# Low Complexity Gaussian Mixture Model-based Block Quantisation of Images

Stephen So and Kuldip K. Paliwal

***Abstract*— In this paper, we present a low complexity version of Gaussian mixture model-based block quantisation for images. The latter coding method has been recently shown to offer considerable performance improvements over the traditional block quantisers where the probability density function (PDF) is assumed to be Gaussian. However, the coder relies on the frequent use of the Karhunen Loève transform (KLT). A coder with less computational complexity can be realised by replacing the KLT with a discrete cosine transform (DCT) which has the advantages of a static transformation matrix, efficient implementation, and good decorrelation characteristics. The results are promising with only a minor drop in performance compared with the KLT-based coder and due to its low complexity, more clusters can be used to make up the performance loss.**

## I. Introduction

BLOCK quantisation or transform coding has been used as a less complex alternative to vector quantisation in the coding of images [11]. As shown in Fig. 1, a block quantiser consists of a set of scalar quantisers operating independently on blocks or vectors which have been operated on by a linear transformation [3]. The role of the linear transformation is to remove as much correlation as possible from the vector components since this improves the efficiency of the scalar quantisers. Assuming the source is jointly Gaussian, the linear transformation which minimises the mean square error (MSE) distortion after quantisation is the Karhunen Loeve̖ Transform (KLT)[1].

The performance of a scalar quantiser depends on whether it matches the probability density function (PDF) of the source. The general assumption is that the source is jointly Gaussian, hence Lloyd-Max non-uniform scalar quantisers are used in traditional block quantisers. This is a reasonable assumption as the transform coefficients are linear combinations of the random variables. As the dimension increases, the distribution tends to approach a Gaussian, irrespective of the shape of the source PDF [11]. However, better performance can be achieved by estimating the PDF of the source and designing quantisers that are adapted to this PDF. Subramaniam and Rao in [9] proposed a block quantiser for speech where the PDF was modelled by a Gaussian mixture model (GMM). The method was subsequently adapted for image coding by Paliwal and So in [7] where an improvement over the benchmark Gaussian block quantiser was reported. It was also shown that increasing the number of clusters in the GMM further reduced the distortion while the bitrate remained unchanged. However, there is heavy use of the KLT and inverse KLT which are computationally expensive. The complexity due to the transformation is proportional to the number of Gaussian mixtures used.

In this paper, we present a modified, low complexity GMM-based block quantiser (GMM-DCT) for images that utilises the discrete cosine transform (DCT) for decorrelating the vectors. This modification leads to a coder that is computationally simpler than the optimal KLT-based coder while maintaining comparable performance. The proposed block quantisation scheme consists of three stages: *PDF estimation*, *bit allocation*, and the *low complexity GMM-DCT block quantiser*. Figs. 2 and 3 show block diagrams of the PDF estimation and bit allocation procedure and the modified GMM block quantiser respectively. Like the JPEG standard [10], images are operated on as blocks of pixels with dimension $8 \times 8 = 64$ since most of the correlation generally is contained within the vicinity of 8 pixels, assuming an isotropic source model [1].

The authors are with the Signal Processing Laboratory, School of Microelectronic Engineering, Faculty of Engineering and Information Technology, Griffith University, Brisbane, QLD 4111 Australia (E-mail: s.so@griffith.edu.au; k.paliwal@griffith.edu.au)

[1] The KLT is also known as the eigenvector transform, Hotelling transform, or Principal Component Analysis in pattern recognition

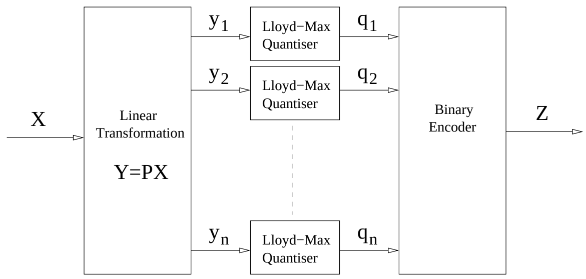


Fig. 1. Traditional Gaussian Block Quantiser

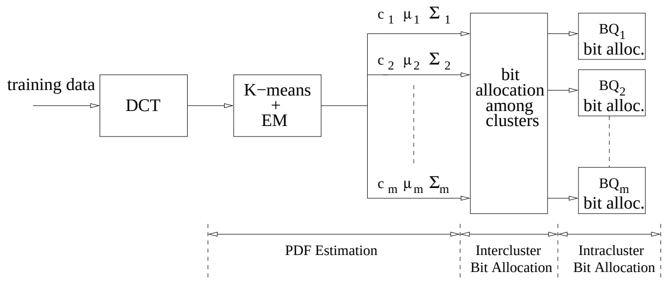


Fig. 2. PDF Estimation and Bit Allocation from Training Images

## II. 2D Discrete Cosine Transform

The DCT transforms vectors, $\boldsymbol{x}$, by multiplying them with a static transformation matrix, $\boldsymbol{D}$, which consists of cosine basis functions as its rows. For image applications (2D), the resulting coefficients are multiplied with the transpose of $\boldsymbol{D}$ in order to transform along the second dimension.

