Directional Discrete Cosine Transforms for Image Coding

Bing Zeng and Jingjing Fu

Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong, emails: {eezeng, jjfu@ust.hk}

Abstract *- Nearly all block-based transform schemes for image and video coding developed so far choose the 2-D discrete cosine transform (DCT) of a square block shape. With almost no exception, this conventional DCT is implemented separately through two 1-D transforms, one along the vertical direction and another along the horizontal direction. In this paper, we develop a new block-based DCT framework in which the first transform may follow a direction other than the vertical or horizontal one, while the second transform is arranged to be a horizontal one. Compared to the conventional DCT, the resulting directional DCT framework is able to provide a better coding performance for image blocks that contain directional edges – a popular scenario in many image signals. By choosing the best from all directional DCT's (including the conventional DCT as a special case) for each image block, we will demonstrate that the rate-distortion coding performance can be improved remarkably.*

1. INTRODUCTION

Digital images undoubtedly contribute a big trunk in digital multimedia data people are enjoying today. Nevertheless, as raw data, each digital image usually needs a large volume of bits to represent, and therefore some sort of compression has to be done before it is transmitted or stored. Over the past three decades, a lot of image coding methods have been developed for this compression task, such as predictive coding, transform coding, vector quantization, and sub-band/wavelet coding. Among various image coding techniques, the block-based transform approach has become particularly successful, thanks to its simplicity, excellent energy compaction in the transform domain, super compromise between bit-rate and quantization errors, etc.

With almost no exception, every transform-based scheme chooses the 2-D discrete cosine transform (DCT) [1], [2] that is applied on non-overlapped image blocks of a square size $N \times N$. In practice, this conventional $N \times N$ DCT is always implemented separately through two N -point transforms, one along the vertical direction and another along the horizontal direction.

Both vertical and horizontal directions are important according to the human visual system (HVS). In the meantime, a lot of blocks in an image do contain vertical and/or horizontal edge(s). Thus, the conventional DCT seems to be the best choice for image blocks in which vertical and/or horizontal edges are dominating. On the other hand, however, there also exist other directions in an image block that are perhaps as equally important as the vertical/horizontal one, e.g., two diagonal directions.

We strongly believe that the conventional DCT would not be the best choice for image blocks in which some directional edges other than the vertical/horizontal ones become dominating. Such believe motivates us to attempt to develop a "directional" DCT framework in this paper so that we can choose the most appropriate directional DCT according to the dominating edge(s) within each individual image block. Our results demonstrate that this framework can indeed improve the coding performance remarkably.

2. THE CONVENTIONAL DCT AND ITS PERFORMANCE LIMITATIONS

As indicated earlier, the conventional $N \times N$ 2-D DCT is always implemented separately by two N -point 1-D transforms. Let us use $\boldsymbol{a} = [a(i,j)]_{N\times N}$ and $\boldsymbol{C}_{N\times N}$ to denote an image block of size $N \times N$ and the $N \times N$ transform matrix for the N -point DCT, respectively. After the conventional DCT, the transformed coefficient block can be expressed as

$$\boldsymbol{A} = \left[A(u,v)\right]_{N\times N} = \boldsymbol{C}_{N\times N} \cdot \boldsymbol{a} \cdot \boldsymbol{C}_{N\times N}^{T}, \tag{1}$$

where

$$\boldsymbol{C}_{N\times N} = \left[c(i,j)\right]_{N\times N},\ c(i,j) = \alpha(i)\cos\left(\frac{(2j+1)}{2N} i\pi\right), \tag{2}$$

$$\alpha(i) = \begin{cases} \sqrt{1/N} & i = 0 \\ \sqrt{2/N} & i \neq 0 \end{cases} \tag{3}$$

It is important to choose the weighting factor $\alpha(i)$ as in Eq. (3) so that the resulting DCT matrix is always unitary. This property is very crucial to the subsequent quantization in the DCT domain, as it guarantees that the mean square error (MSE) yielded by quantization in the DCT domain will not be changed by the IDCT that will be used to generate all reconstructed image blocks.

