Development of integer cosine transforms by the principle of dyadic symmetry

W.-K. Cham, PhD

Abstract: The paper shows how to convert the order-8 cosine transforms into a family of integer cosine transforms (ICTs) using the theory of dyadic symmetry. The new transforms can be implemented using simple integer arithmetic. It was found that performance close to that of the DCT can be achieved with an ICT that requires only 4 bits for representation of its kernel component magnitude. Better performance can be achieved by some ICTs whose kernel components require longer bit lengths for representation. ICTs that require 3 bits or less for representation of their component magnitude are available but with degraded performance. The availability of many ICTs provides an engineer the freedom to tradeoff performance for simple implementation in designing a transform codec.

1 Introduction

Transform coding can achieve a high data compression rate for image data. A transform coder comprises mainly two parts: the first part transforms highly correlated image data into weakly correlated coefficients using an orthogonal transform and the second part performs adaptive quantisation on coefficients to reduce the bit transmission rate. It has been widely accepted that among the many suboptimal orthogonal transforms the discrete cosine transform (DCT) has the best performance in both data compression and filtering for image data [1, 2]. The DCT has also been shown to be asymptotically optimal for a first-order Markov source [3, 4], which is regarded as a good stochastic representation of image data. As block sizes of 8 and 16 are most appropriate for the transform coding of image data, therefore techniques for implementing order-8 and -16 DCTs economically and with fast computational time are very important for the realisation of a transform codec.

The DCT can be implemented using either a programmable processor or dedicated hardware. For an order-n DCT, the magnitudes of the components of the 0th and $(n/2)$th basis vectors can be reduced to unity by suitably scaling components of these basis vectors. However, components of other basis vectors are irrational numbers which cannot be reduced to integers by simple scaling. Being real numbers, the components cause diffi-

culties in both implementation methods. Dedicated hardware that computes the DCT using floating point arithmetic is very complex and expensive. On the other hand, a programmable processor which computes the DCT by executing a sequence of instructions does not usually have instructions that handle real numbers. Although floating point co-processors which have been specially tailored for processing floating point arithmetic can be found on the commercial market, they are also very expensive.

A simple method to eliminate floating point arithmetic is to approximate the real magnitudes of the DCT components by M-bit integers, so that the DCT can be computed using integer multiplications and additions. With each pixel represented by 8 bits, the 1-D order-n DCT requires n^2 $(8 + M)$-bit multiplication and $(8 + M + \frac{1}{2} \log_2 n)$-bit addition operations. The second stage of the 2-D DCT requires n^2 $(8 + 2M + \frac{1}{2} \log_2 n)$-bit multiplication and $(8 + 2M + \log_2 n)$- bit addition operations. Guglielomo has suggested that 7 bits should be enough to represent the magnitudes of order-16 DCT kernel components without causing significant effects on the transformed and reconstructued signal [5]. Suppose the order-8 DCT also requires 7 bits for representation of the components. The 2-D order-8 DCT thus requires 24-bit multiplication and 25-bit addition operations, which are difficult to implement and introduce a lot of delay in the computation. There are other, simpler transforms, such as the Walsh transform [6, 7], slant transform [8] and the high-correlation transform (HCT) [9], whose multiplication and addition operations require shorter bit lengths, but they all perform unsatisfactorily compared to the DCT.

Jones *et al.* [10] found that the order-8 DCT can be approximated using the orthogonal C-matrix transform [CMT] with small performance degradation. The C-matrix transform is computed via the row bit-reversed Walsh transform $[H]$ and the C-matrix $[CM]$ as follows:

$$[CMT] = [CM] \cdot [H]$$

As $[CM]$ is a sparse block diagnonal matrix containing only integers 13, 12, 5, 4, 3, -3, -4 and -5, and as $[H]$ contains only $+1$ and -1, the C-matrix transform can be implemented using simple integer arithmetic. The work was then extended to orders 16 [11] and 32 [12]. These three C-matrices are all derived by trial and error. In this paper, we show that it is possible to replace transform kernel components of the order-8 DCT by a new set of numbers. Such a technique can be applied to any transform, although solutions are not always guaranteed. For the order-8 DCT, infinite new transforms can be obtained. Boundary conditions are imposed to ensure

that the new transforms, while still resembling the DCT, contain only integers in the magnitude of these integers can be very small, the new transforms thus generated are simple to implement and will be referred to here as integer cosine transforms (ICTs).

The sign of each component of the ICTs is the same as that of the DCT whereas the sign of each component of the CMT is the same as that of the Walsh transform. Therefore, the basis vectors of the ICTs are closer to those of the DCT than the CMT. To access the performance of the ICTs, the transform efficiency [9] of a 1-D Markov source, the mean-square-errors [13] of a 2-D Markov source and real images are used as critieria. It was found that the performance of most of the ICTs was very close to that of the DCT and better than that of the CMT. In fact, some ICTs are even superior to the DCT if transform efficiency (see Section 4.1) is used as the criterion. In general, better performing ICTs have components that require longer bit lengths for representation. The availability of many ICTs therefore provides an engineer with the freedom to tradeoff performance for simple implementation in the design of a transform codec. For example, if implementation simplicity is the paramount criterion, he can choose an ICT whose components can be represented using only 2 bits.

