If we analyse the output of the transform ($Y_1 \cdot R$) we see that it includes the N possible product terms which can be formed with n bit numbers. Consequently, using the trigonometric relations expressed in eqns. 3 and 4 and in a more general way in the algorithm of Fig. 2, we can deduce the kernel of eqn. 2, therefore completing the transform of X. That is, if we apply the algorithm of Fig. 2, which will coincide with the transform Y_2, to the output of $Y_1 \cdot R$ we obtain the complete cosine transform, $X = Y_1 \cdot RY_2$. Fig. 4 shows the complete FCT algorithm, where the function reverse ( ) is the bit reversal permutation of a sequence with a length of N. Fig. 5 shows the data flow for the same example as that of Fig. 1. Finally, the inverse fast cosine transform (IFCT) is obtained in a similar way to the algorithm for the FCT. We only have to change the transforms applying a time decimation algorithm.

```
fct_dif (n,x)
int n; double x[];
  {
        int i, j, k, m, s, t, N;
        double a, dcos, cos(double k);

        N = 1 < < n;
        for (i = 1; i < = n; i + +) {
                s = 1 < < (i-1);   t = 1 < < i;
                for (k = 0; k < N/t; k + +) {
                        dcos = 2*cos(3.14159*(4*k + 1)*s/(2*N));
                        for (j = 0; j < s; j + +) {
                                m = k + j*N/s;   a = x[m];
                                x[m] + = x[m + N/t];
                                x[m + N/t] = (a-x[m + N/t])*dcos;
                        }
                }
        }
        reverse (n,x);      /* bit reversal permutation */
        cosine (n,x);       /* see Fig. 2 */
  }
```

Fig. 4 *FCT DIF algorithm*

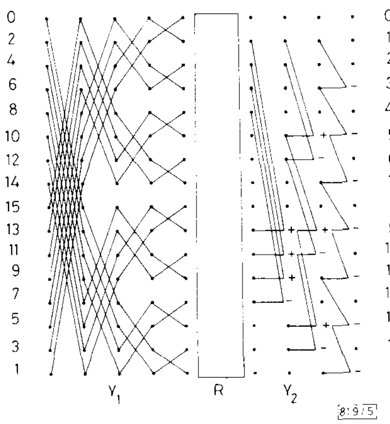

Fig. 5 *Data flow of the FCT DIF algorithm*
$N = 16$

Conclusions: We have presented a new in place, radix 2 and frequency decimation algorithm for the calculation of the fast cosine transform (FCT DIF) of a real sequence of length $N = 2^n$, whose most important properties, compared to Hou's algorithm, are the following: it has the same number of multiplications and of additions/subtractions; it does not require shift operations, as the cosine coefficients have been scaled previously; implements only one bit reversal permutation (instead of the $2n - 1$ permutations of Fig. 1); and, therefore, has a high regularity, which facilitates its implementation in VLSI technology.

Acknowledgment: This work was supported by the Ministry of Education and Science (CICYT) of Spain under contracts TIC88-0094, MIC88-0549 and Xunta de Galicia XUGA80406488.

F. ARGÜELLO
E. L. ZAPATA

Department of Electrónica, Facultad de Fisica
University of Santiago de Compostela
15706 Santiago de Compostela, Spain

19th July 1990

References

1 AHMED, N., NATARAJAN, T., and RAO, K. R.: 'Discrete cosine transform', *IEEE Trans.*, 1974, **C-23**, (1), pp. 90–94
2 HOU, H. S.: 'A fast recursive algorithm for computing the discrete cosine transform', *IEEE Trans.*, 1987, **ASSP-35**, (10), pp. 1455–1461
3 ARTIERI, A., KRITTER, S., JUTAND, F., and DEMASSIEUX, N.: 'A one chip VLSI real time two dimensional discrete cosine transform'. IEEE Int. Symp. on Circuits and Systems, 1988, pp. 701–704
4 SUN, M. T., CHEN, T. C., and GOTTLIEB, A. M.: 'VLSI implementation of a 16 × 16 discrete cosine transform', *IEEE Trans.*, 1989, **CAS-36**, (4), pp. 610–617
5 BAILEY, D. J., and BIRCH, N.: 'Image compression using a discrete cosine transform image processor', *Electronic Engineering*, 1989, **61**, (7), pp. 39–44
6 LIEBSCH, W.: 'Parallel architecture for VLSI implementation of a 2-dimensional discrete cosine transform'. 3th Int. Conf. on Image Processing and its Applications, **307**, 1989, pp. 609–612
7 DUHAMEL, P., GUILLEMOT, C., and CARLACH, J. C.: 'A DCT chip based on a new structured and computationally efficient DCT algorithm'. IEEE Int. Symp. on Circuits and Systems, 1990, pp. 77–79
8 CHO, N. I., and LEE, S. U.: 'DCT algorithms for VLSI parallel implementations', *IEEE Trans.*, 1990, **ASSP-38**, (1), pp. 121–127
9 COOLEY, J. W., and TUKEY, J. W.: 'An algorithm for the machine calculation of complex Fourier series', *Math. Comput.*, 1965, **19**, (4), pp. 297–301