As indicated earlier, the conventional 2-D DCT seems to be the best choice for image blocks in which vertical and/or horizontal edges are dominating. However, it may cause some defects when it is applied to an image block in which other directional edges become dominating. For instance, let us consider two image blocks shown in Fig. 1 where two constant regions are separated along one diagonal direction. Then, it is easy to understand that the conventional DCT

Fig. 1. Two image blocks that contain diagonal edges.

may not be the best choice, as a number of unnecessary non-zero AC coefficients will be generated.

In practice, each image may have some (or many) blocks that contain such diagonal or other directional edges. By recognizing such characteristics, the most recent video coding standard H.264 [5] has developed a number of directional predictions (including the vertical and horizontal ones as two cases, because they also represent directional information) in the coding of all intra blocks – called the intra predictions. These intra predictions have proven to be rather effective in H.264. Nevertheless, it is still the conventional DCT that is used after each intra prediction – which seems to be very unfortunate. We believe that a more reasonable approach should be to apply some "directional" DCT after each directional intra prediction. In a more general context, a "directional" DCT should be applied to an image block in which the corresponding directional edges are dominating. Such a directional DCT framework will be developed in the next section.

3. Directional Discrete Cosine Transforms – The Framework

Following the intra prediction modes used in H.264, we define eight same directional modes (Modes 0-1 and Modes 3-8) for an arbitrary size $N \times N$. Clearly, Modes 0 and 1 are reduced back to the conventional DCT. In the meantime, it is easy to find that only the diagonal down-left mode (Mode 3) and the vertical-right mode (Mode 5) are the essential modes, whereas other modes can be derived by flipping/transposing Mode 3 or 5.

A. Directional DCT for the diagonal down-left mode

As shown in Fig. 2, the DCT's in the first step will be performed along the diagonal down-left direction, i.e., for each diagonal line with $i + j = k$, $k = 0,1,\cdots,2N-2$. There are totally $2N-1$ diagonal down-left DCT's to be done, whose lengths are $[N_k] = [1,2,\cdots,N-1,N,N-1,\cdots,2,1]$. All coefficients after these DCT's are expressed into a group of column vectors

$$\boldsymbol{A}_k = \left[A(0,k),\cdots,A(N_k-1,k)\right]^T, \quad k = 0,1,\cdots,2N-2\,. \tag{4}$$

Notice that each column $\boldsymbol{A}_k$ is of a different length N_k, with the DC component placed at top, followed by the first AC component and so on. To complement the arrangement of coefficients after the diagonal DCT in the first step, the second DCT is arranged to be a horizontal one and thus applied to each row that can be expressed as $[A(u,v)]_{v=u:2N-2-u}$ for $u = 0,1,\cdots,N-1$. The coefficients after the second DCT are pushed horizontally to left and denoted as $[\hat{A}(u,v)]_{v=0:2N-2-2u}$ for $u = 0,1,\cdots,N-1$.

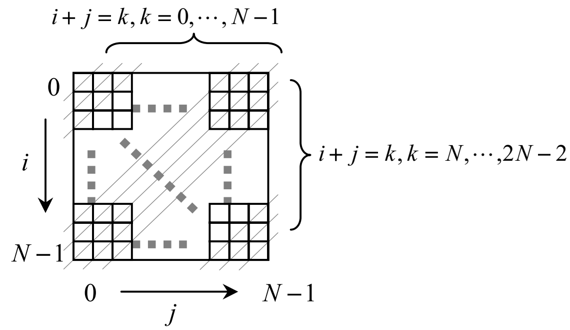


Fig. 2. An $N \times N$ image block in which the first 1-D DCT will be performed along the diagonal down-left direction.

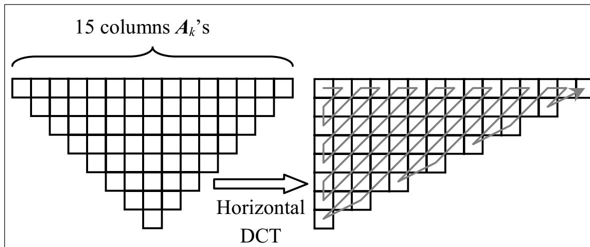


Fig. 3. Example of $N = 8$: arrangement of coefficients after the first DCT (left) and arrangement of coefficients after the second DCT as well as the modified zig-zag scanning (right).