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}\boldsymbol{D}^T \tag{1}$$

$$\boldsymbol{D}_{i,j} = \frac{c_i}{2}\cos\left(\frac{2j+1}{16}i\pi\right) \tag{2}$$

$$c_i = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } i = 0 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

where $\boldsymbol{X}$ are the 2D source vectors, $\boldsymbol{Y}$ are the transformed vectors and $\boldsymbol{D}_{i,j}$ is the $(i,j)$th element of the transformation matrix. The transformation matrix of the DCT contains no terms related to a source statistic such as covariance. Therefore the transformation matrix is static and applied to all vectors.

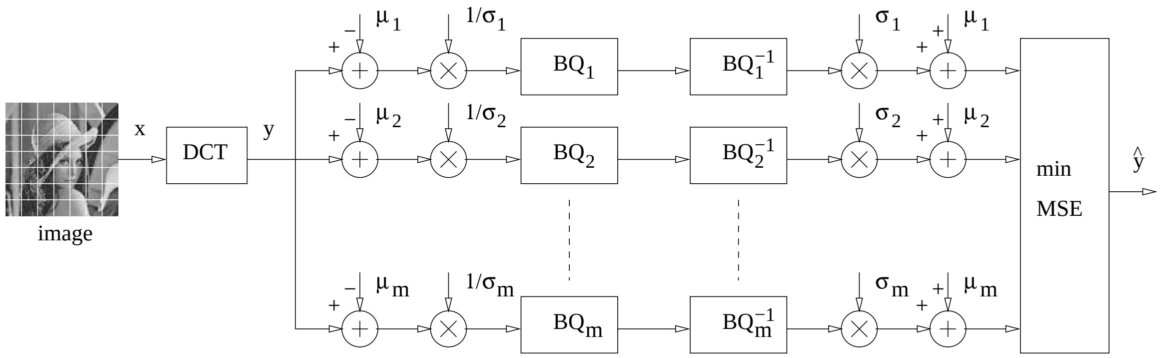


Fig. 3. Low Complexity GMM-DCT Block Quantiser (BQ – block quantiser)

## III. PDF Estimation using Gaussian Mixture Models

GMMs can be used for modelling any arbitrary distribution using multivariate Gaussians as basis functions. The PDF model, $P$, to be estimated is represented by a mixture of multivariate Gaussians $\mathcal{N}(\boldsymbol{y};\boldsymbol{\mu};\boldsymbol{\Sigma})$

$$P(\boldsymbol{Y}|\boldsymbol{\lambda}) = \sum_{i=1}^{m} c_i \mathcal{N}_i(\boldsymbol{y};\boldsymbol{\mu}_i;\boldsymbol{\Sigma}_i) \tag{4}$$

$$\boldsymbol{\lambda} = [m, c_i, \ldots, c_m, \mu_1, \ldots, \mu_m, \Sigma_1, \ldots, \Sigma_m] \tag{5}$$

$$\mathcal{N}(\boldsymbol{y};\boldsymbol{\mu};\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})} \tag{6}$$

where $\boldsymbol{Y}$ are the vectors of transform coefficients, $m$ is the number of mixtures, and $n$ is the dimension of the vectors. $\boldsymbol{\lambda}$ is the set of model parameters consisting of $c_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ which are the weight, mean, and covariance matrix of the $i$th mixture respectively. Note the words 'mixture' and 'cluster' will be used interchangeably in this paper.

The parametric model, represented by parameters, $\boldsymbol{\lambda}$, is initialised by applying the K-means algorithm on the training vectors where $m$ clusters are produced, each represented by a mean, $\boldsymbol{\mu}$, a covariance matrix, $\boldsymbol{\Sigma}$, and cluster weight, $c$. These form the initial parameters for the GMM estimation procedure. Using the Expectation Maximisation (EM) algorithm, the maximum-likelihood estimate of the parametric model is computed iteratively until there is convergence of the log likelihood where a final set of means, covariance matrices, and weights are produced. Since the GMM parameters for the estimated PDF are properties of the source and thus independent of the bit rate, this procedure only needs to be done once and stored for later use [9].