FAST ENCODING ALGORITHM FOR VQ-BASED IMAGE CODING

Indexing terms: Codes and coding, Image processing

The encoding of a VQ-based image coding requires a full codebook search for each input vector to find out the best matched codeword. It is a time consuming process. A fast algorithm for vector quantising image data is proposed. The algorithm is proved powerful.

Introduction: Vector quantisation (VQ) is a widely used technique in many data compression applications. This is because a simple look-up table decoding can be used. The encoding process is a computationally intensive procedure. This limits the applicability of VQ in practical considerations. Many fast algorithms[1–3] have been proposed for reducing the computational complexity of the full search encoding. Chen and Pan[2] proposed a fast search algorithm for vector quantising speech signals. By taking advantage of the high correlation between feature vectors of successive frames, a triangular inequality elimination (TIE) method is employed to prune matching computations of wildly mismatched codewords. A similar idea is employed for vector quantising image signals in this letter. The property of high correlation between contiguous subimages can be incorporated into the TIE method for pruning unnecessary matching computations of encoding. High correlation between contiguous subimages mainly results from the smoothness in low-detail regions as well as the continuity of edges in high-detail regions for image signals. By considering both effects, matching with previously encoded codewords of four neighbouring subimages are first calculated and then used in TIE for eliminating unnecessary matching computations. This is the basic idea of this letter.

Fast encoding algorithm: A fast encoding algorithm for VQ based image coding is presented

Step 1: Partition the image into $N \times N$ subimages denoted by $X(i, j)$, for $i = 1, \ldots, M$; $j = 1, \ldots, M$. The upper leftmost subimage is labelled $X(1, 1)$.

Step 2: Encode $X(1, 1)$ by using conventional full codebook search and denote the encoded codeword as $C(1, 1)$.

Step 3: For encoding $X(i, j)$, the following operations are taken:

(*a*) Calculate the distortions of matching $X(i, j)$ with $C(i-1, j-1)$, $C(i, j-1)$, $C(i+1, j-1)$, and $C(i-1, j)$. Find the codeword with minimum distortion.

(*b*) Use the current minimum distortion and detected codeword in the TIE (to be discussed later) to eliminate matching computations of wildly mismatched codewords.

(*c*) Calculate all the remaining codeword matchings and find the encoded codeword with minimum distortion by the well-known partial distortion elimination (PDE) method.[3]

Step 3 is performed from left to right and from top to bottom. It is also noted that, in Step 3*a*, $C(i-1, j-1)$, $C(i, j-1)$, and $C(i+1, j-1)$ are not used for the encoding of subimages on the first row; and $C(i-1, j-1)$ and $C(i-1, j)$ are not used for subimages on the first column.

We now discuss the TIE method used in Step 3*b*. TIE is a popular codeword matching elimination rule in vector quantisation.[1,2] The form of TIE used in our algorithm is

Let $D(X, C_1)$ be the distance between the encoding vector X and the codeword C_1. If $D(C_1, C_2) > 2D(X, C_1)$, then eliminate the computation of $D(X, C_2)$ because it is always greater than $D(X, C_1)$. When the mean squared error distortion measure is used to replace the distance measure, the condition changes to $D(C_1, C_2) > 4D(X, C_1)$.

The TIE rule will be very efficient if we can initially find a codeword which has small distortion to the encoding vector. This is because many distortion computations can therefore be eliminated. In image coding, because of the properties of smoothness in low-detail regions and continuity of edges in high-detail regions, the best of the previously encoded codewords of four neighbouring subimages in four directions, 45°, 90°, 135°, and 180°, can be taken as the codeword to be applied in TIE for efficiently pruning computations of unnecessary codeword matchings.