Figure 3 shows an example for $N = 8$. Clearly, it is rather reasonable to apply a horizontal DCT in the second step, because the first row contains all DC components and each of other rows contains all AC components with the same index. The right part of Fig. 3 also shows a modified zig-zag scanning that will be used to convert the 2-D coefficient block into a 1-D sequence so as to facilitate the runlength-based VLC.

B. Extension to other directional modes

Extension to other modes is straightforward. For instance, for the diagonal down-right mode (Mode 4), we can simply flip it (either vertically or horizontally), and then the flipped image block will fall into the case as discussed above.

As the second example, let us consider the vertical-right mode (Mode 5) where the block size is selected at 8×8. Referring to Fig. 4, the directional DCT in the first step follows the vertical-right direction and will generate a coefficient block as shown in the middle part of Fig. 4. Next,

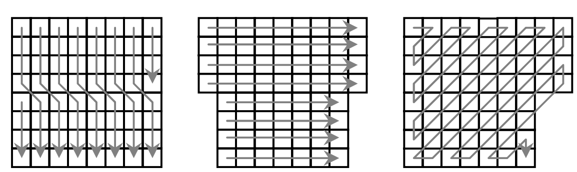


Fig. 4. Directional DCT for the vertical-right mode (Mode 5).

the second DCT will be a horizontal one again (of length 9 for the first 4 rows and 7 for the last 4 rows, respectively) and all coefficients after this DCT will be pushed to left. Finally, a modified zig-zag scanning is defined as in the right part of Fig. 4.

For Modes 6-8, we can simply flip or transpose the image block first and then the manipulated block will fall into the case of Mode 5.

4. DIRECTIONAL DCT WITH DC CORRECTION

As indicated earlier, Mode 0 and Mode 1 are the same as the conventional DCT, and therefore can be applied directly on image blocks. On the other hand, however, Modes 3-8 developed above cannot be applied directly, because they would suffer from the so-called *mean weighting defect* [3]. To understand this in detail, let us consider the simple example of a spatially uniform image block, i.e., all pixels in the block have the same gray value. After a directional DCT (chosen from Modes 3-8) is applied in the first step, all AC coefficients are zero, but the resulting DC coefficients will become different for different directional lines because of the different weighting factors used there. Consequently, the DCT's applied in the second step will unavoidably generate some unnecessary non-zero AC coefficients – which is rather absurd.

To solve this problem, some kind of DC correction has to be accommodated around the second DCT. In the following, we consider the diagonal down-left mode (Mode 3) only and introduce two methods for such DC correction, whereas other directional modes can be handled in the same way.

A. Modifying the weighting factors

The simplest method for the DC correction is to re-define the weighting factor $\alpha(i)$ used in the DCT defined earlier in Eq. (3). It is easy to find that, in order to avoid the mean weighting defect in the diagonal down-left mode, $\alpha(i)$ should be chosen as

$$\hat{\alpha}(i) = \begin{cases} 1/N_k & i = 0 \\ \sqrt{2}/N_k & i \neq 0 \end{cases} \quad k = 0,1,\cdots,2N-2\,, \tag{5}$$

where N_k is the length of the k -th diagonal down-left line. For any of other directional modes (Modes 4-8), the corresponding weighting factor $\alpha(i)$ can be modified similarly.

However, the DCT matrix after the modification on $\alpha(i)$ becomes to be a non-unitary one. On the other hand, the transform coding theory suggests that the use of a non-unitary transform is highly disadvantageous for coding efficiency, because it would suffer from the so-called *noise weighting defect* [3] - some statistics of quantization errors, e.g., spatial distribution of error variances or the frequency characteristics of the error signal, will be weighted in an uncontrollable manner.

B. DC separation and ΔDC correction

To solve this dilemma problem, Kauff and Schuur proposed a novel method that consists of two steps: (1) DC separation and (2) ΔDC correction, in their work on SA-DCT [3]. This method can be readily applied in our case, with some details described below.

