ADAPTIVE TRANSFORM CODING OF IMAGES USING A MIXTURE OF PRINCIPAL COMPONENTS

By

ROBERT DOUGLAS DONY, B.A.Sc., M.A.Sc.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (1995) McMASTER UNIVERSITY
(Electrical Engineering) Hamilton, Ontario

TITLE:	Adaptive Transform Coding of Images Using a Mixture of Principal Components
AUTHOR:	Robert Douglas Dony B.A.Sc. (University of Waterloo) M.A.Sc. (University of Waterloo)
SUPERVISOR:	Dr. Simon Haykin
NUMBER OF PAGES:	xvi, 139

Abstract

The optimal linear block transform for coding images is well known to be the Karhunen-Loève transformation (KLT). However, the assumption of stationarity in the optimality condition is far from valid for images. Images are composed of regions whose local statistics may vary widely across an image. A new approach to data representation, a mixture of principal components (MPC), is developed in this thesis. It combines advantages of both principal components analysis and vector quantization and is therefore well suited to the problem of compressing images. The author proposes a number of new transform coding methods which optimally adapt to such local differences based on neural network methods using the MPC representation. The new networks are modular, consisting of a number of modules corresponding to different classes of the input data. Each module consists of a linear transformation, whose bases are calculated during an initial training period. The appropriate class for a given input vector is determined by an optimal classifier. The performance of the resulting adaptive networks is shown to be superior to that of the optimal nonadaptive linear transformation, both in terms of rate-distortion and computational complexity. When applied to the problem of compressing digital chest radiographs, compression ratios of between 30:1 and 40:1 are possible without any significant loss in image quality. In addition, the quality of the images were consistently judged to be as good as or better than the KLT at equivalent compression ratios.

The new networks can also be used as segmentors with the resulting segmentation being independent of variations in illumination. In addition, the organization of the resulting class representations are analogous to the arrangement of the directionally sensitive columns in the visual cortex.

Acknowledgements

I would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada for its funding of this work through a postgraduate scholarship and the financial support of the Department of Electrical and Computer Engineering, McMaster University.

I would like to thank the numerous members of the Communications Research Laboratory, McMaster University, for their assistance with so many aspects of my work.

I would also like to thank the members of the Department of Radiology at the McMaster University Medical Centre for their help in guiding me through the medical aspects of my research. In particular, I am indebted to Dr. Coblentz for his help and patience in organizing and conducting the radiographic evaluations.

I also wish to thank Dr. Nahmias and the other members of the Medical Imaging and Network Development laboratory in the Department of Nuclear Medicine at the McMaster University Medical Centre for their continuing assistance and guidance.

In addition, I would also like to thank the other members of my supervisory committee, Drs. Capson and Yip for their direction and advice during the course of my research.

I would like to express my gratitude to my supervisor, Dr. Simon Haykin for giving me the freedom to pursue this course of research and the invaluable guidance and support along the way.

But above all, to my wife, Lisa, thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The combination of wide availability of both digital computers and highly efficient communication networks means that tne exchange and processing of digital image data is seeing an unprecedented growth. It seems ironic that, in this day and age of gigabit per second transmission channels and gigabyte storage media, there is a need for image compression. However, the volume of image data being generated and processed keeps growing and there appears no end in sight to this trend.

This is particularly evident in the medical imaging field. Today, a wide range of examinations which produce digital images are commonly used. These include computed tomography (CT), digital subtraction angiography (DSA), digital flurography (DF), magnetic resonance imaging (MRI), ultrasonography (US), nuclear medicine (NM), single photon emission computerized tomography (SPECT), and positron emission tomography (PET). Emerging techniques such as computed radiography (CR) promise to add even more ways of generating digital medical images. It is not uncommon to measure the annual generation of digital imagery for a department of radiology in units of terabytes of data. At these rates, the archiving and retrieval of images for more than a few weeks worth of examinations becomes a significant problem. The successful application of image compression methods in this environment could result in a more effective utilization of today's limited medical resources.

1.1 Current Methods

Needless to say, with the large increase in the generation of digital image data, there has been a correspondingly large increase in research activity in the field of image compression. The goal is to represent an image in the fewest number of bits without losing the essential information content within an image. The study of image compression methods has been an active area of research since the inception of digital imaging. Since images can be regarded as simply two-dimensional signals with the independent variables being two-dimensional space, digital compression techniques for one-dimensional signals can be readily extended to images in many cases.

Two of the main approaches for image compression are vector quantization and transform coding. In vector quantization, the image data are transformed by a nonlinear operator which maps image blocks into codeword indices. Each codeword is a 0-dimensional Voronoi centre and is used as the reconstructed image block for the corresponding index upon decoding. In transform coding, the input data undergo a linear transformation which produces a set of coefficients that are less correlated than the original data. The optimal transformation is the Karhunen-Loève transform (KLT), whose basis vectors are the principal components of the input data.

1.2 New Method

Such "classic" techniques trace their roots back to the beginnings of contemporary statistical signal processing theory. Fundamental to the classical theory are the assumptions of linearity, stationarity, and the sufficiency of second-order statistics or Gaussianity. Mathematical tractability, of course, was the primary justification for these assumptions. However, it has been observed that within the field of statistical signal processing there is a shift away from such restrictive assumptions to an emerging paradigm termed "neurosignal processing" [24]. It is being recognized that the classical assumptions are simply not valid for a large number of signal processing applications. Images, for example, by their very nature are highly nonstationary, with the statistics varying greatly from one region in an image to another. Neural networks provide a framework for developing solutions to signal processing problems which account for the nonlinear, nonstationary, and nongaussian characteristics of many signals.

This thesis proposes a new method of representing the data which incorporates features of both vector quantization (VQ) and the principal components analysis (PCA) of transform coding. A mixture of principal components (MPC) is used to represent the data. The data is partitioned into a number of regions or classes resulting in a nonlinear representation, a characteristic of VQ. Within each class, however, the data are represented by an M-dimensional linear subspace defined by the M principal components of the data within the class. This novel representation is particularly appropriate for use in image compression.

Using a combination of neural network tools and the MPC representation, a new class of neural network is developed by the author. It addresses the problem of the nonstationary nature of image statistics by allowing the transformation to adapt to local changes. A mixture model is used where a region in an image is assumed to come from one of K distributions. Each distribution is represented by M principal components. In the network, each class or distribution is represented by a module which contains the principal components for the class. The class for a given input vector is chosen by a classifier which allows the system to adapt to the different regions in an image. The optimal criteria for adaptation are developed in this thesis and the resulting classifiers are used.

The use of adaptation in an optimal manner allows the networks to compress image data with less distortion relative to the optimal nonadaptive approach. In addition, the network can represent the data in a more efficient manner, resulting in computational savings. As the thesis shows, these advantages can be realized for a variety of image types.

1.3 Organization

The thesis is organized as follows.

Chapter 2: To provide the necessary background, the current state-of-the-art in applying neural networks to the problem of image compression is reviewed. This overview highlights some of the advantages that current neural network approaches have over conventional approaches and thereby justifies continued research in this field. It also details many of the tools which will be used later in developing new approaches.

Chapter 3: A novel neural network based image compression method is developed which addresses the need for adaptive processing in an optimal manner. It is an adaptive block transform coding scheme which uses the subspace classifier to classify input blocks. The network is trained by Hebbian learning in a competitive learning framework. Its performance is shown to be superior to that of the optimal nonadaptive transform coder.

Chapter 4: An alternative method is developed which addresses one of the deficiencies in the previous approach. It is derived from information-theoretic criteria and can be shown to be a special case of the above approach. Not only can the rate-distortion performance of the network surpass that of the nonadaptive approach, but the network has significant computational advantages over "fast" block transform methods. A number of variations on this method are presented and evaluated.

Chapter 5: The performance of the new networks is evaluated in a medical imaging application. Digital chest radiographs are compressed using the new networks. The results, in the context of an educational application, are evaluated by expert Radiologists. In addition, a comparison with the classical approach is performed. The results of the evaluations are presented.

Chapter 6: The use of the new networks as image segmentors is investigated. The classification inherent in the adaptive mechanism of the networks is used to produce segmentation maps of a number of images. These maps and the underlying representations are presented and compared.

Chapter 7: The final chapter highlights the results and contributions made by the research presented herein and concludes the thesis.

Chapter 2

Neural Network Approaches to Image Compression

Recently there has been a tremendous growth of interest in the field of neural networks [60, 40, 32, 28, 23]. Yet the wide range of applications and architectures which fall under this category make a specific definition of the term "neural network" difficult. A neural network can be defined as a "massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use" [23]. Generally, neural networks refer to a computational paradigm in which a large number of simple computational units, interconnected to form a network, perform complex computational tasks. There is an analogy between such a computational model and the functioning of complex neurobiological systems. Higher-order neurobiological systems, of which the human brain is the ultimate example, perform extremely complex tasks using a highly connected network of neurons, in which each neuron performs a relatively simple information processing task.

This model contrasts sharply with the classical serial computational model found in most general-purpose computers in use today. In the serial model, a highly complex processor performs computations in a rigidly serial manner. The way in which the two models are programmed is also fundamentally different. In the classical model, explicit program steps or states are provided to the processor which must account for all possible input states. In contrast, neural networks are trained using examples of data which the networks will encounter. During training, the network forms an internal representation of the state space so that novel data presented later will be satisfactorily processed by the network.

In many applications, the neural network model may have a number of advantages over the serial model. Because of its parallel architecture, neural networks may break down some of the computational bottlenecks which limit the performance of serial machines. Since neural networks are trained using example data, they can be made to adapt to changes in the input data by allowing the training to continue during the processing of new data. Another advantage of training is that since data are presented individually, no overhead is required to store the entire training set. This is particularly important when processing very large data sets, of which images are an example. The high degree of connectivity can allow neural networks to self-organize, which is an advantage when the structure of the data is not known beforehand. Finally, since there is an analogy between neural networks and neurobiological systems, existing biological networks could be used as models for designing artificial neural networks; it is hoped that some of the performance characteristics of the biological network will be inherited by the artificial network.

2.1 Classical Image Compression

Before embarking on a review of the current neural network approaches to image compression, a brief overview of conventional techniques is warranted. Most current approaches fall into one of three major categories: predictive coding, transform coding, or vector quantization. In addition, a combination of these techniques may be applied in a hybrid approach. For more detailed descriptions of these methods, the reader is referred to the following works [18, 29, 30, 31, 47, 49, 50, 59, 66].

2.1.1 Transform Coding

A common approach to image compression is the use of transformations that operate on an image to produce a set of coefficients. The goal of this technique is to choose a transformation for which the set of coefficients after quantizing and encoding is adequate to reconstruct an image with a minimum of discernible distortion.

A simple, yet powerful, class of transform coding techniques is linear block transform coding. An image is subdivided into non-overlapping blocks of $n \times n$ pixels which can be considered as N-dimensional vectors $\mathbf{x}$ with $N = n \times n$. A linear transformation, which can be written as an $M \times N$-dimensional matrix $\mathbf{W}$ with $M \leq N$, is performed on each block with the M rows of $\mathbf{W}$, $\mathbf{w}_i$ being the basis vectors of the transformation. The resulting

M-dimensional coefficient vector $\mathbf{y}$ is calculated as

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2.1}$$

If the basis vectors $\mathbf{w}_i$ are orthonormal, that is

$$\mathbf{w}_i^T \mathbf{w}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2.2}$$

then the inverse transformation is given by the transpose of the forward transformation matrix resulting in the reconstructed vector

$$\hat{\mathbf{x}} = \mathbf{W}^T \mathbf{y} \tag{2.3}$$

The optimal linear transformation with respect to minimizing the mean squared error is the Karhunen-Loève transformation (KLT). The transformation matrix $\mathbf{W}$ consists of M rows of the eigenvectors corresponding to the M largest eigenvalues of the sample autocovariance matrix

$$\Sigma = E[\mathbf{x}\mathbf{x}^T] \tag{2.4}$$

The KLT also produces uncorrelated coefficients and therefore results in the most efficient coding of the data, since the redundancy due to the high degree of correlation between neighbouring pixels is removed. The KLT is related to principal components analysis (PCA), since the basis vectors are also the M principal components of the data. Because the KLT is an orthonormal transformation, its inverse is simply its transpose.

A number of practical difficulties exist when trying to implement the KLT. While the calculation of the covariance estimate and its eigendecomposition do not particularly tax even the most commonly available computing resources today, the algorithms used to do these computations are somewhat complex and therefore not suitable for straightforward hardware implementation. Further, the calculation of the covariance estimate requires $O(N^2)$ calculations per training input. As well, the calculation of the forward and inverse transforms is of order $O(N^2)$ for each image block. Due to these difficulties, fixed-basis transforms such as the discrete cosine transform (DCT) [57], which can be computed in order $O(N \log N)$, are typically used when implementing block transform schemes. The Joint Photographics Expert Group (JPEG) have adopted the linear block transform coding approach for its standard using the DCT as the transformation [72]. The two-dimensional

DCT for an $M \times N$ block, whose (m,n)th element is g_{mn}, is defined as

$$G_{uv} = \frac{2c(u)c(v)}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{mn} \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cos\left[\frac{(2n+1)v\pi}{2N}\right] \tag{2.5}$$

where $u = 0, \ldots, M-1$, $v = 0, \ldots, N-1$, and

$$c(k) = \begin{cases} 1/\sqrt{2} & k = 0 \\ 1 & k \neq 0 \end{cases} \tag{2.6}$$

For a first-order Markov process the DCT is asymptotically equivalent to the KLT [57]. Since this model is applicable for many real world images, the coding performance of the DCT is close to that of the KLT while affording a significant reduction in computational complexity.

2.1.2 Vector Quantization

The process of quantization maps a signal $x(n)$ into a series of K discrete messages. For the kth message, there exists a pair of thresholds t_k and t_{k+1}, and an output value q_k such that $t_k < q_k \leq t_{k+1}$. For a given set of quantization values, the optimal thresholds are equidistant from the values. The concept of quantizing data can be extended from scalar or one-dimensional data to vector data of arbitrary dimension. Instead of output levels, vector quantization (VQ) employs a set of representation vectors (for the one-dimensional case) or matrices (for the two-dimensional case) [1, 18, 20, 39, 48]. The set is referred to as the "codebook" and the entries as "codewords". The thresholds are replaced by decision surfaces defined by a distance metric. Typically, Euclidean distance from the codeword is used. The advantage of vector quantization over scalar quantization is that the high degree of correlation between neighbouring pixels can be exploited. Even for a memoryless system, the coding of vectors instead of scalars can theoretically improve performance.

The standard approach to calculate the codebook is by way of the Linde, Buzo and Gray (LBG) algorithm [39]. Initially, K codebook entries are set to random values. On each iteration, each block in the input space is classified, based on its nearest codeword. Each codeword is then replaced by the mean of its resulting class. The iterations continue until a minimum acceptable error is achieved. This algorithm minimizes the mean squared error over the training set and converges to a local minimum in the MSE energy surface. However, it is not guaranteed to reach the global minimum. In addition, the algorithm is very sensitive to the initial codebook. Furthermore, the algorithm is slow, since it requires an exhaustive search through the entire codebook on each iteration.

With this brief review of conventional image compression techniques at hand, we are ready to consider the role of neural networks as applied to the problem of image compression.

2.2 Transform Coding Using Neural Networks

2.2.1 Linear PCA

One solution to the problems associated with the calculation of the basis vectors through eigendecomposition of the covariance estimate is the use of iterative techniques based on neural network models. These approaches require less storage overhead and can be more computationally efficient. As well, they are able to adapt over long-term variations in the image statistics.

In 1949, Donald Hebb proposed a mechanism whereby the synaptic strengths between connecting neurons can be modified to effect learning in a neuro-biological network [25]. Hebb's postulate of learning states that the ability of one neuron to cause the firing of another neuron increases when that neuron consistently takes part in firing the other. In other words, when an input and output neuron tend to fire at the same time, the connection between the two is reinforced.

For artificial neural networks, the neural interactions can be modelled as a simplified linear computational unit as shown in figure 2.1. The output of the neuron, y, is the sum of the inputs $\{x_1, x_2, \ldots, x_N\}$ weighted by the synaptic weights $\{w_1, w_2, \ldots, w_N\}$, or in vector notation,

$$y = \mathbf{w}^T\mathbf{x} \tag{2.7}$$

Taking the input and output values to represent "firing rates," the application of Hebb's postulate of learning to this model would mean that a weight w_i would be increased when both values of x_i and y are correlated. Extending this principle to include simultaneous negative values (analogous to inhibitory interactions in biological networks), the weights $\mathbf{w}$ would be modified according to the correlation between the input vector $\mathbf{x}$ and the output y.

A simple Hebbian rule updates the weights in proportion to the product of the input and output values as

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha y(t)\mathbf{x}(t) \tag{2.8}$$

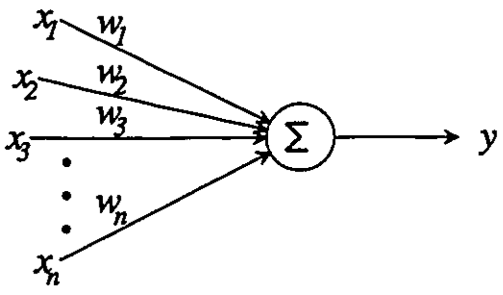


Figure 2.1: Simplified linear neuron.

where α is a learning-rate parameter. However, such a rule is unstable since the weights tend to grow without bound. Stability can be imposed by normalizing the weights at each step as

$$\mathbf{w}(t+1) = \frac{\mathbf{w}(t) + \alpha y(t)\mathbf{x}(t)}{\|\mathbf{w}(t) + \alpha y(t)\mathbf{x}(t)\|} \tag{2.9}$$

where $\|\cdot\|$ denotes the Euclidean norm. This rule has been shown to converge to the largest principal component of the input $\mathbf{x}$ [51, 53, 54, 55]. Oja linearized equation 2.9 using a series expansion to form

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha \left[y(t)\mathbf{x}(t) - y^2(t)\mathbf{w} \right] \tag{2.10}$$

Equation 2.10 has also been shown to converge to the largest principal component [23].

2.2.2 Generalized Hebbian Algorithm

Oja's rule (equation 2.10) has formed the foundation for extending Hebbian learning to simultaneously find the first M principal components. Figure 2.2 shows the architecture of such a system. Each output y_i corresponds to the output of the ith principal component neuron. In vector notation, it can be written as

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2.11}$$

with $\mathbf{y} \in \mathbf{R}^M$, $\mathbf{W} \in \mathbf{R}^{M \times N}$, and $M \leq N$.

Sanger's generalized Hebbian algorithm (GHA) [61, 62, 63] extends Oja's model to compute the leading M principal components using the fact that the computation of any

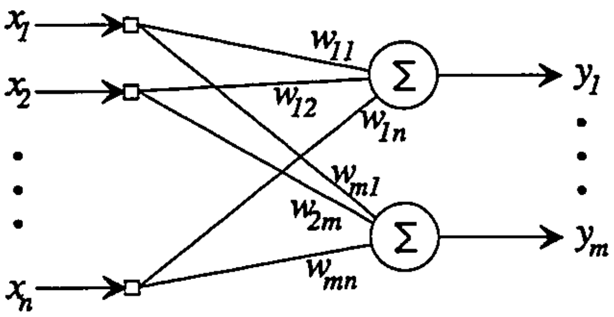


Figure 2.2: *M* principal components linear network.

principal component is identical to that of the first with the data being modified by removing the previous principal components through Gram-Schmidt orthogonalization. The orthogonalization is incorporated into the learning rule to form

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha(t)\left(\mathbf{y}(t)\mathbf{x}^T(t) - \mathrm{LT}[\mathbf{y}(t)\mathbf{y}^T(t)]\mathbf{W}(t)\right) \qquad (2.12)$$

where LT[·] is the lower triangular operator, *i.e.*, it sets all elements above the diagonal to zero. Under the conditions that $\lim_{t\to\infty}\alpha(t) = 0$ and $\sum_{t=0}^{t<\infty}\alpha(t) < \infty$, **W** converges to a matrix whose rows are the *M* principal components [61].

For performance evaluation, Sanger implemented the algorithm using 8×8 input blocks and an output dimension of 8. The network was trained on a 512×512 image using non-overlapping blocks with the image being scanned twice. The learning parameter α was fixed in the range $[0.01, 0.1]$. The coefficients were non-uniformly quantized with the number of bits varying with the sample variance of each coefficient. At a compression of 0.36 bpp, a normalized MSE of 0.043 resulted. When the same matrix **W** was used to code a second independent image, a compression of 0.55 bpp resulted in a normalized MSE of 0.023. Sanger also applied this algorithm to a texture segmentation problem and has used it to model receptive fields.

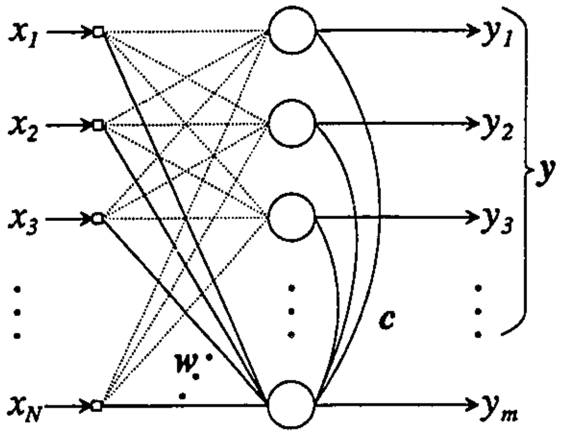


Figure 2.3: Network for the APEX algorithm.

2.2.3 Adaptive Principal Component Extraction

Sanger's method uses only feedforward connections for calculating the M principal components. An alternative approach proposed by Földiák [16] is to use "anti-Hebbian" feedback connections to decorrelate the components. The justification of this approach was based on earlier work by Barlow and Földiák [2] on the visual cortex. Building on this approach and the work of Oja [51], Kung and Diamantaras [34, 10, 35] have developed a sequential solution, called adaptive principal component extraction (APEX), in which the output of the mth principal component y_m can be calculated based on the previous $m-1$ components through

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2.13}$$

and

$$y_m = \mathbf{w}^T\mathbf{x} + \mathbf{c}^T\mathbf{y} \tag{2.14}$$

where $\mathbf{y}$ is the vector of the first $m-1$ components, $\mathbf{W}$ is the weight matrix for the first $m-1$ components, $\mathbf{w}$ is the weight vector for the mth component and $\mathbf{c}$ corresponds to an anti-Hebbian removal of the first $m-1$ components from the mth component. Figure 2.3 shows the architecture of the network.

The learning rule is stated as

$$\Delta\mathbf{w} = \alpha(y_m\mathbf{x} - y_m^2\mathbf{w}) \tag{2.15}$$

and

$$\Delta \mathbf{c} = -\beta(y_m \mathbf{y} - y_m^2 \mathbf{c}) \tag{2.16}$$

Kung and Diamantaras have shown that the weights $\mathbf{w}$ converge to the m-th principal component, given that the first $m-1$ components have already been calculated. As the network converges, the anti-Hebbian weights $\mathbf{c}(t)$ converge to zeroes. The optimal learning parameters α and β are calculated as

$$\alpha = \beta = \left(\sum_{i}^{n} y_i^2\right)^{-1} \tag{2.17}$$

where n is the number of input patterns. This choice of learning-parameters allows the network to adapt to slowly varying changes in the signal statistics. Further, the additional calculation of a further principal component requires only a linear order, $O(N)$, of multiplications per iteration. For testing, the algorithm was applied to a set of $n = 20$ data points of dimension 5. An average squared distance between the principal components extracted from the covariance matrix and those computed using the algorithm was found to be 0.34×10^{-3} after 194 iterations.

Chen and Liu [6] have extended the concept of using feedback connections to extract M principal components simultaneously from the training data, as opposed to the sequential computation of the APEX algorithm. The forward calculation of their network is identical to equation 2.14. In addition, the training rule for the orthogonal weights $\mathbf{c}$ is the same as equation 2.16. The learning rule for the principal component vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ is modified to become

$$\Delta \mathbf{w}_i = \alpha\{\mathbf{B}_i[y_m \mathbf{x} - y_m^2 \mathbf{w}] - \mathbf{A}_i \mathbf{w}_i\} \tag{2.18}$$

where

$$\mathbf{A}_i = \begin{cases} \mathbf{0}, & i = 1 \\ \sum_{j=1}^{i-1} \mathbf{w}_i \mathbf{w}_i^T, & i = 2, 3, \ldots, N \end{cases} \tag{2.19}$$

and

$$\mathbf{B}_i = \mathbf{I} - \mathbf{A}_i \tag{2.20}$$

The matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ perform the orthogonalization during training. Chen and Liu have shown that the weight vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ converge to the M principal components while the anti-Hebbian weights $\mathbf{c}_i$ converge to the zero vector.

2.2.4 Robust Principal Components Estimation

Xu and Yuille [74, 75] have addressed the problem of robustness in the estimation of the principal components. To account for outliers in the training set they first introduce a binary field which includes data from within the distribution and excludes outliers. As such a function is non-differentiable, they propose a weighting function based on a Gibbs distribution to account for the degree of deviation from the distribution a data sample may have. By establishing an energy function to be minimized, $J(\mathbf{x}, \mathbf{W})$, the gradient descent learning rule becomes

$$\mathbf{W} = \mathbf{W} - \alpha D_{\beta,\eta}(\mathbf{x}, \mathbf{W}) \nabla J(\mathbf{x}, \mathbf{W}). \tag{2.21}$$

where $D_{\beta,\eta}(\mathbf{x}, \mathbf{W})$ is a weighting function defined as

$$D_{\beta,\eta}(\mathbf{x}, \mathbf{W}) = (1 + \exp[\beta(J(\mathbf{x}, \mathbf{W}) - \eta)])^{-1} \tag{2.22}$$

which effectively reduces the influence of data points from outside the distribution *i.e.,* those having a large energy function value $J(\mathbf{x}, \mathbf{W})$. The parameter β is a deterministic annealing parameter which starts as a small value and then increases to infinity. The parameter η determines the region considered as being outside the distribution. As for the choice of the energy function $J(\mathbf{x}, \mathbf{W})$, a number of Hebbian rules can be expressed in terms of the gradient of some energy functions.

2.2.5 Discussion of PCA Algorithms

There are a number of advantages which these learning rules have in calculating the M principal components from a data set over standard eigendecomposition techniques. If $M \ll N$, the iterative techniques can be more computationally efficient [52]. As well, because of their iterative nature, they can be allowed to adapt to slowly varying changes in the input stream. A third advantage is that no extra overhead is required to store the data or its higher-order statistics. Finally, if an extra basis were to be required, its computation would be more efficiently performed using the iterative learning rules.

These PCA algorithms using neural networks may be categorized into two classes: *re-estimation algorithms* which use only feedforward connections, and *decorrelating algorithms* which have both feedforward and feedback connections [4]. The GHA is an example of the former. The learning rule of equation 2.12 may be restated as

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t) y_j(t) \left[\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right] \tag{2.23}$$

where $\hat{\mathbf{x}}(t)$ is the *re-estimator* defined by

$$\hat{\mathbf{x}}(t) = \sum_{k=0}^{j} \mathbf{w}_k(t) y_k(t) \tag{2.24}$$

The successive outputs of the network are forced to learn different principal components by subtracting estimates of the earlier components from the input before the data are involved in the learning process. In contrast, the APEX algorithm is a decorrelating algorithm. The anti-Hebbian connections decorrelate the successive outputs, resulting in the computation of different principal components.

Recently, there has been some interest in extending the above approaches to a new class of nonlinear PCA networks in which a sigmoidal activation function is added to the model of a neuron. With such a model, it is possible to extract higher-order statistics from the data; however, the resulting basis vectors lose their orthogonality with respect to each other. While nonlinear PCA has been successfully applied to the separation of sinusoidal signals, their usefulness in image compression may be limited due to the loss of orthogonality.

2.3 Vector Quantization Using Neural Networks

2.3.1 Self-Organizing Feature Map Algorithm

Another class of neural network based approaches to image compression applies Kohonen's self-organizing feature map (SOFM) [33] to the problem of codebook design in vector quantization. Kohonen introduced the concept of classes ordered in a "topological map" of features. In many clustering algorithms such as K-means each input vector $\mathbf{x}$ is classified and only the "winning" class is modified during each iteration [14]. In the SOFM algorithm, the vector $\mathbf{x}$ is used to update not only the winning class, but also its neighbouring classes according to the following rule:

For each vector $\mathbf{x}$ in the training set:

1. Classify $\mathbf{x}$ according to

$$\mathbf{x} \in C_i \;\text{ if }\; \|\mathbf{x} - \mathbf{w}_i\| = \min_j \|\mathbf{x} - \mathbf{w}_j\| \tag{2.25}$$

2. Update the features $\mathbf{w}_j$ according to

$$\mathbf{w}_j(t+1) = \begin{cases} \mathbf{w}_j(t) + \alpha(t)[\mathbf{x} - \mathbf{w}_j(t)], & C_j \in N(C_i, t) \\ \mathbf{w}_j(t), & C_i \notin N(C_j, t) \end{cases} \tag{2.26}$$

where $\mathbf{w}$ is the feature vector, α is a learning parameter in the range $0 < \alpha < 1$, an·l $N(C_i, t)$ is the set of classes which are in the neighbourhood of the winning class C_i at time t. The class features $\mathbf{w}_i$ converge to the class means. The neighbourhood of a class is defined according to some distance measure on a topological ordering of the classes. For example, if the classes were ordered on a two-dimensional square grid, the neighbourhood of a class could be defined as the set of classes whose Euclidean distances from the class are less than some specified threshold. Initially, the neighbourhood may be quite large during training, *e.g.*, half the number of classes or more. As the training progresses, the size of the neighbourhood shrinks until, eventually, it only includes the one class. During training, the learning parameter α also shrinks down to a small value (*e.g.*, 0.01) for the fine tuning (convergence) phase of the algorithm.

2.3.2 Properties of the SOFM Algorithm

The SOFM algorithm has a number of important properties which make it suitable for use as a codebook generator for vector quantization [23].

1. The set of feature vectors are a good approximation to the original input space.

2. The feature vectors are topologically ordered in the feature map such that the correlation between the feature vectors increases as the distance between them decreases.

3. The density of the feature map corresponds to the density of the input distribution so that regions with a higher probability density have better resolution than areas with a lower density.

2.4 Summary

Investigations into the application of neural networks to the problem of image compression have produced some promising results. By their very nature, neural networks are well suited to the task of processing image data. The characteristics of artificial neural networks which include a massively parallel structure, a high degree of interconnection, the propensity for storing experiential knowledge, and the ability to self-organize, parallel many of the characteristics of our own visual system. In contrast, standard approaches to the processing of image data have been based on a serial paradigm of information processing which is more suited to sequential information such as language. As a result, neural network approaches to image compression have been shown to perform as well as or better than standard approaches.

Hebbian learning has formed the basis for a number of iterative methods of extracting the principal components of image data for use as the basis images in block transform coding. Both the GHA and the APEX algorithms have been shown to converge to the M principal components. When only the first few principal components are required, significant computational savings can be realized. Their iterative nature can allow the basis images to adapt over long-term variations. As well, memory requirements are reduced as there is no need to store the entire data set or its second-order statistics.

The ability of SOFM to form ordered topological feature maps in a self-organizing fashion has given it a number of advantages over the standard LBG algorithm for the generation of VQ codebooks. It has been found to be less sensitive to initial conditions, have fast convergence properties and have the ability to produce a lower mean distortion codebook.

However, none of the existing techniques reviewed herein addresses the short-term, region-to-region variation within an image. The local statistics of an image may vary abruptly from an edge to a flat region to a region of texture, all within a relatively small area. Because of the stability-plasticity dilemma, methods of adapting to long-term or macro variations are not suitable for short-term or micro variations. The stability-plasticity dilemma states that there is a trade-off between the adaptivity of a system and its stability [24]. A system which allows adaptation over an extremely short interval may become unstable due to its response to spurious disturbances. On the other hand, ignoring any change allows for a perfectly stable system but does not allow any adaptation.

With the above discussion in mind, we are ready to proceed to address the issue of adaptive image coding using the neural network based tools presented.

Chapter 3

Optimally Integrated Adaptive Learning

For most image compression techniques, the optimal method based on some model of the image statistics is well known. However, the assumptions upon which the conditions for optimality have been based can be called into question. Specifically, the use of global statistics for generating an optimal coding scheme may not be appropriate. The use of adaptation in many compression techniques has resulted in significant improvements in performance. While these improvements clearly indicate that adaptive processing is of merit, there has been inadequate study into the optimality of the adaptation criterion. This chapter develops a new approach to adaptive transform coding in which the criterion for adaptation is shown to be optimal.

3.1 A Spectrum of Representations

The previous chapter has summarized two of the main representations used for image compression, namely principal components analysis (PCA) and vector quantization (VQ). Both these representations, in effect, are the two limits of a potential spectrum of representations. Vector quantization is a zero-dimensional representation of an N-dimensional data set while principal components is a full N-dimensional representation. The author proposes a new approach which combines advantages of these two limiting cases.

3.1.1 Principal Components

The KLT uses up to the full N principal components to represent N-dimensional data. The representation is complete, *i.e.*, if all N components are used, the data are represented *exactly*. Therefore, the representation is continuous since all possible input vectors may be represented. To reiterate, an N-dimensional data vector $\mathbf{x}$ is represented by N coefficients $\mathbf{y}$ which is calculated as

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{3.1}$$

where $\mathbf{W}$ is an $N \times N$ matrix whose ith row is the ith principal component. On reconstruction

$$\mathbf{x} = \mathbf{W}^T\mathbf{y} \tag{3.2}$$

The same holds true for other orthogonal basis functions such as the DCT. Because it uses all the components, it is a very powerful technique due to its complete and continuous representation. The representation is also a linear mapping of the data set. This characteristic affords a high degree of mathematical tractability in the analysis and design of this approach. However, the usefulness of linear techniques on images is limited due to the highly nonlinear nature of most images. For example, the human visual system (HVS) which can outperform any artificial vision system in all but the most trivial tasks gains much of its power through the many nonlinear stages of processing and representation.

