1-1-1997

Index-compressed vector quantisation based on index mapping

Jamshid Shanbehzadeh
Tarbiat Moalem University, Islamic Republic of Ira

Philip Ogunbona
University of Wollongong, philipo@uow.edu.au

Recommended Citation

Shanbehzadeh, Jamshid and Ogunbona, Philip: Index-compressed vector quantisation based on index mapping 1997, 31-38.
https://ro.uow.edu.au/infopapers/2147

Index-compressed vector quantisation based on index mapping

Abstract

The authors introduce a novel coding technique which significantly improves the performance of the traditional vector quantisation (VQ) schemes at low bit rates. High interblock correlation in natural images results in a high probability that neighbouring image blocks are mapped to small subsets of the VQ codebook, which contains highly correlated codevectors. If, instead of the whole VQ codebook, a small subset is considered for the purpose of encoding neighbouring blocks, it is possible to improve the performance of traditional VQ schemes significantly. The performance improvement obtained with the new method is about 3dB on average when compared with traditional VQ schemes at low bit rates. The method provides better performance than the JPEG coding standard at low bit rates, and gives comparable results with much less complexity than address VQ.

Keywords

quantisation, mapping, vector, index, compressed

Disciplines

Physical Sciences and Mathematics

Publication Details

Shanbehzadeh, J. & Ogunbona, P. (1997). Index-compressed vector quantisation based on index mapping. IEE Proceedings: Vision, Image and Signal Processing, 144 (1), 31-38.

Index-compressed vector quantisation based on index mapping

J. Shanbehzadeh
P.O. Ogunbona

Indexing terms: Vector quantisation, Index mapping

Abstract: The authors introduce a novel coding technique which significantly improves the performance of the traditional vector quantisation (VQ) schemes at low bit rates. High interblock correlation in natural images results in a high probability that neighbouring image blocks are mapped to small subsets of the VQ codebook, which contains highly correlated codevectors. If, instead of the whole VQ codebook, a small subset is considered for the purpose of encoding neighbouring blocks, it is possible to improve the performance of traditional VQ schemes significantly. The performance improvement obtained with the new method is about 3dB on average when compared with traditional VQ schemes at low bit rates. The method provides better performance than the JPEG coding standard at low bit rates, and gives comparable results with much less complexity than address VQ.

1 Introduction

The basic vector quantisation (VQ) [10, 14, 15] encoder first partitions the input image into small and similarly sized blocks (vectors), and then compares each vector with a set of predetermined vectors (codebook) to find the best matching vector from the codebook in terms of a similarity measure (mostly MSE). Associated with each vector in the codebook (codevector) is an index, and the encoder stores or transmits the index of the best-matching codevector instead of the image block. The basic VQ decoder reconstructs the vectors by substituting the image vectors indices with their corresponding codevectors from the codebook. VQ achieves its compression capability by representing a block of pixels with the corresponding index in the codebook. The bit rate of the basic VQ is the ratio of the bits required to represent the index of a vector to the vector dimension. The basic VQ has to represent all the possible forms of image blocks by only a small set of fixed vectors at low bit rates; thus the subjective and objective quality of decompressed images is poor [15].

Two important methods of improving the performance of VQ at low bit rates involve the use of some memoryless VQ or VQ schemes with memory. The schemes which consider interblock dependency in the encoding process are generally called VQ schemes with memory; otherwise they are referred to as memoryless VQ. Notable schemes among the class of memoryless VQ suitable for improving the performance of the basic VQ are classified VQ (CVQ) [20] and variable-rate VQ [5, 13, 21, 25]. Important VQ schemes with memory suitable for improving the performance of the basic VQ are variable-block-size VQ (VB–VQ) [3, 22, 23], finite-state VQ (FS–VQ) [1, 2, 7, 12], predictive VQ [13], image adaptive VQ [9] and index-compressed VQ (or sometimes called address VQ) [6, 16, 17].

The first group, memoryless VQ schemes, improves the coding performance of VQ by allocating more bits to active or visually important areas such as edges [20], or by assigning more bits to areas with low probability of occurrence and less bits to others [5, 13, 20, 21]. The second group, VQ schemes with memory, is based on exploiting the interblock correlation [3, 7, 13, 20, 22, 23], nonstationary codebook [9] or a variable-rate coder in combination with interblock correlation-removal schemes [13, 25].

Another approach in VQ schemes with memory is to exploit the correlation of the indices. High correlation in natural images exhibits itself among the indices generated from vector quantising the image. This index correlation has been exploited by lossless [6, 16, 17] (address VQ) or lossy [18, 19] compression schemes. Lossy index compression techniques [18, 19] are based on predicting the indices. This method may result in very unpleasant discontinuities which can easily be noticed in the visually important areas of the image such as edges [18, 19]. Lossless compression of the indices is a combination of a VQ scheme and a lossless index-compression scheme. Therefore this scheme retains the subjective quality of the encoded image. In address VQ, the indices obtained from memoryless vector quantising the image are losslessly compressed by using special codebooks (address codebook). In the form in which address VQ was proposed, it enjoyed the simplicity of memoryless VQ but suffered from the complexity involved in the index compression [8].

The index-compression scheme introduced by Feng and Nasrabadi [6] has three codebooks. The first codebook is used to quantise a mean removed version of the image blocks. The size of this codebook is 128. The indices obtained are sent into a lossless vector-quantisation scheme based on the second codebook (first-address codebook). The indices of each four neighbour-

---

IEE Proceedings online no. 19970984

Paper received 7th May 1996

The authors are with the Department of Electrical and Computer Engineering, University of Wollongong, Northfield Avenue, Wollongong, NSW Australia 2522

ing blocks are compared with a very big codebook of patterns; the number of different patterns is 16 384. If there exists a match with the indices of these four neighbouring blocks they will become candidate for another test; otherwise the index of these blocks has to be transmitted. If there exists a match, instead of four indices, one index from the second codebook can be transmitted.

