Direction Scalability of Adaptive Directional Wavelet Transform: An Approach Using Block-Lifting Based DCT and SPIHT

Yuichi Tanaka, Madoka Hasegawa, Shigeo Kato
Dept. of Information Science
Utsunomiya University
Utsunomiya, Tochigi 321-8585 Japan
Email: {tanaka, madoka, kato}@is.utsunomiya-u.ac.jp

Taizo Suzuki, Masaaki Ikehara
Dept. of Electronics and Electrical Engineering
Keio University
Yokohama, Kanagawa, 223-8522 Japan
Email: {suzuki, ikehara}@tkhm.elec.keio.ac.jp

***Abstract*— Adaptive directional wavelet transform is an effective alternative of the traditional 2-D wavelet transform for image coding. It is able to transform an image adaptively along diagonal orientations as well as conventional vertical/horizontal directions. However, it requires to transmit transform direction information to the decoder side. For image coding at very low bitrates, the bit budget of the direction information degrades a reconstructed image quality. In this paper, a method to construct a scalable bitstream for transform directions is presented. We utilize the fact that the matrix yielded by transform direction indices still contains the original image characteristics. The matrix is transformed by a block-lifting based DCT, then encoded by SPIHT to yield a scalable bitstream. Our method is effective for very low bitrate image coding, and is comparable to the non-scalable one for middle-to-high bitrates.**

I. Introduction

In image and video processing using wavelet transform (WT), multiresolution decomposition is one of the most important features [1]–[3]. It represents an image by several multiresolution subbands. Since most images have higher energy in low-frequency subbands than high-frequency ones, the decomposition is very effective for compression, denoising, etc.

Traditionally, 2-D WT is based on 1-D filterings along vertical and horizontal directions. However, edges usually exist along various directions. Those limited transform directions cause poor directional selectivity in the traditional 2-D WT, especially in compression where the high-frequency subbands are often quantized coarsely. Consequently, the reconstructed image has significant blurry artifacts.

Adaptive directional WT with lifting implementation [4], [5] is one of the most efficient transforms against the directional selectivity problem, and it yields a multiresolution image fully compatible with that of the traditional WT. It applies *directional lifting* in each lifting step. Prediction and updating steps for directional lifting can be in several diagonal orientations as well as traditional vertical/horizontal ones. Lifting factorization always guarantees perfect reconstruction even for directional lifting steps. As a result, it is regarded as a good alternative for the 2-D WT.

The authors have proposed an efficient realization of the adaptive directional WT based on prefilterings of an original image [6]–[8]. The obtained subbands by the prefilterings are used as “reference frames” to calculate transform directions. The method succeeds to reduce the computation complexity significantly compared with that of the previously proposed ones in spite of the comparable image coding performance and simple framework.

However, the adaptive directional WTs share one problem about the transform direction information. Especially at very low bitrates, bitrates for transform directions require high percentage compared to the target bitrates. Finally, the reconstructed image has a lot of annoying artifacts, or sometimes we cannot spend any bits for image textures. Rate-dependent direction determination [5] can partly solve the problem since it provides associated transform direction data for a specific target bitrate. Unfortunately, the bitstream for transform directions cannot be an embedded one. As a result, the rate-dependent determination does not provide a good trade-off between the bitrate for the associated direction information and image textures especially at very low bitrate.

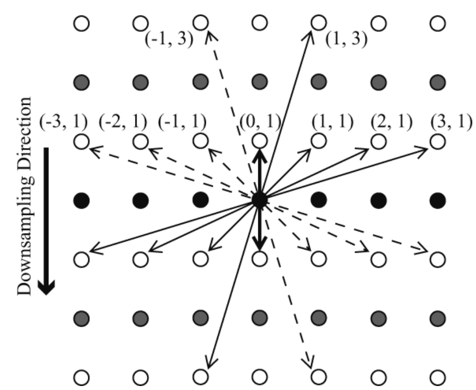


Fig. 1. Directions for the directional WT. Black pixels correspond a row to be transformed.

In [9], an approach has been proposed to tackle this problem. It first calculates transform directions for some target bitrates, and then it merges them to construct a scalable bitstream of the transform directions. It has a layered or level-unit structure, hence the adaptive directional WT can be efficiently used from low to high bitrates. Unfortunately, its structure to yield a scalable bitstream is complex and it still requires to calculate K transform direction sets for K quality layers of the directions.

