code for the node's label is transmitted. For the tree in Fig. 1*a*, its symbolic representation would be in the form shown in Fig. 1*b*. We rewrite it in a symbolic sequence as follows:

$$11011000000010100000100000101100000000010000000 \quad (1)$$

Thus, $4N + 5 = 45$ bits in total are required to transmit this quadtree structure by sending node labels in sequence.

Next, we investigate the performance of the run-length coding algorithm applied to encode the quadtree structure and compare it with that of the above traditional label code. The first step of run-length coding is to compute the length of each run in which all elements have the same symbol and different ones from those in neighbouring runs. From eqn. 1, for example, this step results in a run-length sequence as shown in Fig. 1*c*, which is as follows:

2127111515112816

except for the first digit, '1' enclosed in brackets, to encode the root node.

Secondly if a Huffman coding algorithm is applied to encode the above sequence the rate will be

$$8 \times 1 + 3 \times 2 + 2 \times 3 + 1 \times 4 + 1 \times 5 + 1 \times 5 = 34 \text{ bits}$$

In addition, one extra bit should be added to encode the root node. Therefore, 34 + 1 = 35bits in total are required to transmit the quadtree structure. Comparing with the label code, 10bits are saved.

The above is just a specific example which shows that the information is compressed using the run-length encoding algorithm to encode the quadtree structure instead of sending the binary symbolic sequence directly. Next, we demonstrate that it is true not only for a specific case but in a wider sense.

Let us create a structure in which the labels, except the root label which is always 1, are Bernoulli trials with the probability $P[X = 1] = p$ and $P[X = 0] = (1 - p)$. We choose first K Bernoulli trials in which there are N events which are 1. A quadtree structure could then be created with $4(N + 1) - K$ zeros added as labels of leaf nodes.

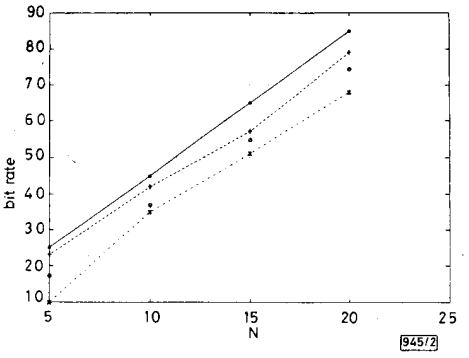

Fig. 2 *Experimental results of run-length code against label code for N ranging from 5 to 20*

* Label code
\+ Maximum
O Average
x Minimum

Fig. 2 shows the results of the performance of run-length code against ordinary label code. The ordinary label code requires $4N + 5$bits to encode a quadtree structure with N non-leaf nodes except the root node. The other three curves display the maximum, average and minimum rate of 10 experiments for the quadtrees created from Bernoulli trials with $p = 0.5$, and $N = 5$, 10, 15 and 20, respectively. The results show that run-length code always performs better than ordinary label code, even in the worst case. The average saving using run-length code instead of ordinary label code is ~10bits per structure for the structures ranging between 25 and 85 nodes.

Conclusions: We have provided the experimental results of a run-length coding algorithm for the coding of the quadtrees generated from Bernoulli trials and compared them with that of ordinary label codes. It is found that the run-length code performs much better. It is demonstrated that the run-length coding algorithm can be used to encode quadtree structures efficiently.

29 July 1993
Electronics Letters Online No: 19931140

H.-S. Wu and J. Barba (*Department of Electrical Engineering, City College of New York, 140 Street and Convent Avenue, New York, 10031, USA*)

