Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Signal Processing: *Image Communication* 20 (2005) 435–446

SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Low-complexity GMM-based block quantisation of images using the discrete cosine transform

Kuldip K. Paliwal, Stephen So*

School of Microelectronic Engineering, Griffith University, Brisbane QLD 4111, Australia

Received 14 November 2003; accepted 3 March 2005

Abstract

While block transform image coding has not been very popular lately in the presence of current state-of-the-art wavelet-based coders, the Gaussian mixture model (GMM)-based block quantiser, without the use of entropy coding, is still very competitive in the class of fixed rate transform coders. In this paper, a GMM-based block quantiser of low computational complexity is presented which is based on the discrete cosine transform (DCT). It is observed that the assumption of Gaussian mixture components in a GMM having Gauss–Markov properties is a reasonable one with the DCT approaching the optimality of the Karhunen–Loève transform (KLT) as a decorrelator. Performance gains of 6–7 dB are reported over the traditional single Gaussian block quantiser at 1 bit per pixel. The DCT possesses two advantages over the KLT: being fixed and source independent, which means it only needs to be applied once; and the availability of fast and efficient implementations. These advantages, together with bitrate scalability, result in a block quantiser that is considerably faster and less complex while the novelty of using a GMM to model the source probability density function is still preserved.

Keywords: Transform coding; Block quantisation; Gaussian mixture models; Discrete cosine transform

1. Introduction

Block quantisation or transform coding, has been used as a less complex alternative to full-search vector quantisation in the coding of images [1]. Proposed by Kramer and Mathews [10] and analysed by Huang and Schultheiss [9], it involves decorrelating the components within a block or vector of samples using a linear transformation before scalar quantising each component independently. When quantising for minimum distortion, the Karhunen–Loève transform (KLT)[1] is the best

*Corresponding author. Tel.: +61 7 3875 3754; fax: +61 7 3875 5198.
E-mail addresses: k.paliwal@griffith.edu.au (K.K. Paliwal), s.so@griffith.edu.au (S. So).

[1]Often called an eigenvector transform, Hotelling transform, or principal component analysis (PCA) in the pattern recognition literature.

0923-5965/$ - see front matter
doi:10.1016/j.image.2005.03.001

transform for correlated Gaussian sources [6]. It should be noted that, while block quantisation and transform coding are generally synonymous, the latter term encompasses coders that use variable rate entropy coding after the quantisation step, like the JPEG standard [18], while block quantisers use non-uniform scalar quantisers of fixed rate with no entropy coding [9,19]. This paper is focused on block quantisation only.

The probability density functions (PDF) of real life sources are rarely Gaussian. Since the scalar quantisers in traditional block quantisers are designed to quantise Gaussian sources efficiently, any PDF mismatch will invariably cause a degradation in performance. Rather than assuming the source PDF to be a standard function such as Gaussian, Laplacian, etc., one can design a quantiser that matches the source PDF as close as possible. The K-means algorithm, otherwise known in literature as the generalised Lloyd algorithm (GLA) and Linde–Buzo–Gray (LBG) algorithm[2], allows one to design vector quantisers which match the PDF of training data via the Lloyd conditions [5,11].

There have been numerous studies in the coding literature on source PDF modelling for quantiser design. These can be broadly classified as either *non-parametric* or *parametric* modelling. Ortega and Vetterli [13] estimated the source model in a non-parametric fashion using piecewise linear approximation. Similarly, multidimensional histograms were used by Gardner and Rao [4] to model the PDFs of line spectral frequencies (LSF) in order to evaluate the theoretical bounds of split vector quantisers. In relation to parametric modelling, Su and Mersereau [16] applied Gaussian mixture models (GMM) in the estimation of the PDF of DC transform coefficients while Archer and Leen [1] used GMMs to form a probabilistic latent variable model from which transform coding design algorithms were derived. On the speech side, Hedelin and Skoglund [8] used GMMs for designing and evaluating high-rate vector quantisers of LSFs while Subramaniam and Rao [17] applied GMMs to improve block quantisation.

In our previous work [15], where we applied the GMM-based block quantiser of [17] to the coding of images, we showed that this quantiser was able to achieve peak signal-to-noise ratio (PSNR) gains of 5–10 dB at high bitrates and 2–4 dB at medium to low bitrates over the traditional block quantiser. Even though the GMM-based block quantiser is not an optimal solution, since it uses the Expectation–Maximisation (EM) algorithm [3] which maximises the likelihood rather than minimising quantiser distortion [1,8], it nevertheless possesses a number of advantages over vector quantisers [17]:

- compact representation of the source PDF which is independent of bitrate;
- bitrate scalability with 'on-the-fly' bit allocation; and
- low search complexity and memory requirements which are independent of the rate of the system.

Because block quantisers are designed for each Gaussian mixture component or cluster, and these clusters overlap each other, a soft-decision scheme is required, where each block is evaluated with the block quantiser of every cluster and the one which incurs the least distortion is chosen. This means that for an m-cluster GMM-based block quantiser, each block needs to be transformed $2m$ times by the KLT. Such a scheme involves a large amount of computation that is proportional to m, which we have shown previously [15] to be an important design parameter in determining the performance of the system. That is, given that there are enough bits, increasing the number of clusters, m, will generally lead to lower distortion for a given bitrate.