The kernel component replacement technique is based on the principle of dyadic symmetry which has been used to generate the HCT and the LCT orthogonal transforms [9], as well as to provide a unified matrix treatment for binary Walsh matrices [7]. Details of the theory of dyadic symmetry will not be covered here. Interested readers may refer to References 7 and 9.

Unless specified otherwise, all vectors in what follows are column vectors.

2   Dyadic symmetry

Definition of dyadic symmetry:

A vector of 2^m elements $[a_0, a_1, \ldots, a_{2^m-1}]$ is said to have the ith dyadic symmetry if and only if $a_j = s \cdot a_{j\oplus i}$, where $\oplus$ is the 'exclusive or' operation, j lies in the range $[0, 2^m - 1]$ and i in the range $[1, 2^m - 1]$, $s = 1$ when the symmetry is even, and $s = -1$ when the symmetry is odd.

For a vector of eight elements, there are seven possible dyadic symmetries. As an example, Table 1 shows the

Table 1: The seven vectors H_s having Sth even dyadic symmetry

S	Vector H_s							
	$[a_0$	a_1	a_2	a_3	a_4	a_5	a_6	$a_7]$
1	$[h_1$	h_1	h_2	h_2	h_3	h_3	h_4	$h_4]$
2	$[h_1$	h_2	h_1	h_2	h_3	h_4	h_3	$h_4]$
3	$[h_1$	h_2	h_2	h_1	h_3	h_4	h_4	$h_3]$
4	$[h_1$	h_2	h_3	h_4	h_1	h_2	h_3	$h_4]$
5	$[h_1$	h_2	h_3	h_4	h_2	h_1	h_4	$h_3]$
6	$[h_1$	h_2	h_3	h_4	h_3	h_4	h_1	$h_2]$
7	$[h_1$	h_2	h_3	h_4	h_4	h_3	h_2	$h_1]$

vectors $H_S = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$ which have the seven even dyadic symmetries.

Theorem of orthogonality: Two vectors U and V are orthogonal if U and V have the same type of dyadic symmetry and one is even and the other is odd.

Proof: It is obvious that the dot product of vectors U and V are zero, therefore U and V are orthogonal.

3   Generation of the order-8 ICTs

Let $[T]$ be the kernel of the order-n discrete cosine transform whose (i, j)th element is $T_n(i, j)$. $T_n(i, j)$ is the jth component of the i-th DCT basis vector and is equal to

$$T_n(i, j) = 1/\sqrt{n} \qquad\qquad \text{for } i = 0$$
$$= (2/\sqrt{n}) \cos \{(i + 0.5)\pi j/n\} \quad \text{for } i \in [1, n - 1]$$

$$(1)$$

where

$$j \in [0, n - 1]$$

The following describes the steps to convert the order-8 DCT kernel into ICT kernels.

Step 1 to express the order-8 DCT kernel $[T]$ in the form of a matrix of variables: Let n of eqn. 1 be 8 so that we obtain an 8×8 matrix

$$[T] = [k_0 J_0, k_1 J_1, k_2 J_2, k_3 J_3,$$
$$k_4 J_4, k_5 J_5, k_6 J_6, k_7 J_7]^t \quad (2)$$

where J_i is the i-th basis vector and k_i is a scaling constant such at $|k_i \cdot J_i| = 1$. Let $J(i, j)$ be the j-th element of J_i. For simplicity, we shall denote $T_8(i, j)$ by $T(i, j)$. As $T(1, 0) = -T(1, 7) = -T(3, 2) = T(3, 5) = -T(5, 1) = T(5, 6) = -T(7, 3) = T(7, 4)$, we may represent the magnitudes of $J(1, 0)$, $J(1, 7)$, $J(3, 2)$, $J(3, 5)$, $J(5, 1)$, $J(5, 6)$, $K(7, 3)$ and $J(7, 4)$ by a single variable, say 'a'. Similarly, all eight basis vectors are expressed as variables 'a', 'b', 'c', 'd', 'e' and 'f' as shown in Table 2. Suppose k_i's are

Table 2: The 8 scaled basis vectors in $|J|$

i	J_i							
0	1	1	1	1	1	1	1	1
1	a	b	c	d	-d	-c	-b	-a
2	e	f	-f	-e	-e	-f	f	e
3	b	-d	-a	-c	c	a	d	-b
4	1	-1	-1	1	1	-1	-1	1
5	c	-a	d	b	-b	-d	a	-c
6	f	-e	e	-f	-f	e	-e	f
7	d	-c	b	-a	a	-b	c	-d

chosen such that d and f are unity, then a, b, c and e are varaibles representing irrational numbers whose values are approximately 5.027, 4.2620, 2.8478 and 2.4142 respectively.

