

# The BinDCT: Fast Multiplierless Approximation of the DCT

Trac D. Tran, *Member, IEEE*

**Abstract**—This paper presents a family of fast biorthogonal block transforms called binDCT that can be implemented using only shift and add operations. The transform is based on a VLSI-friendly lattice structure that robustly enforces both linear phase and perfect reconstruction properties. The lattice coefficients are parameterized as a series of dyadic lifting steps providing fast, efficient, in-place computation of the transform coefficients as well as the ability to map integers to integers. The new  $8 \times 8$  transforms all approximate the popular  $8 \times 8$  DCT closely, attaining a coding gain range of 8.77–8.82 dB, despite requiring as low as 14 shifts and 31 additions per eight input samples. Application of the binDCT in both lossy and lossless image coding yields very competitive results compared to the performance of the original floating-point DCT.

**Index Terms**—BinDCT, DCT, fast multiplierless transforms, JPEG, MPEG.

## I. INTRODUCTION

**B**LOCK transforms have long found application in image and video coding. The current image compression standard JPEG [1] as well as many high-performance video coding standards such as MPEG and H.263 all employ the  $8 \times 8$  discrete cosine transform (DCT) at its transformation stage. From an energy compaction standpoint, the DCT is a robust approximation to the optimal discrete-time *Karhunen–Loève transform* (KLT) of a first-order Gauss–Markov process with a positive correlation coefficient  $\rho$  when  $\rho \rightarrow 1$  [2]. Since the KLT is signal-dependent, and therefore, computationally complex and expensive, the DCT has proven to be a much better alternative in practice. It is signal independent and has linear phase, real coefficients, and fast algorithms.

Exploiting the symmetry of the basis functions, the DCT transform matrix  $P$  can be factored into a series of  $\pm 1$  butterflies and rotation angles as illustrated in Fig. 1. This factorization results in one of the fastest DCT implementation known up to date [2]: eight coefficients  $X[i]$  can be computed using 13 multiplications and 29 additions. However, the DCT is a floating-point transform. It cannot map integers to integers losslessly. More importantly, floating-point implementations in hardware are slow, require more space, and consume more power. Several integer-friendly approximations of the DCT have been proposed in the past [3]. In this paper, we present a

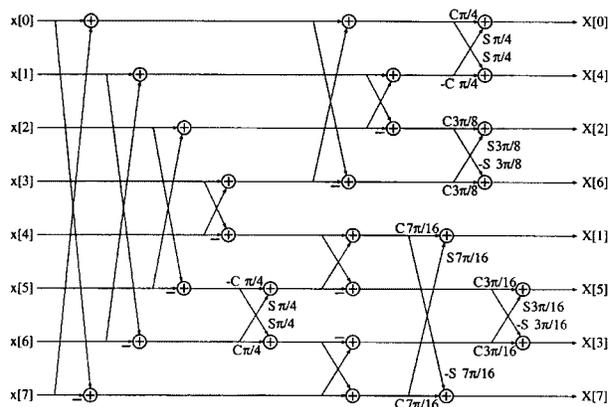


Fig. 1. Forward fast floating-point DCT.

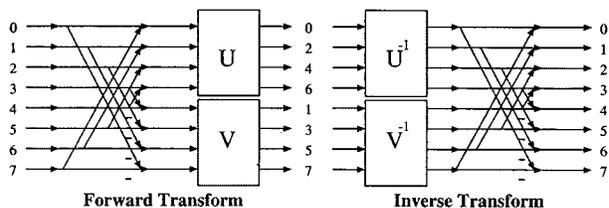


Fig. 2. General solution for perfect reconstruction block transform with symmetric/antisymmetric basis functions.

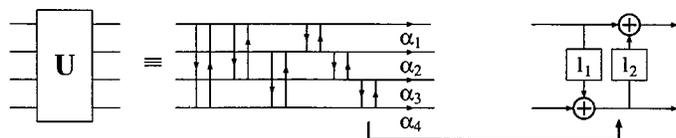


Fig. 3. Parameterization of an invertible matrix via the lifting steps and scaling factors.

novel invertible ladder-based block transform that is almost as efficient as the DCT in energy compaction. Moreover, the new transform not only possesses the integer mapping capability but also has dyadic-rational coefficients that lead to an elegant implementation utilizing only shift-and-add operations.

## II. GENERAL SOLUTION

From a filter bank standpoint, the  $M \times M$  DCT is the most basic  $M$ -channel linear phase paraunitary filter bank (LPPUFB). All  $M$  filters have the same length  $M$ . Its polyphasematrix has order zero (independent of  $z$ ) and can be written in the following form:

$$E_0 = \begin{bmatrix} U & UJ \\ VJ & -V \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I & J \\ J & -I \end{bmatrix}. \quad (1)$$

Manuscript received October 15, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. A. M. Sayeed.

