Comparison of Quantisation Techniques for DCT-based Image Coding

Stephen So and Kuldip K. Paliwal

Abstract— **In this paper, preliminary results of applying different types of scalar quantisers and vector quantisers to lossy image coding are presented. The advantages of these different methods become evident when used in conjunction with a standard DCT-based image coder. It is shown that there is negligible improvement in applying vector quantisation together with the DCT. This comparative study between different quantisation techniques constitutes the first six months of research work and will provide the basis for further research into improving the efficiency of the DCT-based image coder.**

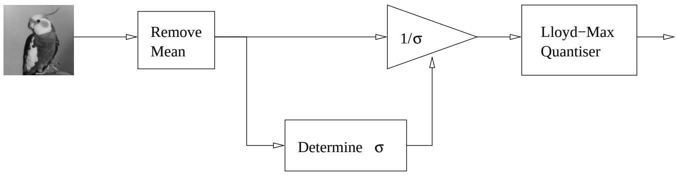

Fig. 1. Image Coding using Lloyd-Max Quantiser

I. Introduction

The DCT or *discrete cosine transform* is the predominant transform used in transform image coding. It is used in image and video coding standards such as JPEG and MPEG. The information loss in the DCT-based coder mostly occurs in the quantisation stage and improvements in rate-distortion performance can be realised by designing more efficient quantisers. It is the aim of this paper to present a comparison of different quantisation techniques used in conjunction of a DCT-based coding system.

II. Quantisation Theory

Quantisation is one of the oldest techniques for coding or compressing a signal. Quantisation is an irreversible process where data values x_i are non-linearly mapped to a finite set of values called *reconstruction values*:

$$y_i = Q[x_i] \quad \text{for } x_i \in S_i \tag{1}$$

with the quantiser Q consisting of reconstruction values $C = \{y_i; i \in \mathbf{I}\}$, intervals $S = \{S_i; i \in \mathbf{I}\}$, and for each cell $S_i = (a_{i-1}, a_i]$ where the a_i's are *thresholds* [1]. The technique was originally motivated in PCM (Pulse code modulation) coding where it was necessary to map continuous analog values to discrete levels. *Scalar quantisation* represents each data value (scalar) with a reconstruction level and hence is a *one-to-one* mapping. It can be showed that significantly improved performance could be achieved if groups of the input values or vectors of arbitrarily large dimension are coded [1]. This *many-to-one* mapping is termed *vector quantisation*.

A. PDF-optimised Non-uniform Scalar Quantisation

The operational distortion-rate performance $\delta(R)$ is related to the average distortion of the quantiser $D(q)$ for an input X with the probability density function or PDF $f_X(x)$:

$$D(q) = \sum_i \int_{S_i} d(x, y_i) f_X(x) dx \tag{2}$$

This indicates that the performance of the quantiser depends very much on the PDF of the input. Therefore, in order to minimise the quantisation error, it is important that the quantiser is designed to match the input PDF.

Lloyd and Max both presented algorithms for designing a quantiser that matches the input PDF. In [2] and [3], optimum reconstruction levels are given for standard PDFs such a Gaussian, Laplacian, and gamma with zero mean and unit variance. To quantise an image using a Lloyd-Max quantiser using the standard PDFs, the mean is subtracted from the data and then divided by the standard deviation (see Figure 1).

The GLA or *generalised Lloyd algorithm* is an iterative procedure for designing non-uniform quantisers to match the PDF of the input data. It consists of quantising the input or training data using arbitrary levels, finding the centroids in each quantisation cell, and recalculating the decision levels as the mid-points between the centroids. This process is continued until a minimum distortion criteria is met [1].

B. Vector Quantisation

Vector quantisers (VQ) have received considerable attention due to their ability to achieve very low bit rates. Rather than using thresholds, vector quantisers use a least distortion criteria for assigning reconstruction values which are stored in a table or *codebook*. The codebook is generated via training and uses different techniques such as the GLA (Generalised Lloyd Algorithm), LBG (Linde-Buzo-Gray) splitting method, PNN (pairwise nearest neighbour), etc. [1] The codebook training designs a vector quantiser that matches the input PDF and results in the least distortion.