In the third step, the indices of four groups of four nearest-neighbouring blocks (i.e 16 neighbouring blocks in a 4 × 4 matrix) are compared with patterns in the third codebook. If there is a match, then instead of the indices of 16 blocks, one index is sufficient. Feng and Nasrabadi used a codebook of size 1024 for this third step. A map is required to record the results of the tests performed in each step of the encoding process.

The main problem of address VQ is the high computation and memory required to generate the address codebooks for lossless index compression. Nassrabadi and Feng [17] solved the computational complexity, but their solution introduced other problems such as the synchronisation between the encoder and decoder, and the computational complexity of reordering the codebook at the receiver and transmitter during the encoding of each block [17].

The other problem of address VQ is the amount of memory required for storing the codebooks in comparison to its counterpart, mean removed VQ. Assuming 8 bits per pixel for each component of the codevectors of the basic mean removed VQ, the ratio of the codebooks for the address VQ to the basic mean removed VQ is about 36. If image quality is to be improved by increasing the codebook size, this ratio will increase considerably. In spite of all the problems listed, the high performance of address VQ, in terms of PSNR at low bit rates, makes it feasible to apply VQ at low bit rates. Using Lena as a test image, Nasrabadi and Feng [17] reported a PSNR of 30.6dB at 0.256 bits per pixel.

This paper introduces a novel VQ scheme based on lossless index compression [index-compressed VQ (IC–VQ)]. IC–VQ enjoys the simplicity of the basic VQ scheme, and only uses the codebook of the basic VQ coder. The theoretical basis of IC–VQ stems from a consequence of the high block–index correlation in natural images; neighbouring image blocks have a high probability of being mapped to highly correlated codevectors. In other words, VQ maps the neighbouring blocks onto a very small subset of the codebook containing codevectors which are relatively highly correlated. If one considers these small subsets of codebook instead of the whole codebook, it is possible to improve the performance of traditional VQ schemes.

In IC–VQ, the image blocks undergo a vector-quantisation step, and the indices generated are sent through a compression step by mapping them onto a subset of the indices. This approach leads to assignment of fewer bits to each image block than in the original VQ coder. The difference between the newly introduced index-compression schemes and address VQ is the direct use of image indices' characteristics to exploit the interblock correlation, rather than generating a global set of vectors to describe the interblock correlation as in address VQ. The index-compression method proposed in this paper is thus image adaptive. In IC–VQ the encoding of the indices requires a codebook which is a subset of the original VQ coder codebook and does not need to be generated separately as is the case in address VQ. Thus IC–VQ saves memory space by performing all the encoding process with one codebook, and requires less complexity when compared with the second and third steps of index compression in address VQ.

2 Neighbouring image blocks' index characteristics

A high correlation among the image-block indices indicates that the probability of a given block having neighbours with about the same index is high. This is because, in VQ, each image block is mapped onto the codevector which has the best similarity, in terms of a specific criterion such as MSE. Consequently, the neighbouring image blocks' indices are mapped onto similar codevectors which form a small subset of the codebook.

A simple experiment was conducted to reveal the relationship among the indices of neighbouring blocks. Consider the image-block arrangement shown in Fig. 1; the block under consideration is labelled M and the neighbours considered are the black blocks to its north and west. Two groups of information were measured; the probability of identically indexed neighbouring blocks and the frequency of occurrence of the neighbouring blocks.

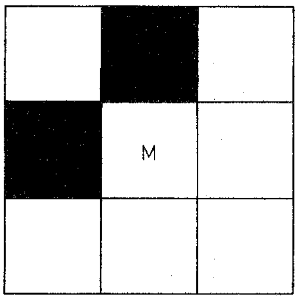

Fig. 1 *Indices of black-coloured blocks are compared with the index of block M*

Tables 1–3 show three groups of probabilities of the identically indexed neighbouring blocks for some standard images obtained from the USC database. These three tables show, respectively, the probability that a block has an identically indexed neighbour located in the north side, west side, or north or west side. These probabilities are obtained by full-search vector-quantising (FSVQ) the images. The average rate of FSVQ is from 1 to 7 bit per vector, the size of the images is 512 × 512 pixels and the block size is 4 × 4 pixels. The results show that the probability of identically indexed neighbouring blocks at low bit rates is high and that increasing the bit rate will reduce this probability.

Fig. 2 shows a plot of the histogram of frequency of the image-block indices against the index of the north or west-side neighbours. This Figure is based on testing the indices of neighbouring blocks using four standard images from the USC database, and the indices are obtained by full-search vector-quantising the images when the codevectors of the codebook are re-ordered based on their energies. The block size for this experiment is 4 × 4 pixels and the codebook size is 128. The result shown in Fig. 2 is empirical evidence that the probability of mapping neighbouring blocks onto a small subset of the VQ codebook is very high.