In the DC separation step, the mean value m of an image block $\boldsymbol{a} = [a(i,j)]_{N\times N}$ is calculated and it will be quantized to $\overline{m}$. Then, m is subtracted from the initial image block, and subsequently, the diagonal down-left DCT is applied to the resulting zero-mean image block $\boldsymbol{b} = [b(i,j)]_{N\times N} = [a(i,j)]_{N\times N} - m$. The coefficients after this step are expressed into a group of column vectors $\boldsymbol{B}_k = [B(0,k),\cdots,B(N_k-1,k)]^T$, where $k = 0,\cdots,2N-2$, and each vector $\boldsymbol{B}_k$ has length N_k as defined before. Next, the horizontal DCT is applied to each row $[B(u,v)]_{v=u:2N-2-u}$, for $u = 0,1,\cdots,N-1$, and the resulting coefficients are pushed horizontally to left and denoted as $[\hat{B}(u,v)]_{v=0:2N-2-2u}$ for $u = 0,1,\cdots,N-1$.

In the ΔDC correction step, the DC component $\hat{B}(0,0)$ will be set to zero and thus will not be transmitted, while all AC coefficients will be quantized – the resulting coefficients are denoted as $[\hat{B}^Q(u,v)]$ with $\hat{B}^Q(0,0) = 0$. Next, the horizontal IDCT is applied on each row of $[\hat{B}^Q(u,v)]$ and the resulting coefficients are denoted as $[B^Q(u,v)]$. Then, a ΔDC correction term is computed as

$$\Delta DC = \sum_{k=0}^{2N-2} \sqrt{N_k}\, B^Q(0,k) \Big/ \sum_{k=0}^{2N-2} \sqrt{N_k}\,, \tag{6}$$

and this correction term will be subtracted from each $B^Q(0,v)$ for $v = 0,1,\cdots,2N-2$.

After the ΔDC correction, the second IDCT will be performed on each column of $[B^Q(u,v)]$ and the results will be placed back to the corresponding diagonal down-left line so as to generate a reconstructed image block of size $N \times N$. Finally, the quantized mean value $\overline{m}$ needs to be added back to the reconstructed image block.

It has been demonstrated in [3] that the ΔDC method is consistently better than modifying the weighting factor by 1-2 dB, and this result has been confirmed in our experiments. Consequently, the ΔDC method is always adopted in our directional DCT-based image coding.

5. EXPERIMENTAL RESULTS

In this section, we provide some experimental results to illustrate our directional DCT framework. To this end, we select four images: "Lena", "Barbara", "Peppers", and "Boat" (of size 512×512), as well as the first frames of four video sequences: "Akiyo", "Foreman", "Stefan", and "Mobile" (of the CIF format). We fix the block size at 8×8 for all test data. For four images, we implement the JPEG quantization/VLC where the quantization scaling factor is selected in the range [0.2, 3.0] with incremental step chosen at 0.2. For four video frames, we implement the H.263 quantization/VLC in which the QP value is selected in the range [3, 31] with incremental step set at 2.

While effective selection of the best directional DCT mode for each image block is very crucial, the primary goal of this paper is to develop the directional DCT framework and demonstrate that it is able to provide a remarkable coding gain. To this end, we leave how to select the best mode effectively as one of our future works, but rather adopt a brute-force method, i.e., we run quantization and VLC for all seven modes and select the best one according to a productive rate-distortion criterion – the product of the MSE and bit-count in each image block.

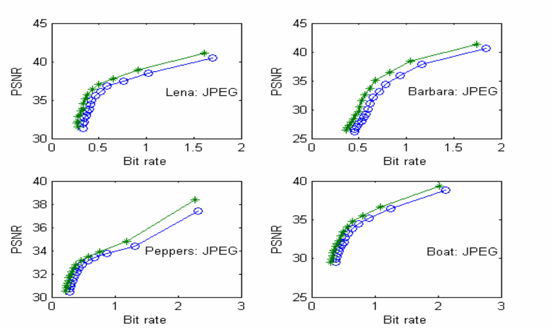


Fig. 5. Experimental results for four images.

