

# Fast directional discrete cosine transform for image compression

**Bo Chen, Hongxia Wang, and Lizhi Cheng**  
 National University of Defense Technology, Department of Mathematics and Systems Science Deya Road, Changsha, Hunan, 410073, China  
 E-mail: chenbo0354@yahoo.com.cn

**Abstract.** A fast directional discrete cosine transform (FD-DCT) is proposed for efficient representation of anisotropic edges in images. The transform is performed on the pre-defined direction lines similar to the intraprediction mode in H.264. Comparing to the directional discrete cosine transform (DDCT) now available, no interpolation is needed in FDDCT; thus, the amount of computation decreases by 80%. Simulation results indicate that the peak signal-to-noise ratios of images compressed using FDDCT are >1 db higher than those using DDCT. © 2010 Society of Photo-Optical Instrumentation Engineers.  
 [DOI: 10.1117/1.3306634]

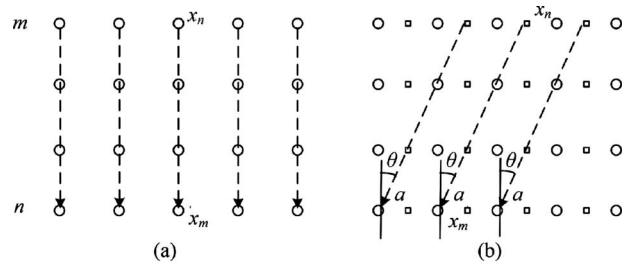
Subject terms: image compression; sparse representation; fast directional transform; discrete cosine transform.

Paper 090584LRR received Jul. 31, 2009; revised manuscript received Dec. 20, 2009; accepted for publication Dec. 22, 2009; published online Feb. 18, 2010.

## 1 Introduction

Discrete cosine transform (DCT) and discrete wavelet transform (DWT) used in image compression are implemented by separable 1-D transforms in the rows and columns of images. The major shortcoming of the separable transform is that it cannot represent the anisotropic edges in the image sparsely. Recently, various lifting-based directional transforms<sup>1-3</sup> are proposed to overcome the problem. In the traditional lifting scheme, the transform matrix is factorized into elementary matrices, which correspond to the operation of adding a multiple of a row/column to another row/column, and the augend and the addend must be in the same column/row.<sup>4,5</sup> While in the directional lifting transform, lifting steps can be performed along edges with any directions; thus, it can approximate edges better and represent images more sparsely (see Fig. 1).

However, it is noted that to transform along edges with arbitrary directions, the fractional pixels are used in adaptive directional lifting and their values are interpolated from neighboring integer pixels. Denote the number of taps of the interpolation filter as  $L$ , then the computation of direction transform is about  $L$  times higher than the original one.<sup>1,2</sup> The high computation cost is a serious drawback in practice, especially for DCT, which has various faster approximations<sup>6</sup> that are quite suitable to portable devices and real-time processing, such as satellite onboard image compression. In this letter, we proposed a fast directional DCT (FDDCT) without interpolation. Compared to the original lifting-based directional DCT (DDCT),<sup>2</sup> the com-



**Fig. 1** Exemplified elementary matrix operation: (a) nondirectional and (b) directional, where the circles are the pixels and squares are half-pixels.

putation of FDDCT decreases by 80%, and the peak signal-to-noise ratios (PSNR) of compressed images are higher.

## 2 Proposed FDDCT

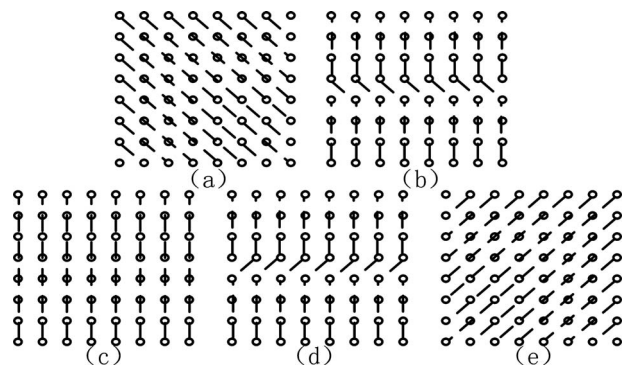
### 2.1 Direction Modes without Interpolation

