Optimally Adaptive Transform Coding

Robert D. Dony, *Student Member, IEEE,* and Simon Haykin, *Fellow, IEEE*

***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. While the use of adaptation can result in improved performance, there has been little investigation into the optimality of the criterion upon which the adaptation is based. In this paper we propose a new transform coding method in which the adaptation is optimal. The system is 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 the subspace classifier. The performance of the resulting adaptive system is shown to be superior to that of the optimal nonadaptive linear transformation. This method can also be used as a segmentor. The segmentation it performs is independent of variations in illumination. In addition, the resulting class representations are analogous to the arrangement of the directionally sensitive columns in the visual cortex.**

I. Introduction

THE study of image compression methods has been an active area of research since the inception of digital imaging. Since images can be regarded simply as 2-D signals with the independent variables defining a 2-D space, digital compression techniques for 1-D signals can be extended to images in many cases. As a result, a number of approaches to the problem are well established [1]–[5]. In addition, there has recently been an interest in applying neural network approaches to the problem of image compression [6].

For most image compression techniques, the optimal method based on some model of the image statistics is well known. For example, for a pth-order linear autoregressive model, the optimal linear predictor for differential pulse-code modulation (DPCM) can be calculated from the image statistics [4]. The LBG algorithm for vector quantization (VQ) generates an optimal set of codewords in the sense that the average distortion is minimized [7], [8]. For transform coding, the Karhunen–Loève transformation (KLT) is the optimal linear transformation in the sense that it minimizes the mean squared reconstruction error [5].

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 paper proposes a new approach to adaptive transform coding in which the criterion for adaptation is shown to be optimal.

The paper is organized in the following manner: Section II reviews the basic techniques of transform coding with some recently developed approaches based on neural network models. Section III presents the framework for adaptive coding and reviews the subspace method of pattern recognition. The new algorithm for optimally adaptive transform coding is introduced in Section IV. The performance of the new method for compressing images is investigated in Section V, and Section VI presents the results when the new method is used for segmentation. Finally, Section VII provides a discussion of the salient points of this method and concludes the paper.

II. Image Compression

Successful image compression techniques must satisfy two conflicting criteria. During the coding phase of image compression, data are transformed from their native format, typically an array of gray level or trichromatic pixels, into a format that requires less bandwidth or storage. At the same time, this transformation must preserve as much information as possible, so that the difference between the original and decoded images is not significant. The significance of such differences must be evaluated within the context of the end use of the image. For example, medical images *must not lose their diagnostic value* under a compression transformation.

A. Transform Coding

One approach to image compression is the use of transformations that operate on an image to produce a set of coefficients. A simple, yet powerful class of transform coding is linear block transform coding. Under this technique, an image is subdivided into nonoverlapping blocks of $n \times n$ pixels. These image blocks can be considered as N-dimensional vectors $\vec{x}$ with $N = n \times n$. A linear transformation, which can be represented as an $M \times N$-dimensional matrix W with $M \leq N$, is performed on each block with the M rows of W, $\vec{w}_i$, being the basis vectors of the transformation. The

Manuscript received December 7, 1993; revised November 20, 1994. This work was supported by the Natural Sciences and Engineering Research Council of Canada through a Postgraduate Scholarship. The associate editor coordinating the review of this paper and approving it for publication was Prof. William A. Pearlman.

R. D. Dony was with the Communications Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada. He is now with the Department of Physics and Computing, Wilfrid Laurier University, Waterloo, Ontario, Canada N2L 3C5.

S. Haykin is with the Communications Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada L8S 4K1.

IEEE Log Number 9413971.

resulting M-dimensional coefficient vector $\vec{y}$ for each image block is calculated as

$$\vec{y} = W\vec{x}. \tag{1}$$

If the basis vectors $\vec{w}_i$ are orthonormal, then the inverse transformation is the transpose of the forward transformation matrix.

The optimal linear transformation with respect to minimizing the mean squared error (MSE) is the KLT. The transformation matrix W consists of M rows of the eigenvectors corresponding to the M largest eigenvalues of the sample autocovariance matrix

$$\Sigma = E[\vec{x}\vec{x}^T]. \tag{2}$$

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 $O(MN)$ for each image block. Due to these difficulties, fixed basis transforms such as the discrete cosine transform (DCT) [9], 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 [10].

Another 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. As the following section will show, these approaches require less storage overhead and can be more computationally efficient. As well, they may adapt over long term variations in the image statistics.

B. Hebbian Learning

Recently, there has been a tremendous growth in interest in neural networks. 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" [11]. One class of neural network learning algorithms, which is of particular interest to the problem of image compression, uses Hebbian learning to extract the principal components from a data set. Since the principal components are the basis vectors of the KLT, they can be used to construct the optimal linear transform. In 1949, Hebb proposed a mechanism whereby the synaptic strengths between connecting neurons can be modified to effect learning in a neuro-biological network [12]. 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.

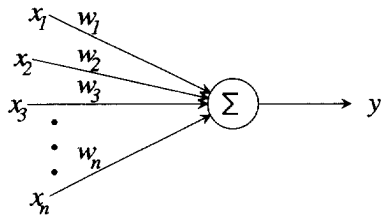


Fig. 1. Simplified linear neuron.

