 3.2 Block diagram of vector quantization depicting the four steps.

3.2 A Final note on Adaptive VQ

The basic VQ creates a single general codebook from a training sequence taken from "a class of images" and uses this same codebook for all images in the class. The expected distortion quantifies the performance of the system. In practice, it is the long term average distortion that is measured. If the process is stationary and ergodic, then the limiting time average distortion is the same as the mathematical expectation. If the input source is indeed stationary and ergodic, the resulting sample average distortion should be nearly the expected value and the same code used on future data should yield approximately the same average. However, the image data may be neither stationary nor ergodic. In fact, there is no generally accepted accurate probability distributions for real image data. In order to use the technique of designing a code for a sufficiently large training sequences and then using the code on future data, a sufficient condition [32] would be that the source be asymptotically mean stationary. So, a more mathematical term for 'a class of images' is 'asymptotically mean stationary sources'.

Thus, even quantization made for a general 'class' of images is adaptive, but adaptive to a larger set of input sources. When image data is introduced that does not fall into this set, poor coding results. An example of this occurs when an image of greater or less exposure appears in the input.

On the other hand, Adaptive VQ (AVQ) is a method of incorporating memory into the vector quantizer. The decoder is informed of the chosen codebook by concatenating the information onto the channel words or in a special side channel. Theoretically, VQ's with memory should be no better than memoryless VQs in terms of minimizing average distortion. However, a VQ with memory may yield the desired

distortion with less complexity. Adaptive VQ would not suffer from the effect of the data not being ergodic or stationary because it adapts to the image statistics. There is some overhead involved in adaptivity. Adaptive VQ requires some of the bit rate to be used to transmit the codebook. The rate calculation is now;

$$\text{rate} = \text{labels} + \text{codebook} \qquad (3.9)$$

The additional bit rate incurred increases with the block size and size of the codebook. This is clearly seen in figure 3.3, where with a vector size of 4 the overhead amounts to only 2 percent of the rate, while with a vector size of 20, a full 72 percent of the rate is taken by the transmission of the codebook. Thus the overhead does put a upper bound on the codebook size. An optimum size was found [33] to be around 9. When the codebook size is greater than 20 the transmission of the codebook renders adaptive VQ inefficient in terms of bit rate that it becomes useless. There is, therefore, a trade off between block size, codebook size and overhead. An additional concern is the need to generate a new codebook for each image.

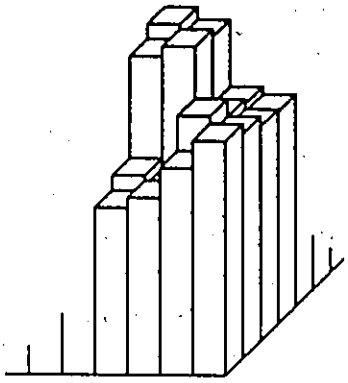
The methods introduced in this thesis are all performed on adaptive vector quantization although the methods are equally applicable to basic VQ. The results presented all factor in the additional bit rate to transmit the codebook.

	Block size	Codebook size	No. of bits	Bit rate	% bit rate in overhead
Basic VQ	4	64	6	1.50	-
Adaptive VQ	4	64	6	1.53	2.0%
Basic VQ	6	181	7.5	1.50	-
Adaptive VQ	6	137	7.1	1.52	7.7%
Basic VQ	9	16384	14	1.56	-
Adaptive VQ	9	512	9	1.51	34.7%
Basic VQ	12	262144	18	1.50	-
Adaptive VQ	12	512	9	1.50	50.0%
Basic VQ	16	1.6 E7	24	1.50	-
Adaptive VQ	16	478	8.9	1.49	62.6%
Basic VQ	20	1.1 E9	30	1.50	-
Adaptive VQ	20	446	8.8	1.52	71.2%
Basic VQ	24	6.8 E10	36	1.50	-
Adaptive VQ	24	388	8.6	1.50	76.0%
Basic VQ	30	3.5 E13	45	1.50	-
Adaptive VQ	30	338	8.4	1.52	81.5%

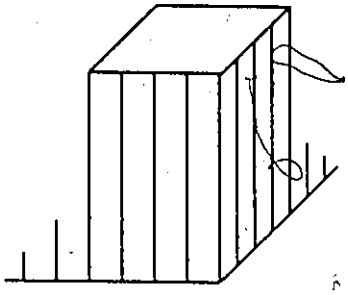
Figure 3.3 The table shows the size of the codebook and thus the label size for an example bit rate of 1.5 bits per pixel. Note how the codebook size goes to a maximum and then decreases with adaptive VQ. The size of the codebook for basic VQ is limited only by computational compexity. The calculations assume a 256 by 256 image with 8 bit pixels.

4.0 A Facet Model for Image Data

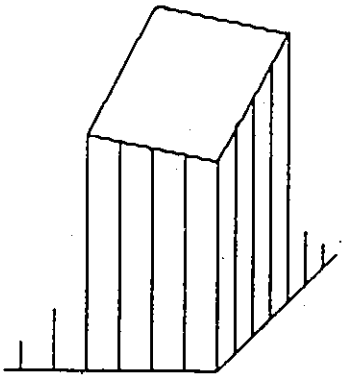
Real images often contain a large amount of order and regularity. Many images can be well represented by a model that regards this order and regularity as a union of small regularly shaped surfaces. If large regions of an image are approximated by these simple surfaces then the image would require fewer bits to represent it than the original. An image model using these surfaces, or facets, to describe sections of the image and polynomials to describe these surfaces will now be explored. In section 4.1 a facet model for images is discussed that represents small blocks of the image as two dimensional polynomial functions. In section 4.2 a review is given of how this facet model has been applied to edge detection, noise cleaning and image analysis. Section 4.3 introduces the use of facet representation as an image compression technique.



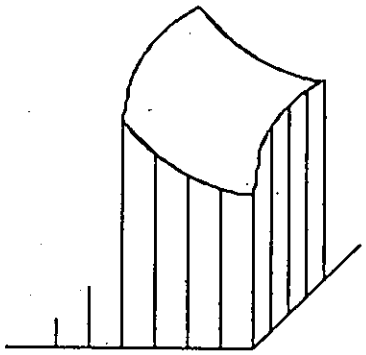
(a) Original grey level values viewed as small surfaces above the image.



(b) Flat facet model approximates surfaces with a polynomial degree zero.



(c) The planar facet model approximates grey levels with polynomial degree one.



(d) A polynomial of degree two approximates grey levels with a curve with no inflection

Figure 4.1 The facet model for image representation viewed as surfaces above the image. Higher order polynomials can represent the image with less error. Any block size can be represented errorlessly if a polynomial of sufficient degree is used.

4.1 A Facet Model for Image Data

> **Facet:** A small face or surface; one of the small planes which forms the sides of a crystal; a flat surface with a definite boundary.[49]

In this section a facet model for image data is presented in which an image is represented by a set of piecewise polynomial functions [50]. The model assumes that the image can be partitioned in the spatial domain into connected regions called facets each of which satisfies certain grey level and shape constraints. The grey levels in each facet are a polynomial function of row-column coordinates of the pixels in the facet.

The grey levels can be viewed as composing a facet above the region, the basic shape of the facet depending on the degree of polynomial used. Facet shapes for polynomials of degree zero, one, and two are shown in figure 4.1. A polynomial of degree zero [51][52] would mean that the surface of the facet would be a flat region. A polynomial of degree one, (planar facet model) would require the surface to be a plane with any orientation. A polynomial of degree two would restrict the facet to a curved surface without inflection.

The facet model proposed by Haralick[53] described an image by irregular shaped regions formed by merging smaller regions with similar characteristics. As the facet model was employed for the purpose of edge detection this was appropriate. In section 4.2 a shape constraint will be imposed to allow the facet representation to be applied for image compression purposes.

A planar facet model will now be explored in more detail. Let Z_x and Z_y be the row column index set of the spatial domain of an image. For any $(x,y) \in Z_x \times Z_y$, let u(x,y) be the grey level value of resolution cell (x,y) and let B(x,y) be the B × B block of resolution cells centered around resolution cell (x,y). Let $\pi = \{\pi_1 ..... \pi_N\}$ be a partition of the spatial domain of $Z_x \times Z_y$ into its facets.

In the planar facet model for every resolution cell $(x,y) \in \pi_n$, there exists a resolution cell $(i,j) \in Z_x \times Z_y$ such that:

1. Shape constraint: $(x,y) \in \mathbf{B}(i,j) \subseteq \pi_n$

2. Region grey tone constraint: $u(x,y) = \alpha_n x + \beta_n y + \gamma_n$

The modelled image is assumed to differ from the real image by the addition of random stationary noise having zero mean and covariance matrix proportional to some constant; i.e.

$$v(x,y) = u(x,y) + \eta(x,y) \tag{4.1}$$

where

$$E[\eta(x,y)] = 0 \tag{4.2}$$

$$E[\eta(x,y)\eta(x',y')] = k\sigma(x-x', y-y') \tag{4.3}$$

The planar facet model fits a sloped plane to each block. This plane results in a least square error approximation of the region. The derivation of the equations for this plane is as follows: Assume the block lengths are odd so that one of the block's pixels is at the center. Let the block be (2L+1) × (2L+1) with the upper left hand corner pixel having value (-L,-L) and let u(x,y) be the grey value at row x, and column

y. For any block entirely contained in a facet;

$$u(x,y) = \alpha x + \beta y + \gamma + \eta(x,y) \tag{4.4}$$

where η(x,y) is noise. A least squares procedure[50] may be used to estimate α, β , and γ , let

$$f(\alpha,\beta,\gamma) = \sum_{x=-L}^{L} \sum_{y=-L}^{L} (\alpha x + \beta y + \gamma - u(x,y))^2 \tag{4.5}$$

Then the least squares estimates for α, β, γ are those which minimize f . Taking partial derivatives and solving the resulting equations gives;

$$\alpha = \frac{3}{L(L+1)(2L+1)^2} \sum_{x=-L}^{L} x \sum_{y=-L}^{L} u(x,y) \tag{4.6}$$

$$\beta = \frac{3}{L(L+1)(2L+1)^2} \sum_{y=-L}^{L} y \sum_{x=-L}^{L} u(x,y) \tag{4.7}$$

$$\gamma = \frac{1}{(2L+1)^2} \sum_{x=-L}^{L} \sum_{y=-L}^{L} u(x,y) \tag{4.8}$$

If the block is 3 × 3, L = 1 and;

$$\beta = \frac{1}{6}[I(+1,\Sigma y) - I(-1,\Sigma y)] \tag{4.9}$$

$$\alpha = \frac{1}{6}[I(\Sigma x,1) - I(\Sigma x,-1)] \tag{4.10}$$

$$\gamma = \frac{1}{9}[I(\Sigma x,\Sigma y)] \tag{4.11}$$

The estimated gray level value for any resolution cell (x,y) in the block is then given by;

$$v(x,y) = \alpha x + \beta y + \gamma \tag{4.12}$$

This shows that α is the slope in the x direction, β is the slope in the y direction, and γ is the average grey level of the block.

4.2 A Review of Applications of the Facet Model

A review of previous work on the facet model is now presented. Haralick[50] first proposed a facet model for those image processing operations that require an image to be in an idealized form. Noise cleaning is a procedure by which a noisy image is operated on in a manner which produces an image which has less noise and has an idealized form. A non-linear relaxation technique to iteratively operate on images and produce a facet image was proposed. By merging regions of similar characteristics, facets of different size and shape would be formed and completely described by polynomials of degree one or of degree two.

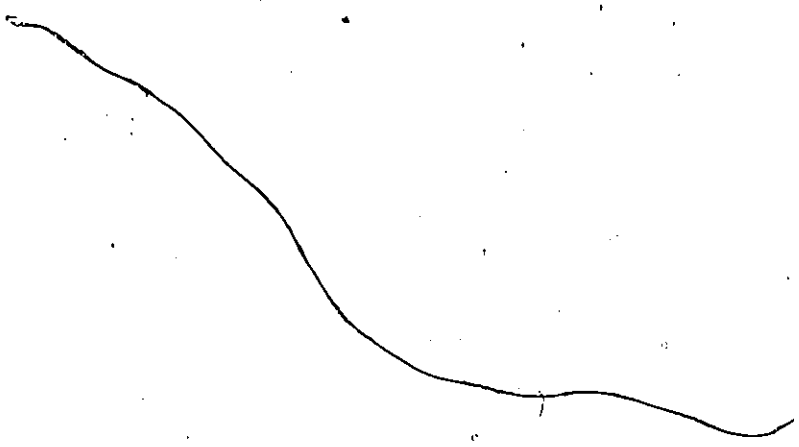
The same facet model was used by Haralick[53] as a method for edge detection and for defining parts of a scene. The approach grows regions by using a significance test to merge similar regions with boundaries ultimately declared between different facets. The regions are a union of B by B blocks of pixels and would have arbitrary size and shape with the narrowest dimension of the facet equal to B. Edges are produced by the boundary points of the facets defined.

A similar method of edge detection and smoothing using facets represented by a polynomial of degree zero was suggested by Nagao and Matsuyama[51] and Tomita and Tsuji[52]. The algorithm looks for the most homogenous neighborhood area around each point in an image and gives each point the average grey level of the selected neighborhood area. This can be regarded as a facet representation using polynomials of degree zero on regions of arbitrary shape and size.

The model has also been used in a pattern recognition application. To classify targets, it is necessary to develop algorithms for systematically analyzing the com-

ponents of a target or an image. A sloped facet model was applied to multi-spectral image analysis[54] to find first order descriptions of targets in terms of their essential components.

It is now proposed to use a facet model for image data compression. A few changes must be made to the previous model to use the facet for image coding purposes. First of all, the regions should be tractable without using any information to describe the region shape. This would preclude irregular shaped boundaries and require regular regions throughout the image so that the block location and shape could always be identified. However, block sizes of variable size are preferred because of the varying statistics of the image. A compromise is, therefore, reached by allowing different block sizes but each of the successively smaller block sizes being a regular division of a larger size. The proposed scheme uses large block sizes for slowly varing areas and smaller blocks for busy areas in the image.



4.3 Image Data Compression Using the Facet Model

In transform coding[10], an image is first divided into blocks of equal size, and then a transform is applied to each block individually to yield a set of coefficients. In general, this transform is linear and can be inverted to recover the original values. The facet representation of an image is, on the other hand, a non-linear, non-unitary transform, which cannot be inverted to recover all the original data. In designing any transform coding system the following parameters must be selected: the coefficients to be transmitted, block size, the bit assignment and quantization strategy.

Results are presented in section 4.3.1 for regular sized shaped facets of degree one, i.e. those described by three coefficients on equally sized blocks. This scheme is made adaptive by varying the size of the facet, the results of which are presented in section 4.3.2. In section 4.4, bit assignment and quantization strategy are discussed.

4.3.1 Fixed Degree Polynomial, Fixed Block Size

The algorithm divides the image into spatially contiguous blocks that are regular in shape. Each block is regarded as a facet and transformed by a polynomial function. The size of the block and degree of the polynomial will determine how much error is produced and how much compression is achieved by the transform. Blocks that transform well are not necessarily blocks with low energy, but low energy blocks transform well. That is, blocks of rapidly changing values will be modeled well provided they can be modeled as a plane. Some edge regions may be characterized as such. An example of an edge that transforms well is shown in figure 4.2(a). An example of an

edge that does not transform well is shown in figure 4.2b. Generally smaller block sizes will transform better. If the block size is small enough for the polynomial chosen then it can be represented errorlessly. Large blocks can be used in regions of slowly varying grey levels.

Experiments are performed using the planar facet representation on block sizes of 2 by 2 and 4 by 4. When 2 by 2 blocks are used, four points are being modeled by the closest plane. As each of these four member blocks cannot be modeled as a plane, some error results. Since a plane is completely described by three non-colinear points, a three member block could have been represented exactly. It will be seen that because of the strong degree of correlation between adjacent points in an image, 2 by 2 blocks are represented with very little error and no observable degradation in subjective quality. Using 4 by 4 blocks with a planar facet model represents 16 points by 3 points and much more error may result. But because of the inherent order the test images (still life images) many large blocks are well represented with very small error.

Results

Three test images, FRA, BAT, TAB are shown in figure 4.3. The images are 256 by 256 with 8 bits per pixel and represent a cross section of characteristics and coding peculiarities. The 'FRA' image has two large uniform areas to the left and the right of the face. The face itself contains some slowly varying grey level regions with some regions of high detail. The constant and slowly varying grey level regions dominate the image statistics as a whole and if uniformity in statistics is assumed for the image, poor coding in the high detail areas results. The 'TAB' image contains many edges characterized by two adjacent uniform regions. It also contains many

slowly varying grey level regions. The edges are in all orientations. The 'BAT' image is a highly detailed image with many edges in the form of lines or as borders between two adjacent regions.

The method is first demonstrated on the three test images using 4 by 4 blocks and 2 by 2 blocks using a polynomial of degree one, i.e. a planar facet representation.

4 by 4 pixel blocks, planar facet. Using 4 by 4 blocks, 16 eight-bit pixels are represented by 3 eight-bit coefficients for a reduction from 8 bits/pixel to 1.5 bits/pixel. The error histograms for each of the test images is shown in figure 4.4. It is seen that a large percentage of the blocks are well represented i.e. yielding a small error. It is also apparent that some blocks are represented poorly and this leads to poor detail in some areas of the image. (see figure 4.5)

2 by 2 pixel blocks, planar facet. Using this size block results in a reduction in the bit rate from 8 to 6 bits. Using a block size of 2 by 2, 4 eight-bit pixels are represented by 3 eight-bit coefficients, giving a reduction in bit rate from 8 bits/pixel to 6 bits/pixel. It is clear from the images in figure 4.7 that even the high detailed areas are coded well. This is apparent by examining the histogram of errors of figure 4.6. In all three images more than 99 percent of the image can be coded with less than .1 percent NMSE.

By comparing the two histograms it can be concluded that a large percentage of the image derives little or no additional benefit by being represented with a smaller block. However, it can also be concluded that some blocks must be represented by small blocks for good image fidelity. This suggests that a combination of block sizes

may have superior results. This possibility is explored in the next section.

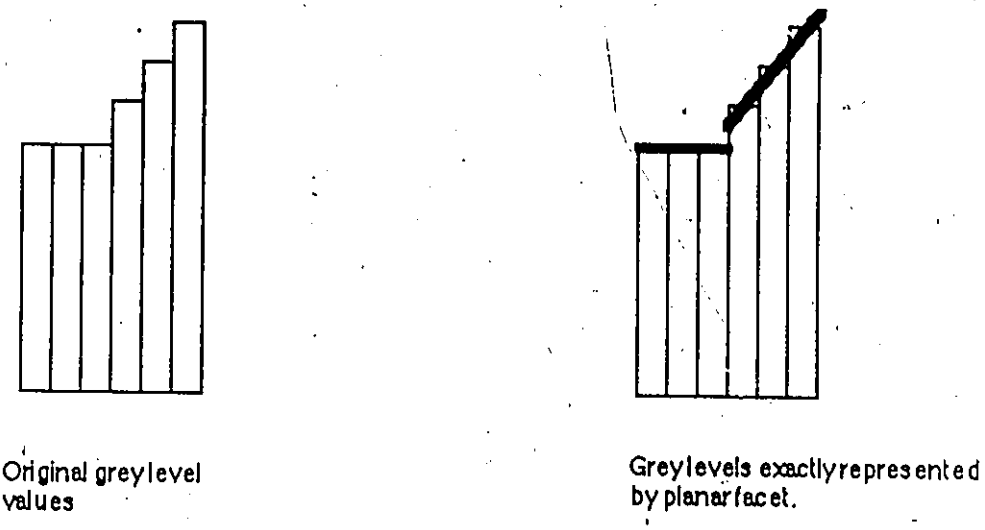


Figure 4.2 A two dimensional version of image blocks that are represented well. Low energy blocks will always be represented well, but high energy blocks may or may not be represented well, depending on some additional statistics.

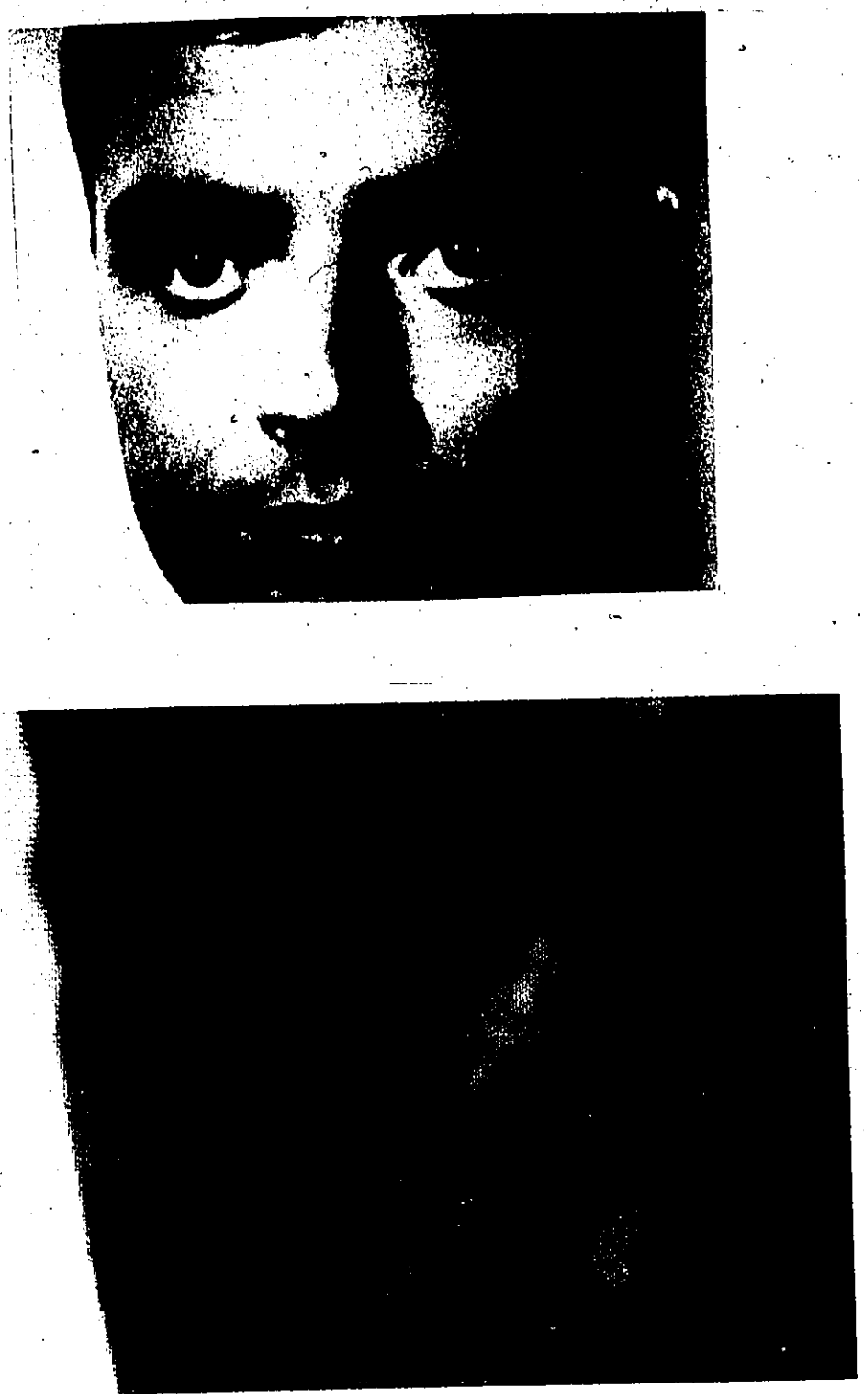


Figure 4.3 Photographs of test images used throughout the thesis. All results presented will follow this same format ; the entire image is photographed followed by an enlargement of a small section. top;FRA,middle;bat,bottom,TAB

