A Mixture of Local PCA Learning Algorithm for Adaptive Transform Coding

Bai-ling Zhang[1], Qian Huang[2] and T.D.Gedeon[1]
[1] Department of Information Engineering
School of Computer Science and Engineering
The University of New South Wales
Sydney 2052 Australia
[2] Department of Electrical Engineering
The South China University of Technology
Guangzhou 610641 P.R.China

Abstract *Karhunen-Loeve transform (KLT) is the optimal linear transform for coding images under the assumption of the stationarity. For images composed of regions with widely varied local statistics, Dony and Haykin proposed a new transform coding method called optimally integrated adaptive learning (OIAL), in which a number of localized KLTs are adapted to regions with roughly the same statistics. The new transform coding method is shown to be superior to the tranditional KLT. However, the performance of OIAL depends on an estimate of the global principal components of the data, which is not only computationally expensive but also impractical in some cases. Another problem of OIAL is that the mean vector in each region is not taken into account, which is required to define a local PCA. In this paper, we proposed an improvement over the OIAL which replaces the winner-take-all (WTA) based clustering by an optimal soft-competition learning algorithm called 'neural gas'. The mean vector in each region is also incorporated. Experiments show a better performance over OIAL.*

1. Introduction

Because digital images require large amount of data to represent, image compresson is needed in order to store and transmit images economically. Many image compression techniques and standards have been proposed. Artificial neural networks have been applied in the area of digital signal processing and more recently in data compression for image coding systems. Neural network models have shown great capability in providing redundancy reduction and data compression of sensory data. This capability can be exploited to perform image coding in two different ways: (a). Transform coding of the input by energy preserving transformation that pack maximum information on a minimum number of samples. (b). Vector quantization of the input data by taking advantage of image redundancy to define a code-book for image blocks.

Among the transform coding techniques, the Karhunen-Loeve transform (KLT) is considered to be better than other linear transforms in the mean square error sense. This is because it employs the second-order statistics of the input data. However, KLT suffers from two problems: it requires large computation effort, and the covariance matrix of the input data might be singular or near singular. Assuming the data can be modelled as a first-order stationary Markov model, the discrete cosine transform (DCT) has been shown to perform as well as the Karhunen-Loeve transform. Because the DCT can be implemented as a fast algorithm, the computational complexity of the KLT is highly reduced. But since not all images can be modelled as a Markov model, a better way to implement the KLT on such image data has to be found [4]. In [5], a neural model approach to perform adaptive calculation of the principal components of the covariance matrix is proposed.

However, the assumption upon which the condition for optimality has 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. Dony and Haykin pro-

0-7803-5871-6/99/$10.00 © 1999 IEEE

posed a new approach to adaptive transform coding based on a mixture of local principal component projections, called optimally integrated adaptive learning (OPAL) [7] which involves two procedures: partition a data set into a number of nonoverlapping regions and each region is represented not by its central point as in clustering but by a localized linear subspace. Their results demonstrated that the OIAL method outperforms the globally optimal linear transform (KLT).

The performance of Dony and Haykin's algorithm seriously depends on the initial condition. It was required that the initial set of transformation matrices should be representative of the distribution space of the training data, otherwise the resulting partition would be suboptimal due to the "underutilization" problem. In [7], an estimate of the global principal components of the data was first made and then a small amount of random variation was added to each class. On the other hand, the definition of OIAL is intuitive in the sense that the centroid vector in each region, which is required to define a local PCA, is ignored. To overcome the problems, in this article we propose an improved mixture model in which the data manifold is partitioned by generalizing an optimal clustering algorithm called 'neural gas' [8]. To guarantee the optimality of the mixture model, we explicitly calculate the mean vector in each region and incorporate it into the coding and decoding process.

2. A Mixture Model of Local Principal Component Representation

Consider a random variable $\mathbf{x}$ in R^L with finite covariance matrix Σ. Without loss of generality, $\mathbf{x}$ is assumed to have zero mean. The r principal components of $\mathbf{x}$'s distribution are the r orthogonal directions in R^L that capture the greatest variation in the distribution. A linear neuron model with weight vector $\mathbf{w}$, input sample $\mathbf{x}$ and output $y = \mathbf{w}^T\mathbf{x}$ can learn the largest principal component [1-2], which can also be achieved by optimizing the criterion

$$\text{minimize } J = E[y^2] \qquad (1)$$

where E stands for expectation. An additional constraint such as $\|\mathbf{w}\| = 1$ is necessary to stabilize the learning rule derived from eqn (1). A stochastic approximation solution of (1) leads to the famous Oja's rule

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y\mathbf{x} - y^2\mathbf{w}_t) \qquad (2)$$

where μ_t is the learning rate at iteration t.