For artificial neural networks, the neural interactions can be modeled as a simplified linear computational unit as shown in Fig. 1. The output of the "neuron," y, is the sum of the inputs $\{x_1, x_2, \cdots, x_N\}$ weighted by the "synaptic strengths" $\{w_{1,2}, \cdots, w_N\}$, or in vector notation

$$y = \vec{w}^T\vec{x}. \tag{3}$$

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 large. Extending this principle to include simultaneous negative values (analogous to inhibitory interactions in biological networks), the weights $\vec{w}$ would be modified according to the correlation between the input vector $\vec{x}$ and the output y. Oja derived a linear Hebbian learning rule for such a simplified neuron model:

$$\vec{w}(t+1) = \vec{w}(t) + \alpha(y(t)\vec{x}(t) - y^2(t)\vec{w}) \tag{4}$$

where t denotes the iteration number. It was shown that the vector $\vec{w}$ converges to the first principal component of the data under this learning rule [13].

Equation (4) has formed the foundation for extending Hebbian learning to simultaneously finding the first M principal components. A simple extension would be to recursively extract the principal components in order. The mth principal component of $\{\vec{x}\}$ can be extracted using (4) by removing the previously computed components through deflation. The recursive application of (4) combined with deflation on $\vec{x}$ results in the computation of the mth principal component. Other approaches have also been developed. Sanger [14] extends the model to compute the M principal components simultaneously by incorporating the deflation into the learning rule. Kung and Diamantaras [15]–[17] propose a recursive solution in which the output of the mth principal component y_m can be calculated based on the previous $m-1$ components through the use of "anti-Hebbian" weights that impose the orthogonality condition. Chen and Liu [18] use a similar network modified to extract M principal components simultaneously from the training data as opposed to recursively. Another approach, taken by Xu and Yuille [19], addresses the problem of robustness in the estimation of the principal components by weighting the training data according to the degree with which each point deviates from the distribution of the training data.

There are a number of advantages that 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 [20]. As well, because of their iterative nature, they can be allowed to adapt to slowly varying changes in the input data. A third advantage is that no extra overhead is required to store the data or its higher-order statistics such as the covariance matrix required for the standard eigendecomposition techniques. Finally, if an extra basis were to be required, its computation would be more efficiently performed using the iterative learning rules.

III. Adaptation

A. Adaptive Processing

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 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 that 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 $\vec{y} = T_{\Omega}(\vec{x})$. If the parameter set were to vary according to the neighborhood around a given data point, $N_{\vec{x}}$, then the transformation can be allowed to adapt to the characteristics of the surrounding data. The transformation can then be represented as

$$\vec{y} = T_{\Omega(N_{\vec{x}})}(\vec{x}). \tag{5}$$

To simplify matters, the statistical variations can be quantized into a finite number of classes. Image pixels are then classified as belonging to one of the classes $\{C_1, C_2, \cdots, C_K\}$. There is a corresponding parameter set $\boldsymbol{\Omega} = \{\Omega_1, \Omega_2, \cdots, \Omega_K\}$, where each element Ω_i describes the characteristics of the corresponding class C_i. The transformation is then represented as

$$\vec{y} = T_{\Omega_i}(\vec{x}), \quad \vec{x} \in C_i. \tag{6}$$

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 [1], [21]–[23], [5]. In some of the earlier work, the adaptation occurs in the quantization stage while the transformation remains fixed [24], [25]. This approach has also been explored in more recent work [26]. Adaptation has also been applied to VQ methods [27]–[29]. 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 the 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 a recursive manner, similar data are grouped together in classes and the resulting representation of each class is used to then reclassify 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.

B. 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 [30]. 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, 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 Euclidean 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 that 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 that define the subspace implicitly define the features of the data set [20]. 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 $\vec{x} \in \mathbb{R}^N$, and $U \in \mathbb{R}^{M \times N}$ is an orthonormal matrix with $M < N$, then the projector P is defined as

$$P = U^T U \tag{7}$$

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

$$\hat{\vec{x}} = P\,\vec{x}. \tag{8}$$

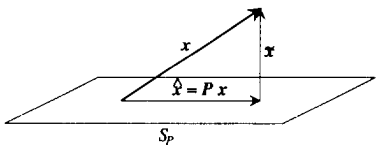


Fig. 2. Projection of $\vec{x}$ by P on S_P.

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

$$S_P = \{\vec{z} | \vec{z} = P\vec{x}, \vec{x} \in \mathbb{R}^N\} \tag{9}$$

and is spanned by the M N-dimensional row vectors of U.

To adequately represent the data, the subspace should match the data as closely as possible. Referring to Fig. 2, this means that the expected norm of the projected vector is maximized, i.e., maximize

$$E[\|P\vec{x}\|]. \tag{10}$$

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

$$E\left[\|\tilde{\vec{x}}\|^2\right] = E\left[\|\vec{x} - \hat{\vec{x}}\|^2\right]. \tag{11}$$

In other words, maximizing the expected norm of the projection is equivalent to finding the transformation that minimizes the MSE. As stated earlier, the linear transformation that 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 that are defined by K subspaces $\{S_1, S_2, \ldots, S_K\}$. Each subspace S_i is defined by its projector P_i, which can be calculated using (7) with the rows of U being the M principal components of the class data. Once the classes are defined, a data vector $\vec{x}$ is assigned to the class under whose projection its norm is maximized:

$$\vec{x} \in C_i \text{ if } \|P_i\vec{x}\| = \max_{j=1}^{K} \|P_j\vec{x}\|. \tag{12}$$

Since the use of (12) 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. However, without knowing *a priori* the required classes, their defining projectors P_i, and their corresponding transformation bases, a learning algorithm is required to extract the appropriate parameters from the dataset.

