A Low Complexity Orthogonal 8×8 Transform Matrix for Fast Image Compression

Ranjan K. Senapati, Umesh C. Pati and Kamala K. Mahapatra

Department of Electronics and Communication Engineering
National Institute of Technology-Rourkela,
Orissa, Pin- 769008, INDIA.
Email:ranjankumarsenapati@gmail.com, ucpati@nitrkl.ac.in, kkm@nitrkl.ac.in

***Abstract*— In this paper, an efficient orthogonal sparse 8×8 transform matrix for low bit-rate image compression is proposed. The transform matrix is made sparse by appropriately inserting additional zeros into the matrix proposed by Bouguezel. The algorithm for fast computation is also developed. It is shown that the proposed transform matrix provides a 20% reduction in computation over the matrix proposed by Bouguezel, and 45% over signed discrete cosine transform (SDCT). By using various test images, it is shown that the rate-distortion performance is also almost comparable to that of above two transform matrices at low bit-rates.**

***Keywords*— Signed discrete cosine transform (SDCT), Image compression, Percentage error energy norm (PEEN).**

I. Introduction

Image Transform methods using orthogonal kernel functions are commonly used in image compression. One of the most widely known image transform method is Discrete cosine transform (DCT), used in JPEG compression standard [1]. Applications such as multimedia, mobile communications, personal digital assistants (PDAs), digital cameras and Internet require a lot of image transmission and processing. Therefore, it is essential to have image compression techniques which should be faster and efficient in terms of subjective and objective image reconstruction capability.

Even though a number of algorithms for fast computation of DCT are available in the literature, there has been a lot of interest towards finding out the approximate integer versions of floating point DCT [2-7]. Cham [2] proposed a family of integer cosine transform (ICTs) using the theory of dyadic symmetry. Cham has shown that the performances of ICTs are close to that of DCT. Dimitrov *et al.* [3] presented a novel architecture for a 2D 8×8 DCT which needs only 24 adders. The architecture allows scalable computation of 2D 8×8 DCT using integer encoding of 1D radix-8 DCT. 8x8 versions of two transformation matrices, one for the coarsest and another for the finest, (represented as $\hat{\boldsymbol{D}}_1$ and $\hat{\boldsymbol{D}}_5$ respectively) approximation levels of exact DCT is proposed in [4]. Using these two matrices a tradeoff of speedup versus accuracy in various bit ranges can be achieved. The performance shows 73% complexity reduction with only 0.2 dB PSNR degradation. Tran [5] has presented a family of 8x8 biorthogonal transforms called binDCT, which are all approximates of popular 8×8 DCT. These binDCT show a coding gain of range 8.77-8.82 dB despite requiring as low as 14 shifts and 31 additions per eight input samples. Tran's 8×8 binDCT shows finer approximations to exact DCT and are suitable for VLSI implementation. Haweel [6] proposed signed DCT (SDCT) by applying signum function to DCT. However, SDCT and its inverse are not orthogonal and it needs 24 additions for transformation. Bouguezel *et al.* [7] has presented an 8×8 transform matrix by appropriately inserting 20 zeros into the elements of $\hat{\boldsymbol{D}}_1$ [4]. A 25% reduction in computation complexity is achieved over SDCT and this matrix is orthogonal.

In this paper we presented a novel 8×8 orthogonal transform matrix which is a modification over the transform proposed by [6] and [7]. The proposed matrix is sparse and has 24 zeros entries. This matrix also satisfies orthogonality property. A fast algorithm for computation of the matrix is also presented. It can be shown that the proposed matrix provides a 20% reduction in computation over the matrix in [7] and 45% over SDCT [6].

The organization of the paper is as follows: Section II presents the introduction about SDCT. The proposed 8×8 transform matrix is presented in Section III. The algorithm for efficient computation is developed in Section IV. Section V shows the application of the proposed transform to a standard JPEG baseline coder for image compression. Simulation result and comparisons are discussed in Section VI. Finally conclusions are given in Section VII.