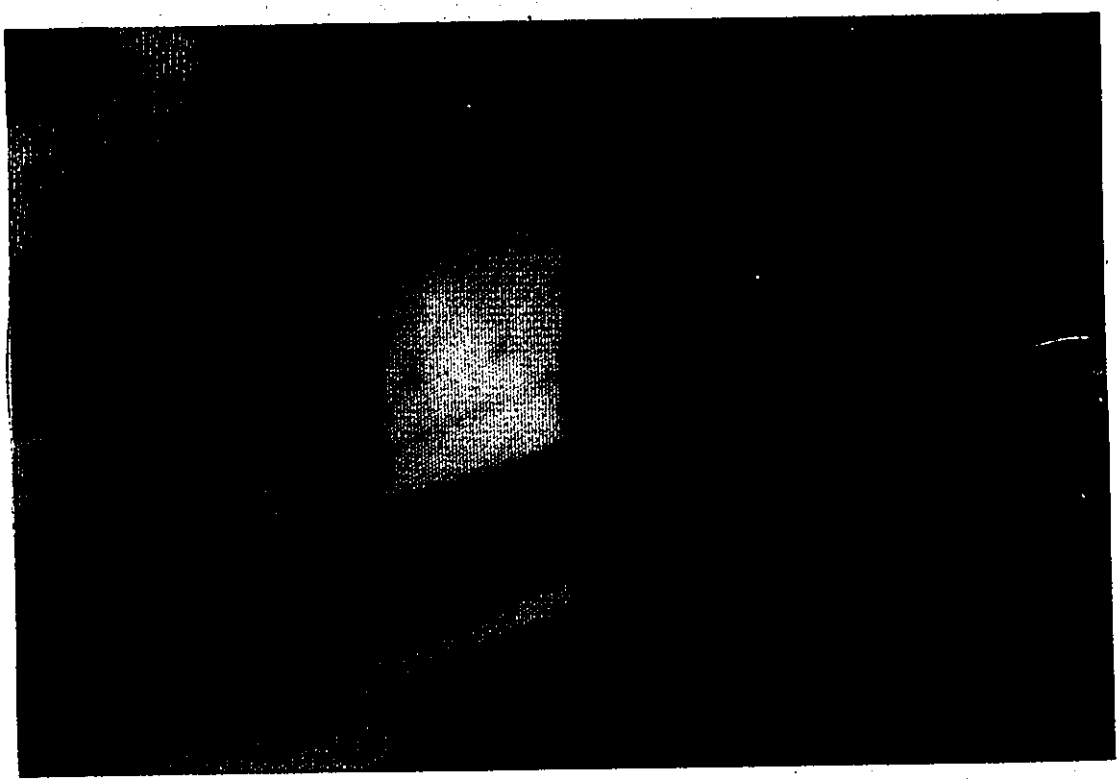
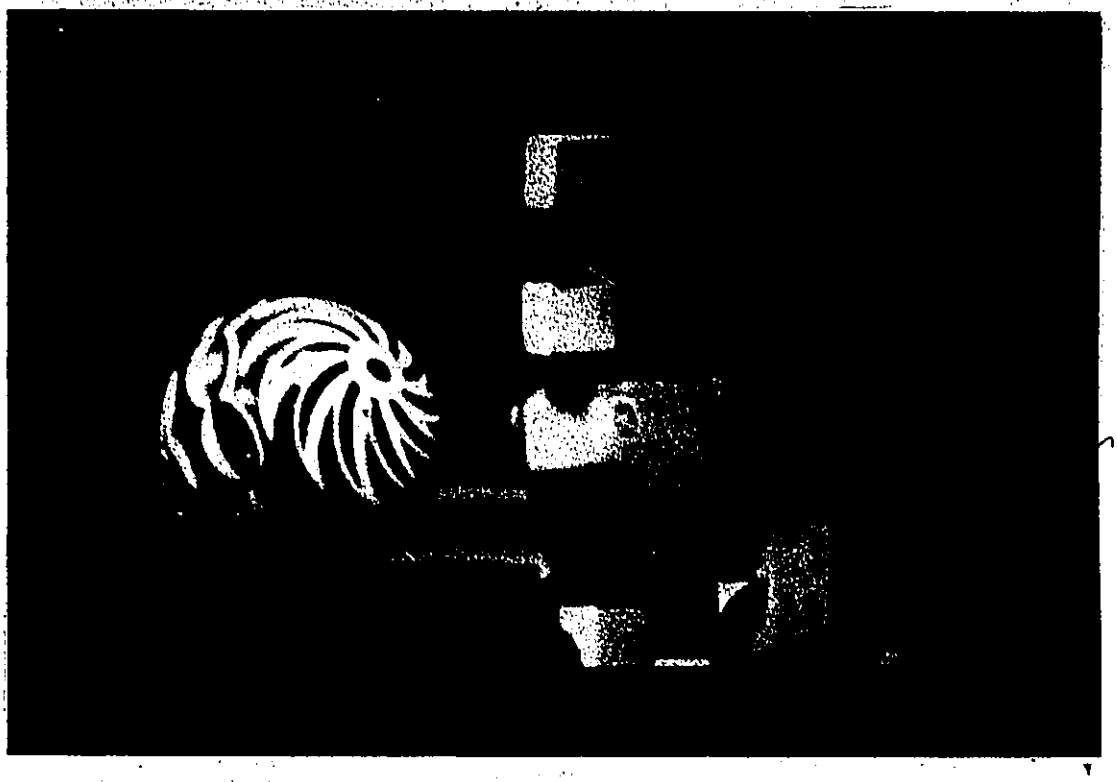


Figure 4.3 (continued)



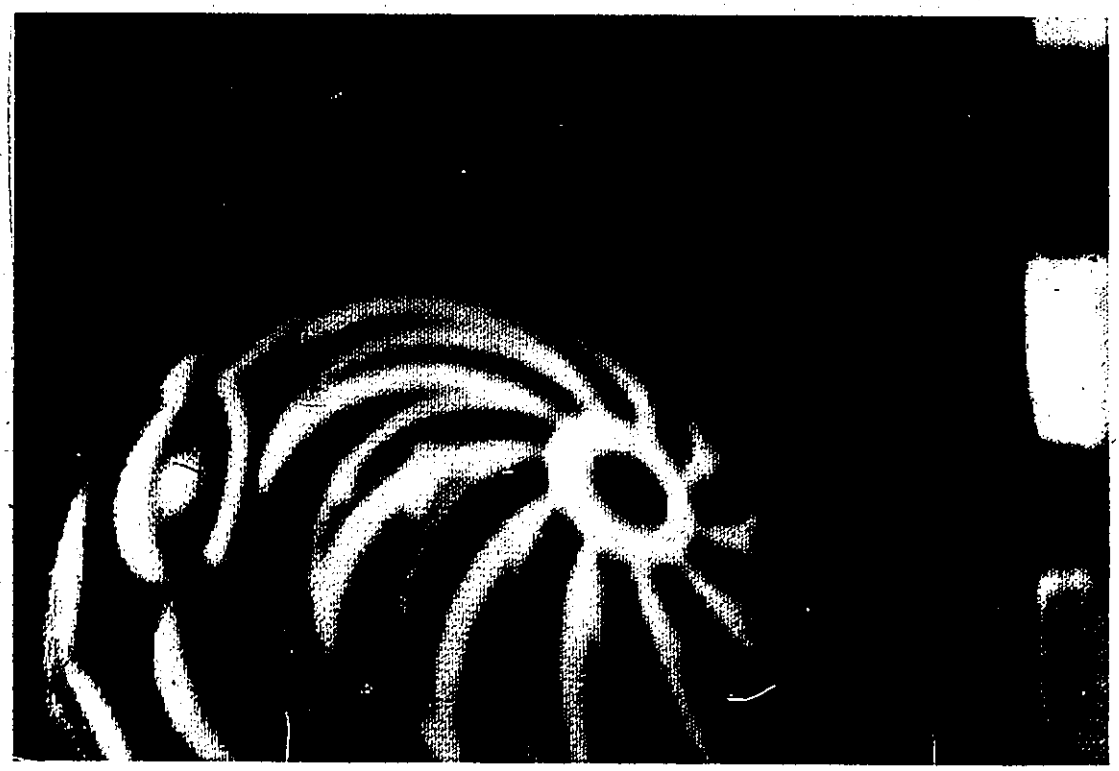
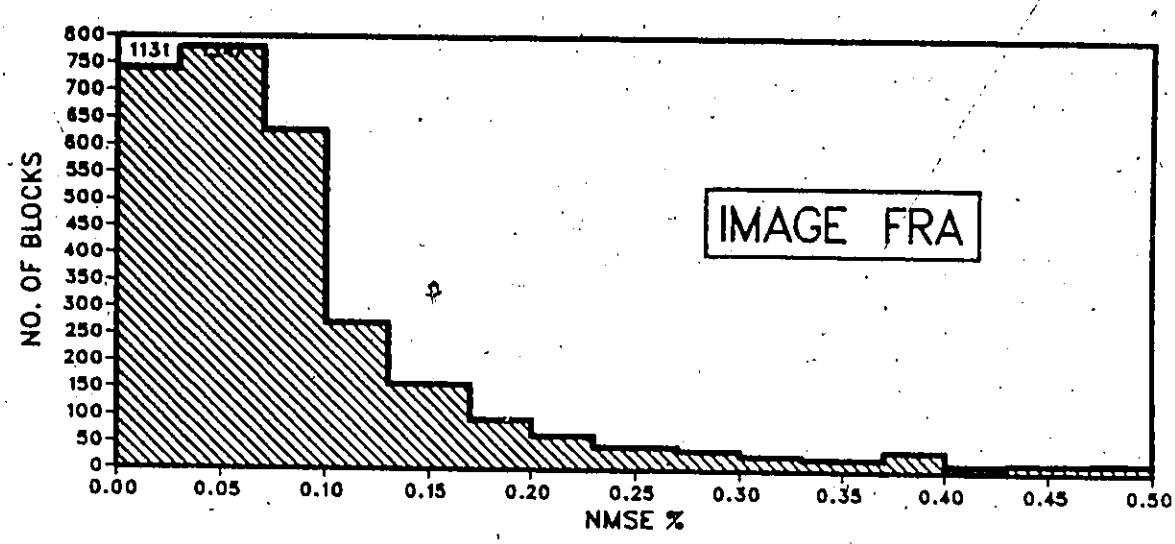
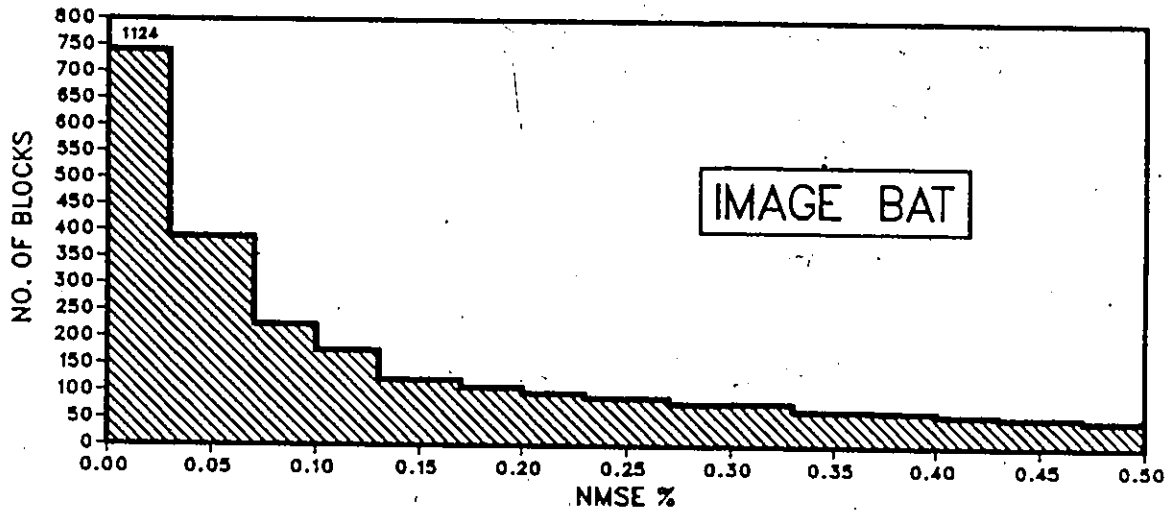


Figure 4.3 (continued)





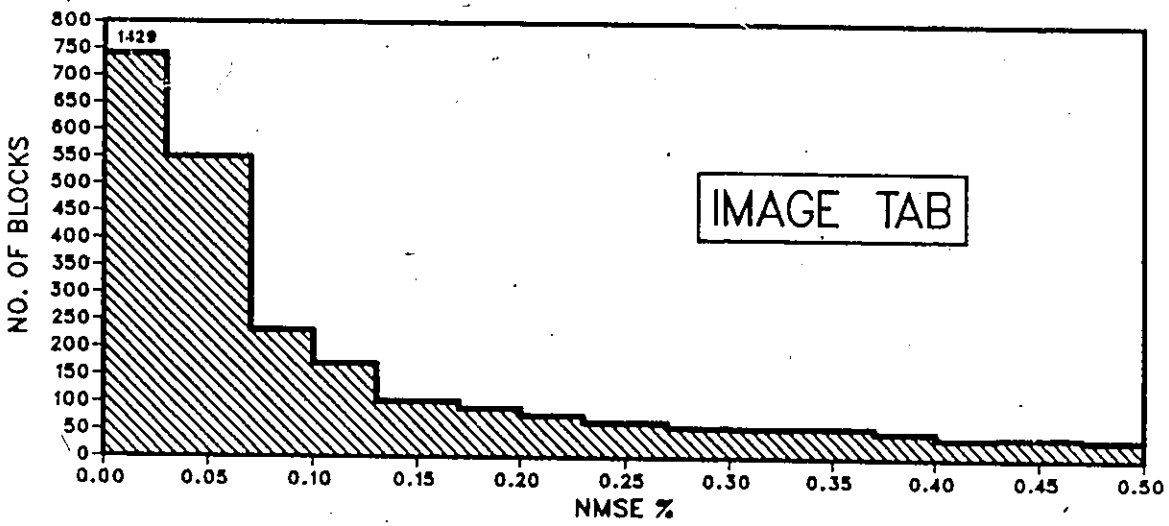


FIGURE 4.4. Histogram of error when the image is represented using 4 by 4 blocks with planar facet representation. Most of the image can be represented with less than .1% NMSE. The blocks that cannot be well represented lead to the artifact that can be seen in the images of of figure 4.7.

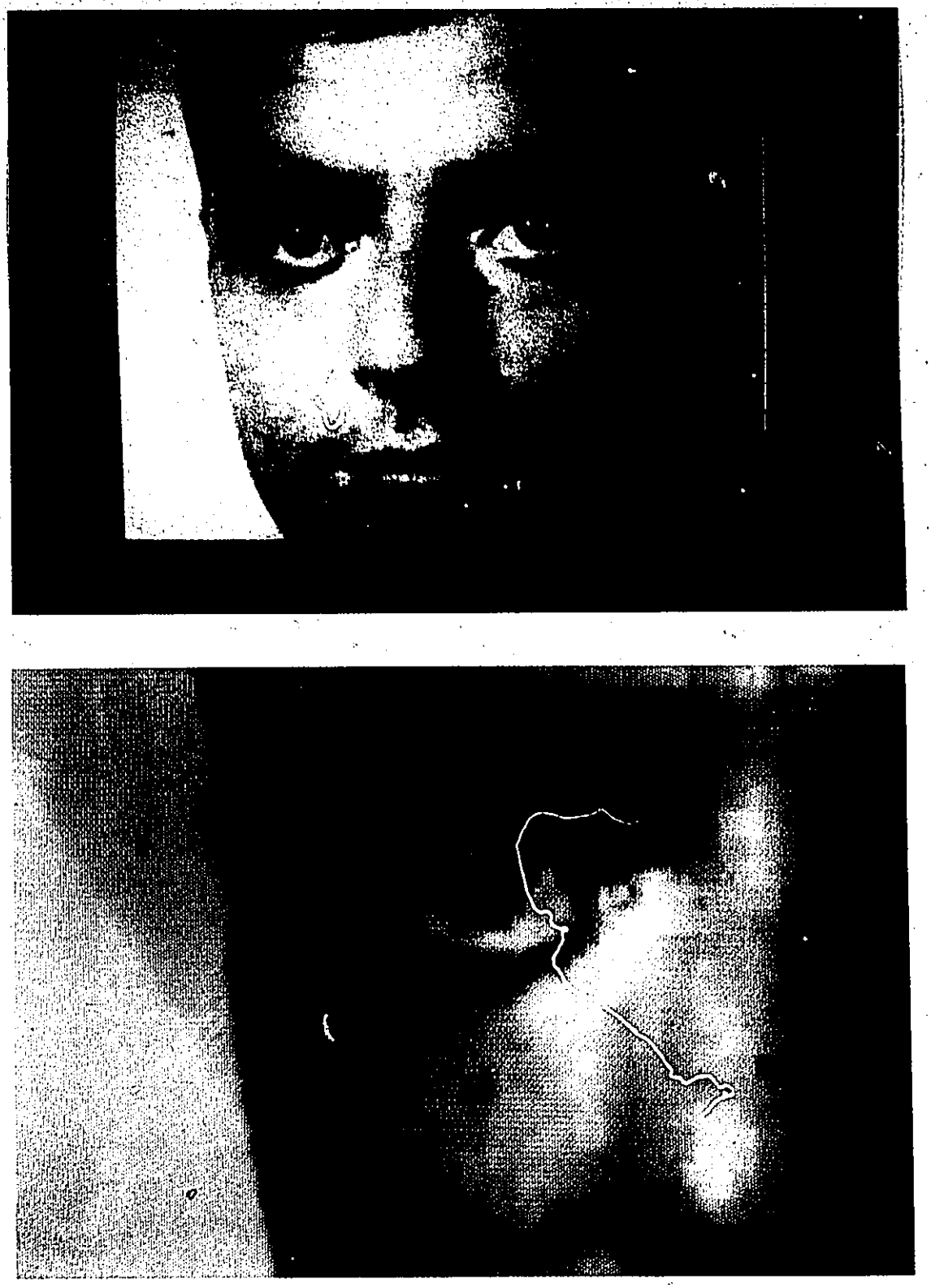
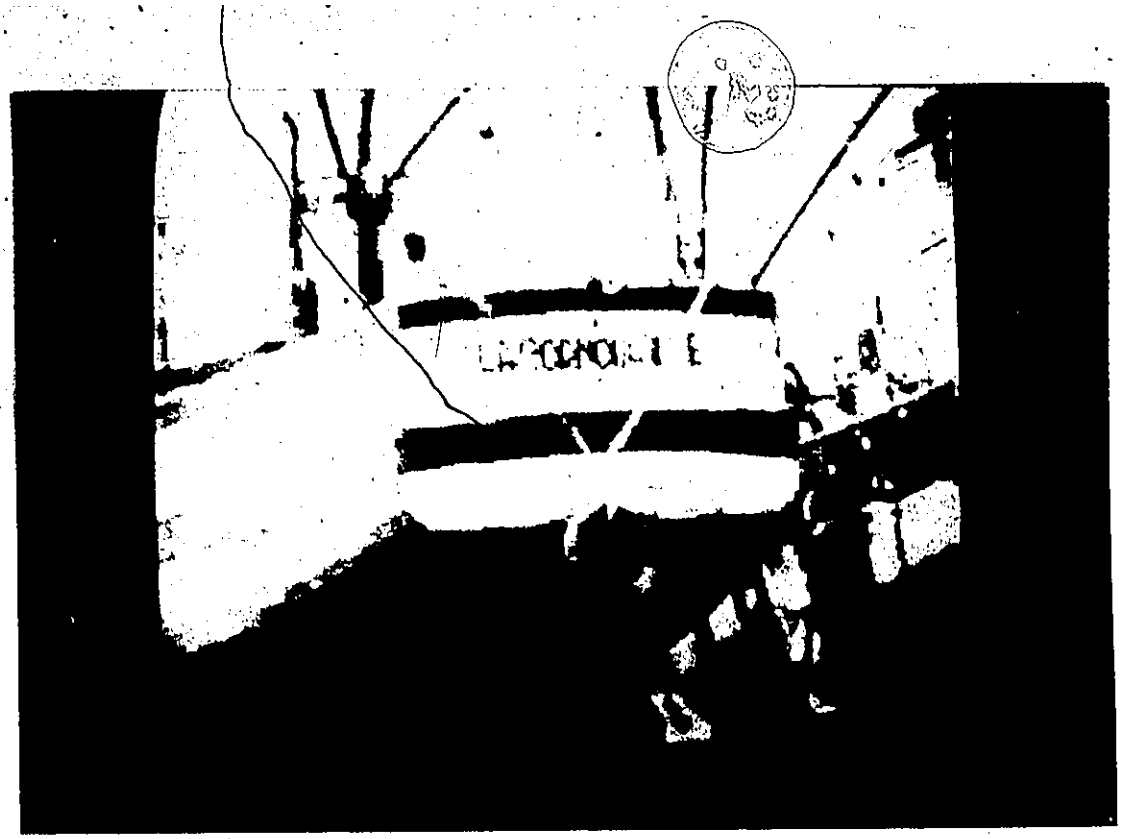


Figure 4.5 Photographs of the three test images represented completely by 4 by 4 blocks, planar facet transformed. The bit rates for all these images is 1.5 bits per pixel.



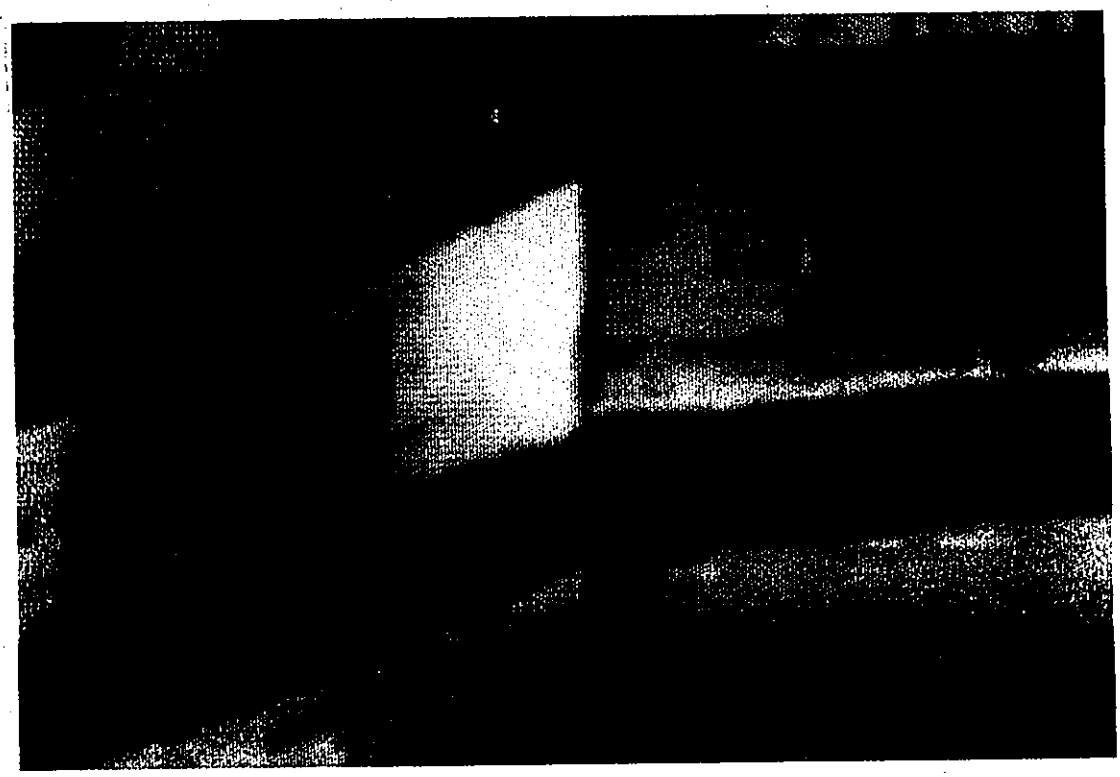


figure 4.5 (continued)

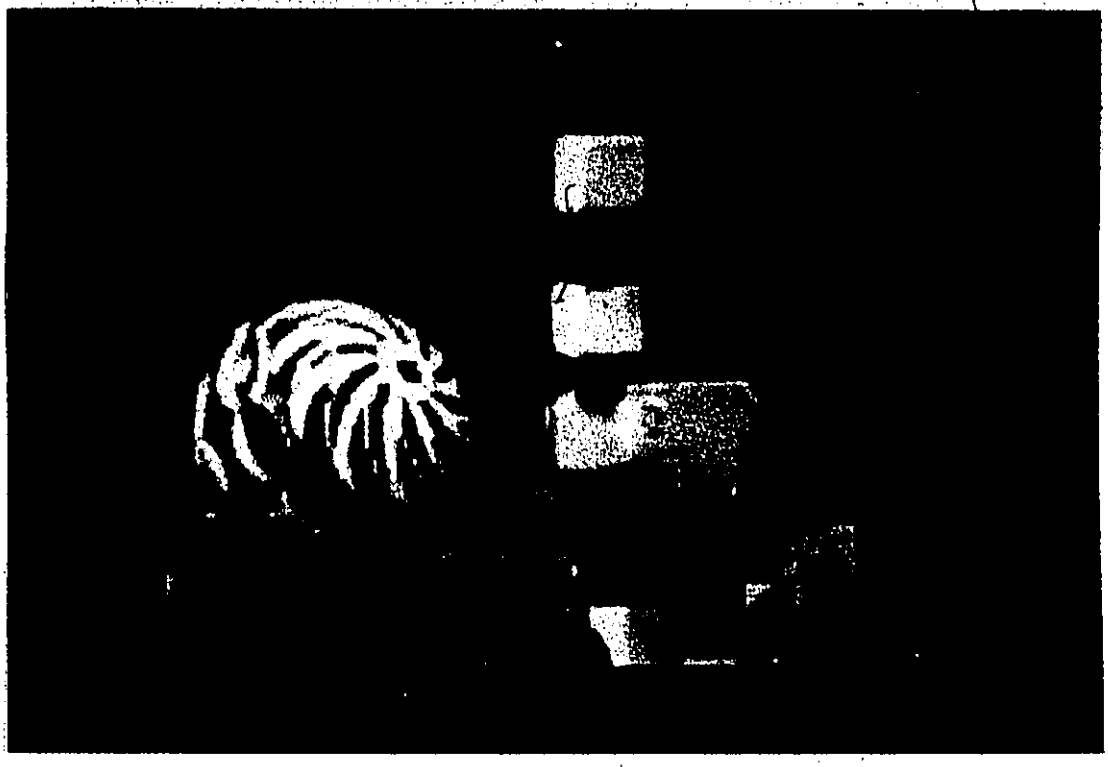
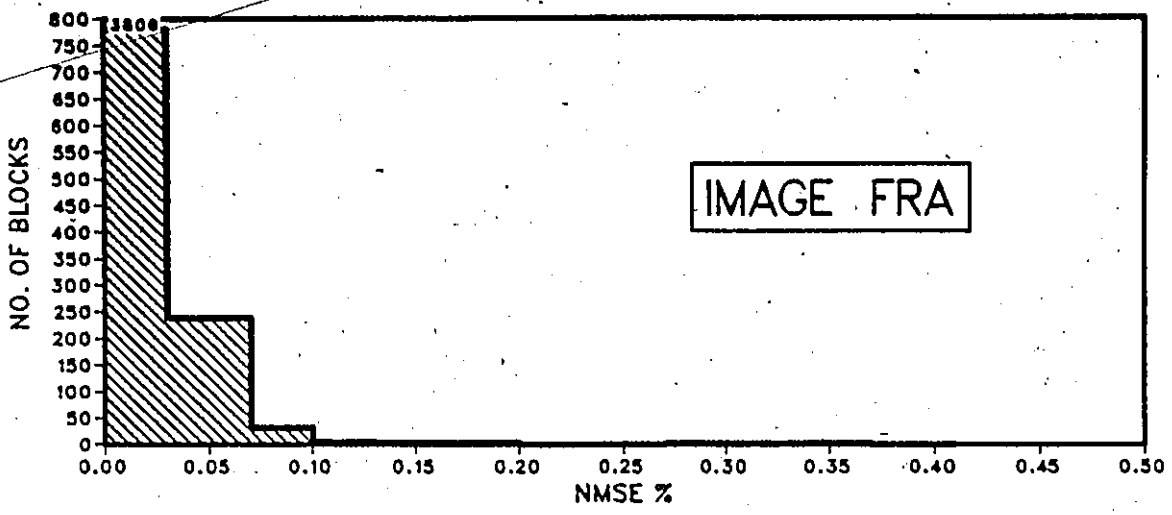
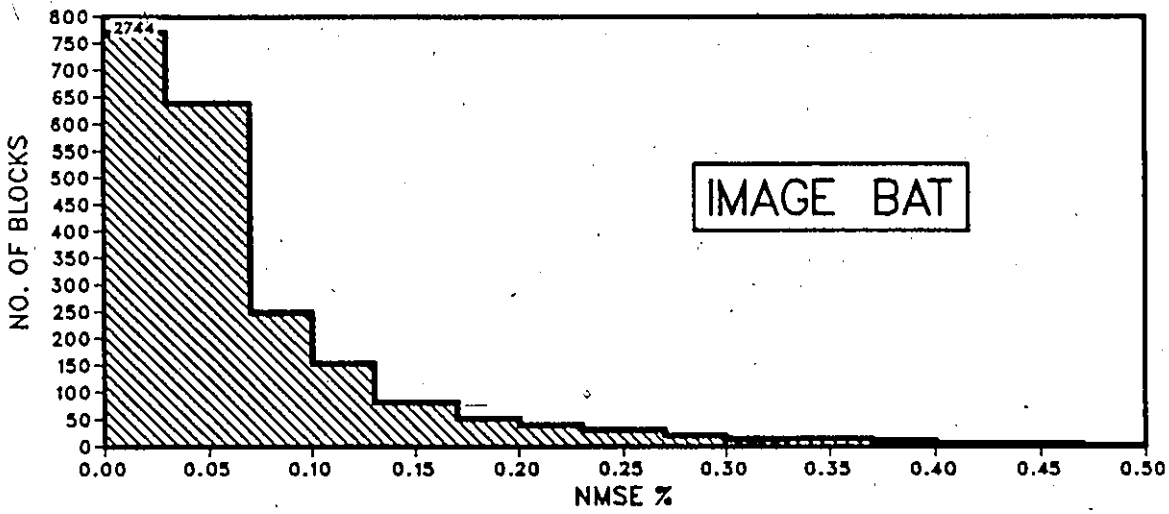


figure 4.5 (continued)