IV. Optimally Integrated Adaptive Learning

A. OIAL Algorithm

A new class of unsupervised learning algorithms is proposed that 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.

The general form of the class of optimally integrated adaptive learning (OIAL) algorithms, which is a generalization of that previously presented in [31], is as follows:

1) Initialize K transformation matrices $\{W_1, W_2, \cdots, W_K\}$.
2) For each training input vector $\vec{x}$:
   a) classify the vector based on the subspace classifier

$$\vec{x} \in C_i \text{ if } \|P_i\vec{x}\| = \max_{j=1}^{K} \|P_j\vec{x}\| \tag{13}$$

   where $P_i = W_i^T W_i$, and
   b) update transform matrix W_i according to

$$W_i = W_i + \alpha Z(\vec{x}, W_i) \tag{14}$$

   where α is a learning parameter, and $Z(\vec{x}, W_i)$ is a learning rule that converges to the M principal components of $\{\vec{x} | \vec{x} \in C_i\}$.
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 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 set 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 that we have used in the experimental section of this paper.

Algorithms based on the above outline will produce K transformation matrices $\{W_1, W_2, \cdots, W_K\}$. Given the appropriate learning rule $Z(\vec{x}, W_i)$ in (14), each matrix will converge to the KLT for that particular class of data. Since the KLT minimizes the mean squared error, each W_i is optimal for its class. The classification rule in (13) is equivalent to finding the transformation that 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 [11]. However, in all the experience the authors have had with the algorithm on "real" data, the algorithm has consistently converged to a satisfactory result every time.

Since there exists a number of learning rules that compute the M principal components of a data set, the choice of $Z(\vec{x}, W_i)$ will depend on the desired computational efficiency and convergence properties. Whether the learning rule used is the linear Hebbian rule of (4) with recursive calculation of the M principal components, or the others mentioned in Section II-B, [14]–[16], [32], [18], [19], 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 P_i would be identical and the performance of the coding and decoding transformations in terms of MSE would also remain unchanged. Therefore, the use of learning rules based on auto-associative backpropagation as proposed by Cottrell [33], [34] or linear gradient descent methods like that proposed by Russo and Real [35] would also result in an optimal set of transformations.

Since the purpose of this paper 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 our present study is the Generalized Hebbian Algorithm (GHA) devised by Sanger [14].

B. System Architecture

Fig. 3 shows the modular architecture for the coding stage of the system. The system consists of a number of independent modules whose outputs are mediated by the subspace classifier. Each module consists of M basis images of dimension $n \times n$ that defines a single linear transformation. The inner product of each basis image with the input image block results in M coefficients per module, represented as an M-dimensional vector $\vec{y}_i$. 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 $\|\hat{\vec{x}}_i\|$ is maximum. The projected vector $\hat{\vec{x}}_i$ is calculated by taking the inverse transformation of the coefficient vector.

The message 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 $\hat{\vec{x}}$ is calculated.

The system efficiently represents both the linear transformation and the classification criterion. The same set of bases 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 system requires no extra overhead in terms of information required to effect the adaptation.

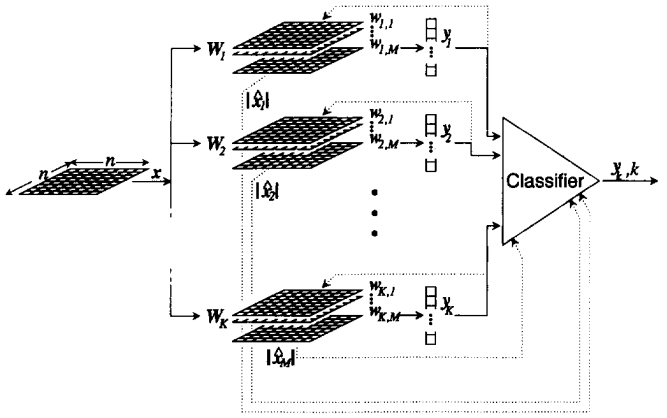


Fig. 3. Modular system architecture of OIAL. Input are blocks of $n \times n$ pixels. The K transformations W_i consist of M basis images of size $n \times n$ and output an M-dimensional vector $\vec{y}_i$. The coefficient vector to be sent is chosen by the subspace classifier based on the maximum norm of the projected vector $\|\hat{\vec{x}}_i\|$.

V. Compression

A. Method

To evaluate the performance of the optimally integrated adaptive learning class of algorithms, a set of experiments were performed. As mentioned in Section IV-A, the learning rule chosen was the GHA. The learning parameter α in (14) for the ith component at iteration k was calculated as

$$\alpha_i(k) = \left(\sum_{l=0}^{k} \gamma^{k-l} y_i^2 \right)^{-1} \tag{15}$$

from [16] where γ is analogous to the "forgetting factor" in the adaptive recursive least squares (RLS) algorithm [36]. For the results presented herein, γ was chosen to be $\gamma = 0.995$. The set of transformation 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.

Fig. 4 shows the magnetic resonance image (MRI) used for training. The image in Fig. 5 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 transformation was also calculated based on the same training data.

Fig. 4. MR image for training.

Fig. 5. MR image for testing.