In this paper, we present a very simple approach to represent a scalable bitstream of transform directions for the adaptive directional WT. It is based on our prefiltering-based method [6]–[8]. The key of this work is to recognize the matrix obtained from indices of transform directions as a small image which preserves original image characteristics. A block-based integer transform, called block-lifting based DCT (BL-DCT) [10], is used to transform the direction matrix. Moreover, SPIHT progressive encoder is employed to yield a scalable bitstream. In the experimental results, our method shows significant image quality improvements at very low bitrates compared with the non-scalable approach.

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

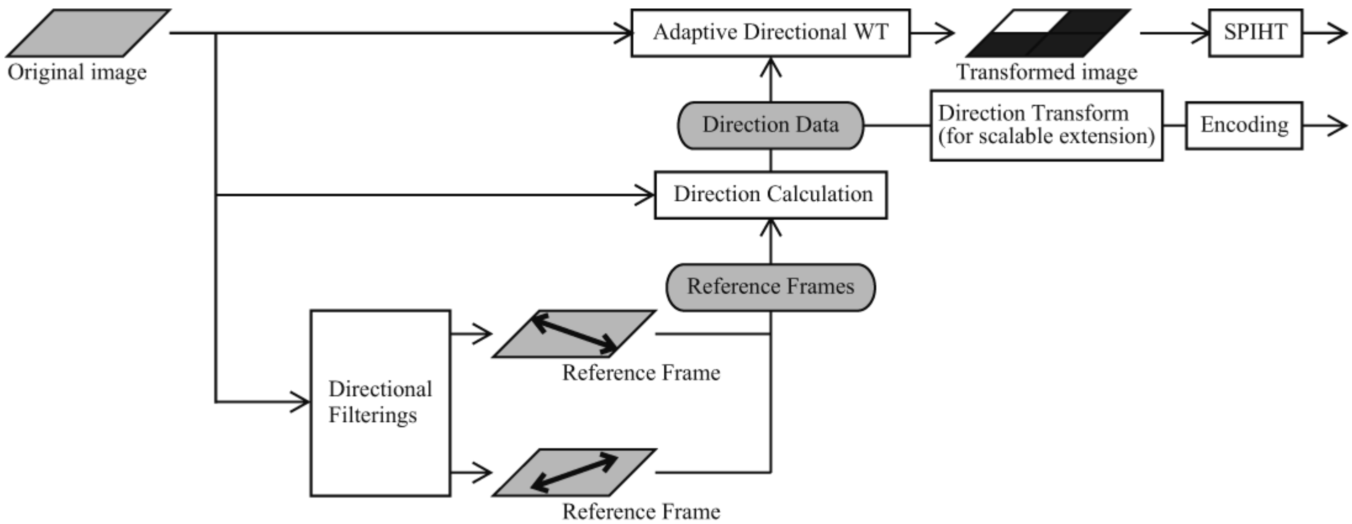


Fig. 2. Framework of D1F-WT. The arrowheads in the reference frames represent the main directions of the diagonal lines in them.

II. Directional Lifting and Adaptive Directional Wavelet Transform

A. Directional Lifting

In this paper, we consider the directional lifting for integer pixel positions [4]. Hereafter, we define the notations of the transform directions of directional lifting as the relative pixel position from the pixel to be transformed. Some typical directions are illustrated in Fig. 1 where the direction for the separable WT (SWT) is defined as (0, 1).

Let $x(m,n)$ and (l_0,l_1) denote the pixel value at (m,n) and a transform direction, respectively. A prediction step with the vertical downsampling is represented as

$$h(m,2n+1) = x(m,2n+1) - P(m,2n) \tag{1}$$

where $h(m,2n+1)$ represents a highpass branch of the directional lifting step and

$$P(m,2n) = p_i(x(m+l_0,2n+1-l_1)+x(m-l_0,2n+1+l_1)) \tag{2}$$

in which p_i is a coefficient for this prediction step. An updating step is given by

$$l(m,2n) = x(m,2n) + U(m,2n+1) \tag{3}$$

where $l(m,2n)$ represents a lowpass branch and

$$U(m,2n+1) = u_i(h(m+l_0,2n-l_1)+h(m-l_0,2n+l_1)), \tag{4}$$

in which u_i is an updating coefficient. Clearly these lifting steps are perfect reconstruction and can be cascaded with other lifting steps similar to SWTs. The resulting subbands are compatible with those using the SWTs. Note that the even (odd) row to be transformed requires neighboring odd (even) rows in each lifting step for perfect reconstruction. Therefore, the directions (1, 2), (-1, 2), etc. cannot be transformed without interpolating pixels. For more generalized (fractional-pel) representations of directional lifting steps, please refer to [5].