A number of unsupervised learning algorithms for extracting multiple principal components or their subspace have been proposed, usually developed from the objective (1) and the following consideration: the second largest principal component also statisfies the minimal reconstruction property with restriction that the second principal component direction must be orthogonal to the first component direction, and so on for the remaining principal component directions. Among many efficient learning algorithms, the Generalized Hebbian Algorithm (GHA) [3] is well-known. For a single-layer network with M linear output units, $L \times M$ matrix $W = [\mathbf{w}(1), \cdots, \mathbf{w}(M)]$ transforms input $\mathbf{x}$ to output $\mathbf{y} = [y_1, \cdots, y_M]^T$, $y_m = \mathbf{w}^T(m)\mathbf{x}$. The learning rule based on GHA is

$$W_{t+1} = W_t + \mu_t(\mathbf{y}\mathbf{x}^T - LT[\mathbf{y}\mathbf{y}^T]W_t) \qquad (3)$$

where LT stands for an operator that sets all elements on or above the diagonal of its matrix argument to zero.

The PCA or subspace method provides a continuous distributed representation. Although important, an inherent weakness of only global linear transform prevents its further applications. In data analysis, clustering or vector quantization (VQ) techniques provide a nonlinear discrete representation, which use a number of local Voronoi centers to represent input vectors. For a set of M reference vectors, $\{\mathbf{v}(1), \cdots, \mathbf{v}(M)\}$, an input vector $\mathbf{x}$ is considered being best matched by one of its reference vector $\mathbf{v}(k)$ in the sense that an appropriately defined distortion measure such as the squared Euclidean distance $\|\mathbf{x} - \mathbf{v}(k)\|^2$ is minimal.

The 'neural gas' algorithm proposed in [8] is an efficient method for solving VQ. In a neural gas model, reference vectors $\mathbf{v}(m)$ are associated with connection weights of neural units and adapted by the relative distances between the neural units within the input space. Each time an input $\mathbf{x}$ is presented, we first make an ordering of the elements of a set of distortions $E_{\mathbf{x}} = \{\|\mathbf{x} - \mathbf{v}(m)\|, m = 1, \cdots, M\}$ and then determine the adjustment of reference vector $\mathbf{v}(m)$. For a given data vector $\mathbf{x}$, we determine the "neighborhood-ranking" ($E_{\mathbf{x}}(m_0)$, $E_{\mathbf{x}}(m_1)$, $\cdots$, $E_{\mathbf{x}}(m_{M-1})$) of the distortion set, which means $\mathbf{v}(m_0)$ is closest to $\mathbf{x}$, $\mathbf{v}(m_1)$ second closest to $\mathbf{x}$, $\mathbf{v}(\mathrm{m_k})$, $\mathrm{k} = 0, \cdots, \mathrm{M} - 1$, the reference vector for which there are k vectors $\mathbf{v}(\mathrm{j})$ with $\|\mathbf{x} - \mathbf{v}(j)\| < \|\mathbf{x} - \mathbf{v}(m_k)\|$. Then each neuron adjusts its weight via a dynamical learning rate which depends on the ranking of its representation capability. Denote the number k associated with each neural unit m by k_m. The following learning rule is the simple neural 'gas' algorithm in [8].

$$\begin{aligned} \mathbf{v}_{t+1}(m) &= \mathbf{v}_t(m) + \mu_t h_\lambda(k_m)(\mathbf{x} - \mathbf{v}_t(m)) \\ m &= 1, \cdots, M \end{aligned} \quad (4)$$

where μ_t is the learning rate, $h_\lambda(k_m)$ is 1 for $k_m = 0$ and decays to zero for increasing k_m with a characteristic decay constant. Algorithm (4) has a number of advantages over other clustering algorithms, including fast convergence and very small distortion errors [8].