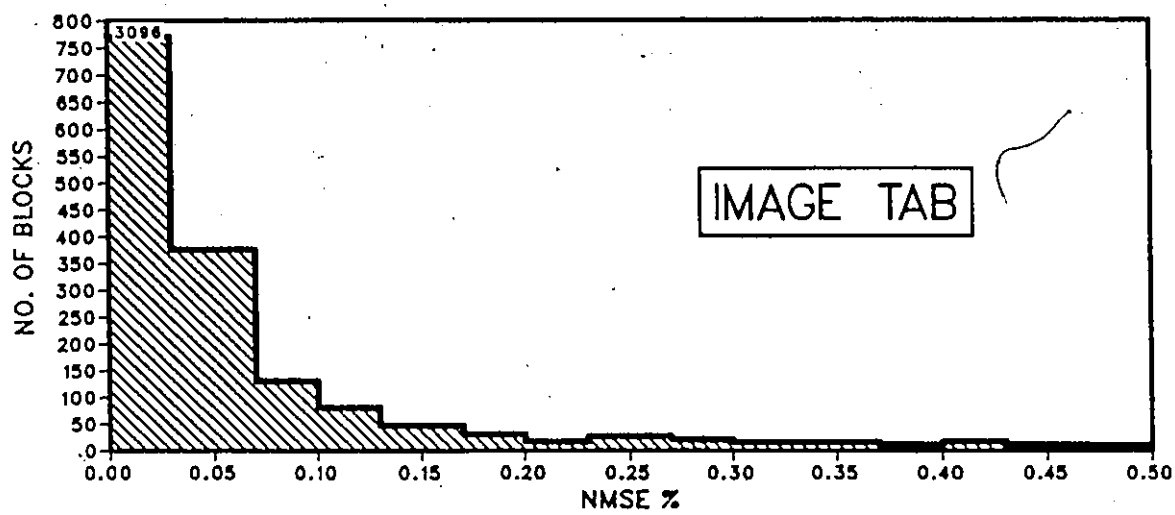


FIGURE 4.6. Histogram of error when the image is represented using 2 by 2 blocks with planar facet representation. Note that the number of blocks that benefit by the smaller block are few, but these blocks make a substantial difference in the subjective quality of the the image, see the images of figure 4.7.

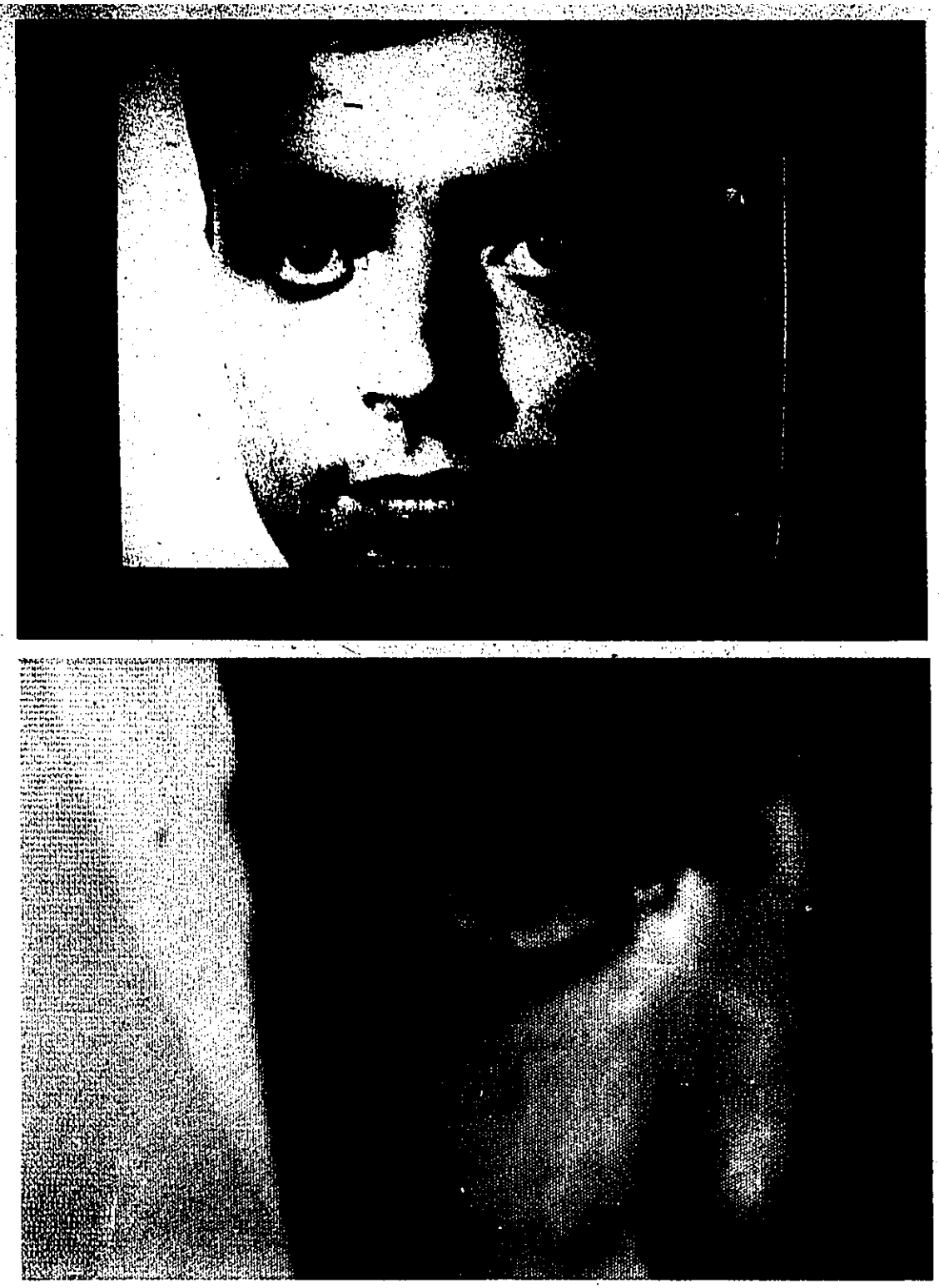


Figure 4.7 Photographs of the three test images represented completely by 2 by 2 blocks, planar facet transformed. The bit rates for all these images is 6.0 bits per pixel.

figure 4.7 (continued)

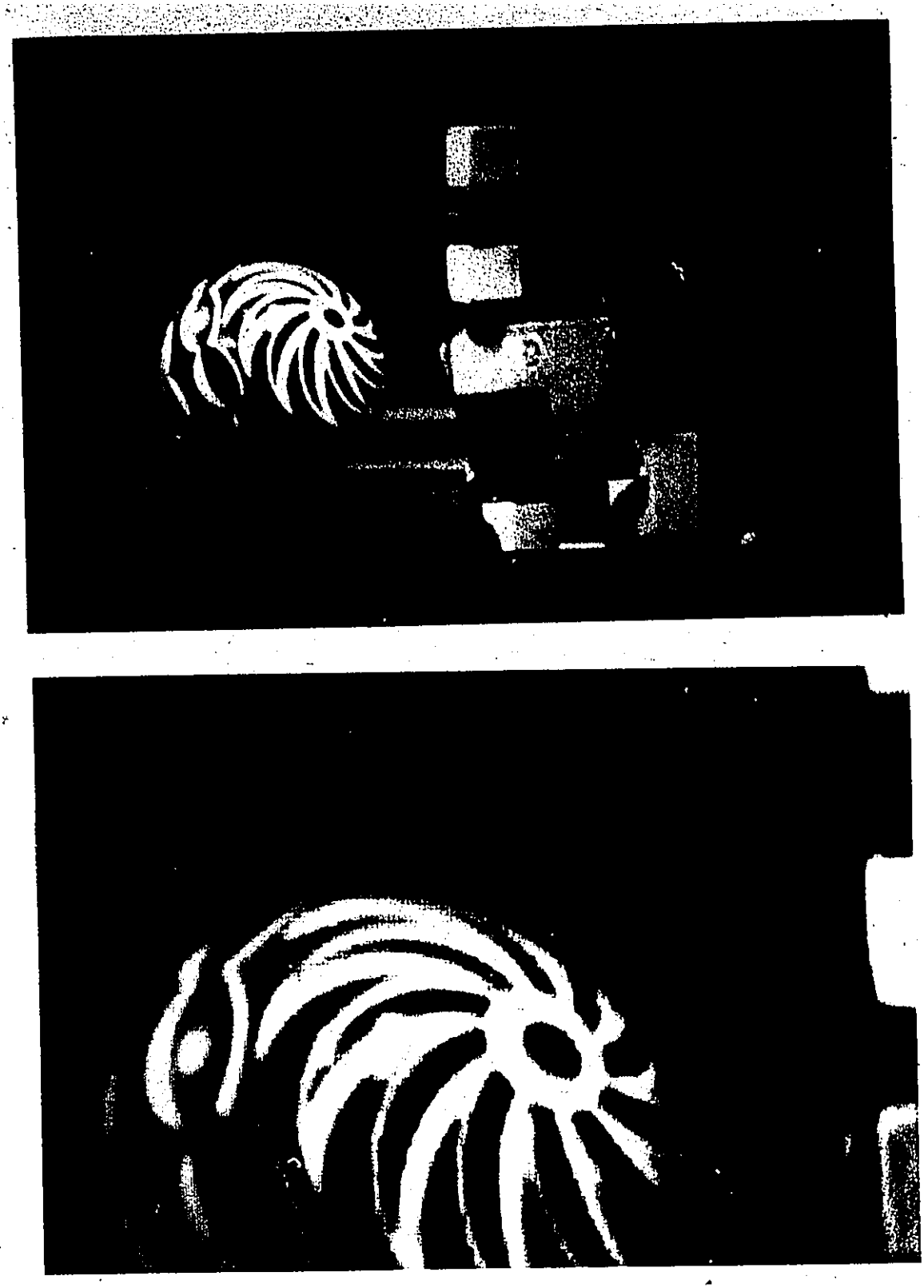


figure 4.7 (continued)

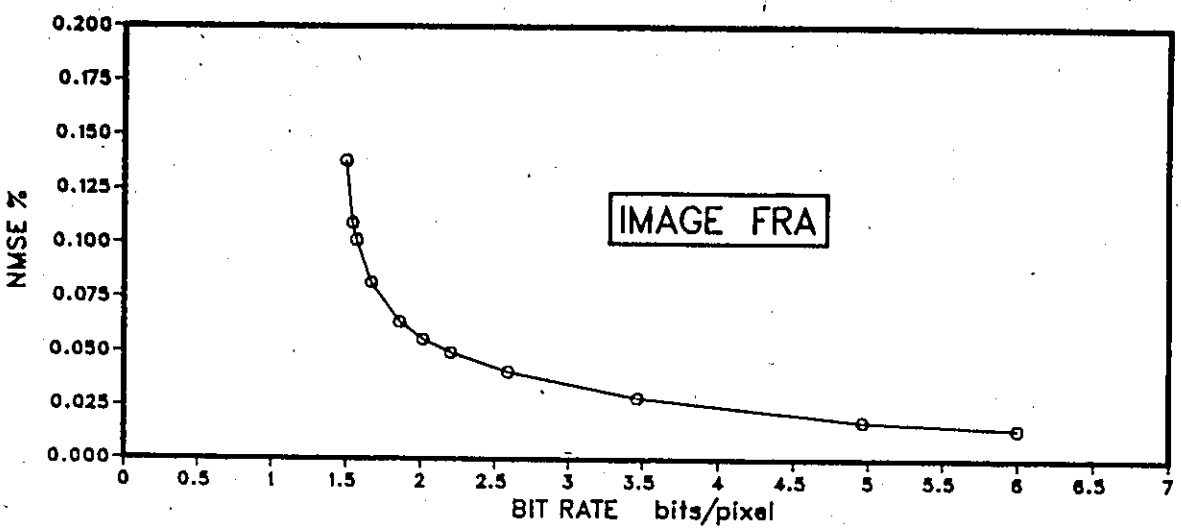
4.3.2 Fixed Degree Polynomial, Variable Block Size

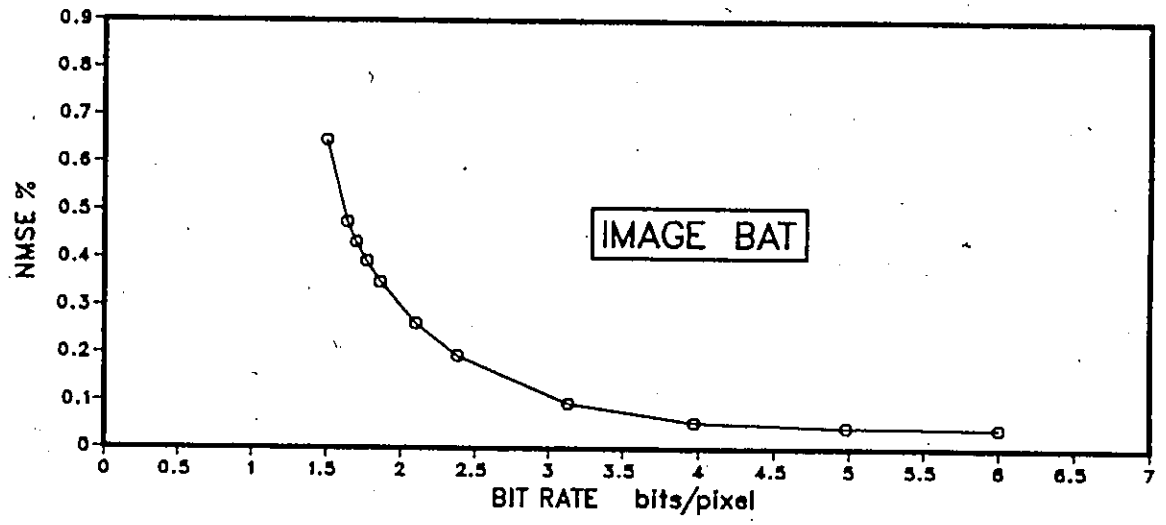
From the experiments in the last section it was apparent that a uniform block size does not represent all parts of an image with equal fidelity. The method now introduced uses different block sizes to code the regions of varying statistics. The image is partitioned into contiguous blocks as before. If the error in a block when represented by a facet exceeds some threshold, then the block is subdivided into smaller regions and represented again by a facet. If the error still exceeds some threshold, the block is split still further. Thus a pyramidal structure is set up with each subdivision creating a new level. This method needs an additional bit per block of overhead to record the binary decision of subdividing. As an example of the overhead, if blocks of size 4 by 4 or 2 by 2 are used, an additional 1/16 bit per pixel is needed. If three levels, 8 by 8, 4 by 4 and 2 by 2 are used, then 1/16 plus 1/64 bits per pixel is added. This method would result in a variable length coder which depends on the image statistics and the error threshold chosen.

Results

This approach is demonstrated on the test images by using blocks of size 4×4 and 2×2. Blocks size of 4 by 4 are transformed using a planar facet model and represented by three coefficients. If the three coefficients result in too much error the block is further divided into 2×2 blocks. Each of the 2×2 blocks are then represented by a planar facet. The results of using variable block sizes are shown in figure 4.8 for different thresholds of error. A change in the error threshold results in a change in the percentage of the image represented by 2 by 2 blocks and in increased error. Note the very definite elbow in the curve, showing the diminishing marginal

gains in further sub-dividing the blocks. Despite the wide range of statistics of the images the elbow occurs around 2 bits per pixel. This means about 12 percent of the image is represented by 2 by 2 blocks. The amount of error that this bit rate creates varies widely from image to image.





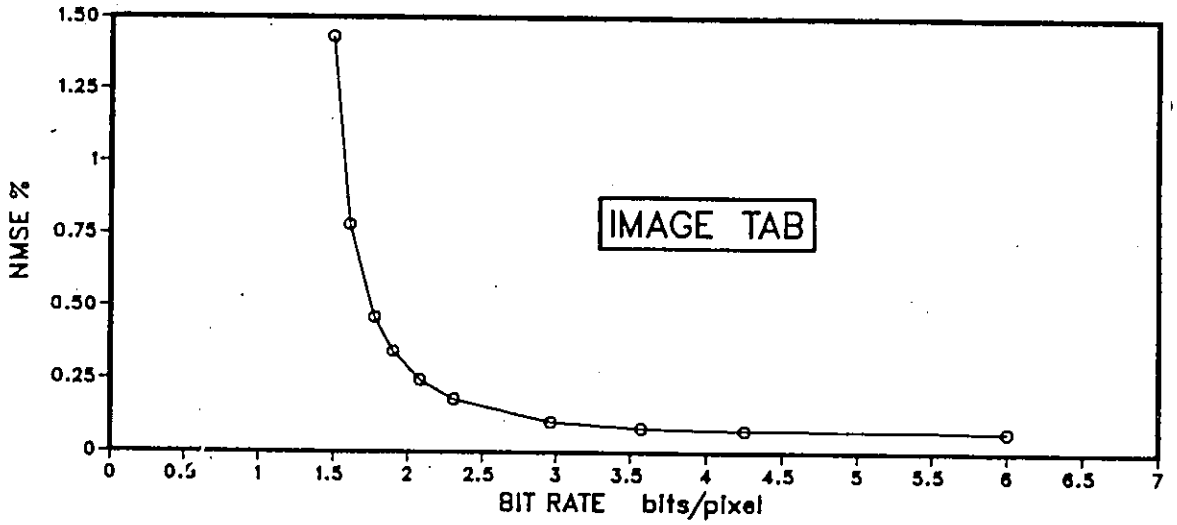


FIGURE 4.8. NMSE VS BIT RATE shows the rapid decrease in the errorcs the bit rate increases. The rate does not include overhead which would add no more than 1/16 to the rates shown.

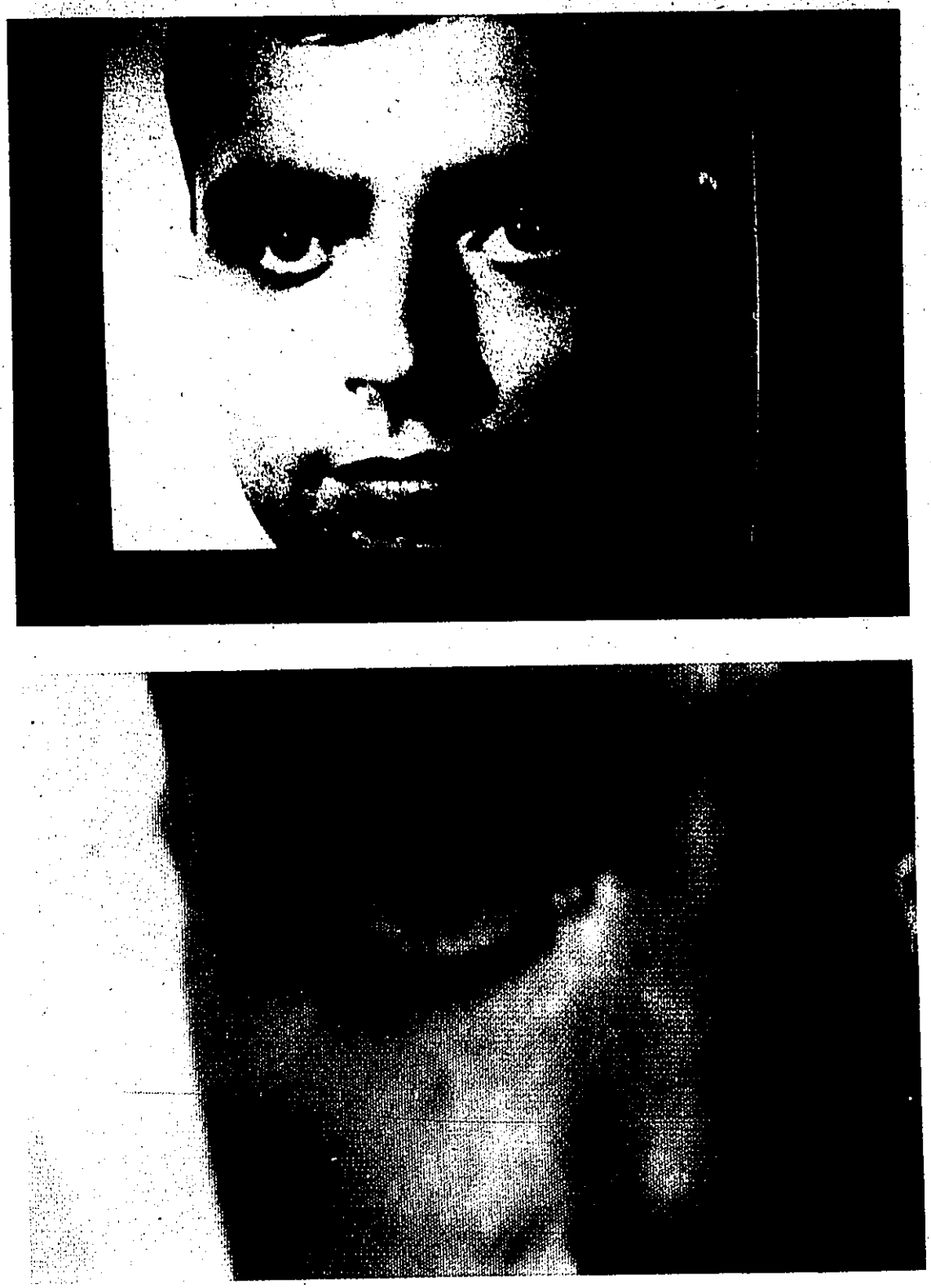


Figure 4.9 Photographs of the three test images represented by 25% 2 by 2 blocks and 75 % 4 by 4 blocks. These images clearly show that it is not necessary to code the entire image by 2 by 2 blocks. The bit rates for all these images is 2.5 bits per pixel.