## IV. Bit Allocation

There are two types of bit allocation that are required: *intercluster bit allocation* and *intracluster bit allocation*. Since the bit allocation is not a computationally expensive procedure, it can be done 'on-the-fly' depending on the chosen bit rate.

### A. Intercluster Bit Allocation

With intercluster bit allocation, quantiser levels need to be assigned to each of the $m$ clusters depending upon to the covariance and probability of that cluster. If the GMM is viewed a composite Gaussian source where each vector is generated by one of the $m$ clusters, then the cluster weights calculated from the EM algorithm also represent the cluster probabilities [9]. For a fixed-rate quantiser, the total number of quantiser levels is fixed [9]:

$$2^{b_{tot}} = \sum_{i=0}^{m-1} 2^{b_i} \tag{7}$$

where $b_{tot}$ is the total number of bits in the bit budget, $b_i$ is the number of bits assigned to cluster $i$, and $m$ is the number of clusters. The average distortion is approximated by:

$$D_{tot} = \sum_{i=0}^{m-1} c_i D_i(b_i) \tag{8}$$

The high resolution approximation[2] for the distortion of a single Lloyd-Max scalar quantiser from an $n$-component block quantiser operating on Gaussian sources is given by:

$$D_i(b_i) = nK\Lambda_i 2^{-2\frac{b_i}{n}} \tag{9}$$

$$\Lambda_i = \left(\prod_{j=0}^{n-1} \lambda_{i,j}\right)^{\frac{1}{n}} \quad \text{for } i = 0, 1, \ldots, m-1 \tag{10}$$

where $n$ is the dimension of the vectors, $m$ is the number of clusters, $\lambda_{i,j}$ is the $j$th variance of cluster $i$, and $K$ is a constant which is approximately equal to $\frac{\pi\sqrt{3}}{2}$ for Gaussian sources.

Using Lagrange multipliers, the average distortion can be minimised under the fixed rate constraint of (7), and the following bit allocation formula is derived:

$$2^{b_i} = 2^{b_{tot}} \frac{(c_i\Lambda_i)^{\frac{n}{n+2}}}{\sum_{i=0}^{m-1}(c_i\Lambda_i)^{\frac{n}{n+2}}}, \quad \text{for } i = 0, 1, \ldots, m-1 \tag{11}$$

where $c_i$ is the weight of cluster $i$.

### B. Intracluster Bit Allocation

After the bits are allocated to each cluster, further bit allocation is performed to assign bits to each of the $n$ components. Following the derivation presented in [3], the total number of bits is fixed:

$$b_i = \sum_{j=0}^{n-1} b_{i,j}, \quad \text{for } i = 0, 1, \ldots, m-1 \tag{12}$$

where $b_{i,j}$ is the number of bits assigned to component $j$ of cluster $i$. Again, using the high resolution approximation for the distortion of a Lloyd-Max scalar quantiser, the average distortion of cluster $i$ is given by:

$$D_i = \frac{1}{n}\sum_{j=0}^{n-1} K\lambda_{i,j} 2^{-2b_{i,j}} \tag{13}$$

[2] One of the two asymptotic results in quantisation theory, *high resolution approximations* assumes very high bitrates and very finely spaced quantiser levels [2].

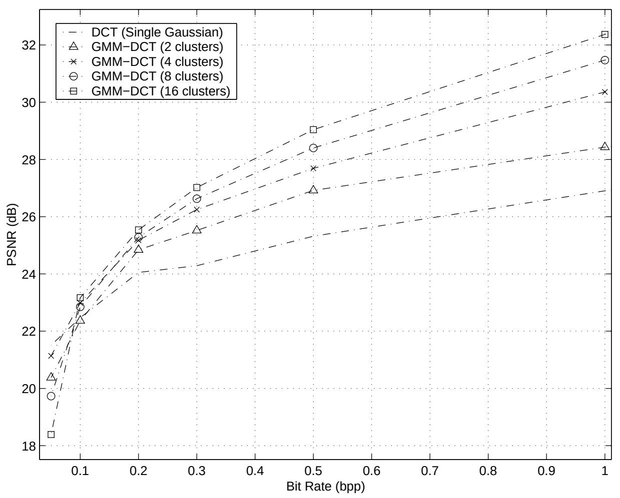


Fig. 4. Distortion Rate Performance of GMM-DCT coder for boat512.raw (512 x 512)