II. Signed DCT

The two dimensional DCT, $T_{DCT}(u,v)$ of order N×N is defined as:

$$T_{DCT}(u,v) = \alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\cos\left[\frac{\pi(2i+1)u}{2N}\right]\cos\left[\frac{\pi(2j+1)v}{2N}\right]$$

$$\text{where, } \alpha(u)\alpha(v) = \begin{cases} 1/\sqrt{N} & \text{for } u,v=0, \\ 2/\sqrt{N} & \text{otherwise.} \end{cases} \qquad (1)$$

The signed discrete cosine transform (SDCT), $T_{SDCT}(u,v)$ is obtained by applying the signum function operator to the elements of DCT obtained form (1).

978-1-4244-9073-5/10/$26.00 ©2010 IEEE

Therefore $T_{SDCT}(u,v)$ is given as:

$$T_{SDCT}(u,v) = \frac{1}{\sqrt{N}} sign\{T_{DCT}(u,v)\}. \tag{2}$$

where, sign{.} is the signum function defined as:

$$sign\{x\} = \begin{cases} +1 & if \quad x > 0, \\ 0 & if \quad x = 0, \\ -1 & if \quad x < 0. \end{cases} \tag{3}$$

Several advantages of SDCT are apparent form (1), (2) and (3). These are (i) All the elements are 0 or ±1, (ii) No multiplication operation or transcendental expressions are required (iii) Unlike WHT [8], RSWT [9] and SDFT [10], the transform order need not be a specific integer or a power of 2. (iv) SDCT maintains the periodicity and spectral structure of its originating DCT and maintains good de-correlation and energy compaction characteristics. It has been verified that only 10% spectral components of SDCT contains 80% of the total signal power compared to 87% signal power in DCT.
As an example, the 8×8 SDCT transform matrix is given by

$$T_{SDCT} = \frac{1}{\sqrt{8}} \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} \tag{4}$$

It is worth mentioning that the forward transform of T_{SDCT}, is almost orthogonal. But its inverse transform is not orthogonal. Hence this matrix is only used for some applications like adaptive filtering where forward transform is employed. For applications like image compression, Hawel [6] proposed another reverse transform matrix called $T^{r}{}_{SDCT}$, where all elements are ±1, ±2, or 0. Like T_{SDCT}, it does not need any multiplication operation for reverse transformation, except shift and sign bit changes. Moreover, 25% of the elements are zeros. Unfortunately, this special feature of the matrix is not valid for all orders of N.

III. Proposed 8×8 Transform Matrix

The proposed 8×8 transform matrix can be obtained by appropriately inserting some 0's and 0.5's into the SDCT matrix in (4). These elements should be arranged in such a way that, the matrix should not be singular and satisfy orthogonality property. The proposed matrix contains 24 zeros, in comparison with 20 zeros with the matrix proposed in [7]. A multiplication of input pixel with 0.5 is just a shift and addition operation. So multipliers are not needed during transform using the proposed matrix. The matrix is shown in (5) below:

$$T = \frac{D}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 & -0.5 & 0.5 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0.5 & 0 & 0 & -0.5 & -0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

where, $D = diag(1,\sqrt{2},2\sqrt{2/5},2,1,\sqrt{2},2\sqrt{2/5},2)$. The above transform matrix can be represented as:

$$T = \hat{D} \times \hat{T} \tag{6}$$

where $\hat{D} = D / 2\sqrt{2}$ and,

$$\hat{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 & -0.5 & 0.5 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0.5 & 0 & 0 & -0.5 & -0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