3.1.2 Vector Quantization

At the other extreme, VQ is a purely discrete representation of the data. Unlike PCA which uses up to the full N principal components, VQ uses only one of a number of Voronoi centres, (a codeword) for each input vector. For a set of K codewords, $\{\mathbf{w}_i | i = 1, \ldots, K\}$, an input vector $\mathbf{x}$ is represented by the ith codeword such that the reconstructed vector, $\hat{\mathbf{x}}$, is

$$\hat{\mathbf{x}} = \mathbf{w}_i \quad \text{where } \|\mathbf{x} - \mathbf{w}_i\| = \min_{j=1}^{K} \|\mathbf{x} - \mathbf{w}_j\| \tag{3.3}$$

Each of the centres or codewords is a zero-dimensional point in the N-dimensional input space. Therefore, the representation under vector quantization is a highly nonlinear function of the input vector.

3.1.3 A Mixture of Principal Components

Between these two extremes lies the mixture of principal components (MPC) [26]. Like VQ, this approach partitions the data set into a number of non-overlapping regions. However, each region is represented not by a zero-dimensional point but by a M-dimensional linear subspace. Like PCA, each subspace is a continuous representation with only M orthogonal components where $0 < M < N$. Each input vector is assigned to the most appropriate partition and then represented by the M basis vectors of that component. This representation can be expressed as

$$\mathbf{y} = \mathbf{W}_i\mathbf{x}, \quad \text{where } \mathbf{x} \in C_i \tag{3.4}$$

where $\mathbf{W}_i$ is an $M \times N$ matrix whose rows are the M principal components of the partition C_i. The reconstructed vector, $\hat{\mathbf{x}}$, is calculated as

$$\hat{\mathbf{x}} = \mathbf{W}_i^T\mathbf{y}, \quad \text{where } \mathbf{x} \in C_i \tag{3.5}$$

The MPC approach combines the features of both PCA and VQ representations. Within a class, an input vector is represented as a continuous, linear combination of the M basis vectors of the subspace in a manner analogous to the PCA representation. But, because of the partitioning of the data into a discrete number of regions or classes, the MPC effects a nonlinear mapping of the data as does VQ.

Figure 3.1 illustrates the relation between the three representations for a two-dimensional example. The PCA approach forms a complete, continuous representation using a linear combination of the two basis vectors. With VQ, the input space is partitioned, in this example, into 10 regions. Each region is represented by a Voronoi centre. Under MPC, the space is also partitioned, in this case into four regions. Within each region, the data is represented by a single basis vector. For higher-dimensional input spaces, the number of basis vectors may be two or more, forming planes, hyperplanes or other higher-dimensional subspaces within the input space.

3.2 Adaptation

The hybrid approach of the MPC representation provides a new method of addressing the problem of adaptation in an image compression context.

A major issue with many image processing applications is their implicit assumption of stationarity. The fallacy of this assumption is the reason why many image processing

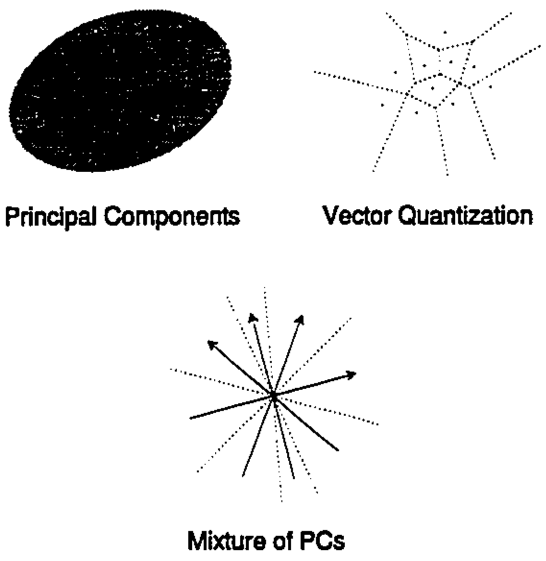


Figure 3.1: A spectrum of representations in two dimensions.

techniques perform poorly in the vicinity of edges since the image statistics around edges tend to be quite different from the global statistics. Methods such as the KLT which are globally optimal are, in effect, locally sub-optimal. Therefore, if processes were made to adapt to local variations in an image, their performance would improve.

To account for variations in the local statistics, a transformation must adapt locally. A transformation $T(\cdot)$ can be allowed to vary by specifying a parameter set Ω such that $\mathbf{y} = T(\Omega, \mathbf{x})$. If the parameter set were to vary according to the neighbourhood around a given data point, $N_{\mathbf{x}}$, then the transformation can be allowed to adapt to the characteristics of the surrounding data. The transformation can then be represented as

$$\mathbf{y} = T(\Omega(N_{\mathbf{x}}), \mathbf{x}) \tag{3.6}$$

To simplify matters, the statistical variations can be quantized into a finite number of classes. Since many neighbourhoods may map to the same class or feature set, the explicit dependence on the neighbourhood may be omittedin equation 3.6. Therefore, regions within an image may be classified as belonging to one of the classes $\{C_1, C_2, \ldots, C_K\}$. There is a corresponding parameter set $\boldsymbol{\Omega} = \{\Omega_1, \Omega_2, \ldots, \Omega_K\}$, where each element Ω_i describes the characteristics of the corresponding class C_i. The transformation is then represented as

$$y = T(\Omega_i, x), \quad x \in C_i \tag{3.7}$$

It has been recognized for some time that the use of adaptation in coding can improve performance and there has been a great deal of success in the use of adaptation for some types of coding techniques [19, 47. 37, 36, 49]. In some of the earlier work, the adaptation occurs in the quantization stage while the transformation remains fixed [67, 69]. This approach has also been explored in more recent work [3]. Adaptation has also been applied to VQ methods [58, 38, 71]. However, in many cases, adaptation has been applied in a rather *ad hoc* manner. For example, "high frequency" components may be coded differently from "low frequency" components. Alternatively, edges of different orientations may be treated separately. In some cases, the adaptation occurs in the quantization stage while the transformation remains fixed. There has yet to be a treatment of the optimality of the criterion upon which the adaptation is based.

The use of classes for adaptation introduces a significant measure of complexity to the process. To begin with, the nature of the classification must be determined. This is not a trivial matter. The classification criterion should somehow be related to the nature of the transformation process. If the classification is inappropriate, then the adaptation may not be optimal. As well, the appropriate parameters for each class must be determined. The parameters should be sufficient to describe what makes a given class unique.

Instead of imposing *a priori* the classes for the adaptation, the data itself should provide the information on how to appropriately perform thc segmentation. In such a self-organizing approach, features of the data are used to compute a measure of similarity between data points and each class. In an iterative manner, similar data are grouped together in classes and the resulting representation of each class is then used to re-classify the data. The problem, of course, is how to determine the appropriate features and measure of similarity, so that the resultant classes form the basis for optimal adaptation.

3.3 Subspace Pattern Recognition

In many classical pattern recognition techniques, classes are represented by prototypical feature vectors and class membership is determined by some transformed Euclidean distance between an input vector and the prototypes [14]. For example, with the K-means and LBG vector quantization algorithms, the classes are represented by their means and the vector-to-class distance is the Euclidean distance between the class mean and an input vector. The class boundaries form closed regions within the input space.

Such class representations are not suitable for use with linear transform coding techniques. If two input vectors were to differ only by a scalar multiple and one of the vectors were adequately represented by a set of basis vectors of a linear transformation, then the same set of bases would also adequately represent the other vector. It would be appropriate, then, that the two vectors belonging to the same class have the same transformation bases. However, under a Euclidian distance-based classifier, the difference in vector norm between the two vectors would mean that they may belong to different classes. Therefore, a classification scheme which is independent of the vector norm of the data is required for adaptive linear transform coding. The linear subspace classifier has this property.

In subspace pattern recognition, classes are represented as linear subspaces within the original data space and the basis vectors which define the subspace implicitly define the features of the data set [52]. The classification of data is based on the efficiency by which the subspace can represent the data as measured by the norm of the projected data.

If the data are represented as $\mathbf{x} \in \mathbf{R}^N$, and $\mathbf{U} \in \mathbf{R}^{M \times N}$ is an orthonormal matrix with $M < N$, then the projector $\mathbf{P}$ is defined as

$$\mathbf{P} = \mathbf{U}^T\mathbf{U} \tag{3.8}$$

with projection of $\mathbf{x}$ by $\mathbf{P}$ being

$$\hat{\mathbf{x}} = \mathbf{P}\mathbf{x} \tag{3.9}$$

The subspace $S_P \subset \mathbf{R}^N$ is defined by

$$S_P = \{\mathbf{z} | \mathbf{z} = \mathbf{P}\mathbf{x}, \mathbf{x} \in \mathbf{R}^N\} \tag{3.10}$$

and is spanned by the M N-dimensional row vectors of $\mathbf{U}$.

To adequately represent the data, the subspace should match the data as closely as possible. Referring to figure 3.2, this means that the expected norm of the projected vector is maximized, *i.e.*, maximize

$$E\left[\|\mathbf{P}\mathbf{x}\|\right] \tag{3.11}$$

Equivalently, the square of the norm of the residual $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is minimized, *i.e.*, minimize

$$E\left[\|\tilde{\mathbf{x}}\|^2\right] = E\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right] \tag{3.12}$$

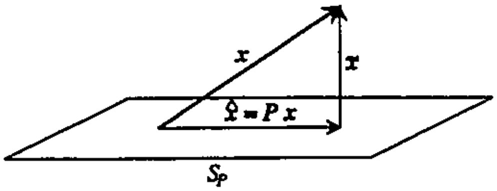


Figure 3.2: Projection of $\mathbf{x}$ by $\mathbf{P}$ on S_P.

In other words, maximizing the expected norm of the projection is equivalent to finding the transformation which minimizes the MSE. As stated earlier, the linear transformation which minimizes the MSE is the KLT. Therefore, the optimal subspace for a data set is the space spanned by the eigenvectors corresponding to the M largest eigenvalues of the data covariance matrix, or equivalently, the M principal components of the data.

For classification purposes, one can define a set of K classes which are defined by K subspaces $\{S_1, S_2, \ldots, S_K\}$. Each subspace S_i is defined by its projector $\mathbf{P}_i$, which can be calculated using equation 3.8 with the rows of $\mathbf{U}$ being the M principal components of the class data. Once the classes are defined, a data vector $\mathbf{x}$ is assigned to the class under whose projection its norm is maximized:

$$\mathbf{x} \in C_i \text{ if } \|\mathbf{P}_i\mathbf{x}\| = \max_{j=1}^{K} \|\mathbf{P}_j\mathbf{x}\| \tag{3.13}$$

Since the use of equation 3.13 results in classes whose membership criterion is independent of the norm of the input data, it may be used in an adaptive linear transform coding scheme.

Based on the subspace classifier, the set of projection matrices partitions the input data space into a number of regions and the M principal components or rows of $\mathbf{U}$ can be used to represent the data vectors within each region. This is simply the MPC representation. Referring to equations 3.4 and 3.5, an input vector $\mathbf{x}$ is assigned a class C_i according to the subspace classifier of equation 3.13. The projection matrix $\mathbf{P}_i$ which defines the class C_i is formed as

$$P_i = \mathbf{W}_i^T\mathbf{W}_i \tag{3.14}$$

where the M rows of $\mathbf{W}_i$ are the M principal components of the data of class C_i. Therefore, the use of the MPC representation allows for optimal adaptation in a transform coding approach to image compression.

3.4 Network Architecture

The incorporation of the subspace classifier into an adaptive transform coding scheme using the MPC representation results in a modular network whose coding stage is shown in figure 3.3. The network consists of a number of independent modules whose outputs are mediated by the subspace classifier. Each module consists of M basis blocks of dimension $n \times n$ which defines a single linear transformation. The inner product of each basis block with the input image block results in M coefficients per module, represented as an M-dimensional vector $\mathbf{y}_i$. Each module corresponds to one class of input data. The choice of class and therefore the coefficient vector to be transmitted along with its class index is determined by the subspace classifier. The selection is based on the class whose projected vector norm $||\hat{\mathbf{x}}_i||$ is maximum. The projected vector $\hat{\mathbf{x}}_i$ is calculated by taking the inverse transformation of the coefficient vector.

The signal is decoded using the same set of transformations. The class index is used to choose the class for the inverse transformation and the resulting reconstructed image block $\hat{\mathbf{x}}$ is calculated.

The network efficiently represents both the linear transformation and the classification criterion. The same set of basis blocks is used to calculate the coefficients for coding and the reconstructed image block for decoding. As well, they *define* the module's class through the linear subspace they span. Therefore, the network requires no extra overhead in terms of information required to effect the adaptation.

3.5 Optimally Integrated Adaptive Learning

The problem, then, is to calculate the optimal set of weights. Without knowing *a priori* the required classes, their defining projectors $\mathbf{P}_i$, and their corresponding transformation bases, a learning algorithm is required to extract the appropriate parameters from the dataset.

A new class of unsupervised learning algorithms is proposed which combines both principal components extraction and competitive learning, and adapts to mixed data from a number of distributions in a self-organizing fashion. The algorithms produce an adaptive linear transformation that is optimal with respect to minimizing the mean squared error between the input data and the decoded data. As such, they are particularly well suited to the task of image compression.

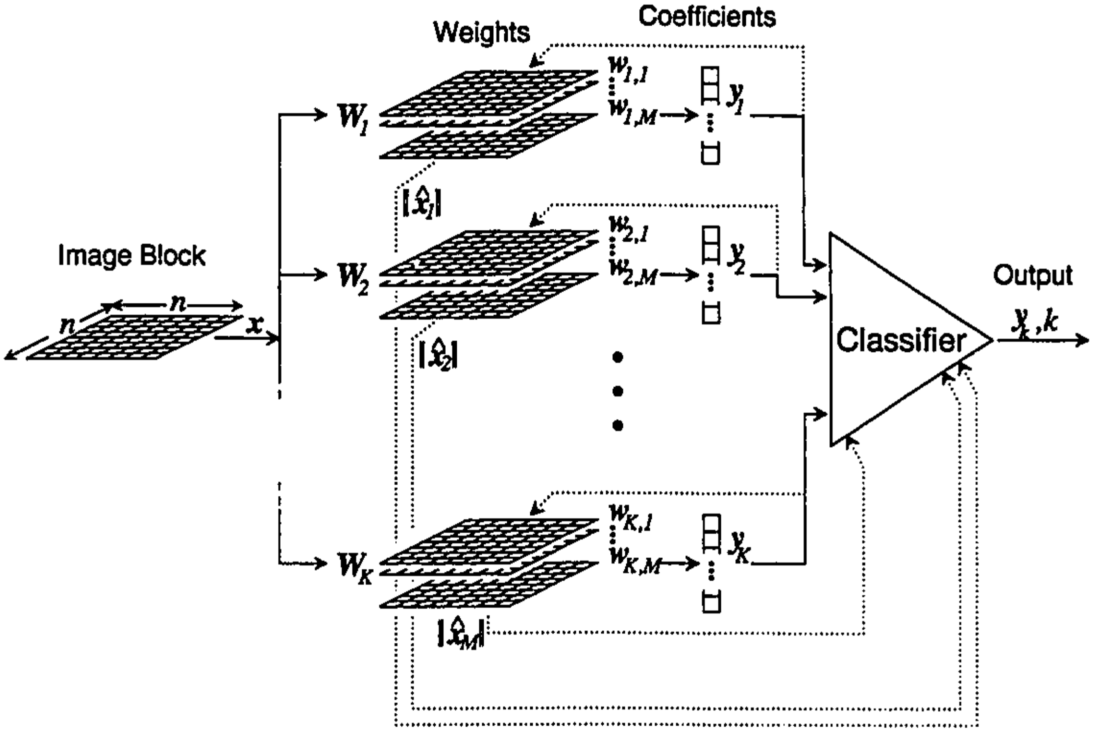


Figure 3.3: Modular system architecture of OIAL. Input are blocks of $n \times n$ pixels. The K transformations $\mathbf{W}_i$ consist of M basis blocks of size $n \times n$ and output an M-dimensional vector $\mathbf{y}_i$. The coefficient vector to be sent is chosen by the subspace classifier based on the maximum norm of the projected vector $\|\hat{\mathbf{x}}_i\|$.

The general form of the class of optimally integrated adaptive learning (OIAL) algorithms is as follows:

1. Initialize K transformation matrices $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K\}$.
2. For each training input vector $\mathbf{x}$:
   (a) classify the vector based on the subspace classifier

$$\mathbf{x} \in C_i \text{ if } \|\mathbf{P}_i\mathbf{x}\| = \max_{j=1}^{K} \|\mathbf{P}_j\mathbf{x}\| \tag{3.15}$$

   where $\mathbf{P}_i = \mathbf{W}_i^T\mathbf{W}_i$, and

   (b) update transform matrix $\mathbf{W}_i$ according to:

$$\mathbf{W}_i = \mathbf{W}_i + \alpha\mathbf{Z}(\mathbf{x}, \mathbf{W}_i) \tag{3.16}$$

   where α is a learning parameter, and $\mathbf{Z}(\mathbf{x}, \mathbf{W}_i)$ is a learning rule such that equation 3.16 converges to the M principal components of $\{\mathbf{x}|\mathbf{x} \in C_i\}$.

3. Repeat for each training vector until the transformations converge.

In the first step, some care must be taken in the choice of the initial set of transformation matrices. They should be representative of the distribution space of the training data. If some of the $\mathbf{W}_i$'s were to be initialized to values corresponding to regions outside of the distribution space, then they would never be used. Hence, the resulting partition would be clearly suboptimal. There are a number of methods to reduce the possibility of this occurring as described here:

- Arbitrarily partition the training sei into K classes and estimate the corresponding transformations using either iterative learning rules or batch eigendecomposition.
- Use a single fixed-basis transformation such as the DCT and add a small amount of random variation to each class to produce a set of unique transformations.
- Use an estimate of the global principal components of the data with a small amount of random variation added to each class.

It is this latter approach which we have used in the experimental section of this paper.

Algorithms based on the above outline will produce K transformation matrices $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K\}$. Given the appropriate learning rule $\mathbf{Z}(\mathbf{x}, \mathbf{W}_i)$ in equation 3.16, each matrix will converge to the KLT for that particular class of data. Since the KLT minimizes the mean squared error, each $\mathbf{W}_i$ is optimal for its class. The classification rule in equation 3.15 is equivalent to finding the transformation which results in the minimum squared error for the particular vector. The combination of these two rules, therefore, produces the optimal set of linear transformations for the resulting partition. Conversely, for the resulting set of linear transformations, the partitioning of the data is optimal with respect to minimizing the MSE.

Whether or not the resulting partitions are optimum raises the following question: has the algorithm converged to the global minimum or a local minimum in the energy surface? The energy surface for the OIAL, because of its nonlinear nature, can be quite complex. Like other nonlinear networks, the proof of convergence to a *global* minimum may not be mathematically tractable [23]. However, in all the experience the author has had with the algorithm on "real" data, the algorithm has consistently converged to a satisfactory result every time.

The satisfactory results may be due to the use of an estimate of the global principal components to initialize the network. The network thus may have a better chance of finding a good local minimum or possibly the global minimum through this incorporation of prior knowledge. The network starts out with an adequate solution, and the training, in effect, fine tunes the network through an adaptative process. If the network were to be initialized to completely random values, the convergance of the network to a useful solution may not occur.

Since there exists a number of learning rules which compute the M principal components of a data set, the choice of $\mathbf{Z}(\mathbf{x}, \mathbf{W}_i)$ will depend on the desired computational efficiency and convergence properties. Whether the learning rule used is the linear Hebbian rule of equation 2.10 with recursive calculation of the M principal components, or the others mentioned in section 2.2, [62, 63, 10, 34, 6, 75, 76], the resulting set of transformations would be the same. In fact, if the algorithm were implemented in a batch mode, the explicit calculation of the eigenvectors of the class covariance matrices would also produce the same transformation bases.

It is also interesting to note that the convergence of the transformation matrices to the M principal components of the class data implies optimality but not *vice versa.* Any orthonormal transformation whose basis vectors span the space defined by the principal components is optimal. For the subspace classifier, the projector $\mathbf{P}_i$ would be identical and the performance of the coding and decoding transformations in terms of MSE would also remain unchanged.

Since the goal at hand is to demonstrate the validity of this technique, the choice of learning rule can be rather arbitrary. At this point, no attempt has been made to evaluate the characteristics of the various learning rules to determine the most appropriate one. Such an evaluation is left for future research. The rule chosen for the present study is the Generalized Hebbian Algorithm (GHA) devised by Sanger [61].

3.6 Evaluation

3.6.1 Method

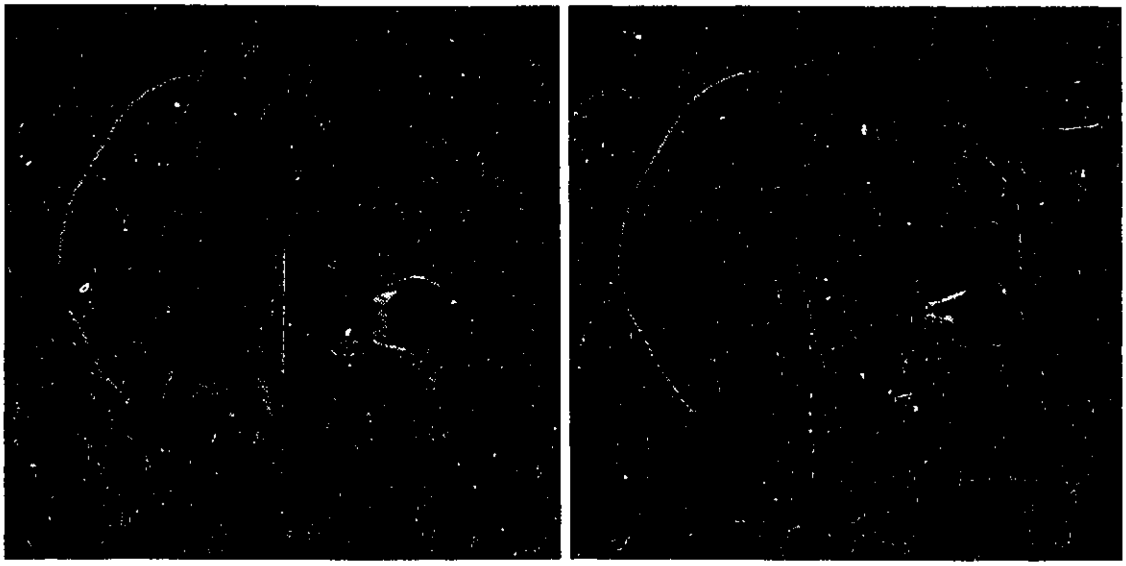
To evaluate the performance of the optimally integrated adaptive learning class of algorithms, a set of experiments were performed. As just mentioned, the learning rule chosen was the GHA. The learning parameter α in equation 3.16 for the ith component at iteration k was calculated using an adaptive form of equation 2.17 [10],

$$\alpha_i(k) = \left(\sum_{l=0}^{k} \gamma^{k-l} y_i^2(l)\right)^{-1} \tag{3.17}$$

where γ is analogous to the "forgetting factor" in the adaptive recursive least squares (RLS) algorithm [21]. For the results presented herein, γ was chosen to be $\gamma = 0.995$. The transformations were initialized to an estimate of the M global principal components with a small amount of random noise (*e.g.*, $\sigma = 0.001$) added to each set of transformations.

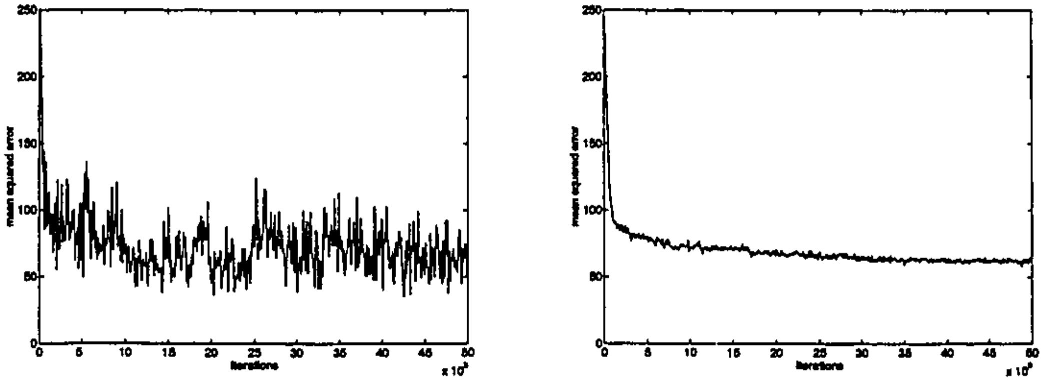
Figure 3.4a shows the magnetic resonance image (MRI) used for training. The image in figure 3.4b was the adjacent section from the same study (patient) and was used for testing. Each image consists of 256×256 pixels with the dynamic range of 8 bits or 256 gray levels. The training image was divided into blocks of 8×8 pixels for an input dimension of $N = 64$. The blocks were overlapped at two pixel intervals for a total number of training samples of 15,625. During training, the samples were presented in random order. A number of system configurations were evaluated. Both the number of coefficients, M, and the number of classes, K, were varied. For comparison, the KLT was also calculated based on the same training data.

A typical learning curve for a system with 4 coefficients and 128 classes is shown in figure 3.5a. Each point represents an average MSE over 10 samples to reduce the block-to-block



(a) (b)

Figure 3.4: MR image for (a) training, (b) testing.



(a) (b)

Figure 3.5: Learning curves for system with 4 coefficients and 128 classes. (a) Typical curve. (b) Ensemble average of 100 learning curves.

variation. The curve shows that within 5000 iterations, the system has formed a sufficient representation of the data to reduce the MSE by approximately one third. The remaining iterations essentially fine tune the system. The ensemble average over 100 such learning curves is shown in figure 3.5b. The same set of initial transformation matrices was used in each training run but the order in which the data were presented varied. This curve shows that the network typically achieves convergence by two to three iterations through the entire training set for this configuration.

The test image was divided into 8×8 non-overlapping blocks. These blocks were transformed by the previously computed system into a set of coefficients, quantized, and then transformed back into image blocks. The coefficients were quantized in a similar manner to that of the JPEG standard [72]. The first coefficient was coded via first order DPCM using a uniform quantizer. The remaining coefficients were coded via PCM using a uniform quantizer. For a given coding rate, the same quantization interval was used for all the coefficients. The quantized data were then Huffman coded with a codebook optimized for Laplacian distributions. The number of bits assigned for the class information was simply $\log_2 K$ bits per block. The data were quantized using a number of intervals resulting in a number of bit rates. For the KLT, the identical coding scheme was used except, of course, that no additional bits per block were required to code the class assignment.

3.6.2 Results

Figure 3.6 shows the experimental rate-distortion curves for the various OIAL network configurations. The distortion is measured in decibels (dB) of peak-signal-to-noise ratio (PSNR) defined as

$$\mathrm{PSNR} = 10 \log_{10} \frac{x_{\mathrm{max}}^2}{E[(x - \hat{x})^2]} \tag{3.18}$$

where x is the original data, $\hat{x}$ is the reconstructed data, and x_{max} is the maximum data value which is in this case $x_{\mathrm{max}} = 255$. The figure shows that the use of adaptation has resulted in improved performance over the non-adaptive KLT. For a given coding rate, there exists a number of OIAL networks which can code the data with less distortion. For example, at 0.25 bits per pixel (bpp), the KLT results in a PSNR of 28.8 dB (MSE of 84.8) while the 4-component, 128-class network has a PSNR of 29.9 dB (MSE of 66.5), an improvement of over 1 dB. Conversely, for a given distortion level, the network can encode the image using fewer bits. For example, at 30 dB PSNR, the KLT can encode the image

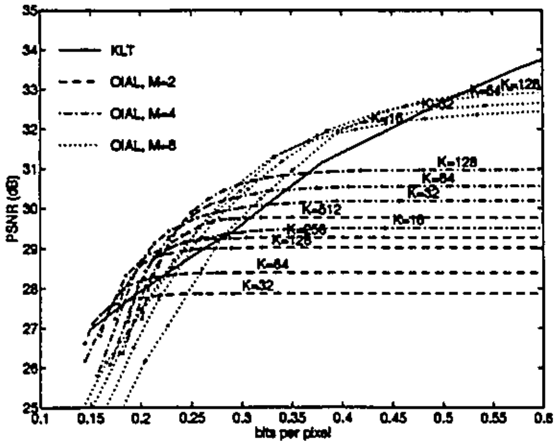


Figure 3.6: Rate-distortion for OIAL and KLT compression.

at 0.321 bpp while the OIAL network can reduce the representation to 0.255 bpp, a savings of over 20%.

Generally, as the number of coefficients increases, the allowable distortion decreases while the bit rate increases. The extra coefficients allows an image to be encoded with a higher fidelity. However, more bits are then required. According to the figure, for coding the image at less than 0.25 bpp, networks with 2 coefficients should be used, at between 0.25 bpp and 0.3 bpp, 4-coefficient networks are best, and over 0.3 bpp, networks with 8 coefficients are required.

The figure also shows that increasing the number of classes decreases the distortion. Since the degree of adaptivity is directly related to the number of classes, this decrease in distortion clearly demonstrates the advantage of using a locally adaptive coding scheme over a non-adaptive method. Of course, an increase in the number of classes also increases the number of bits per pixel because of the extra class assignment information required. It is also interesting to note that the relative improvement due to doubling the number of classes is significantly less with 8 components than with 4 or 2 components.

Table 3.1 shows the rate-distortion performance for the 4-component, 128-class OIAL network. For fine quantization intervals, the coding rate is high while the distortion is low. In this region, an increase in the quantization interval results in a decrease in the bit rate but minimal increase in distortion. As the quantization interval continues to increase,

Quantization Interval	bpp	PSNR
4	0.481	30.96
8	0.419	30.94
12	0.387	30.94
16	0.359	30.87
20	0.340	30.81
24	0.325	30.75
32	0.300	30.62
40	0.282	30.44
48	0.267	30.23
56	0.250	29.90
64	0.236	29.50
80	0.222	28.86
96	0.205	28.05

Table 3.1: Rate-distortion data for 4-coefficient, 128-class OIAL network.

the distortion begins to increase with an accompanying decrease in bit rate. At the coarse quantization region, the increase in distortion becomes significantly greater that the decrease in bit rate as the quantization interval increases.

It is interesting to note that the gray level entropy [49] for the image used for testing is 6.36 bits per pixel. At a quantization iterval of 32 for the 4-coefficient, 128-class OIAL network, the coding rate is 0.3 bpp. Out of those bits, 0.11 bits are used to encode the class information while 0.19 bits are used to encode the coefficient values. By reducing the over-all bit rate by a factor of 21 relative to the gray level entropy, a distortion of only 30.62 dB is incurred.

While performance measures based on squared error provide a quantitative measure of performance and are easily computed, they are no substitute for a qualitative comparison. Figure 3.7a shows details of the resulting image for a coding rate of 0.25 bpp using the OIAL algorithm with 4 coefficients and 128 classes. For comparison, figure 3.7b shows the corresponding details using the KLT at the same rate of 0.25 bpp. The PSNR for the former is 29.9 dB, and for the latter it is 28.8 dB. When examining the detailed structure of the two images, it is clear that the OIAL image preserves more features than the KLT image. In the upper forehead region near the skull, the dark line of the outer table of the skull between the outer white line of the skin and the white line of the diploë is visible in the former, but completely obscured in the latter. The same is true of the detail in the top portion of the orbit. Not only does the KLT lose information, it also introduces texture variations in the brain tissue which are not present in the original nor in the OIAL image. This texture also interferes with the visibility of the folds in the outer portion of the brain. Generally, the boundaries of the image blocks are far more pronounced in the KLT image than in the OIAL image.

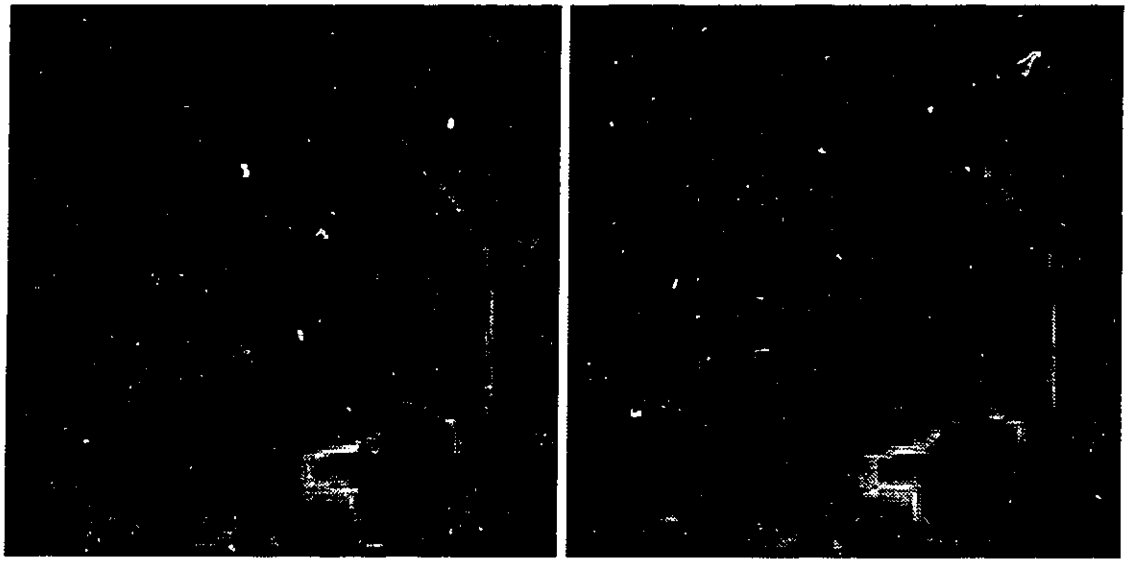
Techniques do exist which can reduce the block effects for block transform coding methods. For example, Malvar's Lapped Othogonal Transform (LOT) uses overlapping blocks to reduce blocking effects [44, 45, 43]. Coifman and Meyer have developed an orthogonal projector to extract overlapping blocks for a transform coding scheme [7]. However, such approaches were not used in this evaluation so that intrinsic differences between the two methods are clearly shown.

