The proposed index mapping can also be applied to many hybrid VQ systems such as mean-predictive VQ (MPVQ) and hierarchical VQ (HVQ) [1]. The index mapping process is similar to that in ordinary VQ except that the cost in eqn. 2 is modified as

$$C_{k,l}(j) = d_u[\beta(i_{k-1,l}) + (\hat{m}_{k-1,l} - \hat{m}_{k,l})\mathbf{1}, \beta(j)] + d_l[\beta(i_{k,l-1}) + (\hat{m}_{k,l-1} - \hat{m}_{k,l})\mathbf{1}, \beta(j)] \quad j \in I \quad (3)$$

where $\hat{m}_{k,l}$ is the mean of the decoded (k,l)th block and $\mathbf{1}$ is a vector of length M with each element equal to 1. In MPVQ, the block mean was predicted from four previously decoded blocks and a codebook with size 128 was used to quantise the residual vector. In HVQ, 2×2 block-means were first vector quantised with a mean codebook of size 64, and then each mean-removed block (of size 4×4) was vector quantised with a residual codebook of size 128. Our index mapping scheme was used to encode the residual codevectors in MPVQ, and both the residual codevectors and mean codevectors in HVQ. The simulation results are shown in Tables 2 and 3.

Table 2: Bit rate reduction performance in MPVQ

Image	Original	Proposed	Gain [%]
Lena	0.3199	0.2670	16.5
Bank	0.3424	0.2672	21.9

Table 3: Bit-rate reduction performance in HVQ

Image	Original			Proposed			Gain [%]
	b_m	b_r	b_t	b_m	b_r	b_t	
Lena	0.0839	0.3350	0.4189	0.0515	0.2713	0.3229	22.9
Bank	0.0877	0.3525	0.4402	0.0554	0.2735	0.3290	25.3

(b_m: bit rate for mean vectors, b_r: bit rate for residual vectors, b_t: total bit rate)

Conclusion: We proposed a simple dynamic index mapping which considerably reduces the index entropy by expoiting interblock correlation. It was shown that it provides a considerable amount of bit-rate reduction not only in ordinary VQ but also in many hybrid VQ systems. Since the proposed index encoding scheme requires only a small amount of additional computation and does not change the quantiser structure, it is expected to be easily applied to other VQ systems.

© IEE 1995
12 June 1995

Electronics Letters Online No: 19951030

Seung Jun Lee, Kyeong Ho Yang, Chul Woo Kim and Choong Woong Lee (*Department of Electronics Engineering, Seoul National University, Korea*)

References

1 GERSHO, A., and GRAY, R.M.: 'Vector quantisation and signal compression' (Kluwer Academic Publisher, 1992)

2 NASRABADI, N.M., and FENG, Y.: 'Image compression using address-vector quantization', *IEEE Trans.*, 1990, **COM-38**, (12), pp. 2166–2173

3 FIORAVANTI, R., FIORAVANTI, S., and GIUSTO, D.D.: 'An efficient neural prediction for vector quantization'. Proc. ICASSP, Adelaide, Australia, 1994, pp. 613–616

4 KIM, T.: 'Side match and overlap match vector quantizers for images', *IEEE Trans.*, 1992, **IP-1**, (2), pp. 170–185

Fast LBG codebook generator for BTC image compression

Kuang-Shyr Wu and Ja-Chen Lin

Indexing terms: Image processing, Data compression, Vector quantisation, Block codes

When the high-mean and low-mean generated by the BTC image compression technique are to be quantised using VQ, the computation time required to search for the nearest centroid can be reduced significantly using the proposed method. Experiments comparing the full-search and EHPM algorithms demonstrate this.

