NOVEL CODEBOOK GENERATION ALGORITHMS FOR VECTOR QUANTIZATION IMAGE COMPRESSION

K. Masselos, T. Stouraitis, C. E. Goutis
VLSI Design Laboratory
Department of Electrical and Computer Engineering
University of Patras, Rio 26500, Greece
tel.: (+)30 61 997323
fax: (+)30 61 994798
email: masselos@ee.upatras.gr

ABSTRACT

Novel algorithms for vector quantization codebook design are presented in this paper. Two basic techniques are proposed. The first technique takes into consideration specific characteristics of the blocks of the training sequence during the generation of the initial codebook. In this way a representative initial codebook is generated. Starting from a high quality initial codebook the iterative optimization procedure converges fast to a representative final codebook which in turn leads to high output image quality. The second proposed technique extends small codebooks computationally. The main idea is the application of simple transformations on the codewords. This technique reduces the memory requirements of the traditional vector quantization making it useful for applications requiring low-power consumption.

1. INTRODUCTION

Vector quantization [3] is an efficient image coding technique achieving low bit rates i.e lower than 1 bit per pixel. The principal component of vector quantization is a codebook of code vectors (codewords). A vector of input samples (block of pixels) is approximated by a representative vector (codeword) of the codebook. The representative codeword is the one that minimizes the distortion among all the codewords in the codebook. Compression is achieved by transmitting or storing the codeword address (index) instead of the codeword itself. The operation of vector quantization is described by the following equation:

$$
\begin{aligned}
&Q : R^k \rightarrow C \\
&C = \{y_1, y_2, \ldots y_N\}, y_i \in R^k \; \forall i = 1,2,\ldots,N \qquad (1) \\
&y_i = Q(x) \quad if\; d(x, y_i) \le d(x, y_j) \\
&for \quad all \quad i, j = 1,2,\ldots,N.
\end{aligned}
$$

where R^k is the k-dimensional space, C is the codebook of N k-dimensional words y_i and d is the distortion criterion used.

Vector quantization has been studied in a variety of contexts but basically for signal coding. A lot of work has been carried out on vector quantization for speech coding and slightly more recently for image and video coding [1]. Results on the direct application of certain vector quantization techniques to image coding prove that vector quantization is a high performance, effective coding technique [7]. Furthermore of equal significance is the role of vector quantization as a component in more sophisticated compression schemes, such as subband coding, predictive coding and even in lossless compression algorithms. In fact vector quantization is used with greater frequency in this way [7].

Especially in its direct applications the performance of vector quantization as far as output image quality is concerned is highly dependent on the codebook. In this paper a new algorithm for codebook design is described. The codebook design procedure is based on characteristics of image blocks. A novel technique for computational extension of small codebooks is also proposed. The basic idea is the application of simple transformations on the codewords during encoding.

The rest of the paper is organised as follows. In section 2 a review of codebook design techniques is attempted. In section 3 the proposed codebook design algorithm is described. In section 4 the technique for computational codebook extension is presented. Finally in section 5 conclusions are offered.

2. CODEBOOK DESIGN TECHNIQUES

The two basic tasks of vector quantization image compression are: a) The design of the codebook and b) The search for the best approximation (codeword) for each block. As far as codebook design a large number of algorithms exist and have been reported in the literature. The most popular technique is the LBG [6] or Generalized Lloyd Algorithm (GLA) [3]. The key point in the codebook design procedure is the design of a good initial codebook which will require a small number of optimization iterations and will lead to a final codebook of good quality.