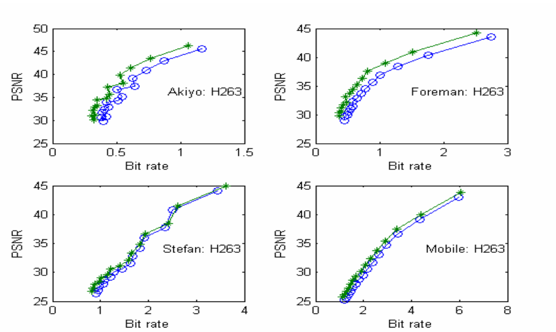


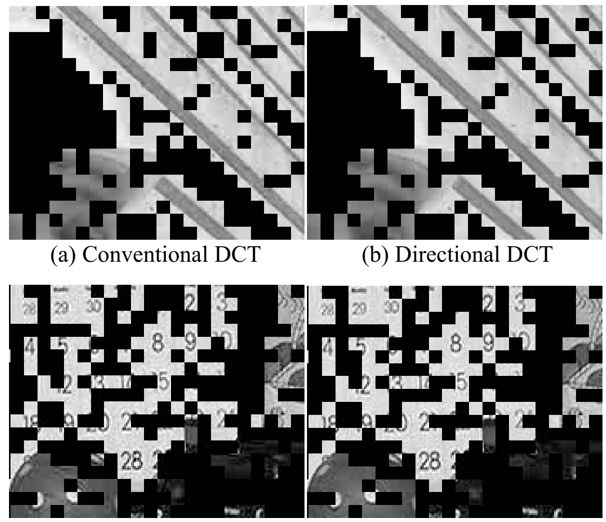
Fig. 6. Experimental results for four video frames.

The simulation results are shown in Figs. 5 and 6, where only image blocks that choose one from Modes 3-8 are included. It is clear that a significant gain has been achieved for all four images as well as for the frames of “Akiyo” and “Foreman”: ranging from about 1 dB in the high bit-rate end to over 3 dB in the low bit-rate end. The gain is also quite noticeable (0.5-1.0 dB) for the frames of “Stefan” and “Mobile”.

Fig. 7 provides some subjective comparisons between the conventional DCT and our directional DCT for the frames of “Foreman” and “Mobile”. All black blocks indicate that Mode 0 has been chosen in our directional DCT so that the conventional DCT and our directional DCT produce the same result. The purpose of blacking these blocks is to highlight all blocks that have chosen a coding mode from Modes 3-8 in our directional DCT scheme. It is clear that the visual quality achieved in our directional DCT has been improved significantly.

6. Conclusions and Future Works

In this paper, we developed a truly directional DCT framework in which the first transform can choose to follow a direction other than the vertical/horizontal one - the default direction in the conventional DCT. To complement the directional DCT in the first step, the second transform is arranged to be a horizontal one. While this new framework can be applied in any case in principle, we focused on seven directional modes in an arbitrary block size of $N \times N$ in this paper, including the vertical and horizontal directions. By selecting the best suited one from seven directional modes for each image block, we demonstrated that a remarkable coding gain has been achieved in the rate-distortion performance.



(a) Conventional DCT (b) Directional DCT

(c) Conventional DCT (d) Directional DCT

Fig. 7. Subjective comparisons between the conventional DCT and our directional DCT.

Our future works include: (1) implementing this new DCT framework for each image block in combination with the best intra prediction chosen in H.264; (2) how to independently select the best directional DCT mode effectively for each image block; (3) in-depth theoretical analysis so as to justify the effectiveness of this new framework; (4) use of some smart padding techniques (such as the one developed in [4]) to solve the mean weighting defect problem, and (5) use of this new DCT framework in motion-compensated frames in video coding.

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE Trans. Computer*, vol. 23, pp. 90-93, 1974.

[2] K. R. Rao and P. Yip, Discrete *Cosine Transform - Algorithms, Advantages, Applications*. London: Academic Press, 1990.

[3] P. Kauff and K. Schuur, “Shape-adaptive DCT with block-based DC separation and ΔDC correction,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, pp. 237-242, June 1998.

[4] G. Shen, B. Zeng, and M. L. Liou, “Arbitrarily shaped transform coding based on a new padding technique,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, pp. 67-78, Jan. 2001.

[5] ITU-T Rec. H.264 | ISO/IEC 14496-10 (AVC), “Advanced video coding for generic audiovisual services”, Mar. 2005.