3.6.3 Generalization

As stated above, the claims of optimality are only valid for the class of images having similar statistical characteristics as the training data. For testing purposes, training and testing were performed on similar images, namely, adjacent sagittal head MRI scans of a single patient. While the general form of the two images is similar, at the block level there are significant differences. The promising results presented above are therefore a good indication that the network *generalizes* well within that particular class of image. In other words, its performance is similar for images outside the training set but within the defined class.

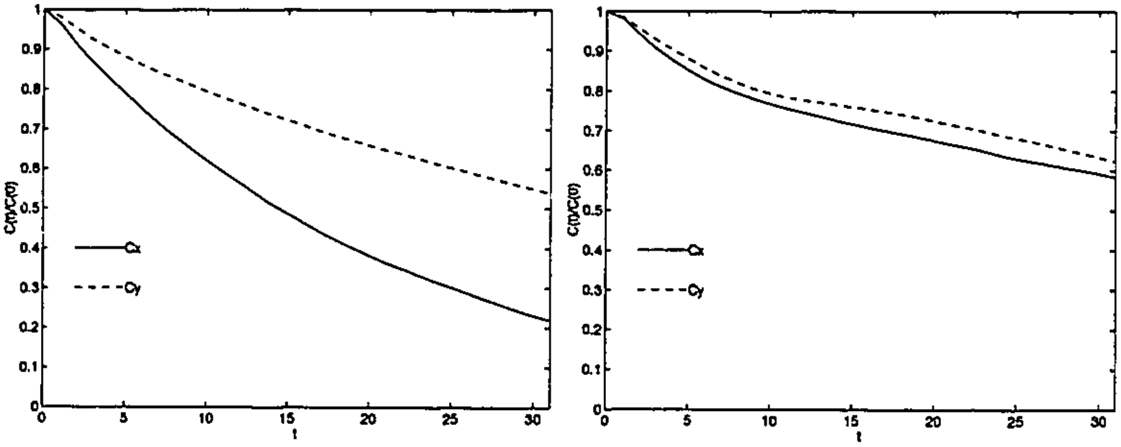
While the "within class condition" may seem restrictive at first, in practice this would not be so. If the encoder and decoder both had a common set of networks, one for each class of images, then the appropriate network would be used by both the encoder and decoder depending on the type of image. For example, in a radiological application there could be separate networks for the various study types, *e.g.*, head MRI, body CT, chest radiograph, *etc.* Because each network generalizes well within its class of image, there is no need to transmit or store a unique network for each particular image.

While there is no claim being made here that there exists a single network configuration which would perform well as a general-purpose image compression scheme across a wide variety of images, it is interesting, nevertheless, to see how well a system trained on one type of image generalizes *outside* that image type. Figure 3.9 shows the Lenna image which



(a) (b)

Figure 3.7: Details of coding at 0.25 bpp. (a) OIAL coding with 128 classes, 4 coefficients per block, PSNR of 29.9 dB. (b) KLT coding, PSNR of 28.8 dB.



(a) (b)

Figure 3.8: Autocovariance for (a) Lenna image and (b) MR image for training. The solid line is the autocovariance in the x direction and the dashed line is the autocovariance in the y direction.

	Lenna	MRI
Size	512×512	256×256
Mean	109.7	51.4
ρ_x	0.955	0.978
ρ_y	0.979	0.982

Table 3.2: Comparison between Lenna image and MR image used for training.

is obviously quite different from the image used for training as shown in figure 3.4a. To begin with, the dimensions of the two are different; the Lenna image is 512 × 512 pixels in size while the MR image is 256 × 256. While both have a dynamic rage of 256 gray levels, the mean pixel value of the former is 109.7 while the mean value of the latter is 51.4. Examining the autocovariance further illustrates the differences. In the x direction, the autocovariance $C_x(\tau)$ is defined by

$$C_x(\tau) = E[(I(x+\tau, y) - m_I)(I(x, y) - m_I)] \tag{3.19}$$

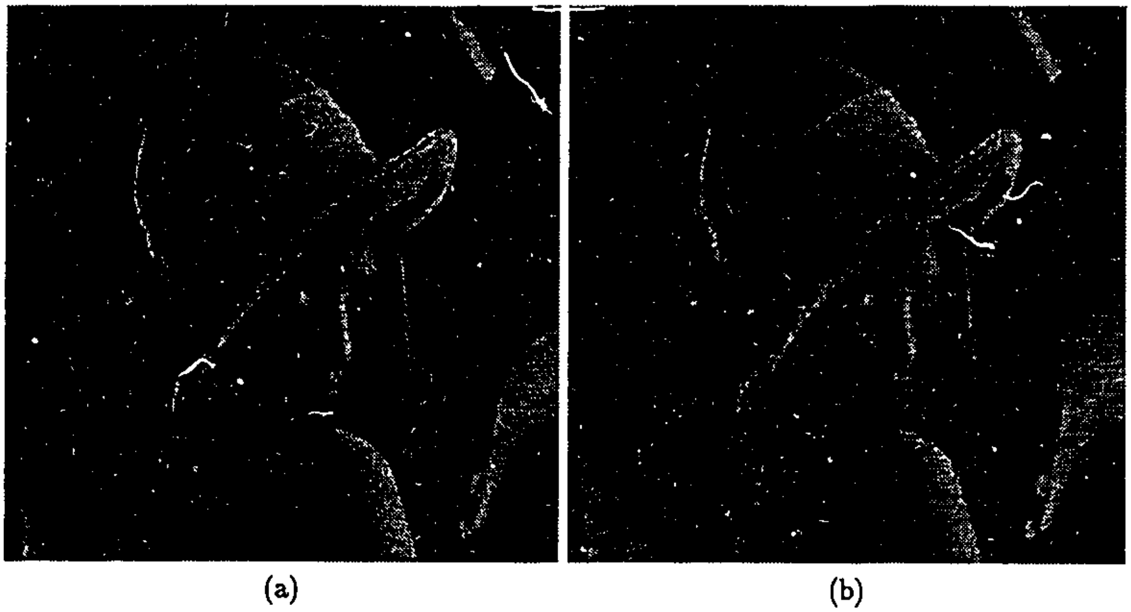
where $I(x, y)$ is the image value at pixel co-ordinate (x, y), m_I is the image mean, and the expectation is taken over all pixels. The autocovariance in the y direction, $C_y(\tau)$, is similarly defined. Figure 3.8 shows the autocovariances for the Lenna image and the MR image used for training. The MR image has a higher degree of correlation than the Lenna image. In addition, the autocovariance in the x and y directions for the Lenna image are quite different. For a stationary Markov-I signal, the autocovariance is given by

$$C(\tau) = \rho^{|\tau|} \tag{3.20}$$

for $0 < \rho < 1$ where ρ is the adjacent correlation coefficient which is the correlation coefficient between adjacent image pixels. Applying this model to the two images under discussion and estimating the adjacent correlation coefficients for the x and y directions for the Lenna image yields $\rho_x = 0.955$ and $\rho_y = 0.979$ respectively. The respective values for the MR image are $\rho_x = 0.978$ and $\rho_y = 0.982$. Table 3.2 summarizes these differences.

The same 4-coefficient, 32-class network used in section 3.6.2 was used to compress the Lenna image at 1 bpp. Figure 3.10a shows the resulting image. The PSNR for this image is 30.2 dB. For comparison, the image was compressed using 4 coefficients of the KLT of *itself* and quantized to the same number of bits (1 bpp). The resulting image is shown in

Figure 3.9: Lenna image for testing generalization.



(a)

(b)

Figure 3.10: Lenna image coded at 0.5 bpp using (a) OIAL, 128 classes, 4 coefficients per block, (b) KLT, 4 coefficients

figure 3.10b and has a PSNR of 28.3. These two images clearly show that the OIAL system trained on a head MR image performs better than the KLT optimized for the specific image being coded. As with figure 3.7, the use of OIAL coding results in less noticeable block effects and better edge preservation. In addition, the OIAL method preserves more texture detail which is particularly noticeable in the feather and the hat band.

3.7 Summary

A new approach to adaptive compression has been developed, based on an optimally integrated adaptive learning (OIAL) class of algorithms. The architecture for such a system consists of a number of modules, each consisting of a number of basis blocks. Each module corresponds to a class of input data and performs a linear transformation on its class data using the bases. Not only do the basis images specify the linear transformations, but they also define the classes by way of the linear subspaces in the input space that each set of bases forms. The system is trained by combining a subspace classifier to identify the appropriate class module and a recursive learning rule which extracts the principal components from the data. Since a transformation whose bases are the M principal components is the minimum MSE linear transformation for compression and the use of the subspace classification method produces the minimum MSE classification, the network will converge to an optimal state in which the over-all MSE is minimized.

The new method addresses some of the deficiencies with current image compression techniques. It has been realized for some time that image processing methods must take into account the mixture of the various region types found within images. Techniques based on global measures of optimality will not perform well on a local level. Therefore, processes must adapt to such local variations. While identifying the need for adaptation, there has been a lack of rigorous treatment of the optimality of the adaptation criteria. The following characteristics of the OIAL approach address this concern:

- The adaptation is optimal, since both the transformation and the classification result in a minimum MSE representation of the data.

- The adaptation of the system during training is self-organizing.

- No assumptions about the importance of or relation between the various regions within an image are imposed beforehand.

- The adaptation is on a microscopic scale. It responds to variations on a block-to-block basis. Other adaptive techniques respond to slowly varying changes over a large number of data points.

- The adaptation criterion is efficiently represented by the system architecture, since each set of basis images serves the dual purpose of defining both the linear transformation and the class representation.

The results presented herein have shown that the new method can outperform the globally optimal linear transform. The same image was coded at the same compression ratio using both the KLT and the new approach. For the new approach, the distortion was reduced and the image quality was improved. Also, more image details were preserved and fewer artifacts were introduced.

The network has also been shown to have good generalization properties. On an image substantially different from the training image, the performance of the network is actually better than the KLT based on that particular image.

Chapter 4

Multi-class Maximum Entropy Coder

The results presented in the previous chapter have shown that the use of an adaptive coding scheme can improve coding performance with respect to the optimal nonadaptive KLT. Furthermore, the optimal classifier in terms of minimizing the MSE for a given number of coefficients per block, M, is the subspace classifier. However, as the following discussion will show, there are some limitations to this approach as a result of how the optimality criterion was stated. This chapter presents an alternative optimality criterion based on Shannon's information theory which leads to another adaptive network for coding images.

4.1 Limitations of OIAL

The goal in the previous chapter was to find both a set of linear transformations, each with M basis vectors for which the mean squared error between the original data and the reconstructed data is minimized, and the corresponding optimal classification rule. It is clear that for a given partitioning of the data, the M basis vectors which minimize the MSE for a class are the M largest principal components or some linear combination thereof. Alternatively, for a given set of linear transformations, the classifier which minimizes the total MSE is the subspace classifier where the subspace of each class is spanned by its transformation basis vectors.

However, this evaluation of optimality deals with only half the story in transform coding. Transform coding consists of two stages. In the first stage, a set of coefficients is calculated using the transformation. Then, the values of the coefficients are coded as a sequence of bits by a quantizer. This second stage is not accounted for in the above approach. The following example illustrates the type of problem this may cause.

Suppose that among K classes, there is a particular class C_k defined by the subspace S_k of at least dimension 2. Now suppose that within that class there are two distinct subclasses C_{k1} and C_{k2} which have different distributions with different coefficient variances. Because the variance of a coefficient may be quite different for the two subclasses, the bit allocations requirements may also be different. For example, the first two variances of the two subclasses may be related as

$$\sigma^2_{1,k1} \gg \sigma^2_{1,k2}$$

and

$$\sigma^2_{2,k1} \ll \sigma^2_{2,k2}$$

Under these conditions, data from subclass C_{k1} would require more bits for y_1, while data from subclass C_{k2} would require more bits for y_2. However, the subspace classifier would not be able to differentiate between the two subclasses and, the resulting bit allocations would therefore not be optimal.

It is also interesting to see the effect of increasing the number of basis vectors, M, per class. At the limit, when $M = N$, the class subspaces are simply the original data space. In this case, there is no adaptation since the projection for each class is the same, namely the identity operator. Therefore, allocating $\log_2 K$ bits per block for the class index is an inefficient use of bandwidth. Furthermore, the computational requirement for classifying is not required. In this case, the optimal network is simply the non-adaptive KLT.

These simple examples demonstrate the inadequacy of neglecting the effects of quantization in deriving an optimally adaptive transform coder. While it is true that one could incorporate the quantization into the projector, the resulting subspaces would no longer be linear. Furthermore, this approach may require different networks optimized, *i.e.*, trained, for different quantization conditions. This is hardly a practical approach.

Alternatively, we could turn to information theory which deals with such issues in a principled manner. It is this latter approach which will now be pursued in establishing the optimal criterion for adaptation.

4.2 Information-Theoretic Approach

The goal of lossy coding is to find a transformation of the image data in which the coding rate of the resulting representation is minimized given an upper limit on the allowable distortion level. The relation between coding rate and distortion is given by the rate-distortion function $R(D)$. It is defined to be the smallest coding rate or number of bits for which the average distortion does not exceed D.

In linear block transform coding, an image block of N pixels in size undergoes a linear transformation which produces a set of N coefficients. The rate-distortion function per image block is, therefore, the sum of the rate-distortion functions of each coefficent,

$$R(D)_{block} = \sum_{i=1}^{N} R_i(D) \tag{4.1}$$

where $R_i(D)$ is the rate-distortion function for the ith coefficient. For block transform coding, the goal is to find a transformation which minimizes the sum in equation 4.1 given the distortion D. To solve this constrained minimization problem, we use Shannon's rate-distortion theory [64].

For a random variable with a given variance, it can be shown that the upper bound of its rate-distortion function is the rate-distortion function for a Gaussian random variable of the same variance [22]. For a squared error distortion measure, the rate-distortion function can be calculated as

$$R(D) = \begin{cases} \frac{1}{2}\log_2(\sigma^2/D) & 0 \leq D < \sigma^2 \\ 0 & \sigma^2 \leq D \end{cases} \tag{4.2}$$

By applying the Gaussian upper bound of equation 4.2 to the rate-distortion function for block transform coding (equation 4.1), it follows that to find the set of orthonormal basis vectors that minimizes the number of bits per block, the following function must be minimized

$$\begin{aligned} R_{block}(D) &\leq \sum_{i=1}^{N} \frac{1}{2}\log(\sigma_i^2/D) \\ &= \frac{1}{2}\log\left(\prod_{i=1}^{N}\sigma_i^2\right) - \frac{N}{2}\log D \end{aligned} \tag{4.3}$$

where σ_i^2 is the variance of ith coefficient, y_i, of the basis vector $\mathbf{w}_i$.

For an orthonormal system, the sum of the coefficient variances is the average energy of the signal. For a given signal it is, in effect, a constant. To minimize the product $\prod \sigma_i^2$ in equation 4.3, given that the sum $\sum \sigma_i^2$ is constant, σ_1^2 should be maximized. Therefore, the solution for the problem at hand is to find $\mathbf{w}_1$ which concentrates as much of the signal's energy as possible into the first coefficient, *i.e.*, maximizes the variance of y_1. The resultant vector $\mathbf{w}_1$ is simply the first principal component of the input data.

In an adaptive approach, there are K classes. Each class, C_i, has a corresponding basis vector, $\mathbf{w}_i$. To minimize the rate-distortion function within each class, the variance of the first coefficient of each class must be maximized. Therefore the optimal set of basis vectors must be the set of principal components for each class.

The assignment of data vectors to the classes must be in accordance with the goal of minimizing the within-class rate-distortion function or equivalently maximizing the variance of the first coefficient. It then follows that a vector be assigned to the class whose weight vector $\mathbf{w}_k$ calculates the largest magnitude coefficient or equivalently the largest squared coefficient. Therefore, the optimal classification rule is

$$\mathbf{x} \in C_k \text{ if } y_k^2 = \max_{i=1}^{K} y_i^2 \tag{4.4}$$

where $y_i = \mathbf{w}_i^T\mathbf{x}$. Because an input vector is assigned to the class based the largest squared value of the coefficient, the use of equation 4.4 as a classifier maximizes the within-class variance. From equation 4.2, maximizing the variance has the effect of maximizing the entropy. Hence, equation 4.4 can be referred to as the maximum entropy classifier.

4.2.1 Discussion

The subspace classifier of section 3.3 assigns a vector to a class based on the maximum norm of the projected vector. Equivalently, the square of the norm may also be used. To reiterate, the square of the norm is calculated as

$$\begin{aligned} \|\hat{\mathbf{x}}\|^2 &= \|\mathbf{W}_k^T\mathbf{y}_k\|^2 \\ &= \mathbf{y}_k^T\mathbf{W}_k\mathbf{W}_k^T\mathbf{y}_k \\ &= \mathbf{y}_k^T\mathbf{y}_k \\ &= \|\mathbf{y}_k\|^2 \end{aligned} \tag{4.5}$$

Using the norm of the projected vector for classifying is equivalent to using the norm of the coefficient vector. Therefore, the maximum entropy classifier is equivalent to the subspace classifier when the dimension of the subspace is $M = 1$.

While the classification criteria of the two approaches has been shown to be equivalent, the maximum entropy classifier requires fewer computations per test. The calculation of y

must be performed for both the maximum entropy classifier and the subspace classifier and requires N multiplications and N additions. However, the calculation of the norm of the projected vector for the subspace classifier requires an additional $2N$ multiplications and N additions.

The maximum entropy classifier may also be viewed as a feature detector. For a given feature to be detected, the optimal filter is the matched filter which is a time reversed version of the feature. Equivalently, it is also the first principal component of the data. For a set of such filters, one for each feature, the filter whose output energy is maximum is considered the winning feature. If the set of weights $\{\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_K\}$ were to be considered a set of such filters, then the maximum entropy classifier is equivalent to a feature detector. Since the weights are the first principal components of the data corresponding to each class, then this classifier is using the most important or most informative feature of each of the classes.

4.2.2 Network Architecture

The incorporation of the maximum entropy classifier into an adaptive transform coding scheme results in the Multi-class Maximum Entropy Coder (McMEC). The modular architecture of the coding stage of the system is shown in figure 4.1 and is similar in form to that of the OIAL system. It consists of a number of independent modules whose outputs are mediated by a maximum entropy classifier. Each module consists of a transformation basis vector (block), $\mathbf{w}_k$, which defines two things: a single linear transformation and a class of input data. The input to the network consists of non-overlapping image blocks of size $n \times n$, $\mathbf{x}$. The inner product of each vector $\mathbf{w}_i$ with the input vector results in a coefficient, y_i, for each module. The choice of class is determined by the maximum entropy classifier that chooses the class for which the square of the coefficient is maximum. The encoder outputs the winning coefficient $\mathbf{y}_k$ and the class index k.

The message is decoded using the same set of transformations at the decoder. The class index is used to choose the class for the inverse transformation and the resulting reconstructed image block $\hat{\mathbf{x}}$ is calculated.

When comparing the McMEC network and the OIAL network, two main differences are evident. First, the McMEC network has only one basis vector per module while the OIAL has M. Secondly, the classifiers are different. With McMEC, there is no need to calculate the projected vector $\|\hat{\mathbf{x}}_i\|$ for each class since the classification is based only on

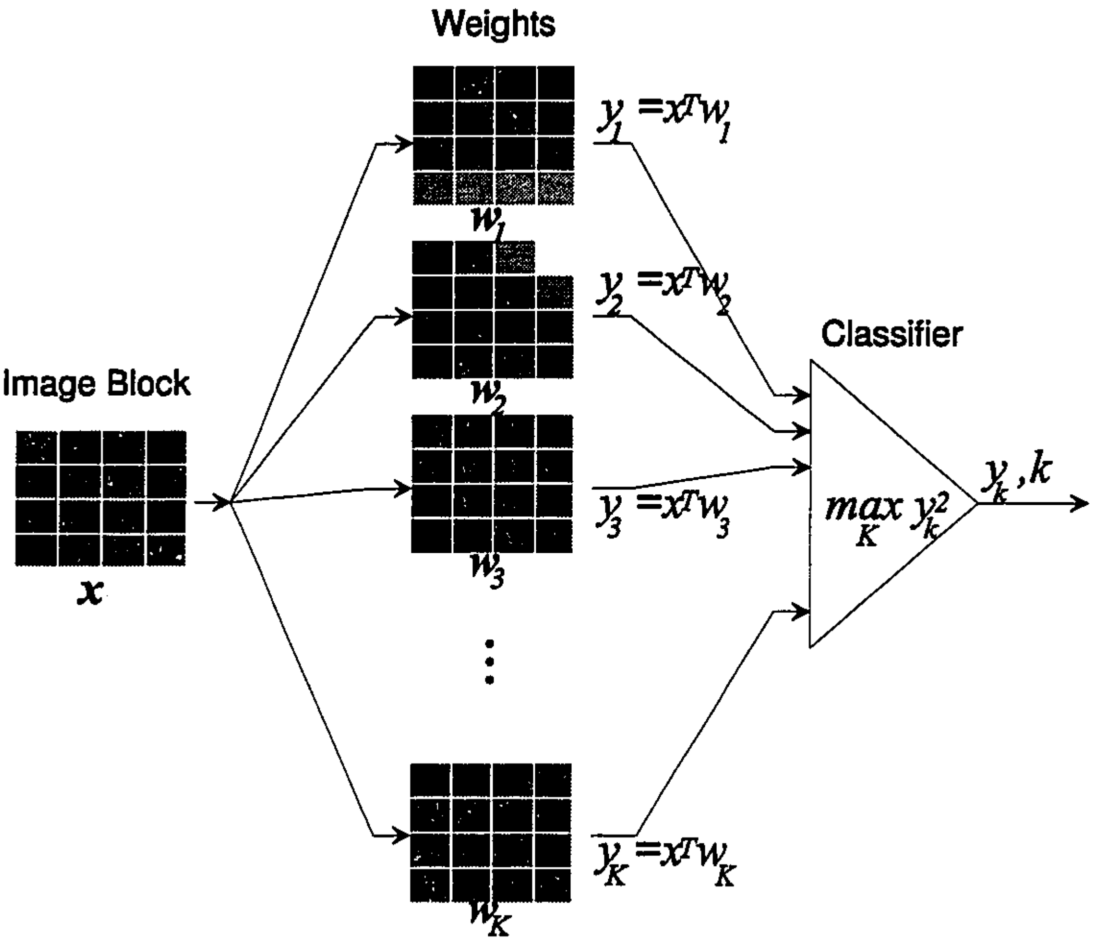


Figure 4.1: Modular architecture of McMEC. Input are blocks of $n \times n$ pixels. Each of the K transformation vectors $\mathbf{w}_i$ of size $n \times n$ outputs a coefficient y_i. The coefficient to be sent by the system is that with the largest magnitude as chosen by the maximum entropy classifier.

the magnitude of the transform coefficients. Both these differences are a result of the differences in the optimality criterion. However, as the discussion in the previous section has shown, in a sense, the McMEC network can be considered as a special case of the OIAL approach.

4.3 McMEC Learning

As described above, the optimal representation of a class in terms of maximizing the information preserved by the network is its principal components. With labeled data, the eigenvectors of the class covariance matrices can be calculated to find the principal component of each class. Similarly, iterative techniques such as those based on Hebbian learning or gradient descent can also be used to calculate the principal component.

Without labelled data, the problem of determining the appropriate classes and their respective principal components is akin to the problem of clustering in classical pattern recognition theory. The OIAL class of learning algorithms as developed in the previous chapter produces an optimal set of classes in a completely self-organizing manner. The resulting set of weights can be used to classify data outside of the training set. A variation on this approach will now be developed for the McMEC network which incorporates a topological ordering of the classes.

4.3.1 Network Topology

In some applications, it may be advantageous to have some similarity between "neighbouring" classes. Kohonen [33] introduced the concept of classes ordered in a "topological map" of features. In many clustering algorithms such as K-means or OIAL, each input vector $\mathbf{x}$ is classified and only the "winning" class is modified during each iteration. As discussed in section 2.3, in Kohonen's self-organizing feature map (SOFM), the vector $\mathbf{x}$ is used to update not only the winning class, but also its neighbouring classes. Each training vector $\mathbf{x}$ is classified according to the minimum Euclidean distance between it and the set of class feature vectors $\{\mathbf{m}_i\}$. The feature vectors of winning class and its neighbouring classes are modified according to their respective vector differences with respect to the input vector. The neighbourhood of a class is defined according to some distance measure on a topological ordering of the classes. For example, if the classes were ordered on a two-dimensional square grid, the neighbourhood of a class could be defined as the set of classes whose Euclidean

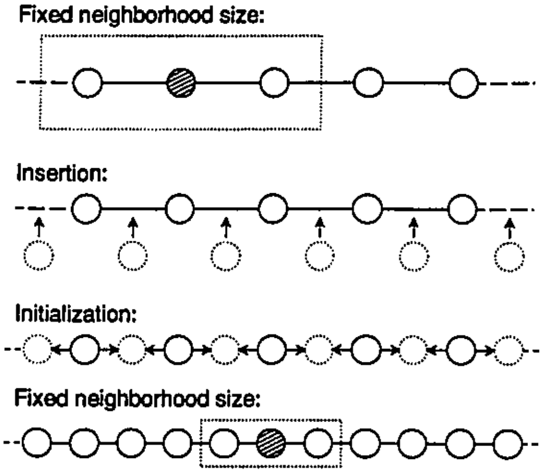


Figure 4.2: Method of network growing by insertion of new nodes.

distances from the class are less than some specified threshold. Initially, the neighbourhood may bequite large during training, *e.g.,* half the number of classes or larger. As the training progresses, the size of the neighbourhood shrinks until, eventually, it only includes the one class.

However, instead of starting with a large update neighbourhood that shrinks during training, the same topological ordering of features can be achieved by growing the network while fixing the neighbourhood size [42, 70, 23]. This results in significant computational savings. Initially the network consists of a small number of classes. Once the network has converged for a given stage, the number of classes is doubled by inserting new modules between the existing ones. The new weights are initialized to the mean of the neighbouring weights and the new network is retrained. These steps are illustrated in figure 4.2.

4.3.2 Training Algorithm

The learning algorithm for the McMEC network can be derived from the OIAL class of learning algorithms developed in the previous chapter. Two modifications are necessary: the classifier is modified from the subspace classifier to the maximum entropy classifier and the topological ordering of the classes is incorporated. The following learning algorithm results:

1. Set the initial number of classes and initialize the first set of weights $\mathbf{w}_i$.

2. Get the next input data vector $\mathbf{x}$ and calculate the coefficient

$$y_i = \mathbf{w}_i^T \mathbf{x} \tag{4.6}$$

   for each class weight $\mathbf{w}_i$.

3. Classify the input vector according to the maximum entropy classifier of equation 4.4.

4. Update the neighbours of the winning class C_k according to the rule:

$$\mathbf{w}_i = \begin{cases} \mathbf{w}_i + \alpha(y_i \mathbf{x} - y_i^2 \mathbf{w}_i) & C_i \in N(C_k) \\ \mathbf{w}_i & C_i \notin N(C_k) \end{cases} \tag{4.7}$$

   where α is a learning rate parameter such that $0 < \alpha < 1$, and $N(C_k)$ is the set of classes that are in the neighbourhood of the winning class C_k.

5. If the network has not converged, go to step 2.

6. If the number of classes K is less than the desired number, double the network by inserting new classes between the existing ones, initialize the new weights to the mean of their neighbours, then go to step 2.

Again, care must be taken in the choice of the initial values of the weights. However, based on comparisons between Kohonen's SOFM and the LBG algorithm [46], the incorporation of the topological ordering of the classes reduces the sensitivity of the algorithm to the initial conditions. In any case, a good initialization scheme would be to set the weights to the d.c. component with a small amount of random noise added to differentiate the classes. The choice of d.c. is a good approximation since that tends to be the dominant component for most images.

Another advantage of using the network growing approach during training is that all the intermediate sized networks are simply a by-product of the training algorithm and so are "free." Before the network is expanded in step number 6, the existing network can be saved and possibly used for coding when a fewer number of classes is desired. Further, if more classes are desired, an existing network may be used to initialize the system thereby saving the computational load of training up to that existing sized network.

Since the above learning rule with a neighbourhood size of one is equivalent to the OIAL algorithms presented in the previous chapter with the number of coefficients $M = 1$

the same optimality issues are valid. The use of Oja's rule in equation 4.7 extracts the principal component of each class which maximizes the variance of its coefficient. For a given set of class basis vectors, the maximum entropy classifier assures that the within-class variance is maximized. The combination of these two rules, both of which maximizes the variance of the class coefficients, ensures a minimum over-all bit rate for a given distortion. Equivalently, for a given bit rate, the distortion is minimized. Again, the issue of global *vs.* local optimality will not be dealt with formally in this thesis due to the mathematical intractability of the problem. However, in practice, the algorithm consistently converged to a useful result every time. In fact, the convergence behaviour was found to be better than that of the OIAL networks due to the incorporation of the topological ordering and the required convergance of only one component.

Suffice it to say that in practice, the convergence properties of the algorithm have been satisfactory.

4.4 Evaluation

4.4.1 Method

To evaluate the performance of the McMEC network, the same data which were used for training in the previous chapter, namely the MR image (figure 3.4a), were used to train the network. The training data consisted of randomly chosen 8×8 image blocks from the training image. The learning algorithm was that presented in section 4.3.2. The initial network size was chosen to be 4 and the weights were initialized to d.c. with a small amount of random variation added. The update neighbourhood was fixed at one, *i.e.*, $N(C_i) = \{C_i\}$. Each intermediate sized network was saved up to the final size of 2048 classes. Again, the learning parameter was calculated as equation 3.17 with $\gamma = 0.95$.

An ensemble average of 1000 learning curves for a network of 128 classes is shown in figure 4.3. As outlined in the learning algorithm of section 4.3.2, the network was initialized using the 64 class network. The figure shows that the initial mean squared error after the network was doubled is very close to that of the final network. Because the initial network incorporates the experiential knowledge gained over the course of training the smaller sized networks, the initial representation is already very close to the optimal representation. The training is essentially a fine tuning of the network. In this example, the first 10 000 iterations result in a 7% reduction in MSE while the remaining iterations reduce the final MSE by a

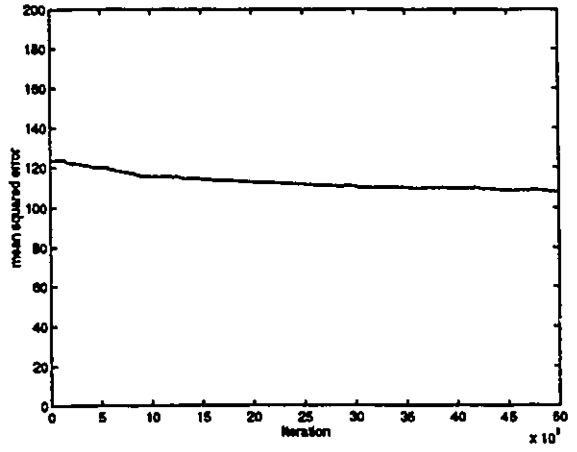


Figure 4.3: Ensemble average of learning curves for McMEC system with 128 classes.

further 7%. This demonstrates the efficiency of using the network growing approach during training when an existing smaller network is available.

For evaluation, the same test image, that shown in figure 3.4b, was used. The quantization and encoding of the resulting coefficients was modified to the approach used in the JPEG standard [73]. The single McMEC coefficient was coded via first order differential pulse-code modulation (DPCM) using a uniform quantizer. The quantized data was then Huffman coded with a codebook optimized for a Laplacian distribution. The coding of the class information was not optimized. The coefficients of the KLT for comparison were similarly coded using first-order DPCM fo: the first coefficient and pulse-code modulation (PCM) for the remaining coefficients with uniform quantization in all cases followed by Huffman coding. A number of quantization intervals were used.

4.4.2 Results

Figure 4.4 shows the error without quantization for the various sized network as a function of the additional number of bits per pixel required to code the class information for each block. As the figure shows, this relation is approximately linear with a slope of 35 dB per bit per pixel. For a network with 2048 classes and 64 pixels per block, the incurred bit rate for the class information is 0.17 bpp but the error reduces from 22.3 dB with no classes (the first component of the KLT) to 28.3 dB, a gain of 6 dB.

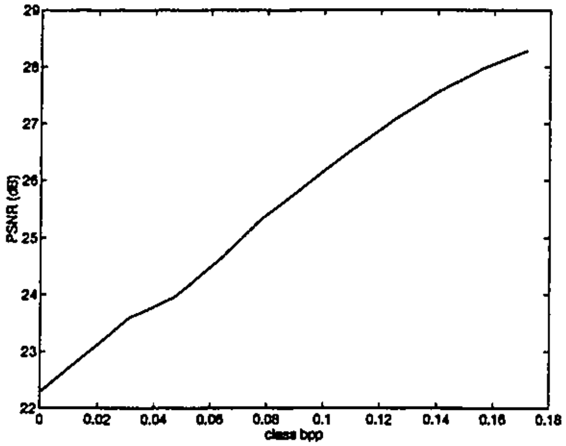


Figure 4.4: Non-quantized distortion *vs.* number of bits per pixel required to code the class information for each 8 × 8 block.

The experimental rate-distortion curves are shown in figure 4.5. The lines with individual points are the measured rate-distortion curves for various class sizes K of the McMEC system. The simple line is for the KLT trained on the same data. Each point represents a result for a specific quantization interval in the uniform quantizer. Table 4.1 shows an example of the rate-distortion data for the 512 class network. As would be expected, the number of bits per pixel decreases as the quantization interval increases. Increasing the quantization interval at fine quantization results in little decrease in PSNR. In this case, the dominant error is due to there being only one coefficient and the added quantization error is insignificant. As the quantization becomes progressively coarser, the over-all distortion begins to increase significantly.