Introduction: The image compression technique BTC uses two values a and b, called the high-mean and low-mean, to replace all values in a block. To increase the compression ratio, the high/low mean pairs are often quantised further through VQ [1, 2]. We concentrate here on the task of LBG [3] codebook generation for the (high/low) mean pairs. A typical LBG cycle contains the steps:

(i) for each data point $\mathbf{q}$, assign $\mathbf{q}$ to the jth cluster if $\|\mathbf{c}_j - \mathbf{q}\| = \min_{1 \le i \le N} \|\mathbf{c}_i - \mathbf{q}\|$

(ii) replace $\{\mathbf{c}_i\}_{i=1}^N$ by the new centroids of the N clusters just formed.

Here, $\mathbf{C} = \{\mathbf{c}_i = (xc_i, yc_i)\}_{i=1}^N$ is the codebook in question. We present the new method for reducing the time needed in step (i). First, evaluate to obtain $\bar{C} = \{\bar{c}_i = (xc_i + yc_i)/2\}_{i=1}^N$ and $C = \{\tilde{c}_i = (xc_i + yc_i)/2\}_{i=1}^N$, which are the sum-projection and subtraction-projection of the codebook to the straight lines $y = x$ and $y = -x$, respectively. Assume that the $\{\bar{c}_i\}_{i=1}^N$ has been sorted. For each data point $\mathbf{q}$ we proceed as follows, to obtain the nearest codeword ($\|\cdot\|$ always denotes a 2-D Euclidean norm $\sqrt{[()^2+()^2]}$, whereas $|\cdot|$ denotes the 1-D absolute value):

Algorithm:

Step 1: Calculate $\bar{q} = (xq+yq)/2$ and $\tilde{q} = (xq-yq)/2$. Use one-dimensional binary searching to search for the $\bar{c}_k$ that satisfies $|\bar{c}_k - \bar{q}| = \min_{1 \le i \le N} |\bar{c}_i - \bar{q}|$. The corresponding $\mathbf{c}_k$ is then used as an initial guess for the current nearest codeword (CNC) of $\mathbf{q}$.

Step 2: Calculate $d_{min} = \|\mathbf{c}_k - \mathbf{q}\|$. Construct the remaining set (RS) by collecting those $\mathbf{c}_i \in \mathbf{C}$ whose $\bar{c}_i$ satisfy

$$|\bar{q} - \bar{c}_i| < (d_{min}/\sqrt{2}) \quad (1)$$

Step 3: Delete $\mathbf{c}_k$ from RS, because $\mathbf{c}_k$ has been checked.

Step 4: If RS is empty, then return CNC as the nearest codeword of $\mathbf{q}$ and exit.

Step 5: In the RS, obtain the $\mathbf{c}_k$ whose $\bar{c}_k$ satisfies

$$|\bar{c}_k - \bar{q}| = \min_{\mathbf{c}_i \in \mathrm{RS}} |\bar{c}_i - \bar{q}| \quad (2)$$

(Unlike in step 1, there is no need to perform a binary search here. Since $\{\bar{c}_i\}_{i=1}^N$ has been sorted, $\{\bar{q}\} \cup \{\bar{c}_i\}_{i=1}^N$ has been sorted after step 1; $\{\bar{q}\} \cup \{\bar{c}_i | \mathbf{c}_i \in \mathrm{RS}\}$ has been sorted after step 2. Therefore, one of the two neighbours of the previous $\bar{c}_k$ must be the candidate.)

Step 6: If $\mathbf{c}_k$ violates any of the following three inequalities:

$$|\tilde{q} - \tilde{c}_k| < (d_{min}/\sqrt{2}) \quad (3)$$

$$|xq - xc_k| < d_{min} \quad (4)$$

$$|yq - yc_k| < d_{min} \quad (5)$$

then go to step 3, else calculate $d = \|\mathbf{c}_k - \mathbf{q}\|$.

Step 7: If $d < d_{min}$ then update d_{min} and CNC with d and $\mathbf{c}_k$, respectively. (Also use this new d_{min} to delete from the RS the $\mathbf{c}_i$ whose $\bar{c}$ violates eqn. 1.)