In this paper, we present a low-complexity GMM-based block quantiser for image coding that does not require the use of multiple transform operations. We will show that this quantiser scheme is not only computationally simpler, but also exhibits similar performance gains over the

[2]The only notable difference between the K-means algorithm and GLA/LBG algorithm is the initialisation used. Since the refinement steps, which are based on the Lloyd conditions for optimality, are essentially the same, we will use the terms 'K-means algorithm', 'GLA', and 'LBG algorithm', interchangeably in this paper.

traditional block quantiser (as we have shown in [15] for the KLT-based scheme) as well as minimal degradation in PSNR when compared to the KLT-based scheme.

This paper is organised as follows: Section 2 briefly describes GMM-based block quantisation as presented in [17] where the KLT is used as the decorrelator. Section 3 shows our modification to this quantiser, which we will refer to as the GMM-DCT-based block quantiser. Results showing the efficiency and performance of the GMM-DCT-based block quantiser are presented and discussed in Section 4. We conclude in the last section on the efficiency of the proposed modified scheme.

2. GMM-based block quantisation

The GMM-based block quantiser can be broken down into three stages: PDF estimation, bit allocation and minimum distortion block quantisation. Figs. 1 and 2 show the PDF estimation and bit allocation procedure and minimum distortion block quantisation, respectively. These are similar to those that appear in [17].

2.1. PDF estimation using Gaussian mixture models

The PDF model, as a mixture of multivariate Gaussians, $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$, can be given by

$$G(\boldsymbol{x}|\boldsymbol{M}) = \sum_{i=1}^{m} c_i \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i), \tag{1}$$

$$\boldsymbol{M} = [m, c_i, \ldots, c_m, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m], \tag{2}$$

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \mathrm{e}^{-1/2(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)}, \tag{3}$$

where $\boldsymbol{x}$ is a source vector, m is the number of mixture components (or, clusters), and n is the dimensionality of the vector space. c_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the weight, mean, and covariance matrix of the ith mixture, respectively. Note the words ‘mixture component’ and ‘cluster’ will be used interchangeably in this paper.

The parametric model, $\boldsymbol{M}$, is initialised by applying the K-means algorithm [11] on the training vectors representing the source distribution and m clusters are produced, each represented by a mean or centroid, $\boldsymbol{\mu}$, a covariance matrix, $\boldsymbol{\Sigma}$, and a cluster weight, c. These form the initial

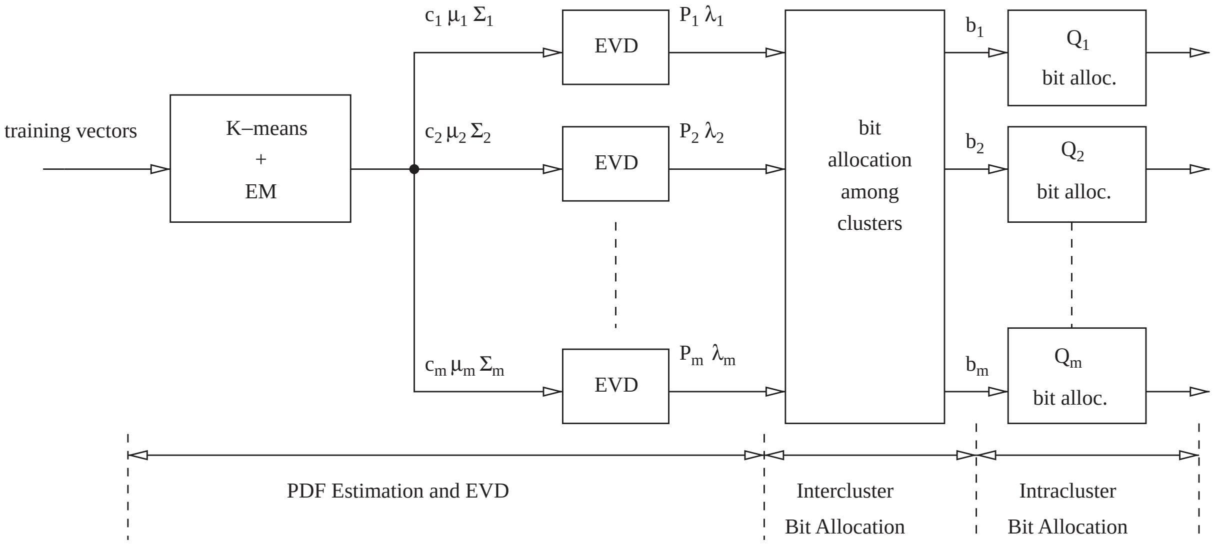

Fig. 1. PDF estimation and bit allocation from training data.