Table 1: Probability of having an identical index with the neighbouring block located in the north-side at various quantisation levels

	Bits per vector						
	7	6	5	4	3	2	1
Airplane	0.4286	0.4586	0.5764	0.6134	0.7062	0.8352	0.9194
Baboon	0.1083	0.1525	0.2538	0.3059	0.4756	0.6656	0.8277
Boat	0.2797	0.3457	0.4630	0.5185	0.6337	0.7644	0.8907
Couple	0.3056	0.3459	0.4836	0.5009	0.6523	0.7934	0.8834
Crowd	0.1900	0.2450	0.3694	0.4143	0.5573	0.7307	0.8890
Man	0.2401	0.3177	0.4483	0.4602	0.6312	0.7782	0.8908
Peppers	0.3855	0.4435	0.5728	0.6413	0.7400	0.8454	0.9298

Table 2: Probability of having an identical index with the neighbouring block located in the west side at various quantisation levels

	Bits per vector						
	7	6	5	4	3	2	1
Airplane	0.5171	0.5899	0.6263	0.7045	0.7850	0.8952	0.9534
Baboon	0.1081	0.1533	0.2045	0.3080	0.4768	0.6681	0.8446
Boat	0.3177	0.3802	0.4436	0.5716	0.6923	0.8180	0.9390
Couple	0.3293	0.3939	0.4514	0.5352	0.6904	0.8184	0.9068
Crowd	0.2980	0.3570	0.4357	0.5418	0.6600	0.8090	0.9332
Man	0.3078	0.3678	0.4420	0.5401	0.6778	0.8213	0.9229
Peppers	0.3647	0.4568	0.5403	0.6720	0.7866	0.8893	0.9564

Table 3: Probability of having an identical index with the neighbouring blocks located in the north or west side at various quantisation levels

	Bits per vector						
	7	6	5	4	3	2	1
Airplane	0.6084	0.6487	0.7438	0.7657	0.8478	0.9219	0.9608
Baboon	0.1752	0.2492	0.3878	0.4663	0.6713	0.8346	0.9187
Boat	0.4044	0.4900	0.6199	0.6684	0.7804	0.8836	0.9539
Couple	0.4697	0.5355	0.6790	0.7059	0.8301	0.9163	0.9563
Crowd	0.2935	0.3736	0.5191	0.5791	0.7160	0.8541	0.9478
Man	0.3550	0.4504	0.6001	0.6258	0.7802	0.8860	0.9475
Peppers	0.5155	0.5850	0.7097	0.7598	0.8389	0.9149	0.9643

3 Index-compressed VQ

The characteristics of indices shown in Section 2 can be used to reduce the coding rate by transmitting or storing a special index (probabilistic index) which describes the neighbourhood situation of a block. In the following consideration, the VQ codebook has N codevectors and the VQ rate is $r = \log_2 N$. Tables 1–3 show that the probability of having identically indexed neighbouring blocks is high, and Fig. 2 shows that the neighbouring blocks are mapped onto a small subset of the VQ codebook; consequently there is need for only a few symbols instead of N to represent the neighbourhood situation of an image block. These symbols are referred to as probabilistic indices.

The index of a given block can be classified as any of three groups: (i) indices with identically indexed west-side neighbour, (ii) indices with identically indexed north-side neighbour (iii) indices whose west-side or north-side neighbour is differently indexed. The last category is further divided into those indices whose west- or north-side neighbours either have a high or low probability of occurrence.

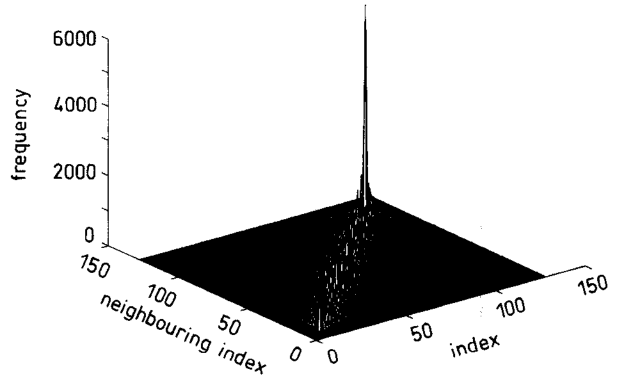

Fig. 2 *Histogram of the indices of the neighbouring blocks with a block which has a specific index*

The first step of IC–VQ exploits the feature of identically indexed neighbouring blocks. The probability that a block has neighbours with the same index is much higher than otherwise at low bit rates; this is evident from Tables 1–3. If the blocks' indices in this category are identified, one only needs to assign the probabilistic index to the rest. The index of each block is compared with its west-side neighbour (i.e. the index of block M is compared with that of the block labelled W in Fig. 3), to identify the blocks that have identically indexed west-side neighbour. If such indices are identical, a specific symbol such as '1' is assigned to them, otherwise another symbol such as '0' is assigned.

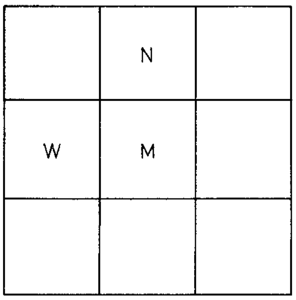

Fig. 3 *In IC–VQ, the index of two neighbours on the north and west sides (blocks N and W) is used to encode the index of block M*