It is known that, in DDCT, interpolation on the fractional pixels is the main reason that causes large amounts of computation. If we can use only the integer pixels to approximate edges, then interpolation will be avoided. In this paper, we propose to use five direction modes to achieve this. These modes are shown in Fig. 2, where no fractional pixels exist and the size of transform matrix is  $8 \times 8$ . The corresponding directional angles are from  $-45$  deg [mode (a), see Fig. 2(a)] to  $45$  deg [mode (e), see Fig. 2(e)].

These direction modes are like the prediction modes used in the intraprediction process in the recent video-coding standard H.264.<sup>7</sup> For example, modes (a) and (b) correspond to the diagonal down-right and the vertical-right prediction in H.264, separately, and the only difference is that polylines are used to avoid interpolating in our modes. It also find that five modes are enough to approximate well most directions of edges in images.<sup>1,2</sup> Supposing that the direction modes of all blocks are selected, the transform can be performed along the direction lines defined in the direction modes (see Fig. 3), and the pixels on the direction line are all integer ones.

### 2.2 Lifting Steps across Adjacent Blocks

Because the direction modes are defined on the individual image blocks, there may exist direction lines consisting of  $<8$  integer pixels, where eight-point fast DCT cannot be



**Fig. 2** Five direction modes in row FDDCT. The circles are integer pixels, and the dashed lines are direction lines.

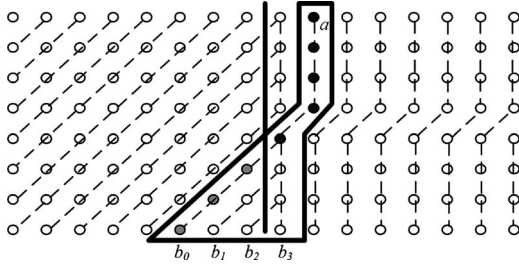


Fig. 3 Direction line across blocks.

performed. Extending the short direction line to the adjacent block is a natural way to solve the problem. But we find that, if the direction modes of adjacent blocks are different, then some pixels will be occupied by more than one direction lines (see Fig. 3). That is to say, lifting steps in several DCTs will modify the value of these pixels simultaneously. Thus, the DCTs on these direction lines across blocks must be modified to make the transform perfect reconstruction.

It is known that DCT can be factored into lifting steps,<sup>6</sup> and the DCT matrix can be expressed as

$$\mathbf{T} = \mathbf{T}_{i_0 j_0}(c_0) \mathbf{T}_{i_1 j_1}(c_1) \dots \mathbf{T}_{i_n j_n}(c_n),$$

$$\mathbf{T}_{i,j}(c) = \mathbf{I} + c \mathbf{e}_i \mathbf{e}_j^T, \quad (1)$$

where  $\mathbf{I}$  is the identity matrix of size  $M$ ,  $\mathbf{e}_n$  is the  $n$ 'th unit vector in  $M$ -dimensional Euclidean space and  $c$ , a real number, is the lifting coefficient. Thus, the lifting matrix

$\mathbf{T}_{i,j}(c)$  is a triangular matrix, and in every lifting step only one element in the input vector is modified.

Because all lifting steps are similar, we just think of one lifting procedure. Take the pixels  $a, b_i, i=0, \dots, 3$  in Fig. 3, for example, where  $a$  is occupied by four direction lines. To avoid modifying  $a$  four times with the multiple of  $b_i$ , we change the lifting procedure as

$$a^1 = a + c \sum_{i=0}^3 w_i b_i, \quad i = 0, \dots, 3,$$

$$\sum_{i=0}^3 w_i = 1, \quad w_i = w_j, \quad i \neq j, \quad (2)$$

where  $a^1$  is output and  $w_i$  are the scaling coefficients. The corresponding inverse transform can be expressed as

$$a = a^1 - c \sum_{i=0}^3 w_i b_i, \quad i = 0, \dots, 3. \quad (3)$$

Obviously, it is perfect reconstruction. In the lifting procedure [Eq. (2)], the mean of pixels on different direction lines is used to predict the target pixel, which often approximates the target better than using a single pixel on one direction line. Furthermore, it should be noted that the transform on the extended direction line is performed between adjacent blocks. Therefore, comparing to DCT, which performs on every block separately, FDDCT can partly remove the correlation between blocks.

