An Overview of Directional Transforms in Image Coding

Jizheng Xu[1], Bing Zeng[2] and Feng Wu[1]
[1] Microsoft Research Asia, Beijing, China
[2] The Hong Kong University of Science and Technology, Hong Kong SAR, China

***Abstract*—Transform-based image coding has been the mainstream for many years, as witnessed in from the early effort in JPEG to the recent advances in HD Photo. Traditionally, a 2-D transform used in image coding is always implemented separately along the vertical and horizontal directions, respectively. However, it is usually true that many image blocks contain oriented structures (e.g., edges) and/or textures that do not follow either the vertical or horizontal direction. The traditional 2-D transform thus may not be the most appropriate one to these image blocks. This well-known fact has recently triggered several attempts towards the development of directional transforms so as to better preserve the directional information in an image block. Some of these directional transforms have been applied in image coding, demonstrating a significant coding gain. This paper presents an overview of these directional transforms as well as a discussion of some existing problems and their potential solutions.**

I. INTRODUCTION

Due to the inhomogeneous property of natural objects and scenes, the correlation of local contents in an image is often directional, i.e., the correlation along one direction is much stronger than those along other directions. On the other hand, human eyes are very sensitive to such directionality, especially for contents that contain strong edges and textures. Thus, efficient representation of directional information is extremely important for high performance image coding.

However, in the exiting popular image coding systems, the representation of directional information is not very efficient. For instance, different transforms are chosen to exploit the correlation within an image, such as DCT in JPEG, wavelets in JPEG 2000, and lapped transform in HD Photo [1]. These transforms have one common property: the 2-D transform is always implemented separately along the vertical and horizontal directions, respectively. Such a separable fashion has significantly reduced the computational complexity when compared with a non-separable 2-D transform, and it also seems to work efficiently to de-correlate the signal when the signal's contents are oriented horizontally or vertically.

Clearly, such an implementation cannot efficiently de-correlate contents that follow neither the horizontal not the vertical direction. The reason is simple: the basis functions of a separable 2-D transform do not support directionalities other than vertical or horizontal. In order to represent such a directional signal, the separable 2-D transform has to use many unidirectional kernels to approximate the signal and thus produce many non-zero high frequency coefficients. The so-called Gibbs artifacts would appear after the quantization on those high frequency coefficients, even when the quantization is very small. More non-zero high frequency coefficients also increase the bit-rate, leading to low coding efficiency.

Although it is not new to develop efficient schemes to better preserve directional information, designing transforms that can accommodate various directionalities has never been an easy task. Basically, the following properties are desired for a good directional transform in image coding:

- Efficient to represent various directional signals. To handle different contents in an image, the transform should be able to exploit the correlations along different directions, and thus the basis vectors should be directional and anisotropic.
- Easy to be implemented. An ordinary image may contain tens of millions of pixels. To handle such a huge number of pixels, the transform should not be complicated.
- Non-redundant. The transformed coefficients should not be redundant. Although the transform itself is not necessary to be a critically-sampled one, it needs a very careful judgment to go over-complete or not.
- Able to take full advantage of exiting coding tools. Image coding has evolved for several decades. Many coding tools are heavily tuned and already very mature. It is very difficult to start from zero to beat the current state-of-the-arts. For example, any new transform that can generate coefficients in a similar structure as those transforms used in the current coding schemes would be highly appreciated.

Recently, several directional transforms have been proposed and applied in image coding [2-13]. Basically, they can be divided into several categories. In the first category, pixels in an image block are re-organized according to a selected direction, and the conventional transforms are then applied [2][3]. More detailed descriptions of these transforms are given in Section II. In the second category, lifting-based approaches are used to change the conventional transforms to be directional [4-10], which will be discussed in details in Section III. In the third category, the directional transform is constructed by a directional prediction and a corresponding data-dependent transform thereafter [11-13], which will be discussed in Section IV. Some concluding remarks are presented in Section V.