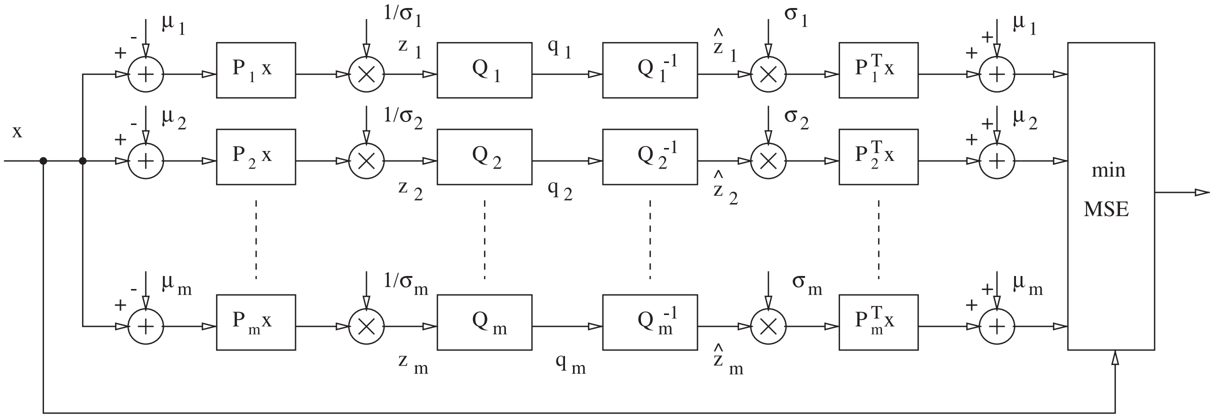

Fig. 2. Minimum distortion block quantisation (Q—block quantiser).

parameters for the GMM estimation procedure. Using the expectation–maximisation (EM) algorithm [3], the maximum-likelihood estimate of the parametric model is computed iteratively and 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 bitrate, this procedure only needs to be done once and stored at the encoder and decoder [17].

2.1.1. The Karhunen–Loève transform and eigenvalue decomposition

The Karhunen–Loève transform (KLT) linearly transforms a vector, $\boldsymbol{x}$, of dimension, n, by multiplying with a square orthogonal matrix, $\boldsymbol{P}$ [9]

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}, \tag{4}$$

where $\boldsymbol{y}$ is the vector of transformed coefficients. Similarly the inverse transform is given by [9]

$$\boldsymbol{x} = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{y}. \tag{5}$$

Eigenvalue decomposition (EVD) is applied to the covariance matrix of each cluster to find the n eigenvalues, $\{\lambda_{i,j}\}_{j=1}^{n}$, where $\lambda_{i,j}$ represents the eigenvalue or variance of the jth decorrelated component of cluster i, as well as the n eigenvectors, $\{\boldsymbol{v}_{i,j}\}_{j=1}^{n}$, which form the rows of the orthogonal matrix, $\boldsymbol{P}_i$, for that cluster. The eigenvalues will be used in the bit allocation procedure of the block quantiser design.

Since the orthogonal matrices and eigenvalues are properties of the source model and thus independent of the bitrate, this procedure only needs to be done once and the parameters are stored at the encoder and decoder [17].

2.2. 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 bitrate by the encoder and decoder. Hence scalability is a feature of the GMM-based block quantiser [17].

2.2.1. Intercluster bit allocation

With intercluster bit allocation, the range of quantiser levels is partitioned to each of the m clusters. For a fixed-rate quantiser, the total number of quantiser levels is fixed:

$$2^{b_{\mathrm{tot}}} = 2^{nb}, \tag{6}$$

where n is the number of components in a block and b is the target bitrate (in bits per sample). These bits are allocated to each cluster, based on the following formula [17]:

$$2^{b_i} = 2^{b_{\mathrm{tot}}} \frac{(c_i A_i)^{n/(n+2)}}{\sum_{i=1}^{m} (c_i A_i)^{n/(n+2)}}, \tag{7}$$

$$A_i = \left[\prod_{j=1}^{n} \lambda_{i,j} \right]^{1/n} \quad \text{for } i = 1, 2, \ldots, m, \tag{8}$$

where b_i and c_i are the number of allocated bits and weight of cluster i, respectively. $\lambda_{i,j}$ is the jth eigenvalue of cluster i.

2.2.2. Intracluster bit allocation

After the bit budget has been partitioned to each cluster, the bits need to be allocated to each of the n components within each cluster using existing bit allocation techniques for block quantisation [17]. In this case, the high resolution formula of [9] is used due to its computational simplicity[3]

$$b_{i,j} = \frac{b_i}{n} + \frac{1}{2} \log_2 \frac{\lambda_{i,j}}{[\prod_{j=1}^{n} \lambda_{i,j}]^{1/n}}$$
$$\text{for } i = 1, 2, \ldots, m, \tag{9}$$

where $b_{i,j}$ is the number of bits allocated to the jth component of cluster i.

2.3. Minimum distortion block quantisation

Fig. 2 shows a diagram of minimum distortion block quantisation. At first glance, it can be seen that it consists of m independent Gaussian block quantisers, Q_i, each with their own orthogonal matrix, $\boldsymbol{P}_i$, and bit allocations, $\{b_{i,j}\}_{j=1}^{n}$. The following coding process is described in [17].