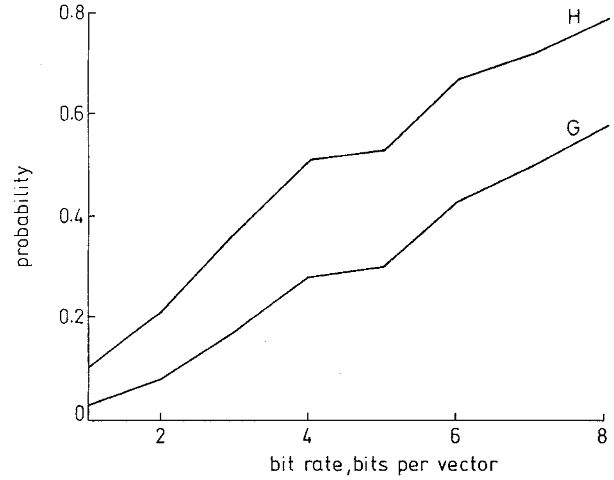

Fig. 4 P_W *and* $P_{(W\cup N)}$ *against rate*
G = probability of not having an identical index to the north- or west-side block
H = probability of not having an identical index to the west-side block

Next, the index of each of the remaining blocks (that failed the first test) is compared with that of its north-side neighbour (i.e. the index of block M is compared with the index of the block labelled N in Fig. 3). The number of blocks in this comparison is $n\bar{P}_W$, where n is the total number of image blocks and $\bar{P}_W$ is the probability that a randomly chosen image block will have an index different from that of its west-side neighbour. The line labelled H in Fig. 4 shows this probability, for a typical image (Lena), for codebooks having bit rates ranging from 1 to 8 bits per vector. As in the previous step, a '1' is assigned to a block if there is a match, and a '0' otherwise.

The number of remaining blocks is $n\bar{P}_{(W\cup N)}$, where $\bar{P}_{(W\cup N)}$ is the probability that a block does not have an index identical to that of any of its west- or north-side neighbours. This probability is plotted (line G) for rates ranging from 1 to 8 bits per vector for the image Lena in Fig. 4. The remaining blocks are divided into two groups: those for which there is a high probability of having a neighbouring block with about the same index and those with a low probability of such occurrence. The division into these two groups is performed in two steps. First, the neighbour to the west or north of the block which most similar index-wise is identified. A '1' is assigned, if the block is most similar to its west-side neighbour and '0' otherwise. Next those blocks are identified which have a neighbouring block with an index with very low probability of occurrence. It is also necessary to transmit or store the complete index for these blocks.

A probabilistic index is assigned to the remaining blocks. The rate of this section depends on the number of probabilistic indices. If the north- or west-side neighbours are used, 1 or 2 bits per vector have to be assigned for all the remaining blocks.

The coding rate of IC–VQ is given as:

$$R = 1 + \bar{P}_W + 2\bar{P}_{(W\cup N)} + rP_{low} + P_{high}\log_2 k \text{ (bpv)} \quad (1)$$

where P_{low} is the probability of having a neighbouring block whose index has a very low probability of occurrence, and r is the rate of a conventional VQ scheme. P_{high} is the probability that the west- and north-side neighbouring blocks are neither identical to the given index nor belong to the group of blocks with a low probability of occurrence, and k is the size of small codebook for neighbouring blocks. P_{low} and P_{high} depend on r and k. Figs. 5 and 6 illustrate these two probabilities for the image Lena, respectively. The expression given in eqn. 1 is derived in the Appendix.

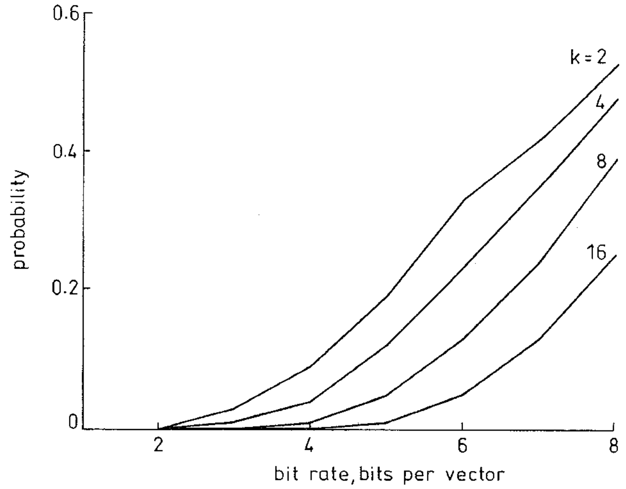

Fig. 5 *Probability that both of the neighbours of a block on the west and north sides have an index with high difference value, depending on k, with the block index*

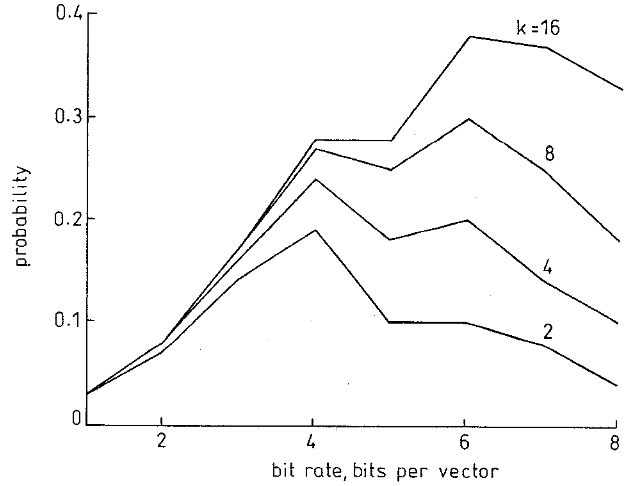

Fig. 6 *Probability that any of two neighbours of a block on the west and north sides have an index with low difference value, depending on k, with the block index*

IC–VQ considers two neighbouring blocks in the compression process, but it is possible to consider three or more neighbours. This may not necessarily improve the performance of IC–VQ, since if there are more than two neighbouring blocks there is a need to transmit or store extra overhead information to represent the neighbouring-block situation.