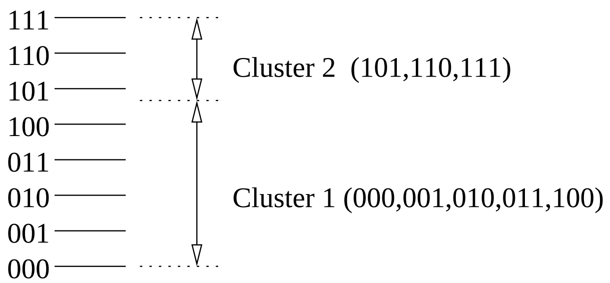


Fig. 5. Example of Quantiser Level Encoding and Cluster Number Encoding

$$\text{for } i = 0, 1, \ldots, m-1$$

Using Lagrange multipliers, the average distortion is minimised under the fixed rate constraint of (12) and the following bit allocation formula is derived:

$$b_{i,j} = \frac{b_i}{n} + \frac{1}{2}\log_2 \frac{\lambda_{i,j}}{\left(\prod_{j=0}^{n-1} \lambda_{i,j}\right)^{\frac{1}{n}}} \tag{14}$$

$$\text{for } i = 0, 1, \ldots, m-1$$

## V. Low Complexity GMM-DCT Block Quantiser

The modified GMM block quantiser is shown in Fig. 3. To quantise a DCT coefficient vector, $\boldsymbol{y}$, using a particular cluster $i$, the cluster mean, $\boldsymbol{\mu}_i$, is subtracted and then divided by the standard deviation, $\boldsymbol{\sigma_i}$, and quantised using a set of $n$ Gaussian Lloyd-Max scalar quantisers (represented as BQ in Fig. 3) as described in [3] with their respective bit allocations. The indicies from the quantiser are decoded, multiplied by the standard deviation and the cluster mean added back to give the approximated transform vector, $\widehat{\boldsymbol{y}}_{\boldsymbol{i}}$. The distortion between this quantised DCT vector and original DCT vector is then calculated, $d(\boldsymbol{y} - \widehat{\boldsymbol{y}}_i)$. For image applications, the simplest distortion measure is mean squared error (MSE). The above procedure is performed for all clusters in the system. The cluster $i$ which produces the vector, $\widehat{\boldsymbol{y}}_i$, that is the least distorted is chosen as the best quantiser and that vector is transmitted as $\widehat{\boldsymbol{y}}$.

In order for the decoder to determine which cluster was used to quantise a particular vector, the cluster number in the form of side information must be present. Rather than explicity sending the cluster number to the decoder, an elegant way of encoding this side information implicitly as well as the transmission of quantiser levels is given in [8]. The entire quantiser level range is partitioned to each of the clusters and vectors that belong to one cluster can only take on codewords within that partition. Fig. 5 shows a simple example of a 2 cluster system. The bit budget is 3 bits which corresponds to 8 quantiser levels. Vectors which were coded using cluster 1 take on the binary codewords $000, 001, 010, 011, 100$ while those coded with cluster 2 can take on the codewords, $101, 110, 111$. The decoder decides which cluster to use for a particular vector by determining which partition the codeword falls into.

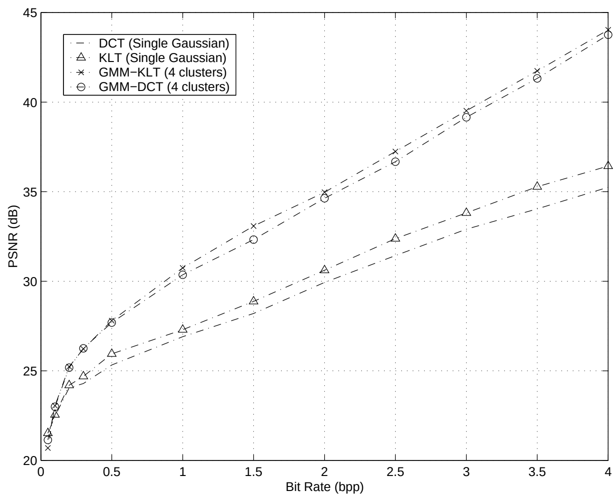


Fig. 6. Comparing D-R Performance of GMM-DCT with GMM-KLT coders for boat512.raw (512 x 512)

## VI. Advantages of using the DCT over the KLT

The discrete cosine transform (DCT) is a good suboptimal alternative to the KLT since it possesses similar decorrelation and energy compaction properties for first-order Markov processes with exponential correlation [6]. Thus replacing the KLT with a DCT as a decorrelator should only incur a slight degradation on the performance.