To quantise a vector, $\boldsymbol{x}$, using a particular cluster i, the cluster mean, $\boldsymbol{\mu}_i$, is first subtracted and its components decorrelated using the orthogonal matrix, $\boldsymbol{P}_i$, for that cluster. The variance of each component is then normalised by the standard deviation to produce a decorrelated, mean-subtracted, and variance-normalised vector, $\boldsymbol{z}_i$

$$\boldsymbol{z}_i = \frac{\boldsymbol{P}_i(\boldsymbol{x} - \boldsymbol{\mu}_i)}{\boldsymbol{\sigma}_i}, \tag{10}$$

where $\boldsymbol{\sigma}_i = \sqrt{\boldsymbol{\lambda}_i}$. These are then quantised using a set of n Gaussian Lloyd–Max scalar quantisers as described in [9] with their respective bit allocations producing indices, $\boldsymbol{q}_i$. They are then decoded to give the approximated normalised vector, $\hat{\boldsymbol{z}}_i$, which is multiplied by the standard deviation and correlated again by multiplying with the transpose, $\boldsymbol{P}_i^{\mathrm{T}}$, of the orthogonal matrix. The cluster mean is then added back to give the reconstructed vector, $\hat{\boldsymbol{x}}_i$.

$$\hat{\boldsymbol{x}}_i = \boldsymbol{P}^{\mathrm{T}} \boldsymbol{\sigma}_i \hat{\boldsymbol{z}}_i + \boldsymbol{\mu}_i. \tag{11}$$

The distortion between this reconstructed vector and original is then calculated, $d(\boldsymbol{x} - \hat{\boldsymbol{x}}_i)$.

The above procedure is performed for all clusters in the system and the cluster, k, which gives the *minimum distortion* is chosen

$$k = \underset{i}{\operatorname{argmin}}\, d(\boldsymbol{x} - \hat{\boldsymbol{x}}_i) \tag{12}$$

It is noted that the search complexity is a function of the number of clusters, m, rather than the bitrate [17]. Thus it is a computational advantage over full search vector quantisers where the search complexity is an exponential function of the bitrate [7].

2.3.1. Quantiser index encoding

Each quantised vector has associated with it, a number identifying which cluster was used for coding. As proposed in [17], this side information can be made inherent in the encoding. For an m-cluster system operating at b bits per vector, $\log_2 m$ bits are required to uniquely identify each cluster. Therefore, on average, $b - \log_2 m$ bits are available for quantising each vector which is equivalent to $2^b/m$ quantiser levels. Hence, our range of quantiser levels has been partitioned into m segments. The decoder can easily determine which cluster the block belongs to by working out which partition the index falls into. This removes the need for extra side information to be transmitted.

[3]Riskin's bit allocation algorithm [14] may be used in place of the high resolution of [9] for better performance.

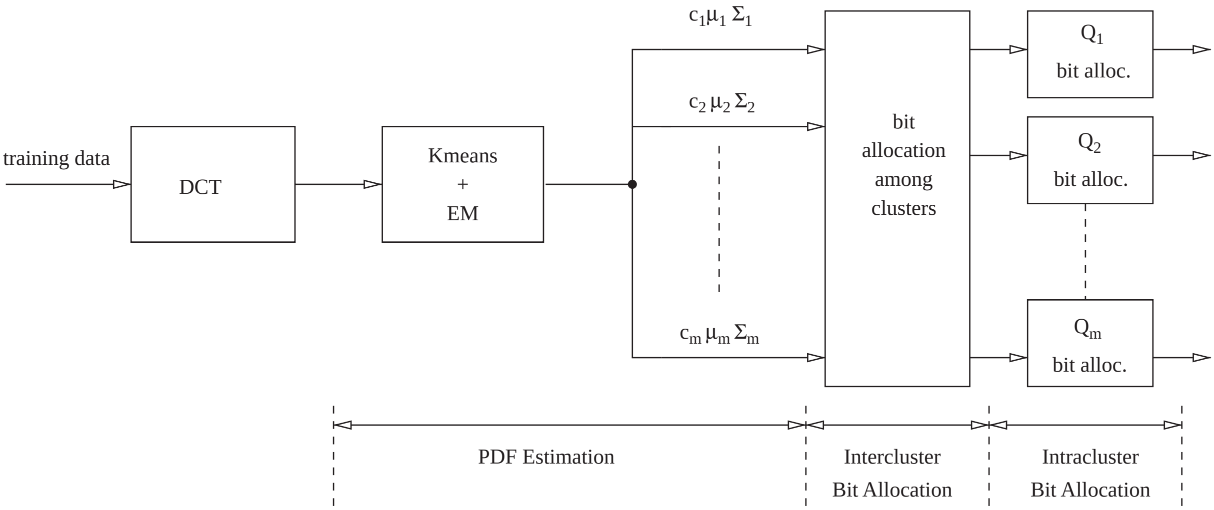

Fig. 3. PDF estimation and bit allocation procedure for the GMM–DCT block quantiser.