4 Performance analysis of IC–VQ

This section discusses the optimum situation for IC–VQ, introduces the method of improving the performance of IC–VQ and compares IC–VQ with its corresponding VQ schemes.

4.1 Minimum rate of IC–VQ

In eqn. 1, P_{low} and P_{high} depend on k and r, and the rest are constants or dependent on r. It is possible to find the optimum k (for a given image) by finding the minimum rate for a specific codebook size. Fig. 7 illustrates the rate against k for codebooks of sizes varying from 16 to 64 (4 to 6 bits per vector) and block size 4 × 4. The optimum k is about 4 for these codebook sizes. In other words, for universal codebooks with sizes varying from 16 to 64 a regional codebook with size of about four codevectors gives the minimum rate.

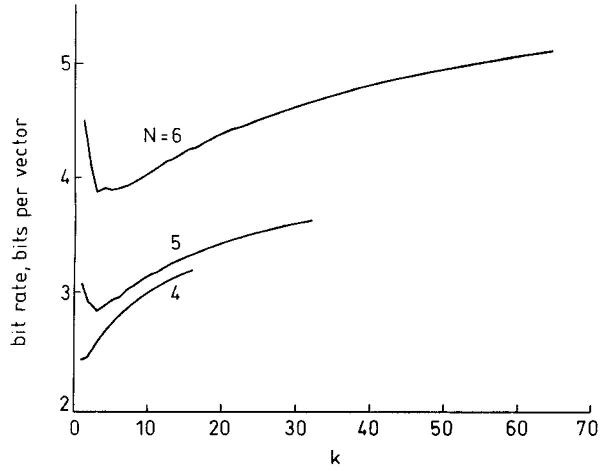

Fig. 7 *Rate against k*

Two cases, (when $k = 1$ and $k = 2^N$) are noteworthy as there is no need to transmit overhead information about the classes with high and low probabilities of occurrence. In the two extreme cases, $k = 1$ and $k = 2^N$, the rate formula can be written as

$$R = 1 + \bar{P}_W + 2\bar{P}_{(W\cup N)} + rP_{low} \text{ (bits per vector) if } k = 1 \quad (2)$$

and,

$$R = 1 + \bar{P}_W + 2\bar{P}_{(W\cup N)} + rP_{high} \text{ (bits per vector) if } k = 2^N \quad (3)$$

These two cases consist of two classes: those blocks with an identical index with their neighbour, and the rest. This consideration results, respectively, in

$$\bar{P}_{(W\cup N)} = P_{low} \quad (4)$$

and

$$\bar{P}_{(W\cup N)} = P_{high} \quad (5)$$

If the extra overhead information is removed, the rate formula can be expressed as

$$R = 1 + \bar{P}_W + r\bar{P}_{(W\cup N)} \quad (6)$$

The main difference between the rate formulae given in eqns. 1 and 6 is the requirement of overhead information for the high- and low-probability classes. Next the condition under which the rate given by eqn. 6 results in an improvement is derived. Note that, under this condition, the optimum k will be 1 or 2^N. The desired condition is easily obtained by finding the difference between the two formulae:

$$\text{Difference} = 2\bar{P}_{(W\cup N)} + rP_{low} + P_{high}\log_2 K - r\bar{P}_{(W\cup N)} \quad (7)$$

If rP_{high} is added to both sides, followed by some manipulation, eqn. 7 can be written as

$$\text{Difference} = 2\bar{P}_{(W\cup N)} + P_{high}(\log_2 K - r)r \quad (8)$$

If the difference in eqn. 8 is less than zero, the rate obtained by eqn. 6 gives a greater improvement. The required condition is

$$r - \log_2 K < \frac{2\bar{P}_{(W\cup N)}}{P_{high}} \quad (9)$$

4.2 Methods of improving the rate of IC–VQ

It is possible to reduce the bit rate by run-length coding the first three terms of eqn. 1, since all these three terms are streams of ones and zeros. Thus, the possibility of having several identical symbols in succession provides a means of improving further the performance of the new scheme. Fig. 8 illustrates the results of IC–VQ based on FSVQ with and without run-length coding. The improvement achieved by employing run-length coding is about 0.04 bit/pixel at the same PSNR.

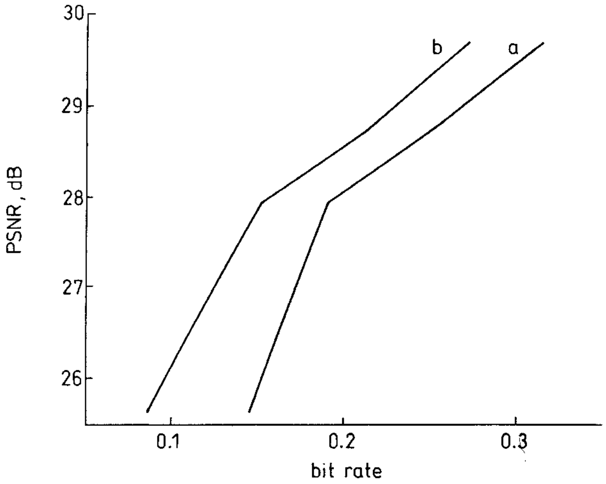

Fig. 8 *Comparing IC–VQ with and without run-length coding: Lena 512 × 512, block size 4 × 4*
a PC-FSVQ without run-length coding
b PS-FSVQ with run-length coding