Step 2 to find the conditions under which J_i and J_j are orthogonal: From Table 2, we can see that each basis vector has at least one dyadic symmetry. Table 3 lists the type of dyadic symmetry present in each basis vector. We then examine the condition under which the ith basis vector J_i and the jth basis vector J_j are orthogonal for all i, j and I/J. For example, we can see that J_0 and J_1 are always orthogonal to each other for all a, b, c and d because J_1 has odd 7th dyadic symmetry and J_0 has even 7th dyadic symmetry. Therefore, by means of the theorem of orthogonality, J_0 and J_1 are always orthogonal. J_0 and J_2 are also always orthogonal for all e and f because

Table 3: Sth dyadic symmetry type in basis vector J_i

	Dyadic symmetry S in J_i						
i	1	2	3	4	5	6	7
0	E	E	E	E	E	E	E
1	—	—	—	—	—	—	O
2	—	—	O	—	—	—	E
3	—	—	—	—	—	—	O
4	O	O	E	E	O	O	E
5	—	—	—	—	—	—	O
6	—	—	O	—	—	—	E
7	—	—	—	—	—	—	O

E and O represent even and odd dyadic symmetry respectively. '—' implies that J_i has no Sth dyadic symmetry

J_2 has odd 3rd dyadic symmetry and J_0 has even 3rd dyadic symmetry. In another example, we can see that J_1 and J_2 are orthogonal to each other for all a, b, c, d, e and f because J_1 has odd 7th dyadic symmetry while J_2 has even 7th dyadic symmetry. The conditions under which J_i and J_j are orthogonal are summarised in Table 4. Table 4 reveals that the only condition that the con-

Table 4: Conditions under which the ith basis vector and the jth basis vector are orthogonal

	i							
1	2	3	4	5	6	7	j
*3	*2	*3	*2	*3	*2	*3	0
	*3	*1	*3	*1	*3	*4	1
		*3	*2	*3	*4	*3	2
			*3	*4	*3	*1	3
				*3	*2	*3	4
					*3	*1	5
						*3	6

*1 if $a \cdot b = a \cdot c + b \cdot d + c \cdot d$
*2 must be orthogonal due to the 3rd dyadic symmetry
*3 must be orthogonal due to the 7th dyadic symmetry
*4 must be orthogonal as their dot product equals zero

stants a, b, c, d and e must satisfy to ensure that the transform $[T]$ be orthogonal is

$$a \cdot b = a \cdot c + b \cdot d + c \cdot d \qquad (3)$$

Eqn. 3 has four variables and has an infinite number of solutions. This implies that an infinite number of new orthogonal transforms can be generated from the DCT.

Step 3 to set up boundary conditions and generate new transforms: Eqn. 1 implies that for the DCT

$$a \geqslant b \geqslant c \geqslant d \quad \text{and} \quad e \geqslant f \qquad (4)$$

To make the basis vectors of the new transforms resemble those of the DCT, inequality exprs. 4 have to be satisfied. Furthermore, to eliminate truncation error due to non-exact representation of the basis components a, b, c, d, e and f, condition expr. 5 has to be satisfied:

$$a, b, c, d, e \text{ and } f \text{ are integers} \qquad (5)$$

Transforms $[T]$ that satisfy the conditions of exprs. 3, 4 and 5 are referred to here as order-8 integer cosine transforms (ICTs). As an example, ICT(5, 3, 2, 1, 3, 1) refers to the order-8 ICT with $a = 5$, $b = 3$, $c = 2$, $d = 1$, $e = 3$, $f = 1$. Fig. 1 shows the basis vectors of this ICT together with those of the DCT.

4 Performance of order-8 ICTs

4.1 Transform efficiency [9] performance

In the transform coding of pictures, transforms are used to convert highly correlated signals into coefficients of

low correlation. Such decorrelation ability may be measured by the transform efficiency η, which is defined on a first-order Markov process of adjacent element correlation ρ. A large η implies a high decorrelation ability. The
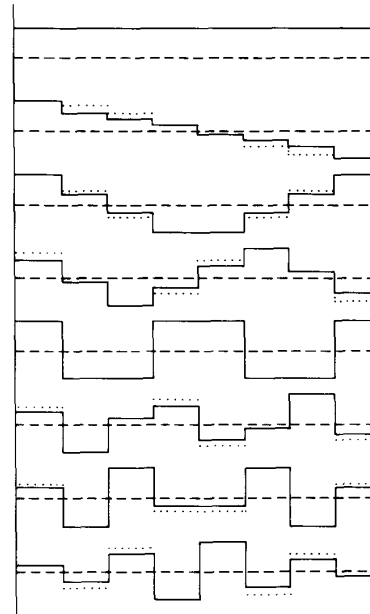

Fig. 1 *Basis vectors of the order-8 DCT and the ICT(5, 3, 2, 1, 3, 1)*

——— ICT
········ DCT

optimal K-L transform which converts signals into completely uncorrelated coefficients has a transform efficiency equal to 100% for all ρ.