The LBG algorithm uses a technique based on splitting for the design of the initial codebook. This technique starts with small codebooks and generates larger ones. More specifically the first codebook consists of only one codeword which is the centroid [3] y of the training sequence. This codebook can be splitted to two codewords y and $y+e$ where e is a vector of the training sequence with small Euclidean magnitude [3]. The new codebook cannot be worse than the first as it includes it. In this way an initial codebook of M codewords can be generated by adding and subtracting the values ke to and from the centroid of the training sequence where $k=1,2,\ldots M/2$. Thus the LBG initial codebook is centroid based. The LBG is based on clustering analysis to optimize the initial codebook. Specifically LBG performs an iterative optimization of the initial codebook. In each iteration every codeword is replaced by the centroid of the blocks of the training sequence that are best approximated by this specific codeword. The iterative optimization procedure stops when the rate of the decrement of the distortion falls below a specific predefined threshold. The LBG algorithm is widely used in image compression systems using vector quantization. The basic disadvantages of the LBG algorithm is the computational complexity because of the iterative optimization procedure and the fact that it does not guarantee global optimal solutions.

In this paper an algorithm that improves LBG performance is presented. The proposed algorithm is based on a different strategy for the design of the initial codebook. Specific features of the blocks of the training sequence are taken into consideration during the design of the initial codebook. This leads to high quality final codebook and to high speed codebook design procedure.

3. PROPOSED ALGORITHM FOR CODEBOOK DESIGN

The proposed algorithm for codebook design adopts a novel strategy for the design of the initial codebook. Specifically it attempts to exploit the special features of the blocks of the training sequence. Several block features can be used. Assuming a size of the training sequence blocks 4×4 pixels (typical size in vector quantization image compression), some simplified representative block features are described by the following equations, where X is a block of the training sequence.

$$Magnitude = \sum_{i,j=1}^{4} X_{ij}^{2} \qquad (2)$$

$$Mean\ value = \frac{\sum_{i,j=1}^{4} X_{ij}}{16} \qquad (3)$$

$$Variance = \sum_{i,j=1}^{4} \left| X_{ij} - Mean\ value \right| \qquad (4)$$

$$Shape = \sum_{i,j=1}^{4} \left| X_{ij} - R_{ij} \right| \qquad (5)$$

Where R is a reference block randomly selected from the training sequence. The shape of a block of the training sequence is in general difficult to be modelled. The angle of each block with respect to a reference can be considered as a good metric of the block shape. However the absolute sum of the differences of the corresponding pixels of the block and the reference block is much simpler and had better performance than the angle during experiments and thus it was the final choice.

Based on these features the initial codebook design procedure requires classification of the blocks with respect to the above features. The selected feature each time is computed for all the blocks in the training sequence. If an initial codebook of M codewords is required the range of values of the selected feature is divided to M subranges. The centroids of the blocks of the training sequence that have feature values in the same subrange are computed. The resulting M centroids form the initial codebook. This procedure for the design of the initial codebook is slightly more complicated than the corresponding used by LBG. Specifically assuming a training sequence of L 4×4 pixels blocks and a desired codebook size of M codewords, LBG requires $16\times L$ additions and one division by L for the computation of the centroid and $16\times(M-1)$ additions/subtractions for the generation of the initial codebook. The proposed technique requires L evaluations of the selected block feature (computational requirements for each block feature are given by eqns (2)-(5)) as well as $16\times L$ additions and M divisions. It can be said that the overhead of the proposed technique equals to L evaluations of the block features minus $16\times(M-1)$ additions/subtractions required by LBG for the initial codebook.

However the proposed technique leads to the design of a more representative initial codebook. This can be proved by the results of table 1. Two test images Lena and Man of 256×256 pixels were encoded by full-search vector quantization using as codebooks the initial codebooks generated by the splitting technique incorporated in LBG and the proposed technique for the different block features.

Algorithm	Image	Codebook	SNR
Proposed (Magnitude)	Lena	256	26.1
Proposed (Mean value)	Lena	256	26.4
Proposed (Variance)	Lena	256	25.4
Proposed (Shape)	Lena	256	25.5
LBG	Lena	256	25
Proposed (Magnitude)	Man	256	23.5
Proposed (Mean value)	Man	256	24
Proposed (Variance)	Man	256	22.7
Proposed (Shape)	Man	256	22.3
LBG	Man	256	19.1

Table 1: Performance of initial codebooks.

The initial codebook generated by the procedure described above is then iteratively optimized using the same procedure as LBG.