II. REORGANIZATION-BASED DIRECTIONAL TRANSFORM

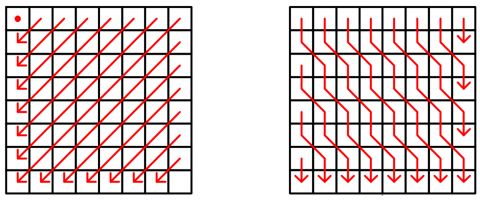
The basic idea on the first category of directional transforms is to reorganize pixels along a certain direction for each 1D transform. The idea is very suitable to block transforms like DCT. Given an image block $\{u(m,n)\}$. Let ψ be the operator that groups some pixels in u to form a 1-D vector so that a 1-D DCT can be carried out on the formed vector. In the next step, another operator ϕ that is complementary (or orthogonal) to ψ takes some transform coefficients obtained in the first step to form a second set of 1-D vectors and then the second 1-D DCT is performed on each vector. In the conventional 2-D DCT, for example, ψ takes each column of u to form a 1-D vector and ϕ takes each row of the transform coefficients after the first 1-D transform.

In general, ψ and ϕ can take data along any direction other than vertical or horizontal, thus producing some directional transforms. Similar to the directional intra-prediction modes defined in H.264, eight directional modes are usually used in directional DCT. Two

978-1-4244-5309-2/10/$26.00

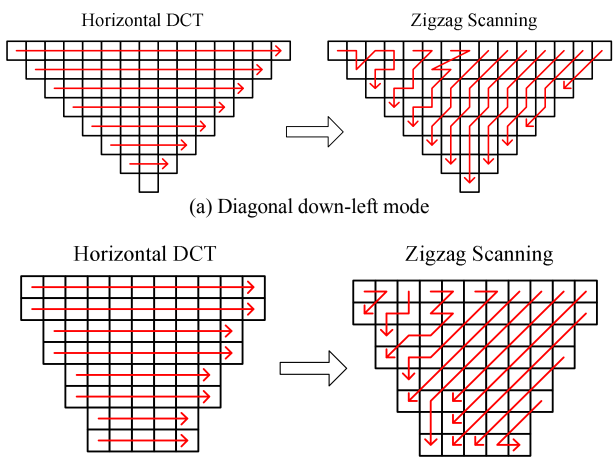
modes are the same as the conventional 2-D DCT. Other modes cover diagonal down-left, diagonal down-right, vertical-right, horizontal-down, vertical-left, and horizontal-up; whereas the DC mode in H.264 is not used here. The modes of the diagonal down-left and vertical-right are depicted in Fig. 1 to explain how the directional DCT works. The other directional modes are processed similarly.

Clearly, ψ in the diagonal down-left mode takes pixels in $\boldsymbol{u}$ along the diagonal down-left direction so that the first 1-D DCT is a directional one along the diagonal direction. All coefficients after the first 1-D DCT's are put as column vectors with the DC component placed at top, followed by the first AC component and so on, see the left part of Fig. 2(a). Then, ϕ groups data into vectors along the horizontal direction to facilitate the second 1-D DCT. As another modification, the zigzag scanning order for each of those modes needs to be redesigned, according to the statistical variances of individual coefficients (obtained over a large set of test images). The right part of Fig. 2(a) shows the new zigzag ordering that will be used in the diagonal down-left mode. Similarly, the right part of Fig. 1 and Fig. 2(b) together show the corresponding steps in the vertical-right mode.



Diagonal down-left　　Vertical-right

Fig. 1. The first 1-D DCT along the specific direction of diagonal down-left and vertical-right (the block size is 8×8).



(a) Diagonal down-left mode

(b) Vertical-right mode

Fig. 2. The second 1-D DCT is implemented along the horizontal direction; whereas the modified zigzag scanning order is shown.