A typical learning curve for a system with 4 coefficients and 128 classes is shown in Fig. 6. Each point represents an average MSE over 10 samples to reduce the block-to-block variation in MSE. 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 Fig. 7. The same set of initial transformation matrices was

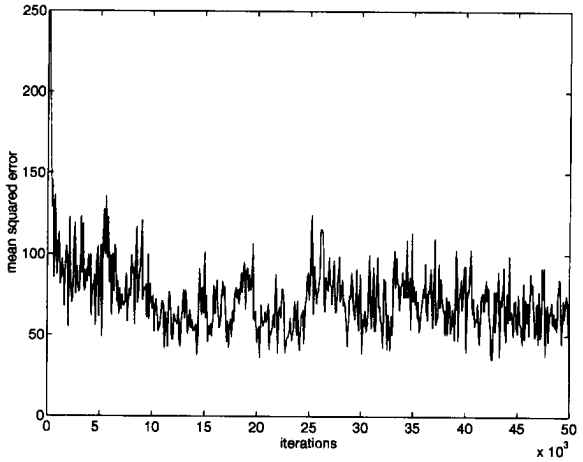


Fig. 6. Typical learning curve for system with four coefficients and 128 classes.

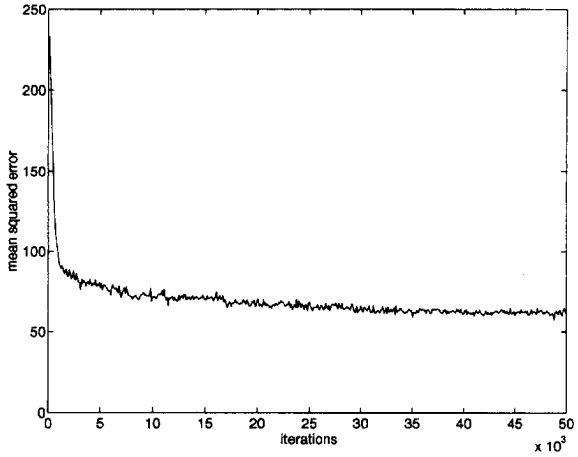


Fig. 7. Ensemble average of 100 learning curves.

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 three to four iterations through the entire training set for this configuration.

The test image was divided into 8×8 nonoverlapping blocks. These blocks were transformed by the previously computed system into a set of coefficients, quantized, and then transformed back into image blocks. Three bit rates were used in the quantization step: 0.625 bits per pixel (b/p) for a compression ratio of 12.8:1, 0.5 b/p for a 16:1 ratio and 0.375 b/p for a ratio of 21.33:1. These bit rates included the side information necessary for each block to convey its class membership. The coding of class information was not optimized. The optimal nonuniform Max quantizer was used with the first coefficient being modeled as a uniform distribution and the others as Laplacians. The estimate of the variances for each coefficient was taken from the training data. Each class was allocated the same number of bits per image block. For both the adaptive approach and the KLT, the number of bits per coefficient were optimally assigned so as to minimize the quantization error [9]. For the KLT, this bit allocation resulted in only a subset of the 64 coefficients for each 8×8 block having nonzero values after quantization. For example, the 0.5 b/p case used only eight coefficients.

TABLE I
CODING DISTORTION FOR VARIOUS OIAL SYSTEM CONFIGURATIONS AS COMPARED WITH KLT

No. Coef.	No. Class	MSE 0.625 bpp	0.5 bpp	0.375 bpp
2	64	95.90	95.90	95.90
2	128	82.91	82.91	82.91
2	256	78.22	78.22	78.22
2	512	70.07	70.07	70.10
4	16	74.85	74.97	81.67
4	32	64.11	64.49	75.91
4	64	58.86	59.50	73.10
4	128	53.72	54.44	71.58
8	16	47.80	67.83	118.10
8	32	45.10	65.15	115.17
8	64	44.70	66.60	122.14
8	128	48.32	72.32	125.57
KLT:		78.92	98.60	135.48

B. Results

Table I shows the MSE for the OIAL algorithm for coding rates of 0.625, 0.5, and 0.375 bits per pixel. Also shown is the MSE for the KLT for the same bit rates.

For most OIAL system configurations as shown in Table I, the use of adaptation has resulted in a reduction in mean squared error over the nonadaptive KLT. When the number of coefficients is small (e.g., 2), there is an improvement for high compression ratios, but for some cases at lower compression, there is a reduction in performance. In these cases, the number of coefficients is inadequate to sufficiently represent each class. As the number of coefficients increases, the mean squared error decreases as shown by comparing the results for two and four coefficients. However, there is a limit to the improvement realized through increasing the number of coefficients alone because of the resulting increase in quantization error. For example, at both 0.5 b/p and 0.375 b/p, doubling the number of coefficients from 4 to 8 for the same number of classes, 128, increases the mean squared error since fewer bits are available to code the coefficient values.

When the number of coefficients is fixed, (e.g., at 4), increasing the number of classes can improve performance. Since the degree of adaptivity is directly related to the number of classes, this decrease in MSE clearly demonstrates the advantage of using a locally adaptive coding scheme over a nonadaptive method. Again, there is a limit to the number of classes that can be used. When the number of coefficients is 8, an increase in the number of classes actually decreases performance at higher compression ratios. In this case, the resulting increase in the number of bits required to represent the class membership information decreases the number of bits remaining to represent the coefficient values. This results in an increase in quantization error.