B. Adaptive Directional Wavelet Transform with Prefiltering

We have recently proposed an efficient realization of the adaptive directional WT [6]–[8]. Its analysis side framework is shown in Fig. 2. Different from the other frameworks [4], [5], our scheme has a directional filtering stage, which transforms an input image before calculating transform directions. These directional filters extract directional information from the image and resulting subbands are used as reference frames to calculate transform directions of the adaptive directional WT since these reference frames indicate the positions of diagonal lines in the image. Finally, both of the multiresolution image and the direction data are used for image coding.

The reference frames are only used on the analysis side to calculate the transform directions. Hence, the synthesis side of this framework is exactly the same as that of the previously proposed ones [4], [5]. In this paper, the directional filtering stage simply uses directional WT highpass filters along two fixed directions $(1,1)$ and $(-1,1)$. We call this adaptive directional WT as D1F-WT, which is the acronym of Directional 1-D Filtering [7], [8].

III. Block-Lifting Based DCT

Block-lifting based DCT (BL-DCT) is a DCT based integer transform which has an arbitrary block size M ($M = 2^n$, $n \in \mathbb{N}$) and achieves higher compression ratio than conventional integer DCTs since it can merge many rounding operators [10]. M-channel BL-DCT type-II is expressed by

$$\mathbf{C}_{II}^{[M]} = \mathbf{P}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & -\mathbf{X}_3\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{X}_2\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{X}_1 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{X}_0\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}\end{bmatrix} \tag{5}$$

where $\mathbf{P}$ and $\mathbf{I}$ are a permutation matrix and the identity matrix, respectively, and $\mathbf{X}_0 = \sqrt{2}\mathbf{C}_{II}^{[M/2]} - \mathbf{I}$, $\mathbf{X}_1 = -1/\sqrt{2}\mathbf{C}_{III}^{[M/2]}$, $\mathbf{X}_2 = \sqrt{2}\mathbf{C}_{II}^{[M/2]} - \mathbf{C}_{II}^{[M/2]2}$ and $\mathbf{X}_3 = \mathbf{C}_{IV}^{[M/2]}\mathbf{C}_{II}^{[M/2]}$. Similarly, M-channel BL-DCT type-IV is presented by

$$\mathbf{C}_{IV}^{[M]} = \begin{bmatrix}\mathbf{V}_0 & \mathbf{V}_1\\ \mathbf{V}_1^T & \mathbf{V}_2\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & -\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{Y}_0\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{Y}_1 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{Y}_0\\ \mathbf{0} & \mathbf{I}\end{bmatrix}$$

where $\mathbf{V}_0 = \mathbf{V}_0^T$, $\mathbf{V}_2 = \mathbf{V}_2^T = -\mathbf{V}_1^{-1}\mathbf{V}_0\mathbf{V}_1$, $\mathbf{Y}_0 = -\mathbf{V}_1^T$ and $\mathbf{Y}_1 = (\mathbf{I} - \mathbf{V}_0)\mathbf{V}_1^{-T}$. The integer-to-integer transform is obtained by applying rounding operators for every block-lifting step. Note that (5) is not a complete lifting structure yet due to $\mathbf{X}_3$. We can achieve the completeness by iterating lifting factorization of $\mathbf{C}_{II}^{[M]}$ and $\mathbf{C}_{IV}^{[M]}$ in $\mathbf{X}_3$ shown as Fig. 3. In this paper, the eight-channel BL-DCT is used to transform the direction matrix.