The rows of the transformation matrix used in the KLT consists of the eigenvectors which are dependent on the source statistics. Therefore in an $m$-cluster GMM-KLT coder, $m$-KLTs and $m$-inverse KLTs need to be performed during the coding process. On the other hand, the transformation matrix of the DCT consists of a set of cosine basis functions which are source-independent. This removes the need for separate transformation operations for each cluster. Also, since the transform space is common for every cluster, Euclidean distances will be the same, hence MSE distortion can be calculated in the DCT domain, removing the need for inverse DCTs. In other words, only one DCT operation needs to be performed in the entire coder, thereby reducing the transformation complexity of the coder by $2m$ times. Since the transformation operation is independent of the number of clusters, then more clusters can be used to make up for the loss in performance, which would have otherwise been computationally exorbitant if the source dependent KLT were used.

## VII. Experimental Results

The training set consists of 14 images of various sizes (128, 256, 512). This totals to 28928 vectors of dimension $8 \times 8$. Lloyd-Max scalar quantisers of up to 256 levels (8 bits) were used. The number of clusters was varied (2, 4, 8, 16) and 20 iterations of the EM algorithm were performed. Fig. 4 shows the distortion-rate characteristic of the GMM-DCT block coder on the image 'boat', which is an 8-bit greyscale image of size $512 \times 512$. Additionally, this image was not part of the training set. The results for a single Gaussian block coder are also presented as a benchmark. This benchmark coder is similar to the one proposed in [3] except that a DCT is applied, rather than a KLT.

Fig. 4 shows that the PSNR of the GMM-DCT coder is at least 2 to 5 dB better at medium bit rates (around 1 bpp) and 1 to 3 dB better at low bit rates (around 0.3 bpp) than the DCT-based transform coder which makes the Gaussian PDF assumption. As the number of clusters is increased, considerable performance gains are realised since with more mixtures, the true PDF of the training data set is estimated better than with a lesser number of mixtures. However, the gains achieved by more accurate modelling of the PDF are compromised by the amount of intrinsic side information that increases as more clusters are used. This is considerably noticeable at very low bit rates (less than 0.1 bpp) where the GMM-DCT coder is performing worse than the basic DCT block coder due to the increased influence of the effective bits required to identify clusters.

A further comparison is shown in Fig. 6 where the GMM-DCT block coder is compared with the GMM-KLT block coder of [7] for 4 clusters as well as the standard Gaussian DCT and KLT coders. It can be seen that as expected, the DCT coders always perform slightly below that of the KLT ($\leq$ 0.76 dB). However, this slight performance degradation is justified by the considerable computational reduction resulting from the use of the DCT.

Figs. 7 and 8 show 'zoomed-in' parts of the original images, 'boat' and 'goldhill' which are both 8 bit greyscale images of size $512 \times 512$ as well as the compressed versions. It can be seen that the traditional block quantiser which assumes the source PDF to be a single Gaussian produces a lot of 'grainy' distortion. When the source PDF is modelled as a 2 cluster GMM, the 'graininess' is reduced, especially in the smooth regions while most of the 'blockiness' appears around the edges. Increasing the GMM to 8 clusters allows it to more accurately model the source PDF. As expected, this further reduced the distortion.

Finally, the computational times of the low complexity GMM-DCT block quantiser and optimal GMM-KLT block quantiser were compared on an Intel Pentium 4 system clocked at 2.4 GHz. At 0.5 bpp and 8 clusters, the GMM-KLT block quantiser took on average 30.5 seconds to quantise 49152 vectors of dimension 64 while the GMM-DCT block quantiser took on average 7.5 seconds.

## VIII. Conclusion

The GMM-DCT block image coder presented in this paper has shown that the modelling of DCT coefficients using GMMs leads to performance gains of 2–5 dB at medium bit rates and 1–3 dB at low bit rates over DCT coders which assume the PDF to be a Gaussian. This demonstrates the same advantages of designing block quantisers which adapt to the true PDF of the source as was seen in the previous KLT-based approaches. Utilising the source independent nature of the DCT, the resulting coder is computationally simpler than the optimum GMM-KLT coder with a slight performance degradation of at most 0.76 dB.