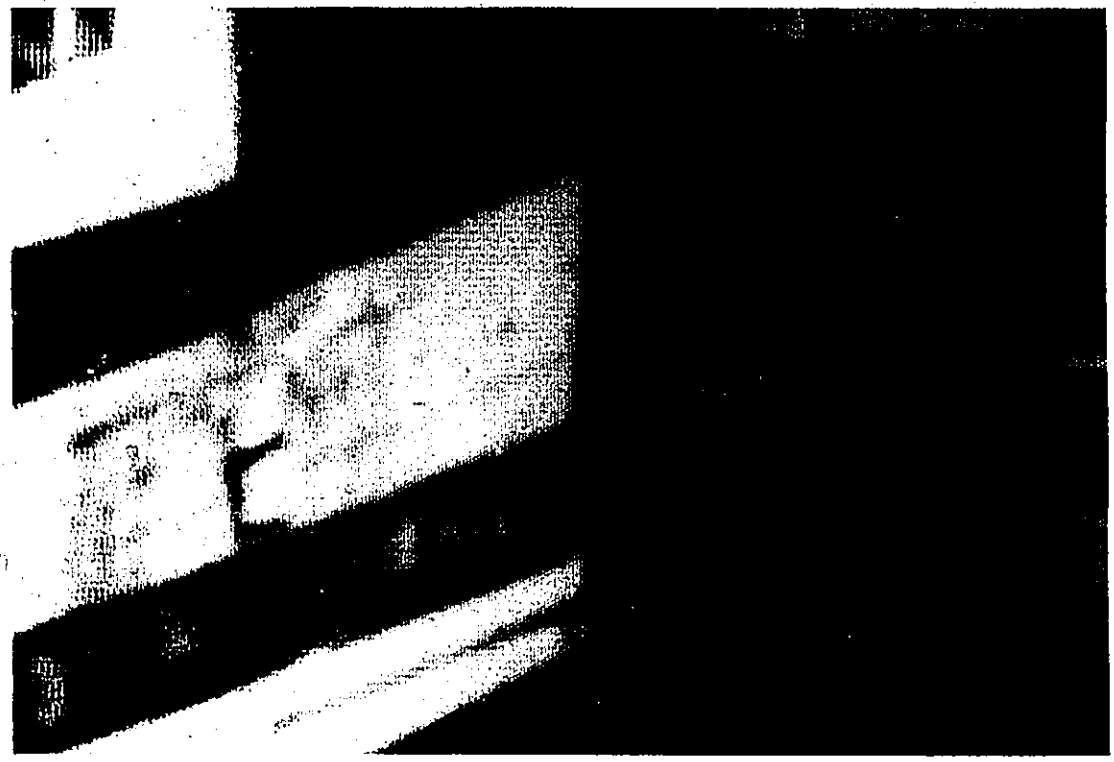
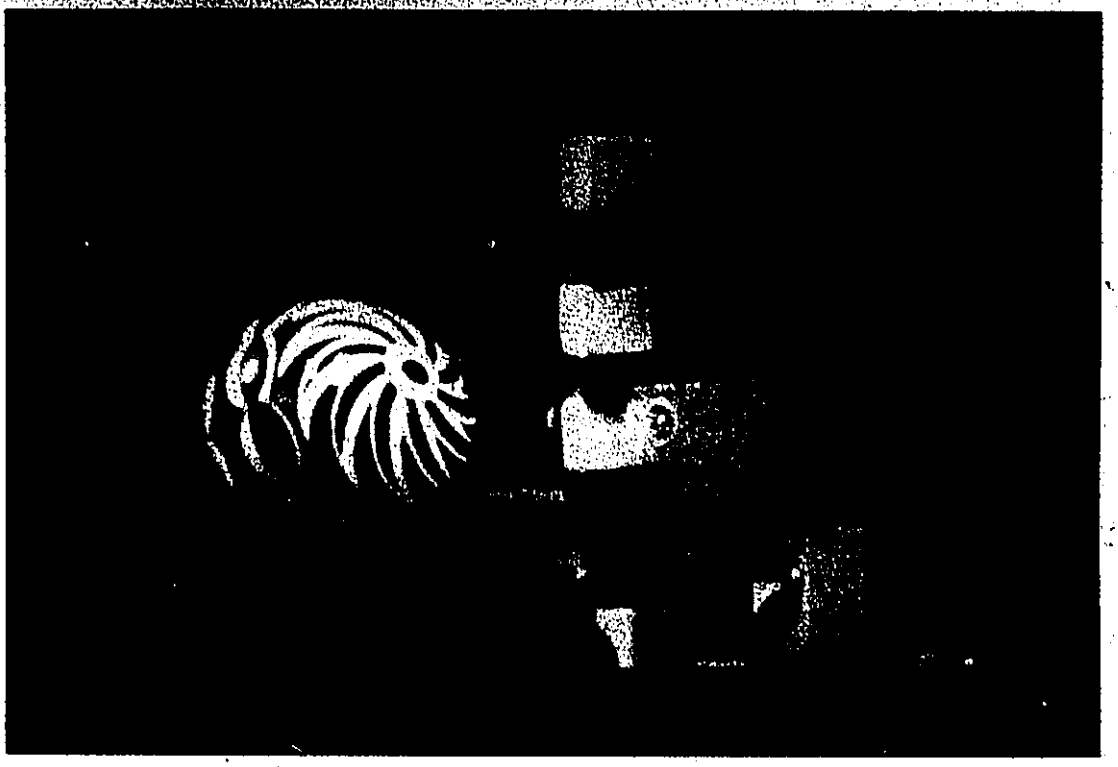


figure 4.9 (continued)



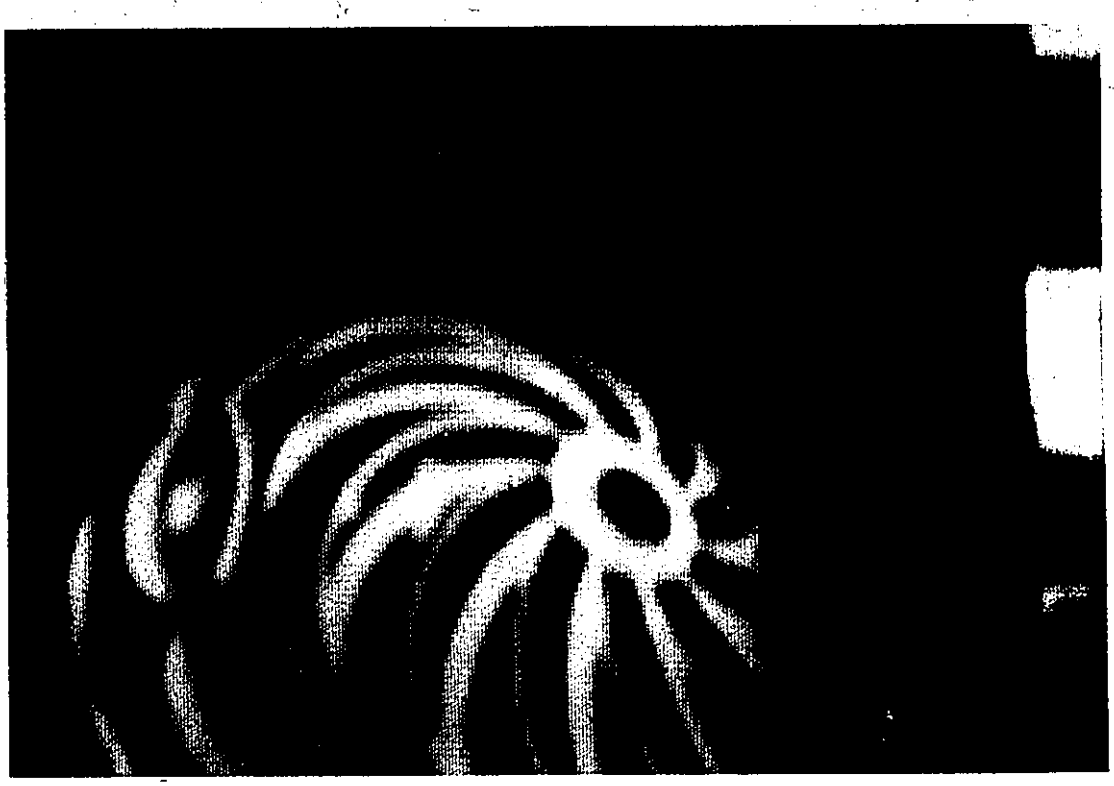


figure 4.9 (continued)

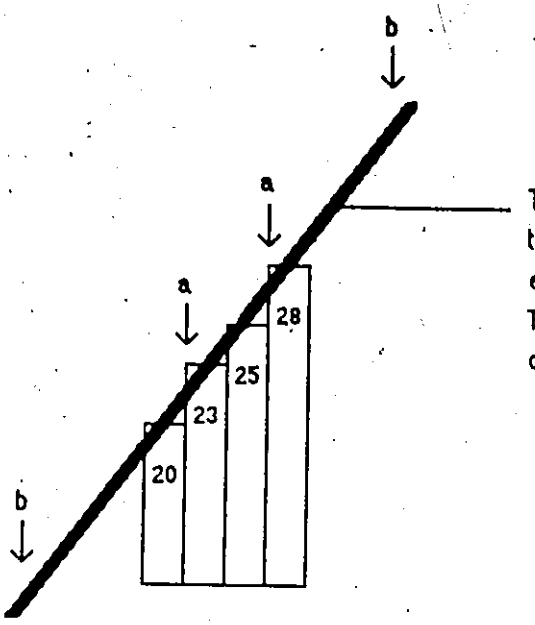
4.4 Quantization and Bit Assignment

The real valued output of the transform operation requires procedures for quantization and bit allocation. It is well known[55] that the quantization operation must be carried out with the distribution of variables in mind. Quantizing to minimize the total average error will lead to quite a different solution to the problem when compared with maximizing the visual acceptability. For the sake of simplicity, all the outputs of the coders in these experiments have been quantized to the nearest integer between 0 and 255, despite the sub-optimality of this scheme. The result of using this quantization scheme has little effect on the final output. This is evident from the histograms of grey levels in figure 4.3. The three images have no grey level values are significantly more frequent then any others, with the exception of grey levels at the extreme ends of the scale, which are less frequent. Thus a uniform quantizer would have results near to an optimum quantizer.

As in the case of quantization, the matter of bit allocation of a set of variables has been extensively treated in the literature[56]. Adaptive transforms will in general allocate different number of bits to different blocks depending on the variation in detail. The adaptive facet transform will allocate the same number of coefficients to each block, but adapts the block size to the variation in detail. This is a sub-optimal scheme because only two block sizes are used.

One final note on the quantization scheme used in the experiments. If we regard the facet model as three points in space defining a plane, the distance between these points determine the quantization error. The further these points are from one another, the lower the quantization error, but the greater the dynamic range of the output required. This is best explained by way of example. Figure 4.10a shows 4

points that all lie on the same plane and therefore can be exactly represented by the planar facet. However, the three coefficients that represent this block are not integers and must be quantized. Figure 4.10b shows the same four pixels being represented by the planar facet with the points chosen further from one another. Note the quantization error in the second case is much less. When the cases arise that the four grey level values are closer to 255 or 0 , then the magnitude of the planar facet coefficient would be greater then 255 or less that 0. This is because the points chosen to represent the plane are well out of the spatial block (see diagram). The same effect could be accomplished by additional quantization bins. Observing the histogram of the test images, one notes that most distributions of real image will have few gray levels at the extremes, at 0 and 255. Using this premise, the three points were chosen to be on a circle with twice the diameter of the block it defines. For the few cases that the output exceeds 255 or goes below 0 the output is hard limited to these values.



Using points aa to represent a line that best fits the grey values results in the two points 21.5 and 26.5. These values are then quantized to 21 and 26 resulting in reconstructed values of 20,22,25,27 to represent the original block of 20,23,25,28.

Using points bb to represent a line that best fits the given grey level values results in points 10.3 and 37.3. These values are quantized to 10 and 37 resulting in a reconstruction vector of 20,23,25,28.

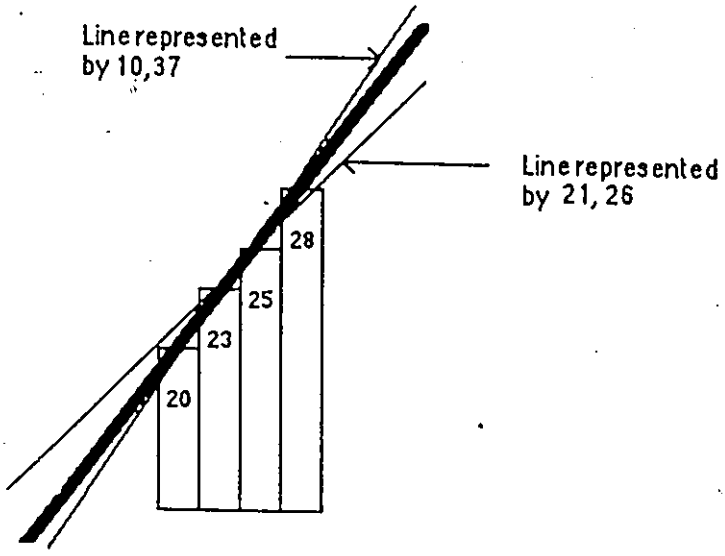


Figure 4.10. THe figures show the difference in quantization error due to the positioning of the points taken to represent the line. By using points bb less quantization error is seen. Also by using points bb, leads to the possiblity of having the points lie outside of the range of the grey levels.

4.5 Conclusion

A facet model for image coding which uses a set of piecewise polynomial functions to represent regularly shaped blocks of an image is introduced. Representing an image by a polynomial function of fixed degree and fixed dimension can lead to regions being represented by different degrees of fidelity. By allowing the regions to have different sizes, the error tends to be more uniform in error content. The results of the experiments show, that for a class of still images, mixing 4×4 and 2×2 blocks results in subjectively excellent reproduction of coded images.

5.0 Planar Facet Vector Quantization

Results were given in chapter 4 for planar facet representation of images with scalar quantization of the vector elements. Bit rates between 1.5 and 6.0 bits per pixel were shown with some visual distortion appearing at rates below 2.0 bits per pixel. Shannon's coding theorem suggests that a lower bit rate can be had, within the bounds of the given distortion, by using vector quantization. Results are now presented for images coded using vector quantization on a planar facet representation (PFVQ), reducing the bit rate to between 0.3 and 1.5 bits per pixel.

For the set of test images examined, the planar facet representation requires that for high detail regions of the image the smallest spatial block represented by a plane be 2 by 2 pixels in order to have no subjective visual distortion. Slowly varying regions be represented with 4 by 4 blocks with the same level of average distortion. The percentage of the image that requires the smaller blocks depends on the image statistics and on the level of distortion. In section 5.1.1, experiments are reported on the percentage of the image that require 2 by 2 blocks for a given distortion . The experiments are performed by having a single codebook size and varying the percentage of 2 by 2 and 4 by 4 blocks. These results are then compared to the rate distortion curves of the scalar case given in figure 4.8.

Because the lowest distortion using VQ is that of the best fitting plane (best in a NMSE error sense), it would be expected that distortion vs codebook size would see a diminishing gains phenomena as the codebook size increases and the error approaches its minimum. Section 5.1.2 gives results of experiments for distortion vs codebook size for a fixed percentage of 2 by 2 blocks. Section 5.1.3 compares PFVQ using a codebook for each block size with PFVQ using a single codebook. Because a variable block size tends to transform both slowly and rapidly changing regions onto the same plane, one would expect that a single codebook would have similar results

as compared with a partitioned codebook. Finally, in section 5.2, PFVQ is compared with the cosine transform, a quadtree method, and a partitioned codebook method.

The block diagram for PFVQ is shown in figure 5.1. Equal sized contiguous blocks of 4 by 4 are represented by the planar facet using the transformation given by equations 4.9 to 4.11. This produces a set of vectors with three coefficients for each of the spatial blocks. Each of the vectors is now inverse transformed using equation 4.12 and an approximation to the original 4 by 4 is produced. The three coefficients are retained. For each of the reconstructed blocks, the NMSE is calculated with respect to the original and compared to some given threshold. If the representation results in an error greater than the threshold then the block is split into four blocks of size 2 by 2 and a new representation found for each block. The vector set is now vector quantized according to the algorithm given in section 3.1. There are two possibilities for the codebook generation; a separate codebook for each block size or a single codebook for both block sizes.

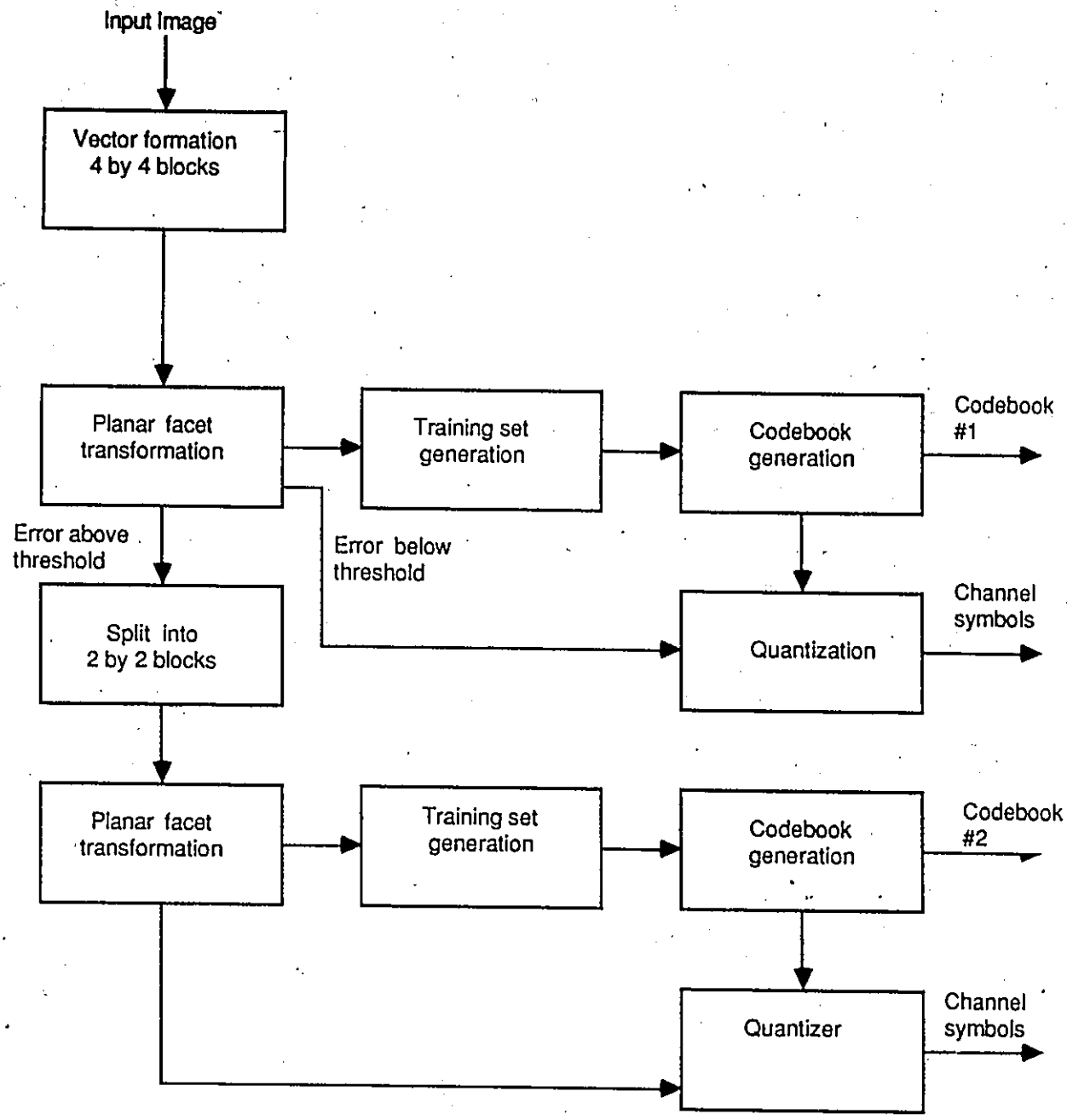


Figure 5.1a Block diagram of planar facet vector quantization with partitioned codebook

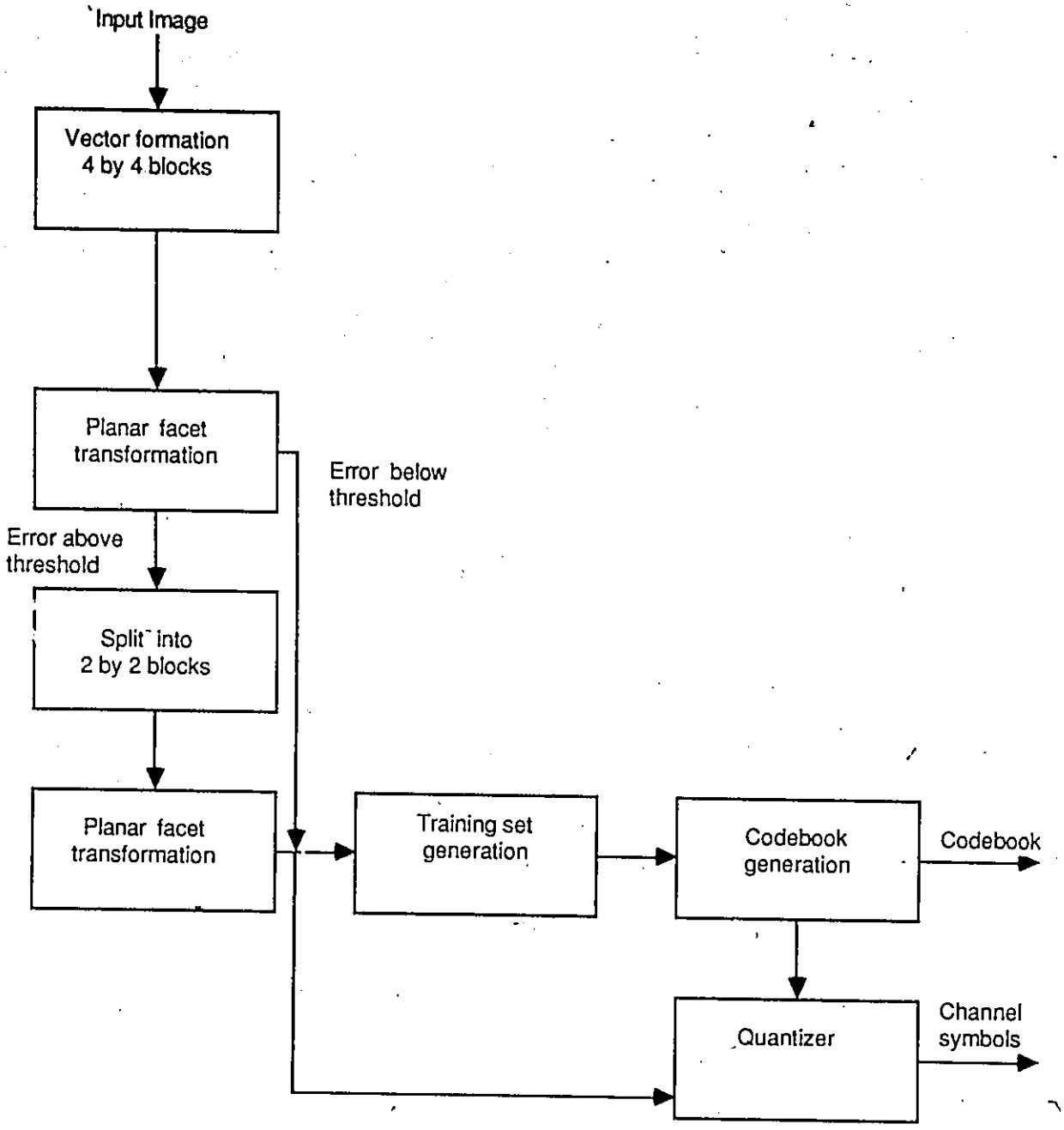


Figure 5.1b Block diagram of planar facet vector quantization with a single codebook.

5.1 Results and Discussion

5.1.1 Spatial Block Size for PFVQ

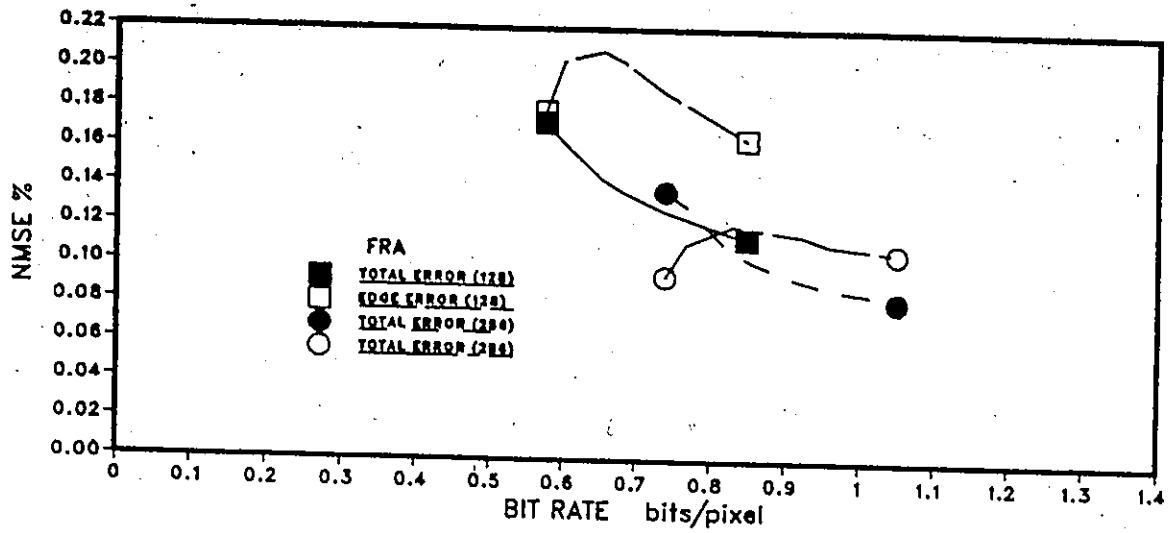
The first set of experiments are performed to determine the percentage of the image that requires 2 by 2 blocks for good visual quality. In the experiments in chapter 4, a definite elbow was observed in the rate distortion curve as the percentage of 2 by 2 blocks increased, leading to the conclusion that certain regions in the image derive little benefit form being coded with smaller blocks. This same property is tested now using vector quantized planar facet representation. The results of this experiment are given in figure 5.2a and the reconstructed images are given in figure 5.2b.