While PCA provides a global, linear transform of the data, clustering or VQ offers a local, nonlinear mapping between the data and the representation. In practice, these two basic forms of data representation can be combined in an appropriate way to establish some kind of nonlinear distributed representations. A mixture model of local PCA is such a combination, which partitions the data set into a number of K regions and each region C_k is represented by a respective M_k-dimensional linear subspace $\mathcal{L}^{(k)}$. In other words, each input vector is assigned to the most appropriate partition and then represented by the M_k basis vectors of the region. Specifically, this representation can be expressed as

$$\mathbf{y}^{(k)} = \mathbf{W}^{(\mathrm{k})^T}(\mathbf{x} - \bar{\mathbf{x}}^{(k)}), \text{ if } \mathbf{x} \in C_k, \ k = 1, \cdots, K \quad (5)$$

where $\mathbf{W}^{(\mathrm{k})}$ is an $L \times M_k$ matrix whose columns are the M_k principal components of the partition C_k, $\bar{\mathbf{x}}^{(k)}$ is the mean vector of region C_k. The reconstructed vector $\hat{\mathbf{x}}^{(k)}$ is calculated as

$$\hat{\mathbf{x}}^{(k)} = \mathbf{W}^{(\mathrm{k})}\mathbf{y}^{(\mathrm{k})} + \bar{\mathbf{x}}^{(\mathrm{k})}, \text{ if } \mathbf{x} \in \mathrm{C_k}, \ \mathrm{k} = 1, \cdots, \mathrm{K} \quad (6)$$

The reconstruction error

$$\begin{aligned} E^{(k)} &= \|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|^2 \\ &= (\mathbf{x} - \bar{\mathbf{x}}^{(k)})^T P^{(k)^T} P^{(k)}(\mathbf{x} - \bar{\mathbf{x}}^{(k)}) \quad (7) \\ k &= 1, \cdots, K \end{aligned}$$

measures the distance between $\mathbf{x}$ and the subspace $\mathcal{L}^{(k)}$, where $P^{(k)} = 1 - \mathbf{W}^{(k)}\mathbf{W}^{(k)^T}$ is the projection matrix of $\mathcal{L}^{(k)}$. $E^{(k)}$ can be termed as reconstruction distance [9].

Input space can then be partitioned by a competition among these PCA type representations on the basis of the reconstruction distances $E_\mathbf{x} = \{\|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|^2, k = 1, \cdots, K\}$, K is the number of subspaces. Each time an input $\mathbf{x}$ is presented, we first make an ordering of the elements of $E_\mathbf{x}$ and then determine the adjustment of each subspace $\mathcal{L}^{(k)}$, $k = 1, \cdots, K$. In other words, we make a ranking $(E_\mathbf{x}^{(k_0)}, E_\mathbf{x}^{(k_1)}, \cdots, E_\mathbf{x}^{(k_{K-1})})$ of the reconstruction error set, with $\hat{\mathbf{x}}^{(k_0)}$ being closest to $\mathbf{x}$, $\hat{\mathbf{x}}^{(k_1)}$ being second closest to $\mathbf{x}$, $\hat{\mathbf{x}}^{(k_l)}$, $l = 0, \cdots, K-1$ being the reconstructed vector for which there are k_l vectors $\hat{\mathbf{x}}^{(j)}$ with $\|\mathbf{x} - \hat{\mathbf{x}}^{(j)}\| < \|\mathbf{x} - \hat{\mathbf{x}}^{(k_l)}\|$. Specifically, each subspace adjusts its projection via a dynamical learning rate which depends on the ranking of its reconstruction error. Denote the number d associated with each projection k by d_k, then (4) can be extended as:

$$\begin{aligned} W_{t+1}^{(k)} &= W_t^{(k)} + \mu_t h_\lambda(d_k)(\mathbf{y}^{(k)}(\mathbf{x} - \bar{\mathbf{x}}^{(k)})^T \\ &\quad -LT[\mathbf{y}^{(k)}\mathbf{y}^{(k)^T}]W_t^{(k)}) \quad (8) \\ &\quad k = 1, \cdots, K \end{aligned}$$

where $h_\lambda(d_k)$ is 1 for $d_k = 0$ and decays to zero for increasing d_k. In the simulations we choose the dynamical adaptation step $h_\lambda(d_k) = exp(-d_k/\lambda)$, with λ being a decay constant, which is same as in the original 'neural gas' algorithm [8].