An input vector is quantised by finding the closest reconstruction vector in the codebook and then replacing it with its index. However, the storage requirements and computational complexity have hindered the use of vector quantisers. For a vector quantiser of rate R and a vector dimension k, the size of the codebook is 2^{kR}. Using structured codebooks can solve these problems. *Multistage vector quantisers*, often called *residual vector quantisers*, solve the problem of exponentially growing codebooks for large vector dimensions. Referring to Figure 2, the input image is coarsely quantised using vector quantiser $Q1$. The difference or *residual* between the input and reconstructed image is found and is quantised using another vector quantiser $Q2$. If C_n is the codebook size of quantiser n in an N-stage multistage VQ, the total codebook size is given as a summation $\sum_{n=0}^{N-1} C_n$ rather than a product $\prod_{n=0}^{N-1} C_n$.

III. DCT-based Image Coding

The DCT or *discrete cosine transform* is one of the most popular transforms used in image coding since it achieves near-

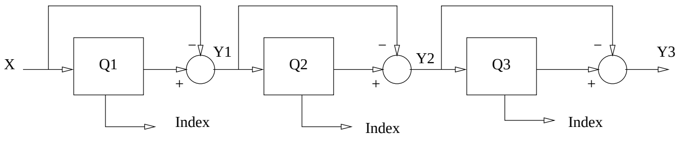

Fig. 2. Multistage Vector Quantiser (Three stage)

optimal energy compaction and decorrelation as well as being image independent. It is noted that for typical image data, the two-dimensional covariance function, which indicates the amount of correlation, decays rapidly beyond 8 pixels. With the aim of the DCT being to decorrelate the image pixels, 8×8 blocks are therefore used. The JPEG (Joint Photographic Experts Group) image coding standard performs a DCT on each 8×8 block, scalar quantises the DCT coefficients using fixed quantisation tables, performs zigzag RLE (runlength encoding), and Huffman encodes AC coefficients while DC coefficients are differentially Huffman encoded to take advantage of inter-block correlation [4].

IV. Designing Efficient Quantisers for DCT Coefficients

Uniform scalar quantisers and standard Lloyd-Max scalar quantisers perform poorly on DCT coefficients since the PDF is highly skewed toward the low magnitude coefficients. This is due to the fact that the DCT compacts the image energy into a few coefficients while the majority of coefficients are close to zero.

Improvements can be realised by designing a non-uniform scalar quantiser that adaptively matches the PDF of each DCT block. However, quantiser parameters such as the reconstruction values need to be transmitted as side information. The large amount of side information significantly offsets the improvement in performance.

There are also difficulties in applying a vector quantiser to the DCT-based system as the 8×8 blocks constitute 64 dimensional vectors. Unstructured vector quantisers handling such large dimensional vectors have exorbitant memory and computational requirements, hence multistage vector quantisers may be a likely candidate.

V. Experiments and Results

The 8-bit 512×512 greyscale image, *Lena*, was used in the experiments. The quantisation techniques examined include the following:

- Uniform scalar quantiser
- Lloyd-Max non-uniform scalar quantiser designed for a Gaussian PDF
- Non-uniform quantiser optimised for input PDF using GLA
- Full-search vector quantiser (unstructured)
- Multistage vector quantiser
- Uniform scalar quantiser + block DCT
- Multistage vector quantiser + block DCT

From the graph in Figure 3, the scalar quantiser designed using the GLA algorithm outperforms all other techniques. This shows that a scalar quantiser that is designed to match the input PDF generally performs better. The Lloyd-Max quantiser