3. The modified GMM-based block quantiser using the DCT

By observing Fig. 2, we can see that a transformation and inverse transformation is performed for all clusters. Each cluster, with its own local statistics, will possess its own unique transformation matrix. In addition to this, because the orthogonal bases of the transform space vary between clusters, then any distance measure such as MSE will need to be performed in the original space, thus the need for inverse transformations. It all comes down to the source-dependent nature of the KLT. We may remove the extra steps of computation by replacing the KLT with a source independent transform that possesses similar properties. If we assume that the role of the KLT is to decorrelate samples before scalar quantising them independently, and that the source is Gauss–Markov with a high degree of correlation, then the discrete cosine transform (DCT) is a good candidate as it has been shown to approach the optimality of the KLT under these conditions [12].

The DCT transforms vectors, $\boldsymbol{x}$, by multiplying them with a transformation matrix, $\boldsymbol{D}$, which consists of cosine basis functions. For image applications, a separable two-dimensional DCT is commonly used [2]

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}\boldsymbol{D}^{\mathrm{T}} \tag{13}$$

$$\boldsymbol{D}(i,j) = c_i \cos\left(\frac{2j+1}{2n}\, i\pi\right) \tag{14}$$

$$c_i = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } i = 0, \\ \sqrt{\frac{2}{n}} & \text{otherwise,} \end{cases} \tag{15}$$

where $\boldsymbol{X}$ is the $p \times p$ block of pixels[4], $\boldsymbol{Y}$ is the transformed matrix 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 and hence the same matrix can be applied to all vectors generated from all sources.

Apart from replacing the KLT with the DCT as well as the resulting structural changes in the minimum distortion block quantisation to reduce computational complexity, the modified GMM-based block quantiser uses the same bit allocation and quantiser index encoding as that described in

[4]Note that $p \times p = n$.

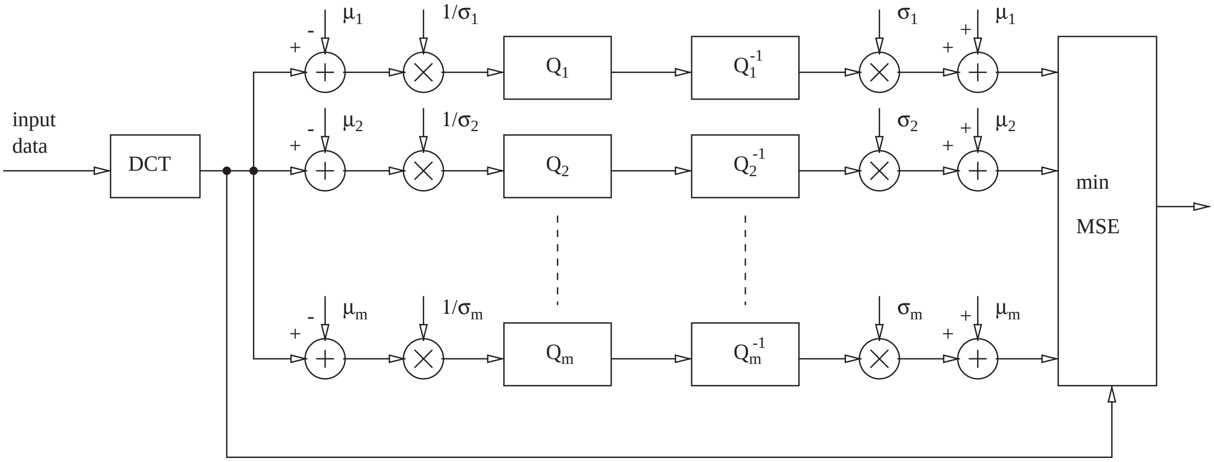

Fig. 4. Schematic of the modified GMM block quantiser based on DCT (Q—block quantiser).

the previous sections on the original KLT-based scheme.

3.1. PDF estimation

Fig. 3 shows the block diagram of the PDF estimation procedure for the GMM–DCT-based block quantiser. By using a GMM, the PDF is modelled as a composite source of Gaussians. Assuming that these mixture components are themselves Gauss–Markov processes, then applying a DCT on each mixture component should not impact much on the distortion performance. Since the DCT is source independent and fixed, then only one transform needs to be performed. Therefore, each image block is transformed first by the DCT and the GMM is estimated based on the DCT coefficients. It is assumed that the vectors will be decorrelated by the DCT, hence only diagonal covariance matrices are used in the EM algorithm.

3.2. Minimum distortion block quantisation

Fig. 4 shows the block diagram of minimum distortion block quantisation for the GMM–DCT-based coder. As the DCT bases are constant, there is no need for multiple transformations as well as inverses for distance measurements. It is helpful to compare the computational complexity (in flops[5]) of the two types of transformations. The computational complexity of a single KLT is $2p^4 - p^2$ or 8128, assuming 8×8 blocks. For an m-cluster system, the number of flops is therefore $4mp^4 - 2mp^2$ (including inverse transformation). For the case of the DCT, the computational complexity is constant at $4p^3 - 2p^2$ or 1920 for a block of 8×8 for all values of m. Therefore, for a 16 cluster GMM-based block quantiser, the number of flops used for the transformation step in the DCT-based scheme is less than 1% of that required in the KLT-based scheme.