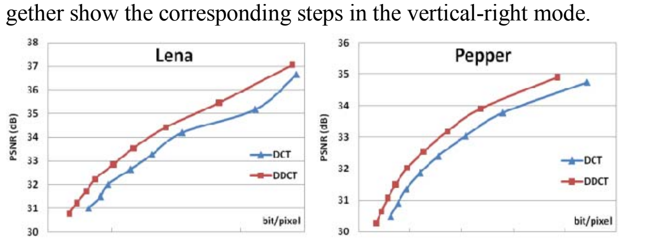


Fig. 3. R-D performances of directional DCT in comparison with JPEG.

For each block, the directional mode is selected in terms of rate-distortion cost similar to H.264. The experimental results from [2] are shown in Fig. 3, which the JPEG's quantization and VLC table are adopted. It is clear that directional DCT indeed produces a remarkable coding gain over the popular JPEG. Recently, directional DCT has been reported to be integrated into the intra coding of H.264 [3]. Similar performance gain has been achieved in the scheme too. The performance upper bound for 2-D directional sources with block-based transforms is derived in [17], which justifies the improvement in practical image coding.

III. LIFTING-BASED DIRECTIONAL TRANSFORMS

There are various transforms that can be factorized into lifting operators. In this section, we will show that those transforms can be turned into corresponding directional transforms by performing each lifting operator directionally. The lifting-based implementation of the CDF 5/3 wavelet for 1-D signals is illustrated in Fig. 4. The first lifting operator uses the even-numbered signals to predict the odd-numbered ones. The prediction residues are the high-pass coefficients. Then, the high-pass coefficients are updated to the odd-numbered signals to generate the low-pass coefficients. The inverse transform can be implemented by reversing the order of lifting operators and the tap of each operator.

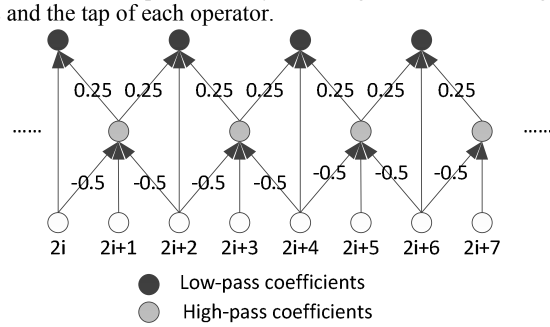


Fig. 4. Lifting-based implementation of the CDF 5/3 wavelet.

For a 2D signals, to perform a 2-D wavelet transform, conventionally a vertical 1-D wavelet and a horizontal one are applied sequentially. This equals to perform lifting steps vertically and horizontally. Rather than to perform the whole transform directionally, it is much easier to perform each lifting step instead. Thus, the basic idea of constructing a directional wavelet transform is to turn all lifting operators that compose the original wavelet transform to be directional. Fig. 5 illustrates a directional lifting operator compared to a conventional one. In both lifting operators, the immediately upper and lower rows of signals are used to predict/update the current row of signals. The difference is that in the directional one, the correspondences are found following a given direction θ. For example, when θ=45°, for the current pixel, the correspondence on the upper row is the upper-right pixel. For a prediction operator, when the correlation is strong along some direction, to perform the prediction along that direction will yield residue with much less energy. Thus, more compact energy can be achieved, which is preferable in image coding. It is analogical to the motion compensation procedure in video coding if we consider each row of signals as a video frame. When along the direction we cannot find a signal but somewhere between two signals, as what is often done in motion compensation, an interpolation is utilized to generate the signal located at the place [4][5]. The other approach to deal with such a case is to find a correspondence on other rows along the direction if a signal can be found in the end [6]. Such an approach may be more efficient when the correlation is very anisotropic.

The inverse of a directional lifting operator is also a directional operator, which is along the same direction but with the reverse lifting weight. Concatenating all the inverse lifting operators together, the inverse directional transform can be exactly generated. Similarly, the horizontal transform of a 2D wavelet can also be performed directionally. Since each lifting operator is simple, such a divide-and-conquer strategy makes it possible to generate directional transforms, even for some complex wavelets.