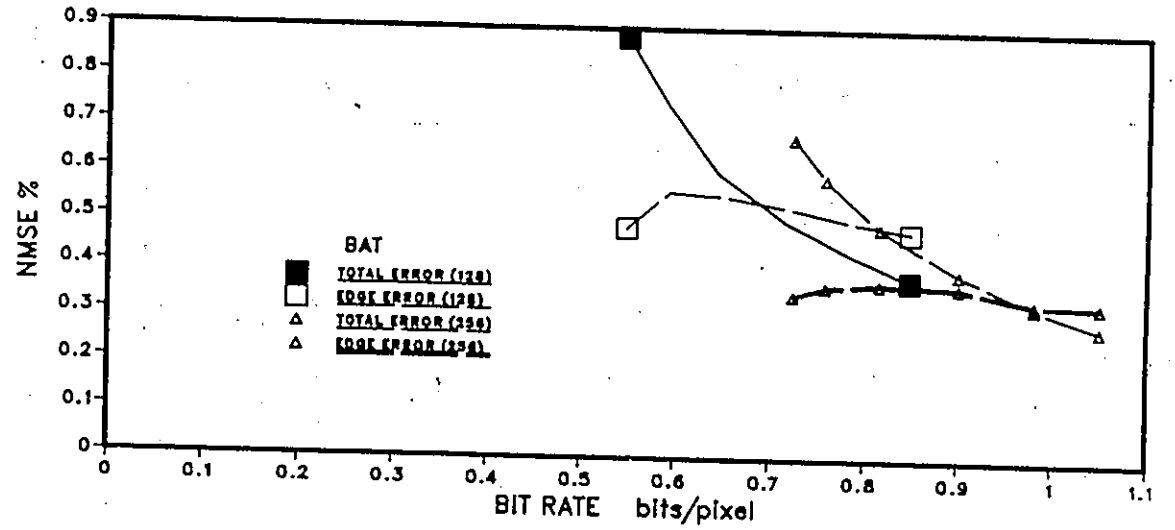
Discussion

It is clear that edge areas are not well represented when represented by large block sizes, while non-edge regions are well represented. This problem is also present to a certain extent in scalar quantization but is accentuated for two reasons in vector quantization. First of all, for each block there is a best fitting plane (best in a NMSE sense) that may introduce some distortion, depending on the grey level values of the block. Vector quantization will represent the best plane by some approximation which will approach but cannot do better than the best fitting plane. Thus the lower bound of distortion for PFVQ is that which is introduced by the best fitting plane, and cannot do better no matter what the codebook size. Secondly, vector quantizers tend to cluster the low probability vectors (the edges) in with the higher probability vectors leading to poor edge representation. As more of the image uses 2 by 2 blocks

the error is more uniformly spread though the image and leads to edge areas being well represented.

The facet representation using fixed polynomial degree and variable block size requires an error threshold to be chosen for the decision of dividing the block further of not. The threshold of error chosen is a NMSE /compression trade off and should be chosen consistent with the size of the codebook. Vector quantization will approximate the best fitting plane by the closest plane represented in the codebook, however the best fitting plane may still introduce error. Using a large codebook for with a high threshold of error may result in only a small reduction of the overall average NMSE and little improvement of subjective quality. This diminishing gains characteristic can be seen in figure 5.2 as the reduction of error is only marginal as the bitrate increases. The optimum combination of codebook size and error threshold must be determined from the final subjective quality of the image.





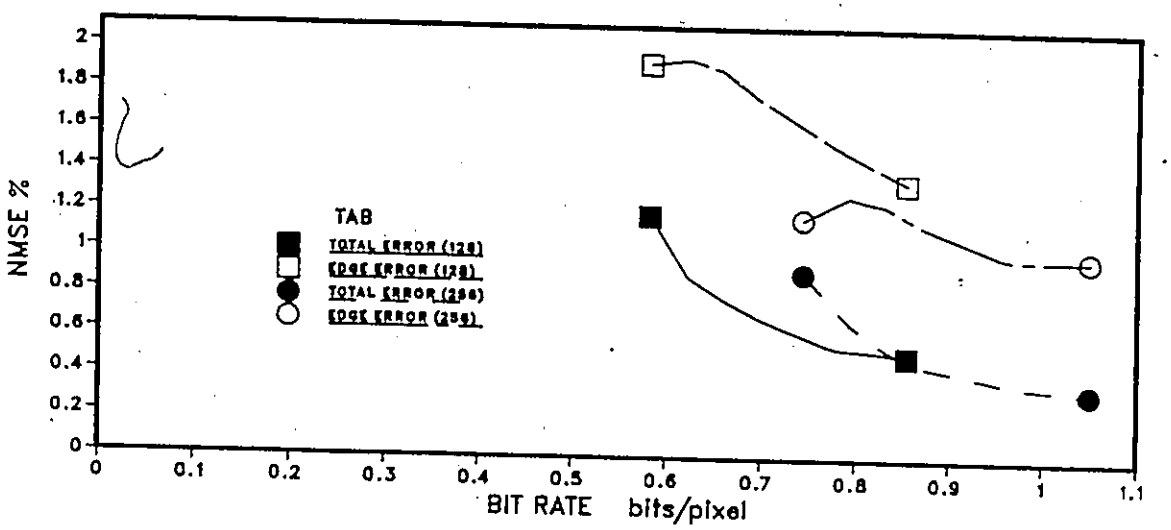


FIGURE 5.2. NMSE VS BIT RATE for vector quantized planar facet. The number in the brackets in the legend indicates the codebook size. These curves are examples of two codebook sizes, all codebook sizes showed the same concave curve for edge error

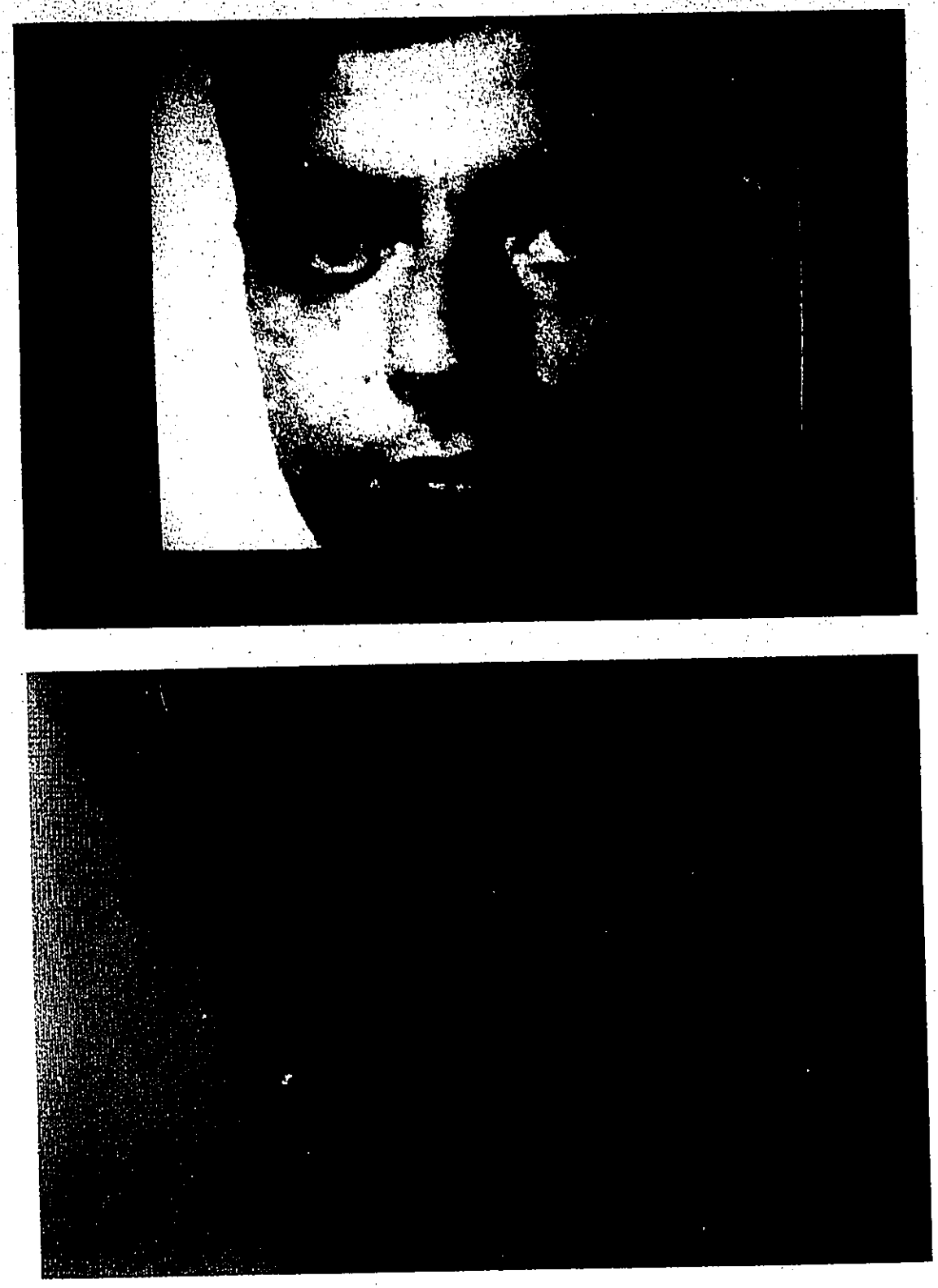


Figure 5.2b Photographs of reconstructed images keeping a fixed 25% of the image as 2 by 2 blocks. The three images shown here are; Top; 64 codewords used in codebook. Middle; 256 codewords used in codebook. Bottom; 512 codewords used in codebook.

figure 5.2b (continued)

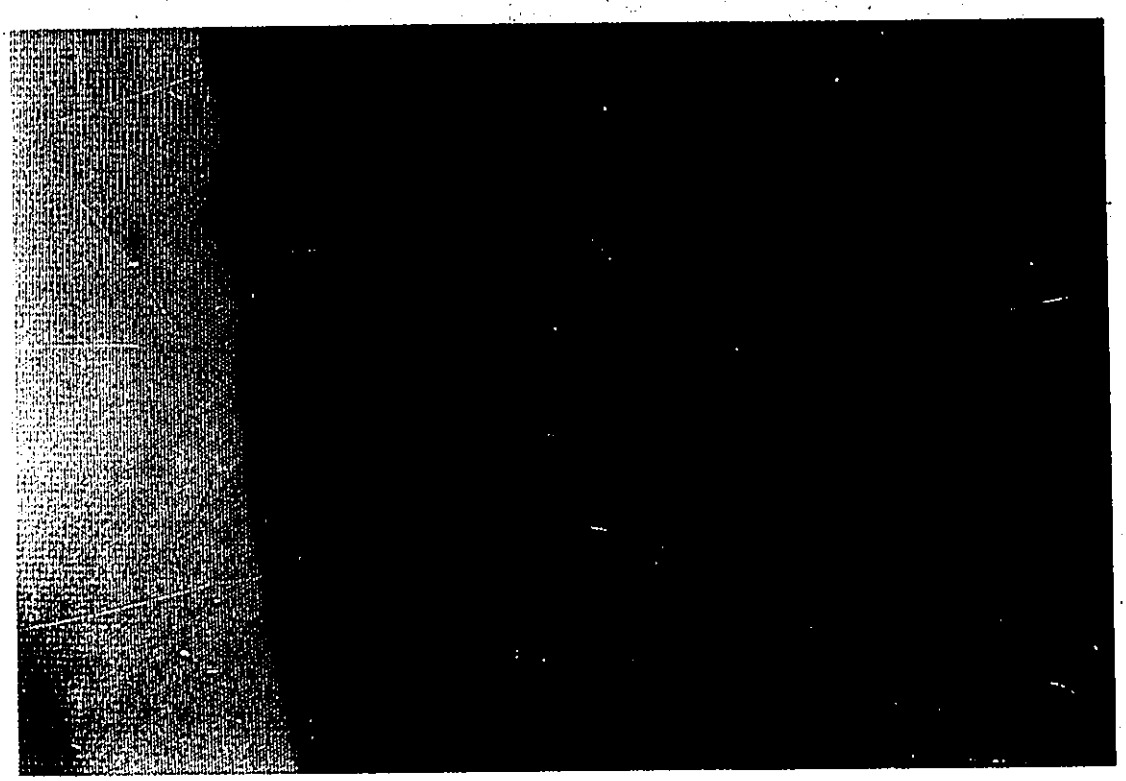


figure 5.2b (continued)

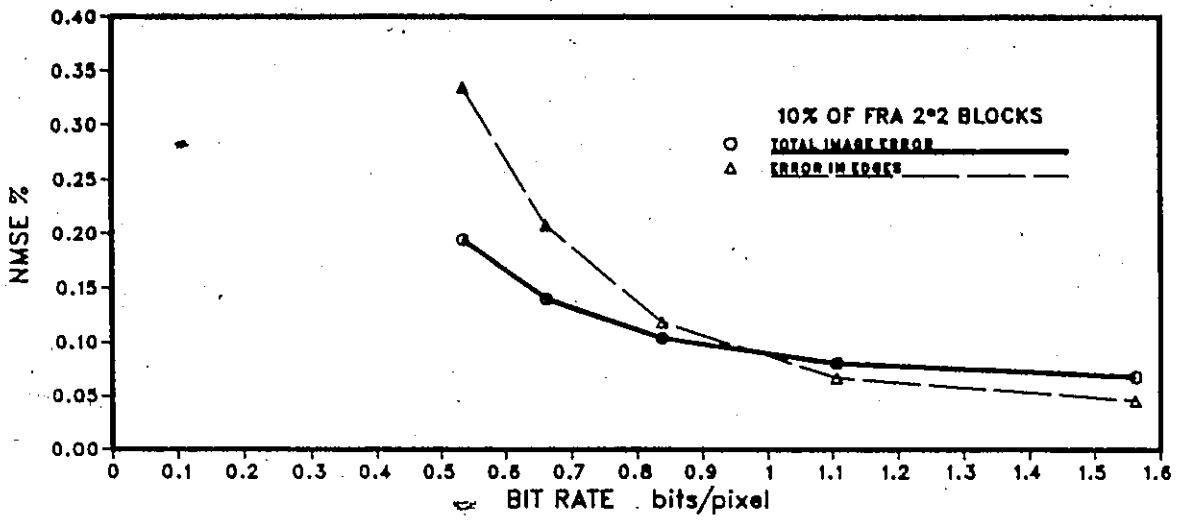
5.1.2 Codebook Size for PFVQ

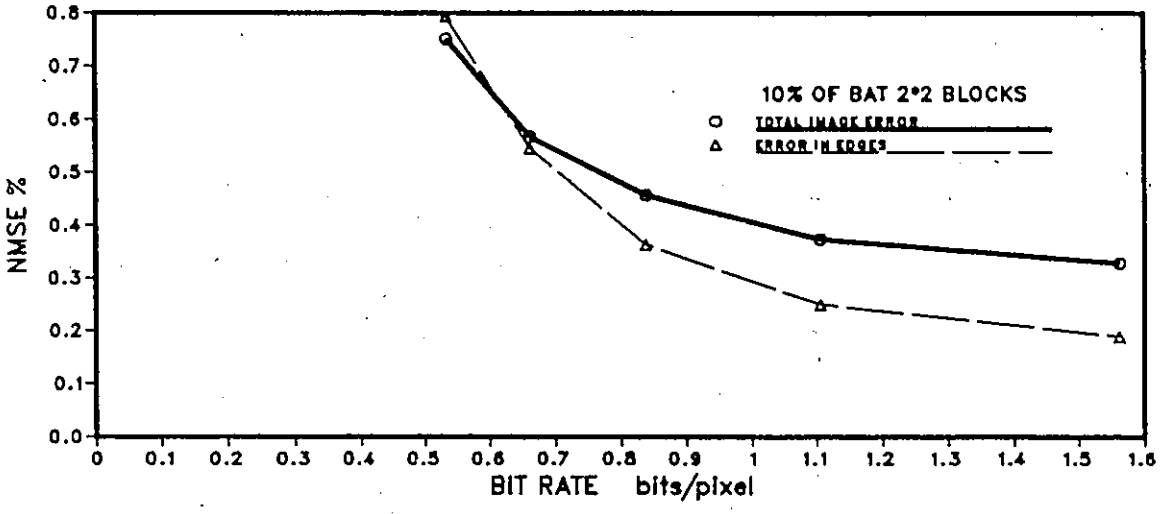
The second set of experiments uses a constant percentage of 2 by 2 blocks while the codebook size is varied. For the three test images, two thresholds are demonstrated. One leading to 10 percent of the image being divided into 2 by 2 blocks and one leading to 25 percent of the image being split to 2 by 2 blocks. Because of the wide range of statistics within the test images, these percentages were reached using different thresholds. To get 10 percent of the image, the thresholds in NMSE percent are: .23 for FRA, 1.7 for BAT and 2.8 for TAB. For 25 percent the thresholds were: .1 for FRA, .76 for BAT and .55 for TAB. That is to say, that blocks with error above these thresholds were considered edges.

Discussion

As with the scalar quantization case, it can be seen that certain blocks derive no benefit from a split to 2 by 2 blocks. This is reflected in the decreasing reduction in the NMSE as the bit rate increases. For each image an explanation is warranted in terms of the individual image statistics. The image FRA saw an increase in the overall NMSE of the image and an increase in the error in the edges when going form 10 to 25 percent of the image using 2 by 2 blocks. This is because 25 percent of the image as 2 by 2 blocks was unnecessary for good coding. Many blocks were coded as 2 by 2 blocks that could have been sufficiently well coded as 4 by 4. The additional overhead at 2 by 2 increased the overall R(d) curve. For the image BAT, the overall error decreased while the error in the edges remained about the same when the amount of 2 by 2 blocks increased from 10 to 25 percent. This is because for this image, 10 percent 2 by 2 was insufficient and many of the edges were coded as 4 by 4, increasing the overall NMSE. At 25 percent most of the edges were divided into 2

by 2 blocks and resulted in a decrease in NMSE. Finally, the coded TAB image with 25 percent 2 by 2 had increased error in both the edges and overall error. This is because, while the image contains many edges, only a few are sharp edges, most of the pixels lie in areas of constant slope and are, therefore, generally well coded using 4 by 4 blocks.





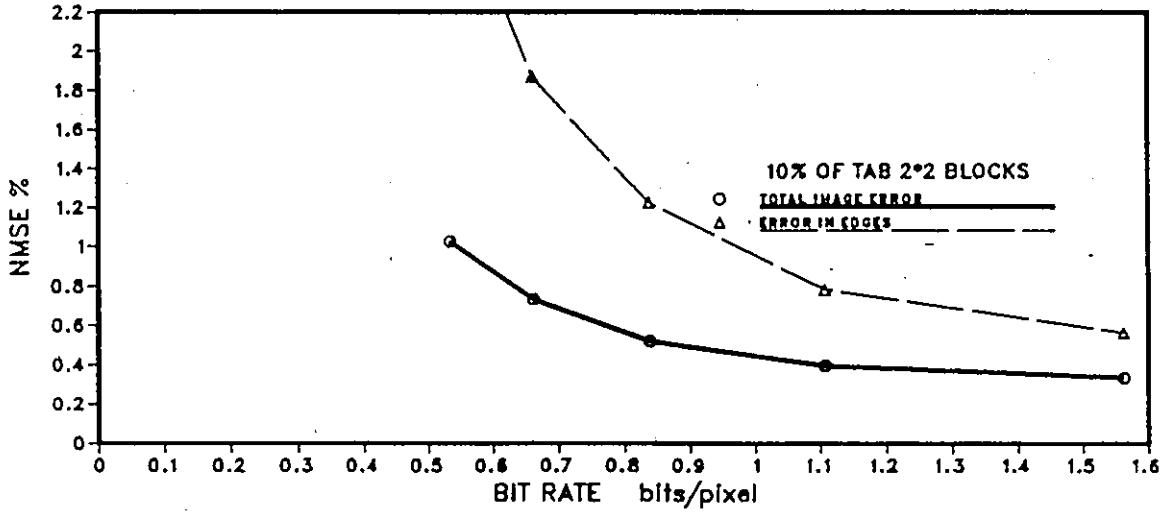


FIGURE 5.3. NMSE VS BIT RATE for vector quantized planar facet.The thick curves show the total error, the lighter curves shows the error in just the edges. The 2 by 2 block content is constant at 10 percent

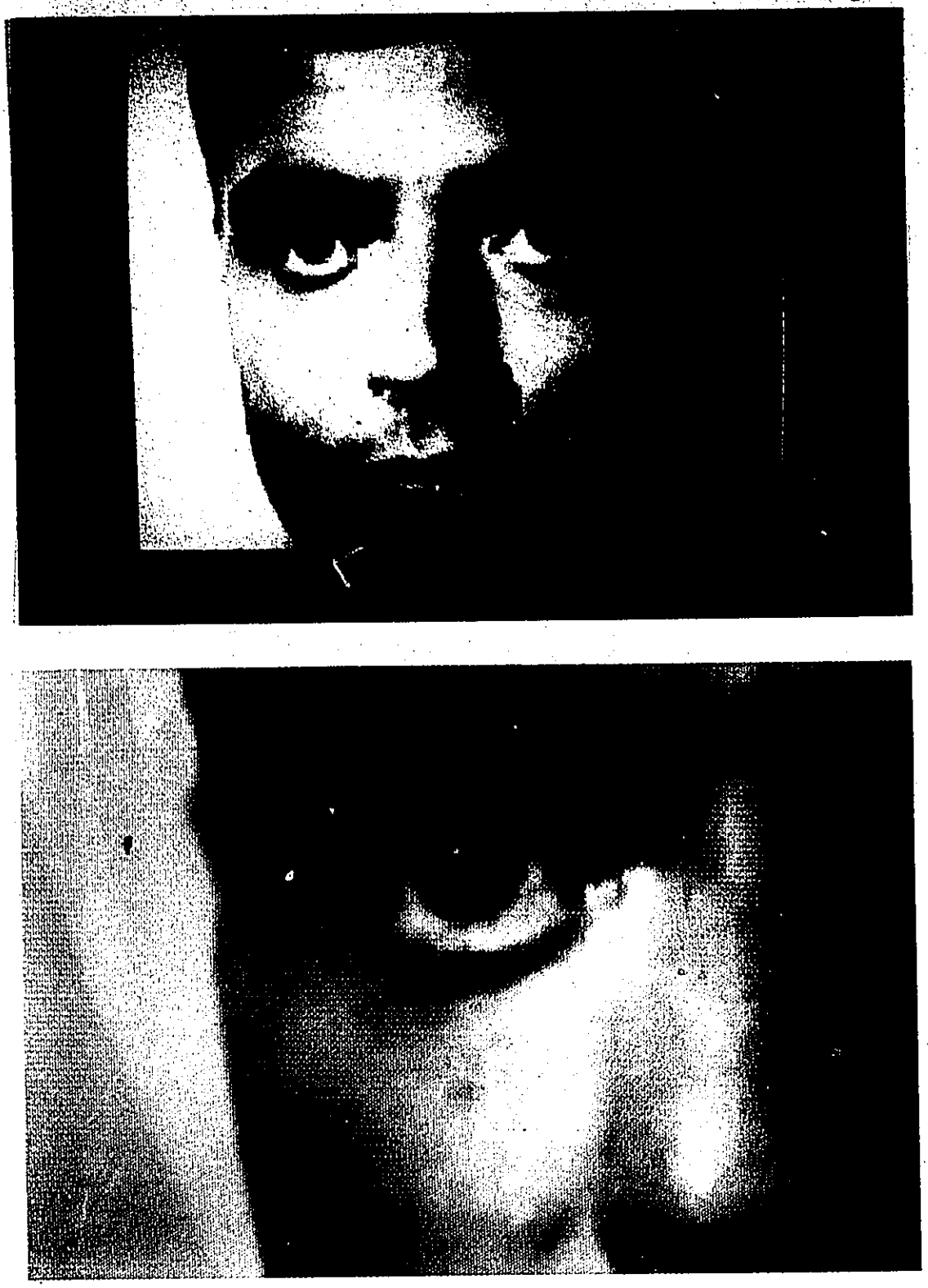


Figure 5.4 Photographs of reconstructed images using 10% 2 by 2 blocks and 128 codewords for each codebook.