5 Simulation results and discussion

The coding performance of IC–VQ based on both FSVQ and TSVQ coder is compared with traditional VQ schemes, JPEG [23] and address VQ. As IC–VQ is a combination of VQ with a lossless-coding scheme, this method has been compared with another combination of VQ with another lossless coding scheme: Huffman coding [11]. In most cases, the objective performance superiority of the new scheme is demonstrated. The PSNR has been computed by the formula

$$PSNR = 10\log_{10}\left(\frac{255^2}{mse}\right) \quad (10)$$

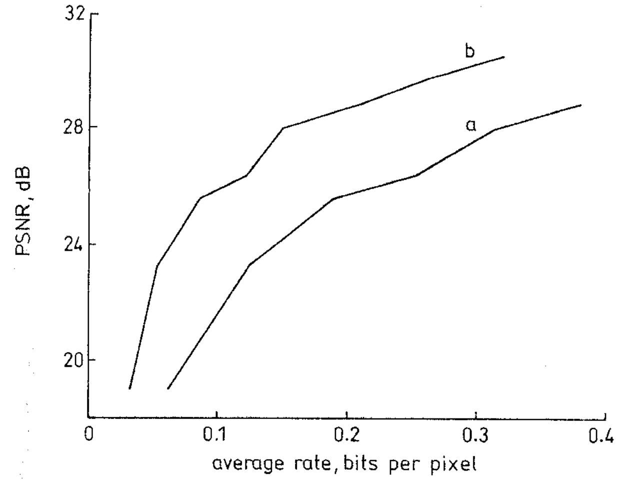

Fig. 9 *Results of FSVQ and IC–VQ based on FSVQ*
a FSVQ
b IC–VQ

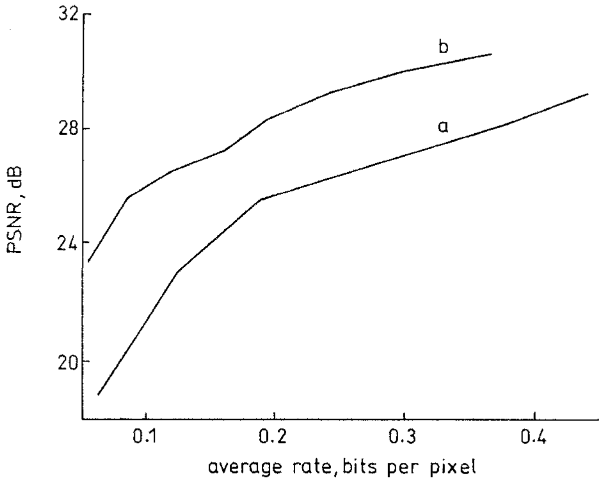

Fig. 10 *Results of TSVQ and IC–VQ based on TSVQ*
a TSVQ
b IC–VQ

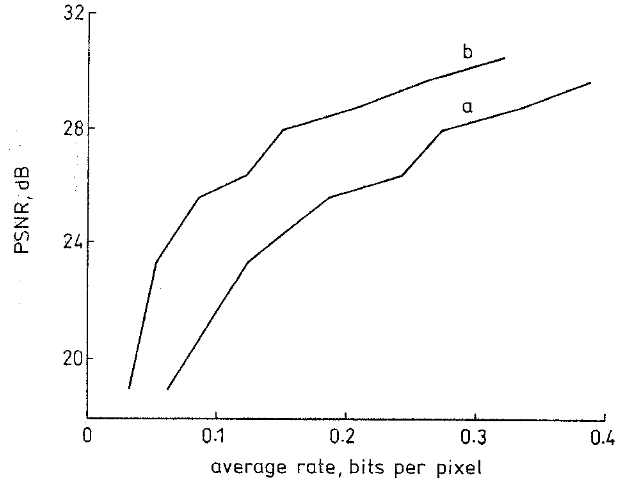

Fig. 11 *Results of FSVQ with Huffman coding of the indices and IC–VQ*
a FSVQ with Huffman coding of the indices
b IC–VQ based on FSVQ

5.1 Comparison between IC–VQ and traditional VQ schemes

Figs. 9 and 10 show the results of IC–VQ and traditional VQ schemes (FSVQ and TSVQ) at bit rates ranging from 0.05 to 0.35 bit per pixel. A similar comparison is shown in Figs. 11 and 12; in this case the indices of traditional VQ schemes are Huffman encoded. The improvement acheived by the new scheme over traditional schemes is on average more than 3dB. This improvement is more significant at bit rates from about 0.06 to 0.2 bit per pixel. Figs. 11 and 12 show about the same improvement as Figs. 9 and 10, but at bit rates of more than 0.2 bit per pixel the

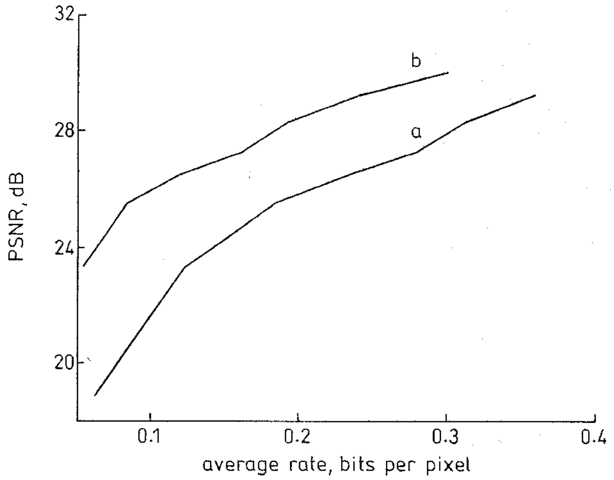

Fig. 12 *Results of TSVQ with Huffman coding of the indices and IC–VQ*
a TSVQ wuth Huffman coding of the indices
b IC–VQ based on TSVQ