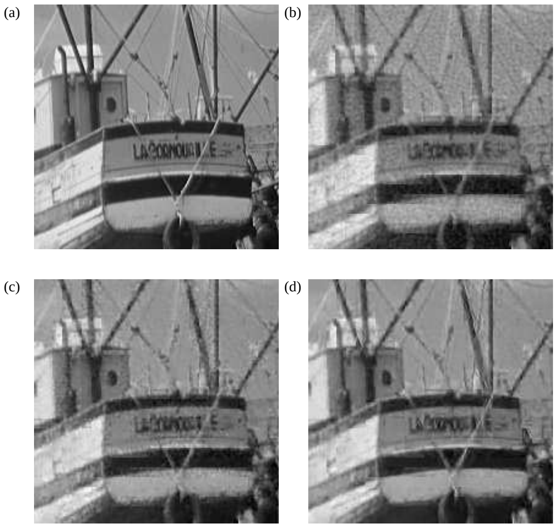


Fig. 7. Part of the image 'boat' ($512 \times 512$) (a) Original Image (b) Compressed at 0.5 bpp using traditional block quantiser (PSNR 25.32 dB) (c) Compressed at 0.5 bpp using 2 clusters (PSNR 26.92 dB) (d) Compressed at 0.5 bpp using 8 clusters (PSNR 28.4 dB)

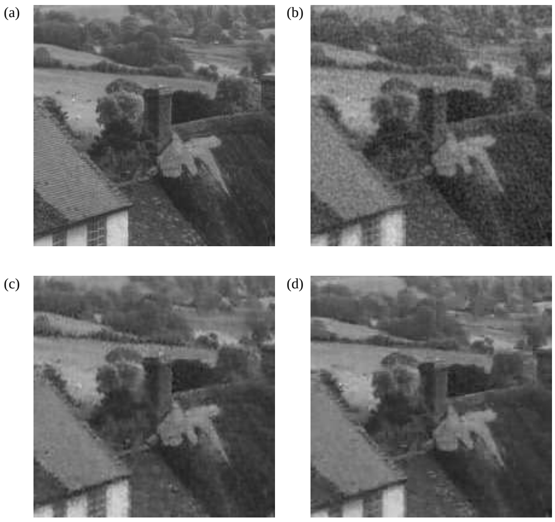


Fig. 8. Part of the image 'goldhill' ($512 \times 512$) (a) Original Image (b) Compressed at 0.5 bpp using traditional block quantiser (PSNR 28.16 dB) (c) Compressed at 0.5 bpp using 2 clusters (PSNR 29.48 dB) (d) Compressed at 0.5 bpp using 8 clusters (PSNR 30.15 dB)

## References

[1] V. Bhaskaran & K. Konstantinides, *Image and Video Compression Standards*, 2nd Ed., Kluwer International Series, Boston, 1997.

[2] R.M. Gray & D.L. Neuhoff, "Quantization", *IEEE Trans. Inform. Theory*, Vol. 44, No. 6, Oct. 1998.

[3] J.J.Y. Huang & P.M. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables", *IEEE Trans. Commun. Syst.*, Vol. CS-11, pp. 289–296, Sept. 1963.

[4] S.P. Lloyd, "Least Square Quantization in PCM", *IEEE Trans. Inform. Theory*, Vol. IT-28, No. 2, pp. 129–137, Mar. 1982.

[5] J. Max, "Quantising for Minimum Distortion", *IRE Trans. Inform. Theory*, Vol. IT-6, pp. 7–12, Mar. 1960.

[6] A.N. Netravali & J.O. Limb, "Picture Coding: A Review", *Proc. IEEE*, Vol. 68, No. 3, pp. 366–406, Mar. 1980.

[7] K.K. Paliwal & S. So, "Efficient Block Coding of Images Using Gaussian Mixture Models", *Proceedings of the Workshop of Signal Processing and Applications 2002*, Brisbane, Dec. 2002.

[8] K.K. Paliwal & S. So, "Fractional Bit Encoding for Gaussian Mixture Model-Based Block Quantisation", *To be published*, Dec. 2002.

[9] A.D. Subramaniam & B.D. Rao, "PDF Optimized Parametric Vector Quantization of Speech Line Spectral Frequencies", *IEEE Trans. Speech Audio Processing*, Vol. 11, No. 2, pp. 130–142, Mar. 2003.

[10] G.K. Wallace, "The JPEG Still Picture Compression Standard", *Communications of the ACM*, Vol. 34, No. 4, pp. 30-44, Apr. 1991.

[11] P.A. Wintz, "Transform Picture Coding", *Proc. IEEE*, Vol. 60(7), pp. 809–820, July 1972.