Step 8: Go to step 3.

The $\mathbf{c}_k$ obtained in step 1 is deleted in step 3, and hence the $\mathbf{c}_k$ obtained in step 5 is different. Furthermore, after the deletion of the codewords $\mathbf{c}_i$ in step 7, the $\bar{c}_i$ of every codeword $\mathbf{c}_i$ remaining in the RS must satisfy $|\bar{q} - \bar{c}_i| < (d_{min}/\sqrt{2})$. Since the next $\mathbf{c}_k$ obtained in the next execution of step 5 is also chosen from the RS, then $\mathbf{c}_k$ must also satisfy

$$|\bar{q} - \bar{c}_k| < (d_{min}/\sqrt{2}) \qquad (6)$$

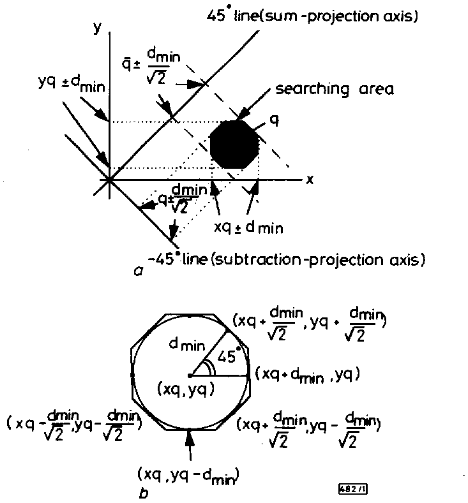

Fig. 1 *Searching area for nearest codeword and detail of shaded portion*

a Searching area for nearest codeword
b Detail of shaded portion in *a*

Eqns. 3 – 6 together imply that the $\mathbf{c}_k$ that could be a CNC must fall in the shaded area enclosed by the eight-sided regular polygon shown in Fig. 1*a*. The reason there is a $\sqrt{2}$ term in eqns. 3 and 6 is shown in Fig. 1*b*, which is an enlarged picture of the shaded region sketched in Fig. 1*a*. The circle has a radius d_{min} and touches the upright eight-sided regular polygon eight times. The sum-projection of the two touching points

$$\left(xq+\frac{d_{min}}{\sqrt{2}}, yq+\frac{d_{min}}{\sqrt{2}}\right) \text{ and } \left(xq-\frac{d_{min}}{\sqrt{2}}, yq-\frac{d_{min}}{\sqrt{2}}\right)$$

are

$$\left(xq+\frac{d_{min}}{\sqrt{2}}+yq+\frac{d_{min}}{\sqrt{2}}\right)/2=\frac{xq+yq}{2}+\frac{d_{min}}{\sqrt{2}}=\bar{q}+\frac{d_{min}}{\sqrt{2}}$$

and

$$\left(xq-\frac{d_{min}}{\sqrt{2}}+yq-\frac{d_{min}}{\sqrt{2}}\right)/2=\frac{xq+yq}{2}-\frac{d_{min}}{\sqrt{2}}=\bar{q}-\frac{d_{min}}{\sqrt{2}}$$

respectively. In the above algorithm, d_{min} becomes smaller and smaller, and only those $\mathbf{c}_k$s satisfying eqns. 3 –6 simultaneously will need to evaluate the 2-D distance $\|\bar{\mathbf{c}}_k - \mathbf{q}\|$ (the only exception is the $\mathbf{c}_k$ found in step 1 where eqns. 3 – 6 are not checked at all because d_{min} is not yet defined). The idea used in the algorithm is quite obvious. First, if the current nearest codeword is $\mathbf{c}_k$ and $d_{min} = \|\mathbf{c}_k - \mathbf{q}\|$, then only those $\mathbf{c}_j$ with the property $\|\mathbf{c}_j - \mathbf{q}\| \le d_{min}$ are possible to be the nearest codeword of $\mathbf{q}$. In other words, only those $\mathbf{c}_j$ interior to the circle centred at $\mathbf{q}$ and with a radius d_{min} (see Fig. 1*b*) are possible. Secondly, the circular disc is interior to the eight-sided-polygon constrained by eqns. 3 – 6 (see Fig. 1). Hence, only those $\mathbf{c}_j$ interior to the polygon are possible. The circular disc is a more accurate estimation (of the codeword) than the polygon. However, use of the circular disc is very time-consuming (many vector-distances are evaluated).