As the number of classes increases, the performance of the system improves. The relation between the number of classes and the distortion as shown in figure 4.4 is also evident in figure 4.5. When the number of classes is small, the McMEC network performs significantly worse than the KLT. However, as the class size increases, the performance gap shrinks. For the network with 2048 classes, the performance is very close to that of the KLT. Considering that the McMEC network is coding the image using only one coefficient while the KLT uses up to 64 coefficients, the similarity in performance is remarkable. If this trend were to continue, then a network of 4096 classes would perform even better. However, as the network size increases, so does the coding complexity. This issue will be dealt with in more

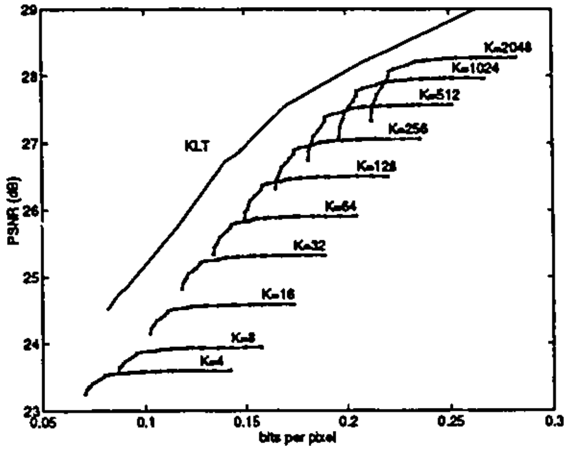


Figure 4.5: Rate-distortion curves for McMEC system. Lines with points are measured rate-distortion curves for various class sizes K while the simple solid line is the KLT.

Quantization Interval	bpp	PSNR
4	0.251	27.56
8	0.236	27.56
16	0.221	27.55
32	0.207	27.52
40	0.201	27.50
48	0.199	27.47
56	0.192	27.42
64	0.190	27.38
72	0.189	27.33
80	0.188	27.25
88	0.186	27.18
96	0.184	27.10
104	0.183	27.02
112	0.182	26.97
120	0.181	26.90
128	0.181	26.75

Table 4.1: Rate-distortion data for 512-class McMEC network.

detail later in this chapter.

Referring to table 4.1, at a quantization interval of 48 for the 512-class McMEC network, the coding rate is 0.199 bpp. This is a reduction by a factor of 32 over the gray level entropy of 6.36 bits per pixel for the testing image. Out of these bits, 0.141 are used to encode the class information while 0.058 bits are used to represent the coefficient values.

Again, while squared error is not without some merit as a performance measure, a qualitative comparison with the optimal nonadaptive KLT is still warranted. Figure 4.6a shows details of the result of coding the image of figure 3.4b with the McMEC network using a 2048 class system, quantized to 0.22 bpp, a compression ratio of 36:1. The resulting PSNR was 28.1 dB. Figure 4.6b shows the results using the KLT at the same bit rate; the PSNR was 28.3 dB. Even though the two measures of squared error are relatively close, the types of error are significantly different. The fidelity of edges is much better when the McMEC approach is used. This is particularly illustrated around the skull. The block effects of the KLT introduce a jaggedness to the line of the skull which is not as pronounced in the McMEC results. As well, both methods introduce a block-like texture in the brain tissue. It is interesting to note that while the squared error is higher for the McMEC approach, from a perceptual point of view, the fidelity of many features is actually better.

The improvement clearly justifies the use of adaptation. The KLT forms a representation based on an average over the entire image. As a result, the contribution from areas with strong edges and areas of no edge activity are all lumped together. When the KLT encodes edges at low bit rates, there is not enough information to adequately represent such regions due to the global nature of the transformation. On the other hand, with McMEC, each type of region within an image is optimally represented by its own class. Therefore, the coder can reconstruct each type of region with greater fidelity.

4.5 McMEC With Implied D.C. Component

4.5.1 Approach

The results of the previous section have shown that the coding performance of the McMEC method can approach or even surpass that of the KLT. However, the range of distortion levels for which this performance comparison is made may be too high for some applications.

To decrease the distortion with the KLT, the step size of the quantizer can simply be decreased. Of course, rate-distortion theory tells us that this will also result in more bits required to encode the image. Because of the nonlinear nature of the McMEC system, the relation between the coding rate and the distortion is not so simple. For a given network, a decrease in the quantization interval will decrease the distortion only up to a certain point. This upper bound in distortion is a result of the fact that the transformation for each class is not complete, *i.e.*, irreversible, since only one out of a possible N basis vectors are used. The omission of the remaining $N - 1$ basis vectors results in a "truncation error." The total distortion is then a sum of truncation error plus quantization error. So even with no quantization error, there will still be distortion.

One approach to reduce the distortion would be to add more classes. Figure 4.4 shows that an increase in the number of classes results in a decrease in the truncation error. This relation appears to be a simple linear function of the logarithm of the number of classes (class bits per pixel) and the logarithm of the distortion (PSNR in dB). However, the number of classes cannot grow without bound. To begin with, doubling the number of classes doubles the encoding time, not to mention the training time. Practical limitations such as processor speed and time requirements for encoding would place an upper bound on the network size. As well, the number of weights for any neural network-based approach is limited by the number of training data available [23]. A practical limit is given by

$$N > \frac{W}{\epsilon} \tag{4.8}$$

where N is the number of training samples, W is the number of weights, in this case the product of the number of classes and the block size, and ϵ is an accuracy parameter. Therefore, for a finite number of training samples and a given accuracy, the number of classes is limited.

A second approach would be to add more basis vectors to each class. In this case, the network becomes the OIAL network of chapter 3, where M is the number of basis vectors per class. The results in the previous chapter show that this approach does work for higher bit rates and therefore can lower distortion levels. However, the discussion at the start of this chapter points out that the adaptivity from a rate-distortion criterion may be suboptimal when $M > 1$.

Instead of, say, two unique basis vectors for each class, it may be possible to identify a component common to most classes. For most images, the strongest component, *i.e.*, the principal component, tends to be the d.c. component. When examining the class basis vectors created in section 4.4.1 it was found that there was a strong d.c. component in most of them. For the network of 512 classes, the angle between the average of all the basis vectors and the d.c. vector was only 3.6°. A third approach, then, would be to extract the d.c. component and encode it separately, while passing the residual through a McMEC encoder. The computation of the residual would not be necessary if the class basis vectors contained no d.c. component. This can be accomplished by explicitly removing the d.c. component from the input data during training. The resulting network is shown in figure 4.7. It is identical to that of the McMEC network shown in figure 4.1 except for the added calculation and output of the d.c. component of the input block.

4.5.2 Results

The training of the McMEC network with an implied d.c. component (IDC-McMEC) was performed in the same manner as described in section 4.4.1 except that the d.c. component of every training block was removed before presenting it to the network. Again, the image in figure 3.4b was used to evaluate the performance. For the rate-distortion evaluation, the d.c. coefficient was coded using DPCM with a uniform quantizer and the coefficient calculated by the network was coded using PCM with a uniform quantizer.

The rate-distortion curves for the resulting networks are shown in figure 4.8. The lines with individual points are the measured rate-distortion curves for various class sizes K of the IDC-McMEC system. The simple line is the KLT trained on the same data. Each point represents the result for a specific quantization interval in the uniform quantizer. When comparing the rate distortion curves for both the McMEC and IDC-McMEC networks, it is clear that the addition of the separate d.c. component allows for a decrease in squared error. Of course, with the decrease in distortion comes an increase in the bit rate. However, compared to the KLT, there is a marked improvement. For a network of 256 classes or larger, the IDC-McMEC approach out-performs the KLT in terms of rate-distortion.

Figure 4.9 shows details of the results of coding at 0.286 bpp (a compression ratio of 28:1) for a 2048-class IDC-McMEC network and the KLT. The PSNR for the former is 29.6 dB and 29.3 dB for the latter. As with the OIAL and McMEC networks, the IDC-McMEC results show better edge and line resolution than the results for the KLT. Again, this is most noticeable at the line defining the skull. Furthermore, some of the details in the folds of the brain are better preserved under the IDC-McMEC compression and the texture-like artifacts are reduced.

Generally, the use of the two-component network allows a greater fidelity on reconstruction compared with the single component network. In addition, the performance of the IDC-McMEC network relative to the KLT is much better with respect to the squared error distortion measure. While both the one- and two-component versions of the network preserve many of the finer structures of the images better than the KLT, the two component network would be preferred in applications where less distortion would be tolerated because of its improved performance.

4.6 Tree-Structured McMEC

4.6.1 Approach

The OIAL, McMEC, and IDC-McMEC networks all require an exhaustive search through the entire set of classes during coding to determine the optimal class for each input image block. As a result, the time or computational load required for encoding the signal varies linearly with the number of classes. If the number of classes were to double, the encoding of an image would take twice as long. As the network size becomes quite large, the time required for encoding may become prohibitive.

An effective technique for reducing search complexity is to organize the class modules in a tree structure as shown in figure 4.10 [5]. For an m-ary tree, each step in the search algorithm searches the current m nodes for the best match, and then continues the search on the m children of the winning node. This process continues until a leaf node, *i.e.*, a node without any children, is reached. For an m-ary tree with l levels, there are m^l leaf nodes in a fully balanced tree. For a McMEC network, each leaf node represents a final class with $K = m^l$. The use of a tree-structured search reduces the search complexity from K comparisons to $m \log_m(K)$ comparisons. Table 4.2 gives some examples of the required number of comparisons for a number of tree configurations. The table shows the dramatic savings in search time which the tree-structured approach affords.

For coding, the total number of nodes in an m-ary tree with K leaf nodes is given by

$$T = \frac{(K-1)m}{m-1} \tag{4.9}$$

Each node consists of a single basis vector. For each input vector $\mathbf{x}$, the m coefficients are calculated, one for each of the m nodes at the first level. The winning node is chosen by the maximum entropy classifier and the search moves down one level in the tree to the m children of the winning node. When the winning node is a leaf node, its coefficient is

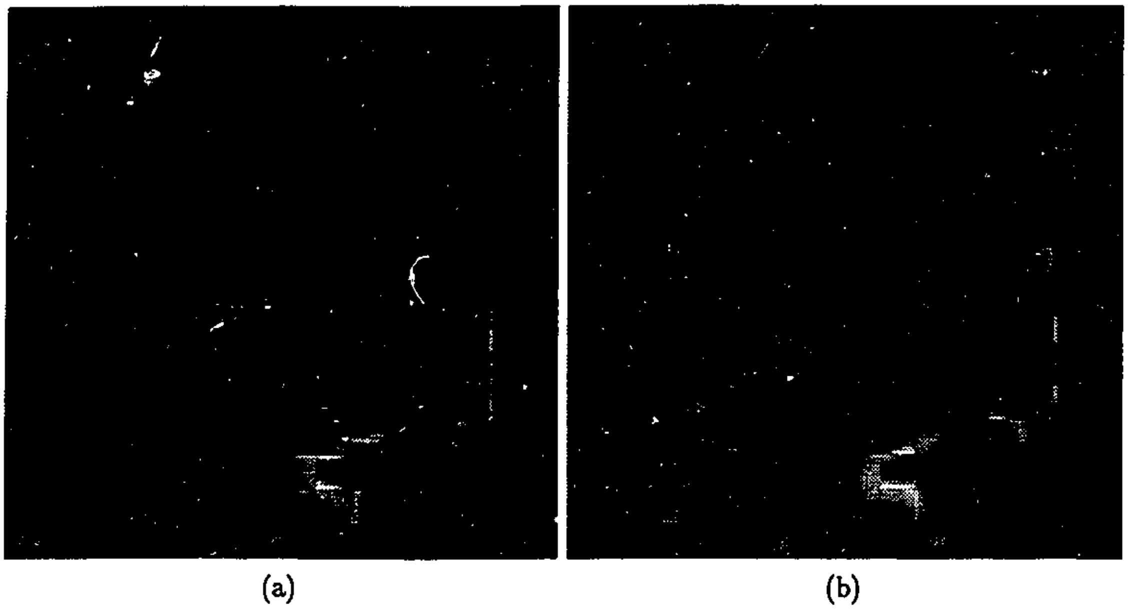


Figure 4.6: Details of coding at 0.22 bpp, (a) McMEC with 2048 classes, PSNR of 28.1 dB., (b) KLT, PSNR of 28.3 dB.

No. of Classes	Full Search	m=2 Tree	m=4 Tree	m=8 Tree
4	4	4	4	
8	8	6		8
16	16	8	8	
32	32	10		
64	64	12	12	16
128	128	14		
256	256	16	16	
512	512	18		24
1024	1024	20	20	
2048	2048	22		

Table 4.2: Number of comparisons for a tree-structured search for a number of tree configurations.

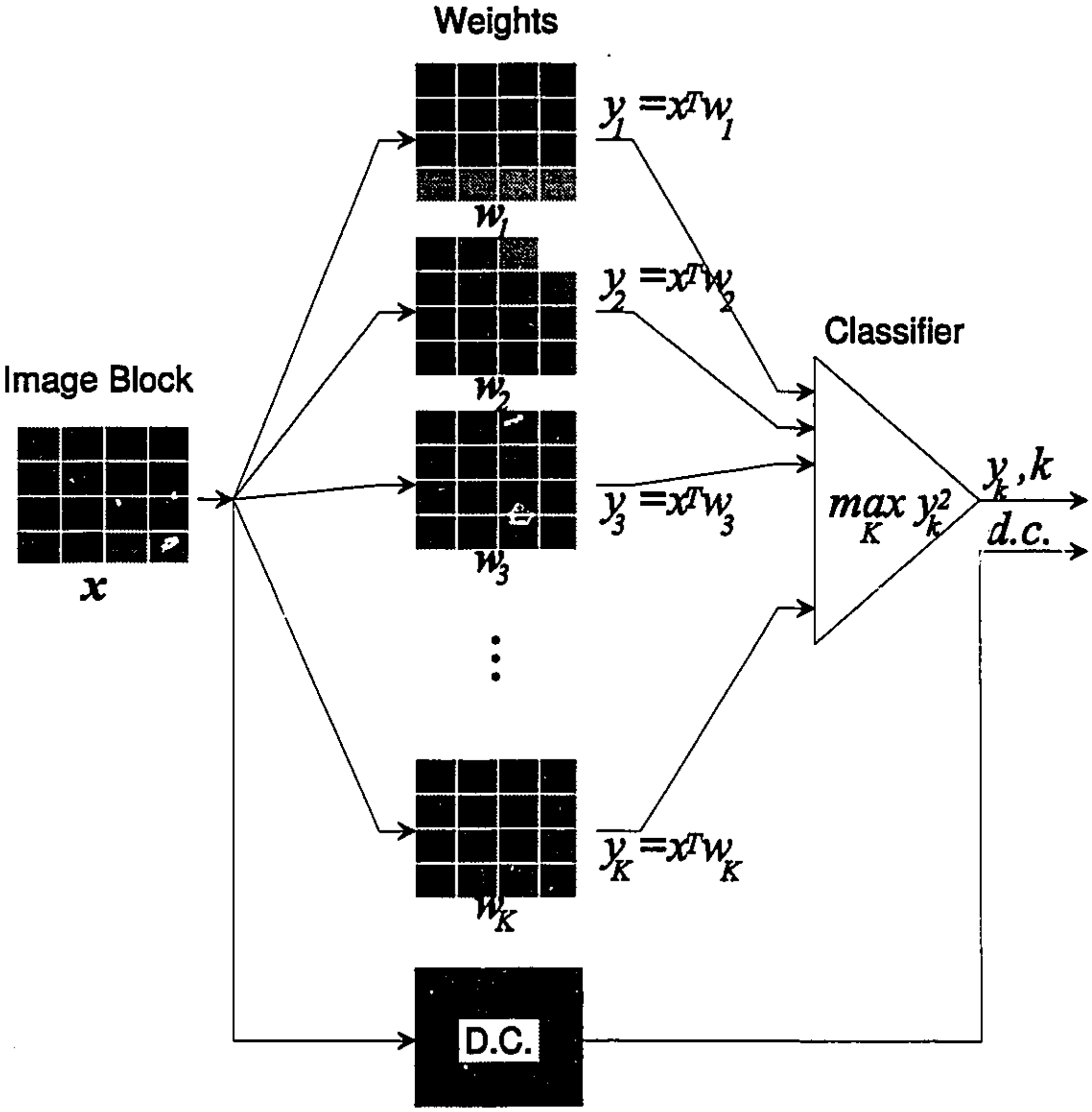


Figure 4.7: Architecture of Implied D.C. McMEC (IDC-McMEC) network.

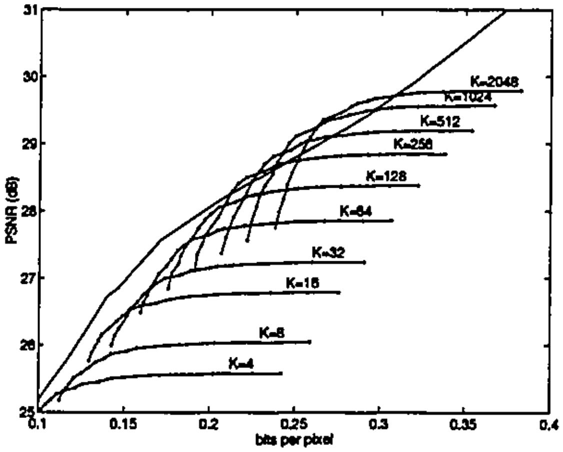
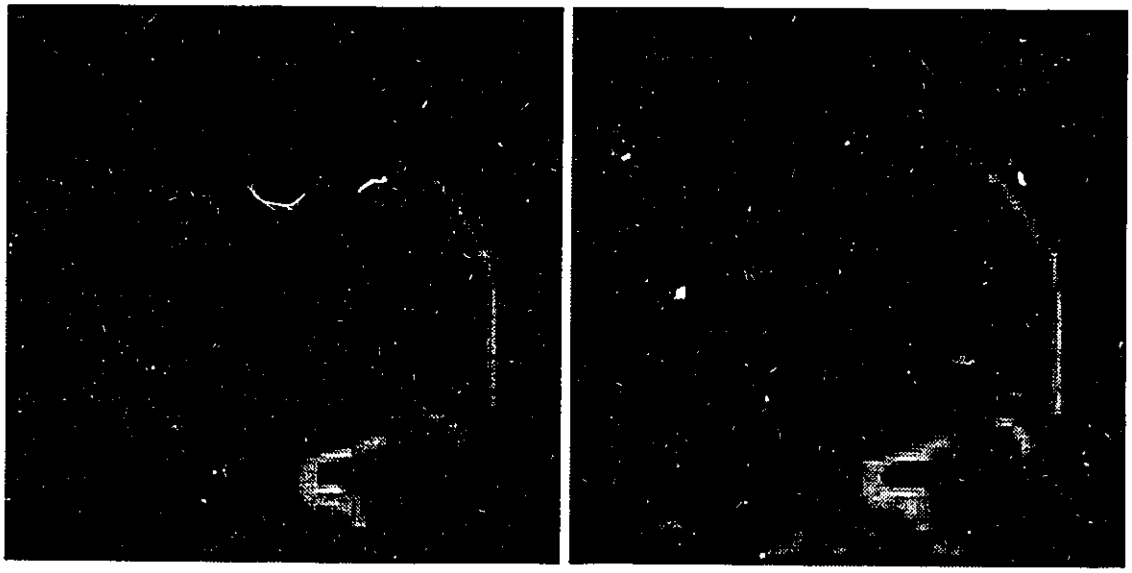


Figure 4.8: Rate-distortion curves for IDC-McMEC system. Lines with points are measured rate-distortion curves for various class sizes K while the simple solid line is KLT.



(a) (b)

Figure 4.9: Details of coding at 0.29 bpp for (a) IDC-McMEC with 2048 classes, with a PSNR of 29.6 dB and (b) KLT, with a PSNR of 29.3 dB

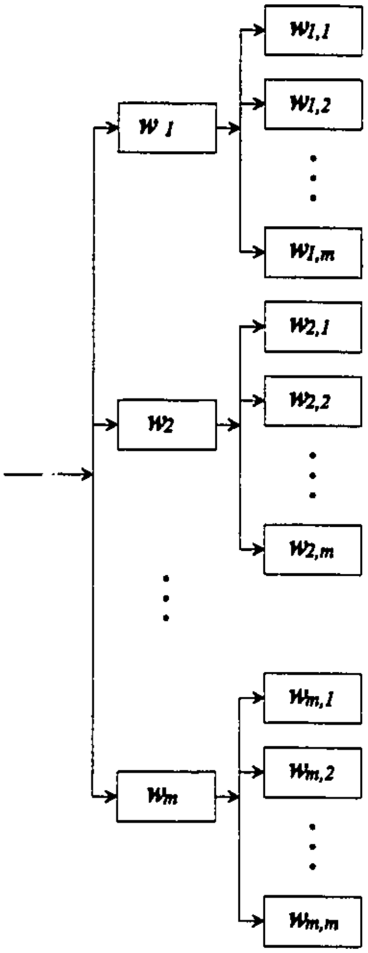


Figure 4.10: Architecture of Tree Structured McMEC network.

output by the encoder. Since there are K leaf nodes, there are K final classes. For decoding, only the K leaf nodes are required and the decoding proceeds in the same manner as the full-search approach.

4.6.2 Training

To train such a tree-structured network, a network growing approach may be used as illustrated in figure 4.11. Initially, the network consists of one level with m classes. During each stage in training, only the leaf nodes are modified. As each input vector is presented to the network, it is classified using the tree-structured search technique described above. The weights of the winning leaf node are modified using the Hebbian learning rule of equation 2.10. New input vectors are presented to the network until the weights in the leaf nodes converge. If another layer is required, the set of weights for each leaf node is duplicated m times to form the m children nodes. A small amount of random variation is then added to each set of new child's weights to differentiate the sibling nodes from each other. The training continues. This process is repeated until the desired network size is grown.

One of the advantages of this tree growing approach is that all the intermediate sized networks are available after training. If the final number of classes for an m-ary tree is $K = m^l$ where l is the final number of layers, networks of class size m^1, m^2, $\ldots$, and m^{l-1} are contained within the final network. For one of these smaller sized networks, the classification tree-search simply stops at the appropriate intermediate layer and the output is the coefficient of that layer's winning node.

If a larger network is required, the tree growing method can use an existing smaller network for initialization. In this manner, the new network already begins with a representation of all the experiential knowledge gained during the training of the smaller network. By initializing the leaf node weights to their parents' values, the same effect is achieved; the new weights are initialized to near optimum values. The training process becomes, in effect, a fine tuning of the weights.

4.6.3 Results

To evaluate the performance of the tree-structured McMEC (TS-McMEC) a number of network configurations were trained using the same training data as described in section 4.4.1. The single coefficient method was used, *i.e.*, the d.c. component was not removed as

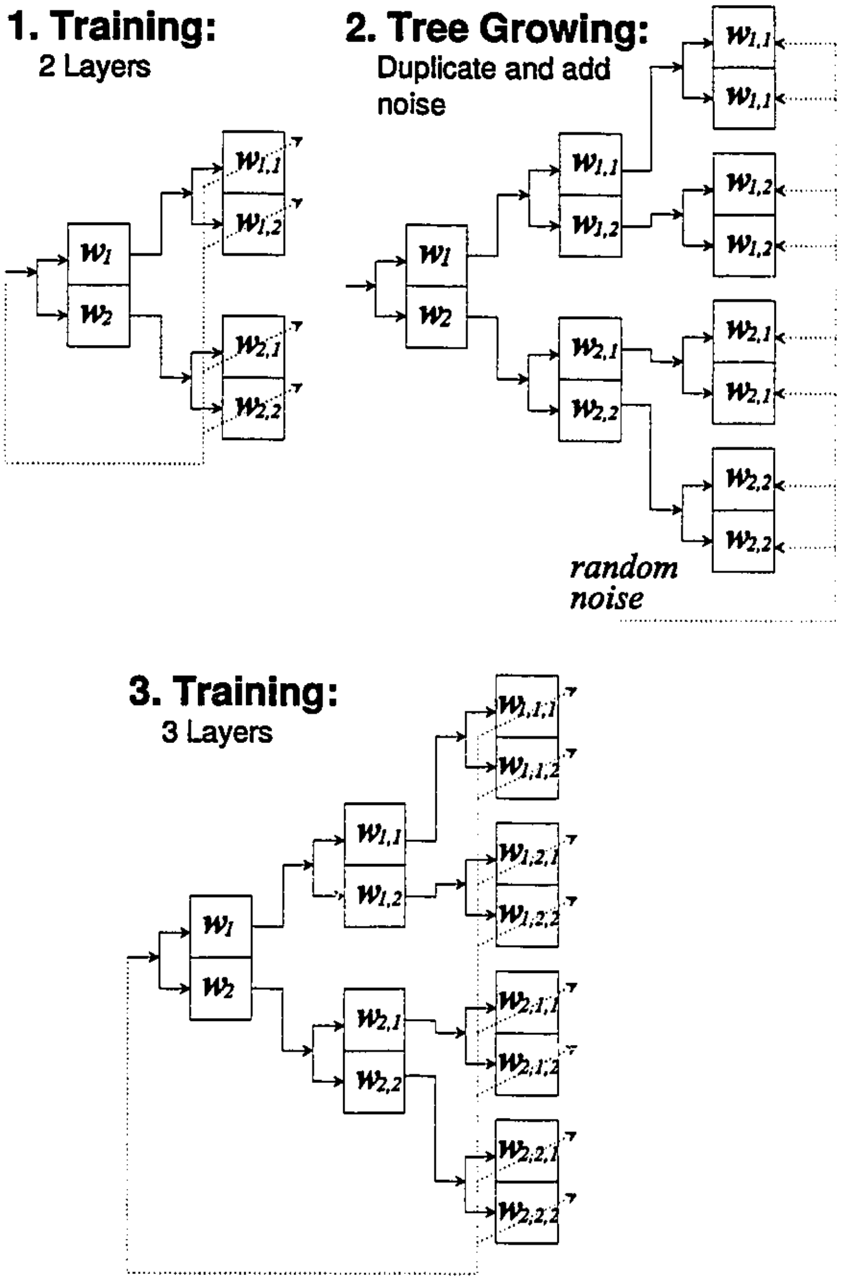


Figure 4.11: Training of a tree-structured network.

No. of Classes	Full Search	m=2 Tree	m=4 Tree	m=8 Tree
4	23.6	23.5	23.6	
8	23.9	24.2		23.9
16	24.6	24.7	24.8	
32	25.3	25.3		
64	25.9	25.8	26.1	26.4
128	26.5	26.2		
256	27.1	26.5	27.1	
512	27.6	26.8		27.8

Table 4.3: PSNR for TS-McMEC compared to full search McMEC without quantization.

in the IDC-McMEC method. Three types of tree architectures were trained: a binary tree (m=2), a quad tree (m=4), and an oct tree (m=8). The final number of leaf nodes were 512, 256, and 512 respectively. Except for the method of growing the network using the tree growing method as described above and the use of the tree-structured search for the final class, all the training parameters of section 4.4.1 for the full-search McMEC network training were used.

The performance of resulting networks were evaluated using the MR image shown in figure 3.4b. To see if the substantial reduction in search complexity incurs any additional distortion, a comparison of the error due to truncation is sufficient. Therefore, there is no need for an exhaustive comparison of the final quantized error rates, hence the coefficients were not quantized. The resulting PSNR is shown in table 4.3 along with the corresponding distortion measures for the standard full-search McMEC networks. In most cases, the use of the tree-structured network resulted in no significant difference in squared error. In fact, the error actually decreased for some cases.

The use of a tree-structured network can significantly reduce the encoding complexity of the McMEC technique. For a 512-class network, the number of comparisons to classify an input vector is reduced from 512 down to 18 for a binary tree or 24 for an oct ary tree. As the above data indicate, this substantial improvement incurs virtually no increase in distortion.

4.7 Computational Issues

One of the main reasons why the KLT has not found wide acceptance as a compression method is due to its computational complexity. For a block of N pixels, N coefficients may be calculated. For each coefficient, N multiplications and N additions are required. Therefore, $2N$ floating-point operations (flops) are required for each pixel. For an 8×8 block, the KLT requires 128 flops per pixel. For decoding, the same number of operations are also necessary.

Coding based on the DCT has gained widespread use due to the existence of fast algorithms for its computation. For example, based on the implementation presented in appendix A.2 of [57], an 8×8 forward DCT transformation requires 22 multiplications and 26 additions per block or equivalently 12 flops per pixel. The inverse DCT requires the same number of operations. This computational savings by almost a factor of 10 comes with a minimal loss in coding efficiency. It is well known that the DCT is asymptotically equivalent to the KLT for a first-order Markov process [57]. Since this model is a good approximation for most images, this means that the basis vectors of the DCT and KLT are similar and therefore their performance is similar.

Figure 4.12 illustrates this similarity for the MR images previously used. The solid line is the rate-distortion curve for the KLT based on the estimates of the covariance matrix from the training image in figure 3.4a. The KLT was tested on the image shown in figure 3.4b. The dashed line is the rate-distortion curve for the DCT on the same image (figure 3.4b). The figure clearly shows that the performance of the two transformations is almost identical.

An analysis of the computational complexity of the McMEC approach yields some interesting results. Because an image can be represented by substantially fewer components in an adaptive transform scheme than a standard block transform such as the KLT or DCT, fewer computations are required. For decoding the one coefficient McMEC network with a block size of N pixels, only N multiplications per block are required or equivalently, one flop per pixel. For the IDC-McMEC network, N new additions are required for decoding for a total of two flops per pixel. Therefore, the decoding complexity of the McMEC approach is $1/2N$ or $1/N$ that of the KLT. For the standard block size of 8×8, the use of the McMEC network affords a computational savings of $1/6$ or $1/12$ over the fast DCT at the decoder.

Both the KLT and the DCT are symmetric coders, *i.e.*, the same computational load is

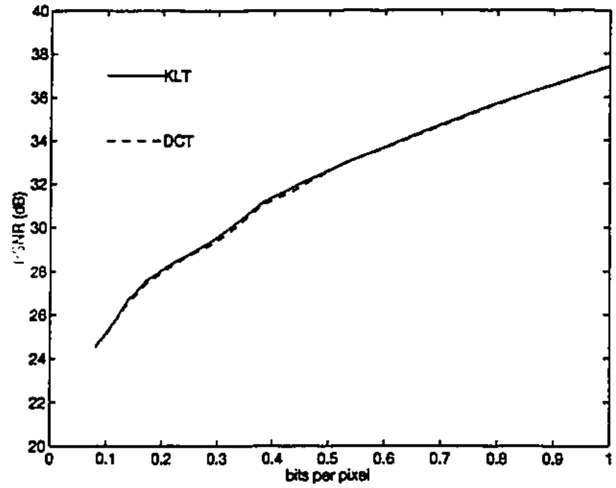


Figure 4.12: Comparison of coding performance of DCT and KLT.

required for both encoding and decoding an image. Because of the adaptive nature of the McMEC approach, it is an asymmetric coder due to the additional calculations required for classification at the encoder. For a full-search network with K classes, K separate coefficient calculations must be performed. For the one coefficient network, KN multiplications and KN additions per block or $2K$ flops per pixel are required for encoding an image. Since the d.c. component is calculated independently of the network on the implied d.c. approach, only N more additions per block are required for a total of $2K+1$ flops per pixel. For typical network sizes of 512 or more classes, the McMEC encoder incurs a substantial penalty in terms of computational complexity even compared to the "slow" KLT.

As the results of the previous section demonstrate however, a significant savings in computation can be realized through the use of a tree-structured network. For an m-ary tree of l levels with $K = m^l$ leaf nodes, ml coefficient calculations are required per input block for an average of $2ml$ flops per pixel. For an implied d.c. network, $2ml + 1$ flops per pixel are required. Table 4.4 shows the required number of floating-point operations per pixel for a number of network configurations with 512 classes and an 8×8 block size. Also shown are the equivalent number for the DCT and the KLT. The use of tree-structured networks substantially reduces the computations required for encoding with respect to a full-search network. Compared to the KLT, the McMEC network requires between 28% and

Method	Encode	Decode
Full-search McMEC	1024	1
Full-search IDC-McMEC	1025	2
Binary Tree McMEC	36	1
Binary Tree IDC-McMEC	37	2
Oct Tree McMEC	48	1
Oct Tree IDC-McMEC	49	2
KLT	128	128
DCT	12	12

Table 4.4: Number of floating-point operations per pixel for various McMEC network configurations of 512 classes, the DCT, and the KLT for an 8×8 block size.

38% fewer computations for encoding. However the encoder still requires more computations compared to the DCT, albeit only three to four times as many for this approach.

The calculations of the forward and inverse transformations, while having the greatest impact on the total complexity, represent just two parts of the entire encoding and decoding process. Once the coefficients are calculated, they must be encoded in an efficient manner as a set of discrete messages or series of bits. One of the common approaches, which was used in the experimental sections of this chapter, is to quantize the coefficients and then encode them in a lossless manner using, for example, Huffman coding. For a uniform quantizer, each coefficient must be divided by the quantization interval and then rounded. For the Huffman encoder, a look-up table operation is performed to encode the quantized coefficient based on a pre-calculated set of codes. Since these two operations must be performed for each coefficient, the DCT and the KLT both require up to N of these operations, whereas the McMEC method requires only one or two. While the quantizer and lossless encoder may ignore a number of coefficients at higher compression ratios, a substantial number still must be processed at typical coding rates further adding to the encoder complexity.

Upon decoding, the message or bit stream must be decoded by a Huffman decoder and then multiplied by the quantization interval before the coefficients undergo the inverse transformation. The order of complexity of a Huffman decoder is proportional to the number of bits in the message. Therefore, the computational requirements of decoding the Huffman code would be similar for all the approaches, since the number of bits would be the same at the same compression ratio. However, because each coefficient needs to be multiplied by the quantization interval, the fact that the McMEC approach uses fewer coefficients means that there would be additional savings in computation when decoding the bit stream before the inverse transformation.