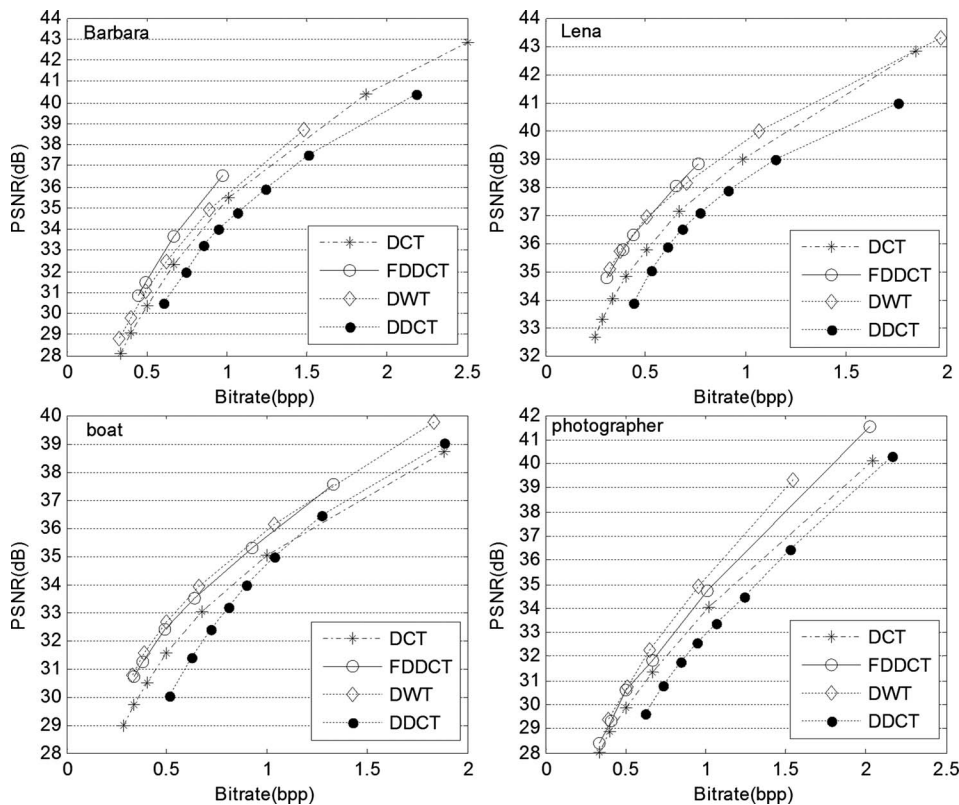
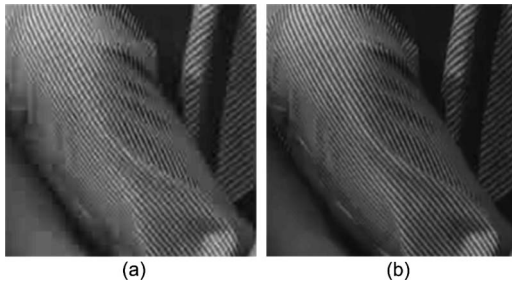


Fig. 4 PSNR versus bit-rate curves of different transforms.



**Fig. 5** Parts of Barbara compressed with DCT and FDDCT (bit rate: 0.5 bits/pixel): (a) is compressed with DCT and (b) is compressed with FDDCT.

At last, it must be reminded that the direction modes of blocks need to be selected before FDDCT. As stated above, the FDDCT coefficients of a block are only influenced by its adjacent blocks, which is similar with DDCT. Thus, the same dynamic programming algorithm in Ref. 2 is used to select the direction modes in FDDCT to minimize the rate-distortion cost.