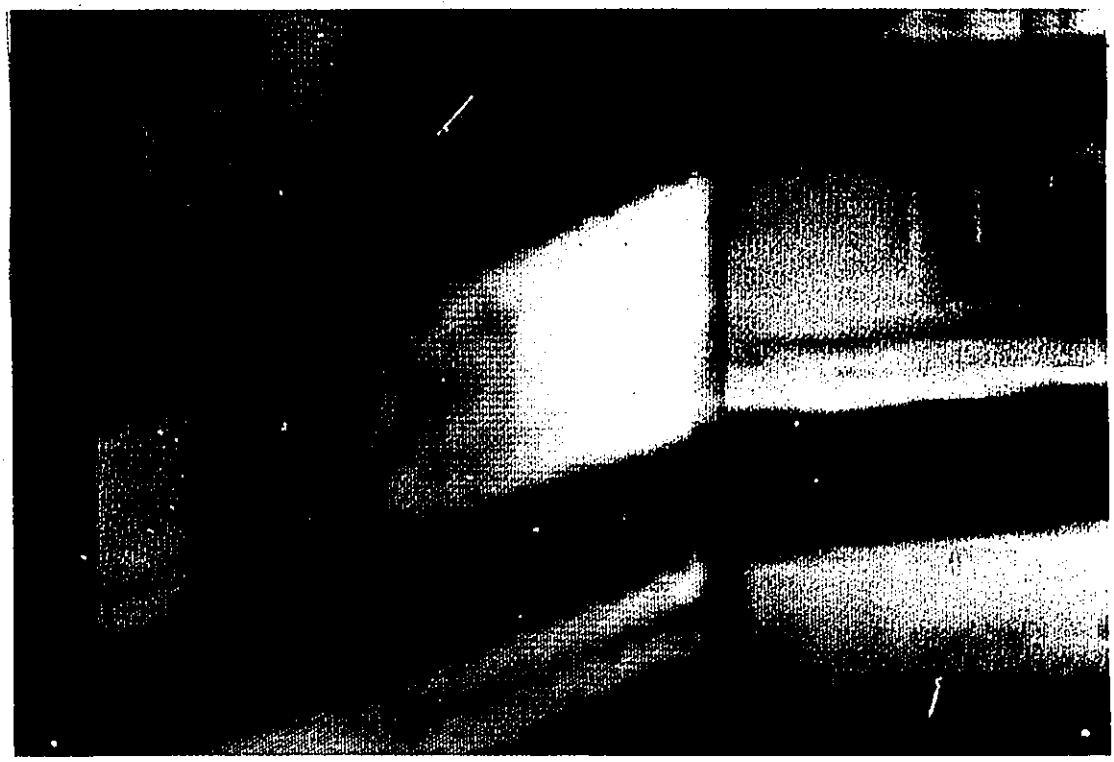


figure 5.4 (continued)

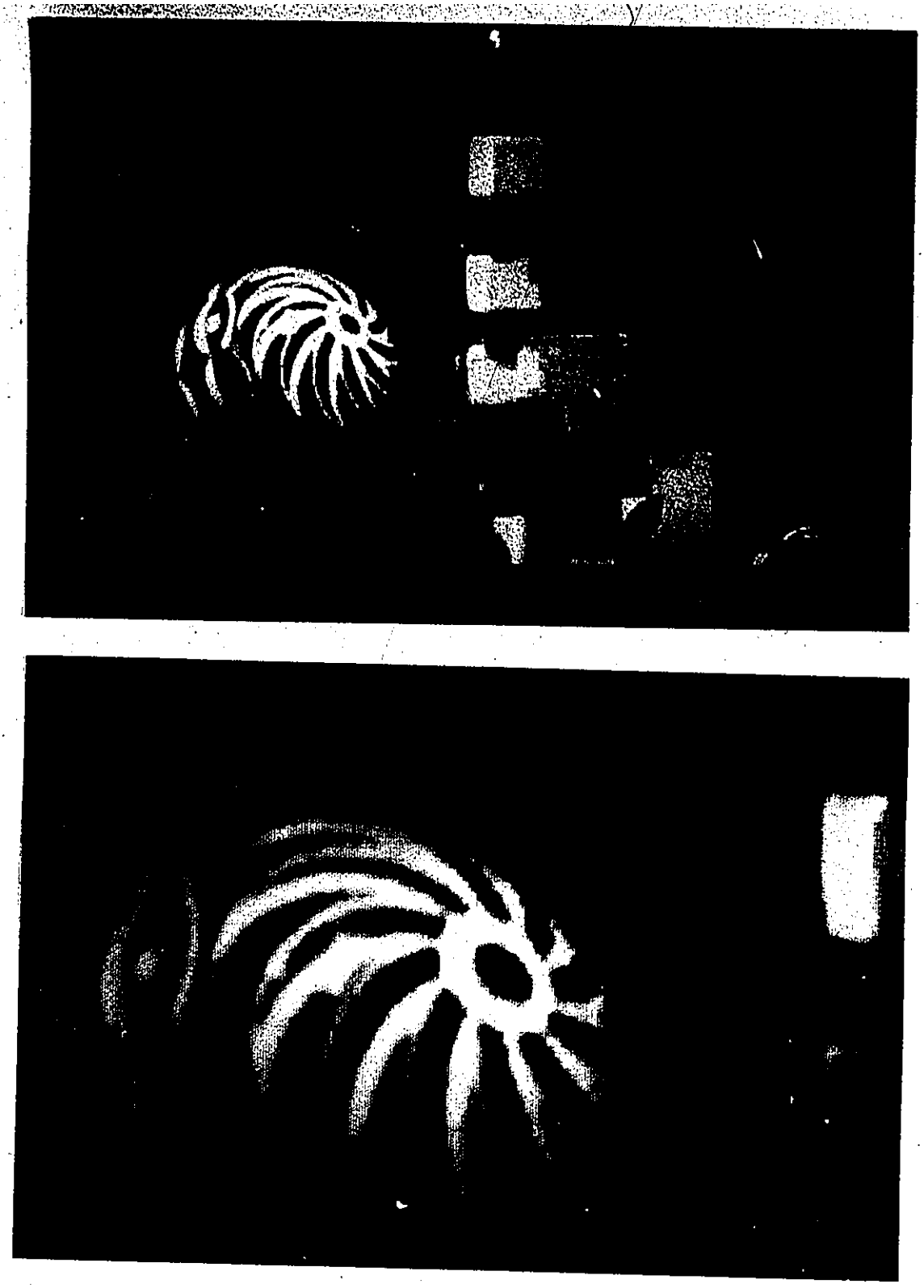
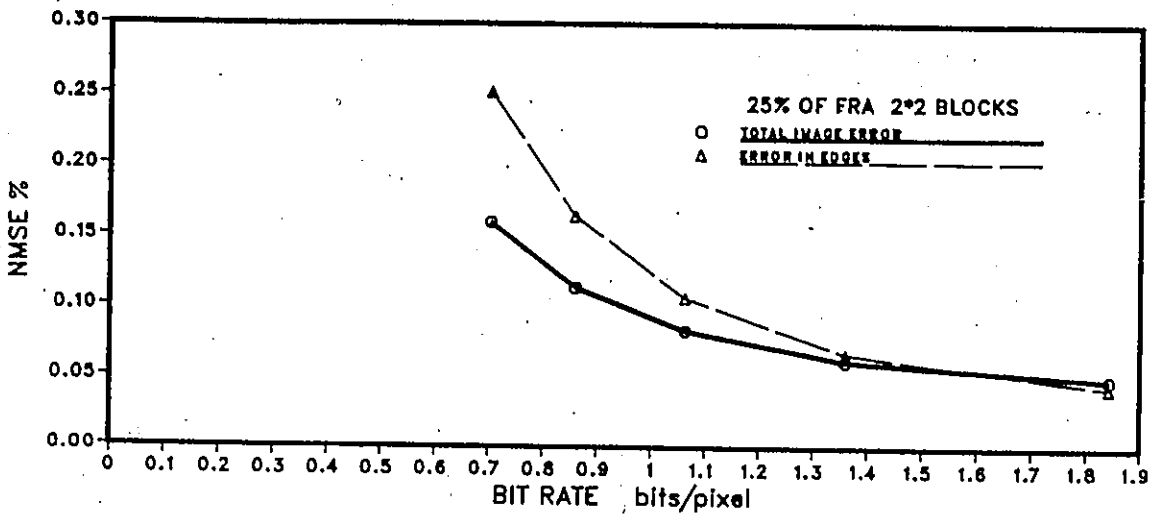
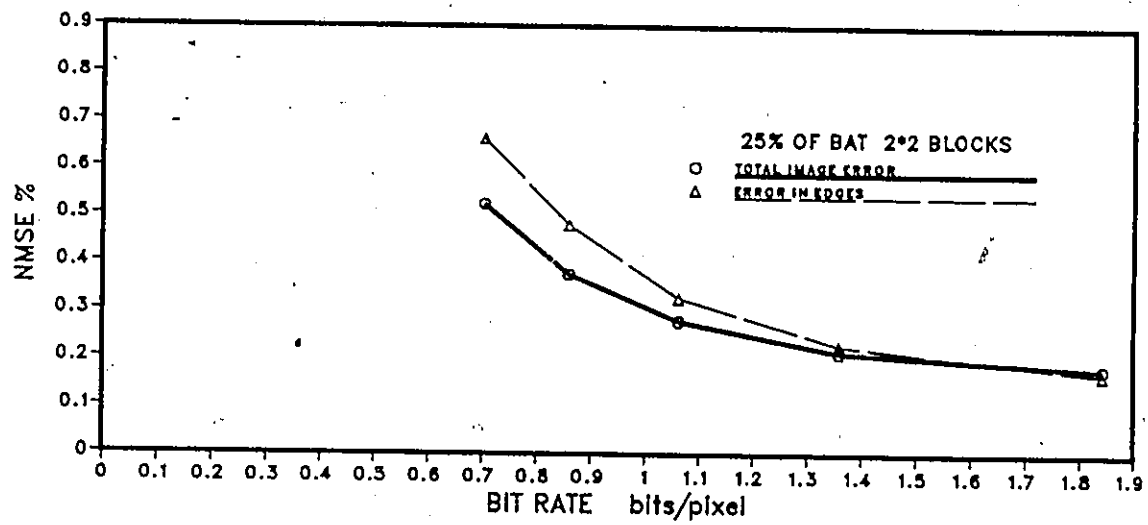


figure 5.4 (continued)





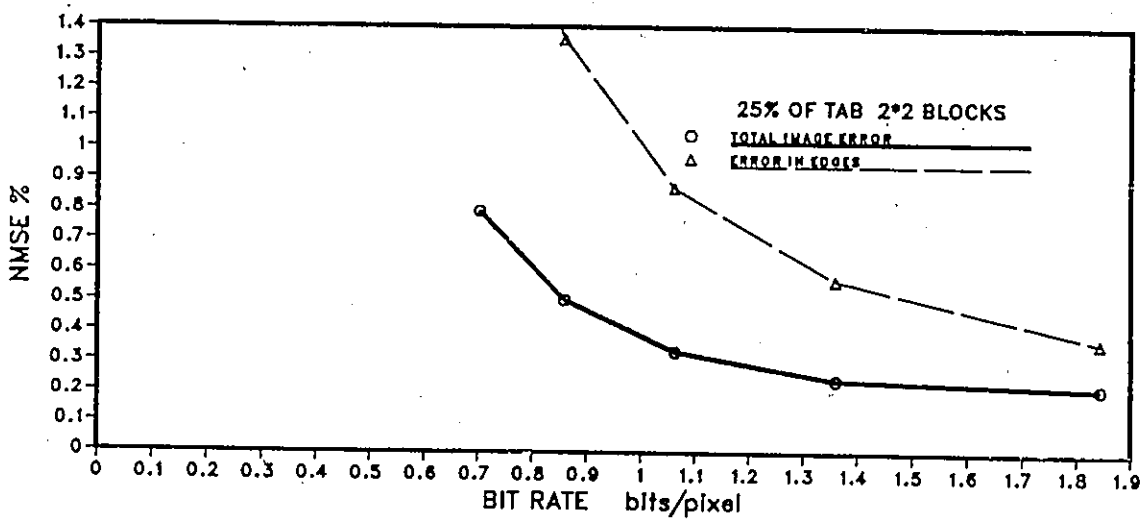


FIGURE 5.5. NMSE VS BIT RATE for vector quantized planar facet. The thick curves show the total error, the lighter curves shows just the error in the edge blocks. The percentage of the image that uses 2 by 2 blocks is held at 25%

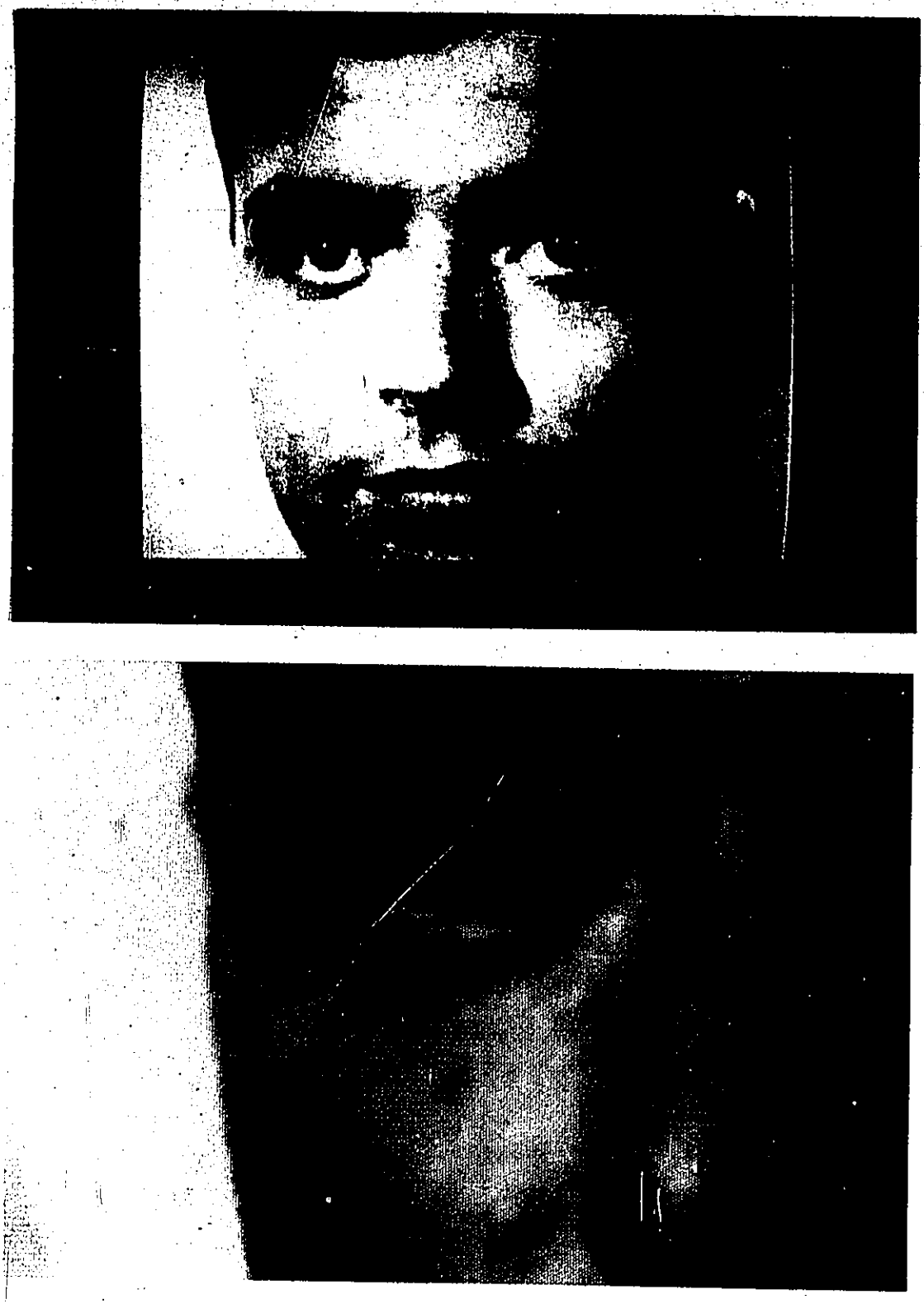


Figure 5.6 Photographs of reconstructed images using 25% 2 by 2 blocks and 128 codewords for each codebook.

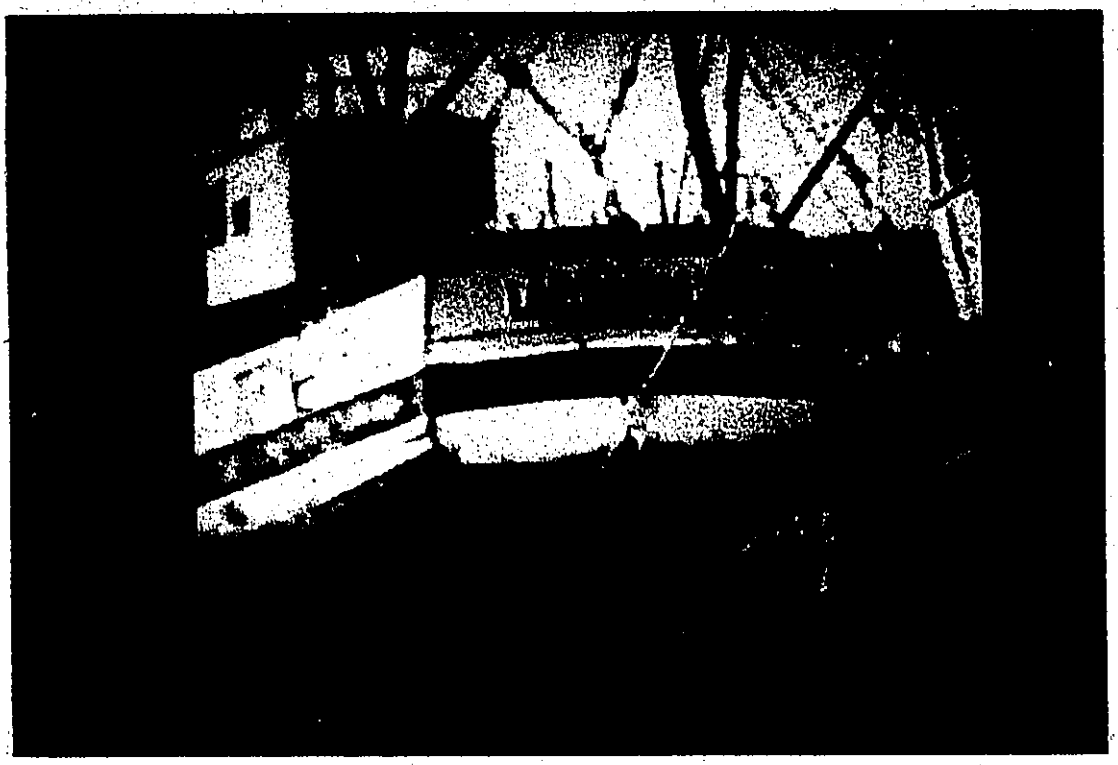


figure 5.6 (continued)

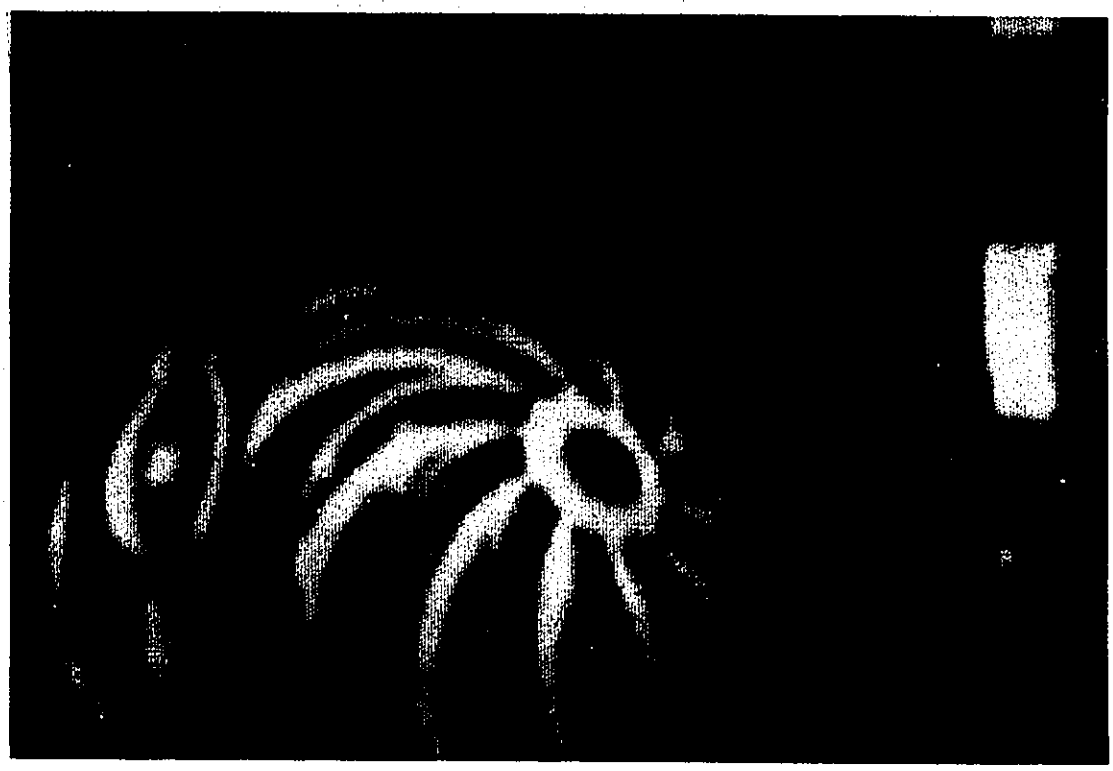


figure 5.6 (continued)

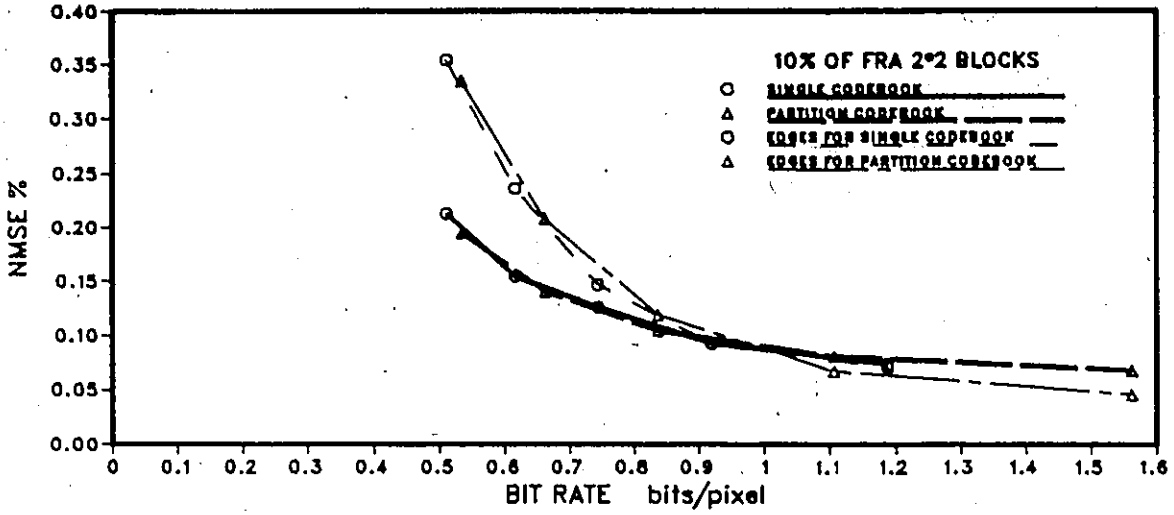
5.1.3 Partition vs Single Codebook for PFVQ

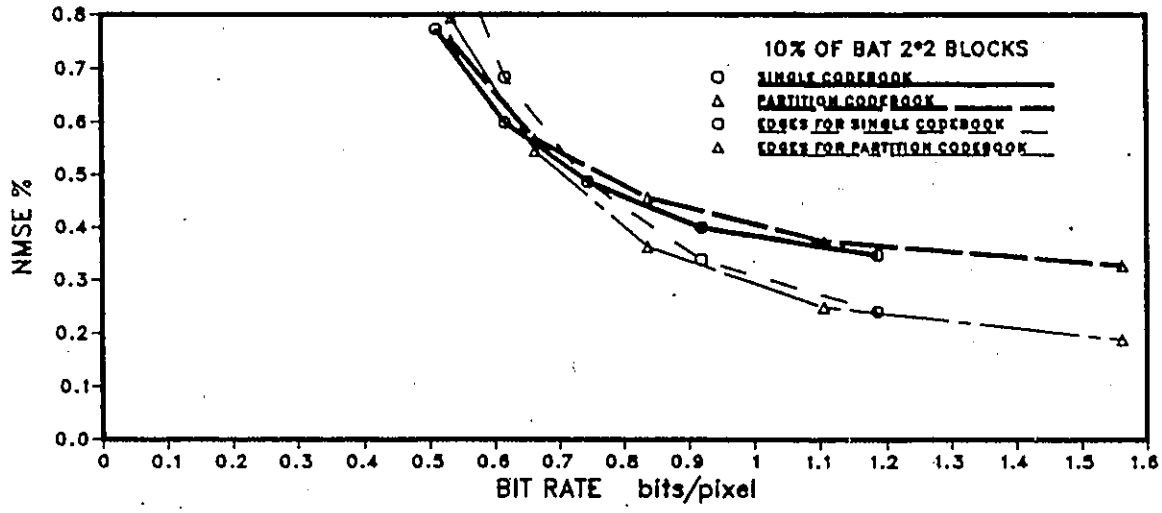
The final set of experiments is used to determine whether a partitioned codebook or a single set is necessary for quantization. If the transform maps both the large and the small blocks onto vectors of similar statistics then a single codebook should give similar results to a partitioned codebook. An example of when both 2 by 2 and 4 by 4 blocks are mapped onto the same slope is shown in figure 5.12. The difference in grey level for the 4 by 4 block could be twice as much as with a 2 by 2 block and would still be mapped onto the same slope. Figure 5.7 and 5.8 show a single and partition codebooks used for training sequence generation. The experiments were done using 10 percent 2 by 2 and 25 percent 2 by 2.

Discussion

From these two graphs it can be seen that there is no advantage in terms of the NMSE in using a partitioned codebook for the FRA or the BAT image, however for low bit rates the partition codebook has slightly less NMSE for the case of TAB in both the 10 percent and the 25 percent case. As is well known NMSE is not well correlated to subjective quality. A lower overall NMSE may not result in a better subjective image. Examining the error in the edge curves in figure 5.7 and 5.8, it is seen that for the 10 percent 2 by 2 case no advantage in gained by using a partitioned codebook. There is a marked decrease in the error in the edges in the TAB image, and a marginal decrease in the error in the edges for the FRA image for a single codebook. An examination of the images in figure 5.9 verifies that no increase in subjective quality is gained by using a partitioned codebook. There is a slight anomaly in the TAB image R(D), as the overall error increases but the error in the edges decreases when using a single codebook. This is due to the nature of edges in

this image. TAB contains many sloped regions, areas where the change in the pixel values are intermediate in magnitude. These vectors strongly influence the cluster centers when vector quantizing leaving an overall increase in error but a decrease in edge error. Error in the edges does increase in the FRA as observed in TAB, when 25 percent of the image is 2 by 2 blocks. At that point some of the uniform sloped areas start to influence the cluster centers. The uniformly sloped areas are found on the left and the right of the subjects head. It should be noted that there is a slight increase in error when ever a single codebook is used. This would be expected as the partitioned codebook would contain at least similar codewords to what was in the single codebook plus additional codewords for a better selection of codewords and thus a lower error. However, any advantage in NMSE by using a partitioned codebook is offset by the additional overhead needed for the extra codebook. This shows that the two vector sets from the two block sizes similar in statistics.