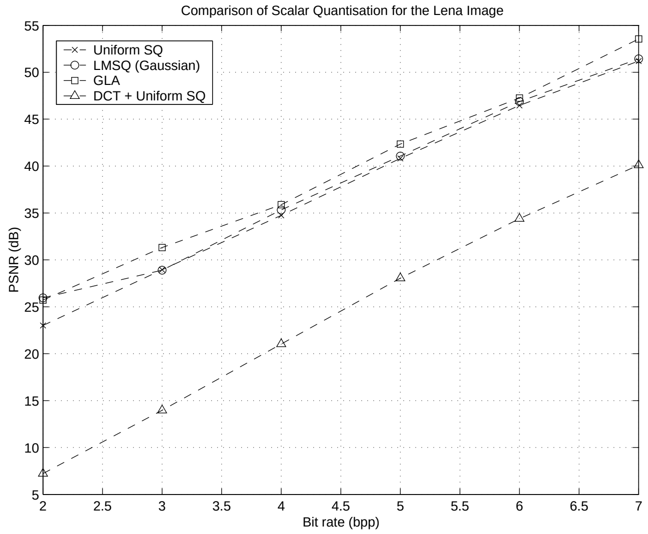

Fig. 3. Comparison of Scalar Quantisation Techniques

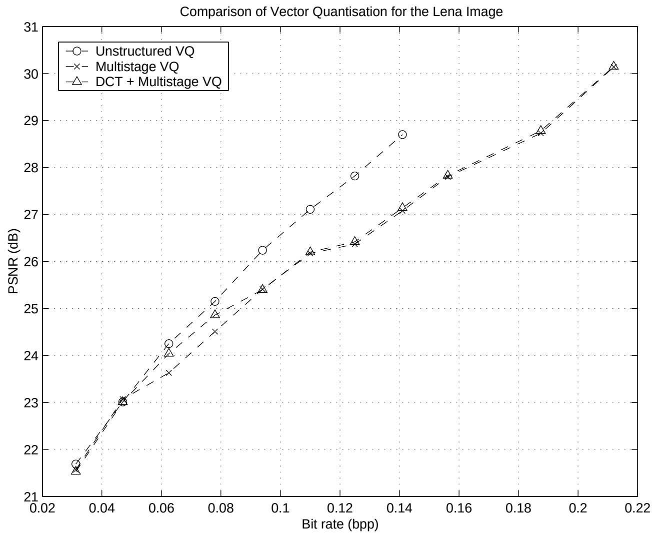

Fig. 4. Comparison of Vector Quantisation Techniques

Fig. 5. Uniform Scalar Quantisation (28.95 dB at 3 bpp)

Fig. 6. Non-uniform Scalar Quantisation using GLA (31.31 dB at 3 bpp)

Fig. 7. Unstructured Vector Quantisation (27.82 dB at 0.125 bpp)

Fig. 8. Multistage Vector Quantisation (26.37 dB at 0.125 bpp)

Fig. 9. DCT + Multistage Vector Quantisation (26.42 dB at 0.125 bpp)

(Gaussian) does not show considerable improvement over the uniform scalar quantiser. This is explained by the fact that the PDF of the image is not approximated accurately with a Gaussian. The DCT + Uniform SQ performs the worst since the PDF of the coefficients is highly skewed toward the smaller coefficients and this results in a large granular error.

Figure 4 shows that even though multistage VQ allows higher bit rates to be achieved without an exponentially growing codebook, there is a sacrifice in the PSNR. However, the unstructured VQ uses a considerably larger amount of memory and computation at bit rates close to 0.14 bpp compared with the multistage VQ. Unstructured vector quantisers operating at above the rate of 0.14 bpp become impractical.

The DCT + Multistage VQ shows negligble improvement. This can be attributed to the fact that the DCT decorrelates the data and this significantly reduces the efficiently of VQ [1].

VI. CONCLUSION

While higher bit rates and large dimensional vectors are handled by multistage vector quantisers, they show negligible improvements when used in conjunction with the DCT-based coder.

Scalar quantisers that are designed to match the input data PDF have shown to outperform the uniform SQs. It is expected that DCT coefficients can be efficiently coded using these quantisers. Further work would involve making improvements over the existing JPEG standard, which uses fixed quantisation tables, by designing PDF-matching scalar quantisers that adapt to each DCT block.

REFERENCES

[1] R.M. Gray & D.L. Neuhoff, ‘Quantisation’, *IEEE Transactions on Information Theory*, Vol. 44, No. 6, October 1998.

[2] J. Max, ‘Quantising for Minimum Distortion’, *IRE Trans. Inform. Theory*, Vol. IT-6, pp. 7-12, March 1960.

[3] M.D. Paez & T.H. Glisson, ‘Minimum Mean-Squared-Error Quantisation in Speech PCM and DPCM System’, *IEEE Trans. Commun.*, vol. COM-20, pp.225-230, April 1972.

[4] G.K. Wallace, ‘The JPEG Still Picture Compression Standard’, *Communications of the ACM*, Vol. 34, No. 4, Association of Computing Machinery, pp. 30-44, April 1991.