Let the n-dimensional vector X be a sample from a one-dimensional, zero-mean, unit-variance first-order Markov process with adjacent element correlation ρ, and covariance matrix $[C_X]$ where the (i, j)th element of $[C_X]$ is $\rho^{|i-j|}$. The efficiency of the transform $[T]$ is defined on the transform domain covariance matrix $[C_Y]$ of vector Y where

$$Y = [T]X$$

$$[C_Y] = E[Y \cdot Y^t]$$

$$= [T][C_X][T]^t$$

$$= \begin{bmatrix} s_{11} & \cdots & \cdots & s_{1n} \\ \vdots & & & \vdots \\ s_{n1} & \cdots & \cdots & s_{nn} \end{bmatrix}$$

$$\text{Efficiency } \eta = \frac{\sum_{i=1}^{n} |s_{ii}|}{\sum_{p=1}^{n} \sum_{q=1}^{n} |s_{pq}|}$$

The DCT, which is widely accepted as the best suboptimal transform, has the highest transform efficiency of the well-known suboptimal transforms for ρ close to unity. A computer search has been performed to find the set of (a, b, c, d) that gives the highest transform efficiency for a less than or equal to 255, and (e, f) equal to (1, 0), (4, 1), (3, 1), (2, 1) and (1, 1). It was found that $e = 3$ and $f = 1$ always gives a higher transform efficiency for the same (a, b, c, d). Table 5 lists the twelve order-8 ICTs that

have the highest transform efficiencies for ρ equal to 0.9 and a less than or equal to 255. It can be seen that all twelve ICTs have high transform efficiencies than the

Table 5: The twelve order-8 ICTs that have the highest transform efficiencies for ρ equal 0.9 and a less than or equal to 255

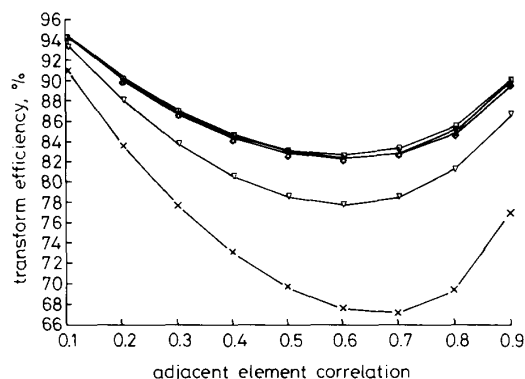| Transform efficiency	Transform
90.221	ICT(230, 201, 134, 46, 3, 1)
90.220	ICT(175, 153, 102, 35, 3, 1)
90.219	ICT(120, 105, 70, 24, 3, 1)
90.217	ICT(185, 162, 108, 37, 3, 1)
90.217	ICT(250, 219, 146, 50, 3, 1)
90.215	ICT(65, 57, 38, 13, 3, 1)
90.213	ICT(55, 48, 32, 11, 3, 1)
90.213	ICT(205, 180, 120, 41, 3, 1)
90.212	ICT(140, 123, 82, 28, 3, 1)
90.211	ICT(215, 189, 126, 43, 3, 1)
90.210	ICT(75, 66, 44, 15, 3, 1)
90.208	ICT(235, 207, 138, 47, 3, 1)
89.836	DCT
86.785	CMT
85.842	slant transform
84.097	HCT
77.140	Walsh transform

Fig. 2 *Transform efficiency for different values of adjacent element correlation*

△ DCT
× Walsh transform
◇ ICT(230, 201, 134, 46)
+ ICT(55, 48, 32, 11)
□ ICT(10, 9, 6, 2)
▽ CMT

order-8 DCT. Fig. 2 shows how the transform efficiency of various transforms varies with adjacent element correlation ρ. As shown in Fig. 2, the transform efficiencies of the DCT and the ICTs are very close to each other and are always better than those of the CMT and the Walsh transform for adjacent element coefficients between 0.1 and 0.9.

4.2 Basis restriction mean-square-error performance

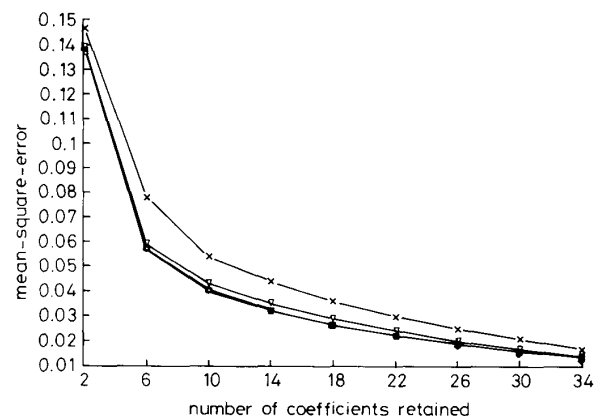The data compression ability of a transform can be measured by means of the basis restriction mean-square-error

Fig. 3 *Basis restriction mean-square-errors of transforms for adjacent element correlation ρ equal to 0.95*

□ ICTs
◇ KLT
△ DCT
× Walsh transform
▽ CMT

[13]. Consider a two-dimensional zero-mean unit-variance nonseparable isotropic Markov process with covariance function

$$C_x(i, j; p, q) = E[x_{i,j} \cdot x_{p,q}] = \rho^{\sqrt{(i-p)^2 + (j-q)^2}}$$

where ρ is the adjacent element correlation in the vertical and horizontal directions. Let the n by n matrix $[X]$ be a sample of the Markov process. Suppose $[X]$ is transformed into $[C]$ by transform $[T]$, i.e.