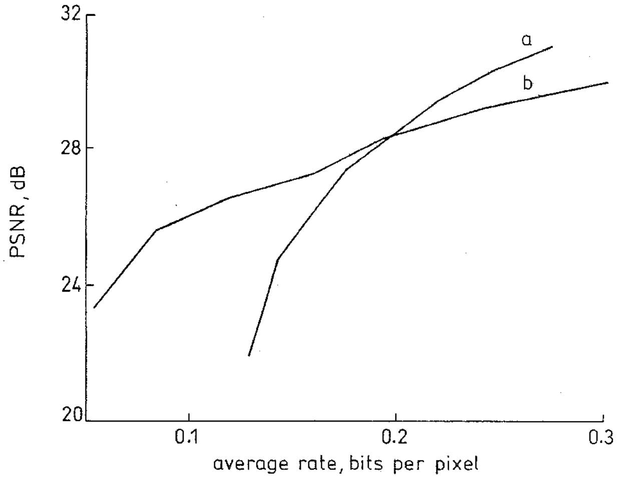

Fig. 13 *Results of JPEG and IC–VQ based on TSVQ*
a JPEG
b IC–VQ based on TSVQ

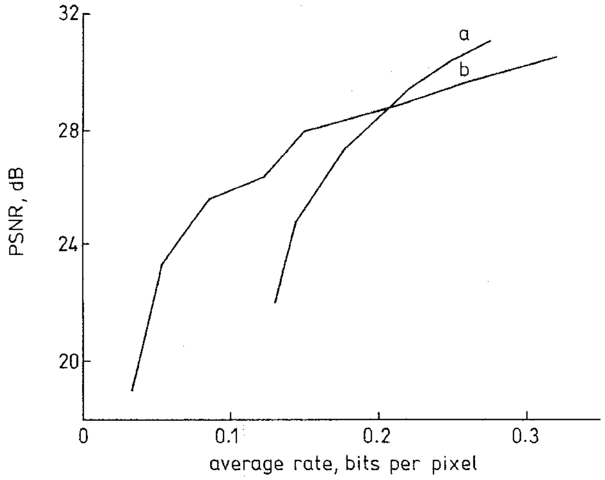

Fig. 14 *Results of JPEG and IC–VQ based on FSVQ*
a JPEG
b IC–VQ

difference is less than the previous case. The reason adduced for the better performance of IC–VQ over a combination of traditional VQ schemes (with Huffman coding of the indices) is that both traditional VQ and Huffman coding disregard the index dependency.

5.2 Comparison between IC–VQ and JPEG

Figs. 13 and 14 show the results of JPEG (line A) [24] and IC–VQ based on TSVQ and FSVQ (line B), respectively. The performance of IC–VQ based on TSVQ and FSVQ outperforms JPEG at bit rates less than 0.19 bit per pixel and 0.22 bit per pixel, respectively. The performance difference at around 0.15 bit per pixel is about 2 – 3dB. The reasons for the superiority of IC–VQ are as follows. At low bit rates IC–VQ gains most of its performance improvement through exploiting the interblock correlation. The quantisation procedure of JPEG adapts the transform-domain coefficients based on the available low number of bits. A consequence is the similarity of corresponding AC coefficients in neighbouring blocks. This similarity is a source of the quantisation-domain interblock correlation exploited by IC – VQ, but neglected by JPEG. The down side of this is that IC–VQ is more sensitive to channel noise than JPEG.

5.3 Comparison between IC–VQ and address VQ

A simple comparison can show the superiority of this scheme over address VQ in terms of implementation. This scheme consists of a simple memoryless VQ encoder and a simple method of mapping neighbouring blocks onto subsets of the VQ codebook based on comparing the indices. These requirements are much less than the complexity of address VQ, which was previously described. Another important advantage of this system is its flexibility to operate on a wide range of compression ratios using a small amount of space for the memoryless-VQ codebook. While for address VQ any change of the compression ratio requires not only some memory space for the memoryless-VQ codebook but also a large amount of memory space for the address codebook, and if the time required to generate the address codebook is considered the advantage of IC–VQ over address VQ becomes obvious. The simulation results show about 0.7dB difference in PSNR at about the same rate (0.26 bit per pixel).

6 Conclusion

A novel coding scheme, IC–VQ, which is capable of providing significant results at low bit rates, is proposed. IC–VQ exploits the interblock correlation by applying a lossless coding scheme on the indices of a vector-quantised image to achieve performance improvement. High interblock correlation in natural images has a direct influence on the probability of the indices of neighbouring blocks, in a way that their indices belong to a small subset of the codebook. This characteristic has been employed in a lossless scheme to improve further the performance of traditional simple VQ schemes. The coding performance of this scheme has been compared with that of traditional VQ schemes, address VQ, and JPEG at low bit rates. This scheme outperforms these coding schemes in terms of objective performance or implementation.