The computational savings of the McMEC method upon decoding are substantial. This approach, though, incurs an increase in computational complexity at the encoder. The added encoding complexity may or may not be significant depending on the type of application. For a one time, point-to-point transmission where an image is encoded, transmitted, and then decoded, the McMEC network would require three to four times the computations over-all than the DCT. However, if an image were to be encoded once and decoded many times, the extra complexity of the encoder would be more than offset by the reduced complexity of the decoder. This model of compressed image use is valid for a wide range of applications. One example is a database of archived images. An image may be encoded once, but many users may wish to view the image a number of times. In this case, the encoding time may be immaterial but the users would want the decoding of an image to be performed as quickly as possible. For example, a physician may tolerate only a second or two delay in reconstructing a previously archived radiologic image. In addition, the computational power available to the user may not be large, for example a personal computer. Any reduction in the computational requirement of decoding is then even more important.

4.8 Comparison Between OIAL and McMEC

While the McMEC approach developed in this chapter and the OIAL approach of the previous chapter are both cases of the MPC representation, there are some differences between the two which warrant comparison. To compare the performance of the two approaches, it is important that the comparison be performed on an equal basis. To begin with, the number of weights should be equal. It is obvious that a network with more weights has the potential to represent the data in a more complete fashion. As a result, a larger network may have an unfair advantage over a smaller one despite a difference in network configuration. Computational requirements should also be the same. All things being equal, a computationally more complex method has a potential edge over a less sophisticated approach.

Both the McMEC network of section 4.2.2 and the IDC-McMEC network of section 4.5 have K network weights for a K class network. An OIAL network with M components and K classes has $M \times K$ weights. Therefore, for the same number of weights, an OIAL

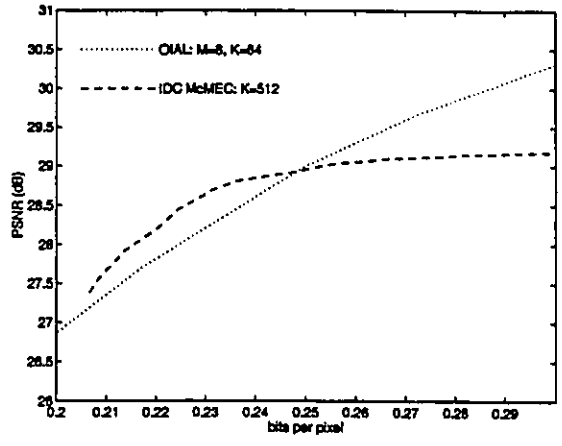


Figure 4.13: Rate-distortion comparison between OIAL and IDC-McMEC.

network will have fewer classes relative to a McMEC network. For the comparison, an IDC McMEC network with 512 classes will be used. For an OIAL network with 8 components, this means that only 64 classes are allowed to maintain the same number of weights (512).

Table 4.4 shows that the computational requirements of encoding for the 512 class IDC-McMEC network is 1025 flops per pixel. The requirements for the equivalently sized OIAL network is 1024 flops per pixel for the forward transformations plus an additional 16 flops per pixel to calculate the squared norm of the coefficient vector for classification. The norm of the coefficient vector is equivalent to the norm of the projected vector in the subspace classifier. This gives a total of 1040 flops per pixel for the OIAL network. Therefore, the encoding complexity of both methods are similar.

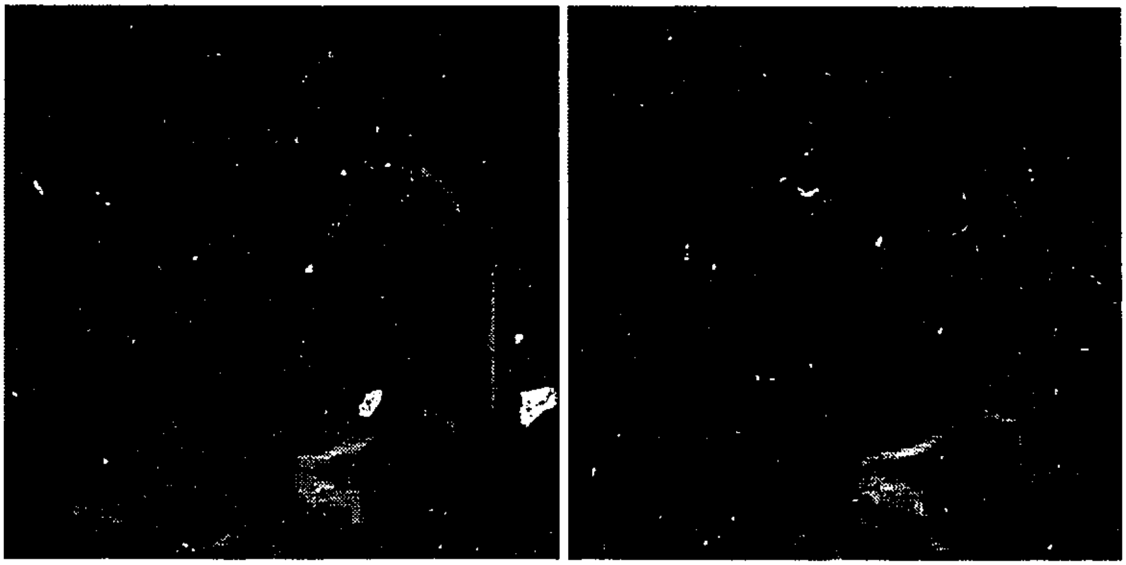
The 512-class IDC-McMEC network of section 4.5 and the 8-component, 64-class OIAL network of section 3.6 were used to compress the test image of figure 3.4b at a variety of bit rates. The resulting rate-distortion curves are shown in figure 4.13. The figure shows that the use of a single adaptive coefficient can result in reduced distortion over the use of multiple coefficients. This is consistent with the theory outlined in section 4.2. The use of a single coefficient results in the optimal adaptation which translates into improved coding performance.

The figure also shows that at higher bit rates, the multiple coefficient approach overtakes the single coefficient method. With only one adaptive component, the representation of the data is limited. Even neglecting quantization error, there may still be significant error in the representation. With multiple components, a more accurate representation is possible. This simply demonstrates the power of a multiple component approach to representation such as principal components. These results also illustrate the flexibility that the spectrum of representations using the MPC approach allows. At one extreme, a one-dimensional subspace allows for the optimal adaptation which improves performance at low bit rates. As the dimensionality increases, the adaptivity may decrease, but the representation becomes more faithful to the original data space.

To compare the nature of the distortion between the two networks, details of the compressed test image at identical rates (0.25 bpp) and squared error (PSNR of 29.0 dB) are shown in figure 4.14. When comparing the edges of the skull, the McMEC image tends to preserve the continuity of the lines and edges better than the OIAL image which displays more block effects in this vicinity. The OIAL method appears to introduce a large degree of texture-like distortion in the brain region which is not as noticeable in the McMEC version. However, some of the details around the orbit are reproduced better in the OIAL image.

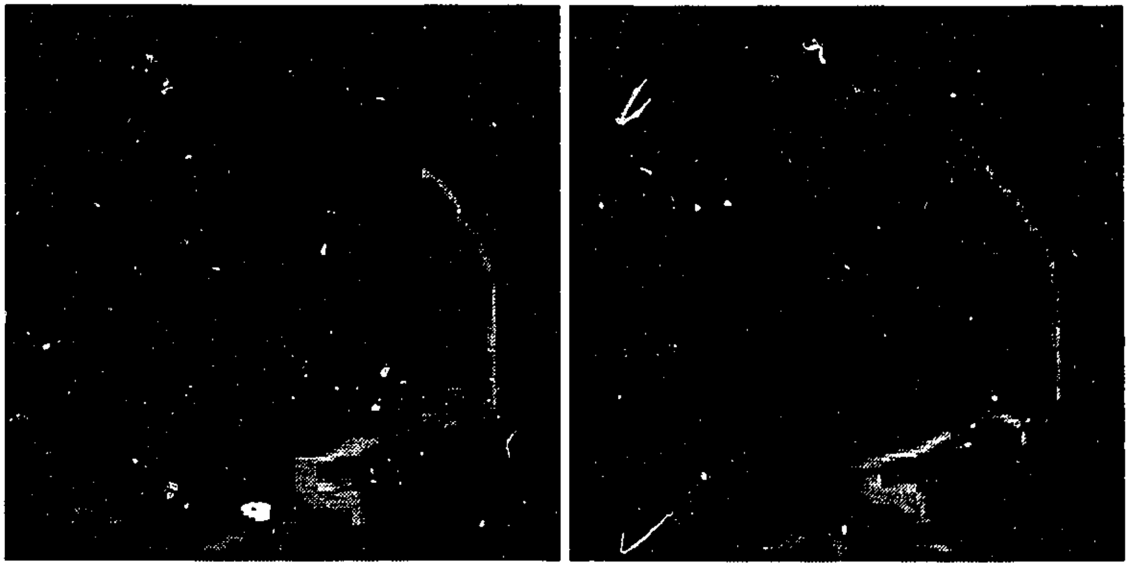
At a bit rate of 0.33 bpp, the squared error of the OIAL is significantly lower in comparison to the McMEC. For the OIAL, the PSNR is 31.0 dB while the McMEC has a PSNR of 29.2 dB. The details for the two methods at this rate are shown in figure 4.15. Comparing these results to those at the lower bit rate shown in figure 4.14, the quality of the McMEC image increases marginally with the greatest improvement evident in the brain region. The OIAL shows a greater degree of improvement with an increase in the bit rate, especially around the skull and the orbit. When comparing the two compression methods at the higher bit rate, a number of differences is evident. The well-defined edges and lines at the skull and orbit are better preserved in the OIAL version. For example, the dark line between the skull and the diploë is more distinct in the OIAL version than in the McMEC version. However, the texture-like distortion is still quite evident in the brain region of the OIAL image, while the McMEC image shows no such distortion. So, despite the increase in squared error, some regions are still reproduced more faithfully by the McMEC approach.

While this investigation has attempted to compare the two approaches on an equal footing, there remains a number of fundamental differences. The computing resources required for decoding are significantly different. For the IDC-McMEC, 2 flops per pixel are required while the decoding of an image using an 8-coefficient OIAL network requires 16 flops per pixel. The difference in training is also significant. The use of Oja's rule is a simple application of Hebbian learning to extract the first principal component from a data set. However,



(a) (b)

Figure 4.14: Details of coding at 0.25 bpp. (a) IDC-McMEC coding with 512 classes, PSNR of 29.0 dB. (b) OIAL coding with 8 coefficients, 64 classes, PSNR of 29.0 dB.



(a) (b)

Figure 4.15: Details of coding at 0.33 bpp. (a) IDC-McMEC coding with 512 classes, PSNR of 29.2 dB. (b) OIAL coding with 8 coefficients, 64 classes, PSNR of 31.0 dB.

the extension to multiple components is not so straightforward as evidenced by the number of approaches to this problem surveyed in section 2.2. The extraction of multiple components raises the issue of whether or not the components should be extracted simultaneously as with the GHA or sequentially as with APEX. The effect these two approaches have on the convergence rate or stability of training is an open issue.

Based on the above results and discussion, it can be concluded that, where possible, the one component McMEC or, even better, the IDC-McMEC is the preferred approach. In summary:

- The adaptation is optimal which results in better coding performance.
- For a given network size, the computational complexity is less — marginally so for the encoder and significantly so for the decoder.
- The complexity of the training is significantly reduced when only the first principal component needs to be extracted.

4.9 Summary

As chapter 3 has shown, the use of adaptation in coding images can allow better performance over the best nonadaptive coder. However, adaptation based on more than one basis vector per class may be suboptimal. From an information-theoretic point of view, the optimal adaptation occurs when there is only one basis vector per class, that being the principal component of the class. The maximum entropy classifier assigns a vector to the class whose basis vector calculates the largest magnitude coefficient. It acts as a feature detector, or equivalently, a matched filter. The use of the first principal component ensures that the information retained from the original is maximized. As well, maximum entropy classifier has computational advantages over the subspace classifier.

The performance of the McMEC network approaches that of the KLT. Considering that the McMEC network is encoding the image using only one coefficient while the KLT used up to 64, this shows that the network is indeed efficiently representing the information contained in the original image. Further, the network is better at preserving finer structures and details in an image due to its adaptive nature despite some relative increase in squared error.

It was noted that the basis vectors of every class of the McMEC network contained a significant d.c. component. To improve the coding fidelity, the d.c. component can be removed from the data and coded separately. This allows the network a greater latitude in adaptation. The resulting over-all performance of the IDC-McMEC network can surpass that of the KLT. Again, the network was shown to have less distortion in fine detailed regions of an image.

The McMEC networks have significant computational advantages over other block transforms upon decoding. For an 8 × 8 block size, the KLT requires up to 128 flops per pixel for decoding and the fast DCT requires 12. In contrast, the McMEC method requires 1 or 2 flops per pixel. On encoding, however, the adaptive nature of the McMEC network results in an increase in computations per pixel over the DCT. A tree-structured network can be employed to help improve the efficiency of the encoder with little if any cost in terms of increased distortion. Compared with the KLT, only 28% to 38% of the computations are required for encoding, while three to four times the computations are required relative to the DCT. For many applications of image coding, an image is encoded once and decoded many times. In this environment, the use of the McMEC network would allow a substantial reduction in the computational requirements.

In comparing the McMEC approach to the OIAL of the previous chapter, a number of advantages are evident. First, the adaptation is optimal due to there being only one component per class. This optimal adaptation results in improved performance for a network of a fixed size. Secondly, the computational complexity for decoding is significantly less. Finally, the training complexity is also reduced.

Chapter 5

Application to Medical Imaging

While the two previous chapters have demonstrated in a somewhat convincing fashion the usefulness of the new approach to image compression, the evaluations presented were of limited scope. To fully explore and evaluate the usefulness of any technique, not just image compression, the technique must be applied to a legitimate problem and its performance in that environment assessed.

One of the most demanding application areas for the use of image compression is the compression of medical images. The implications of introducing any sort of distortion in this class of image are grave. There are numerous legal and regulatory issues which are of concern because of this [73]. As a result, there is an argument for the use of lossless compression in this field. However, this approach is of limited usefulness due to the theoretical limits on the maximum allowable compression. The study of the use of lossy techniques to overcome this limit is therefore warranted.

For these reasons, the application of image compression to medical imaging was chosen as the environment in which a substantial evaluation of the new compression technique was conducted.

5.1 The Need for Image Compression

Within the medical imaging field, there continues to be tremendous growth in the availability and use of digital images. The number of imaging modalities in use includes such diverse methods as computed tomography (CT), digital subtraction angiography (DSA), digital flurography (DF), computed radiography (CR), magnetic resonance imaging (MRI),

Modality	Images per Exam	Image Resolution	Megabytes per Exam
CT	30	512 × 512 × 12 bits	16
MRI	50	256 × 256 × 12 bits	6.5
US	36	512 × 512 × 6 bits	9.5
PET	62	128 × 128 × 16 bits	2
SPECT	50	128 × 128 × 8 or 16 bits	0.8 or 1.6
DSA	20	1024 × 1024 × 8 bits	20
DF	15	1024 × 1024 × 8 bits	15
CR	2	2048 × 2048 × 10 bits	16

Table 5.1: Typical data volumes for digital examinations.

ultrasonography (US), nuclear medicine (NM), single photon emission computerized tomography (SPECT), and positron emission tomography (PET). With this increasing use of digital imaging comes the need to handle larger volumes of digital data. It is estimated that in the United States alone the volume of digital image data being captured per year is on the order of petabytes (*i.e.,* 10^{15} bytes) [73]. Even for a single digital radiology department in a 1500-bed university hospital, approximately 2.5 terabytes of data are produced per year [73]. Table 5.1 illustrates some of the typical data volumes generated by the various digital diagnostic imaging techniques [73].

Even with gigabyte storage media, the archiving of such data volumes still poses a problem. Currently, optical jukeboxes have a top storage capacity on the order of 100 gigabytes. While this storage medium allows for relatively fast access, *i.e.,* seconds, a jukebox could be full in a matter of weeks. For longer term storage, digital tape is commonly used. However, since the capacity of a single tape is only 2 to 5 gigabytes, a large number of tapes, on the order of one hundred per year, must be created and manually archived.

The transmission time of such large data sets also becomes an issue. In many cases, 2 seconds is the maximum allowable time for the transmission and display of an image [73]. For a typical local area network (LAN) with a peak bit rate of 10 megabits per second (10 Mb/s) operating at an average 50% of the peak rate, the transmission of a single CR image would require 13 seconds. For teleradiology applications using a T1 link at 1.544 Mb/s, the delay would be much greater and if an Integrated Services Digital Network (ISDN) operating at 128 kb/s were used, such an image would require approximately 10 minutes to transmit.

It is obvious, then, that there is a need in the diagnostic imaging field for some type of image compression. Even if a modest 10:1 compression were achieved, an optical jukebox may be able to store an entire year's worth of image data. At this rate too, the transmission of a CR image over a LAN would take under 2 seconds and over ISDN, the transmission time would be reduced to a more tolerable 1 minute.

The question, of course, is how much compression can be achieved? For "lossy" image compression methods, this is the same as asking how much distortion can be introduced in the reconstructed image? If absolutely no distortion can be tolerated, then "lossless" compression must be employed. However, since the maximum compression ratio for lossless compression is fundamentally upper-bounded by the entropy of the image, ratios of only 2:1 to 4:1 can be achieved [73]. Therefore, the usefulness of lossless compression is limited and the performance of lossy techniques must be investigated.

To answer the question of how much distortion is tolerable, the end-use of the images must properly be defined. For example, one of the main uses of medical images is for diagnosis. Diagnostic images must clearly show the critical features of a pathology to enable a diagnostician, typically a Radiologist, to identify correctly the pathology and its attributes. In addition, no spurious features should be present which could lead to a misdiagnosis. For a compressed image to be used in a diagnostic setting, its diagnostic value must not be degraded. The compression technique must preserve the features in an image upon which a diagnosis is made. As well, it must not introduce artifacts which could contribute to an incorrect diagnosis.

How can this distortion be measured? A common measure of distortion is mean squared error (MSE), or equivalently, peak-signal-to-noise-ratio (PSNR). However, as many researchers have rightly pointed out, despite its popularity as a distortion measure, MSE can be a poor indicator of the subjective quality of reconstruction [31, 49]. As a result, perceptually based criteria may be more appropriate. One example is the mean opinion score (MOS) [49]. A number of subjects view an image and rate its quality on a five-point scale of "bad," "poor," "fair," "good," or "excellent." The MOS is simply the average rating assigned by all the subjects. In the case of diagnosis, receiver operating characteristics (ROC) curves can be employed [15, 9]. ROC curves plot the relation between the probability of the correct detection of a pathology (P_d) and the probability of the false detection of a nonexistent pathology (a false alarm) (P_{fa}). As the distortion of the compression system increases, P_d will decrease for a given P_{fa}. Again, the method of evaluation will depend on the end-use being considered.

5.2 Application Area

The new image compression approach presented here will be applied to the educational use of medical images. Currently, Radiology residents acquire their diagnostic skills through the examination of actual clinical images containing various pathologies as well as normal images. Typically, these images are on film and are stored in a library of images. With the growth in digital imaging, it is now possible to store such a library of images digitally in a computer database. The residents would then be free to call up any of the images and display them on a suitable cathode-ray tube (CRT) display and examine them at their convenience. The database may reside in a central location on a network or it may be put on a CD-ROM for portability and at-home study on a personal computer. Such flexibility is particularly well suited to a problem-based learning environment. In addition, a number of imaging modes may be combined to allow the residents to compare the different types of examinations. With the addition of three-dimensional imaging and three-dimensional anatomical models, it would be possible to allow a resident to interactively perform any number of "virtual" examinations.

The evaluation criteria for this environment are quite different from, say, a diagnostic environment. In the educational environment, the diagnosis or pathology is given beforehand. It is sufficient that an image show clearly the pathology in question or the characteristics of a normal image. The diagnostic value is not the criterion by which the performance of an image compression technique should be evaluated. Instead, it is the over-all quality of the image and the visibility of the pathology as judged by an experienced Radiologist which must be measured.

Such a qualitative evaluation can be performed using the MOS method introduced in the previous section. While a simple judgement of "acceptable" or "unacceptable" would be adequate, a five-point scale of quality would provide more information with "excellent," "good," and "fair" being acceptable and "poor" and "bad" being unacceptable. A number of representative images is required for such a study. The images must include not only pathologies, but normal variations as well. The images would be compressed to a variety of degrees and the resulting images, including the originals, would be evaluated independently by a number of expert Radiologists. Of course, on presentation, the evaluators would not be told the degrees of compression or which image is the original. The highest compression ratio at which all the evaluations are rated to be "fair" or better, *i.e.*, acceptable, would determine the maximum allowable compression ratio for educational use for the examination type being evaluated.

5.3 Examination Type

For consistency, only one type of examination will be considered for the evaluation. The most demanding examination from a radiological imaging perspective is the chest radiograph. A chest image must contain a wide range of densities from the very unattenuating (dark) regions in the lungs to the very attenuating (light) region of the mediastinum containing the spine and heart. Within each region, there must be sufficient local contrast to show the detailed structures. In many cases, the features which a Radiologist looks for in a radiographmay not even be visible to the untrained eye. As well, a chest radiograph may contain a wide variety of pathologies ranging from the very subtle pnuemothorax to very obvious deformations. Therefore, the chest radiograph provides one of the most rigorous tests of any image compression scheme.

The fundamental principles of radiographic image formation have changed little since Roentgen produced the first X-ray images in the late Nineteenth Century. In an X-ray tube, as shown in the schematic of figure 5.1, electrons are emitted by the cathode's heated filament, are accelerated by the potential difference applied between the anode and the cathode, and strike the anode producing short-wavelength radiation referred to as X-rays. The spot on the anode on which the electron beam is focused is referred to as the focal spot and acts as a point source of X radiation. As the radiation passes through the subject, some of it is absorbed. The degree of absorption varies with the composition and density of the matter. For example, bones absorb more of the radiation than soft tissue. Behind the subject is placed a sheet of photographic film which is sensitive to X radiation. Upon development, the optical density of the film provides a record of the radiation to which the film has been exposed, and hence the degree to which the radiation has been attenuated in passing through the patient.

Computed radiography (CR) is an emerging imaging technique which produces digital images using X radiation [68, 41]. Instead of a photographic film as the recording medium, an imaging plate (IP) is used. When an X-ray photon strikes a phosphor molecule in the IP, this molecule undergoes a chemical change. The change causes the molecule to fluoresce when illuminated. The plate is "read" by a laser scanner which records the intensity of this

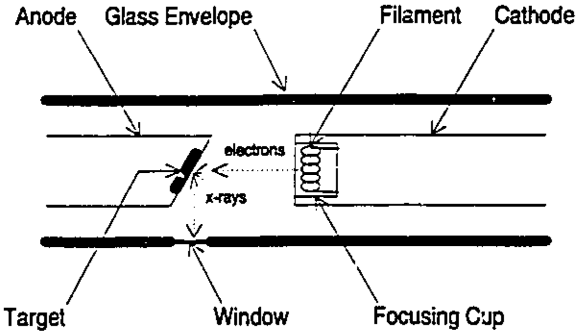


Figure 5.1: Schematic of an X-ray tube.

fluorescence and converts it to an array of digital pixel values.

CR has a number of advantages over conventional film-based radiography:

- The exposure-density relation is linear for a wide range of exposures. The dynamic range in X-ray dose of this linear response is on the order of 1 : 10^4. Because this latitude is much greater than that of photographic film, regions which would be over- or under-exposed on a conventional radiographic film would be faithfully recorded under CR.

- Since the raw data are digital, an image can be enhanced by a plethora of digital image processing techniques. These include global contrast enhancement, local contrast enhancement, edge enhancement, and other forms of digital filtering operations.

- The fact that the raw data are in digital form means that there is no possibility of introducing distortion during the digitization process. For example, a digital image obtained by digitizing an exposed photographic film will incorporate the nonlinear contrast response of the film. On the other hand, the response of the CR system is linear.

Because of the above reasons, digital CR chest radiographs were chosen to evaluate the new image compression method.

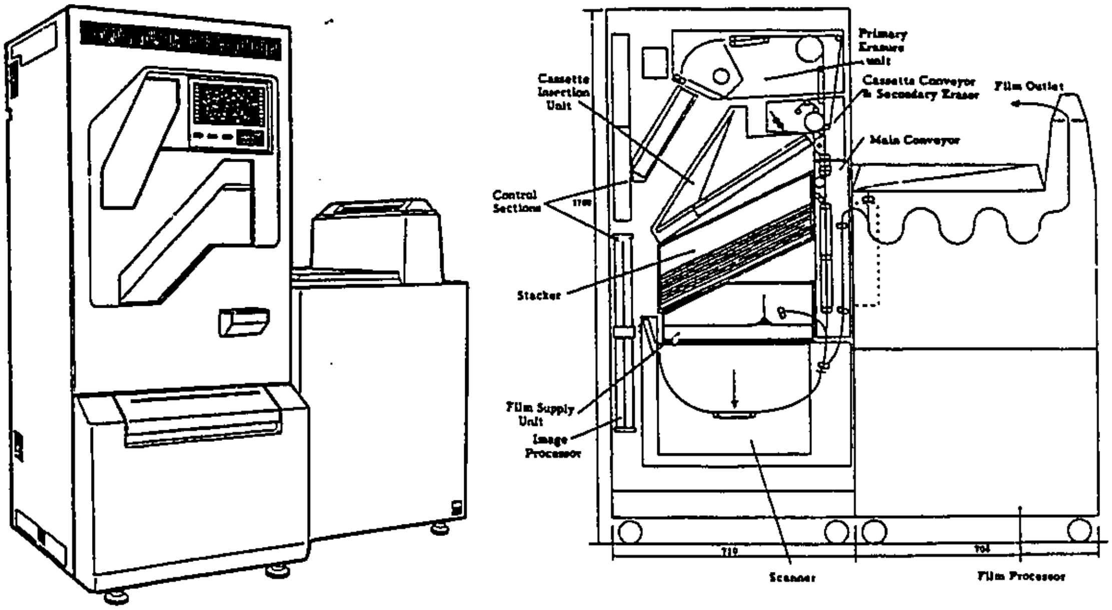


Figure 5.2: Diagram of the Fuji AC1 CR system (from [16]).

5.4 Materials and Method

All images used for this evaluation were clinical images generated by the Fuji AC1 CR system located in the Department of Radiology at the McMaster University Medical Centre (MUMC). A diagram of the system is shown in figure 5.2. The exposure is obtained in the standard manner with a conventional X-ray tube except that an IP is used as the recording medium instead of film. The IP is read and digitized by the CR system. The digital image is then transferred to a disk attached to a Sun workstation.

The images are nominally 2048 × 2048 pixels in size with 10 bits per pixel. The useful image area varies from image to image. The Medical Imaging and Network Development (MIND) Laboratory in the Department of Nuclear Medicine at MUMC has developed a program called MUMC Display which allows the viewing of these images in a number of formats with variable gray level mappings for contrast enhancement. The program also allows the printing of images to a digital film recorder.

5.4.1 Training

The two images chosen for training are shown in figures 5.3 and 5.4. These were chosen as "typical" chest radiographs. A number of IDC-McMEC networks were trained using these images in the same manner as described in sections 4.4.1 and 4.5.2 with the input blocks being randomly chosen from both images. The number of classes in the networks ranged from 2 to 1024 in steps of integer powers of 2. The following different block sizes were also used, 4 × 4, 5 × 5, 6 × 6, and 8 × 8. The variety of block sizes allowed a greater range of compression ratios to be investigated. The KLT was also generated from the same data for a single block size of 8 × 8.

For a preliminary evaluation, the rate-distortion curves for the various networks were experimentally found using a third representative chest image shown in figure 5.5. The coefficients were quantized in the same manner as described in sections 4.4.1 and 4.5.2, namely, first-order DPCM for the d.c. coefficient and PCM for the second coefficient with a uniform quantizer for both using a common quantization interval. Similarly, the rate-distortion curve for the KLT was also determined. The resulting curves are shown in figure 5.6. Again, these curves show that the McMEC networks can perform better than the KLT. The figure also shows the effect of varying the size of the image blocks. For a given number of classes, decreasing the block size reduces the distortion. This is simply because the input dimension is smaller and is therefore represented better by a set of subspaces of fixed dimension (*e.g.*, 2 for the IDC-McMEC).

5.4.2 Opinion Score Evaluation

For the evaluation, nine clinical chest radiographs were chosen by a senior Radiologist who specializes in chest radiology. This set was chosen to contain a variety of pathologies as well as normal images and images with high contrast objects such as surgical staples or jewelery. The original images are shown in figures 5.7 through 5.15. The characteristics of the images are summarized below.

- **Image 805.** Subsegmental atelectasis: bands of opacity, horizontal in image left lung and diagonal in image right lung.

- **Image 7731.** Interstitial disease and pleural fluid: diffuse, patchy opacities and edge created by pleural fluid in image left lung.

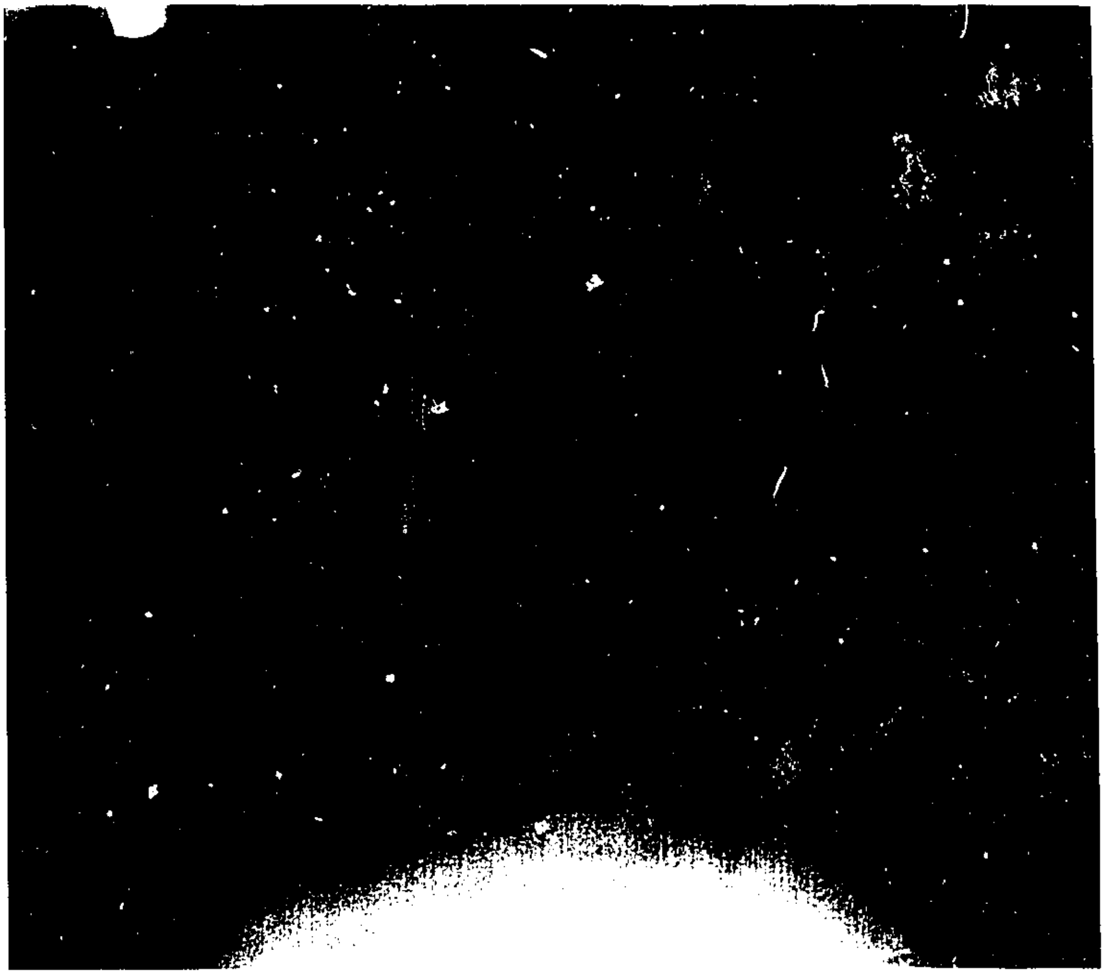


Figure 5.3: Chest 1. One of the two chest ,·diographs for training.

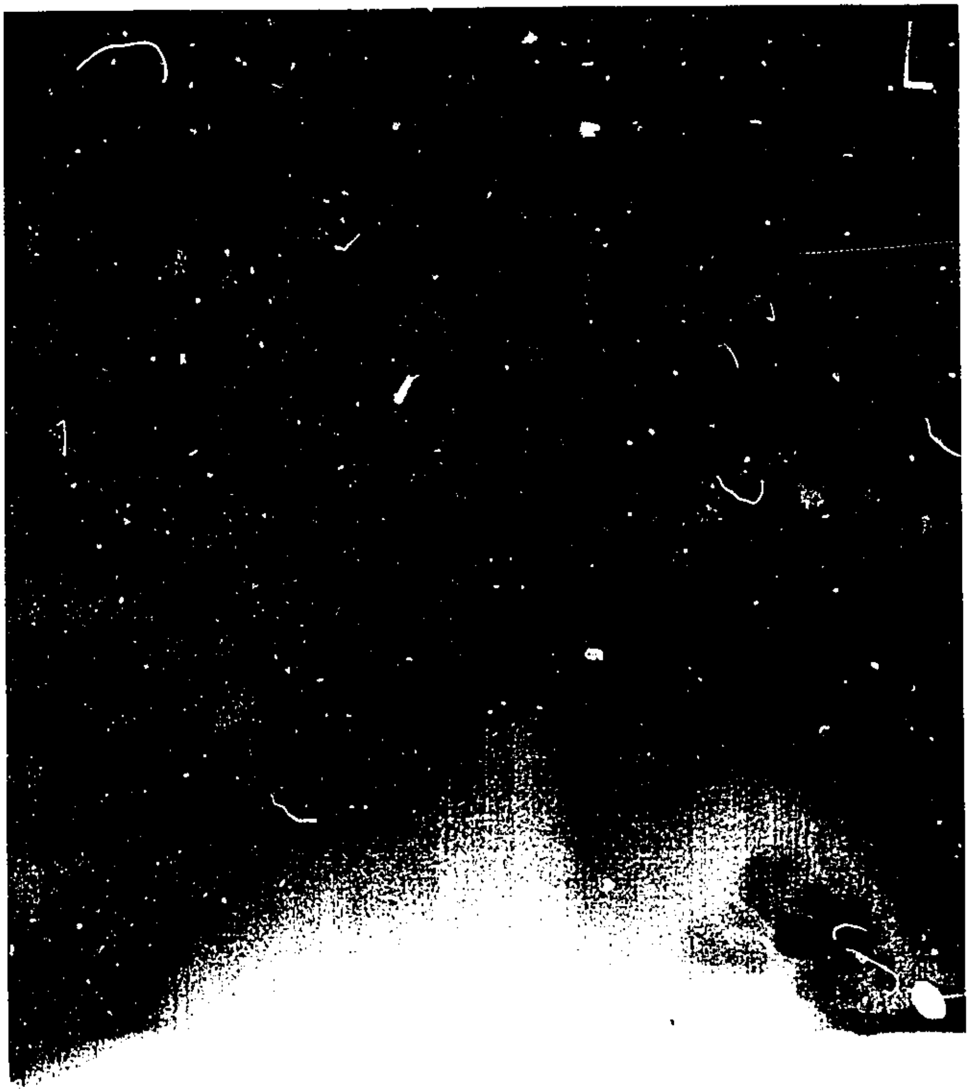


Figure 5.4: Chest 2. The other chest radiograph for training.