Fig. 8 shows the distortion versus compression ratio calculated from the test data for the KLT and OIAL compression methods. For the OIAL data, the system configuration that results in the minimum squared error value for each compression ratio is used in the comparison. For the same coding rate, the use of OIAL decreases the MSE by 40–50% over the KLT. If the acceptable distortion was fixed at a MSE of 60, for example, the compression ratio could be improved from 10:1 for the KLT to 19:1 for the OIAL.

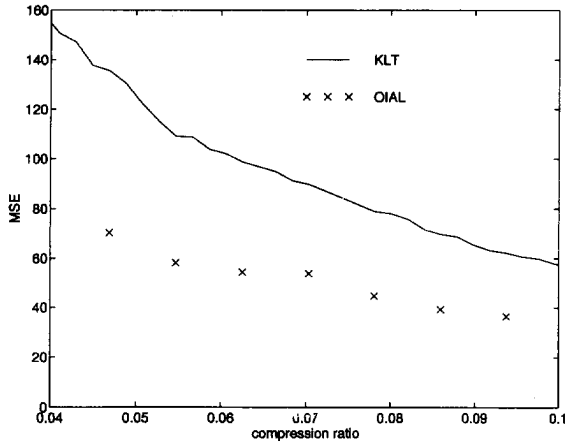


Fig. 8. Distortion versus compression ratio for OIAL and KLT compression.

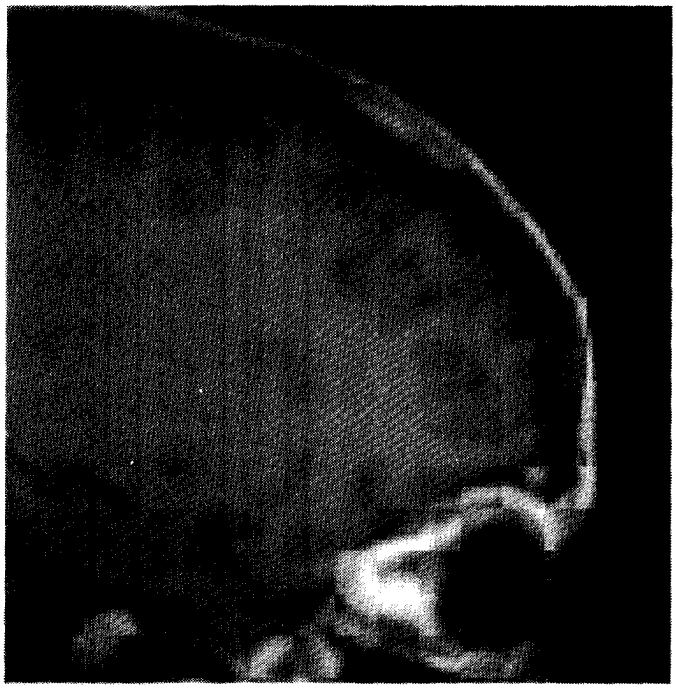


Fig. 9. Details of OIAL coding with 128 classes, four coefficients per block, at 0.5 b/p.

Although performance measures based on squared error provide a quantitative measure of performance and are easily computed, they are no substitute for a qualitative comparison. Fig. 9 shows details of the resulting image for a coding rate of 0.5 b/p using the OIAL algorithm with four coefficients and 128 classes. For comparison, Fig. 10 shows the corresponding details using the KLT at the same rate of 0.5 b/p. 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 that are not present in the original nor in

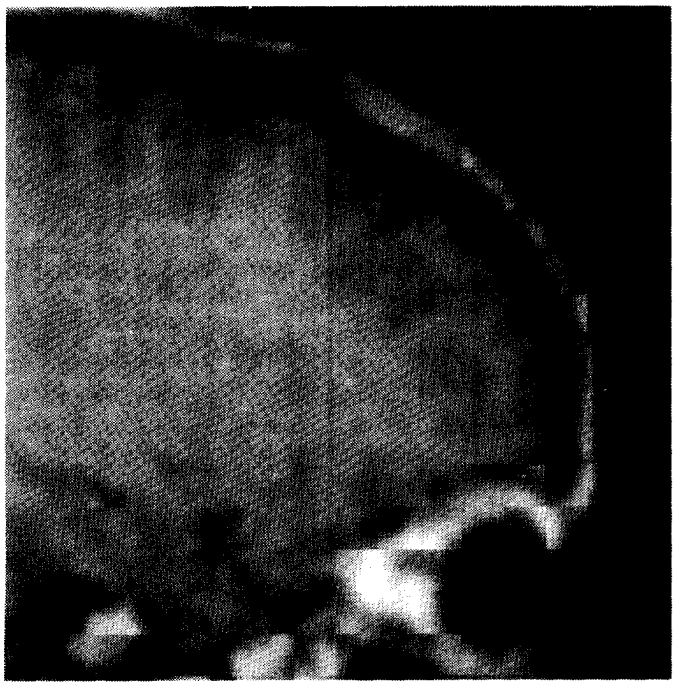


Fig. 10. Details of KLT coding at 0.5 b/p.

Fig. 11. Lenna image for testing generalization.

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. Although techniques exist that can reduce the block effects for block transform coding methods, they were not used so that the differences between the two methods are more clearly shown.