The fact that more representative initial codebooks are designed and used by the proposed algorithm leads to faster iterative optimization procedure of the initial codebooks in comparison to LBG. This means that the proposed algorithm converges faster to its final solution which is in general better than the corresponding of LBG (in terms of SNR of the output image). The comparison (using the same training sequence and the same completion threshold for the iterative optimization procedure) of the two algorithms is shown in table 2.

The smaller number of iterations required by the proposed algorithm implies a reduced computational complexity in comparison to LBG. The number of operations required by an iteration are $L\times M\times16$ additions, $L\times M\times16$ subtractions and

$L \times M \times 16$ multiplications where L is the size of the training sequence and M the codebook size. From table 2 it is obvious that significant savings in number of iterations are achieved by the proposed algorithm. Thus the savings in computational complexity resulting from the smaller number of iterations are far larger than the computational overhead introduced in the design of the initial codebook (which is smaller than the computation required by a single iteration). This is very important for applications using a codebook updating strategy [4]. The reduced computational complexity may decrease the area and power consumption [2] of the hardware that generates the codebooks (update procedure). Another important point is that the two block features used for the design of the initial codebook can also be used to reduce the size of the final codebook by merging codewords with relative magnitude and shape values. This can lead to increase of the compression ratio without significant drop of the final quality depending on the amount of merging to be performed.

Algorithm	Image	Codebook	Iterations	SNR
Proposed (Magn.)	Lena	256	27	29.7
Proposed (Mean)	Lena	256	19	29.5
Proposed (Varian.)	Lena	256	25	29.8
Proposed (Shape)	Lena	256	29	29.9
LBG	Lena	256	70	29
Proposed (Magn.)	Man	256	18	26.9
Proposed (Mean)	Man	256	18	27
Proposed (Varian.)	Man	256	20	26.8
Proposed (Shape)	Man	256	30	26.7
LBG	Man	256	42	26.6

Table 2: Performance of the two algorithms.

Output images of proposed and LBG algorithms are shown in figures 1, 2.

4. COMPUTATIONAL CODEBOOK EXTENSION

The basic disadvantage of classical full-search vector quantization is its computational complexity which is proportional to codebook size N. Another important point is that classical full-search vector quantization can be characterized as a rather “static” encoding procedure since each image block is approximated by one codeword of a predefined codebook. A technique aiming at reducing the memory requirements of vector quantization while being more dynamic during encoding is proposed. The basic idea is the use of small codebooks ($N \leq 64$ as a codebook of 256 words at least is usually required by most image and video processing applications) generated by any codebook design algorithm which are computationally extended. This is achieved by applying simple transformations on the codewords during encoding. The idea of using transformations during encoding is used by fractal image compression [5]. The encoding procedure requires for each block, a search for the set of codeword/transformations, that minimizes the distortion criterion. In the decoding, the transformations selected during encoding are applied on the selected codeword.

To illustrate the proposed technique two simple transformations are selected: Shifting transformation [5] and isometries [5]. Shifting transformation simply adds a constant value to all the pixels of a codeword or image block. As constant value the difference of the mean values of the image block under encoding and the candidate codeword is used. Isometries are transformations that simply shuffle the pixels of an image block or codeword in a deterministic way. There are eight canonical isometries [5]. These transformations are applied to all the codewords during the encoding of an image block. Each block is encoded by the index of the selected codeword, the constant value used for shifting and the index of the selected isometry. The use of more than one parameters for the encoding of each block introduces a penalty in the compression ratio in comparison to classical vector quantization (although the proposed technique uses smaller codebooks in general). To keep the number of bits required for the representation of each block as low as possible the following selection procedure is used. A condition related to the size of the selected shifting constant is set. This means that the selected shifting constant should be smaller than a selected value V. For the representation of the constant, Log_2V bits would be required however $Log_2V\text{-}2$ could be used without introducing significant penalty in terms of the quality of the output image. For example for shifting constants smaller or equal to 64, 6 bits are required however 4 bits can be also used. In this way for each image block only the pairs codewords-constants with constants smaller than V can be used to approximate the block. From these pairs the one that minimizes the distortion is selected. This is a joint optimization procedure of both the output image quality and the compression ratio. As far as the isometries are concerned simulation results proved that only 4 isometries can achieve almost the same quality as the 8 canonical isometries.