Figure 5.5: Chest 3. Chest radiograph for preliminary testing.

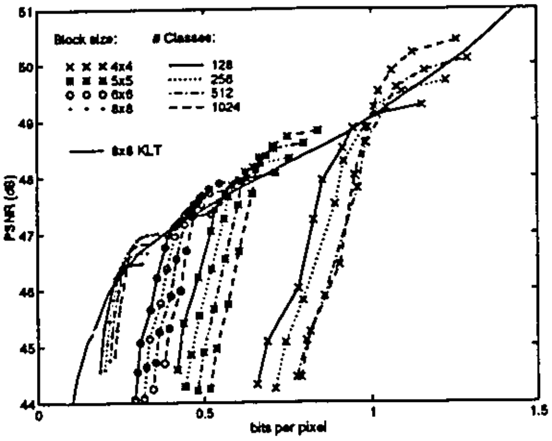


Figure 5.6: Rate-distortion curves for chest radiograph.

- **Image 9502.** Subtle pneumothorax: faint line in apex of image left lung.
- **Image 9789.** Normal, high contrast objects.
- **Image 10555.** Air space disease: white opacities in top of image left lung, and top and middle of image right lung.
- **Image 16916.** Normal: chosen to test for visualization of mediastinal line and edges.
- **Image 19158.** Interstitial and air space disease: diffuse, patchy opacities in image right lung
- **Image 26366.** Normal: chosen for support equipment and subdiaphramatic detail including contrast material in kidney.
- **Image 31159.** Image left upper lobe volume loss.

Each of the nine images was compressed at 10:1, 20:1, 30:1, and 40:1 relative to the useful image area coded at 10 bpp. The networks and the quantization intervals used for these ratios were chosen based on the data of the preliminary evaluation shown in figure 5.6. These parameters are summarized in table 5.2. As an example, figures 5.16 through 5.20 show the 512 × 512 details of a portion of image 16916 and the corresponding area at 10:1, 20:1, 30:1, and 40:1 compression, respectively.

Figure 5.7: Image 805.

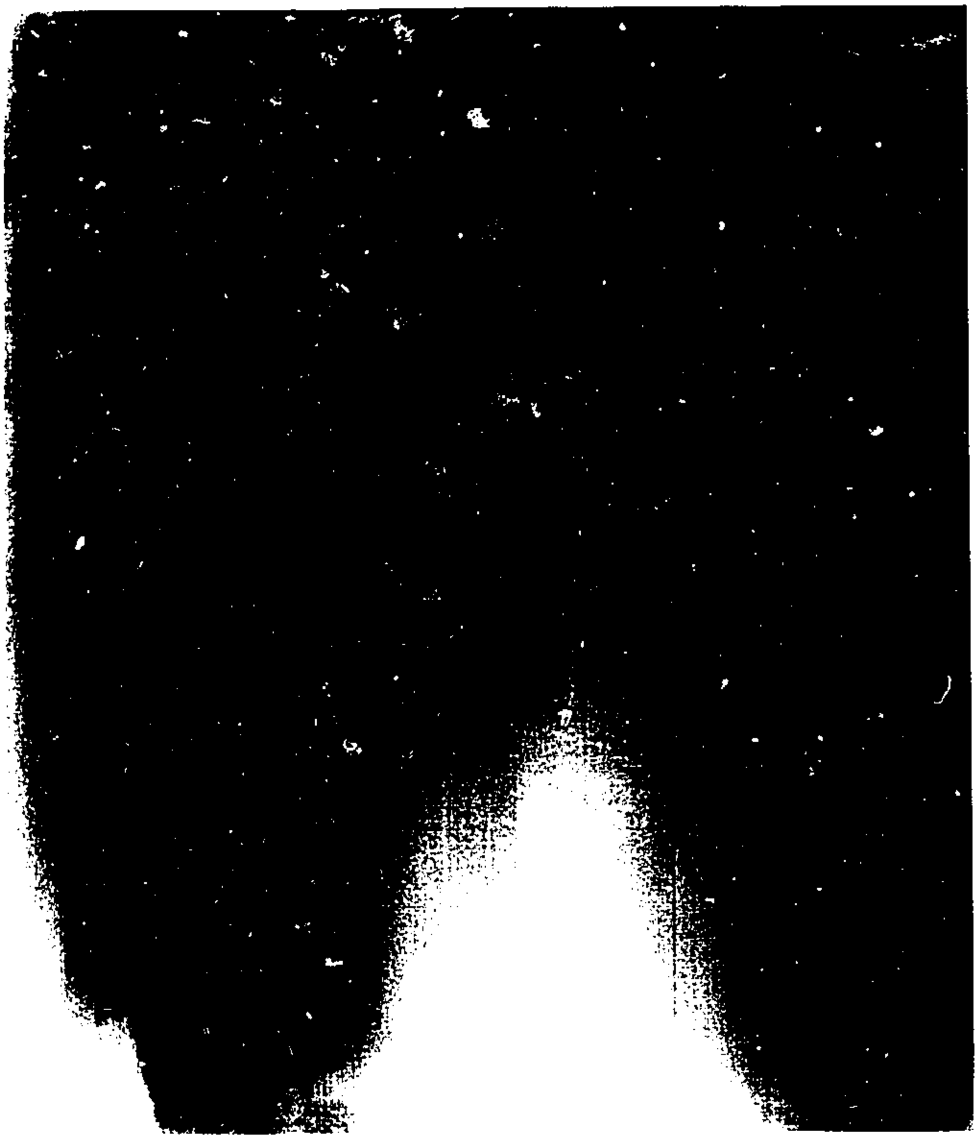


Figure 5.8: Image 7731.

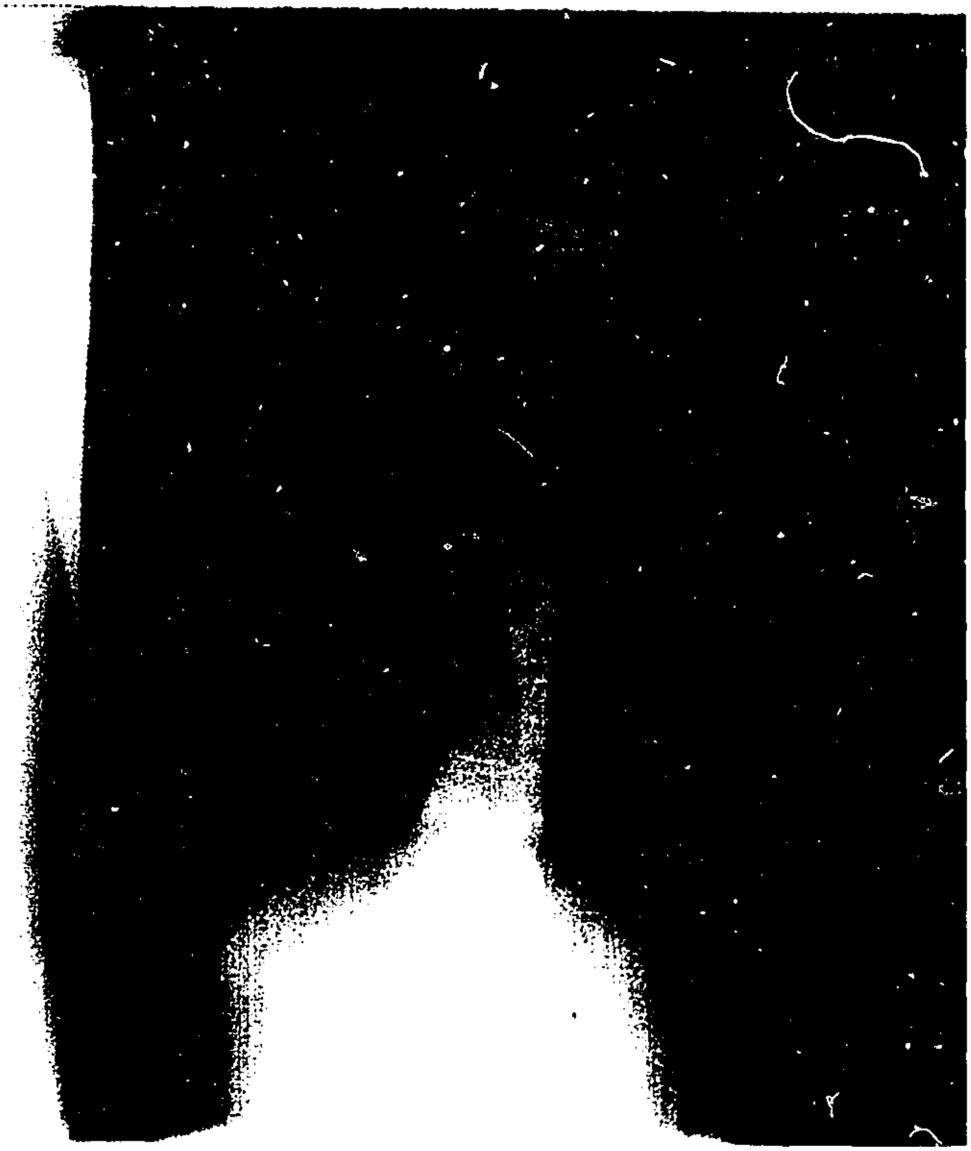


Figure 5.9: Image 9502.

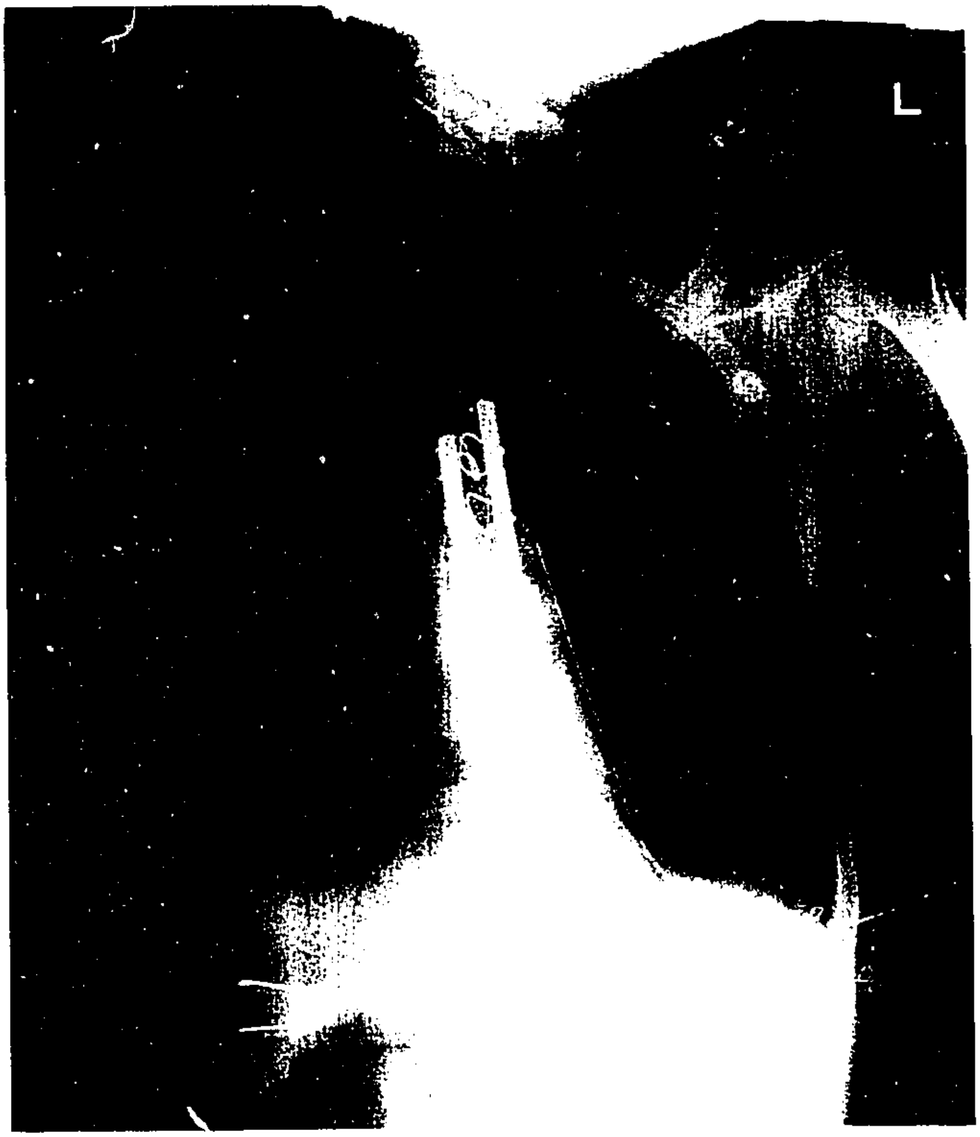


Figure 5.10: Image 9789.

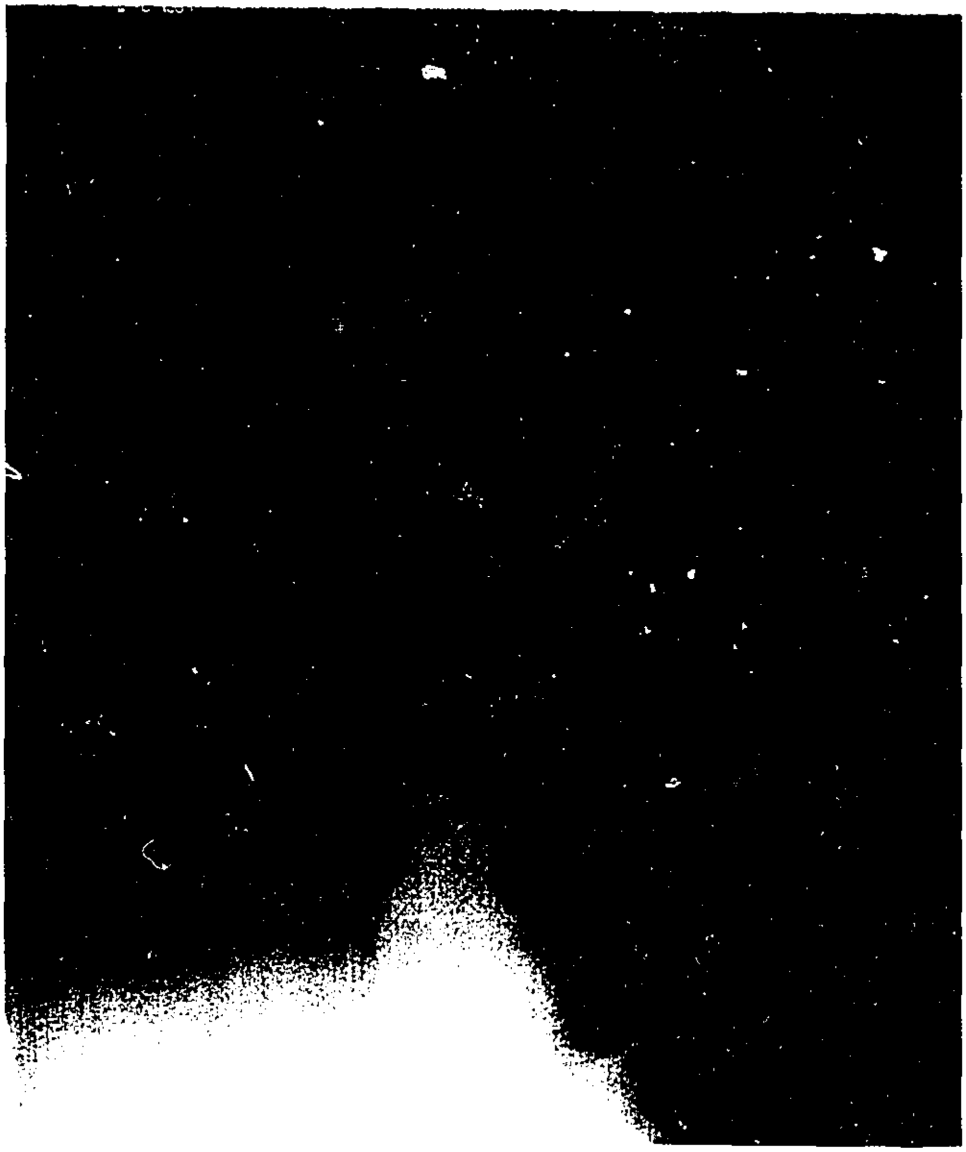


Figure 5.11: Image 10555.

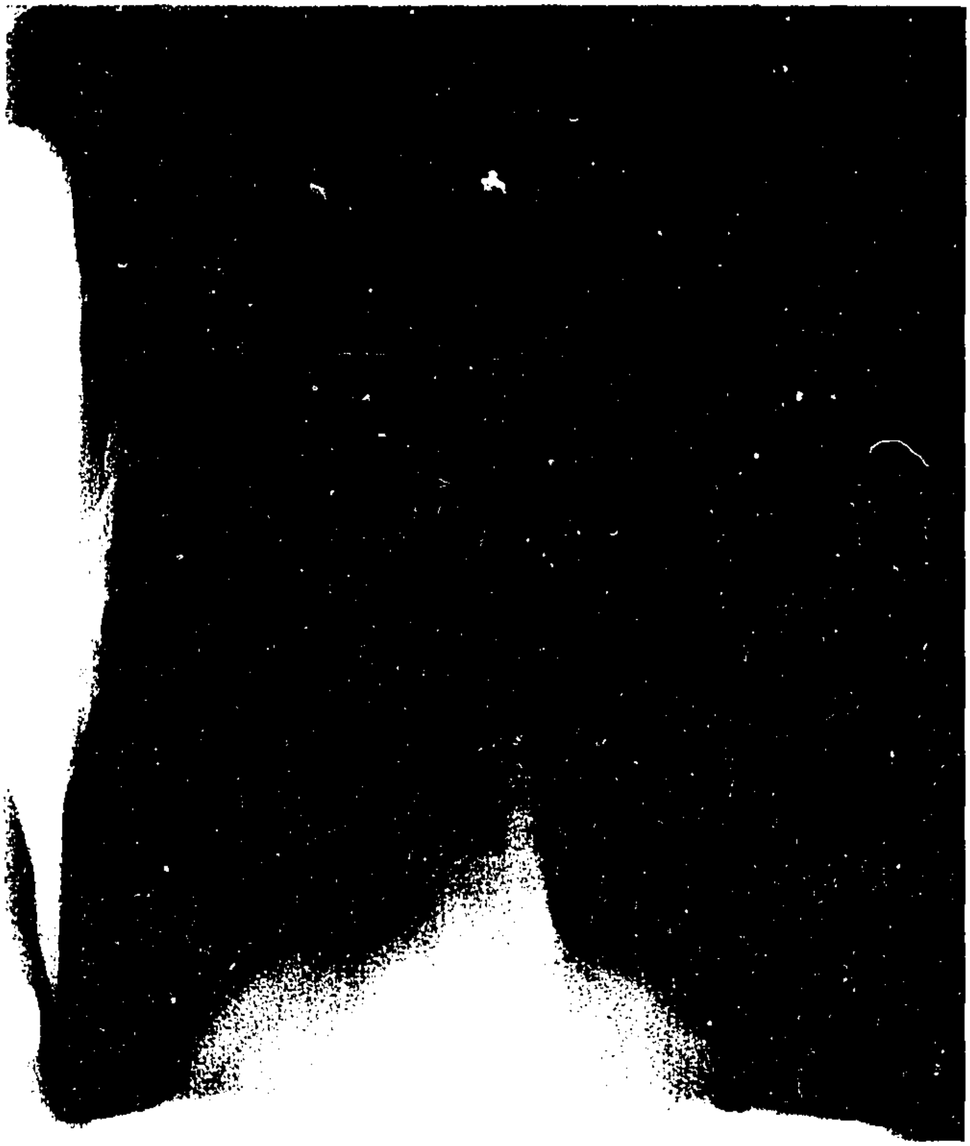


Figure 5.12: Image 16916.

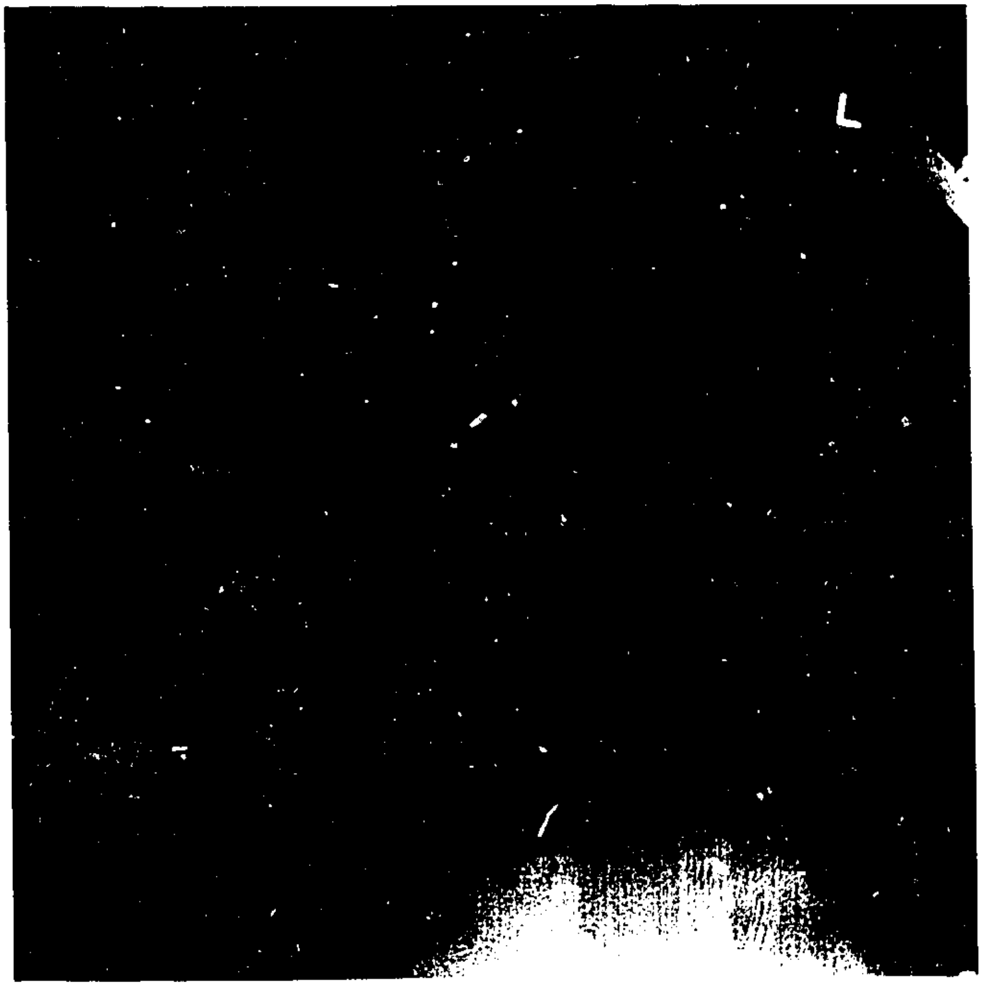


Figure 5.13: Image 19158.

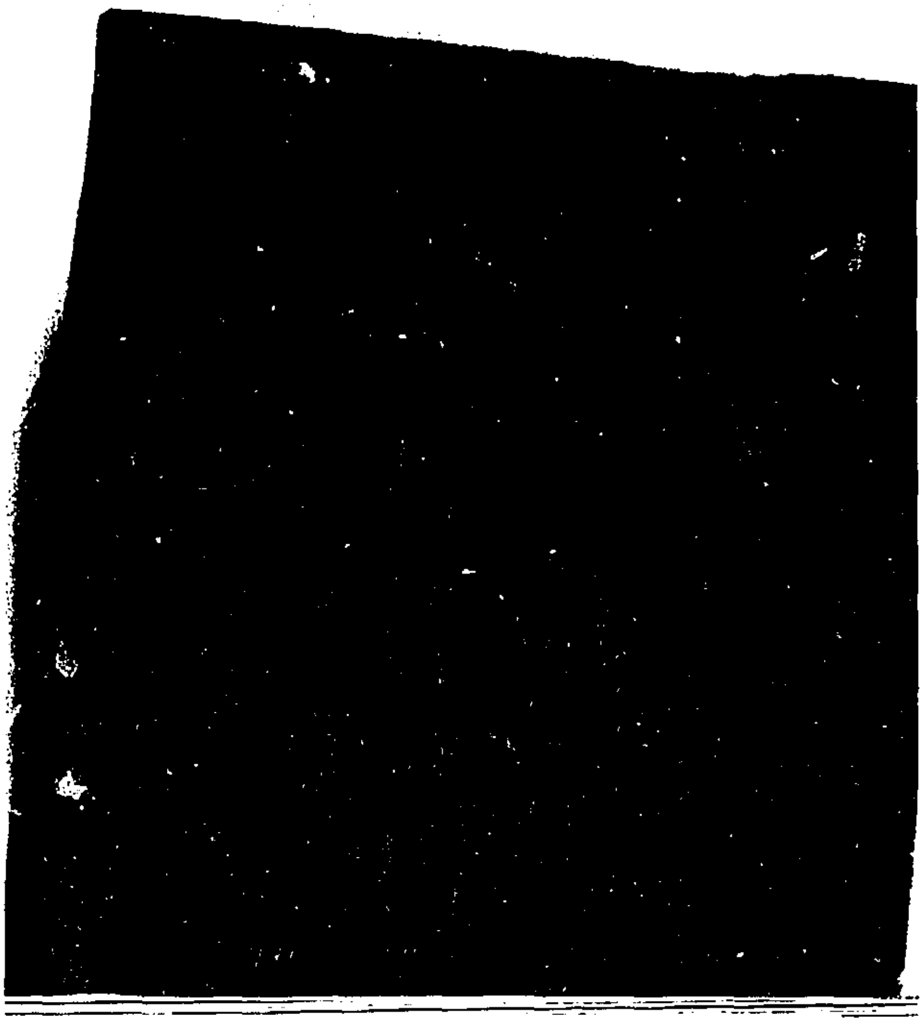


Figure 5.14: Image 26366.

Figure 5.15: Image 31159.

Figure 5.16: Details of image 16916.

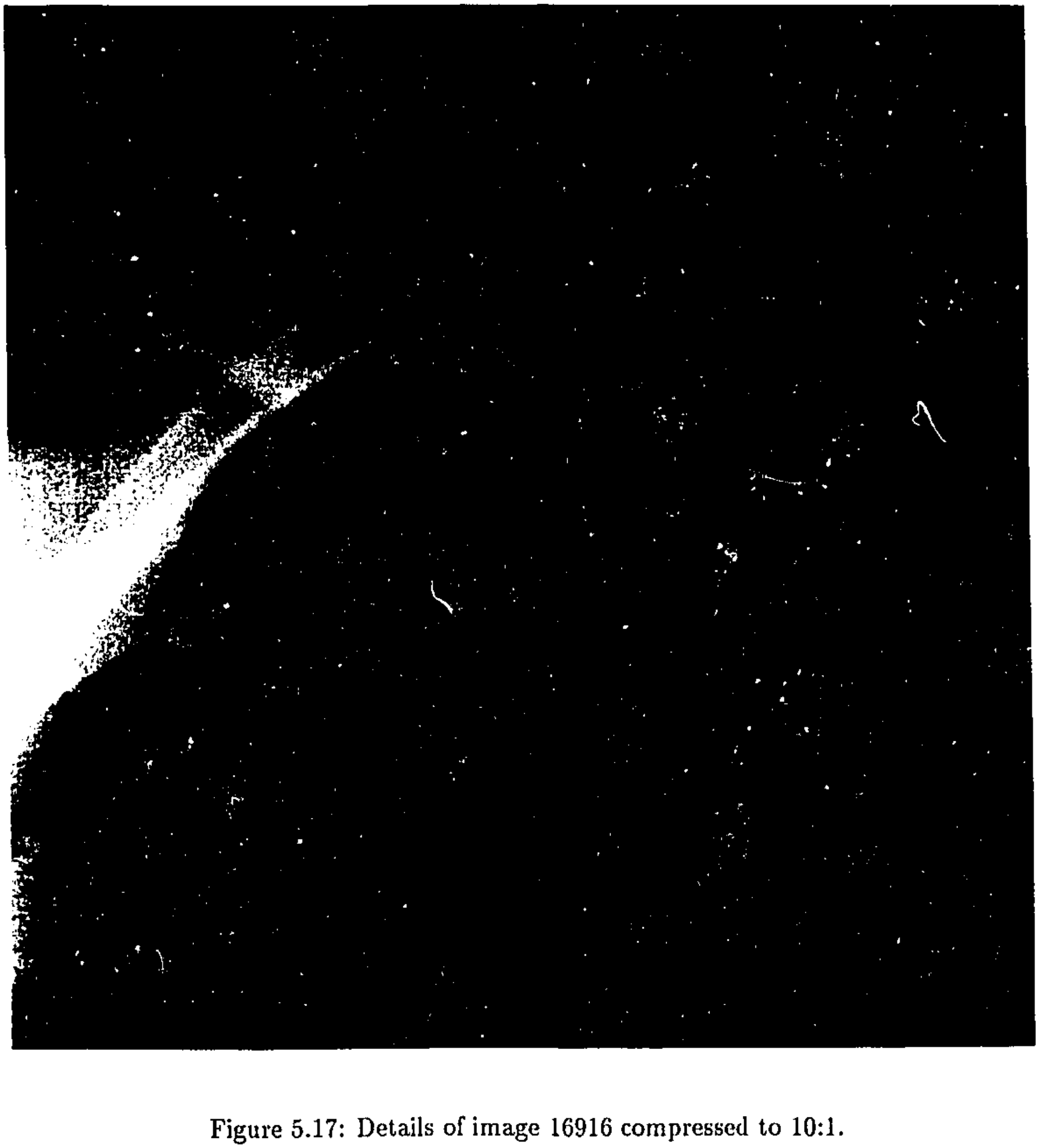


Figure 5.17: Details of image 16916 compressed to 10:1.

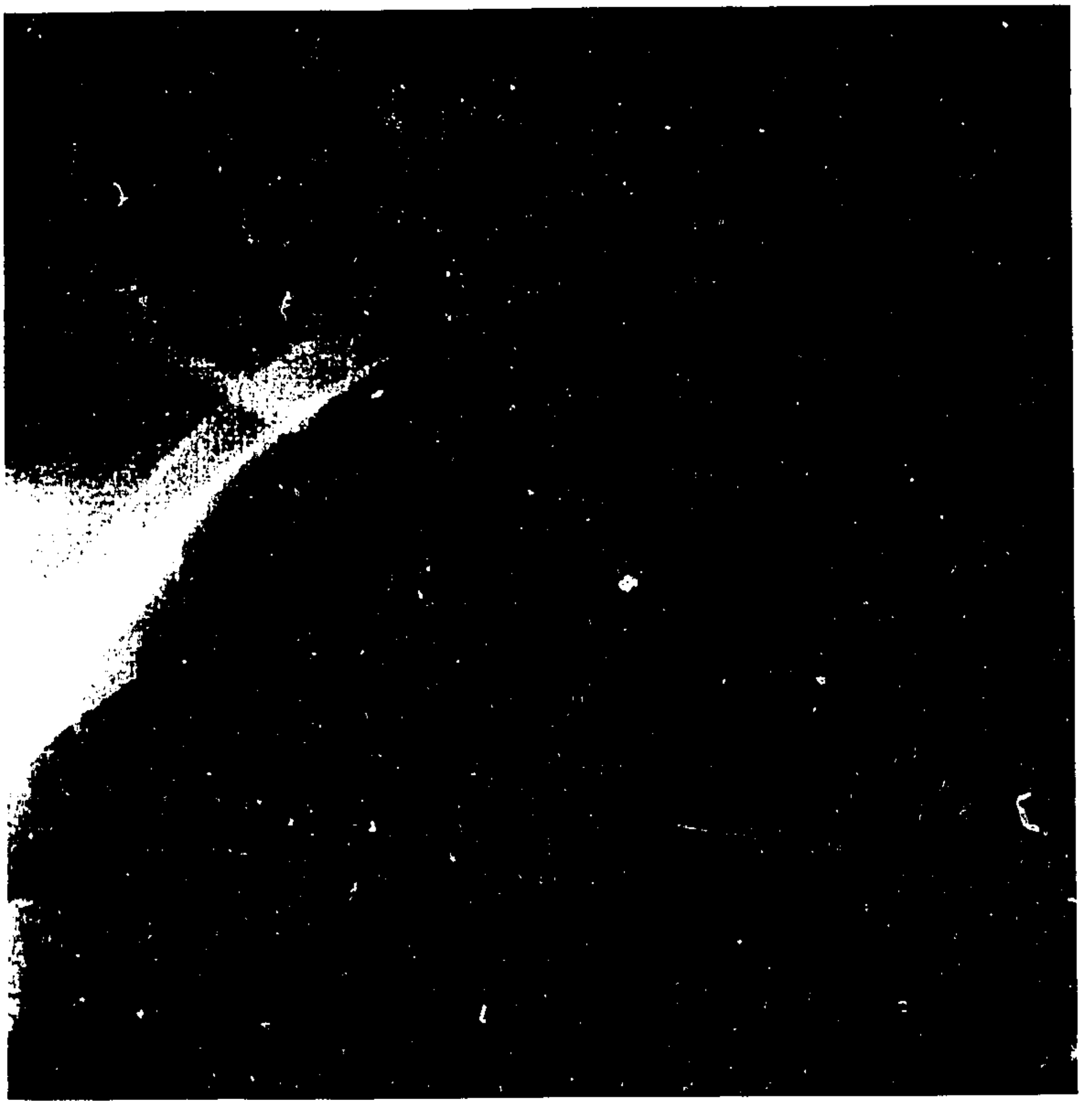


Figure 5.18: Details of image 16916 compressed to 20:1.