C. 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, we have trained and tested 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. That is, 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 x-ray, 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 we do not claim that there exists a single network configuration that 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. Fig. 11 shows the Lenna image, which is obviously quite different from the image used for training as shown in Fig. 4. Fig. 12 shows the resulting compressed image using the same network (four coefficients and 128 classes) and bit rate (0.5 b/p) as that used for the image shown in Fig. 9. The MSE for this image is 54.9. For comparison, the image was compressed using the KLT of *itself* and quantized to the same number of bits (eight coefficients with 0.5 b/p). The resulting image is shown in Fig. 13 and has a MSE of 71.0. 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 Figs. 9 and 10, 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.

VI. Segmentation

A. Image Formation Model

One very useful property from an image processing perspective of the subspace method of classification is that the classification is independent of the norm of the data vector $\vec{x}$, i.e., for any scalar multiple α, if $\vec{x} \in C_i$ then $\alpha\vec{x} \in C_i$. This is a significant property because it is known that the image formation process can be modeled as a multiplicative process [37]

$$L(x, y) = E(x, y)\rho(x, y) \tag{16}$$

where L is the luminance of the formed image, E is the illumination falling on a scene, and ρ is the reflectance. A similar model is valid for images formed via transmission as well as reflection. If the illumination were to vary much

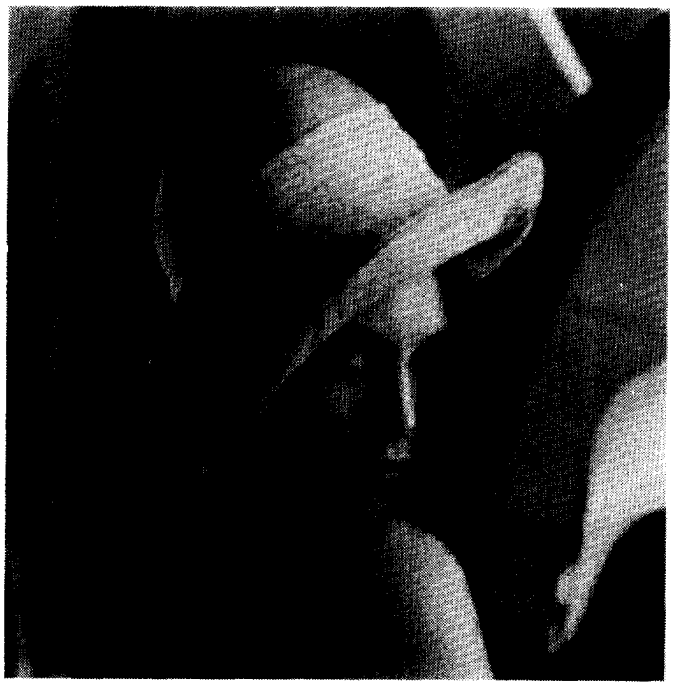


Fig. 12. Lenna image with OIAL coding, 128 classes, four coefficients per block, 0.5 b/p.

Fig. 13. Lenna image with KLT coding, 0.5 b/p.

more slowly than the reflectance, a valid assumption for most images, then for a small neighborhood $N(x,y)$, (16) can be rewritten as

$$L(x,y) = E_{N(x,y)}\rho(x,y). \tag{17}$$

Typically, the goal in image analysis is to determine characteristics about the underlying physical properties of the scene being imaged. These are inferred from the reflectivity of the scene. Therefore, it is the reflectivity that conveys the information about the scene, and any variations in the illumination can be considered as noise. This is the justification for a class of image processing called "homomorphic processing" [38].

Using vector notation to represent the luminance values of a set of neighboring pixels as $\vec{x}$, the image of one region, or feature, is formed as

$$\vec{x}_1 = E_1\vec{\rho}_1 \tag{18}$$

which is the vector equivalent of (17). If the same feature, having the same reflectance, $\vec{\rho}_1$, were to appear elsewhere under different illumination conditions, E_2, its image would be $\vec{x}_2 = E_2\vec{\rho}_1$. If some image analysis process were performed on these image vectors, one would expect the same result since both $\vec{x}_1$ and $\vec{x}_2$ were created by the same underlying feature $\vec{\rho}_1$. For many classical pattern recognition approaches, this would not be the case since they 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 $\vec{x}_1$ and $\vec{x}_2$ may be quite large and result in the codeword representations of the two vectors being different. Subspace methods, however, would treat the two vectors identically since they would both project to the same subspace independently of the illumination values E_1 and E_2. It could be argued, then, that subspace 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 [39]. We have no problem in recognizing that the 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. The linear subspace classifier has exactly this property.

B. Class Representation

As shown in Section III-B, the optimal representation of a class in terms of maximizing the "within class similarity" is the subspace spanned by the M principal components of the class data. With labeled data, the eigenvectors of the class covariance matrices can be calculated and used to form the projection matrices defining the linear subspaces for each class. Similarly, iterative techniques such as those based on Hebbian learning can also be used to calculate the principal components. Alternatively, linear minimization methods such as gradient descent can be used to find the optimal projection matrix without the explicit calculation of the eigenvectors.

Without labeled data, the problem of determining the appropriate classes and their respective linear subspaces is akin to the problem of clustering in classical pattern recognition theory. The OIAL class of learning algorithms as introduced in Section IV-A 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.