Thus the number of bits required for the encoding of each block is $Log_2N+(Log_2V\text{-}2)+2$ where N is the codebook size. The shifting constant is quantized to Log_2V-2 bits (The dynamic range of the shifting constant is still 0-V). Finally 2 bits are required to encode the selected isometry. Classical vector quantization requires Log_2N bits for the encoding of an image block however the proposed technique uses codebooks with far smaller size N. During decoding the shifted constant is added to the selected codeword and then the selected isometry is applied on the shifted codeword. Results of this technique are shown in table 3. The codebooks were generated by LBG.

Image	Codebook	Bits/pixel	SNR
Lena	64	0.75	30
Lena	32	0.6875	29.5
Lena	64	0.6875	29.6
Lena	32	0.625	29.1
Man	64	0.75	27.1
Man	32	0.6875	26.7
Man	64	0.6875	26.9
Man	32	0.625	26.6

Table 3: Performance of proposed technique.

Output images of the proposed technique are shown in figure 2. The proposed technique reduces significantly the memory requirements (memory size-number of memory accesses) of classical full-search vector quantization while achieving comparable output image quality. This is very important for applications requiring low-power consumption as memory related power consumption forms an important part of the total power budget [2]. So for power consumption reduction it is preferable to replace memory related computation with pure arithmetic computation [2]. This is exactly what the proposed technique does. Another important point of the proposed technique is that the transformation parameters (shifting constant, isometry) depend on the image block under encoding. Thus the small codebook is somehow adapted each time to the specific block under encoding. This is a kind of "dynamic" nature of the proposed scheme in comparison to vector quantization.The proposed encoding technique can be used in other vector quantization schemes such as tree-search vector quantization [3].

5. CONCLUSIONS

In this paper some new ideas related to vector quantization image compression were presented. A novel codebook design algorithm that exploits specific characteristics of the blocks of the training sequence was described. The results prove that the design of a representative initial codebook is very important for the performance of the codebook design algorithm, leading to both high output image quality and high design speed. The use of specific characteristics of the training sequence blocks can help significantly towards this direction.

A new technique for computational extension of small codebooks was also presented. The basic idea is that the encoding task becomes computation-based rather than memory-based which may be very important for specific applications like applications requiring low-power consumption. The quality achieved is comparable to that of classical full-search vector quantization while using smaller codebooks.

6. REFERENCES

[1] M. Barlaud, P. A. Chou, N. M. Nasrabadi, D. Neuhoff, M. J. T. Smith, J. W. Woods, "Introduction to the Special Issue on Vector Quantization", IEEE Trans. on Image Processing - Special issue on Vector Quantization, Vol.5, No.2, pp. 197-199, February 1996.
[2] A. Chandrakasan, R. Brodersen, "Low Power Digital CMOS Design", Kluwer Academic Publishers 1995.
[3] A. Gersho, R. M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers 1992.
[4] Y. H. Hu, "Optimal VLSI Architecture for Vector Quantization", proc. of the IEEE ICASSP 1995, pp. 2853-2856.
[5] A. E. Jacquin, "Image Coding Based on Fractal Theory of Iterative Contractive Image Transformations", IEEE Trans. on Image Processing, Vol.1, pp. 18-30, January 1992.
[6] Y. Linde, A. Buzo, R. M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. on Communications, Vol.28, pp. 84-95, 1980.
[7] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review", IEEE Trans. Commun., vol. 36, pp. 957-971, Aug. 1988.

Fig. 1: (Left to right) Outputs of proposed codebook design algorithm (N=256), LBG (N=256) and proposed codebook computational extension technique (N=32, 0.625 bits/pixel) for image Lena.

Fig. 2: (Left to right) Outputs of proposed codebook design algorithm (N=256), LBG (N=256) and proposed codebook computational extension technique (N=64, 0.75 bits/pixel) for image Man.