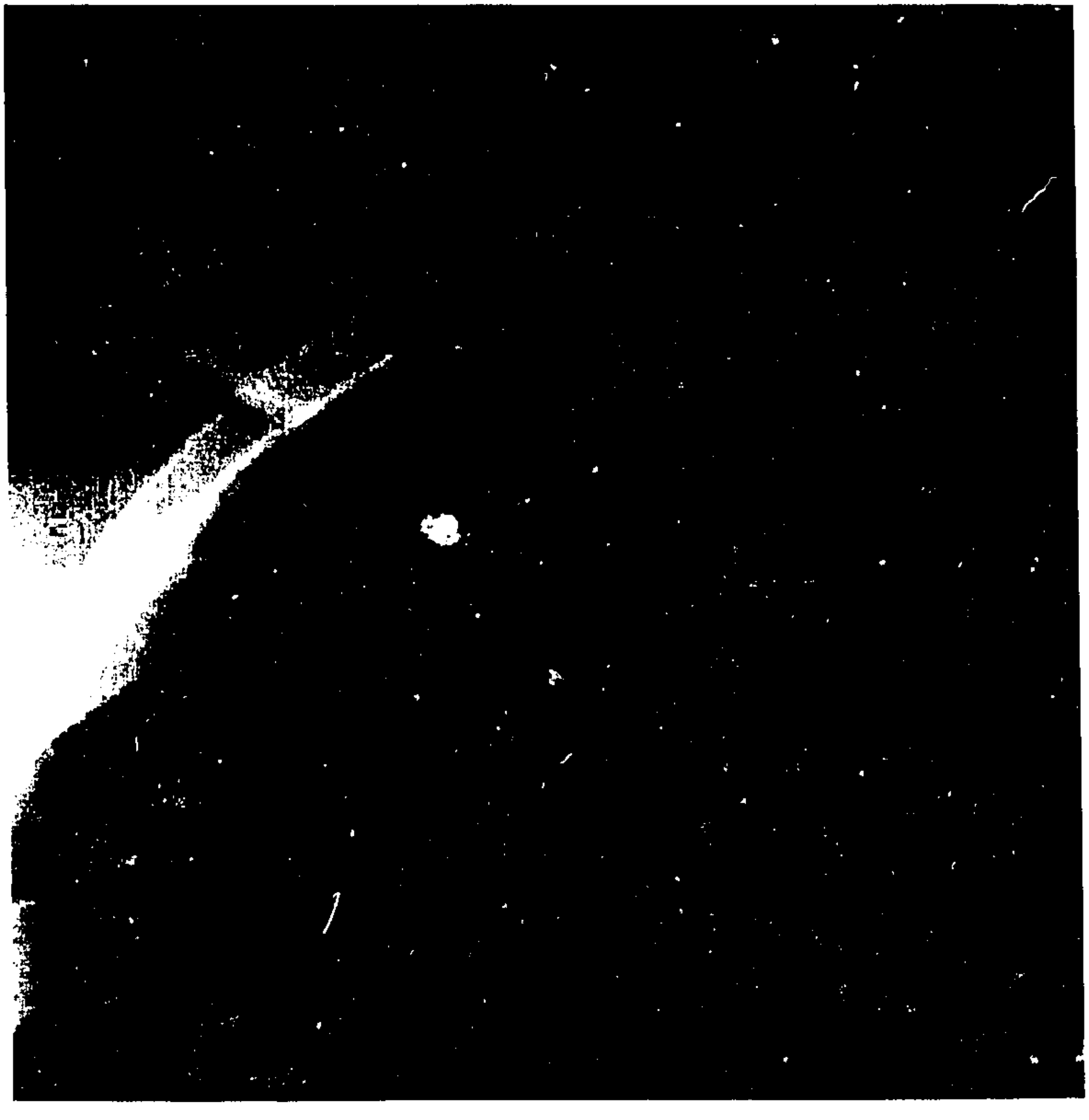


Figure 5.19: Details of image 16916 compressed to 30:1.

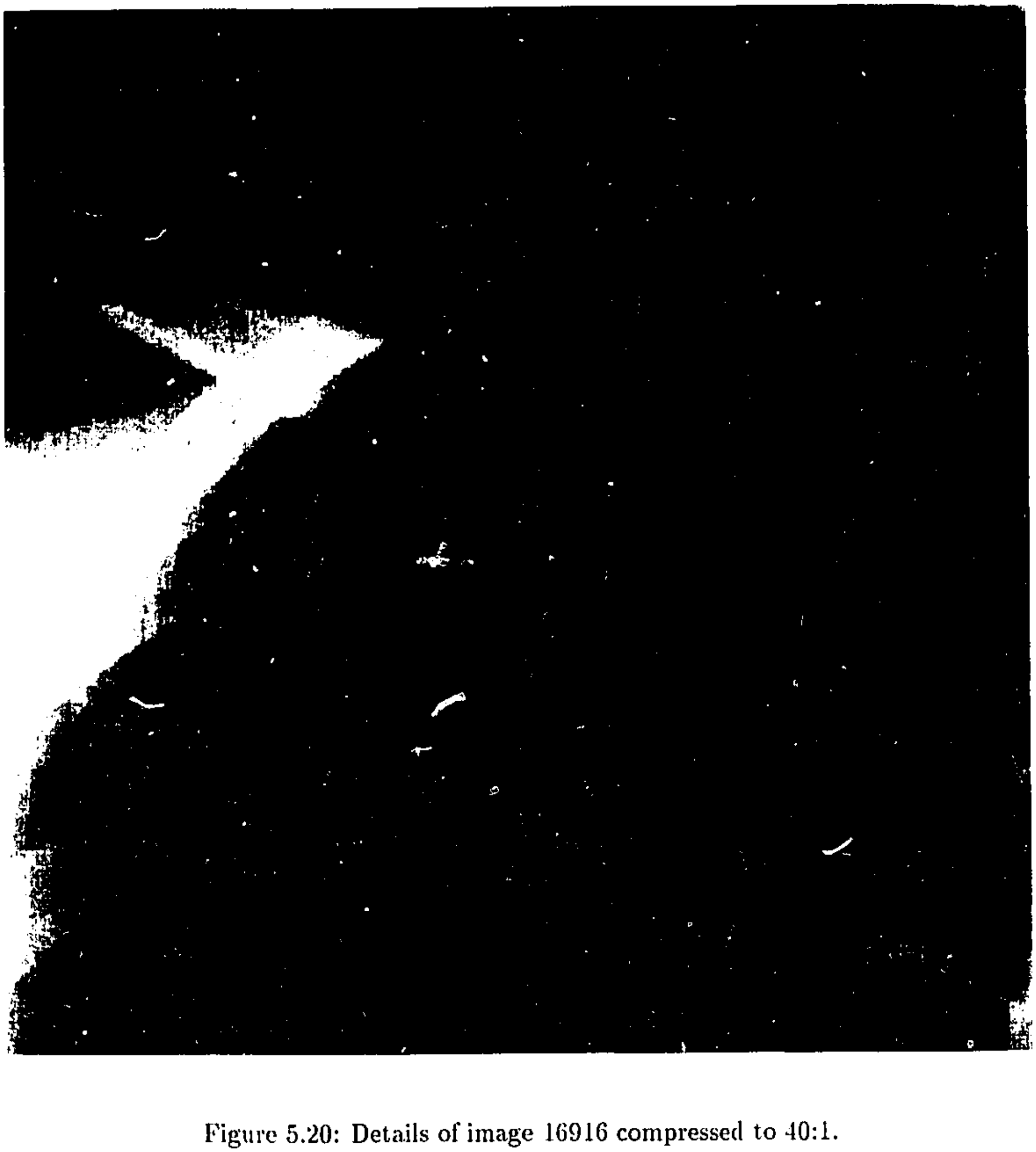


Figure 5.20: Details of image 16916 compressed to 40:1.

Compression Ratio	Block Size	Number of Classes	Quantization Interval
10:1	4×4	512	12.0
20:1	6×6	1024	14.1
30:1	8×8	1024	7.7
40:1	8×8	256	23.8

Table 5.2: Network parameters for the various compression ratios.

The resulting 36 images as well as the nine original images were each printed onto photographic film using a laser film recorder for a total of 45 films. Film was chosen as the presentation medium over a CRT display for a number of reasons:

- Radiologists are most comfortable with the use of film.
- Film has a much larger dynamic range in density than a CRT.
- The laser film recorder also has a better spatial resolution than most CRTs.
- Films are portable and light boxes for viewing are located throughout any radiology department.
- A number of films may be viewed at once on a large light box.

To evaluate the quality of the images, seven physicians took part in the study. Four were senior Radiologists, one of them, who initially selected the images, is a chest specialist. One of the other participants was a physician specializing in Nuclear Medicine, and the final two were Radiology residents.

The sessions for each of the evaluators were conducted as follows. The five versions of an image, including the original, were presented to the evaluator simultaneously, in random order, on a large light box. The degree of compression for each film was hidden from the evaluator. The evaluator was given a sheet where he or she was asked to chose an opinion score for each of the five images, rating both the over-all image quality and the visibility of the pathology. If the physician did not detect a pathology, he or she was asked to rate the visibility of some landmark features. A five-point opinion sco‹e was used with 5 being "excellent," and 1 being "bad." This procedure was repeated for each of the nine images. As a result, each degree of compression was assigned 63 opinion scores.

5.4.3 Comparison to KLT

In order to compare the new compression technique to the globally optimal KLT, a second set of evaluations were performed. To compare the two techniques, the relative distortion of two images compressed to the same degree using the two different approaches must be compared. In this case, it would be sufficient to judge one as being better than the other. If more than one compression ratio were used, an ordinal ranking based on the relative distortion would indicate whether one method was superior to the other.

Two compression ratios were chosen for this second evaluation, namely 30:1 and 40:1. These ratios represent the highest acceptable compression as shown by the results for the previous evaluation, presented in the following section. For the McMEC approach, the same images previously produced at these rates .vere used. The KLT was used to compress the original nine images to the same compression ratios, and these images were printed on the laser film recorder. Figures 5.21 and 5.22 show the same details as figure 5.16 for the KLT compression of 30:1 and 40:1, respectively. Therefore, each image had four versions for a total of 36 films.

It was decided that only one participant, the chest Radiologist, was required for this evaluation. Not surprisingly, he had the most "critical eye" in evaluating the subtle differences between the images in the previous evaluation. For this set of images, the differences are even less visible.

The evaluation was conducted as follows. The four versions of an image were presented to the evaluator simultaneously, in random order, on a large light box. The degree and method of compression was hidden from the evaluator. The evaluator was asked to sort the four images in order of the amount of perceived distortion. It was acceptable to have more than one image ranked the same. This procedure was repeated for each of the nine images.

5.5 Results

5.5.1 Opinion Score Evaluation

After the first set of evaluations, the data were collected and summarized. The results for the rating of image quality are shown in table 5.3. For each compression ratio, the table shows the percentage of times images were assigned a given opinion score. Sixty-three scores were assigned for each compression ratio. As the table shows, there is little difference

Figure 5.21: Details of image 16916 compressed to 30:1 using the KLT.

Figure 5.22: Details of image 16916 compressed to 40:1 using the KLT.

	1:1	10:1	20:1	30:1	40:1
excellent (5)	0.37	0.43	0.35	0.29	0.26
good (4)	0.53	0.48	0.57	0.60	0.52
fair (3)	0.10	0.10	0.08	0.11	0.19
poor (2)	0.0	0.0	0.0	0.0	0.03
bad (1)	0.0	0.0	0.0	0.0	0.0

Table 5.3: Fraction of times images of a given compression ratio were assigned a given opinion score rating image quality.

	1:1	10:1	20:1	30:1	40:1
excellent (5)	0.42	0.41	0.35	0.32	0.33
good (4)	0.48	0.44	0.56	0.57	0.47
fair (3)	0.10	0.14	0.10	0.11	0.19
poor (2)	0.0	0.0	0.0	0.0	0.02
bad (1)	0.0	0.0	0.0	0.0	0.0

Table 5.4: Fraction of times images of a given compression ratio were assigned a given opinion score rating the visibility of the pathology.

between the ratings of the original and the 10:1 and 20:1 versions. In fact, the data show that the quality of the 10:1 image was actually rated higher than the original. At 30:1, the rating drops slightly with fewer images scored as excellent. However, there are still no unacceptable scores at this compression ratio. It is only at a compression of 40:1 that some images, albeit just 2 out of 63, were rated as unacceptable. Even still, over one quarter of the images evaluated at this degree of compression were scored as excellent. For all of the compression rates, no images were scored as bad.

The results for the rating of the visibility of the pathology in an image are similarly shown in table 5.4. Again, the differences in scoring between the original and the 10:1, 20:1, and even 30:1 are relatively minor. At none of these compression ratios were any images scored as unacceptable. At 40:1, one image was rated in the unacceptable range as poor. Despite this, one third of the images were still assigned the top score of excellent. As was the case for the other scoring criterion, no images were assigned the score of bad at any of the compression ratios evaluated.

Further summarizing the data, the plot of the mean opinion score taken across all images

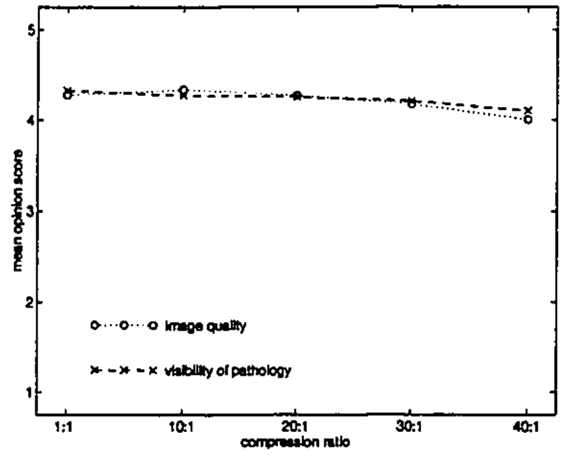


Figure 5.23: Mean opinion score across all images and evaluators.

and evaluators for both scoring criteria is shown in figure 5.23. For both criteria, the MOS at the various degrees of compression remains quite close to that of the original. For image quality, the MOS for the original is 4.28 and only drops to 4.01 at 40:1. The MOS for the pathology visibility is 4.33 for the original and 4.10 for the 40:1 compression ratio.

The comments made by the evaluators were similarly encouraging. Many times the evaluators, after examining the five versions of an image, would comment that there was no difference between the images, or if any, that it was very slight. It was not uncommon for the compressed versions to be scored higher than the original. In fact, a number of times an evaluator would indicate with a high degree of certainty one of the versions as the best, showing the details most clearly, and this version would turn out to be the 40:1 version.

In many instances, when an evaluator scored an image relatively low, he or she would comment that the scoring was based on the distortions visible in high contrast artificial objects in the image. Such objects include surgical staples, necklace chains, catheter tubes, and electrode disks. The image in figure 5.10 contains a number of such objects. The chest Radiologist was particularly adept at identifying such differences. Even then, the evaluators were quick to point out that the differences in the biological structures such as tissue structures, blood vessels, and bones, were far less visible, if visible at all.

Image	McMEC 30:1	KLT 30:1	McMEC 40:1	KLT 40:1
805	1	2	3	3
7731	1	2	3	3
9502	1	3	2	3
9789	2	1	2	2
10555	1	2	1	2
16916	1	2	1	2
19158	1	2	2	2
26366	1	2	3	3
31159	1	2	3	3

Table 5.5: Ordinal ranking of perceived image quality.

5.5.2 Comparison to KLT

The data collected in the second evaluation are shown in table 5.5. For each image, the numbers indicate the ordinal ranking that each of the different versions of the image were assigned. In all but one case, the McMEC 30:1 version was ranked the highest and in all cases, the KLT 40:1 was assigned the lowest rank. When comparing the two techniques at the same compression ratio, in 17 of the 18 cases, the McMEC version was rated better than or equal to the KLT version. At 30:1, the McMEC was rated better than or equal to the KLT 8 times out of 9 and at 40:1, this was the case for all 9 image pairs. It is interesting to note that for 4 of the 9 images, the 40:1 McMEC version was ranked better than or equal to the 30:1 KLT version. This data unequivocally demonstrate the superiority of the McMEC approach over the KLT in terms of reduced distortion for a given compression ratio.

In commenting on the differences between the two types of compression techniques, the evaluator noted that the KLT images appeared grainier relative to the McMEC images. Figure 5.24 shows details of the absolute value of the error with respect to the original of image 16916 compressed to 40:1 using McMEC and the KLT. The KLT error image shows a slightly more uniform distribution of error compared with the McMEC error image. This distribution is consistent with the description of grainy. The distribution of error for the McMEC approach is not so uniform. In somewhat flat regions, the error is relatively small while in "busy" areas such as edges, the error is larger. It is well known that the human visual system is relatively more sensitive to error in flat regions than it is to error in busy regions [49]. Therefore, it appears that the McMEC takes advantage of this characteristic to improve the perceived quality of a compressed image.

5.6 Summary

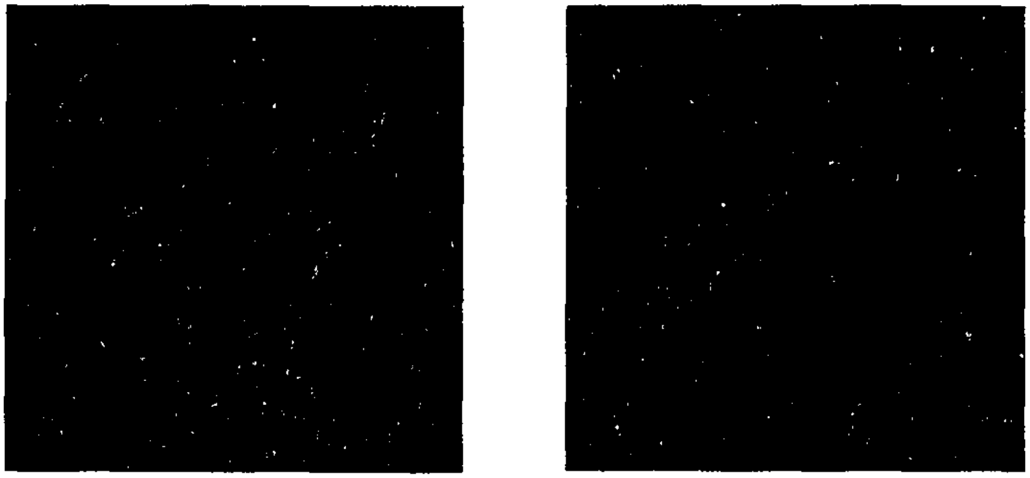
The application of image compression to the medical imaging environment is a very challenging problem. The distortion introduced by a lossy technique must be evaluated carefully. The repercussions of introducing such distortion into an image used for diagnosis can be life-threatening. However, there is a real need for image compression in the medical field due to the large volumes of digital image data currently being generated and the potential growth in the field for the foreseeable future.

One application for image compression in this environment is the use of medical images for educational purposes. This is quite different from the diagnostic use of medical images. In this application, the nature of the pathology portrayed in the image is known beforehand. The purpose of the image is to show this pathology and its attributes in a sufficiently faithful manner, such that it can be identified and its characteristics ascertained. This particular application and its requirements have not received much attention in the research community.

To determine the usefulness of the McMEC approach to image compression, the following investigation was performed. Nine representative clinical chest radiographs, acquired digitally using the Fuji CR system, were chosen for evaluation. The images were compressed using various IDC-McMEC networks trained on two representative images outside the evaluation set. Four degrees of compression were used: 10:1, 20:1, 30:1, and 40:1. Seven physicians evaluated the images. The five versions of each image were shown simultaneously to each evaluator, who was asked to rate each one on a five-point scale for image quality and visibility of pathology. To determine the relative performance of the new approach to the globally optimal KLT in this application, one of the Radiologists was later asked to rank, in order of the severity of the distortion, four versions of each of the nine image: McMEC at 30:1 and 40:1, and the KLT at 30:1 and 40:1.

In the first evaluation, it was found that the mean opinion score differed very little from that of the original across all the compression ratios. Only for the 40:1 versions were there any unacceptable ratings. Even then, these images received a substantial number of top ratings. Many times, the evaluators commented on how little difference there was if any between the images. Occasionally, the physician would choose the 40:1 image as the best. When compared to the KLT, the McMEC versions were ranked better than or as good as the KLT versions 17 times out of 18. In addition, four out of nine times the 40:1 McMEC version was ranked as good as or better than the 30:1 KLT.

It can be concluded, then, that it is possible to compress digital chest radiographs to between 30:1 and 40:1 with an IDC-McMEC network without losing the fidelity necessary for an educational use. In fact, it may be possible to further compress some images, since the number of unacceptable scores at 40:1 was very low and the number of top scores was still quite high. Based on the comparison between the new method and the KLT, the data show conclusively that for a given degree of compression, the distortion of the new approach is less than that of the KLT.



(a) (b)

Figure 5.24: Details of the absolute error with respect to the original of image 16916 compressed to 40:1 using (a) McMEC and (b) KLT.

Chapter 6

Self-Organizing Segmentor and Feature Extractor

A useful by-product of coding an image with either the OIAL or the McMEC networks is the class assignment for each input block. Therefore, these techniques may be applied to the problem in pattern recognition of segmenting an image into a number of distinct regions.

In classical pattern recognition techniques, there are two general phases in the analysis of data. First, features are extracted from the data. Typically, a number of attributes are measured and then some transformation is used to decorrelate and extract only significant features thereby reducing the dimensionality. For example in edge detection, the local pixel values may be transformed by a set of gradient operators to produce a number of directional gradient images. However, in many applications, this step resembles more an art than a science since, despite extensive analysis, the appropriate transformation may not be readily apparent. In the second stage, an analysis of the data is performed to partition the feature space into classes of data. In the case of edge detection, a threshold is typically applied to the gradient images based on some predetermined minimum gradient magnitude. Generally, the partitions tend to be based on a distance metric with respect to some representation of the classes. For example, in many clustering techniques, the classes are represented by the mean of the class data, and the metric is the Euclidean distance in the feature space. The class partitions form bounded regions in the feature space and the form of the decision surfaces between the regions are determined by the metric used.

In subspace pattern recognition, both the feature extraction and class representation phases are combined. As discussed in section 3.3, the basis vectors of the class define both the class by the subspace they define, and the features of the data. The classifier defines *unbounded* regions due to the insensitivity of the classifier to the vector norm of the data. The classification of data is based not on some form of Euclidean distance, but by the efficiency with which a subspace can represent the data as measured by the norm of the projected data. For the maximum entropy classifier, these attributes also hold true. Further, the representation of the classes by their first principal component and the assignment of data based on the largest magnitude of the class output assures that the information preserved by the network is maximized.

Because of the above characteristics of the classifiers upon which the adaptive nature of the OIAL and McMEC networks are based, their use as segmentors warrants some investigation.

6.1 Image Formation Model

The most interesting property from an image processing perspective of both the subspace and maximum entropy methods of classification is that the classification is independent of the norm of the data vector $\mathbf{x}$. For any scalar multiple α, if $\mathbf{x} \in C_i$ then $\alpha\mathbf{x} \in C_i$. This is a very significant characteristic in light of the actual process of image formation. It is known that the luminance at a particular point in an image, $L(x, y)$, can be modelled as the following product [65]:

$$L(x,y) = E(x,y)\rho(x,y) \tag{6.1}$$

where $E(x, y)$ is the illumination falling on that point, and $\rho(x, y)$ is the reflectance of the physical object at that point. In real images, the illumination may vary across an image or it may vary between a number of images. However, this variation must be on a much larger scale than the variations in reflectance for the contents of an image to be useful. Therefore, for a small neighbourhood around a point, $N(x, y)$, the illumination may be considered a constant, $E_{N(x,y)}$, but its value, from region to region, may change. Equation 6.1 can then be rewritten as

$$L(x,y) = E_{N(x,y)}\rho(x,y) \tag{6.2}$$

Typically, the goal in image analysis is to determine characteristics about the underlying physical objects of the scene being imaged. These are inferred from the reflectivity of the scene. Therefore, it is the reflectivity which conveys the information about the scene, and any variation in the illumination can be considered as noise. This is the justification for a class of image processing called "homomorphic processing" [56].

Using vector notation to represent the luminance values of a neighbourhood of pixels as $\mathbf{x}$, the image of this neighborhood, or feature, is formed as

$$\mathbf{x}_1 = E\rho \tag{6.3}$$

which is the vector equivalent of equation 6.2. If the same feature, having the same reflectance, ρ, were to appear elsewhere under different illumination conditions, αE, its image would be

$$\mathbf{x}_2 = \alpha E\rho \tag{6.4}$$

If some image analysis process were performed on these image vectors, one would expect the same result since both $\mathbf{x}_1$ and $\mathbf{x}_2$ were created by the same underlying feature. For many classical pattern recognition approaches, this would not be the case since many use scale-dependent metrics like Euclidean distance. For example, in vector quantization, Euclidean distance is used to measure the distance between input vectors and the codewords. As a result, the distance between $\mathbf{x}_1$ and $\mathbf{x}_2$ may be quite large and result in the codeword representations of the two vectors being different. Both the maximum entropy and subspace methods, however, would treat the two vectors identically, since they would both project to the same subspace independently of the illumination values E and αE. It could be argued, then, that these methods act *directly* on the physical properties of the objects being imaged rather than *indirectly* on the illumination dependent image.

Illumination independence is a very important characteristic of the human visual system [8]. We have no problem in recognizing that the different retinal images formed by the same object under a wide range of illumination conditions do in fact correspond to the same object. It would be very hard indeed for us to function as we do if our visual system did not behave in such a manner. So, for an artificial system processing image data, such independence on variations in illumination would be similarly advantageous. Both the linear subspace classifier and the maximum entropy classifier have exactly this property.

6.2 OIAL Network as Segmentor

6.2.1 Method

As with the McMEC network, the concept of topologically ordered classes can be incorporated into the OIAL class of algorithms. Referring to the algorithm presented in section 3.5, the step for updating the transformation bases, (namely equation 3.16), can be modified to

$$\mathbf{W}_j = \begin{cases} \mathbf{W}_j + \alpha \mathbf{Z}(\mathbf{x}, \mathbf{W}_j), & C_j \in N(C_i) \\ \mathbf{W}_j, & C_j \notin N(C_i) \end{cases} \tag{6.5}$$

where C_i is the winning class according to the subspace classification rule of equation 3.15, and $N(C_i)$ is the set of classes which are in the neighbourhood of class C_i.

A simple topology appropriate for this investigation is a linear arrangement in which the distance between two classes is simply the absolute value of the difference between the class indices. To avoid discontinuities in the topology at the ends, a circular topology could be used where the first and final classes are adjacent. It is this circular arrangement which is used for the following investigation. A system consisting of 32 classes ($K = 32$) with 2 coefficients per class ($M = 2$) was trained on samples from the training MR image shown in figure 3.4a. The training samples consist of overlapping blocks of 8×8 pixels ($N = 64$) as described in section 3.6.1. The learning rule was that shown above in equation 6.5 with $\mathbf{Z}(\mathbf{x}, \mathbf{W}_j)$ chosen as Sanger's GHA. The initial neighbourhood size was 3/4 the size of the entire system or 24 classes. The size of the neighbourhood decreased by two classes for each iteration through the training set.

6.2.2 Results

Once the network was trained, the resulting sets of basis vectors were examined. It was found that the first basis vector of each class was very close to d.c. The mean across all classes of the d.c. gain of the first basis vector was 7.9424, which is very close to the normalized d.c. gain of $\sqrt{N} = 8$. The minimum and maximum d.c. gains across all classes were 7.7056 and 8.0, respectively. As expected, the second basis vectors contained very little d.c. with a mean d.c. gain of only -0.0320. The value of all the weights of the second basis vector across all the classes varied from -0.3854 to 0.5259.

Figure 6.1 shows the second coefficient basis blocks of the system. The class number progresses left to right, top to bottom with the top left class being arbitrarily chosen as class

1. Black represents -0.4 and the lightest colour 0.5. The basis images were colour-coded to match the class assignments shown in subsequent segmentation maps of test images. Examining the class weights shows the preference of the segmentor for edge and line features. This is a rather interesting result as no *a priori* conditions were imposed as to what features were important in the image. In the human visual system, edges and lines are two of the primary features used to construct higher-order representations of scenes [45]. Even when one looks at the original images in figure 3.4, the areas to which one's attention is initially drawn corresponds to areas which the segmentation network has represented as being important, namely, edges and lines. The extraction of important features was accomplished entirely through a self-organizing mechanism. The figure also shows the high degree of similarity between adjacent classes. Bases with similar directional sensitivities tend to be near each other. The sequence of the basis blocks shows how the orientations of the classes progresses through 180°. This arrangement of feature orientations is remarkably similar to the way in which the visual cortex is arranged.

In a classic set of experiments, Hubel and Wiesel recorded the response of neurons in the mammalian visual cortex to a variety of optical stimuii using microelectrodes [27]. They found that groups of neurons arranged in columns responded only to very specific stimuli. In particular, a column would only respond when the eye was presented with lines of a particular orientation. If the angle varied even slightly, the column of cells would stop firing. In terms of the spatial organization of these columns, it was found that the angle of sensitivity differed only slightly (about 10°) between adjacent columns. Further, as the electrode was moved along, the direction of change in the angle, either positive or negative, remained the same. This continuity of change in angle sensitivity persisted, in some cases, up to 270° along a line of columns.

Referring to figure 6.1, the above characteristics of the visual cortex are mimicked by the set of bases. Each basis block is, in effect, a feature detector. The features corresponding to the bases are either lines or edges of a specific orientation. When comparing adjacent classes, the angles of the features are similar. As well, the angles change in a somewhat regular manner as the class number progresses.

It is also interesting to note the distribution of the orientations in the basis set. The training image shown in figure 3.4a contains more vertical and diagonal features than horizontal ones. This non-uniform distribution of orientations is reflected in the basis set. Most of the basis images have either a vertical or diagonal orientation. Only a few correspond to horizontally oriented features, namely classes 25, 26, and 27.

This network was used to segment the test image of figure 3.4b. The segmentation was performed by taking the surrounding 8×8 block for each pixel in the image, classifying the block, and replacing the central pixel by the resulting class value. Since the class topology was circular, the class values were coded by colour with the colour of a class i being the hue at an angle of $i/K \times 360°$ on a colour circle, where K is the total number of classes which in this case is $K = 32$. The intensities were weighted by the magnitude of the second coefficient for each block. Figure 6.2 shows the resulting class map; the same colour coding was used in figure 6.1.

The figure clearly shows the preference of the segmentor for edge and line features. In most areas of the image, it is acting as either an edge or line detector. The edges around the skull, the orbit, and sinus cavity are dramatically shown.

The continuity of the colour transitions shows the high degree of similarity between neighbouring classes. Since the class indices were coded as a spectrum of colours, similar colours indicate similar classes. Starting at the base of the skull in the lower left of the image and going around the skull in an anti-clockwise direction, the colours progress from green to yellow, orange, to red, to violet at the top of the skull, to blue, and finally back to green at the forehead. Throughout the image, too, features with the same orientations are consistently segmented with the same class. For example, the horizontal features around the the top and bottom of the orbit are mapped to the same classes as the horizontal features at the top of the skull.

The class assignment of a feature with a particular orientation is independent of whether it has a positive going direction or negative going direction. For example, vertical edges are labelled green whether the gradient is to the right or to the left. The d.c. gain of the second basis vector of each class is approximately zero. Since the representation of an input vector by a subspace and hence its classification is independent of the sign of the coefficients, multiplying the basis vectors by -1 does not change the classification. Therefore, changing the gradient direction of an edge by 180° does not effect its classification.

To examine the effect of the network on an image significantly different from the training image, it was used to segment the Lenna image of figure 3.9. Figure 6.3 shows the resulting segmentation map. Despite the fact that the network was trained on a very different image, the classification of features is the same as that of figure 6.2. Again, the network has identified perceptually important features. It has picked up all the significant edges and lines and has classified them in a consistent manner. Horizontal features are blue to purple, vertical features are green, diagonals in the top-right to bottom-left orientation are orange and the opposite diagonals are blue to green. As was the case for the compression results using the OIAL network, these results indicate that the segmentation properties of the network appear to generalize well across different images.