It can be seen that the matrix T satisfies the property of orthogonality, i.e. $T^{-1} = T^{t}$ where t denotes the transpose operation. So we can use the same matrix for image encoding and image decoding.
Let X be an 8×8 block of image data and Y be its corresponding matrix in transformed domain. Then the forward transform operation will be

$$Y = TXT^{t} = \hat{D}(\hat{T}X\hat{T}^{t})\hat{D}^{t}. \tag{8}$$

Since T is orthogonal, we can reconstruct the image by using the reverse transform given as:

$$X = T^{t}YT = \hat{T}^{t}(\hat{D}^{t}Y\hat{D})\hat{T}. \tag{9}$$

IV. Algorithm for Fast Computation

The proposed sparse matrix can decomposed into three sparse matrices as:

$$\hat{T} = \hat{T}_3 \times \hat{T}_2 \times \hat{T}_1, \tag{10}$$

where, $\hat{T}_1, \hat{T}_2$ and $\hat{T}_3$ are given by:

$$\hat{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix},$$

$$\hat{T}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix},$$

$$\hat{T}_3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The fast computation of the proposed transform matrix requires only 14 additions and two shift operations. Whereas, DCT [7] needs 18 additions and 2 shift operations. The SDCT [6] matrix requires 24 additions only. Therefore, the proposed matrix has 20% and 45% saving in computation than DCT [7] and SDCT respectively. It has also in-place computation capability.

V. Application to JPEG Image Compression

We will apply the above transform matrix in a standard JPEG baseline encoder which is shown in Fig.1. Since the quantization operation is applied after transformation using proposed matrix, the diagonal term of the matrix $\hat{D}$ from (7) can be merge into the quantizer. This is done so that $\hat{T}$ is the only source of computation in the transformation stage.

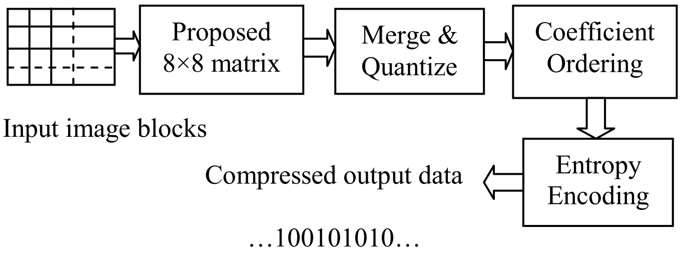


Fig. 1. Implementation of proposed transform matrix on the 8×8 blocks of image data.

VI. Simulation Results and Discussion

To evaluate the performance of the proposed transform matrix, Peak signal to noise ratio (PSNR) and percentage error energy norm (PEEN) are the metrics used for comparison. They are expresses as:

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

$$\text{where} \quad MSE = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(X_{ij} - X'_{ij})^2 \qquad (11)$$

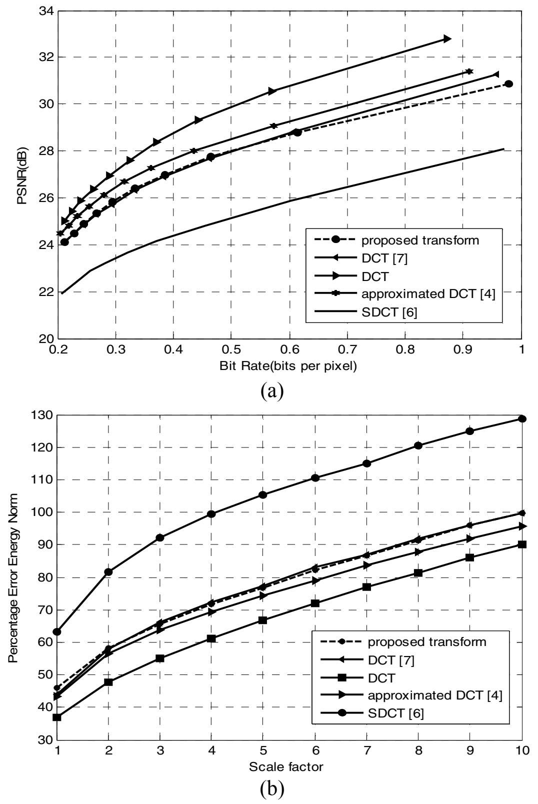
$$PEEN = \sqrt{\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(X_{ij} - X'_{ij})^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}X_{ij}^{\ 2}}} \times 100. \qquad (12)$$

Symbols X_{ij} and X'_{ij} are original and reconstructed pixel - values at the location (i, j) respectively. $M \times N$ is the size of the image.