The author is with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218-2686 USA (e-mail: trac@jhu.edu).

Publisher Item Identifier S 1070-9908(00)05100-2.

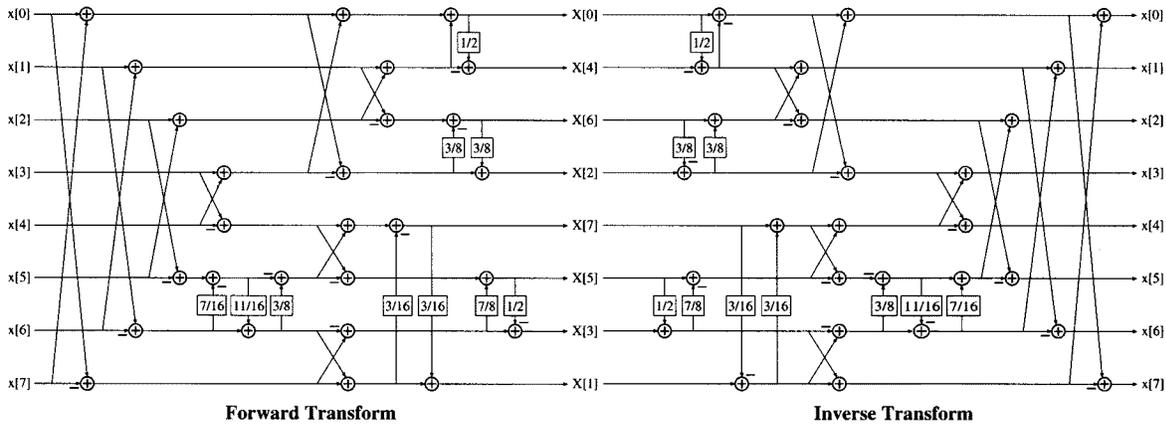


Fig. 4. BinDCT, version A.

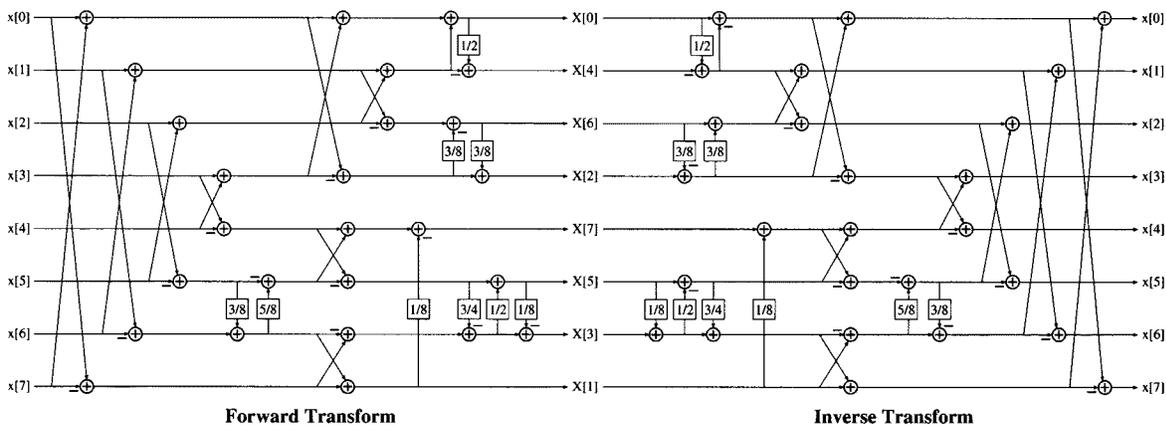


Fig. 5. BinDCT, version B.

It is clear that  $\mathbf{E}_0$  is orthogonal if and only if  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal. For  $\mathbf{E}_0$  to represent the DCT, we need two special orthogonal matrices. However, any choice of orthogonal  $\mathbf{U}$  and  $\mathbf{V}$  does result in an  $M$ -channel  $M$ -tap LPPUFB. In the more general biorthogonal case,  $\mathbf{E}_0$  must be invertible. From the factorization above, it is clear that  $\mathbf{E}_0$  is invertible if and only if  $\mathbf{U}$  and  $\mathbf{V}$  are invertible. In other words, the factorization in (1) covers the complete class of all  $M$ -channel  $M$ -tap biorthogonal LPPB's as long as the matrices  $\mathbf{U}$  and  $\mathbf{V}$  are invertible. The general structure is depicted in Fig. 2.

The challenge now is how to characterize these invertible  $\mathbf{U}$  and  $\mathbf{V}$  matrices using the fewest number of independent parameters. It is well known that every  $N \times N$  orthogonal matrix can be factored into  $N(N-1)/2$  rotations. We can establish a similar result for invertible matrices: any  $N \times N$  invertible matrix can be completely characterized by  $N(N-1)$  shears (also known as *lifting steps* or *ladder structures*),  $N$  diagonal scaling factors, and possibly a permutation matrix. It is not too difficult to see how one can systematically factor any invertible matrix using permutation, diagonal scaling, and shearing. This is simply the Gauss-Jordan elimination process or the LU factorization.