4. Experimental results

During the training process, 18 greyscale 8-bit images of size 512×512 were used as the training set. The dimension of the vectors was 8×8 and in total, 73,728 vectors were available for training. 20 iterations of the EM algorithm were performed while Gaussian Lloyd–Max scalar quantisers of up to 256 levels (8 bits) were used. The test set includes all the training images plus an additional six images that are not part of the training set.

Fig. 5 shows the distortion rate characteristic of the GMM–DCT-based block quantiser on the

[5] In this study, we consider each multiplication and addition to represent one floating point operation (flop).

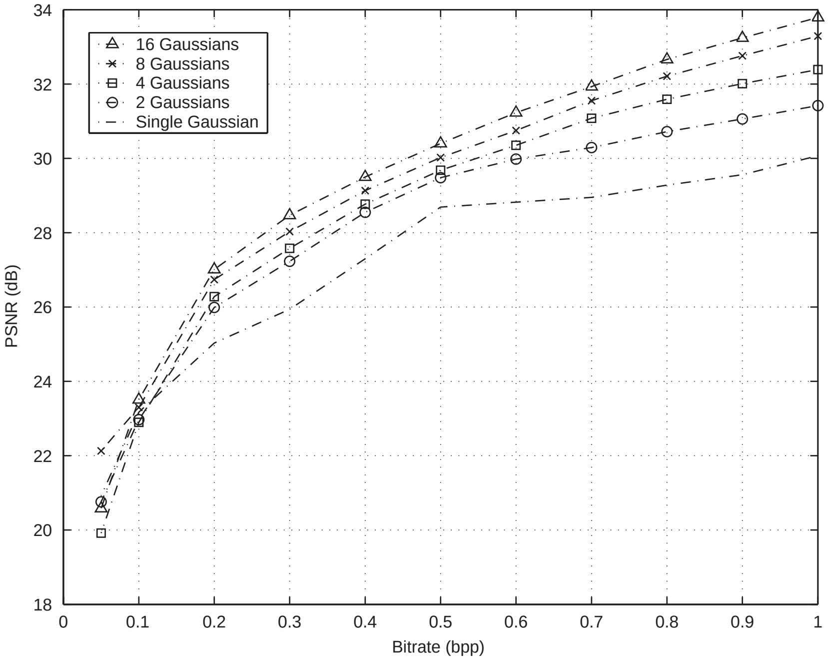

Fig. 5. Operating distortion-rate performance for the image 'goldhill' (part of training set).

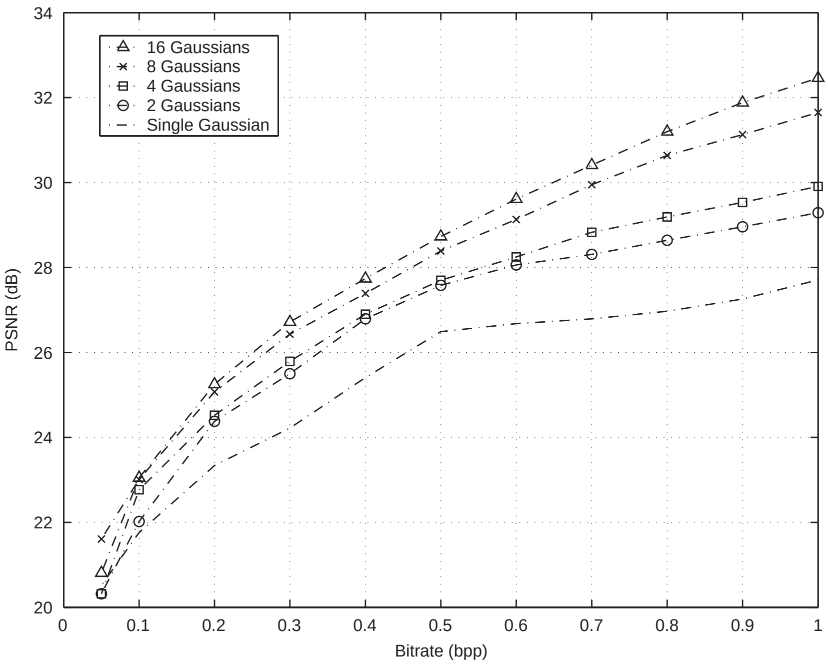

Fig. 6. Operating distortion-rate performance for the image 'boat' (not part of the training set).