3. Learning Adaptive Local Linear Transforms for Image Coding

In [7], an unsupervised learning algorithm called optimally integrated adaptive learning (OIAL) was proposed that combines both principal components extraction and competitive learning. The algorithm produces a number of linear transforms which are local to different regions. OIAL can be outlined as follows:

1. Initialize K transformation matrices $\{W^{(1)}, W^{(2)}, \cdots, W^{(K)}\}$.
2. For each training input vector $\mathbf{x}$:
   - classify the vector based on the subspace classifier

     $$\mathbf{x} \in C_i \quad \text{if } Q^{(i)}\mathbf{x} = \max_{j=1}^{K} Q^{(j)}\mathbf{x} \quad (9)$$

     where $Q^{(i)} = W^{(i)}W^{(i)^T}$.
   - update transformation matrix $W^{(i)}$ according to

     $$W^{(i)} = W^{(i)} + \alpha Z(\mathbf{x}, W^{(i)}) \quad (10)$$

     where $Z(\mathbf{x}, W^{(i)})$ is an appropriate adaptation algorithm for learning the M principal components of $\{\mathbf{x} | \mathbf{x} \in C_i\}$.
3. Repeat for each training vector until the transformations converge.

OIAL is a mixture model of local PCAs, which concurrently perform WTA competition and principal component projections. Intuitively, WTA competition has a number of disadvantages, for example, the underutilization problem. In [7], the initial set of transformation matrices are required to 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. To overcome this problem, [7] proposed to first make an estimation of the global principal component of the data distribution, then to each class a small amount of random variation is added as the initial transformation matrix. This not only requires extra computation, but also makes it hard to demonstrate the soundness.

Another problem of OIAL is that it does not consider the mean vector in each region, which is required to define a local PCA or KLT. Though a mixture model of local PCA may be introduced in different ways, the local linear transforms algorithm in [9] is preferred here, which can be outlined as follows:

1. Partition the input space R^L into K disjoint regions $\{C^{(1)}, \cdots, C^{(K)}\}$
2. Compute the local covariance matrices
$$\Sigma^{(k)} = E[(\mathbf{x}-E\mathbf{x})(\mathbf{x}-E\mathbf{x})^T | \mathbf{x} \in C^{(k)}];\ k = 1, \cdots, K \quad (11)$$
and their eigenvectors $\mathbf{e}_l^{(k)}, l = 1, \cdots, \mathrm{L}$. Relabel the eigenvectors so that the corresponding eigenvalues are in descending order $\lambda_1^{(k)} > \lambda_2^{(k)} > \cdots > \lambda_L^{(k)}$.
3. Choose a target dimension M and retain the M leading eigenvectors for the encoding.

Based on the above general scheme, [9] proposed to partition the input space by VQ and discuss two distortion measures for guiding the partition process. The optimal projection partition [9] is difficult to adaptively proceed as it needs to estimate the covariance matrix in each partition. Instead, we prefer the suboptimal one, called Euclidean partition [9], which builds a VQ on the basis of Euclidean distance. From our discussion of the mixture model in the last section, our learning algorithm for implementing local linear transforms can then be summarized as follows:

1. Applying algorithm (8) on the training data to partition the input space. After convergence, K reference vectors $\bar{\mathbf{x}}^{(k)}, k = 1, \cdots, K$ are obtained. Each $\bar{\mathbf{x}}^{(k)}$ can be considered as the centroid of the corresponding region $C^{(k)}$.

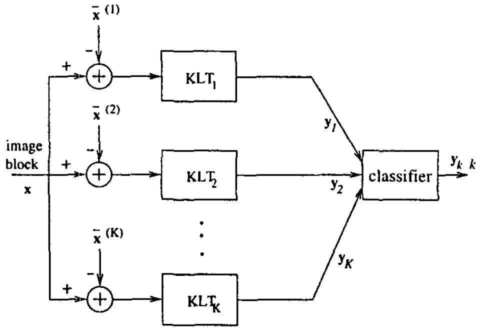


Figure 1: Modular architecture of adaptive transform coding. Input data are blocks of $L \times L$ pixels. The kth localized transformations KLT$_k$ consists of M basis images of size $L \times L$, and output is an M dimensional vector $\mathbf{y}_k$. The coefficient vector to be transmitted is selected by comparing the reconstruction distances.