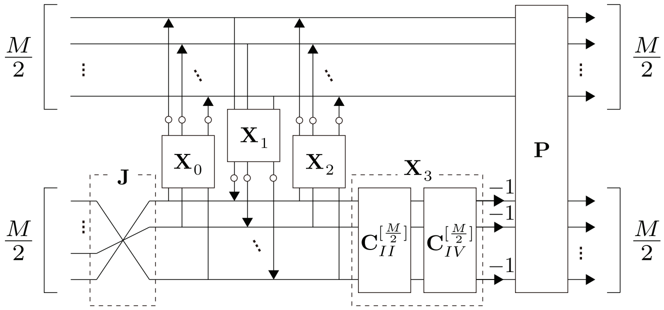
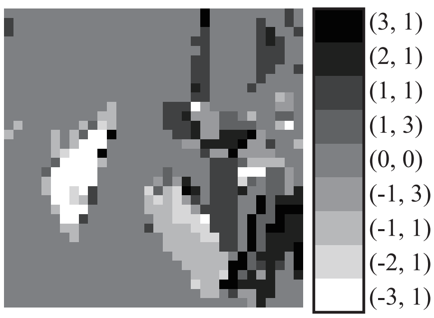
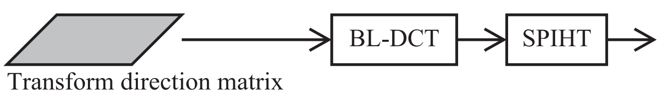


Fig. 3. M-channel BL-DCT (white circles: rounding operations).



(a)



(b)

Fig. 4. Transformation of direction matrices. (a) a transform direction example for *Barbara*. (b) Framework of scalable encoding of the direction matrix.

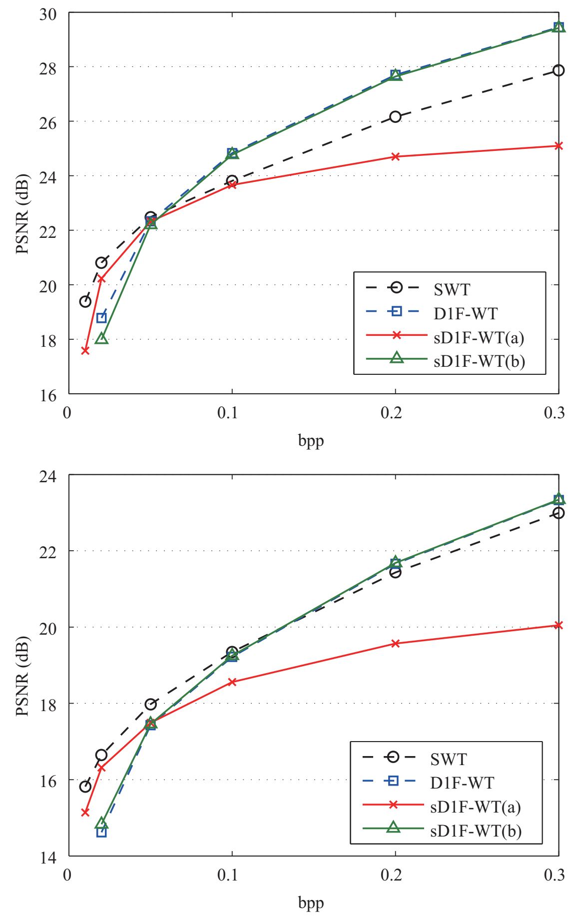


Fig. 5. PSNR comparisons. (Top) *Barbara*. (Bottom) *Bike*.

IV. Direction Scalability Using BL-DCT and SPIHT

The D1F-WT calculates its transform directions in a block-based fashion. They constructs a transform direction matrix whose each element contains an index value of the transform direction. Conventionally, a direction vector is obtained by raster scanning of the matrix, and then the vector is encoded by runlength coding [7], [8]. However, the resulting vector does not have a rate-scalable property. For very low bitrate image coding, the non-scalable direction data severely affect to a reconstructed image quality since we can spend very few bits for textures at that rate. We resolve this problem by using *dual* SPIHT encoding that one is used for textures and the other for transform directions.

A transform direction matrix for *Barbara* by the D1F-WT is depicted in Fig. 4(a). Clearly a transform direction corresponds to the image feature itself. Consequently, the direction matrix is regarded as a small image. Thus, we simply transform the direction data by the BL-DCT and the transformed image is encoded by the scalable encoder SPIHT. The framework of direction matrix transformation is shown in Fig. 4(b).

In the synthesis side, the direction matrix is reconstructed by using a received scalable bitstream. Indeed a compressed direction matrix does not match the actual one. Therefore, the advantage of the scalable direction is for the very low bitrate case. Additionally, the transform direction matrix is usually very smaller than the original image. For example, if we determine the directions for every 16×16 block in a 512×512 image, the transform direction matrix is of the size 32×32. For such a small image, block-based transform is more suitable than the normal filter banks (e.g., 9-7 filters) which requires signal extensions at image boundaries. Moreover, integer transforms are desired since a transform direction matrix strictly contains only integer values. They are the reason that we apply the BL-DCT to the direction transformation.