$$[C] = [T] \cdot [X] \cdot [T]^t$$

where the elements of $[X]$ and $[C]$ are $x_{i,j}$ and $c_{u,v}$, respectively. The covariance function of $[C]$ is

$$C_c(u, v; r, s) = E[c_{u,v} \cdot c_{r,s}]$$

$$= \sum_i \sum_j \sum_p \sum_q C_x(i, j; p, q)$$

$$\cdot T(u, i) \cdot T(v, j) \cdot T(r, p) \cdot T(s, q)$$

Hence, we have the variance of $c_{u,v}$ equal to

$$\sigma_c(u, v) = C_c(u, v; u, v)$$

Let Ω be the set containing M index pairs (u, v) corresponding to the largest M $\sigma_c(u, v)$. The basis restriction

Table 6: Basis restriction mean-square-errors for adjacent element correlation equal to 0.95

no. of coefficients retained	KLT	DCT	ICT(230, 201, 134, 46, 3, 1)	ICT(55, 48, 32, 11, 3, 1)	ICT(10, 9, 6, 2, 3, 1)	CMT	Walsh
2	0.1372	0.1381	0.1381	0.1381	0.1382	0.1387	0.1468
6	0.0567	0.0572	0.0573	0.0573	0.0573	0.0587	0.0785
10	0.0406	0.0409	0.0410	0.0410	0.0410	0.0431	0.0541
14	0.0320	0.0322	0.0323	0.0323	0.0323	0.0348	0.0441
18	0.0263	0.0264	0.0266	0.0266	0.0266	0.0287	0.0361
22	0.0221	0.0222	0.0223	0.0223	0.0224	0.0238	0.0300
26	0.0189	0.0189	0.0190	0.0190	0.0190	0.0198	0.0251
30	0.0160	0.0160	0.0162	0.0162	0.0162	0.0165	0.0205
34	0.0136	0.0136	0.0137	0.0137	0.0137	0.0140	0.0170

Table 7: Mean-square-errors (MSE) of the ICTs that have the highest transform efficiencies

Transform or ICT(a, b, c, d)	Transform efficiency	MSE of 'Girl'	MSE of 'House'	MSE of 'Stone'
DCT	89.8	10.6	22.3	17.3
CMT	86.8	10.8	22.9	17.7
slant transform	85.8	10.6	22.7	17.7
HCT	84.1	12.0	24.8	19.2
Walsh transform	77.1	15.8	32.3	25.1
a is 255 or less and needs 8 bits or less for representation				
ICT(230 201 134 46)	90.2	10.8	22.4	17.3
ICT(175 153 102 35)	90.2	10.8	22.4	17.3
ICT(120 105 70 24)	90.2	10.8	22.4	17.3
ICT(185 162 108 37)	90.2	10.8	22.4	17.3
ICT(250 219 146 50)	90.2	10.8	22.4	17.3
a is 127 or less and needs 7 bits or less for representation				
ICT(120 105 70 24)	90.2	10.8	22.4	17.3
ICT(65 57 38 13)	90.2	10.8	22.4	17.3
ICT(55 48 32 11)	90.2	10.8	22.4	17.3
ICT(75 66 44 15)	90.2	10.8	22.4	17.4
ICT(85 75 50 17)	90.2	10.8	22.4	17.3
a is 63 or less and needs 6 bits or less for representation				
ICT(55 48 32 11)	90.2	10.8	22.4	17.3
ICT(10 9 6 2)	90.2	10.8	22.5	17.3
ICT(55 51 34 11)	90.1	10.9	22.6	17.4
ICT(45 39 26 9)	90.1	10.7	22.4	17.3
ICT(45 42 28 9)	90.1	10.9	22.6	17.4
a is 31 or less and needs 5 bits or less for representation				
ICT(10 9 6 2)	90.2	10.8	22.5	17.3
ICT(25 24 16 5)	89.9	11.0	22.7	17.4
ICT(25 21 14 5)	89.8	10.7	22.4	17.3
ICT(24 21 15 4)	89.6	10.8	22.5	17.3
ICT(26 24 15 6)	89.6	10.8	22.6	17.4
a is 15 or less and needs 4 bits or less for representation				
ICT(10 9 6 2)	90.2	10.8	22.5	17.3
ICT(15 15 10 3)	89.4	11.1	22.9	17.6
ICT(15 12 8 3)	89.1	10.7	22.3	17.3
ICT(14 12 9 2)	89.1	10.9	22.6	17.4
ICT(12 10 6 3)	88.3	10.6	22.5	17.4
a is 7 or less and needs 3 bits or less for representation				
ICT(6 6 3 2)	83.2	11.3	23.4	18.0
ICT(5 3 2 1)	81.1	11.0	23.4	18.3
ICT(7 4 3 1)	80.0	11.0	23.6	18.4
ICT(3 2 1 1)	80.0	11.5	24.3	19.0

$e = 3; f = 1$

mean-square-error is defined as

$$e(M) = 1 - \left\{ \sum_{u, v \in \Omega} \sigma_c(u, v)^2 \Big/ \sum_u \sum_v \sigma_c(u, v)^2 \right\}$$