2. Initialize K matrices $\{W^{(1)}, W^{(2)}, \cdots, W^{(K)}\}$ using random values.
3. For each training input vector $\mathbf{x}$:
   (a) Calculate the reconstruction distances $E_{\mathbf{x}} = \{\|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|^2, k = 1, \cdots, K\}$, where $\hat{\mathbf{x}}^{(k)} = \bar{\mathbf{x}}^{(k)} + W^{(k)} W^{(k)^T} (\mathbf{x} - \bar{\mathbf{x}}^{(k)})$.
   (b) Update each transform matrix $W^{(k)}$ according to the algorithm (8).
4. If converged, stop; Otherwise, go to Step 3.

After training is complete, our algorithm produces K transformation matrices $\{W^{(1)}, W^{(2)}, \cdots, W^{(K)}\}$ and each matrix converges to the localized KLT for the corresponding data class. K reference vectors $\bar{\mathbf{x}}^{(k)}$ approximately represent the centroids of the partitioned data regions. For encoding images, a modular architecture similar to the OIAL is shown in Fig.1. In the system, a number of K localized KLT modules consists of M basis images of dimension $L \times L$. The inner product of each basis image with input image block (corresponding reference vector subtracted) results in M coefficients per module, which is represented as the M-dimensional vector $\mathbf{y}_i$. To choose the class and the corresponding coefficient vector to be transmitted, we can simply compare the reconstruction distance $\|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|$ and select the module with minimal distance.

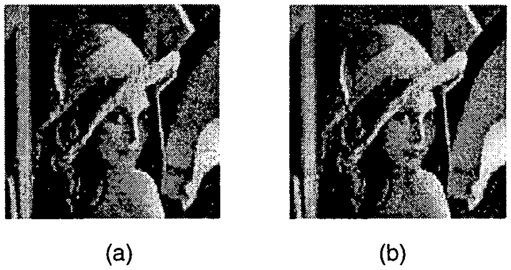
Figure 2: Original Lena image.

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

4. Simulations

We have performed a set of experiments of image coding based on the mixture models of local KLT. The first image is the typical Lena image with 256×256 pixels and $0 - 255$ gray levels, as shown in Fig.2. During training, the image was divided into blocks of 8×8 pixels for an input dimension of $L = 64$. The blocks were randomly sampled and presented to the network. Training proceeded with 50,000 random samples. The learning parameter μ in (8) is initially set to 0.5 and then dynamically decreases to 0.05. The decay constant λ in the dynamical adaptation step $h_\lambda(d_k)$ changes from 20 to 0.01. The time dependence for μ and λ is taken as the same form as $g(t) = g_i(g_f/g_i)^{t/t_{max}}$ [8], in which t is the current adaptation step, t_{max} is a predefined maximum adaptation step, i.e., $t_{max} = 50,000$ in our experiments. The subscripts i and f stands for initial value and final value, respectively, *i.e.*, $\mu_i = 0.5$, $\mu_f = 0.05$, $\lambda_i = 20$, $\lambda_f = 0.1$.

During decoding, the image was also divided into 8×8 nonoverlapping blocks, which were then transformed by the previously computed system into a set of coefficients, quantized and then transformed back into image blocks. We first experimented with the OIAL coding algorithm, which has 128 classes and four coefficients per class. Using an estimate of the global principal components of the data with a small amount of random variation added to each class as the initial set of transformation matrices, OIAL algorithm yields a decoded image as shown in Fig.3(a), with PSNR=29.6. If we randomly initialize the $W^{(i)}$'s by small random values, the decoded image will become poorer, as illustrated in Fig.3(b), with PSNR=27.



(a)

(b)

Figure 3: Lena image with OIAL coding for 128 classes and four coefficients per class. (a). Initialization of transform matrices by estimate of the global PCA, PSNR=29.6 (b). Randomly initialization of the transform matrices, PSNR=27.

In our experiment, we use the peak signal to noise ratio (PSNR) in dB given as

$$\text{PSNR} = 10\log_{10}[\frac{255^2}{\text{MSE}}] \tag{12}$$

to measure the decoded image quality, where 255 is the maximum intensity (for 8-bit intensity) and MSE is the mean square error between the original image and decoded image. In defining the average bits per pixel, we simply suppose that each local PCA's coefficient has been coded with 8 bits. Another $\frac{\log_2 K}{L}$ bits per pixels is needed to transmit the class index. Therefore, for the 8×8 image blocks and 128 classes, the average bits per pixel is $\frac{2\times8}{8\times8} + \frac{\log_2 128}{8\times8} = 0.36$ bits/pixel for 2 coefficients per class, and $\frac{4\times8}{8\times8} + \frac{\log_2 128}{8\times8} = 0.61$ bits/pixel for 4 coefficients per class, and so on. The optimal coding of class information is not considered here, which can be found in [5].