In some applications, it may be advantageous to have some similarity between "neighboring" classes. Kohonen [40] 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 $\vec{x}$ is classified and only the "winning" class is modified during each iteration. In Kohonen's self-organizing feature map (SOFM), the vector $\vec{x}$ is used to update not only the winning class, but also its neighboring classes. Each training vector $\vec{x}$ is classified according to the minimum Euclidean distance between it and the set of class feature vectors $\{\vec{m}_i\}$. The feature vectors of winning class and its neighboring classes are modified according to their respective vector differences with the input vector. The neighborhood 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 2-D square grid, the neighborhood 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 neighborhood may be quite large during training, e.g., half the number of classes or more. As the training progresses, the size of the neighborhood shrinks until, eventually, it only includes the one class. During training, the learning parameter α also shrinks.

C. Self-Organizing Segmentor

The concept of topologically ordered classes can be incorporated into the OIAL class of algorithms. Referring to the algorithm presented in Section IV-A, the step for updating the transformation bases (namely (14)), can be modified to

$$W_j = \begin{cases} W_j + \alpha Z(\vec{x}, W_j), & C_j \in N_{C_i} \\ W_j, & C_j \notin N_{C_i} \end{cases} \tag{19}$$

where C_i is the winning class according to the classification rule of (13), and N_{C_i} is the set of classes that are in the neighborhood of class C_i.

As with Kohonen's network, we start with a large neighborhood initially. The large neighborhood size allows each class to be affected by a large number of training data vectors. Since the data would be fairly representative of the entire training set, each class will begin to converge to the globally optimal representation of the data, irrespective of the initial conditions. This helps reduce the possibility of the initial conditions allowing null classes to form. Null classes can form when some initial class representations fall outside the range of the possible data values. As the neighborhood size shrinks, the globalizing effect reduces and the differences among the classes become enhanced. Connected groups of classes become tuned to the variations within the data set as the training data have a more regional effect on the topology. The effect of individual data points becomes more and more localized until, eventually, each training data point updates only one class. The result is a set of classes in which the similarity between classes is correlated with the distance between them.

Another advantage of such a topological arrangement is the ease with which new classes can be incorporated within an existing set of transformations. A new class can be added by inserting it between two existing classes and initializing it to the union of the neighboring classes. Unlike other techniques that increase the number of classes by splitting existing classes, this technique preserves the existing classes when new classes are added. In addition, it removes the problem of how best to split an existing class.

D. Results

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 that is used for the following investigation.

A system consisting of 32 classes with two coefficients per class was trained with the training data described in Section V-A using the updating rule of the self-organizing segmentor, (19). The initial neighborhood size was $3/4$ the size of the entire system or 24 classes. The size of the neighborhood decreased by two classes for each iteration through the training set.

Once the system was trained, it was used to segment the test image of Fig. 5. 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 color with the color of a class i being the hue at an angle of $i/K \times 360^\circ$ on a color circle, where K is the total number of classes that in this case is $K = 32$. The intensities were weighted by the value of the second coefficient for each block. Fig. 14 shows the resulting class map.

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. 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 [41]. Even when one looks at the images in Figs. 4 and 5, the areas to which one's attention is initially drawn corresponds to areas that the segmentation network has represented as being important, namely, edges and lines. This extraction of important features was accomplished entirely through a self-organizing mechanism.

The continuity of the color transitions shows the high degree of similarity between neighboring classes. Since the class indices were coded as a spectrum of colors, similar colors 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 colors 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

Fig. 14. Segmentation map of test image with 32 classes, two coefficients per class. Color indicates class membership, intensity is weighted by value of the second coefficient for each block.

features around the top and bottom of the orbit are mapped to the same classes as the horizontal features at the top of the skull.

Fig. 15 shows the second coefficient basis images of the system. Since the first coefficient in every class is approximately the dc component, they were not shown. The basis images were color coded to match the class assignments shown in Fig. 14. Again, this figure clearly 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 images shows how the orientations of the classes progress 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 stimuli using microelectrodes [42]. 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 Fig. 15, the above characteristics of the visual cortex are mimicked by the set of bases. Each basis image 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 Fig. 4 contains more vertical and diagonal features than horizontal ones. This nonuniform 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.

VII. Discussion and Conclusions

A new approach to adaptive compression is proposed in this paper, based on an optimally integrated adaptive learning

Fig. 15. Map of second coefficient basis images for 32-class, two-coefficient network. The class number progresses left to right, top to bottom.

(OIAL) class of algorithms. The architecture for such a system consists of a number of modules, each consisting of a number of basis images. 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 that 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 overall 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 or relationships of 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 adaptation and resulting representation are independent of variations in illumination over an image since the subspace classifier is insensitive to the vector norm of the data.

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 MSE was reduced and the image quality was improved. Also, more image details were preserved and fewer artifacts were introduced.

The use of the new method as a segmentor has great potential. In the results presented in Section VI, the system extracted perceptually important features from the test image in a completely self-organizing fashion. The classification of similar features was consistent across the entire test image. The use of a topological ordering of the classes during training resulted in similar classes being close together in a manner analogous to the ordering of directionally sensitive columns in the visual cortex.

While most of the results in this paper have been restricted to one particular imaging mode, namely, MRI, there is nothing to suggest that there should not be similar improvements for images from other classes. As long as both the training and test data come from the same population of images with similar distributions of data, similar results should be attained. The optimality of the adaptation mechanism means that the use of this technique will result in a minimum mean squared error distortion.

ACKNOWLEDGMENT

The authors wish to thank Dr. G. Cottrell and the anonymous reviewers for their constructive input and recommendations for improvements to the paper.