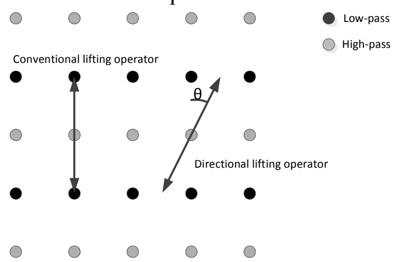


Fig. 5. Illustration of a directional lifting operator.

Based on the lifting-based approach, directional DCT [7] and directional lapped transforms [8] are also proposed to bring directional basis functions into those transforms. These directional functions are generated along the given direction that each lifting operator follows. On the other hand, these directional basis functions still preserve the frequency properties of the DCT and lapped transforms. Thus, directional DCT and directional lapped transforms are a direction-frequency analysis of the image signals.

In practical coding using lifting-based directional transforms [3-8], the image to be coded is divided into blocks. For each block, one direction from a set of predefined directions is assign and the transform will be along that direction within the block. The reason to do so is because the encoder has to notify the decoder about the directions in every region of the image. Limiting the number and spatial granularity of the directions can reach an optimal coding performance by trading off the adaptivity of the transform to represent the whole image and the number of bits of the direction notification overhead.

Lifting-based directional transforms have several advantages. Firstly, they preserve the advantages of lifting-based implementation of transforms. The lifting-based implementation of a transform is often also a fast implementation scheme of that transform. Lifting-based implementations do not require much memory thanks to in-place calculation of lifting-based approach. Secondly, the flexibility of lifting-based approaches makes it easy to handle boundaries and even different subsampling patterns. For example, the authors of [5] present directional wavelet transforms using quincunx subsampling. In [4], variable block size can be combined with the directional wavelet to provide more adaptive transform in coding, thanks to the flexibility. Thirdly, theoretically the direction θ can be arbitrary in the directional transform, which allows very precise match between the transform and the correlation of the signals. Fourthly, the directional transforms have also the same coefficient structures as the corresponding conventional transforms. Thus, they can be easily integrated into existing coding systems. For example, directional wavelet transforms can replace the conventional ones in JPEG2000 [3-6]. The directional DCT can be integrated into JPEG [7] and directional lapped transforms can improve HD Photo [8]. Fifthly, compared to reorganization-based directional transforms, lifting-based directional transforms can be done continuously and across image block boundaries. Thus, they can more easily remove the inter-block redundancy and reduce blocking effects effectively. But this point also brings a disadvantage. The complexity of the direction decision is rather high to achieve optimal since the transforms on different blocks are not independent. The optimization problem of direction decision can be approximately solved using dynamic programming [7][8]. However, it is still more complex than the direction decision procedure in the case that each block has an independent transform, e.g., reorganization-based directional transforms described in Section II. Cross-block operations also place constraints on the directional transform. For example, neighboring blocks may not choose much different directions and subsampling patterns in directional wavelet transforms for the above reason.

As an illustration for the effectiveness of transforms in this category, the coding performance comparisons with image Foreman and Barbara between the conventional CDF 5/3 wavelet in JPEG2000 and the corresponding directional wavelet are shown in Fig. 6 [4]. Over 2dB's gain has been observed with these natural images. Recently, improvements on the transforms are also presented by training the interpolation filters for the sub-pixel position directions and the weights of the wavelet transforms according to the local statistics of the image [9][10]. Such improvements can offer further coding performance gain up to 0.98dB [10]. The lifting-based directional transform design is also introduced into H.264 [13], which also significantly improves the coding performance.

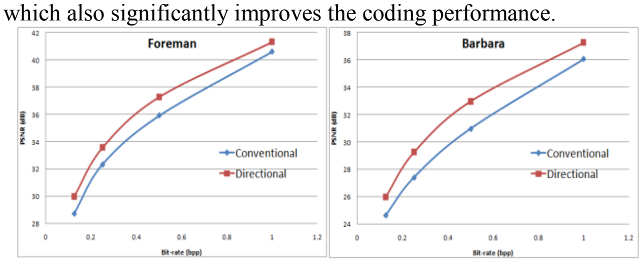


Fig. 6. Coding performance comparisons with natural images between the conventional and the directional wavelet. The data are extracted from [4].