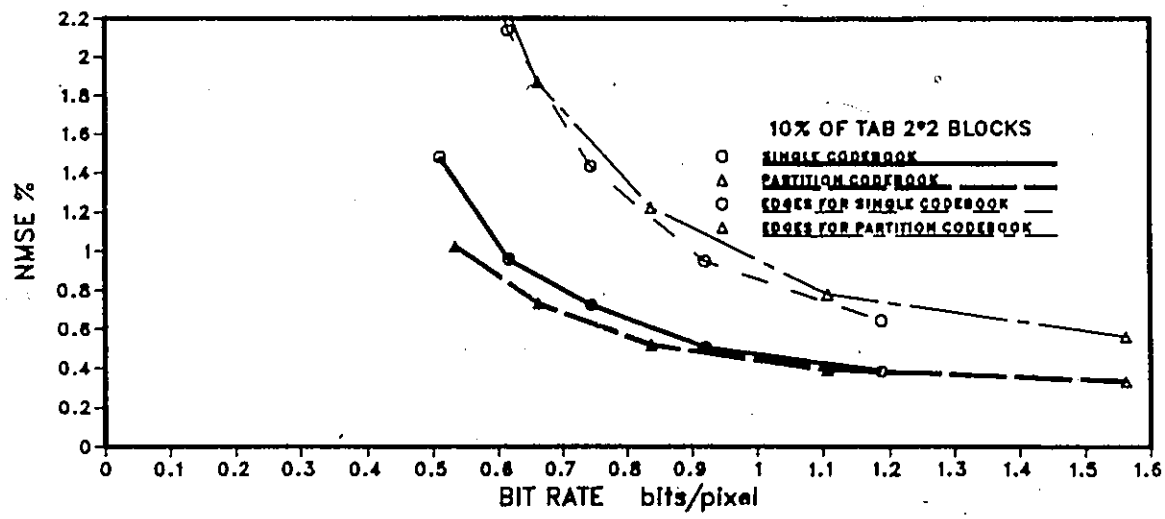
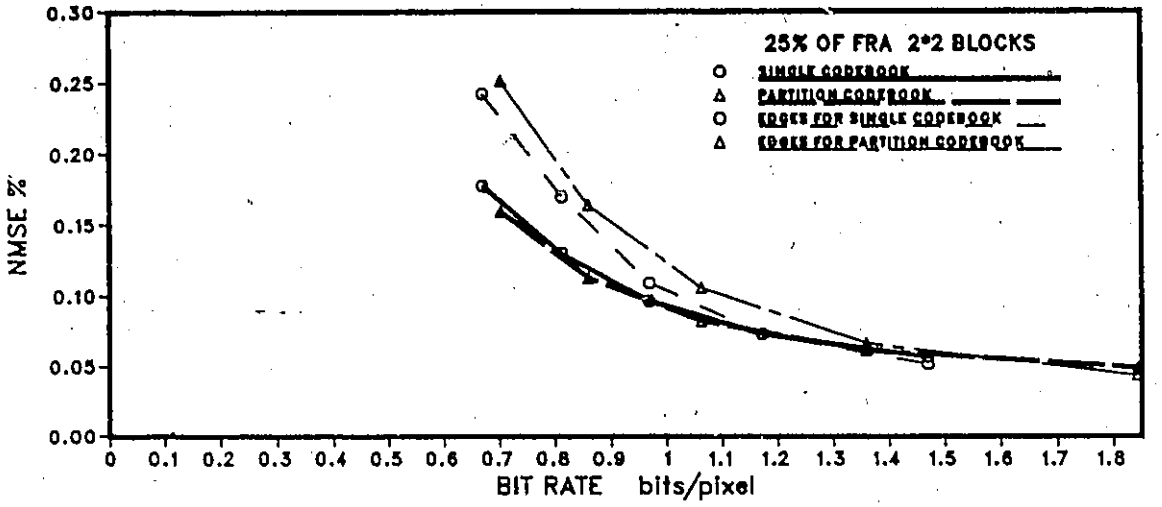
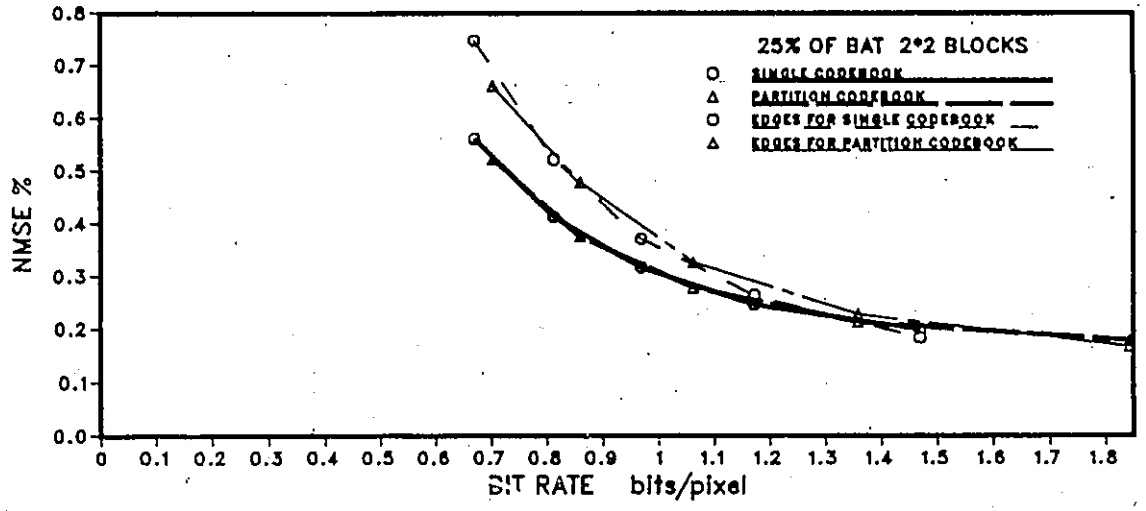


FIGURE 5.7. NMSE VS BIT RATE for vector quantized planar facet. The thick curves show the total error, the lighter curves shows the error in just the edges. Note how the error in the edges does not increase whena single codebook is used





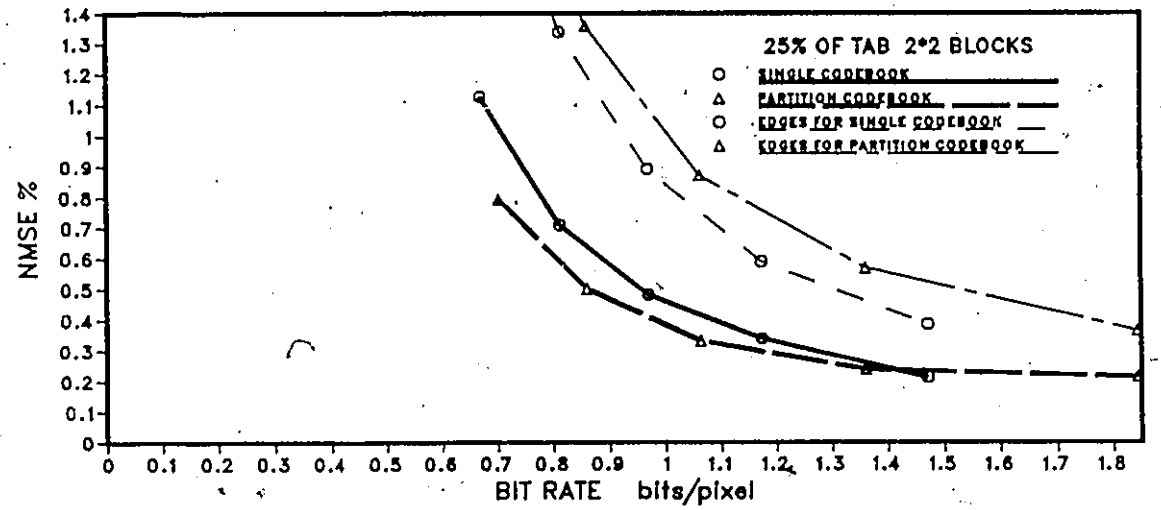


FIGURE 5.8. NMSE VS BIT RATE for vector quantized planar facet. The thick curves show the total error, the lighter curves shows just the error in the edge blocks. Note that the error in the edges does not increase when a single codebook is used.

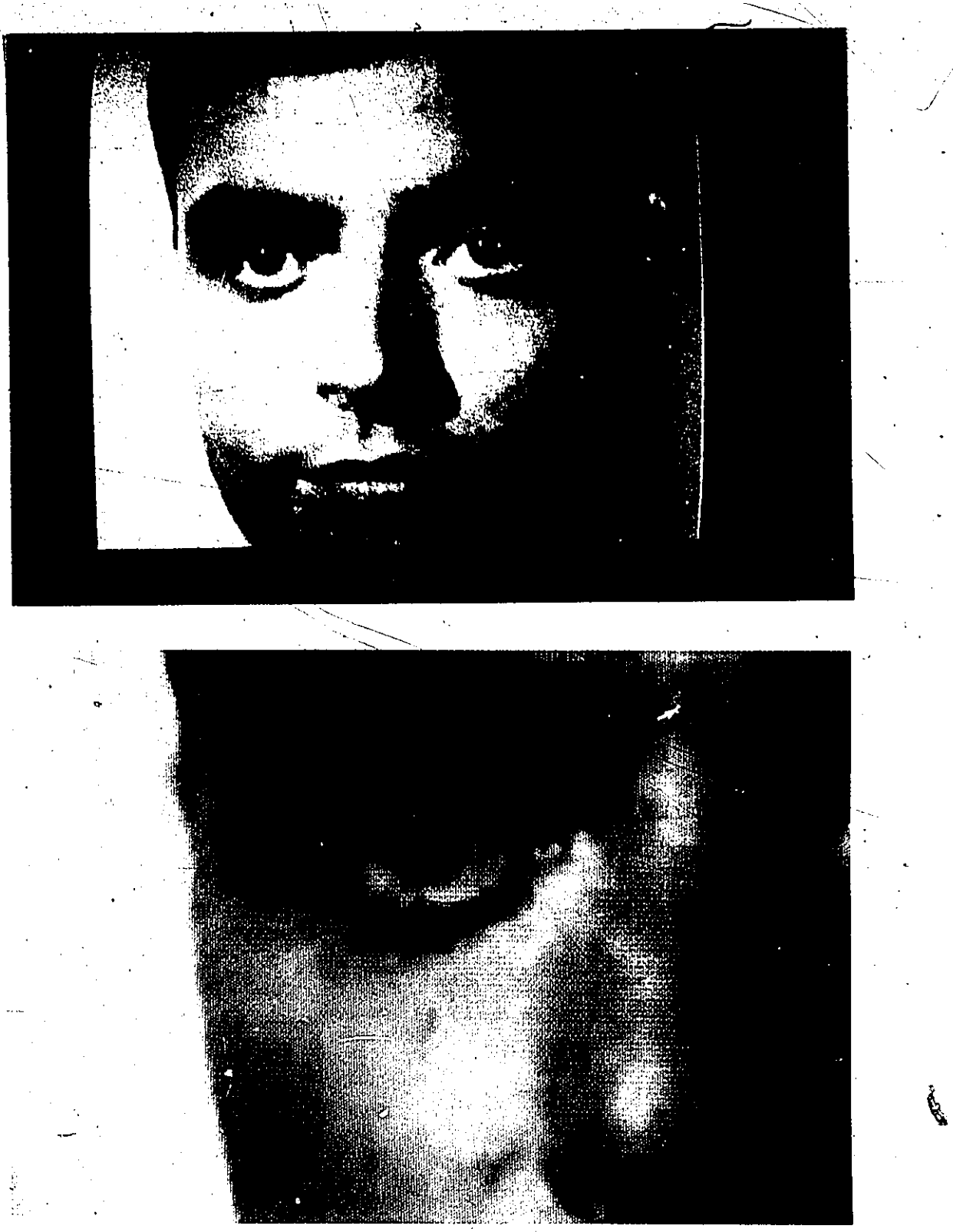
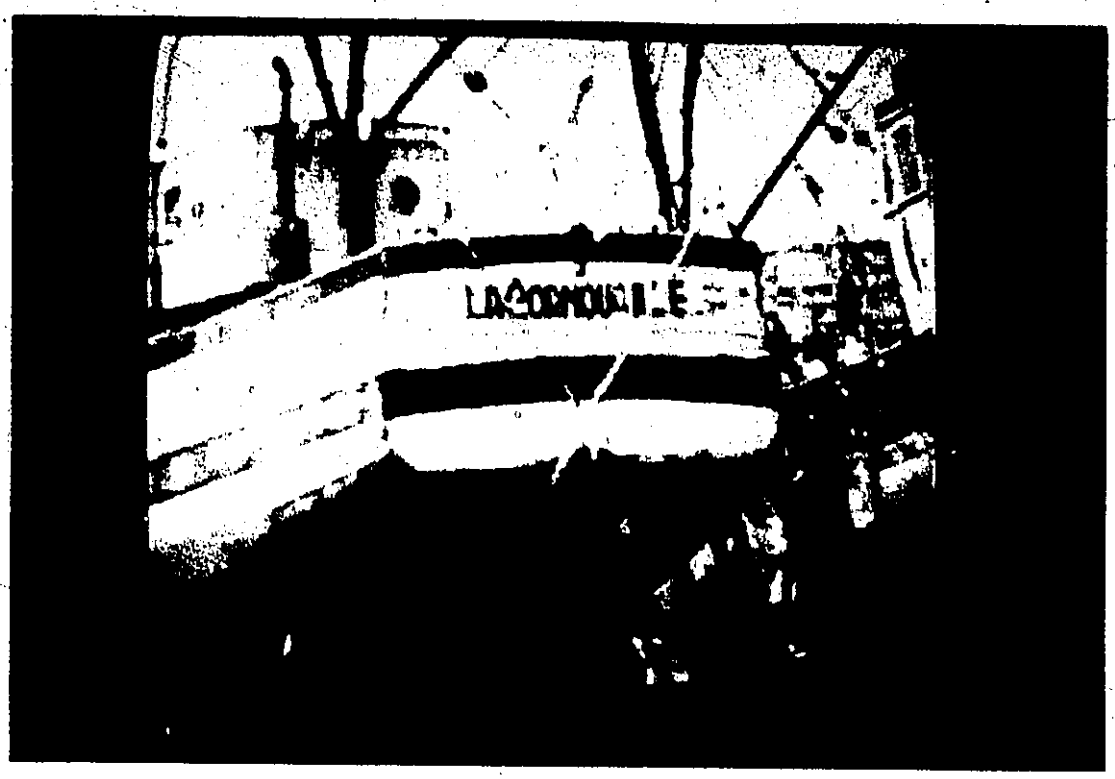


Figure 5.9 Photographs of reconstructed images using VQ with only a single codebook for both the 2 by 2 blocks and the 4 by 4 blocks. These images used 25% 2 by 2 blocks. Compare these images to the images of figure 5.6, which use two codebooks.



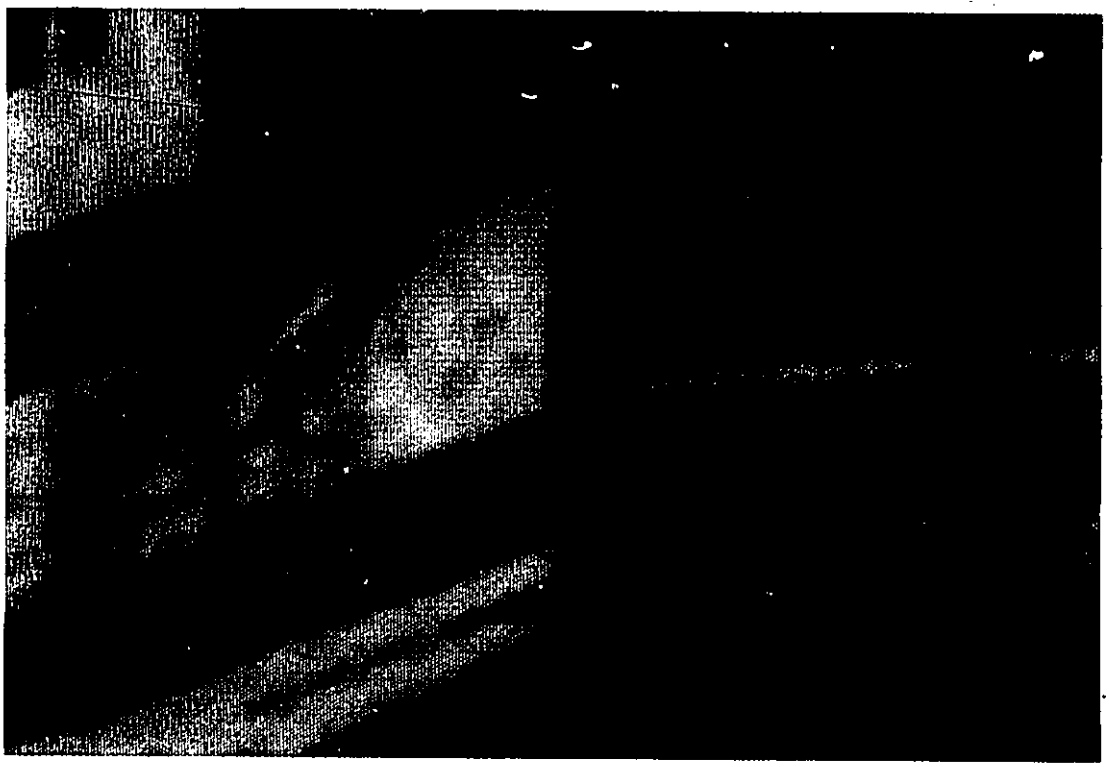
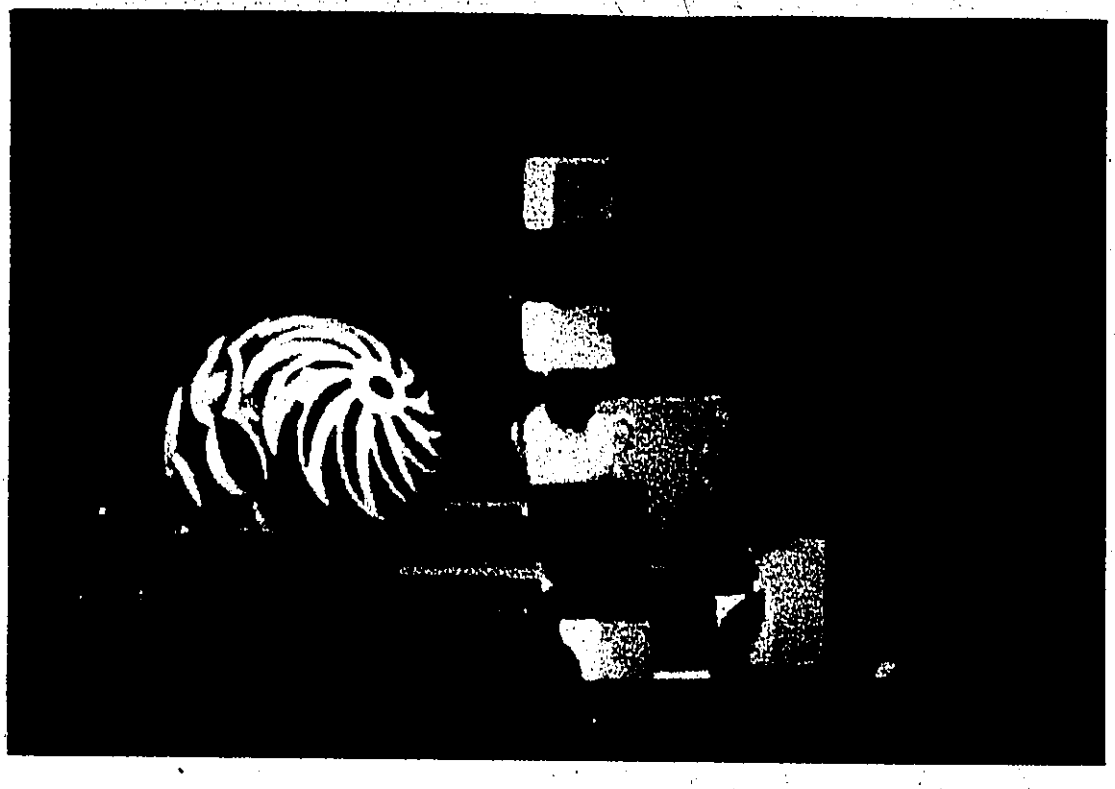


figure 5.9 (continued)



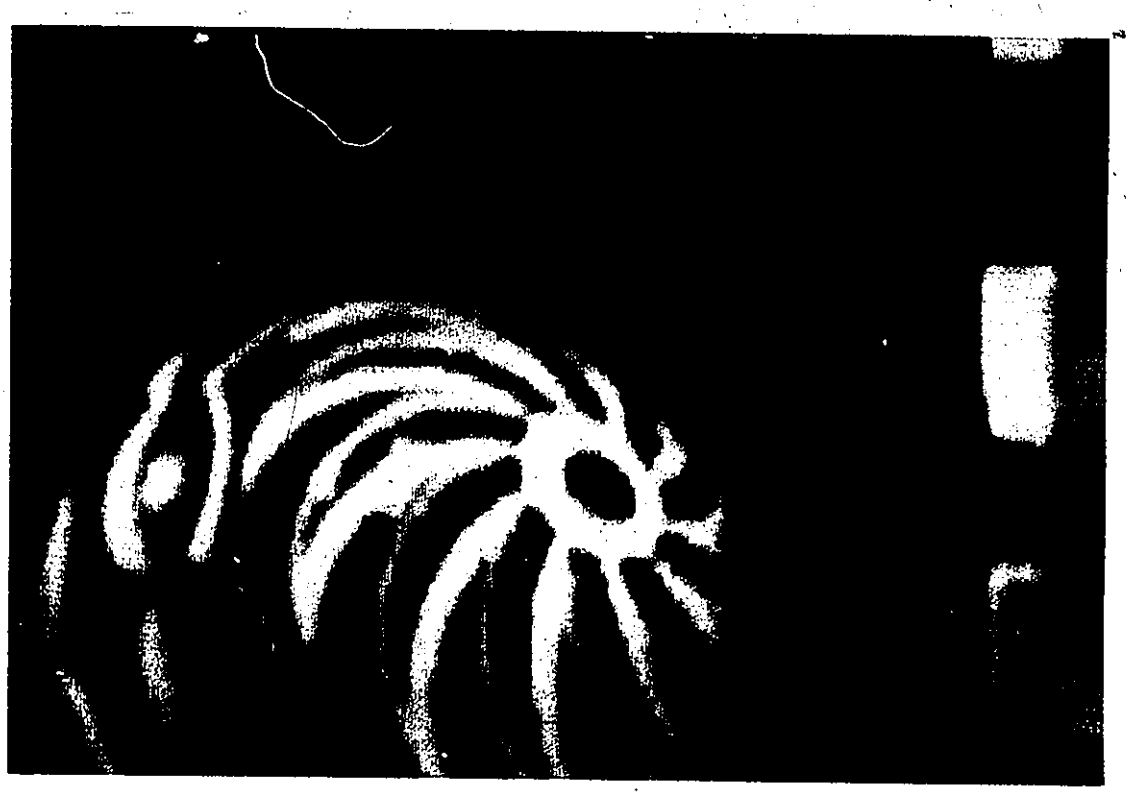


figure 5.9 (continued)

5.1.4 Additional Discussions

Some additional comments are now given on PFVQ. While Shannon's theory states that the rate distortion can be reduced to a minimum by extending the block length, there are some practical limitations to the size of block that can be used. A study by Baker[54] showed that for non-adaptive memory-less VQ, the size of vector was limited to about 20 because of constraints on memory and computation.

Adaptive VQ results in a lower NMSE and better subjective quality[53] when compared with basic VQ. However, a study by Boucher [53] showed that adaptive VQ coding schemes are even more restrictive in the maximum size of vector. This is because the codebook takes too much of the bit rate to take full advantage of the larger codewords. Transmitting the codebook reduces the effective rate (the portion of the rate devoted to transmitting the labels) to such an extent that the optimum size of the vector for adaptive VQ was found[53] to be about 9. The upper limit seems to be a codeword of 16, above which the codebook takes too much of the transmitted information to allow efficient coding. This limitation is clearly seen from table 5.13 where the bit rate is constant at 1.5 bits/pixel. As the size of the block length gets larger more and more of the transmitted information is overhead. With a block size of 20, 72 percent of the transmitted information is required to transmit the codebook.

The codebook in adaptive VQ contains highly redundant data and does not exploit inter-block correlations. A codeword that represents a slowly varying region of the image may consist of highly correlated data. As the vectors get larger, more redundant data is contained in the channel code. The PFVQ works to reduce the redundant data in the codebook.

By first representing the image by its facets, large spatial blocks can be used without the accompanying redundancy of larger vector codebooks. By reducing the percentage of the bit rate devoted to transmit the codebook, more of the bit rate can be used to transmit the labels. Using the planar facet representation with a block size of 16 allows 2048 representative codewords in the codebook compared with 512 for adaptive VQ with the same block size.

In the review of vector quantization in chapter 3, vector quantization was divided into four processes, vector formation, training sequence generation, codebook generation, and quantization. The Planar facet representation is a technique to improve image coding quality by enhancing the vector formation, and training sequence generation steps.