Fig. 3 and Table 6 show comparisons of the basis restriction mean-square-errors of various transforms for ρ equal to 0.95. The ICTs being tested are ICT(230, 201, 134, 46, 3, 1), ICT(55, 48, 32, 11, 3, 1) and ICT(10, 9, 6, 2, 3, 1) whose basis restriction mean-square-errors, as shown in Table 6, are nearly the same and so are represented using one curve in Fig. 3. Fig. 3 shows that the basis restriction mean-square-errors of the ICTs, the KLT and the DCT are very close and always smaller than those of the CMT and the Walsh transform. Table 6 shows that the basis restriction mean-square-error of the KLT is in fact smaller than that of the DCT which in turn is smaller than those of the ICTs.

4.3 Mean-square-error performance on real images

Tests have also been performed using real images. Images are first divided to form horizontal order-8 vectors and then transformed. With five high-sequence coefficients truncated, the transformed vectors are inverse transformed back into the spatial domain. The mean-square-error between the original and the processed image is used as the critierion for assessing the performance of a transform. Table 7 lists the results for the mean-square-error test performed on the three images shown in Fig. 4.
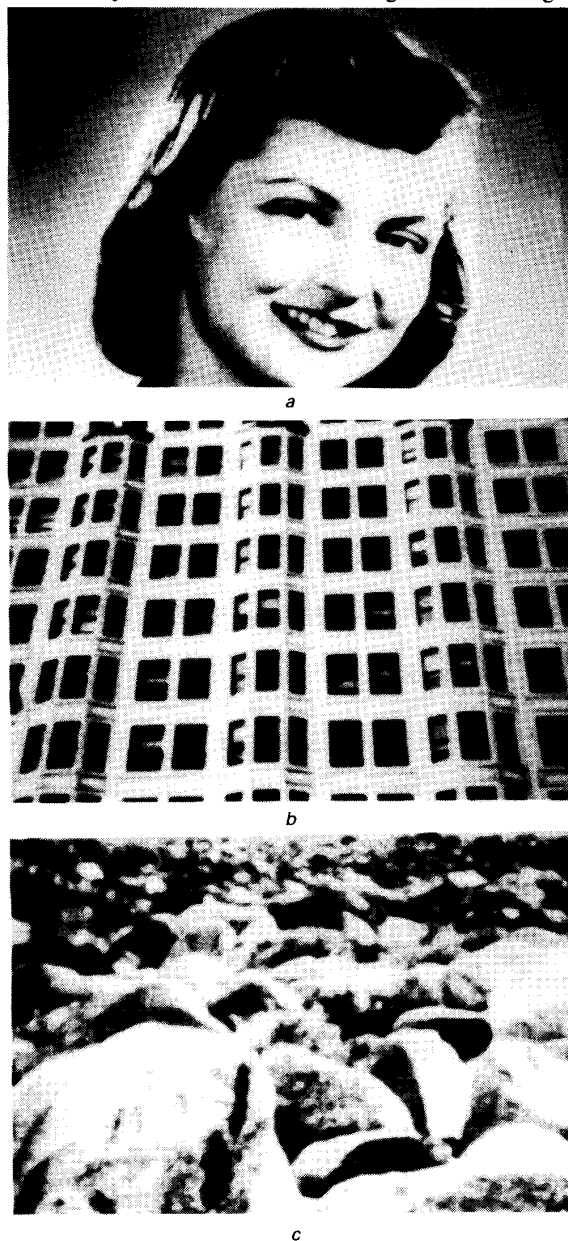

Fig. 4 The three images used in the mean-square-error test
a Girl
b House
c Stone

It was found that the mean-square-error performance of the transforms varies slightly from image to image. For image 'Girl', the slant transform has the smallest mean-square-error and is followed by ICT(12, 10, 6, 3, 3, 1), the DCT and then ICT(15, 12, 8, 3, 3, 1). For image 'House', the DCT has the smallest mean-square-error and is followed by ICT(15, 12, 8, 3, 3, 1) and ICT(25, 21, 14, 5, 3, 1). For image 'Stone', ICT(25, 21, 14, 5, 3, 1), ICT(45, 39, 26, 9, 3, 1) and ICT(55, 48, 32, 11, 3, 1) produce the smallest mean-square-errors.

ICT(10, 9, 6, 2, 3, 1), whose implementation complexity is about the same as that of the CMT, has smaller mean-

square-errors than the CMT, for all three images. The Walsh transform, which has the smallest transform efficiency and the largest basis restriction mean-square-error, has the largest mean-square-error for all three image. In general, transforms having larger transform efficiencies and smaller basis restriction mean-square-errors usually have smaller mean-square-errors in the real-image test. However, it should be noted that results drawn from the real-image test are only used to confirm the results based on the stochastic processes and should not be generalised.