Fig. 7. 'Goldhill' image (part of training set): (a) original 8-bit image; (b) single Gaussian DCT at 0.5 bpp (28.69 dB); (c) GMM–DCT with 2 clusters at 0.5 bpp (29.48 dB); (d) GMM–DCT with 16 clusters at 0.5 bpp (30.40 dB).

image 'goldhill', which was part of the training set. The results for a single Gaussian block quantiser are also presented as a baseline. This baseline block quantiser is similar to the one described in [9] except that a DCT is applied, rather than a KLT, to decorrelate the vectors which are then coded by a set of n Lloyd–Max Gaussian quantisers. Its purpose in the experiment is to show how much improvement has resulted from using multiple Gaussians to model the PDF rather than just a single one. Fig. 5 shows that the PSNR of the GMM–DCT-based block quantiser is at least 1–4 dB better at medium bitrates (around 1 bpp) and 1–2.5 dB better at low bitrates (around 0.3 bpp) than the basic DCT-based block quantiser. As the number of clusters is increased, performance gains are realised since with more clusters, the true PDF of the training data set is estimated better than with a lesser number of clusters. 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 bitrates (less than 0.1 bpp) where the GMM–DCT-based block quantiser is performing worse than the basic DCT-based block quantiser due to the increased influence of the effective bits required to identify clusters.

Fig. 6 shows the distortion rate characteristic of the GMM–DCT-based block quantiser on the image 'boat', which was not part of the training set. At medium bitrates (around 1 bpp), the GMM–DCT-based block quantiser is about 2–5 dB better and about 1–2.5 dB better at low bitrates (around 0.3 bpp) than the single Gaussian

Fig. 8. 'Boat' image (not part of the training set): (a) original 8-bit image; (b) Gaussian DCT-based block quantiser at 0.5 bpp (26.49 dB); (c) GMM–DCT-based block quantiser using 2 clusters at 0.5 bpp (27.58 dB); (d) GMM–DCT-based block quantiser using 16 clusters at 0.5 bpp (28.73 dB).

block quantiser. We can see that the PDF model works well for images which are not part of the training set.

The improvements to image quality resulting from more accurate modelling of the PDF can be seen in Fig. 7 where the image 'goldhill' was compressed using traditional block quantisation (single Gaussian) as well as using the improved GMM-based block quantiser at a bitrate of 0.5 bits per pixel. In Fig. 7(b), it can be seen that the sky as well as the ground looks very grainy. Also, there is a large degree of blocked distortion observable in region of the houses, especially on the edges of the white house in the centre of the picture. By using a 2 cluster GMM to model the PDF, one can see in Fig. 7(c) that the sky and ground area appear much smoother. There is some apparent distortion observable in the fields immediately behind the houses. In Fig. 7(d), where a 16 cluster GMM is used, the sky and ground are considerably smoother and less noisy and distortions in the fields has been considerably reduced. Similarly, Fig. 8 shows the image 'boat' which is not part of the training set. In Fig. 8(b), where a traditional DCT-based block quantiser was used, the smooth areas such as the sky and the black bottom of the boat are very grainy. There is also block distortion on the white sides of the boat. It can be observed that in Figs. 8(c) and (d), as more clusters are used, the graininess of the smooth regions has been reduced as well as the block distortions.

Table 1 shows three sets of PSNRs for all images in the training set, at a rate of 1 bit per pixel.

Table 1
Coding results for images in the training set using different techniques

Image name	PSNR (dB) at 1 bpp		
	DCT-1	DCT-16	KLT-16
Baboon	24.36	26.53	26.76
Barbara	24.89	28.00	28.80
Fruits	27.75	33.10	33.56
Lena	29.24	35.44	35.66
Peppers	28.89	35.29	35.39
Sailboat	27.68	32.07	32.22
Man	26.34	29.34	29.52
Goldhill	30.07	33.80	33.90
Bridge	26.48	28.63	28.72
Jet	27.28	33.90	34.21
Pyramid	29.04	35.77	35.94
Aero	29.19	34.87	35.10
Einstein	30.83	37.76	38.01
Hat	28.77	35.09	35.24
London	29.74	36.28	36.32
Tekboat	21.97	25.73	26.00
Tekrose	20.16	23.15	23.38
Loco	23.43	27.75	27.99

Table 2
Coding results for images not part of the training set using different techniques

Image name	PSNR (dB) at 1 bpp		
	DCT-1	DCT-16	KLT-16
Boat	27.71	32.46	32.59
Kids	24.43	26.95	26.99
Crowd	20.02	23.91	24.14
Hat	28.77	35.09	35.25
Mill	24.14	27.45	27.65
Vegas	29.49	35.37	35.59

Table 3
Comparison of average processing times (in seconds)

No. of clusters	GMM–DCT	GMM–KLT
2	6.5	14.8
4	8.6	30.6
8	13.2	59.5
16	21.5	120

'DCT-1', 'DCT-16', and 'KLT-16' are the traditional DCT-based (single Gaussian), GMM–DCT-based (16 clusters) and GMM–KLT-based (16 clusters) block quantisers, respectively. From these results, it can be observed that the benefits of modelling the PDF as a mixture of Gaussians are more or less retained in the modified DCT-based version. A gain of approximately 6–7 dB in the images 'Einstein', 'aero', 'hat', etc. has been achieved. As expected, due to the suboptimal decorrelating nature of the DCT, there is a slight degradation in the performance compared with the KLT-based version, though it is not significant. Therefore, it can be concluded that each Gaussian mixture component of the GMM has characteristics which are similar to Gauss–Markov processes and hence, the DCT is able to approximate the KLT in terms of decorrelation and energy packing ability.