### 3 Simulation Results

Extensive experiments have been conducted to evaluate the performance of FDDCT. We report the experimental results of four common gray images: Lena, Barbara, Boat, and Photographer, which have different characteristics. FDDCT is compared to DCT and DWT (CDF9/7 used in JPEG2000) and DDCT in our experiments. The set partition coding developed by us in Ref. 8 is used to code the coefficients of FDDCT, DCT, and DWT, which is suitable for both the block transform and the subband transform. The direction mode of a block is first predicted from the coded modes of coded neighboring blocks and then is coded by the arithmetic coding.

Figure 4 shows that PSNRs of images compressed with FDDCT are close to those with DWT. And compared to DCT, the coding gain of FDDCT ranges from 0.4 to 1.5 dB. The compression results of DDCT-based method in Ref. 2 are also given in Fig. 4, whose PSNRs are >1 dB lower than the proposed FDDCT based one. The decompressed Barbara is partly shown in Fig. 5. It shows that the edges in the image compressed by FDDCT are less disturbed, and the block artifacts are nearly eliminated by FDDCT. The reason should be that, in FDDCT, many direction lines are extended to the adjacent blocks; thus, the blocks are not transformed individually. It is helpful to reduce block artifacts in the compressed images.

Finally, we give the computational complexity of FDDCT. If the direction modes of all blocks in the image are same, then no special lifting steps are needed; thus, the computation of FDDCT is the same as that of DCT. But, in practice, different blocks generally have different direction modes. As a result, the computation depends on the edges in the image. In other words, the computation of FDDCT is image dependent.

**Table 1** Executing time (second) of different transforms.

Size	Transform		
	DCT	FDDCT	DWT
512×512	0.006547	0.008672	0.032030
1024×1024	0.030160	0.038900	0.129220
2048×2048	0.131870	0.179953	0.557810

Here, we perform several transforms on many standard gray-scale images on a 2.93-GHz Pentium IV. The average executing times of DCT, DWT, and FDDCT are given in Table 1. It shows that the time of FDDCT is in direct proportion to the size of the image, which is ~1.4 times than that of DCT and one-third of DWT. Comparing to DDCT, whose computation is eight times than that of DCT (eight-tap interpolation is used), the computation decreases by ~80%.

### 4 Conclusion

By choosing the direction modes consisting of integer pixels, a new directional DCT called FDDCT for image compression is proposed in this paper. Because interpolations in DDCT are avoided, the computation of FDDCT decreases ~80% compared to original DDCT. Furthermore, the use of interblock information in FDDCT derives less block artifacts and better rate-distortion performance in the compressed images.

### Acknowledgments

The authors are grateful to the anonymous reviewers for their comments, which have helped us to improve this work. This study is supported by the National Natural Science Foundation of China (Grant No. 10601068).

### References

- W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Trans. Image Process.* **16**(2), 416–427 (2007).
- H. Xu, J. Xu, and F. Wu, "Lifting-based directional DCT-like transform for image coding," *IEEE Trans. Circuits Syst. Video Technol.* **17**(10), 1325–1335 (2007).
- L. Zhang, D. Wang, and A. Vincent, "Adaptive zero-tree structure for curved wavelet image coding," *Opt. Eng.* **45**, 27–30 (2006).
- D. L. Liang, L. Cheng, and Z. H. Zhang, "General construction of wavelet filters via a lifting scheme and its application in image coding," *Opt. Eng.* **42**, 1949–1955 (2003).
- Y.-J. Chen and K. S. Amaratunga, "M-channel lifting factorization of perfect reconstruction filter banks and reversible M-band wavelet transforms," *IEEE Trans. Circuits Syst. Video Technol.* **50**(12), 963–976 (2003).
- J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.* **49**(12), 3032–3044 (2001).
- Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 ISO/IEC 14496-10 (AVC) (Mar. 2005).
- B. Chen, L. Cheng, and H. Wang, "LBT based low complexity image compression method," in *Proc. of Int. Conf. on Pattern Recognition*, Hong Kong, Vol. 1, pp. 941–944, IEEE Computer Society, Hong Kong (2006).