5 Implementation

Let $[T]$ be an ICT. From eqn. 2, we have

$$[T] = [K][J] \tag{7}$$

$$[T]^{-1} = [T]^t = [J]^t[K] \tag{8}$$

where $[K]$ is a diagonal matrix whose (i, i)th element is k_i and $[J]$ is a matrix whose ith basis vector is given by Table 2. As $[T]$ is an ICT, therefore $[J]$ contains only integers. Fig. 5 shows an adaptive 1-D transform coding
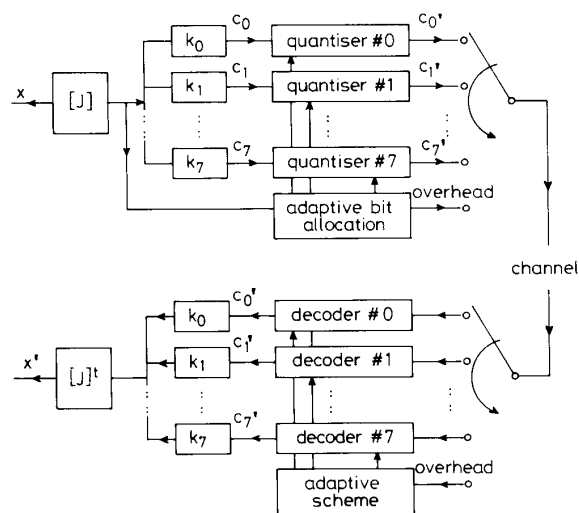
Fig. 5 Adaptive 1-D transform coding system utilising an order-8 ICT $[J]$

system that utilises an order-8 ICT. In the transmitter, signal vector X is first transformed by $[J]$ and then each transformed coefficient is multiplied by the corresponding k_i to form c_i. Coefficient c_i is then quantised by quantiser $\#i$ under the control of an adaptive scheme to form the quantised coefficient c_i', which is then multiplexed to the channel. At the receiver, each c_i' is demultiplexed and decoded separately into a form suitable for the multiplication process and the inverse transform. The quantisation process at the transmitter and the decoding process at the receiver are most likely performed using microprocessors by means of table-lookup. This implies that the multiplication processes at the transmitter can be easily incorporated into the quantisation process by modifying the input entries of the quantisation lookup table; the multiplication processes at the receiver can be incorporated into the decoding process by modifying the output entries of the decoding lookup table. Therefore, real number multiplications may be completely eliminated in an ICT transform coder.

To speed up the transformation process, the integer transform $[J]$ should be implemented using a dedicated

chip. The complexity of the chip depends on the magnitude of the basis vector component a (see Table 2) because a determines the maximum number of shifts required in each multiplication. For example, multiplication operations within ICT(10, 9, 6, 2, 3, 1) require only three shifts, whilst the multiplication operations within the DCT require six shifts if its kernel components are represented using seven bits [5]. With fewer shift operations, an ICT(10, 9, 6, 2, 3, 1) chip is not only simpler but also faster. As shown in Sections 4.1 to 4.3, the performance of the ICT(10, 9, 6, 2, 3, 1) with respect to transform efficiency, basis restriction mean-square-error and mean-square-error is better than that of the CMT and is very close to that of the DCT. Therefore, ICT(10, 9, 6, 2, 3, 1) is a good substitute for the order-8 DCT. In cases where simplicity and speed of the transform chip are of paramount importance, ICT(5, 3, 2, 1, 3, 1) and ICT(3, 2, 1, 1, 3, 1) are other choices. A transform codec designer can therefore tradeoff transform compression ability for computation speed and implementation simplicity by choosing an appropriate transform. Furthermore, as shown in Fig. 6, all ICTs can also be computed using a
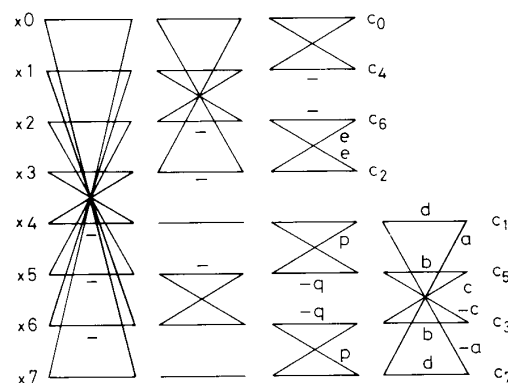
Fig. 6 Fast computational algorithm for order-8 ICT where $p = (b + c)/2a$ and $q = (a - d)/2c$

fast computational algorithm which is very similar to that of the DCT [3] and has 4 iterations.

6 Generalisation to larger block sizes

An order-$2n$ orthogonal transform $T_{2n}(i, j)$ can be generated from an order-n $T_n(i, j)$ transform as follows:

(a) the first n basis vectors of $T_{2n}(i, j)$:

$$T_{2n}(i, 2j) = T_n(i, j)$$

and

$$T_{2n}(i, 2j + 1) = T_n(i, j) \quad \text{for } j \in [0, n - 1]$$

(b) the last n basis vectors of $T_{2n}(i, j)$:
(i) $T_{2n}(i + n, 2j) = T_n(i, j)$
and

$$T_{2n}(i + n, 2j + 1) = - T_n(i, j)$$

$$\text{for } j \in \{0, 2, 4, \ldots, n - 2\}$$

(ii) $T_{2n}(i + n, 2j) = - T_n(i, j)$
and

$$T_{2n}(i + n, 2j + 1) = T_n(i, j)$$

$$\text{for } j \in \{1, 3, 5, \ldots, n - 1\}$$

It was found that the basis vectors of $T_{2n}(i, j)$ generated using the above method resemble those of the $T_n(i, j)$. The

order-8 ICT may therefore be generalised to any order-2^m ICT for $m > 3$. Computer programs have been used to search for order-16 ICTs whose transform efficiency is the highest for a limited to 7, 15, 31, up to 255 and e equal to 3 and f equal to 1. It was found that ICT(246, 222, 147, 50, 3, 1) has the highest transform efficiency for a less than or equal to 255 and ICT(10, 9, 6, 2, 3, 1) has the highest transform efficiency for a less than or equal to 128. However, as given by Table 8, the transform effi-

Table 8: Transform efficiency of the order-16 ICTs and other well known transforms

Transform efficiency	(a, b, c, d)
82.3	DCT
74.1	slant transform
73.9	(246, 222, 147, 50)
73.8	(10, 9, 6, 2)
73.7	CMT
68.4	HCT
60.9	Walsh transform

ciency performance of the order-16 ICTs is inferior to that of the other well known order-16 orthogonal transforms. This could be because the order-16 ICTs have only eight levels in their kernel whilst the DCT and the slant transform have 16 levels.

7 Conclusion

In this paper, the concept of dyadic symmetry has been used to modify the order-8 DCT and generate many new integer orthogonal transforms which are called the integer cosine transforms (ICTs). The basis vectors of these transforms are similar to those of the DCT and all ICTs can be implemented by integer arithmetic. The implementation complexity of an ICT depends on the number of bits required to represent the magnitude of its kernel components. In comparison with the CMT, which is an integer approximation of the DCT, an ICT whose transform kernel contains only 10, 9, 6, 2, 3, and 1 has similar implementation complexity but higher decorrelation ability and better mean-square-error performance. Even better performance can be achieved by some ICTs whose kernel components require longer bit lengths for

representation. ICTs that require three bits or less for representation of their component magnitudes are available but have a less satisfactory performance. The availability of many ICTs thus provides an engineer with the freedom to tradeoff performance for simple implementation in designing a transform codec. The order-8 ICT can be computed using a fast computational algorithm which is similar to that of the order-8 DCT but requires only integer multiplication and addition operations. The order-8 ICTs can be generalised to transforms of larger block sizes, however the performance cannot match that of the order-8 transforms.

8 References

1 CHEN, W., and SMITH, C.H.: 'Adaptive coding of monochrome and color images', *IEEE Trans.*, 1977, **COM-25**, (11), pp. 1285–1292
2 HEIN, D., and AHMED, N.: 'On a real-time Walsh-Hadamard/cosine transform image processor', *IEEE Trans.*, 1978, **EMC-20**, (3), pp. 453–457
3 AHMED, N., NATARAJAN, T., and RAO, K.R.: 'Discrete cosine transform', *IEEE Trans.*, 1974, **C-23**, (1), pp. 90–93
4 SHANMUGAM, K.S.: 'Comments on "Discrete cosine transform"', *IEEE Trans.*, 1975, **C-24**, (7), p. 759
5 GUGLIELMO, M.: 'An analysis of error behavior in the implementation of 2-D orthogonal transform', *IEEE Trans.*, 1986, **COM-34**, (9), pp. 973–975
6 FINO, B.J., and ALGAZI, V.R.: 'Unified matrix treatment of the fast Walsh-Hadamard transform', *IEEE Trans.*, 1976, **C-25**, (11), pp. 1142–1146
7 CHAM, W.K., and CLARKE, R.J.: 'Dyadic symmetry and Walsh matrices', *IEE Proc. Commun., Radar & Signal Process.*, 1987, **134**, (2), pp. 141–144
8 PRATT, W.K., CHEN, W., and WELCH, L.R.: 'Slant transform image coding', *IEEE Trans.*, 1974, **COM-22**, (8), pp. 1075–1093
9 CHAM, W.K., and CLARKE, R.J.: 'Application of the principle of dyadic symmetry to the generation of orthogonal transforms', *IEE Proc. F, Commun., Radar & Signal Process.*, 1986, **133**, (3), pp. 264–270
10 JONES, H.W., HEIN, D.N., and KNAUER, S.C.: 'The Karhunen-Loeve discrete cosine and related transforms obtained via the Hadamard transform'. Proc. Intl. Telemetering Conference, Los Angeles, November 1978, **14**, pp. 87–98
11 SRINIVASAN, R., and RAO, K.R.: 'An approximation to the discrete cosine transform for $N = 16$', *Signal Process.*, 1983, **5**, pp. 81–85
12 KWAK, H.S., SRINIVASAN, R., and RAO, K.R.: 'C-matrix transform', *IEEE Trans.*, 1983, **ASSP-31**, (5), pp. 1304–1307
13 JAIN, A.K.: 'Advances in mathematical models for image processing', *Proc. IEEE*, 1981, **69**, (5), pp. 502–528