IV. DATA-DEPENDENT DIRECTIONAL TRANSFORMS

In the intra-frame coding of H.264, the state-of-the-art video coding standard, actually the directional correlation can be partly exploited by the intra prediction. In the intra prediction, the reconstructed pixels on the neighboring blocks are used to predict the current block in one of the predefined modes. For example, when the block size is 4x4, besides DC prediction, there are still eight modes which allow the prediction to be along eight different directions. However, the intra prediction only removes the directional redundancy among neighboring blocks. It does not exploit the directional correlation within the current block. Thus, after the intra prediction, there are still directional correlations left in the prediction residues.

In [14], it has been shown that prediction can also be considered as a part of the transform in the block-based coding structure. The paper also proposes that the inter-block prediction and the transform within the current block should be jointly designed. To fully remove the directional redundancy within the image, mode-dependent directional transform (MDDT) [11] is proposed in the framework of H.264. Instead of using just one transform, i.e., the DCT in H.264, MDDT uses different transforms according to the prediction direction of the image blocks. As an example, since for 4x4 blocks there are nine different prediction modes, according MDDT designs nine corresponding transforms applied on the residue after the directional prediction. The transform for each direction, which can be designed to be separable or non-separable, is got by training the KLT based on the signals of the same mode. Thus, the transform is data-dependent. By applying intra prediction to remove inter-block directional redundancy and MDDT to remove that redundancy within the current block, the whole prediction and transform provide an efficient solution to exploit the directional correlation within H.264 framework. Actually, theoretically, when both the inter-block prediction and intra-block transform are optimal, the whole block-based coding system can achieve its optimality [15].

Fig. 7 shows the coding performance comparison between MDDT and DCT in the H.264 High Profile framework. It illustrates that even after the directional prediction, there is much redundancy related to each direction. Having different transforms for different directions in MDDT can achieve about 0.5dB's gain compared to the state-of-the-art. Recently, in [12], the authors present that the coding performance can still be much improved by utilizing more data-dependent directional transforms for each mode. It may mean that nine transforms may not be enough to fully exploit the directional correlation within images. Finer directional transforms are desired.

One advantage of such data-dependent directional transforms is that they are very suitable for block-based image coding, e.g., intra-frame coding in H.264. But since the transform matrices may rather be arbitrary according to the training of the data, the implementation of the transforms may not as efficient as those lifting-based directional ones.

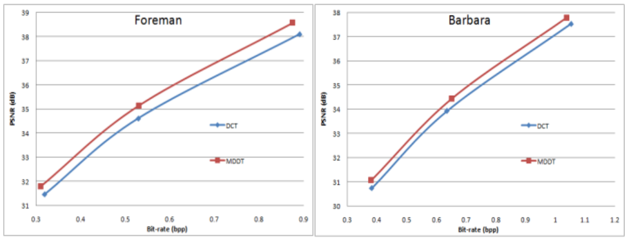


Fig. 7. Coding performance comparison between MDDT and DCT.

V. Conclusions and discussions

In this paper, we overview the directional transforms that have been shown to be effective in image coding. Although directionalities in images have been known for a long time, how to effectively utilize those directionalities in practical coding systems remained a problem, until recently new directional transforms have been created and superior coding results over the state-of-the-art systems have been shown. One of the directional transforms, MDDT, has been adopted in the recent standardization process of the video coding standard beyond H.264 [16]. With the development of new directional transforms, directionalities may be the next big thing for next generation coding standards and systems. Based on the evidences that have been shown, we think that directional transforms could influence more other coding standards in the near future.

There are three categories we have overviewed: reorganization-based, lifting-based, data-dependent directional transforms. The former two categories tend to perform a specific transform along a given direction. The directional transforms of the third category are specially designed for block-based coding schemes, in which the directional transform and the inter-block directional prediction provide an efficient solution to exploit the directional correlation within the whole image. In spite of different designs to exploit the directional correlation of images, these directional transforms show significantly coding performance improvements over conventional ones and dramatically improve the state-of-the-art coding systems, especially for images with much anisotropic content.