References

1 VAISEY, D.J., and GERSHO, A.: 'Variable block-size image coding'. Proc. IEEE, ICASSP'87, 1987, pp. 1051–1054
2 VAISEY, D.J., and GERSHO, A.: 'Image compression with variable block size segmentation'. *IEEE Trans.*, 1992, **SP-40**, pp. 2040–2060
3 BOXERMAN, J.L., and LEE, H.J.: 'Variable block-size vector quantization of grayscale images with unconstrained tiling'. Proc. IEEE, ICASSP'90, 1990, pp. 2277–2280
4 COHEN, Y., LANDY, M.S., and PAVEL, M.: 'Hierarchical coding of binary images', *IEEE Trans.*, 1985, **PAMI-7**, pp. 284–298
5 LIU, S., and HAYES, M.: 'Segmentation-based coding of motion difference and motion field images for low bit-rate video compression'. Proc. IEEE, ICASSP'92, 1992, pp. III-525-III-528
6 SULLIVAN, G.J., and BAKER, R.L.: 'Efficient quadtree coding of images and video'. Proc. IEEE, ICASSP'91, 1991, pp. 2661–2664
7 KUNT, M., IKONOMOPOULOS, A., and KOCHER, M.: 'Second generation image coding techniques'. Proc. IEEE, 1985, pp. 549–574
8 KUNT, M., BENARD, M., and LEONARDI, R.: 'Recent result in high-compression image coding'. *IEEE Trans.*, 1987, **CAS-34**, pp. 1306–1336

Extension of the BCH decodable class of DCT codes

J. Shiu and J.-L. Wu

Indexing terms: BCH codes, Discrete cosine transforms

The discrete cosine transform (DCT) has already been applied to real number error control coding. In the Letter the class of BCH decodable DCT codes is enlarged to form a parity check matrix.

Introduction: In [1], the discrete cosine transform (DCT) has been applied to the problem of real number error control coding. Specifically, a parity check matrix $\boldsymbol{H}$ for an (N, K) linear block code can be formed by selecting the first $(N - K)$ rows of an N-point DCT matrix, and the corresponding generating matrix wlll consist of the rest of the rows. For simplicity, let $d = N - K$ and DCT_N denote the N-point DCT matrix. The syndrome vector $\boldsymbol{S}$ of a received vector v is calculated by

$$\boldsymbol{S} = \boldsymbol{v} \cdot \boldsymbol{H}^T = \boldsymbol{e} \cdot \boldsymbol{H}^T \quad (1)$$

where $\boldsymbol{e}$ is the error vector with v nonzero elements, $e_{i1}, e_{i2}, \ldots, e_{iv}$. As shown in [1], eqn. 1 can be factorised as $\boldsymbol{S} = \boldsymbol{Y} \cdot \boldsymbol{X} \cdot \boldsymbol{C}$ where $\boldsymbol{Y}$ is a $1 \times v$ row vector consisting of the nonzero elements in $\boldsymbol{e}$, $\boldsymbol{X}$ is a $v \times d$ matrix with $X_{ij} = X_j^i = \cos^i (2i_j + 1)\pi/2N$ as its elements, where i_j is the location of the jth error, and $\boldsymbol{C}$ is a $d \times d$ upper triangular matrix in the following form:

$$\begin{bmatrix} C_{0,0} & 0 & C_{2,0} & 0 & \ldots & \\ 0 & C_{1,1} & 0 & C_{3,1} & \ldots & \cdot \\ 0 & 0 & C_{2,2} & 0 & \ldots & \cdot \\ 0 & 0 & 0 & C_{3,3} & \ldots & \cdot \\ & & \vdots & & \vdots & \\ 0 & 0 & 0 & 0 & \ldots & C_{d-1,d-1} \end{bmatrix} \quad (2)$$

Because matrix $\boldsymbol{C}$ is always invertible, this DCT code can be decoded by the decoding algorithms for BCH codes [2] with the modified syndromes $\boldsymbol{S}' = \boldsymbol{S} \times \boldsymbol{C}^{-1}$.

In this Letter the class of BCH decodable DCT codes is further enlarged. Briefly speaking, in addition to the above method for

selecting rows as parities, both selections of rows 0, a, $2a$,..., $(d-1)a$ and a, $3a$, $5a$,..., $(2-1)a$, provided that these rows are distinct, also form a parity check matrix of a BCH decodable DCT code.