Table 2 shows three sets of PSNRs for images that were not part of the training set, at a rate of 1 bit per pixel. We can see that the PDF model represented by the GMM works quite well for these images. For the images 'hat', 'vegas', and 'boat', there are gains about approximately 4–6 dB from the single Gaussian block quantiser. Also, compared with the KLT-based scheme, there are slight degradations of less than 0.5 dB due to the suboptimality of the DCT.

At the expense of slightly degraded performance, using the DCT results in less computations. Table 3 shows the average time, in seconds, it takes to code 19 test images at a bitrate of 1 bit per pixel. In total, 77824 image blocks were coded on an Intel Pentium 4 system running at a clockspeed of 2.4 GHz. The number of clusters was increased (2, 4, 8, 16) and the performance times are averaged. It can be seen that the DCT version of the coder is considerably faster than the KLT-based one. It is expected that the differences in time would be even more if better optimised DCT algorithms were used.

5. Conclusion and further work

In this paper, a GMM-based block quantiser, based on the discrete cosine transform, was presented. The assumption made was that each Gaussian cluster in the GMM can be approximated by a Gauss–Markov process, which would allow the DCT to approach the optimality of the KLT. The advantages of the DCT are its source independent nature and computational efficiency. With the transformation matrix being independent of the source, only a single DCT needs to be performed on the image blocks rather than individual KLTs and inverse KLTs for each Gaussian cluster. Also, there are many efficient implementations of the DCT available. Therefore, a GMM-based block quantiser based on the DCT is computationally less complex than a KLT-based one. Experimental results show that the GMM–DCT-based block quantiser provides improvements, similar to those observed in the GMM–KLT-based versions over the traditional single Gaussian block quantiser. Therefore, the Gauss–Markov assumption appears reasonable with gains of 6–7 dB observed for some images.

The accurate modelling of multimodal PDFs using GMMs is a relatively new concept in coding. Further work would involve exploring its use in other types of coding where sources possess multimodal PDFs.

References

[1] C. Archer, T.K. Leen, A generalized Lloyd-type algorithm for adaptive transform coder design, IEEE Trans. Signal Processing 52 (1) (January 2004) 255–264.
[2] V. Bhaskaran, K. Konstantinides, Image and Video Compression Standards, second ed., Kluwer International Series, Boston, 1997.
[3] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Royal Stat. Soc. 39 (1977) 1–38.
[4] W.R. Gardner, B.D. Rao, Theoretical analysis of the high-rate vector quantization of LPC parameters, IEEE Trans. Speech Audio Processing 3 (September 1995) 367–381.
[5] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Massachusetts, 1992.
[6] V.K. Goyal, Theoretical foundations of transform coding, IEEE Signal Processing Mag. 18 (5) (September 2001).
[7] R.M. Gray, D.L. Neuhoff, Quantization, IEEE Trans. Inform. Theory 44 (6) (October 1998).
[8] P. Hedelin, J. Skoglund, Vector quantization based on Gaussian mixture models, IEEE Trans. Speech Audio Processing 8 (4) (July 2000) 385–401.
[9] J.J.Y. Huang, P.M. Schultheiss, Block quantization of correlated Gaussian random variables, IEEE Trans. Commun. Syst. CS-11 (September 1963) 289–296.
[10] K.P. Kramer, M.V. Mathews, A linear coding for transmitting a set of correlated signals, IRE Trans. Inform. Theory (Corresp) IT-17 (November 1971) 751–752.
[11] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. 28 (1) (January 1980) 84–95.
[12] A.N. Netravali, J.O. Limb, Picture coding: a review, Proc. IEEE 68 (3) (March 1980) 366–406.
[13] A. Ortega, M. Vetterli, Adaptive scalar quantization without side information, IEEE Trans. Image Proc. (1996).
[14] E.A. Riskin, Optimal bit allocation via the generalized BFOS algorithm, IEEE Trans. Inform. Theory 37 (2) (1991) 400–402.
[15] S. So, K.K. Paliwal, Efficient block coding of images using Gaussian mixture models, in: Proceedings of the Fourth Australasian Workshop on Signal Processing and Applications 2002, September 2002, pp. 71–74.
[16] J.K. Su, R.M. Mersereau, Coding using Gaussian mixture and generalized Gaussian models, in: IEEE International Conference on Image Processing, Lausanne, Switzerland, 1996, pp. 217–220.
[17] A.D. Subramaniam, B.D. Rao, PDF optimized parametric vector quantization of speech line spectral frequencies, IEEE Trans. Speech Audio Processing 11 (2) (March 2003) 130–142.
[18] G.K. Wallace, The JPEG still picture compression standard, Commun. ACM 34 (4) (April 1991) 30–44.
[19] P.A. Wintz, Transform picture coding, Proc. IEEE 60 (7) (July 1972) 809–820.