The image provided in Figs.4(a) and (b) show the decoding quality of our learning algorithm for the cases of two coefficients per class and four coefficients per class, respectively. The initial transform matrices $W^{(i)}$'s are all randomly initialized with small values for each elements and the decoded image quality is clealy superior to the image in Fig.3(b) and slightly better than the image in Fig.3(a). From Fig.4 we can also find that increasing the number of coefficients per class yields a higher peak signal to noise ratio. However, as is pointed out in [7], there is a limit to the improvement realized through increasing the number of coefficients alone because of the resulting increase in quantization error. The results of aplying our al-

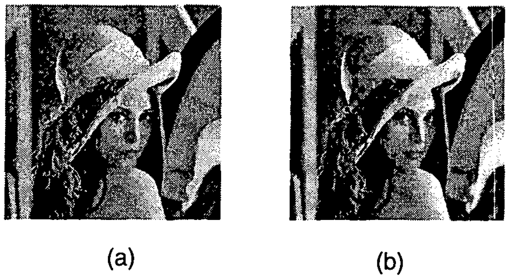


Figure 4: Lena image coded with our learning algorithm for 128 classes, with (a). two coefficients per class, PSNR=30.7; and (b). four coefficients per class, PSNR=32.

Figure 5: Original F16 jet fighter image.

gorithm to encode an F-16 jet fighter image Fig.6 are demonstrated in Fig.7 (a) and (b), corresponding to the cases of two coefficients per class and four coefficients per class, respectively. This result show no significant difference between the two coefficients and four coefficients, from both their PSNR's and subjective quality.

5. Conclusion

In this paper, we proposed an approach to adaptive compression on the basis of a mixture of local PCA model. The learning algorithm improves the performance of the optimally integrated adaptive learning (OIAL) system. The architecture of the encoding scheme is similar to that of OIAL, which consists of a number of modules corresponding to localized PCAs. Each module in the system specializes in a class of data and perform a linear transformation on its class data using the basis images. The training involves concurrently making ranks of the "reconstruction distances" calculated from the localized PCAs and dynamically performing PCA learning for different modules. The simulation results have shown that our algorithm outperforms the OIAL.

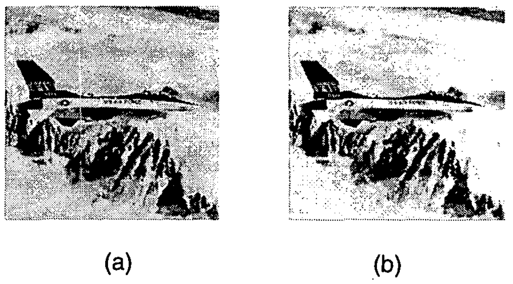


Figure 6: F16 image coded with our learning algorithm for 128 classes, with (a). two coefficients per class, PSNR=29; and (b).four coefficients per class, PSNR=30.3.

References

[1] E.Oja, "Simplified neuron model as a principal component analyzer," *J. Math. Biol.,* vol. 15, pp. 267-273, 1982.

[2] E.Oja, "Neural networks, principal components and subspace," *Int. J. Neural Syst.,* vol.1, pp. 61-68, 1989.

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

[4] L.Torres-Urgell, R.L.Kirlin, "Adaptive image compression using Karhunen-Loeve transform," *Signal Processing,* vol.21, pp.303-313, 1990.

[5] H.M.Abbas, M.M.Fahmy, "Neural model for Karhunen-Loeve transform with application to adaptive image compression," *IEE Proceedings-I,* vol.140, pp.135-143, 1993.

[6] S.Haykin, "Neural networks: A comprehensive foundation," New York: Macmillan, 1994.

[7] R.D.Dony, and S.Haykin, "Optimally adaptive transform coding," *IEEE Transactions on Image Processing,* vol.4, pp.1358-1370, 1995.

[8] T.M.Martinetz, S.G.Berkovich and K.J.Schulten 1993, "'Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks,* 4, 558-568.

[9] N.Kambhatla, and T.K.Lee, "Dimension reduction by local principal component analysis," *Neural Computation,* vol. 9, pp. 1493-1516, 1997.