Our BL-DCT based approach requires a simple structure compared with the previous layer/level-unit based method [9]. It can be applied for tree-based adaptive directional wavelet transform [4], [5] which constructs a tree for transform directions by using a cost function with Lagrangian multiplier λ. The resulting tree is highly rate-dependent, thus in the scalable transform direction case, optimal trees for some λ's are first constructed and then they are utilized to obtain a scalable bitstream. In contrast to that, we just need to calculate one transform direction matrix for high bitrates. Then the BL-DCT and SPIHT can construct a scalable direction matrix for low bitrates.

V. Experimental Results

In this section, our proposed approach is applied for image coding and compared with the SWT and the non-scalable D1F-WT [7]. Both of the adaptive directional WTs are based on D1F-WT. Two 512×512 images, *Bike* and *Barbara*, are used for this experiment. We tested two scalable D1F-WT; one spends 0.1 bpp for the 32×32 transform direction matrix, whereas the other reconstructed the matrix

Fig. 6. Comparison of reconstructed image qualities. From left to right, SWT, D1F-WT [7], sD1F-WT(a) and sD1F-WT(b).

losslessly. We refer these two cases as sD1F-WT(a) and sD1F-WT(b), respectively.

Fig. 5 shows PSNR comparisons of two images. Clearly in very low bitrates, sD1F-WT(a) shows comparable performance to the SWT, whereas the D1F-WT and the sD1F-WT(b) are significantly worse than these two transforms. In 0.1 bpp or higher, the D1F-WT and the sD1F-WT(b) present very similar PSNRs and they outperforms the SWT and the sD1F-WT(a). The reconstructed visual quality of *Barbara* image is compared in Fig. 6. It is clear that the reconstructed images of the SWT and the sD1F-WT(a) are better in 0.01–0.02 bpp. Especially in 0.01 bpp, only these two transforms could reconstruct the image. In contrast to that, the D1F-WT and sD1F-WT(b) preserve textures in the image better than the other two for 0.1 bpp.

Note that we can estimate an optimal truncation point of the direction bitstreams by using the similar method to [9]. Therefore, the proposed approach can choose the best one between the sD1F-WT(a) and (b). The bitstream for the direction data is easily embedded into that for the texture data. As a result, the sD1F-WT can achieve good performance from very low to high bitrates.

VI. Conclusions

In this paper, we propose a simple approach to obtain a scalable bitstream for transform directions of adaptive directional WT. It uses the BL-DCT and SPIHT to make a scalable bitstream from the viewpoint that the transform direction matrix can be recognized as a small image. In the experimental results, our method shows a possibility of a flexible bitrates for the direction matrix. Especially in very low bitrates, our scalable method presented significant image quality improvements than the non-scalable one.

References

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. NJ: Prentice-Hall, 1993.

[2] M. Vetterli and J. Kovačevic, *Wavelets and subband coding*. NJ: Prentice-Hall, 1995.

[3] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*. MA: Wellesley-Cambridge, 1996.

[4] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform for image compression," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1289–1302, 2007.

[5] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 416–427, 2007.

[6] Y. Tanaka, M. Hasegawa, and S. Kato, "Highpass-filtering based adaptive directional wavelet transform," in *Proc. 27th Picture Coding Symposium*, 2009.

[7] Y. Tanaka, M. Hasegawa, S. Kato, M. Ikehara, and T. Q. Nguyen, "Adaptive directional wavelet transform using pre-directional filtering," in *Proc. ICIP'09*, 2009, pp. 1–4.

[8] ——, "Adaptive directional wavelet transform based on directional prefiltering," *IEEE Trans. Image Process., accepted*, 2010.

[9] T. Xu, C.-L. Chang, and B. Girod, "Scalable direction representation for image compression with direction-adaptive discrete wavelet transform," in *Proc. Visual Communication and Image Processing*, 2007.

[10] T. Suzuki and M. Ikehara, "Design of Block Lifting-Based Discrete Cosine Transform Type-II and IV," in *Proc. DSP/SPE 2009*, 2009, pp. 480–484.