The parameterization of an arbitrary invertible matrix is illustrated in Fig. 3 (drawn for  $N = 4$ ). Back to our general  $M$ -channel biorthogonal block transform with linear phase basis functions, the transform can be proven to consist of

$((M^2/2) - M)$  lifting steps,  $l_i$ , and  $M$  diagonal scaling factors  $\alpha_i$ . Typically, these scaling factors can be folded into the quantization stepsizes of the encoder.

### III. MULTIPLIERLESS SOLUTIONS

The lifting scheme offers versatility in constructing fast transforms that can map integers to integers by placing a *floor* (or *round*, or *ceiling*) in each lifting step. If the lifting step is chosen to be dyadic [i.e., a rational that can be written in the form of  $(k/2^m)$ ;  $k, m \in \mathcal{Z}$ ], the nonlinear operation can be incorporated into the division using binary bit shift. Division by  $1/2^m$ , followed by a truncation, is equivalent to a binary shift by  $m$  places. Thus, multiplierless filter banks can be easily constructed using this method.

From the fast DCT implementation in Fig. 1, it is clear that the integer-unfriendly components of the DCT are its rotation angles  $\{(\pi/4), (3\pi/8), (7\pi/16), (3\pi/16)\}$ . To construct integer transforms, we approximate the DCT's rotation angles by appropriate dyadic lifting steps, as discussed in the previous section. Unconstrained optimization can be used to search for the optimal lifting parameters from which dyadic ones are approximated. Three different fast versions of such multiplierless block transforms are presented in Figs. 4–6. The forward and inverse transform coefficients of the simplest version (C) are tabulated in Table I. All three transforms approximate the DCT closely,

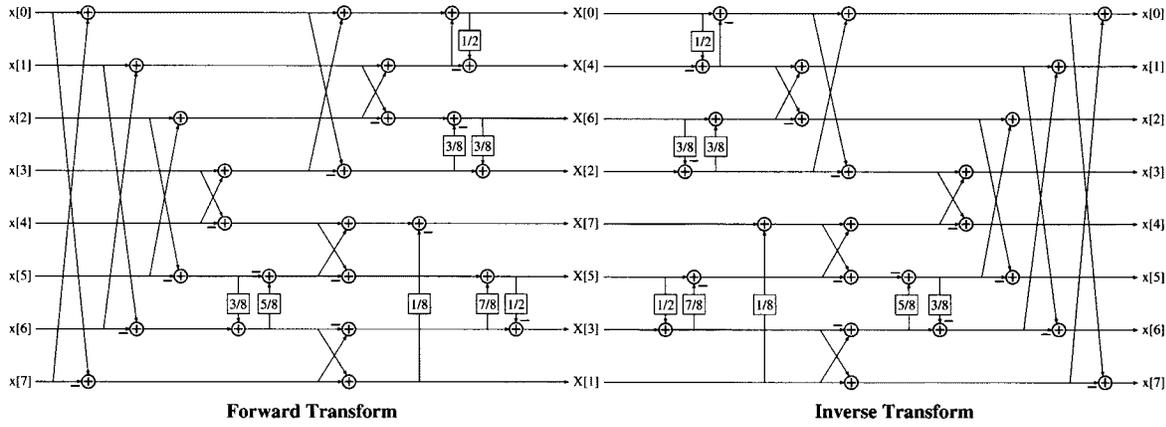


Fig. 6. BinDCT, version C.

TABLE I  
FORWARD TRANSFORM MATRIX  $P$  AND  
THE INVERSE TRANSFORM MATRIX  $G$  OF THE binDCT-C

Forward Transform Matrix							
1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2
1/2	1/2	3/16	0	0	-3/16	-1/2	-1/2
55/128	3/16	-3/16	-55/128	-55/128	-3/16	3/16	55/128
9/32	-1/8	-19/64	-1/4	1/4	19/64	1/8	-9/32
1/4	-1/4	-1/4	1/4	1/4	-1/4	-1/4	1/4
7/16	-3/4	7/32	1/2	-1/2	-7/32	3/4	-7/16
-3/16	1/2	-1/2	3/16	3/16	-1/2	1/2	-3/16
-1/16	1/4	-13/32	1/2	-1/2	13/32	-1/4	1/16
Inverse Transform Matrix							
1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/4
1/2	13/32	1/4	1/16	-1/16	-1/4	-13/32	-1/2
1/2	3/16	-3/16	-1/2	-1/2	-3/16	3/16	1/2
1/2	-7/32	-3/4	-7/16	7/16	3/4	7/32	-1/2
1/2	-1/2	-1/2	1/2	1/2	-1/2	-1/2	1/2
1/4	-19/64	1/8	9/32	-9/32	-1/8	19/64	-1/4
-3/16	55/128	-55/128	3/16	3/16	-55/128	55/128	-3/16
0	3/16	-1/2	1/2	-1/2	1/2	-3/16	0