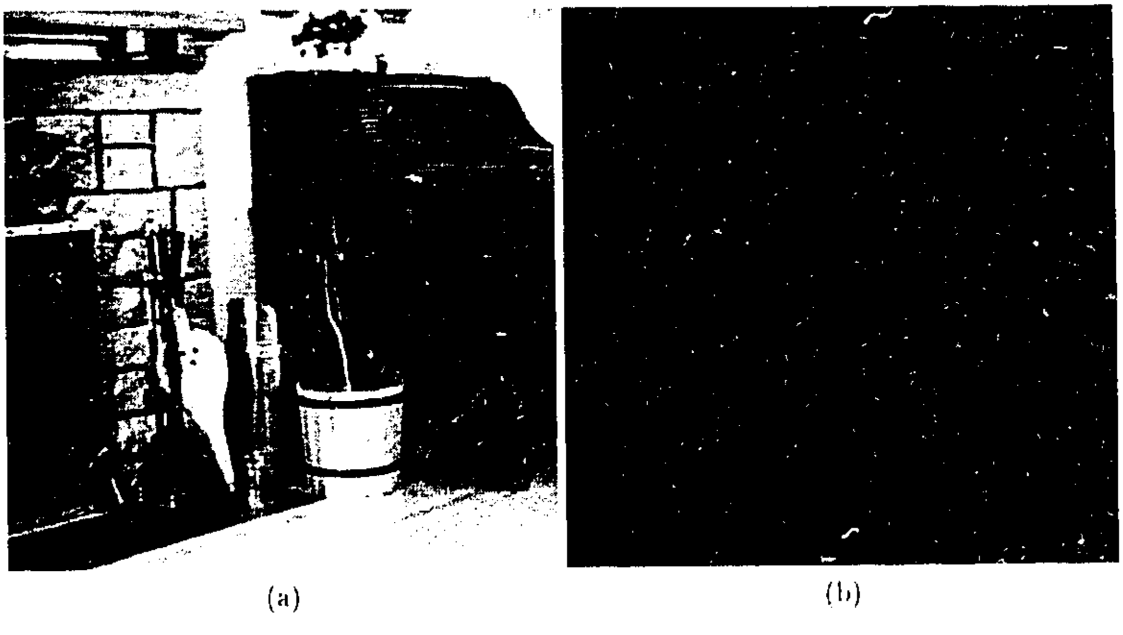
To test the performance of the segmentation under different illumination conditions, the network was used to segment the images shown in figure 6.4. Both images are 400 × 400 pixels in size with 256 levels of gray. One image was captured with the scene being brightly illuminated while the other was captured after the lights were turned off. The respective segmentation maps are shown in figure 6.5. Because of the difference in dynamic range between the two input images, the intensity weighting of the segmentation map by the magnitude of the second coefficient was normalized for consistency. The hue as the indicator of class assignment, however, was not modified. As these two segmentation maps show, a substantial change in illumination has very little effect on the assignment of the features in a image to the various classes. As per the discussion in section 6.1, this result is consistent with what would be expected.

Of course, in practice, there are limitations to the illumination independence characteristic of this segmentation technique. A completely black image would have an arbitrary segmentation since a zero vector projects equally well in all the subspaces. Even with a very small input vector, any noise could substantially change the direction and hence the classification of the vector. Since actual digital images are inherently quantized in the illumination dimension, the quantization noise for extremely poorly illuminated images may then present such a problem. Generally, if the signal-to-noise ratio is large, with the noise including both quantization noise and any sensor noise, then the illumination independence characteristic of the segmentor is valid. However, as the illumination drops to the point where the SNR approaches 1, then the noise will begin to interfere with the performance of the segmentor and the illumination independence fails. At the other extreme, too much illumination may saturate the sensor thereby distorting the signal. Such distortion may degrade the performance of the segmentor.

Figure 6.1: Map of second coefficient basis blocks for 32 class, 2 coefficient OIAL network. The class number progresses left to right, top to bottom.

Figure 6.2: OIAL segmentation map of test image with 32 classes, 2 coefficients per class. Colour indicates class membership, intensity is weighted by the magnitude of the second coefficient for each block.

Figure 6.3: OIAL segmentation map of Lenna image with 32 classes, 2 coefficients per class.



(a)

(b)

Figure 6.4: Test images of same scene under different illumination.

6.3 McMEC Network as Segmentor

6.3.1 Method

The McMEC network used to investigate the segmentation performance was the single coefficient network with 32 classes which was trained as described in section 4.4.1 and used in the evaluation of compression performance in section 4.4.2.

6.4 Results

Examining the set of of basis vector weights for the McMEC network yields some important differences between the OIAL representation which uses two coefficients, and the McMEC representation which uses only one coefficient. In the former, the first basis vector for every class was approximately the d.c. vector. Because they were all the same, it was the second basis vectors which effectively defined each class and, as was noted before, the second basis vectors had no d.c. component. For the McMEC network, there is only one basis vector which defines each class. In this network, each vector has a strong d.c. component. The mean d.c. gain across all classes was 7.4112 and varied substantially from a minimum of 5.8688 to a maximum of 8.0. Because of the strong d.c. component in each vector, all the values of the weights were positive and varied from 0.0129 to 0.2705.

The resulting class weights are shown in figure 6.6. The class number progresses left to right, top to bottom with the top left class being class 1. Again, the figure was colour-coded to match the class assignments shown in subsequent segmentation maps. This figure shows the preference of the network for not only lines and edge, but flat areas as well. In the OIAL network, flat regions were sufficiently represented by only the first coefficient. Therefore, there was no need for such a representation in the second basis vector and, hence, the zero d.c. gain. With only one basis vector per class, flat regions must be represented explicitly. Again, the extraction of these perceptually important features was accomplished in an entirely self-organizing manner as no conditions were imposed beforehand on the importance of specific types of features.

The progression of features across adjacent classes is quite regular. The orientations of the features of adjacent classes are similar. This shows that the network growing approach used to train this network, as illustrated in figure 4.11, produces a topologically ordered map of features. Again, there are parallels between the organization of this feature map and

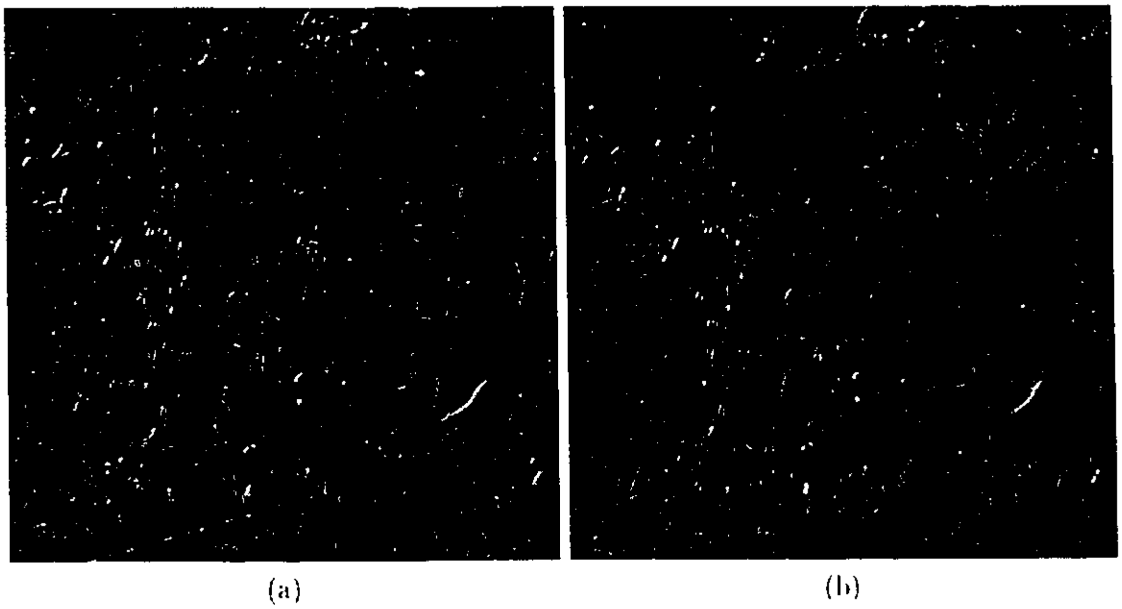


Figure 6.5: OIAL segmentation maps of (a) brightly illuminated test image, (b) dimly illuminated test image.

Figure 6.6: Map of basis images for 32 class McMEC network. The class number progresses left to right, top to bottom.

the regular progression in the orientation sensitivity of the columns in the visual cortex.

The distribution of features in the feature map, as in the OIAL representation of figure 6.1, corresponds to the relative distribution of the same features in the training image. Vertical and diagonal features are well represented as they are quite prevalent in the training image. As well, the number of relatively flat features in the map corresponds to the relative predominance of slowly varying regions in the image.

The segmentation map of the MR image in figure 3.4b for this network is shown in figure 6.7. As before, the class assignments were colour-coded using 32 colours from a continuous colour circle. Since the McMEC network does not have a second non-d.c. coefficient, the intensity of the segmentation map was not weighted.

One of the first noticeable features of this segmentation map is the apparently random distribution of class assignments in the background. In this area, the values of the pixels are low, but there is some degree of variation due to noise. Because of the relatively flat characteristic of the background, one would expect it to be assigned to a uniformly flat class. However, adding a degree of variation to the relatively small magnitudes of the blocks in this region substantially changes the angle of the input vector and therefore can change the class assignment. This effect clearly illustrates the degraded performance of the segmentor in very dark regions with low SNR. Such a random variation is also present in the background region in the segmentation map for the OIAL network as shown in figure 6.2, but is not visible due to the weighting by the magnitude of the second coefficient.

The foreground figure of the head, however, is well classified. The edges and lines around the skull and orbit are all distinctly assigned to the appropriate classes. Comparing this figure with the colour-coded feature map of figure 6.6, one can see that the classification of features in the image correspond to those in the feature map. As previously mentioned, one of the differences between the sets of features for the OIAL and McMEC networks is the presence of almost pure d.c. features in the latter. The relatively smooth areas in the brain tissue are classified as such. In the OIAL segmentation map, such flat regions are arbitrarily classified, but again, this variation is not visible in the segmentation map due to the intensity weighting by the second coefficient.

A further difference between the two segmentation maps is evident around the edge of the skull. The difference is well illustrated when comparing the class assignments along a line at right angles to the skull, crossing the top of the forehead, and going from the brain tissue just inside the skull to the outside of the skull. In the McMEC segmentation map, the negative-going edge as the line crosses the edge of the brain, the flat black region between the brain and the skull, and the positive-going edge on the inside edge of the skull are all classified differently with the colours green, blue and purple assigned, respectively. In the OIAL segmentation map, negative-going edges are classified the same as positive-going edges. This difference, again, is due to the presence in the former and the absence in the latter of a d.c. component in the basis vectors which effectively define the classes.

The McMEC network was used to segment the Lenna image, and the resulting segmentation map is shown in figure 6.8. The figure clearly shows that the network has captured the essential information required to represent an image, since the details in the original image are readily identifiable in the segmentation map. The features in the image are classified in a consistent manner with areas of slowly varying intensity similarly labeled, and the various edges in the image are labeled according to their orientation. Since there are no very dark background regions in the image, there is no failure of the segmentor due to a low SNR.

Again, to test the illumination independence of the network, it was used to segment the two images of the same scene under different illumination conditions as shown in figure 6.4. The resulting segmentation maps are shown in figure 6.9. As was the case for the OIAL network, the McMEC segmentation maps for the two images are virtually indistinguishable from each other. However, the random class assignments in the very dark regions for both images show how the segmentor can fail when the illumination is so low that there is very little signal in the image. But for most regions in the poorly lit image, there is still sufficient signal upon which to make an appropriate classification.

6.5 Summary

The use of both the OIAL and McMEC networks as segmentors yields some interesting results. The networks have been shown to extract features from the test image in a completely self-organizing fashion. No *a priori* assumptions were imposed as to the importance of any type of feature, yet the features extracted by the networks, namely, flat regions, edges, and lines, are all of perceptual importance.

The classification of similar features was consistent across an image. This consistency of classification was valid not only for images similar to the training image, but also for images with substantially different characteristics.

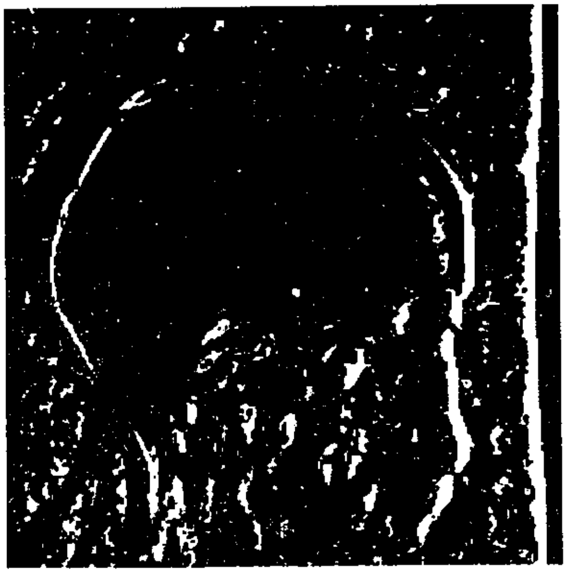


Figure 6.7: McMEC segmentation map of test image with 32 classes. Colour indicates class membership.

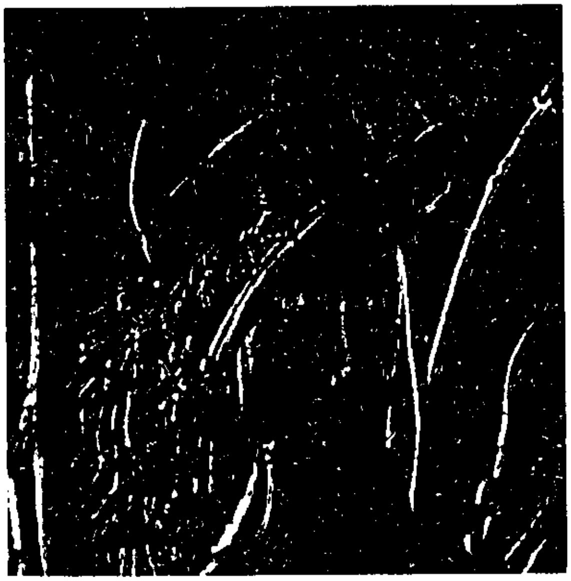


Figure 6.8: McMEC segmentation map of Lenna image with 32 classes.

The topological ordering of the classes during training resulted in like classes being close together in a manner analogous to the ordering of directionally sensitive columns in the visual cortex. Both the network growing approach to training or the more conventional shrinking neighbourhood approach achieved such an ordering.

The segmentation was also shown to be independent of variations in illumination. Considering that the human visual system has such a characteristic, these results are significant. Further, by examining this independence in the context of the multiplicative image formation model, it can be shown that these segmentors are effectively operating directly on the underlying physical properties of the scene as opposed to indirectly on the luminance values of the image which may be corrupted by noise in the illumination component. Segmentors based on distance measures such as Euclidean distance do not share this property.

While the above characteristics are valid for both networks, there are some significant differences between them. These differences stem from how the two networks incorporate the d.c. information in their network.

The OIAL network in this investigation represented the classes with two components per class. It was found that the the first component was approximately d.c. for all the classes, while the second components, which then effectively defined the classes, had no d.c. As a result, features which differed by a negative sign in the don-d.c. components, for example, positive- and negative-going edges of the same orientation, were assigned the same class. In addition, slowly varying regions were arbitrarily classified.

The McMEC network, on the other hand, used only one component per class so that each of those components contained a significant amount of d.c. In this case, features such as edges with opposite directions were assigned different classes. Further, flat or slowly varying regions were well represented in the set of features and therefore were distinctly classified.

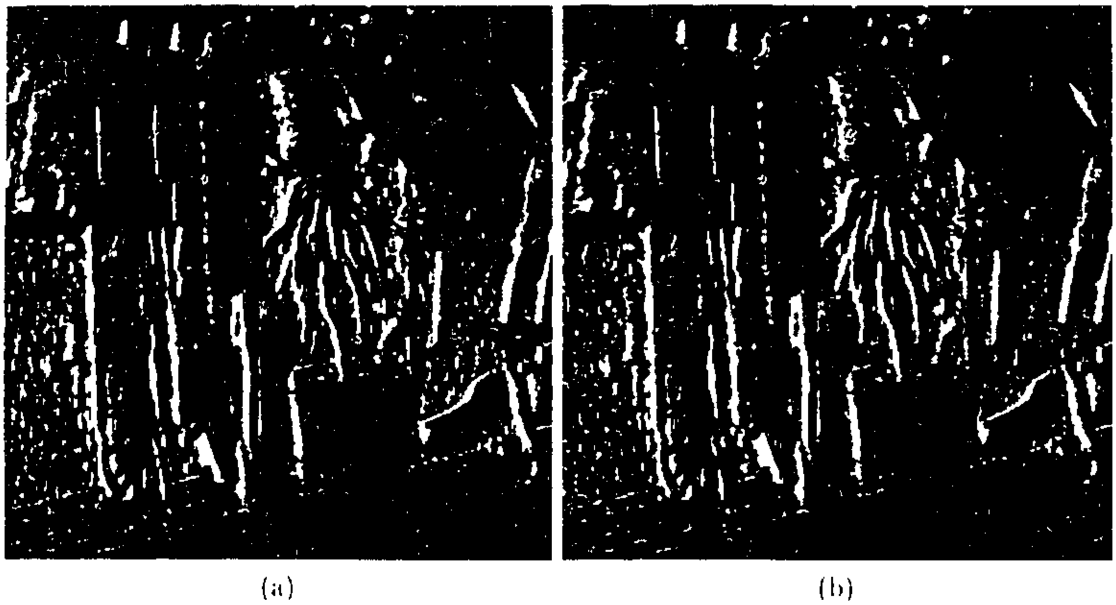


Figure 6.9: McMEC segmentation maps of (a) brightly illuminated test image, (b) dimly illuminated test image.

Chapter 7

Conclusion

The main objective of this thesis is to apply neural network approaches to the problem of image compression. A number of characteristics of neural networks as signal processing tools make them particularly well-suited to this problem:

- Their nonlinear nature can be used when the assumption of linearity in image models is not valid. When used as nonlinear predictors for example, performance improvements can be realized over linear methods.

- They may be allowed to adapt over long-term variations in the signal. The iterative learning, whether during the initial training phase or the following adaptive phase, extracts the most important features from the data in a computationally efficient manner and without the overhead of estimating and processing higher-order statistics. For example, Hebbian learning rules can extract principal components without the need to estimate the covariance matrix and perform an eigendecomposition.

- The self-organizing ability of these networks can be used to create topologically ordered maps of features. The structure of these maps may be exploited to improve coding performance and/or coding complexity. Further, these maps may mimic some of the characteristics of biological neural networks.

Neither the classical nor current neural network-based image compression schemes adequately address the locally nonstationary nature of images. Most classical techniques are based on optimizing some global measure of performance which assumes a stochastically stationary model. However, images contain a rich mixture of many types of regions. Techniques based on a global measure of optimality will not perform well in such regions where the statistics differ from the global estimate. While there has been recognition of the need for adaptation in the compression of images, there has not been a sufficient treatment of the optimality of such adaptation.

A new approach to data representation, a mixture of principal components (MPC) was developed which, together with principal component analysis (PCA) and vector quantization (VQ), form a spectrum of representations. At one extreme is PCA, which is a completely linear transformation, using up to N dimensions of the original data space in a continuous representation. It is a very powerful representation, and has found widespread use in data compression. At the other extreme, VQ is a highly nonlinear approach, which partitions the data space into a number of discrete regions. It uses zero-dimensional Voronoi centres to represent the data. Between these two extremes is the MPC representation. It partitions the data space into a number of discrete regions which form subspaces of the original space. As such, it is a nonlinear technique. However, within each subspace, the data are represented by the M principal components of the subspace and therefore this representation has some of the characteristics of PCA. Because MPC incorporates features of both PCA and VQ, its use in image coding is justified.

This thesis has presented a number of novel neural network based image compression methods, which address the concern of optimality in adaptation using the MPC representation. These networks are modular in design. Each module consists of a set of one or more basis vectors which performs a linear transformation on the input data. In addition, each module represents a class of input data and the basis vector(s) of each module defines the class. During coding, an input vector is transformed by each module and the coefficient(s) of the winning class, as chosen by a classifier along with the class index, is output by the network. The classifier for the OIAL network developed in chapter 3 is the subspace classifier. An input vector is assigned to the class whose subspace best represents the vector. The subspace for each class is defined by the transformation basis vectors of the class. The McMEC approach and its variants as presented in chapter 4 uses only one basis vector per class and chooses the class which has the largest magnitude coefficient.

These networks have the following significant characteristics.

- The adaptation is optimal. The OIAL network minimizes the mean squared error representation of the data. The McMEC approach maximizes the information retained by the network. While two different criteria were used, the McMEC network turns out to be a special case of the OIAL network.

- The adaptation is self-organizing, since no assumptions on the importance of or the relation between the regions, or their defining features are imposed beforehand.

- The adaptation responds to differences between regions in an image on a block-to-block basis. Other adaptive approaches can only adapt to long-term variations in the data.

- The adaptation criteria are efficiently represented by the networks, since the set of weights in each module defines not only its linear transformation but also the class of the module itself.

- The performance of these adaptive networks can surpass that of the optimal nonadaptive KLT.

- The networks have some computational advantage in terms of complexity at the decoder over the KLT and the fast DCT. While the complexity of encoding is worse than the DCT, the encoding complexity may be not as important in applications where an image is encoded once and decoded many times.

The networks were applied to the problem of compressing digital chest radiographs for an educational application. Nine images were compressed to 10:1, 20:1, 30:1, and 40:1. Six Radiologists and a Nuclear Medicine physician rated the image quality and the visibility of pathology in the different versions of the images in a blind evaluation. A second experiment used the 30:1 and 40:1 versions of the images compressed with the new network and the 30:1 and 40:1 versions using the KLT. An expert Radiologist ranked in order, based on the degree of distortion, the four versions of each image. The results of these evaluations can be summarized as follows:

- Compression ratios of between 30:1 and 40:1 can be realized without an unacceptable loss in image quality.

- Even at 40:1, the images received a substantial number of top ratings.

- The evaluators commented many times on how little difference there was between any of the different versions of the images.

- When comparing the new method to the classical KLT, 17 out of 18 times the quality of the images compressed using the new technique was judged to be as good as or better than the KLT.
- Further, four out of nine times the 40:1 version using the new method was ranked as good as or better than the 30:1 KLT.

The use of these networks as segmentors has yielded some interesting results which may be summarized as follows:

- Perceptually important features, namely, flat regions, lines, and edge, were extracted by the networks in a completely self-organized fashion.
- The classification of similar features was consistent across a variety of images.
- The topological ordering of classes had some characteristics similar to the arrangement of directionally sensitive columns in the visual cortex.
- The segmentation was shown, both theoretically and experimentally, to be independent of changes in the illumination of a scene.
- The two-component OIAL network and the one-component McMEC network differed in the way they represented d.c. information. As a result, the latter network segmented flat regions well and separately classified features which differed by 180° in orientation.

In conclusion, the new networks developed in this thesis represent a fundamentally new representation of data using the MPC. These networks can be applied not only to image compression but also to segmentation. Their use has been shown to result in significant improvements over existing methods due to their optimally adaptive nature.

Appendix A

List of Abbreviations

APEX: Adaptive principal component extraction

bpp: Bits per pixel

CR: Computed radiography

CRT: Cathode-ray tube

CT: Computed tomography

d.c.: Direct current

DCT: Discrete cosine transform

DF: Digital flurography

DPCM: Differential pulse-code modulation

DSA: Digital subtraction angiography

flops: Floating point operations

GHA: Generalized Hebbian algorithm

IDC-McMEC: Implied d.c. multi-class maximum entropy coder

IP: Imaging plate

JPEG: Joint Photographics Experts Group

KLT: Karhunen-Loève transform

LAN: Local area network

LBG: Linde, Buzo and Gray

LOT: Lapped orthogonal transform

McMEC: Multi-class maximum entropy coder

MIND: Medical Imaging and Network Development

MOS: Mean opinion score

MPC: Mixture of principal components

MRI: Magnetic resonance imaging

MSE: Mean squared error

MUMC: McMaster University Medical Centre

NM: Nuclear medicine

OIAL: Optimally integrated adaptive learning

PCA: Principal components analysis

PCM: Pulse-code modulation

PET: Positron emission tomography

PSNR: Peak signal-to-noise ratio

ROC: Receiver operator characteristics

SNR: Signal-to-noise ratio

SOFM: Self-organizing feature map

SPECT: Single photon emission computerized tomography

TS-McMEC: Tree-structured multi-class maximum entropy coder

US: Ultrasonography

VQ: Vector quantization

Bibliography

[1] Hüseyin Abut, editor. *Vector Quantization*. IEEE Press, New York, NY, 1990.

[2] H. Barlow and P. Földiák. Adaptation and decorrelation in the cortex. In R. Durbin, Miall C., and G. Michison, editors, *The Computing Neuron*, pages 54-72. Addison-Wesley, Reading, MA, 1989.

[3] Atilla Baskurt and Isabelle Magnin. Adaptive coding method of X-ray mammograms. In *SPIE Vol. 1444 Image Capture, Formatting, and Display*, pages 240–249, San Jose, CA, February 24-26 1991.

[4] S. Becker and M. Plumbley. Unsupervised neural network learning procedures for feature extraction and classification. To appear *Int. J. Applied Intelligence*, 1996.

[5] J.H. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.

[6] H. Chen and R. Liu. Adaptive distibuted orthogonalization process for principal components analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing '92*, pages II 283–296, San Francisco, CA, March 23-26 1992.

[7] R. R. Coifman and Y. Meyer. Nouvelles bases orhonomées de $L^2(\mathbf{R})$ ayant la structure du systeèm de Walsh. *Preprint, Yale University*, 1989.

[8] Tom N. Cornsweet. *Visual Perception*. Academic Press, New York, NY, 1970.

[9] P. C. Cosman, C. Tseng, R. M. Gray, R. A. Olshen, L. E. Moses, H. C. Davidson, C. J Bergin, and E. A. Riskin. Tree-structured vector quantization of CT chest scans: Image quality and diagnostic accuracy. *IEEE Trans. on Medical Imaging*, 12(4):727–739, December 1993.

[10] K. I. Diamantaras. *Principal Component Learning Networks and Applications.* PhD thesis, Princeton University, October 1992.

[11] Robert D. Dony and Simon Haykin. Self-organizing segmentor and feature extractor. In *Proc. IEEE Int. Conf. on Image Processing*, pages III 898–902, Austin, TX, November 13-16 1994.

[12] Robert D. Dony and Simon Haykin. Neural network approaches to image compression. *Proc. IEEE*, 83(2):288–303, February 1995.

[13] Robert D. Dony and Simon Haykin. Optimally adaptive transform coding. To appear *IEEE Trans. Image Processing*, 1995.

[14] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[15] James P. Egan. *Signal Detection Theory and ROC-Analysis.* Academic Press, New York, NY, 1975.

[16] P. Földiák. Adaptive network for optimal linear feature extraction. In *Int. Joint Conf. on Neural Networks*, volume 1, pages 401–405, Washington, DC, 1989.

[17] Fuji Photo Film Co., Ltd., Tokyo, Japan. *Fuji Computed Radiography System FCR AC-1 Product Specification*, 2nd edition, February 1990.

[18] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Norwell, MA, 1992.

[19] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing.* Addison-Wesley, Reading, MA, 1977.

[20] Robert M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1:4–29, 1984.

[21] Simon Haykin. *Adaptive Filter Theory.* Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1991.

[22] Simon Haykin. *Communication Systems.* Wiley, New York, NY, 3rd edition, 1994.

[23] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan, New York, NY, 1994.

[24] Simon Haykin. Neurosignal processing: A paradigm shift in statistical signal processing. Submitted to *IEEE Signal Processing Magazine*, 1995.

[25] D. O. Hebb. *The Organization of Behavior.* Wiley, 1949.

[26] Geoffry E. Hinton. Personal communucations, 1995.

[27] David H. Hubel and Torsten N. Wiesel. Brain mechanisms of vision. *Scientific American*, pages 130–146, September 1979.

[28] Don R. Hush and Bill G. Horne. Progress in supervised neural networks: What's new since Lippmann. *IEEE Signal Processing Magazine*, 10(1):8–39, January 1993.

[29] Anil K. Jain. Image data compression: A review. *Proc. IEEE*, 69(3):349–389, March 1981.

[30] N. S. Jayant and P. Noll. *Digital Coding of Waveforms.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[31] Nikil Jayant, James Johnston, and Robert Safranek. Signal compression based on models of human perception. *Proc. IEEE*, 81(10):1385–1421, October 1993.

[32] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, 3rd ed. edition, 1988.

[33] Teuvo Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, September 1990.

[34] S. Y. Kung and K. I. Diamantaras. A neural network learning algorithm for adaptive principal component extraction (APEX). In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 90*, pages 861–864, Alburqurque, NM, April 3-6 1990.

[35] S. Y. Kung, K. I. Diamantaras, and J. S. Taur. Adaptive principal component extraction (APEX) and applications. *IEEE Trans. Signal Processing*, 42(5):1202–1217, May 1994.

[36] Murat Kunt, Michel Bénard, and Riccardo Leonardi. Recent results in high-compression image coding. *IEEE Trans. Circuits and Systems*, CAS-34(11):1306–1336, November 1987.

[37] Murat Kunt, Athanassios Ikonomopoulos, and Michel Kocher. Second-generation image-coding techniques. *Proc. IEEE*, 73(4):549–574, April 1985.

[38] Tsu-Chang Lee and Allen M. Peterson. Adaptive vector quantization using a self-development neural network. *IEEE J. on Selected Areas in Communications*, 8(8):1458–1471, October 1990.

[39] Yoseph Linde, Andrés Buzo, and Robert M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Communications*, 28(1):84–95, January 1980.

[40] Richard P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4:4–22, April 1987.

[41] Bruce W. Long. Image enhancement using computed radiography. *Radiologic Technology*, 61:276–280, 1990.

[42] S. P. Luttrell. Image compression using a neural network. In *Proc. IGARSS '88*, pages 1231–1238, Edinburgh, Scotland, September 13-18 1988.

[43] Henrique S. Malvar. *Signal Processing with Lapped Transforms.* Artech House Inc., Norwood, MA, 1992.

[44] Henrique S. Malvar and David H. Staelin. Reduction of blocking effects in image coding with a lapped orthogonal transform. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 88*, pages 781–784, New York, NY, April 11-14 1988.

[45] Henrique S. Malvar and David H. Staelin. The LOT: Transorm coding without blocking effects. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-37(4):553–559, April 1989.

[46] David Marr. *Vision.* W. H. Freeman, New York, NY, 1982.

[47] Jean D. McAuliffe, Les E. Atlas, and Carlos Rivera. A comparison of the LBG algorithm and Kohonen neural network paradigm for image vector quantization. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing '90*, pages 2293–2296, Alburquerque, NM, April 3-6 1990.

[48] Hans Georg Musmann, Peter Pirsch, and Hans-Joachim Grallert. Advances in picture coding. *Proc. IEEE*, 73(4):523–548, April 1985.

[49] Nasser M. Nasrabadi and Robert A. King. Image coding using vector quantization: A review. *IEEE Trans. Communications*, 36(8):957–971, August 1988.

[50] Arun N. Netravali and Barry G. Haskell. *Digital Pictures: Representation and Compression.* Plenum Press, New York, NY, 1988.

[51] Arun N. Netravali and John O. Limb. Picture coding: A review. *Proc. IEEE*, 68(3):366–406, March 1980.

[52] Erkki Oja. A simplified neuron model as a principal component analyzer. *J. Math. Biology*, 15:267–273, 1982.

[53] Erkki Oja. *Subspace Methods of Pattern Recognition.* Research Studies Press Ltd., Letchworth, U.K., 1983.

[54] Erkki Oja. Neural networks, principal components, and subspaces. *Int. J. Neural Systems*, 1(1):61–68, 1989.

[55] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[56] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *J. Math. Analysis and Applications*, 106:69–84, 1985.

[57] Alan V. Oppenheim and Ronald W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[58] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Academic Press, New York, NY, 1990.

[59] E.A. Riskin, T. Lookabaugh, P.A. Chou, and R.M. Gray. Variable rate vector quantization for medical image compression. *IEEE Transactions on Medical Imaging*, 9(3):290–298, September 1990.

[60] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing*, volume I & II. Academic Press, San Diego, CA, 2nd edition, 1982.

[61] D. E. Rumelhart and J. L. McClelland, editors. *Parallel Distributed Processing.* MIT Press, Cambridge, MA, 1986.

[62] Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

[63] Terence D. Sanger. An optimality principle for unsupervised learning. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 11–19, 1989.

[64] Terence D. Sanger. Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input. *Biol. Cybern.*, 63:221–228, 1990.

[65] C. Shannon. Coding theorems for a discrete source with a fidelity criterion. In *IRE Natl. Conv Rec.*, pages 142–163, 1959.

[66] Thomas G. Stockham Jr. Image processing in the context of a visual model. *Proc. IEEE*, 60(7):828–842, July 1972.

[67] James A. Storer and Martin Cohn, editors. *Proc. Data Compression Conference*, Snowbird, UT, March 30 - April 2 1993. IEEE Computer Society Press.

[68] M. Tasto and P. A. Wintz. Image coding by adaptive block quantization. *IEEE Trans. Communications*, 19(6):957–972, 1971.

[69] Y. Tateno, T. Linuma, and M. Tkano, editors. *Computed Radiography.* Springer-Verlag, Tokyo, Japan, 1987.

[70] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley, Reading, MA, 1974.

[71] Khoanh K. Truong. Multilayer Kohonen image codebooks with a logarithmic search complexity. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing '91*, pages 2789–2792, Toronto, Canada, May 14-17 1991.

[72] Jaques Vaisey and Allen Gersho. Image compression with variable block size segmentation. *IEEE Trans. Signal Processing*, 40(8):2040–2060, August 1992.

[73] G. K. Wallace. Overview of the JPEG (ISO/CCITT) still image compression standard. In *Proceedings of the SPIE, vol. 1244*, pages 220–233, Feb. 1990.

[74] Stephen Wong, Loren Zaremba, David Gooden, and H. K. Huang. Radiologic image compression — a review. *Proc. IEEE*, 83(2):194–219, February 1995.

[75] Lei Xu and Alan Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. Technical Report 92-3, Harvard Robotics Laboratory, February 1992.

[76] Lei Xu and Alan L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. Neural Networks*, 6(6):131–143, January 1995.