The standard test images used in our experiment are Lena and Cameraman, each of 256×256 dimensions. The ‘Rate-distortion’ (PSNR vs. Bit rate) and ‘PEEN vs. Scale factor’ plot of Lena image are shown in Fig. 2 (a) and (b). The parameter ‘Scale factor’ is a multiplication term that is used with quantization table to control the bit rate in JPEG. Similarly the ‘Rate-distortion’ and ‘PEEN vs. Scale factor’ plot of Cameraman image is shown in Fig. 3 (a) and (b). Comparison is made between proposed 8×8 transform, DCT [7], DCT, approximated DCT [4], and SDCT [6] for both the images. It has been shown from Fig. 2 and Fig. 3, PSNR and PEEN of proposed transform outperforms SDCT over all bit rates. It follows closely with DCT [7].

The sparseness of the proposed matrix seems to be seriously affecting compression performance for higher bit rates (lower compression) on both the images. For instance, at bit rates greater than 0.3 bpp a PSNR reduction is almost 0.5-1 dB with respect to approximate DCT [4] on both images. Fig 4 (a) and (b) shows the reconstructed images of Lena and Cameraman images at a scale factor 8 (0.2 bpp approx) using the proposed transform matrix. Generally, at low bitrates low complexity transforms are preferable.

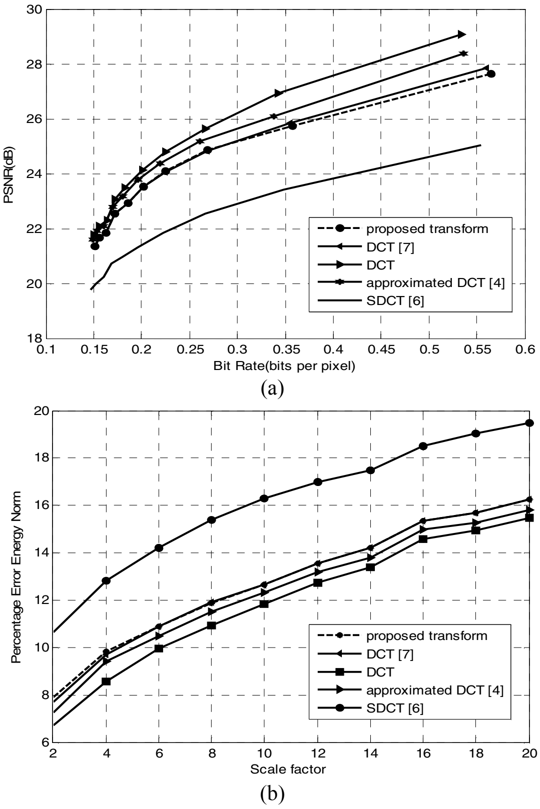
Wavelet kinds of methods have several advantages over DCT based methods at the cost of computational complexity. For example, state-of-the-art image compression algorithms such as EZW [11], SPIHT [12], and SPECK [13] have 1-2db PSNR gain over conventional DCT for a wide range of bit rates, absence of blocking artifacts, progressive transmission capabilities, precise rate control etc. As our goal is to have low complexity compression, the proposed matrix can be applicable to low power and portable devices such as PDAs, digital camera and mobile phones, where conventional DCT transform or wavelet coding algorithms are not suitable. The conventional DCT whose basis elements are floating point numbers needs extensive computations. Therefore, the proposed transform matrix can be an alternative to DCT for these devices owing to its sparseness and integer elements.