REFERENCES

[1] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977.

[2] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366–406, Mar. 1980.

[3] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349–389, Mar. 1981.

[4] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed, vol. I, II. San Diego, CA: Academic, 1982.

[5] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*. New York: Plenum, 1988.

[6] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, pp. 288–303, Feb. 1995.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[8] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, pp. 957–971, Aug. 1988.

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

[10] G. K. Wallace, "Overview of the JPEG (ISO/CCITT) still image compression standard," in *Proc. SPIE*, Feb. 1990, vol. 1244, pp. 220–233.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[12] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[13] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, pp. 267–273, 1982.

[14] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 459–473, 1989.

[15] S. Y. Kung and K. I. Diamantaras, "A neural network learning algorithm for adaptive principal component extraction (APEX)," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, NM, Apr. 3–6, 1990, pp. 861–864.

[16] K. I. Diamantaras, "Principal component learning networks and applications," Ph.D. dissertation, Princeton Univ., Princeton, NJ, Oct. 1992.

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

[18] H. Chen and R. Liu, "Adaptive distibuted orthogonalization process for principal components analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing '92*, San Francisco, Mar. 23–26, 1992, pp. II 283–296.

[19] L. Xu and A. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," Harvard Robotics Lab., Tech. Rep. 92-3, Feb. 1992.

[20] E. Oja, *Subspace Methods of Pattern Recognition*. Letchworth, UK: Research Studies, 1983.

[21] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.

[22] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE*, vol. 73, pp. 549–574, Apr. 1985.

[23] M. Kunt, M. Bénard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 1306–1336, Nov. 1987.

[24] M. Tasto and P. A. Wintz, "Image coding by adaptive block quantization," *IEEE Trans. Commun.*, vol. COM-19, pp. 957–972, 1971.

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

[26] A. Baskurt and I. Magnin, "Adaptive coding method of x-ray mammograms," in *Proc. SPIE: Image Capture, Formatting, Display*, San Jose, CA, Feb. 24–26, 1991, vol. 1444, pp. 240–249.

[27] E. A. Riskin, T. Lookabaugh, P. A. Chou, and R. M. Gray, "Variable rate vector quantization for medical image compression," *IEEE Trans. Med. Imaging*, vol. 9, pp. 290–298, Sept. 1990.

[28] T.-C. Lee and A. M. Peterson, "Adaptive vector quantization using a self-development neural network," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1458–1471, Oct. 1990.

[29] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Processing*, vol. 40, pp. 2040–2060, Aug. 1992.

[30] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[31] R. D. Dony and S. Haykin, "Optimally integrated adaptive learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, Apr. 27–30, 1993, pp. I 609–612.

[32] T. D. Sanger, "An optimality principle for unsupervised learning," in *Advances in Neural Information Processing Systems 1*, D. S. Touretzky, Ed. 1989, pp. 11–19.

[33] G. W. Cottrell, P. Munro, and D. Zipser, "Learning internal representations from gray-scale images: An example of extensional programming," in *Proc. 9th Annu. Conf. Cognitive Soc.*, July 16–18, 1987, pp. 462–473.

[34] G. W. Cottrell and P. Munro, "Principal components analysis of images via back propagation," in *Proc. SPIE: Visual Commun. Image Processing '88*, 1988, vol. 1001, pp. 1070–1077.

[35] L. E. Russo and E. C. Real, "Image compression using an outer product neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing '92*, San Francisco, Mar. 23–26, 1992, pp. II 377–380.

[36] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[37] T. G. Stockham, Jr., "Image processing in the context of a visual model," *Proc. IEEE*, vol. 60, pp. 828–842, July 1972.

[38] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[39] T. N. Cornsweet, *Visual Perception*. New York: Academic, 1970.

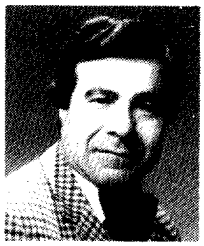
[40] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, Sept. 1990.

[41] D. Marr, *Vision*. New York: W. H. Freeman, 1982.

[42] D. H. Hubel and T. N. Wiesel, "Brain mechanisms of vision," *Sci. Am.*, pp. 130–146, Sept. 1979.

Robert D. Dony (S'92) received the B.A.Sc. and M.A.Sc. degrees in 1986 and 1988, respectively, from the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada. In 1995, he received the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada.

In 1988, he joined Imaging Research Inc., where he was Manager of Imaging Development. Presently, he is with the Department of Physics and Computing, Wilfrid Laurier University, Waterloo, ON, Canada. His research interests include image processing, image compression, neural networks, and both artificial and biological learning systems.



Simon Haykin (F'82) received the B.Sc. degree with first-class honors in 1953, the Ph.D. degree in 1956, and the D.Sc. degree in 1967 from the University of Birmingham, UK, all in electrical engineering.

He is the Founding Director of the Communications Research Laboratory and Professor of Electrical and Computer Engineering at McMaster University, Hamilton, ON, Canada. His research interests include nonlinear dynamics, neural networks, and adaptive filters and their applications. He is the editor of *Adaptive and Learning Systems for Signal Processing, Communications and Control*, a new series of books for Wiley-Interscience.

In 1980, Dr. Haykin was elected Fellow of the Royal Society of Canada. He was awarded the McNaughton Gold Medal, IEEE (Region 7), in 1986.