Although there has been shown much benefit in image coding, directional transforms for image coding still need further investigations to fully exploit the directionality of images. For reorganization-based directional transforms, how to support finer directions to sub-pixel level and how to efficiently exploit inter-block correlation still need to be studied. For lifting-based directional transforms, the main problems are to reduce the complexity of the direction decision and circumvent the constraints placed by cross-block operations. For data-dependent directional transforms, finding regular expressions and a finer directional representation are of interest.

Acknowledges

The authors would like to thank Ms. Jingjing Fu and Ms. Xiulian Peng for kindly generating some experimental results.

References

[1] S. Srinivasan, C. Tu, Z. Zhou, D. Ray, S. Regunathan and G. Sullivan, "An introduction to the HD Photo technical design," JPEG document wg1n4183, Apr. 2007.

[2] B. Zeng and J.-J. Fu, "Directional discrete cosine transforms - a new framework for image coding," *IEEE Trans. Circ. Syst. for Video Technology,* pp. 305-313, Mar. 2008.

[3] C.-L. Chang and B. Girod, "Direction-adaptive partitioned block transform for image coding," *IEEE Int. Conf. Image Processing*, 2008.

[4] W. Ding, F. Wu, X. Wu, S. Li and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Trans. Image Processing,* pp. 416-427, Feb. 2007.

[5] C.-L. Chang, A. Maleki, and B. Girod, "Adaptive wavelet transform for image compression via directi*onal quincunx lifting," Proc. IEEE Int. Workshop on Multimedia Signal Processing*, Shanghai, China, November 2005.

[6] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform for i*mage compression," IEEE Trans. Image Processing, vol. 16*, no. 5, pp. 1289-1302, May 2007.

[7] H. Xu, J. Xu and F. Wu, "Lifting-based directional DCT-like transform for image coding," *IEEE Trans. Circ. Syst. for Video Technology,* pp. 1325-1335, Oct. 2007.

[8] J. Xu, F. Wu, J. Liang and W. Zhang, "Directional lapped transforms for image coding," *Data Compression Conference,* pp. 142-151, Mar. 2008.

[9] Y. Liu and K. N. Ngan,"Weighted Adaptive Lifting-Based Wavelet Transform for Image Coding," *IEEE Trans. Image Processing,* pp. 500-511, Apr. 2008.

[10] W. Dong, G. Shi and J. Xu, "Adaptive Nonseparable Interpolation for Image Compression With Directional Wavelet Transform," *IEEE Signal Processing Letters*, vol. 15, pp. 233-236, 2008.

[11] Y. Ye and M. Karczewicz, "Improved H.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," *IEEE Int. Conf. Image Processing,* 2008.

[12] X. Zhao, L. Zhang, S. Ma and W. Gao, "Rate-Distortion Optimized Transform," ISO/IEC JTC1/SC29/WG11 MPEG m16926, Oct 2009, Xi'an, China.

[13] X. Peng, F. Wu and J. Xu, "Directional Filtering Transform," *IEEE Int. Symp. Circuits and Systems,* 2009.

[14] J. Xu, F. Wu and W. Zhang, "Intra-predicitve transforms for image coding," *IEEE Int. Symp. Circuits and Systems,* 2009.

[15] J. Xu, F. Wu and W. Zhang, "On the coding gain of intra-predictive transforms," *IEEE Int. Conf. Image Processing,* 2009.

[16] "34th VCEG Meeting Report," document VCEG-AH01, Study group 16 question 6, Turkey, Jan. 2008.

[17] S.-Y. Zhu, S.-K. Au Yeung, and B. Zeng, "R-D performance upper bound of transform coding for 2-D directional sources," *IEEE Signal Processing Letters*, vol. 16, no. 10, pp. 861-864, 2009.