To realise the TIE method, a table showing the distortion between any pair of codewords is required. Every row of this table must be stored in increasing order so the TIE elimination can be efficiently applied. This can be accomplished off-line without affecting the efficiency of the encoding process. Extra memory space to store such a table is still required. This is the major price to be paid for applying the TIE method.

Table 1 ELIMINATION EFFICIENCIES FOR ENCODING 'LENA' IMAGE

Block size	TIE	PDE	TIE + PDE
2 × 2	94·85%	60·89%	97·99%
4 × 4	87·13%	68·97%	96·01%

Simulation: The efficiency of the proposed fast encoding algorithm for VQ based image coding is examined by simulation. A 512 × 512 'Lena' image was used in the simulation. The image is firstly divided into 2 × 2 subimages. A codebook containing 1024 codewords is then generated from these vectors using the well known LBG algorithm.[4] The effectiveness of applying the TIE and PDE methods in the fast encoding algorithm is tested using this codebook. The elimination efficiencies of applying TIE and PDE are listed in Table 1. It can be seen from the Table that 94·85% codeword matchings are eliminated by TIE and 60·89% operations in the remaining codeword matchings are saved by PDE. By examining the distribution of the number of computed codeword matching for encoding a subimage, we find that less than 32 codeword matchings were performed for most subimages. The average number of operations used to encode a vector by the conventional full search and by the fast encoding algorithm are shown in Table 2. Compared with the conventional method, only 2·01% of both multiplications and additions and 13·02% of comparisons are required by the fast encoding algorithm. This shows that the proposed algorithm is very efficient. 4 × 4 subimages were finally considered. A codebook with 256 codewords was generated. Simulation results are also shown in Tables 1 and 2. The overall elimination efficiency of applying the fast encoding algorithm is 96·01%. Compared with the previous case, it has a slightly lower efficiency. This is because a lower correlation exists for contiguous 4 × 4 subimages.

Table 2 COMPARISONS OF COMPUTATIONAL EFFICIENCIY

Method	Block size	Additions	Multiplications	Comparisons
Conventional	2 × 2	7168	4096	1023
method	4 × 4	7936	4096	255
Proposed	2 × 2	1·57%	2·01%	13·02%
algorithm	4 × 4	3·76%	3·99%	76·30%

Conclusions: A fast encoding algorithm for VQ based image coding has been presented. Simulation results showed that 97·99% savings of both multiplications and additions and 86·98% saving of comparisons were achieved for the case of using subimages of 2 × 2 and codebook of size 1024. The price paid is an increase of memory space to store the distortion table for TIE elimination. This algorithm may also be applied to a VQ based coding for DCT transformed images.

Acknowledgment: This work is funded by the National Science Council, Republic of China.

S.-H. HUANG
S.-H. CHEN

30th July 1990

Department of Communication Engineering and Centre for Telecommunication Research
National Chiao Tung University
Hsinchu, Taiwan 30039
Republic of China

References

1 VIDAL, E.: 'An algorithm for finding nearest neighbours in (approximately) constant average time', *Pattern Recog. Lett.*, 1986, **4**, pp. 145–157

2 CHEN, S. H., and PAN, J. S.: 'Fast search algorithm for VQ-based recognition of isolated words', *IEE Proc. I*, 1989, **136**, pp. 391–396

3 BEI, C.-D., and GRAY, R. M.: 'An improvement of the minimum distortion encoding algorithm for vector quantization', *IEEE Trans.*, 1985, **COM-33**, pp. 1132–1133

4 GRAY, R. M.: 'Vector quantization', *IEEE ASSP Mag.*, 1984, pp. 4–28

NOISE CHARACTERISTICS OF POLARISATION SENSITIVE OPTICALLY PREAMPLIFIED RECEIVERS

Indexing terms: Optical receivers, Noise

Using a new expression for the noise variance of optically amplified systems it is shown that the theoretical SNR degradation caused by uncontrolled polarisation at the input to optically preamplified receivers is minimal for gain sensitivities less than 4 dB. This result is experimentally confirmed.

Introduction: Semiconductor optical preamplifiers can increase the sensitivity of receivers in broadband optical communication systems.[1] Optical amplifiers generally have different gains in the TE and TM modes, and unless the input signal state of polarisation is controlled both the output signal and signal related noise powers may vary with time. In a practical system, control of the input signal state of polarisation may be undesirable because of the insertion loss and increased component count, or impossible when signals are received from more than one optical source. We present a