7 References

1 ARAVIND, R., and GERSHO, A.: 'Low-rate image coding with finite-state vector quantization', *ICASSP*, 1986, pp. 137–140
2 ARAVIND, R., and GERSHO, A.: 'Image compression based on vector quantization with finite state memory', *Opt. Eng.*, 1987, **26**, pp. 570–580
3 BOXERMAN, J.L., and LEE, H.J.: 'Variable block-sized vector quantization of grayscale images with unconstrained tiling', *ICASSP*, 1990, pp. 2277–2280
4 BUZO, A., GRAY, A.H., GRAY, R.M., and MARKEL, J.D.: 'Speech coding based upon vector quantization', *IEEE Trans.*, 1980, **ASSP-28**, pp. 562–574
5 CHOU, P.A., LOOKABAUGH, T., and GRAY, R.M.: 'Entropy-constrained vector quantization', *IEEE Trans.*, 1989, **ASSP-37**, pp. 31–42
6 FENG, Y., and NASRABADI, N.M.: 'A dynamic address-vector quantization algorithm based on interblock and intercolor correlation for color image coding', *ICASSP*, 1988, pp. 1755–1758
7 FOSTER, J., GRAY, R.M., and DUNHAM, M.O.: 'Finite-state vector quantization for waveform coding', *IEEE Trans.*, 1985, **IT-31**, pp. 348–359
8 GERSHO, A., and GRAY, R.M.: 'Vector quantization and signal compression' (Kluwer Academic Publishers, Massachusetts, 1992)
9 GOLBERG, M., BOUTCHER, P.R., and SHLIEN, S.: 'Image compression using adaptive vector quantisation', *IEEE Trans.*, 1986, **COM-34**, pp. 180–187
10 GRAY, R.M.: 'Vector quantization', *IEEE ASSP Mag.*, 1984, pp. 4–29
11 HUFFMAN, D.A.: 'A method for the construction of minimum redundancy codes', *Proc. Inst. Radio Eng.*, 1952, **40**, pp. 1098–1101
12 KIM, T.: 'New finite state vector quantizers for images', *ICASSP*, 1988, pp. 1180–1183
13 LOOKABAUGH, T., RISKIN, V, CHOU, P.A., and GRAY, R.M.: 'Variable rate vector quantization for speech, image, and video compression', *IEEE Trans.*, 1993, **COM-41**, (1)
14 MAKHOUL, J., ROUCOS, S., and GISH, H.: 'Vector quantization in speech coding', *Proc. IEEE*, 1985, **73**, pp. 1551–1588
15 NASRABADI, N.M., and KING, R.A.: 'Image coding using vector quantization: A review', *IEEE Trans.*, 1988, **COM-35**, pp. 957–971
16 NASRABADI, N.M., and FENG, Y.: 'A multilayer address vector quantization technique', *IEEE Trans.*, 1990, **CAS-37**, (7), pp. 912–921
17 NASRABADI, N.M., and FENG, Y.: 'Image compression using address-vector quantisation', *IEEE Trans.*, 1990, **COM-38**, pp. 2166–2173
18 POGGI, G.: 'Address-predictive quantization of images by topology-preserving codebook ordering', *Eur. Trans. Telecomun. Relat. Technol.*, 1993, **4**, pp. 423–434
19 POGGI, G.: 'Generalized-cost-measure-based address-predictive Vector quantization', *IEEE Trans. Image Process.*, 1996, **5**, (1), pp. 49–55
20 RAMAMURTHI, B., and GERSHO, A.: 'Classified vector quantization of images', *IEEE Trans.*, 1986, **COM-34**, pp. 1105–1115
21 RISKIN, E.A., and GRAY, R.M.: 'A greedy tree growing algorithm for the design of variable rate vector quantizers', *IEEE Trans.*, 1991, **ASSP-39**, (11) pp. 2500–2507
22 VAISEY, J., and GERSHO, A.: 'Variable block-size image coding', *ICASSP*, 1988, pp. 1051–1054
23 VAISEY, J., and GERSHO, A.: 'Image compression with variable block size segmentation', *IEEE Trans.*, 1992, **ASSP-40**, (8), pp. 2040–2060
24 WALLACE, G.K.: 'The JPEG still picture compression standard', *IEEE Trans.*, 1992, **CE-38**, (1), pp. 18–34
25 RISKIN, E.A., LOOKABAUGH, T., CHOU, P.A., and GRAY, R.M.: 'Variable rate vector quantization for medical image compression,', *IEEE Trans. Med. Imaging*, 1990, **9**, pp. 290–298

8 Appendix: Rate of index compressed VQ

A description of the IC–VQ algorithm is given in Section III, and the rate is derived in this Appendix. The rate for the first step of IC–VQ is 1 bit per vector. The rate for the second step of IC–VQ is $\bar{P}_W$ bit per vector since, in the second step, the index of each of the remaining blocks (which failed the first test) is compared with that of its north-side neighbour, and the information regarding the results of the comparison has to be transmitted. This information requires 1 bit per vector, and as there are $n\bar{P}_W$ blocks, $\bar{P}_W$ bit per vector is the required rate. The rate for the third step is

$\bar{P}_{(W \cup N)}$ bit per vector, because the number of remaining blocks is $n\bar{P}_{(W \cup N)}$, and 1 bit per vector is required to identify the blocks which have a neighbouring block with an index with very low probability of occurrence. As it is also necessary to transmit or store the complete index for these blocks, the total rate of this case becomes $(\bar{P}_{(W \cup N)} + rP_{low})$ bit per vector, where P_{low} is the probability of having a neighbouring block whose index has a very low probability of occurrence, and r is the rate of a conventional VQ scheme. The rate to identify the index of blocks with high probability of occurrence is $P_{high}\log_2 k$ bit per vector, where P_{high} is the probability that the west- and north-side neighbouring blocks are neither identical to the given index nor do they belong to the group of blocks with a low probability of occurrence, and k is the size of the small codebook for neighbouring blocks. The total rate of this algorithm is hence

$$R = 1 + \bar{P}_W + 2\bar{P}_{(W \cup N)} + rP_{low} + P_{high}\log_2 k \,(\text{bpv})$$