(a)

(b)

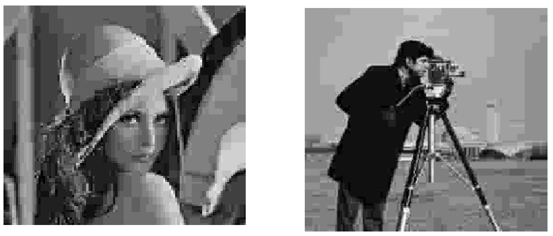
Fig. 2. (a) Rate-Distortion plot and (b) PEEN vs. Scale factor plot of Lena image.



(a)

(b)

Fig. 3. (a) Rate-Distortion plot and (b) PEEN vs. Scale factor plot of Cameraman image.



(a) (b)

Fig. 4. Reconstructed images at a 'Scale factor' of 8 (bit rate is 0.2 bits per pixel approx.) using proposed transform matrix. (a) Lena, (b) Cameraman.

VII. Conclusions

In this paper, we proposed an orthogonal sparse transform for image compression application. A fast algorithm for computation is also developed. The proposed 8x8 transform matrix needs 14 additions and 2 shift operations, which is 20% lesser and 45% lesser computations for transformation than DCT [7] and SDCT algorithms respectively. It is also demonstrated that the reconstructed image quality of the proposed algorithm is comparable with other integer cosine transform algorithms at low bitrates. The basis of the proposed algorithm is based on integers, and made sufficiently sparse (keeping in view of subjective and objective performances), so that the transformation will be faster than floating point DCT and other integer based DCT. It can be very much suitable for fast VLSI implementation.

References

[1] W.B.Pennebaker and J.L.Mitchell, "JPEG Still Image Compression Standard", *Chapman & Hall*, New York, 1993.

[2] Cham, W.K, "Development of integer cosine transforms by the principle of dyadic symmetry", *IEE Proc, I, Commun. Speech Vis.*, 1989, 136, (8), pp. 276-282.

[3] Dimitrov. V, Wahid. K, and Jullien. G, "Multiplication free 8x8 2D DCT architecture using algebraic integer encoding", *Electron. Lett.*, 2004, 40, (9), pp. 1310-1311.

[4] Lengwehasatit K, and Ortega A, "Scalable variable complexity approximate forward DCT", *IEEE Trans. Circuits Syst. Video Technology.*, 2004, 14, (11), pp. 1236-1248.

[5] Tran T.D., "The BinDCT: Fast multiplierless approximation of the DCT", *IEEE Signal Process. Lett.*, 2000, 7, (6), pp. 141-144.

[6] Haweel T.I., "A new squre wave transform based on the DCT", *Signal Process.*, 2001, 81, pp. 2309-2319.

[7] S.Bouguezel, M.O.Ahmad and M.N.S.Swamy., "A low complexity 8x8 transform for image compression", *IEEE Electronic Lett.*, 9th October 2008, vol. 44, no. 21.

[8] K.G.Beauchamp, Application of Walsh and related functions, *Academic press*, New York, 1984.

[9] J.Pendar, D.Cover., "New squre wave transform for signal processing, *IEEE Trans. Signal Process*, 40, August 1992, pp. 2095-2097.

[10] T.I.Haweel, A.M. Alhasan, A simplified squre wave transform for signal prosessing", *Contemp. Math*. 190 (1995).

[11] Shapiro JM, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Transaction on Image Processing*, 1993, 41(12): 3445-3462.

[12] Said A, Pearlman W A, "A new fast efficient image codec based on set partitioning on hierarchical trees", *IEEE Transaction on circuits and systems for video technology*, Jun 1996, vol. 6, pp. 243-250.

[13] W A Pearlman, A Islam, N Nagaraj, A Said, "Efficient low complexity image coding using with a set partitioning embedded block coder", *IEEE Transactions on circuits systems and video technology*, 14(11), 2004, pp. 1219-1235.