TABLE II  
COMPARISON OF TRANSFORM COMPLEXITY (OPERATIONS NEEDED PER  
EIGHT TRANSFORM COEFFICIENTS) AND CODING GAIN [WITH  $AR(1)$   
IMAGE MODEL  $\rho = 0.95$ ]

Transform	No. of Multiplications	No. of Additions	No. of Shifts	Coding Gain
8 × 8 WHT	0	24	0	7.95 dB
8 × 8 DCT	13	29	0	8.83 dB
8 × 8 binDCT-A	0	36	19	8.82 dB
8 × 8 binDCT-B	0	31	14	8.77 dB
8 × 8 binDCT-C	0	30	13	8.77 dB

TABLE III  
OBJECTIVE CODING RESULT COMPARISON (PSNR IN dB)

Comp. Ratio	Lena		Goldhill		Barbara	
	DCT	binDCT-B	DCT	binDCT-B	DCT	binDCT-B
Lossless	-	4.565 bpp	-	5.092 bpp	-	4.994 bpp
1:8	39.91	39.44	36.25	35.82	36.31	35.71
1:16	36.38	36.02	32.76	32.45	31.11	30.67
1:32	32.90	32.74	30.07	29.87	27.28	26.90
1:64	29.67	29.61	27.93	27.79	24.58	24.24

yet they can be implemented using simple binary arithmetic. Hence, they are named the binDCT. The energy compaction of the new transforms ranges from 8.77–8.82 dB coding gain for  $AR(1)$  image model with  $\rho = 0.95$  (the DCT achieves 8.83 dB). Eight transform coefficients can be computed using as low as 13 shifts and 30 additions (version C). Table II compares the complexity and performance of the binDCT's. The Walsh–Hadamard transform and the DCT serve as comparison benchmarks.

IV. APPLICATION IN IMAGE CODING

The two novel transforms are evaluated in an image coding experiment where the floating-point implementation of the DCT provides the benchmark. To encode the transform coefficients, we use a progressive transmission zerotree coder, where each block of transform coefficients is treated analogously to a wavelet tree, as described in [4]. The DC subband can be further decomposed using several wavelet iterations. Integer wavelets can be employed to keep the transform stage entirely integer-based.

The objective coding results are presented in Table III. The binDCT is only around 0.1–0.5 dB below the DCT in the peak

SNR's of the reconstructed images. In terms of visual image quality, the binDCT offers similar performances to that of the DCT.

V. CONCLUSIONS

Compared to the DCT, the novel block transform described in this paper offers numerous advantages.

- 1) The binDCT has a fast, elegant implementation utilizing only shift-and-add operations. No multiplication is needed. Eight transform coefficients can be computed using as low as 13 bit shifts and 30 additions.
- 2) The binDCT can map integers to integers with exact reconstruction. This property is pivotal in transform-based lossless coding and allows a unifying lossy/lossless coding framework.
- 3) In our software implementation, the binDCT is already three times faster than the floating-point DCT. Much higher speed is expected in hardware implementations.
- 4) The multiplierless property of the binDCT allows efficient VLSI implementations in terms of both chip area and power consumption.

- 5) The binDCT approximates the DCT very closely. Perceptual quantization matrices and coding strategies designed specifically for the DCT can be applied to the binDCT immediately without any modification.
- 6) The binDCT has reasonably high coding performances. Generalizing the concept to longer filter lengths (resulting in lapped transforms) is also relatively straightforward [5].

#### REFERENCES

- [1] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*. New York: Van Nostrand Reinhold, 1993.
- [2] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic, 1990.
- [3] W.-K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *Proc. Inst. Elect. Eng., Part I*, vol. 136, pp. 276–282, Aug. 1989.
- [4] T. D. Tran and T. Q. Nguyen, "A progressive transmission image coder using linear phase filter banks as block transforms," *IEEE Trans. Image Processing*, vol. 8, pp. 1493–1507, Nov. 1999.
- [5] T. D. Tran, " $M$ -channel linear phase perfect reconstruction filter bank with integer coefficients," in *SPIE Int. Symp. Wavelet Applications in Signal and Image Processing*, Denver, CO, July 1999.
- [6] —, "Fast multiplierless approximation of the DCT," in *33rd Annu. Conf. Information Sciences and Systems*, Baltimore, MD, Mar. 1999, pp. 933–938.
- [7] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Applicat.*, vol. 4, pp. 247–269, 1998.