5.2 Comparison with Other Methods

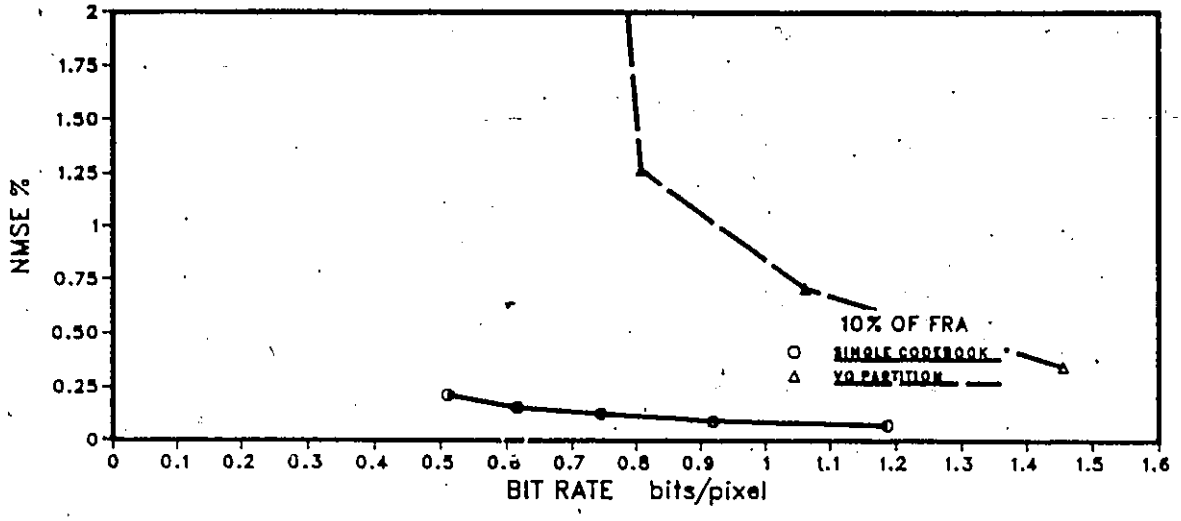
Vector quantization applied to the facet representation (PFVQ) is now compared to three other techniques. The first comparison is made with a VQ technique proposed by Gersho and Ramamurthi[16] that splits the codebook into "edge" and "texture" vectors for training sequence generation and quantization. The technique has been modified to make it adaptive to assure a fair comparison. The second comparison is made with cosine block transform[61], and finally, VQ applied to the facet representation. is compared to a quadtree technique proposed by Wang and Goldberg[10].

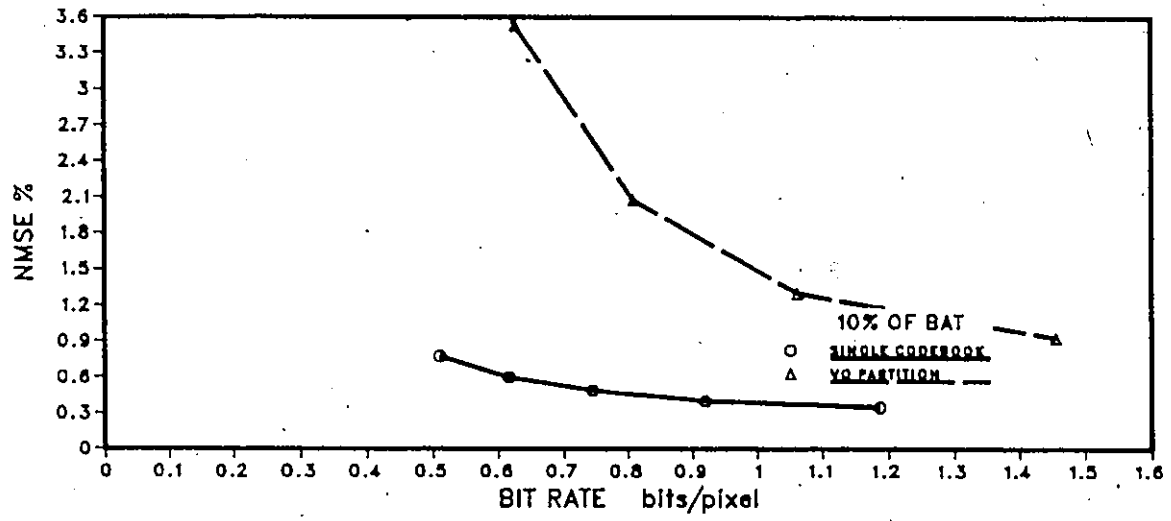
Gersho and Ramamurthi[16] proposed a method of overcoming the blocking effect of the coded image by a partition code. A simple classifier was used to distinguish between vectors of one of two classes, regarded as the 'edge' and 'texture' vectors, with separate codebooks for each. The separation of edge and texture is on the basis of the variance (or energy) of the vector. All vectors with variance below a threshold value are considered textures and above the threshold are considered as edges. In order to ensure a fair comparison of this partitioned codebook technique and PFVQ, certain parameters must be kept constant. When comparing coded images each method will devote the same percentage of the bit rate to the 'edges' and the 'shade' classes. For example, if an image is coded with 10 percent edges in PFVQ then 10 percent will also be used for the partition codebook method. Keeping these parameters constant assures that the comparison using NMSE relates well with subjective quality. It is well known that the initial codebook and the algorithm play a large role in obtaining good results. Thus the same algorithm, indeed the same Fortran program is used for each of these methods. Figure 5.10 shows the R(D) curves in the coded images using the PFVQ with a single codebook compared with VQ using a partitioned codebook

for blocks of size 3 by 3 and 4 by 4. The PFVQ results in less error than the VQ with partitioned spatial codebook. The largest advantage between these two methods occurs when the codebook is small at low bit rates. There are two reasons for this; first PFVQ using 3 coefficients could never result in less NMSE than that incurred in the original transformation. At higher bit rates the error approaches the lower limit of NMSE. If the codebook size continued to increase the upper curve would eventually cross over. Secondly, at low bit rates with large block sizes, the overhead required to transmit the codebook takes a larger percentage of the bit rate in the partition codebook method and PFVQ has its greatest advantage. The main reason for the improved performance is the reduction in redundancy in the codebook and the more efficient use of codeword for regions of slowly varying grey level values.

Block transform coding involved the two-dimensional transform of 16 by 16 pixel blocks. The transform frequency coefficients are Max-quantized [61] using a Gaussian distribution model. The coefficients are quantized using a different number of levels according to their variance, and according to the required bit rate [62]. The mean component is quantized to 8 bits. The error in the coded image is shown in figure 5.11[33]. PFVQ outperforms the cosine transform because, while this unitary transform results in coefficients are much more decorrelated, the coefficients are scalar quantized, while using the planar facet transformation coefficients that are vector quantized. Also the planar facet in the form used codes the texture region much more efficiently because of the variable size block. It is likely that a cosine transform used to replace the planar facet in the algorithm would result in a more efficient code, albeit with increased complexity. This is because the transform results in more decorrelated coefficients while the variable block size would assure a high degree of adaptivity. This possibility is left for future considerations.

Finally, a comparison is made with a technique using vector quantization on images in quadtree form[60]. A mean quadtree representation of an image is first built up by forming a sequence of reduced-size images by averaging over 2 by 2 blocks. A difference quadtree is then built by taking the differences between successive levels in the mean quadtree. Vector quantization is applied to the difference quadtree of the image on a level-by-level basis. All the nodes in the difference quadtree starting from the top level and ending at the bottom level are sent. The kth approximation image can be formed by adding the information of level k to the previously reproduced (k-1)th approximation. Figure 5.11 shows this method compared with PFVQ with a single codebook. The better performance of the PFVQ can be attributed to the higher level of adaptivity to image statistics. The quadtree in this non-adaptive form requires the same amount of bits regardless of the activity in the region. Thus in real image, where many regions are of constant grey level, the quadtree method produces less efficient code. It is likely, that if adaptivity was introduced in the quadtree, perhaps by use of a cutset, that this method would produce less error than the PFVQ.





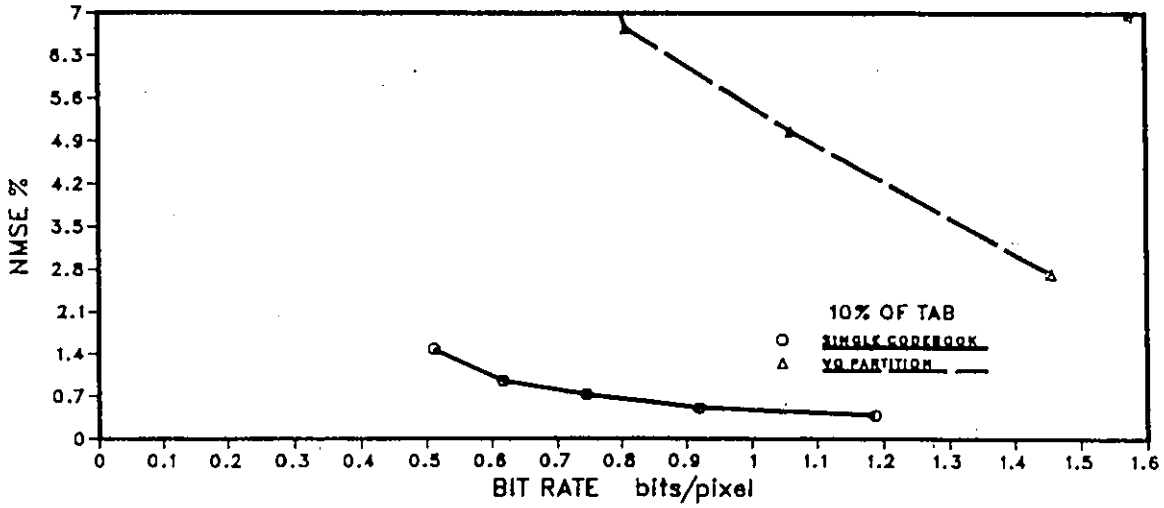


FIGURE 5.10. NMSE VS BIT RATE for vector quantized planar facet vs VQ with a partitioned codebook
The poor performance of VQ with a partitioned codebook largely due to the redundancy in the codebook

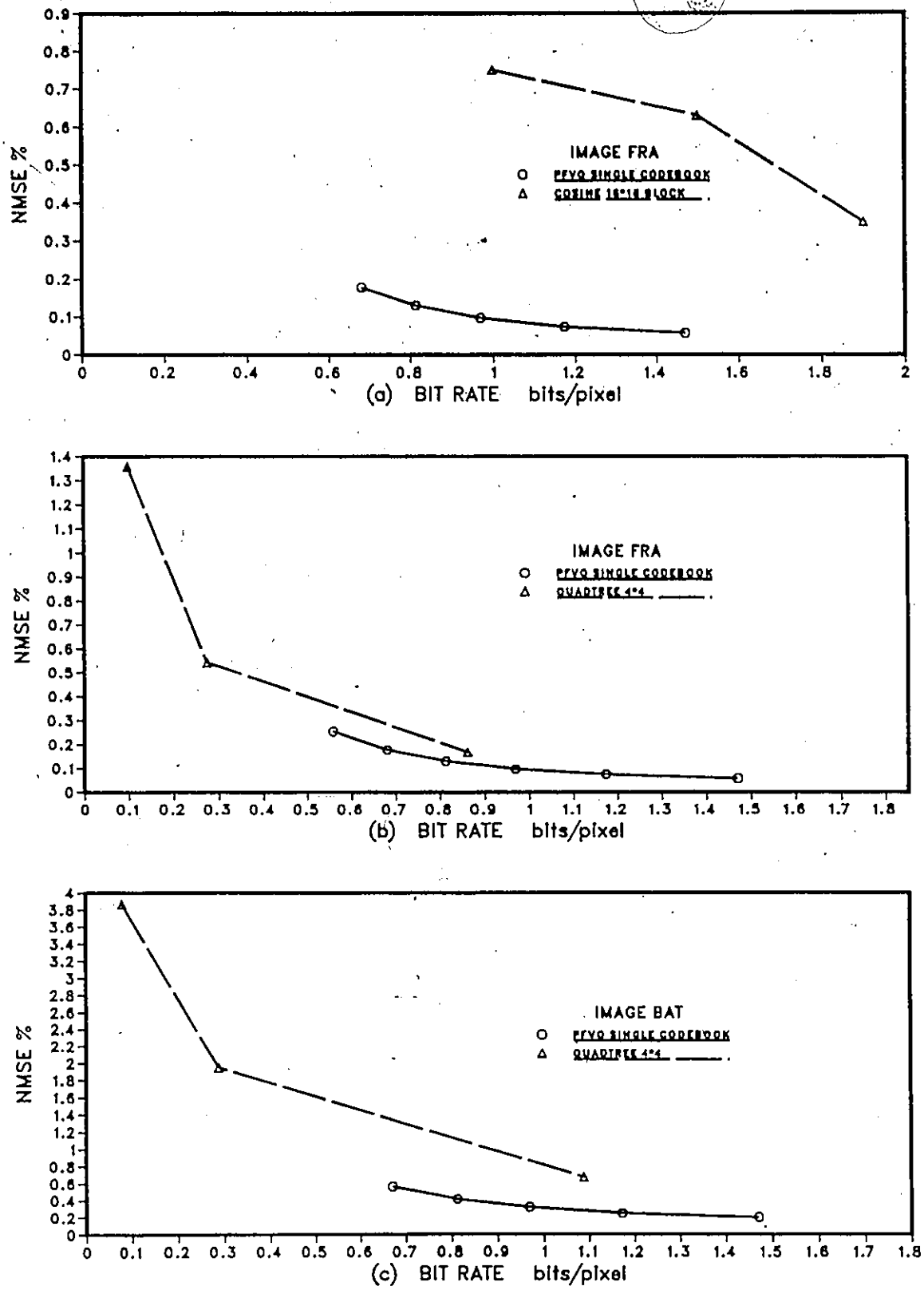


FIGURE 5.11. NMSE VS BIT RATE for vector quantized planar facet compared with block cosine in figure a, and a quadtree method[60] in figure b and c.

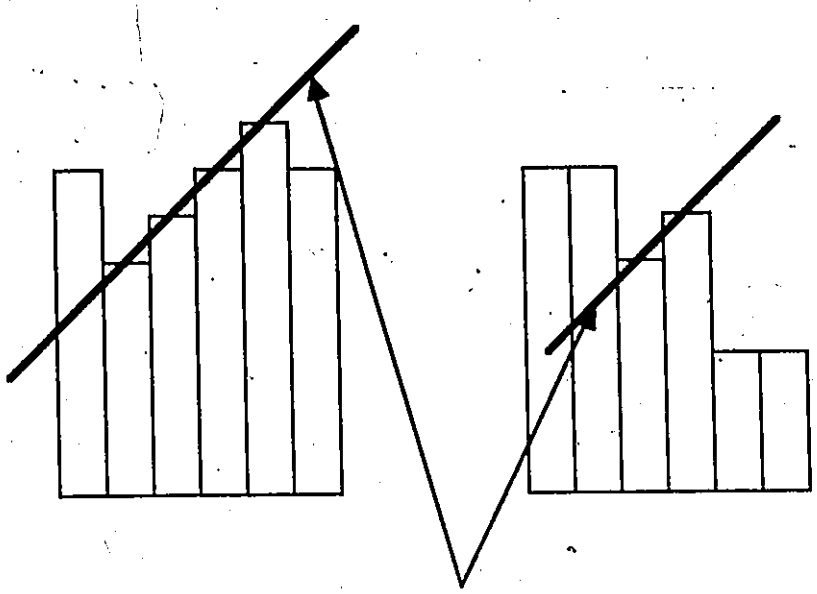


Figure 5.12 A one dimensional version of a 4 by 4 and 2 by 2 block mapping onto the same slope. This allows many vectors to be similar and results in a single codebook being sufficient.

	Block size	Codebook size	No. of bits	Bit rate	% bit rate in overhead
Adaptive VQ	4	64	6	1.53	2.0%
PFVQ	4	64	6	1.53	1.5%
Adaptive VQ	6	256	8	1.52	12.3%
PFVQ	6	256	8	1.42	6.57%
Adaptive VQ	9	512	9	1.51	34.7%
PFVQ	9	512	9	1.18	15.79%
Adaptive VQ	16	512	9	1.49	62.6%
PFVQ	16	512	9	.75	25.0%
Adaptive VQ	20	446	8.8	1.52	71.2%
PFVQ	20	446	8.8	.57	24.8%
Adaptive VQ	24	389	8.6	1.50	71.2%
PFVQ	24	389	8.6	.50	28.39%

Figure 5.13 The table shows the bit rate for the same size codebook for PFVQ and adaptive VQ. By reducing the redundancy in the codebook, more of the bit rate is left for the labels, thus increasing the maximum size block that can be used. The calculations were based on a 256 by 256 pixel image with 8 bits per pixel.

6.0 Conclusions and Recommendations for Further Work

A facet model for image coding which uses a set of piecewise polynomial functions to represent regularly shaped blocks of an image is introduced. Representing an image by a polynomial function of fixed degree and fixed dimension can lead to regions being represented by different degrees of fidelity. By allowing the regions to have different sizes, the error tends to be more uniform in error content. The results of the experiments show that for a class of still images, mixing 4 by 4 and 2 by 2 blocks results in subjectively excellent reproduction of coded images. A key observation in these scalar quantized representations is that the smallest block necessary for no subjectively detectable distortion was 2 by 2 blocks, but the majority of blocks in the images could be transformed using 4 by 4 blocks.

Using vector quantization the bit rate was reduced to 0.3 to 1.5 bits per pixel. For PFVQ like the scalar quantized case, some regions of high detail required as small as 2 by 2 blocks to code well. One notable observation was that the common problem of false contouring that is often accompanied with VQ was not as prevalent in the case of PFVQ. As in the case of scalar quantization many blocks did not require a split to 2 by 2 and a split would only make the code less efficient. In addition using a larger codebook then necessary would also result in a wasting of bit rate. Often there was practically no advantage in using a codebook of 1024 codewords instead of 512 because 512 was already close to the minimum possible error achievable when a block is being represented by a plane. One other noteworthy point was the fact the a single codebook would often lead to the same or better R(d) than when a codebook was used for each block size. This was due to the fact that the vector set for each block size were similar in statistics.

Several questions remained unanswered in this investigation. First and foremost is the selection of the optimum threshold for splitting the block. This question was

empirically examined by the first and second set of experiments but no clear conclusion were drawn. Secondly, unlike most VQ schemes, where larger codebooks mean lower distortion, a larger codebook would only bring the distortion closer to the best fitting plane. The optimum size codebook for the threshold should be investigated further.

Another possibility for further work is using a variable polynomial degree and a variable block size to suit the image statistics more exactly.

Also, as suggested in section 5.2, the same technique of variable block size could be used with any transform and should be investigated further.

References

[1] Shannon, C., 'Coding theorems for a discrete source with a fidelity criterion', *IRE Natl. Conv. Rec.*, pp.142-163, 1959.

[2] Sakrison, D.J.,Halter,M., and Mostafavi,H., 'Properties of the human visual system as related to the encoding of images',in *New Direction in Signal Processing in Communications and Control* (J.Skwirsynski ed.), pp. 83-98, Noordhoff, Leiden, 1975.

[3] Hall,C.F., and Hall,E.L., 'A non-linear model for the spatial characteristics of the human visual system', *IEEE Trans. Sys. Man. Cybernet.* SMC-7, pp. 161-170, 1977.

[4] Lukas,F., and Budrikis, Z,L., 'Picture quality prediction based on visual model', *IEEE Trans. Commmun., COM-30*, pp.1679-1692, 1982.

[5] Campell, F.W., and Robson,J.G., 'Applications of Fourier analysis to the visibility of gratings', *J. Physiol.* (London), 197, pp. 551-556, 1968.

[6] Grey,A.H.,Grey.,R.M., and Markel,J.D., 'Compression of optimal quantizers of speech reflection coefficients', *IEEE Trans. Acoust. Speech Signal Processing*, vol.ASSP-25, pp.9-23, Feb.,1977.

[7] Itakura F.,Saito S., 'Analysis synthesis based on the maximum likelyhood method', *Proc. 6th Int. Congr. Acoust.*, Tokyo, Japan, pp. C-17-20, 1968.

[8] Itakura F.,Saito S., 'A statistical method for estimation of speech spectral density and format frequencies', *Electron. Commun. Jap.*, vol. 53-A, pp. 36-43, 1970.

[10] Clarke, R.J., 'Transform coding of images', *Microelectronics and Signal Processing, Academic Press, N.Y., 1985.*

[11] Pratt,W.K., (Ed.), *'Image transmission techniques', Academic Press, N.Y., 1979.*

[12] Rosenfeld,A., and Kak, A.C., 'Digital picture processing', *Computer Science and Applied Mathematics*, 2nd edition, vol.1, Academic Press, 1982.

[13] Netravali,A.N., 'Interpolative coding using a subjective criterium', Proceedings of the IEEE, Vol. 55, pp. 253-263, Mar. 1963.

[14] Dubois,E., Prasada,B. and Sabri, M.S., 'Image sequence coding', *Image Sequence Analysis*, (Ed.) Huang, T.S., Chapter 3, 1982.

[15] Habibi, A., 'Hybrid coding of pictorial data', *IEEE Trans.on Commun.*, Vol. Com-22, No. 5, pp. 614-624, May 1979.

[16] Gersho, A., Ramamurthi, B., 'Image coding using vector quantization', in *IEEE Proceeding ICASSP*, pp. 428-431, May 1982.

[17] Baker, R.L., and Grey, R.M., 'Image compression using non-adaptive spatial vector quantization', *IEEE Conference on Circuits Systems-Computer*, pp.55-61, 1983.

[18] Murakami, T., Asai., and Yamazaki, E., 'Vector quantizer of video signals' , Electronic Letters 7, pp 1005-1006, Nov. 1982.

[19] King, R.A., Nasrabadi, N.M., 'Image coding using vector quantization in the transform domain', *Pattern Recognition Letters*, Vol. 1, pp. 323-329 July 1983.

[20] Sun, H.F., Goldberg, M., 'Image coding using LPC with vector quantization', *IEEE Proceedings of Int. Commun. and Energy Conf.*, pp. 266-269, Montreal, Canada, Oct. 1984.

[21] Goldberg, M., and Sun, H.F. 'Image sequence coding techniques using transforms vector quantization', *IEEE Trans. on Communication*

[22] Buzo,A., Grey, A.H., and Grey R.M., and Markel, J.D., ' Speech coding based upon vector quantization', *IEEE Trans. on Acoust., Speech, and Sign. Proc.*, ASSP-28, pp-367-376, Aug., 1980.

[23] Boucher, P., and Goldberg, M., 'Color image compression by adaptive vector quantization', *IEEE Proc. ICAASP*, pp. 29.6.1-4, San Diego, Mar., 1984.

[24] Kurtman, C.M., 'Redundancy reduction- A practical method of data compression', *Proc. of the IEEE*, Vol. 55, pp-253-263, Mar. 1963.

[25] Andrew, H.C. and Pratt, W.K., 'Fourier transform coding of images', *Proceeding Hawaii Int. Conf. System Science*, pp. 677-679, Jan., 1968.

[26] Juang, B.H., and Grey, A.H., 'Multiple stage vector quantization for speech coding', *Proc. of the IEEE Int. Conf. on ASSP*, pp. 597-600, Apr. 1982.

[30] Lloyd, S.P., 'Least square quantization in PCM', *Bell Labs. Technical Note.* (1957). Published in Bell Labs special issue on quantization, Mar. 1982.

[31] Buzo,A, Grey, A.H., Grey, R.M. and Markel, J.D., 'Speech coding based upon vector quantization', *IEEE Trans. on ASSP*, Vol. ASSP-28, No. 5, pp.562-574, Oct. 1980.

[32] Fontana,R.J., 'Stochastic stability for feedback quantization schemes', *IEEE Trans. on Information Theory*, IT-28, pp. 248-254, Mar. 1982.

[33] Boucher, P., 'Adaptive vector quantization of pictorial data', *Theses for M.A.Sc.*, Dept. of Elect. Eng. Univ. of Ottawa., 1984.

[49] Webster's Dictionary 1986

[50] R. Haralick and L. Watson, 'A Facet Model For Image Data', *Proceedings IEEE Computer Society Conference on Pattern Recognition and Image Processing*, August 6-8, 1979, Chicago, Illinois

[51] M. Nagao and T. Matsuyyama, 'Edge preserving smoothing', *Fourth International Joint Conference on Pattern Recognition*, Kyoto, November 1978,pp. 432-440.

[52] F. Tomita and S. Tsujji, 'Extraction of Multiple regions by smoothing in selected neighborhoods', *IEEE Trans. Sys. Man. Cybernet., SMC-7*, No. 2, 1977, 107-109

[53] R.M. Haralick, 'Edge and region analysis for digital image data', in *Image Modeling* (Azriel Rosenfeld, Ed.), pp. 171-184, Academic Press, New York, 1981.

[54] E.A. Cohen, 'Generalized sloped models useful in multispectral image analysis', *Comp. Vision, Graphics, and Image Processing*, pp 171-190, Academic Press, New York, 1985.

[55] Tescher, A., 'Transform image coding', in *Image Transmission Techniques*, ed. Pratt, W.K., pp. 113-155, Academic Press, N.Y. 1979.

[56] Huang, J.J.Y., and Schultheiss, P.M., 'Block quantization of correlated Gausian random variables', *IEEE Trans. on Commmun.* COM-11, pp. 289-296, 1963.

[60] Wang,L., and Goldberg,M. 'Progressive image transmission using vector quantization on images in quadtree form',*30th midwest Symposium on circuits and systems*, 1987.

[61] Max J., 'Quantizing for minimum distortion', *IRE Trans. Information Theory*, vol IT-6, pp. 7-12, March 1960.

[62] Chen, W.H., and Smith, H., 'Adaptive coding of Monochrome and Color Images'. *IEEE Transactions on Communications*, COM-25, no.11, Nov 1977, pp. 1285-1292.