Experimental results: We compared the proposed algorithm with the full-search and an elegant algorithm EHPM [4] which also reduces the number of vector-distance ($\|\cdot\|$) computations. Table 1 shows the total number of vector-distance computations (tnovdc) required to make the LBG converge. Since there is some overhead (eqns. 1 – 5) for reducing the number of $\|\cdot\|$ computations in our method, to be fair to the other two methods we also list in Table 1 the total CPU time needed in LBG (including both steps 1 and 2). Regardless which of the two criteria is used to compare the performance, the proposed method shows better results than the full-search and EHPM algorithms. All three methods obtain the same (final) codebook and require the same number of LBG cycles. The only difference is the CPU time.

Table 1: Comparison of total number of vector-distance computations (tnovdc) and CPU time

Codebook sizes	Methods and performance		Tested images			
			Lena	Baboon	Couple	Crowd
256	FS	tnovdc	41943040	16777216	37748736	5452952
		time(s)	2091	2180	2277	2647
	EHPM	tnovdc	4091906	8244463	6194767	3606481
		time(s)	54	91	120	51
	ours	tnovdc	865928	1270239	651093	1059592
		time(s)	38	53	51	38
512	FS	tnovdc	83886080	58720256	75497472	109051904
		time(s)	3386	3151	3590	3362
	EHPM	tnovdc	4424626	6170464	9654954	3943527
		time(s)	61	74	125	56
	ours	tnovdc	832203	954876	1665988	924708
		time(s)	40	40	75	38

Machine used is SUN-SPARC 10. FS: 'full search'

Conclusions: The proposed fast nearest codeword searching algorithm accelerated the vector quantisation of the high/low means generated in BTC compression. The method condensed the searching area, and reduced the total CPU time required for the LBG.

Acknowledgment: This study was supported by the National Science Council, Republic of China, under contract NSC83-0404-E009-081.

© IEE 1995
25 May 1995
Electronics Letters Online No: 19950978

Kuang-Shyr Wu and Ja-Chen Lin (*Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China*)

Ja-Chen Lin: corresponding author

References

1 UDPIKAR, V.R., and RAINA, J.P.: 'BTC image coding using vector quantization', *IEEE Trans.*, 1987, **COM-35**, (3), pp. 352–356
2 WEITZMAN, A., and MITCHELL, H.B.: 'An adaptive BTC-VQ image compression algorithm operating at a fixed bit-rate', *Signal Process. Image Commun.*, 1993, **5**, pp. 287–294
3 LINDE, Y., BUZO, A., and GRAY, R.M.: 'An algorithm for vector quantizer design', *IEEE Trans.*, 1980, **COM-28**, (1), pp. 84–95
4 GUAN, L., and KAMEL, M.: 'Equal-average hyperplane partitioning method for vector quantization of image data', *Pattern Recognit. Lett.*, 1992, **13**, pp. 693–699

Multicomponent heterodyne laser Doppler anemometer using chirp-modulated Nd:YAG ring laser and fibre delay lines

J.W. Czarske and H. Müller

Indexing terms: Doppler velocimetry, Anenometers, Ring lasers, Solid lasers, Optical delay lines

A two-dimensional directional laser Doppler anemometer using a frequency modulated Nd:YAG laser is presented for the first time. The magnitude and sign of the fluid velocity were determined by employing the heterodyne technique without having to use an external frequency shifter.