Extension of the BCH decodable class of DCT codes: As defined in eqn. 1, if the jth row of DCT_N is contained in the parity check matrix $\boldsymbol{H}$, then we will have a parity check equation or syndrome as follows:

$$S(j) = \sum_{i=0}^{N-1} e_i \cos j\frac{(2i+1)\pi}{2N} = \sum_{i=0}^{N-1} e_i \sum_{\substack{n=0 \\ (j=n):\text{even}}}^{j} C_{j,n}\cos^n \frac{(2i+1)\pi}{2N} \quad (3)$$

Then, for index ja, the parity check equation becomes

$$S(ja) = \sum_{i=0}^{N-1} e_i \sum_{n=0}^{j} C_{j,n}\cos^n a\frac{(2i+1)\pi}{2N} \quad (4)$$

Because it can easily be verified that

$$\cos(2N+j)\frac{(2i+1)\pi}{2N} = -\cos j\frac{(2i+1)\pi}{2N}$$
$$\cos(N+j)\frac{(2i+1)\pi}{2N} = -\cos(N-j)\frac{(2i+1)\pi}{2N} \quad (5)$$

we can find the associated row of DCT_N for index ja. Following similar reasoning in [1], the rows corresponding to indices 0, a, $2a$, $3a$,..., $(d-1)a$ will form a parity check matrix of a BCH decodable DOT code. For example, take $N-1$ for a and suppose that $d = N-K = 6$. Then we will have indices 0, $N-1$, $2N-2$, $3N-3$, $4N-4$, $5N-5$, $6N-6$ which correspond to the rows 0, $N-1$, 2, $N-3$, 4, $N-5$, 6 of DCT_N. The corresponding parity check equations or syndromes are $S(0)$, $S(N-1)$, $-S(2)$, $-S(N-3)$, $S(4)$, $S(N-5)$, $-S(6)$. These syndromes, after being modified by multiplying $\boldsymbol{C}^{-1}$, can be used as input to a decoding algorithm for BCH codes to find the error locators $X_l = \cos(N-1)(2i_l+1)\pi/2N$ and the error values $Y_l = e_{il}$ as suggested in [1].

There is still another way for forming the desired parity check matrix. Consider eqn. 3 again. Because $C_{j,n}$ is zero if $j-n$ is not even, by dropping these zero terms and substituting n by $2n+1$, we will have the following parity check equation:

$$S((2j+1)a) = \sum_{i=0}^{N-1} e_i \sum_{n=0}^{j} C_{2j+1,2n+1}\cos^{(2n+1)} a\frac{(2i+1)\pi}{2N} = \sum_{i=0}^{N-1} e_i \cos a\frac{(2i+1)\pi}{2N}\sum_{n=0}^{j} C_{2j+1,2n+1}(\cos^2 a\frac{(2i+1)\pi}{2N})^n \quad (6)$$

It suggests that the rows of DCT_N corresponding to indices a, $3a$, $5a$,..., $(2d-1)a$ also form a parity check matrix of a BCH decodable DCT code. As a concrete example, let $a = 3$, $N = 16$ and $d = 6$. The sets of selected indices and their corresponding rows and syndromes are, respectively, {3,9,15,21,27,33}, {3,9,15,11,5,1}, and {$S(3)$, $S(9)$, $S(15)$, $-S(11)$, $-S(5)$, $-S(1)$}. A BCH decoding algorithm can use these syndromes to calculate the error locators $X_l = \cos^2 3(2i_l+1)\pi/2N$ and the error values $Y_l = e_{il}\cos 3(2i_l+1)\pi/2N$.

Note that, in both cases, it is not necessary for a to be relatively prime to N. However, if it is, the index set {0, a, $2a$,...,$(N-1)a$} will specify the set of rows {0, 1, 2,..., $N-1$}. Therefore, we can select up to N rows as parities. If a is not relatively prime to N, the index set will be smaller. Consequently, only codes with a higher rate can be defined. For the second case, because indices $(2N-2j-1)a$ and $(2j+1)a$ specify the same rows, {a, $3a$, $5a$ $(2N-1)a$} and {a, $3a$, $5a$,...,$(N-1)a$} will specify the same set of rows. From this set of rows, which has at most $N/2$ elements, only codes with rates higher than 1/2 can be defined.

4 August 1993
Electronics Letters Online No: 19931149
J. S. and J.-L. Wu (*Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 10764, Taiwan, Republic of China*)

References

1 WU, J.-L., and SHIU, J.: 'Real-valued error control coding by using DCT'. *IEE Proc.* I, 1992, Vol. 139 (2), pp. 133–139
2 BLAHUT, R.E.: 'Theory and practice of error control codes' (Addison-Wesley, 1983)

Improved method for calculating mitigation bandwidth of DSPN signals

F. Amoroso

Indexing terms: Spread spectrum communication, Mobile radio systems

Previous work has defined the *mitigation bandwidth* of DSPN as employed to combat fading over the dense scatterer mobile digital link. The Letter gives a simple, elegant method for computing mitigation bandwidth. No sacrifice of precision is conceded, and closed-form expressions are given for BPSK and MSK chip modulation.

Introduction: A mitigation bandwidth [1] has been found to measure the potential of a direct sequence pseudonoise (DSPN) signal to reduce multipath fading over a mobile digital link in very dense scatterers. With propagation delay spread Δ and typical chip pulse $u(t)$, having a Fourier transform $U(\omega)$ the standard deviation of the received signal power at a moving antenna is given by

$$\sigma_n(|s(t)|^2) = \frac{1}{T}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{|U(\omega_1)|^2|U(\omega_2)|^2}{1+[(\omega_1-\omega_2)\Delta]^2}d\omega_1 d\omega_2\right]^{\frac{1}{2}} \quad (1)$$

where $u(t)$ and $U(\omega)$ have been normalised to energy T

$$\int_{-\infty}^{\infty}|u(t)|^2dt = \int_{-\infty}^{\infty}|H(\omega)|^2d\omega = T \quad (2)$$

to ensure unit mean received power for all choices of chip duration T. The previous correspondence [1] evaluated $[\sigma_n(|s(t)|^2)]^2$ for a 'Nyquist' chip pulse having (necessarily) rectangular $U(\omega)$, and showed the mitigation bandwidth W_m to be $2/T$. Calculations presented here will extend the result of that previous correspondence to show that for two additional basic chip modulation types the asymptotic relationship between the variance of received power and the chip bandwidth also takes the form

$$\lim_{T\to 0}[\sigma_n(|s(t)|^2)]^2 = 1/(W_m\Delta) \quad (3)$$

for the appropriate W_m. Thus the mitigation bandwidths of all three chip modulation types play compatible roles as the 'figure of merit' that measures the potential of the specific DSPN type to reduce the received signal variance in a given dense scatterer environment, once the chip rate has been increased to some large multiple of the inverse delay spread $1/\Delta$.

Problem statement: The analytical problem addressed here is the development of a more elegant, yet perfectly general, method of computing the mitigation bandwidths of typical chip modulahon types found in DSPN signalling. The complexities attendant to the obvious, direct application of eqn. 1 for the computation of W_m are quickly encountered in the case of a rectangular chip pulse of duration T. This pulse corresponds to classical binary phase shift keyed (BPSK) modulation at chip rate $1/T$. With the normalisation indicated in eqn. 2 the spectral density is

$$U^2(\omega) = \frac{T^2}{2\pi}\frac{\sin^2(\omega\frac{T}{2})}{(\omega\frac{T}{2})^2} \quad (4)$$

which leads to the following expression [see note 1] for $[\sigma_n(|s(t)|^2)]^2$ subsequent to the evaluation of the inner integral in eqn. 1: