DIRECTION-ADAPTIVE TRANSFORMS FOR CODING PREDICTION RESIDUALS

Robert A. Cohen[1], Sven Klomp[2], Anthony Vetro[1], Huifang Sun[1]*

[1]Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, {cohen, avetro, hsun}@merl.com
[2]Leibniz Universität Hannover, Germany, klomp@tnt.uni-hannover.de

ABSTRACT

In this paper, we present 2-D direction-adaptive transforms for coding prediction residuals of video. These Direction-Adaptive Residual Transforms (DART) are shown to be more effective than the traditional 2-D DCT when coding residual blocks that contain directional features. After presenting the directional transform structures and improvements to their efficiency, we outline how they are used to code both Inter and Intra prediction residuals. For Intra coding, we also demonstrate the relation between the prediction mode and the optimal DART orientation. Experimental results exhibit up to 7% and 9.3% improvements in compression efficiency in JM 16.0 and JM-KTA 2.6r1 respectively, as compared to using only the conventional H.264/AVC transform.

Index Terms— Video coding, discrete cosine transforms, directional transforms, H.264/AVC, prediction residuals

1. INTRODUCTION

Many popular image and video coding standards use the separable 2-D Discrete-Cosine Transform (2-D DCT) to compact the energy in blocks of data prior to entropy coding. For example, JPEG [1] applies an 8×8 2-D DCT on image-intensity blocks. H.264/AVC [2] uses an integer transform related to the 2-D DCT to transform residuals from Inter- and Intra-predicted blocks. The DCT was chosen because under certain conditions, it closely matches the performance of the optimal Karhunen-Loève transform (KLT). Since the separable 2-D DCT applies 1-D DCTs horizontally and vertically, it is well-suited for transforming blocks that contain horizontal and vertical features. Oblique features, however, can exhibit artifacts after the DCT coefficients are quantized, inverse quantized, then inverse transformed. This behavior was further verified in [3], which examined the suboptimal performance of the conventional 2-D DCT as compared to the optimal KLT, when transforming directional textures.

To improve coding performance on blocks with oblique features, several directional transforms have been developed [4–8]. In [4], a separable Directional 2-D DCT (DDCT) was used to transform image blocks. First, 1-D transforms were applied isotropically over an entire block. For example, in an 8×8 block, a vertically-oriented 2-D DCT would first apply a set of 1-D DCTs with length-8 down each column of the block. The resulting block of coefficients would then be operated upon by a second set of 1-D DCTs, applied across each row. For a diagonal directional DCT, the first set of 1-D DCTs would be applied along paths oriented at 45°. The second set of transforms, however, were not applied perpendicular to the first set, i.e. 135°. Each of the initial 1-D DCTs produces a DC coefficient. So, for an 8×8 block, there were 15 DC coefficients after the first transform pass. The second DCT pass would therefore start with a length-15 DCT along all the DC coefficients. Subsequent passes would be shorter, culminating with what would be equivalent to a length-1 transform on the last AC coefficient. For larger blocks, the second pass of 1-D transforms in the DDCT could become rather long. The Direction-Adaptive Partitioned Block Transform (DA-PBT) [5] improved upon the DDCT by partitioning the DC and AC coefficients after the first transform pass. A set of shorter transforms comprised the second pass. The DA-PBT also improved upon the coefficient scanning order.

In H.264/AVC, the 2-D block transform is applied to prediction residuals. For the case of Intra-slice coding, the residual for a block is computed after performing predictions from neighboring blocks. In intra-predicted blocks, the correlation tends to be higher in the same direction as the selected prediction mode. This behavior was leveraged in [7, 8], which use non-separable and separable KLTs respectively, trained on video data to improve the coding of Intra-predicted residuals. With this KLT-based Mode-Dependent Directional Transform (MDDT), however, training sets must be processed to obtain transforms that are optimal for a given codec structure. If a new method is developed for generating prediction residuals, a new set of KLTs would need to be generated.

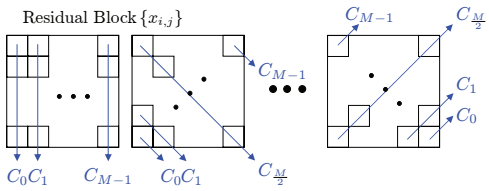
Inter-coded blocks typically are motion-compensated prediction residuals. These residuals tend to be more sparse than blocks found in the original image or video frame. In [6] it was observed that the correlation of data along the direction of features in a motion-compensated prediction residual is much higher than the correlation in other directions. The authors therefore only applied a first set of directional 1-D transforms on the residuals. As noted earlier, however, the number of DC coefficients for oblique directions can become quite high for large blocks.

In this paper, we present 2-D direction-adaptive transforms for both Inter and Intra-prediction residuals that use reduced-complexity secondary transforms on DC coefficients along with folded 1-D transform paths to eliminate the less-efficient shorter transforms. After introducing the Direction-Adaptive Residual Transform (DART) in the next section, we present how it is used in a rate-distortion optimized framework to code both Inter and Intra-prediction residuals. We then present experimental results showing both coding performance and how the optimal transform direction is related to the Intra-prediction direction.

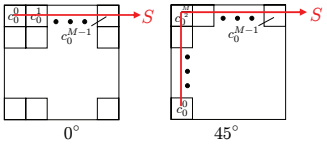
2. DIRECTIONAL TRANSFORMS FOR CODING INTER AND INTRA-PREDICTION RESIDUALS

For coding prediction residuals, we would like to exploit correlation along the direction of features in a block as was done in [4] and [5], while avoiding the generation of high-frequency transform coefficients by limiting transforms applied orthogonally to these features, as was done in [6]. In this section, we present the structure of these

*The author performed this work while at Mitsubishi Electric Research Laboratories



(a) Primary directional transforms



(b) Example secondary transforms

Fig. 1. Direction-Adaptive Residual Transform

transforms along with improvements to reduce the number of DC coefficients that need to be quantized.

2.1. Directional transform structure

Our proposed Direction-Adaptive Residual Transform (DART) consists of two passes: A primary transform pass comprising a set of isotropic 1-D DCTs, followed by a limited secondary pass applied along only the DC coefficients generated during the primary transform.

The primary transform pass of the $N \times N$ DART, for selected orientations, is shown in Fig. 1(a). The input to the primary pass is an $N \times N$ block of prediction residual data $\boldsymbol{X} = \{x_{i,j}\}$, where i and j respectively denote the row and column. 1-D DCTs are then applied along aligned paths within the block The outputs of the primary pass are sets of 1-D DCT coefficients $\boldsymbol{C}_m, m = 0, 1, \ldots M-1$ corresponding to each 1-D DCT. The coefficients along the path for $\boldsymbol{C}_m$ are $c_k^m, k = 0, 1, \ldots K_m$, where K_m is the length of transform path $\boldsymbol{C}_m$.

As shown in Fig. 1(a), Direction 0 consists of $M = N$ 1-D DCTs applied vertically within the block. Thus, there are $K = N$ coefficients generated by each of these transforms. If we employ a total of 16 different directions representing orientations from 0° to 180°, then Direction 4 corresponds to a diagonal down-right path. The primary pass for this orientation contains $M = 2N - 1$ 1-D transforms. The length K_m of each path varies from N down to 1. The directions between 0° and 45° generally have more than N 1-D transforms and fewer than $2N-1$. As in [4], we apply scaling factors to normalize the coefficients after the primary pass, to account for the varying path lengths.

In the traditional 2-D DCT a second pass would be applied over all the coefficients, producing one DC coefficient and $N^2 - 1$ AC coefficients. Our focus, however, is on coding prediction residuals, which as shown in [6] exhibit less correlation in directions orthogonal to the features in the block. Unlike [4] and [5], therefore, we bypass the secondary transform across the AC coefficients. We only apply one secondary transform $\boldsymbol{S}$ along the DC coefficients $c_0^m, m = 0, 1, \ldots M - 1$, as shown in Fig. 1(b). The coefficients output after the primary and secondary transforms are therefore

$$\begin{aligned} \text{Secondary DC coef:} \quad & s_0 \\ \text{Secondary AC coefs:} \quad & s_m \qquad m = 0, 1, \ldots M-1 \\ \text{Primary AC coefs:} \quad & c_k^m \quad \begin{cases} m &= 0, 1, \ldots M-1 \\ k &= 1, 2, \ldots K_m\,. \end{cases} \end{aligned} \tag{1}$$

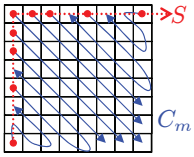


Fig. 2. Reduction of short paths via folding

As with the traditional 2-D DCT, the DART outputs one DC and $N^2 - 1$ AC coefficients, but the AC coefficients are generated without applying secondary transforms across the less-correlated primary AC coefficients. Tho benefits of this approach are utilized when the features in the block are aligned with direction of the primary transform paths. As described in Section 3, a rate-distortion process is used to match the transform direction to the texture in a block.

2.2. Transform path folding

As one can observe in Fig. 1(a), several of the oblique directional transforms include short 1-D DCT paths near the corners. For example, Direction 4, oriented at 45°, includes DCTs of length one and two. A one-point DCT can be regarded as a simple scaling without any particular orientation, and the coding gain of a two-point DCT is rather small.

In [5], the directional transforms used in DA-PBT were improved by partitioning the blocks. However, very short transform paths could still exist within the partitions. In contrast, we reduce the number of DC coefficients output by the primary transform pass by folding or reflecting the shortest 1-D transform paths, such as those near the corners. An example of this folding for the 45° orientation for an 8×8 block is shown in Fig. 2. Since the segments of the transform paths $\mathbf{C}_m$ are still aligned in the same direction, we can improve the coding efficiency in these regions while maintaining the directionality of the primary transform pass. Since the number of DC coefficients $c_0^m, m = 0, 1, \ldots M$ is reduced, the secondary transform is shortened as well. In the example shown here, the length of the secondary transform inside an 8×8 block is reduced from 15 to 9.

3. CODING OF INTER AND INTRA-PREDICTED BLOCKS IN THE H.264/AVC FRAMEWORK

Now that we have outlined the structure of the DART we will look at how it is used within the H.264/AVC framework to code Intra- and Inter-prediction residuals.

3.1. Transform selection and coding

In H.264/AVC, most macroblocks undergo a rate-distortion optimized decision process [9]. For Intra-coded blocks, residuals can be computed by predicting the current block using data from previously coded or reconstructed blocks. There are several prediction modes, such as vertical, horizontal, diagonal down-right, etc. that indicate the direction from which neighboring samples are used to predict samples in the current block. The best intra-prediction mode is chosen via a Rate-Distortion (R-D) optimized process which minimizes a cost function similar to $J_i = Di + \lambda R_i$, where D is an estimate of the distortion generated by a given prediction mode i, and R_i is an estimate of the number of bits needed to encode the residual and associated data using that mode.

With DART enabled, for each Intra-prediction mode i, a cost is $J_{i,d}$ is computed, where d denotes the transform used to compute the prediction residual. For example, with an eight-direction DART configuration, d is an index that iterates over the default 2-D

DCT transform and eight directional transforms. The best transform, $\underset{d}{\operatorname{argmin}}\,(J_{i,d})$ is selected to be used with mode i. After this process is repeated for all Intra-prediction modes, the mode that minimizes the R-D cost is chosen, along with the corresponding directional transform.

Now that the optimal transform has been selected, an index corresponding to this choice must be encoded into the bit-stream. To compress this index, we use CABAC [10] as implemented in H.264/AVC with three additional contexts. One context is used to code one bit that indicates whether the default 2-D DCT or the directional transform is selected. No additional data has to be coded for DCT. When DART is selected, we compute a prediction d_P of the optimal transform direction d. This can be done using neighboring blocks or other already coded data. Section 4.1 gives a practical example for such predictor. We next subtract the predicted index d_P from d. For the eight-direction case, the resulting difference index can be wrapped into the range of $[-3, 4]$, since the transformations are modulo 180°. This difference index is binarized using a truncated unary (TU) code [10]. The first bit of the TU codeword represents whether d_P and d are equal or not. Therefore, this bit is coded using its own CABAC context. Experiments have shown that the remaining bits are almost equally distributed and thus, jointly coded in the third context. The same applies for the sign bit, which is coded using the same context.

For Inter-coded blocks, the process is similar, except that the macroblock is partitioned, and motion estimation is performed on each partition to generate a motion-compensated prediction residual and corresponding distortion for the macroblock. For each partitioning mode, we iterate over the DART directions and use R-D optimization to choose the best direction associated with each mode. The optimal mode and its corresponding directional are then signaled. The optimal transform direction depends upon the texture in the prediction residual, which is generally dependent upon the data in the picture, not the motion vectors or partitioning mode. So, the CABAC contexts for Inter-coded blocks correspond to the DCT/DART decision flag bit, the most-significant bit of the DART direction index, and the remaining bits of the DART direction index.

3.2. Quantization of directional transform coefficients

In H.264/AVC, the coefficient scanning order for quantizing 8×8 and certain other transform types begins with the DC coefficient, followed by the lowest-frequency AC coefficients, and continuing through to the highest-frequency AC coefficients. With DART, we reorder the coefficients so that s_0, the DC component of the secondary transform $\boldsymbol{S}$ is quantized first. We then follow with s_m, the AC coefficients of the secondary transform. We then continue with the primary transform AC coefficients c_k^m. Thus, the coefficient scanning order is equivalent to the coefficient lists shown in (1).

It is important to note that the transform alone, as specified by H.264/AVC is not exactly the same as an integer approximation to a 2-D DCT. As described in [9], the 2-D DCT in H.264/AVC undergoes a post-scaling process to facilitate computation using integers. This scaling is incorporated into the quantizer, so together, the forward and inverse transforms and quantizers correspond to an integer 2-D DCT. If we replace the 2-D DCT with DART, the scaling used in the H.264/AVC quantizer will apply improper scaling factors to the DART coefficients. To compensate, we need to apply forward and inverse scaling. For example, using the notation of [9], the scaling for $v_{i,j}$, an $N \times N$ DART output, would be

$$v'_{i,j} = v_{i,j} \cdot \frac{(1 << \text{q\_bits})}{\text{MF} \cdot \text{QStep}}\,. \tag{2}$$

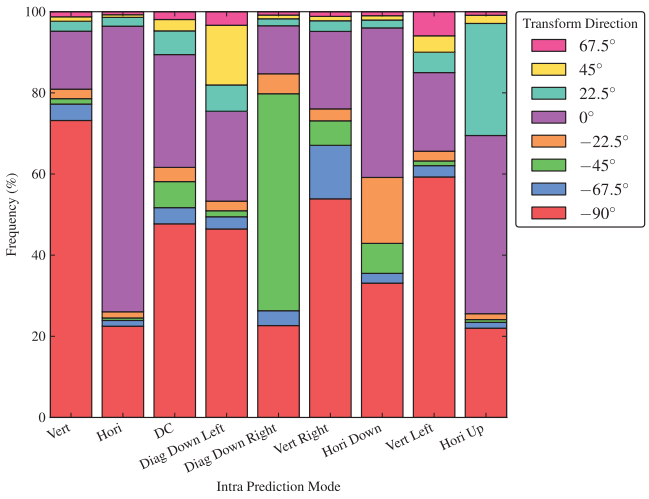


Fig. 3. Frequency of selected transform directions for Intra modes

4. EXPERIMENTAL RESULTS

For our experiments, we were interested in three aspects: The relation between the direction of the Intra-prediction mode and the optimal direction of DART, the improvement in coding efficiency using this new transform, and observations on subjective improvements.

4.1. Relation of directional transform to Intra-prediction residuals

Earlier works such as [7, 8] mention that the directionality of KLTs generated with MDDT tend to be correlated with the Intra-prediction direction. The KLTs, however, are trained on prediction residuals computed using these modes. In contrast, the structure of DART is independent of the data. To see if there is a relation between the Intra-prediction mode and the optimal directional transform, we incorporated DART into the JM 16.0 H.264/AVC reference software [11] for the 8x8 Intra transform. We coded 150 frames each of the *Foreman* and *Flower Garden* sequences using all Intra slices, with QP values of 25, 29, 33, and 37. As described earlier, H.264/AVC uses 9 Intra-prediction modes to calculate an estimate for the current block from neighboring pixel values. In Fig. 3, the frequency of the selected transform direction for each Intra-prediction mode is shown for all these simulations combined. It can be clearly seen, that direction -90° and 0° are preferable for Intra-prediction mode *Vertical* and *Horizontal*, respectively. Not only the principal directions are distinguishable, e.g. -45° dominates while $DiagonalDownRight$ is used and directions between 0° and 22.5° are preferred for *Horizontal Up*. These Simulations show that the selected transform direction d is indeed correlated with the Intra-prediction mode. Thus, we can take advantage of the Intra-prediction mode to choose the most probable transform direction. This direction can act as prediction for CABAC encoding of the transform direction as described in Section 3.

4.2. Coding performance

For evaluating coding performance, we incorporated DART into I and P picture coding in JM-KTA 2.6r1 [11]. Configuration parameters were limited to 8×8 transforms and partition modes, and the other KTA extensions were not enabled. The encoder used a rate-distortion optimized decision process to choose among the conventional transform and multiple DART directions. The picture struc-

Table 1. Coding performance, reference is JM-KTA 2.6r1

Sequence	fps	Intra Prd	Num Frs	DART IbBbP BD-Rate (%)	DART IbBbP BD-PSNR (dB)	DART IbBbP Avg BD-Rate	MDDT IbBbP BD-Rate (%)	MDDT IbBbP BD-PSNR (dB)	MDDT IbBbP Avg BD-Rate
2560x1600						**-2.40**			**-3.41**
Traffic	30	28	297	-3.23	0.14		-2.69	0.12	
PeopleOnStreet	30	28	149	-1.57	0.08		-4.13	0.21	
1920x1080						**-4.62**			**-2.00**
Kimono1	24	24	237	-2.21	0.23		-2.21	0.10	
ParkScene	24	24	237	-4.03	0.21		-1.67	0.07	
Cactus	50	48	497	-4.10	0.13		-2.77	0.08	
BasketballDrive	50	48	497	-3.54	0.12		-2.24	0.07	
BQTerrace	60	60	597	-9.20	0.20		-1.09	0.02	
1280x720						**-5.62**			**-1.53**
Spincalendar	60	60	497	-6.33	0.27		-1.07	0.04	
BigShips	60	60	149	-5.99	0.18		-0.79	0.02	
City	60	60	149	-8.95	0.44		-1.79	0.07	
Crew	60	60	149	-4.03	0.13		-2.77	0.09	
Jets	60	60	149	-3.65	0.15		-1.76	0.08	
Night	60	60	149	-4.84	0.20		-1.90	0.08	
Raven	60	60	149	-5.54	0.28		-0.65	0.03	
832x480						**-6.66**			**-1.33**
BasketballDrill	50	48	497	-6.31	0.29		-1.95	0.09	
BQMall	60	60	597	-6.40	0.34		-1.41	0.07	
PartyScene	50	48	497	-9.27	0.42		0.42	-0.02	
RaceHorses	30	28	297	-4.67	0.23		-2.37	0.12	
352x288						**-6.68**			**-0.12**
Paris	30	28	297	-7.73	0.46		2.29	-0.13	
Foreman	30	28	297	-6.35	0.33		-1.63	0.08	
Mobile	30	28	297	-6.72	0.36		-0.29	0.01	
Tempete	30	28	297	-5.94	0.27		-0.84	0.04	

ture was hierarchical-B (IbBbP), with Intra QP values of 25, 29, 33 and 37 for all resolutions except 1920×1080, which used 25, 28, 31 and 34. DART was not applied to B- or b-pictures because we found that associated performance gain was too small to be worth the additional encoding complexity. All additional side information signaled in the bit-stream is included in the reported rates. The resulting BD-Rate and BD-PSNR performance metrics [12] are shown in Table 1, where an identically configured JM-KTA 2.6r1 without DART was used as a reference. The BD-Rate gains ranged from about 1.6% to 9.3%.

As mentioned earlier, we also implemented DART in JM 16.0. When comparing this implementation to an unmodified JM 16.0, repeating these experiments yielded average BD-Rate gains of 1.0%, 1.4%, 2.0%, 4.2%, and 5.0% for the highest to lowest picture resolutions listed in Table 1, respectively. JM 16.0 contains many bug fixes and improvements that are not in JM-KTA 2.6R1, which is based upon JM 11.0.

The directional transforms also improved the subjective performance when applied to features that are aligned with the transform. Fig. 4 shows corresponding *Spincalendar* frames decoded using the H.264/AVC and DART codecs. In both cases, grain or noise present in large flat areas tends to be flattened by the quantizer. Since H.264/AVC uses the traditional 2-D DCT, grain or camera noise that is present in the input video is coded along with the sharp edges that are neither horizontal nor vertical. This is visible as noise along the center black diagonal in Fig. 4(a). With DART, however, the transform that the R-D optimization process selected is aligned with the edge, resulting in a more consistent quantization on either side of the boundary.

5. CONCLUSIONS

In this paper, we have introduced DART, a directional transform that is well-suited for coding prediction residuals by performing a set of aligned 1-D transforms, combined with path-folding of short transforms, followed by a secondary transform along only the DC coefficients. With these techniques, we improve upon earlier works which either performed no secondary transform, or performed secondary transforms across AC coefficients that had reduced correlation due to the directional nature of features unique to prediction residuals. Unlike the approaches that use the KLT, DART requires no training, and for Intra-predicted blocks, the optimal transform direction is correlated to the Intra-prediction mode. In addition to yielding gains of up to 9.3% in JM-KTA 2.6R1 and up to 7% in JM 16.0 over a wide variety of sequences, DART also provided visible improvements along strongly-oriented features in the decoded video. Proposed future research would include extending this work to larger transform and partition sizes for higher-resolution sequences.

(a) H.264/AVC, 3.43 Mbps, 33.96 dB (b) DART, 3.26 Mbps, 34.04 dB

Fig. 4. *Spincalendar* decoded frames (cropped)

6. REFERENCES

[1] "Digital compression and coding of continuous-tone still images. Requirements and Guidelines," ISO/IEC International Standard 10918-1, CCITT T.81, September 1992.

[2] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, 2003.

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

[4] B. Zeng and J. Fu, "Directional discrete cosine transforms: A new framework for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 305–313, March 2008.

[5] C.L. Chang and B. Girod, "Direction-adaptive partitioned block transform for image coding," in *Proc. Int. Conf. on Image Processing (ICIP'08)*, 2008, pp. 145–148.

[6] F. Kamisli and J. S. Lim, "Transforms for the motion compensation residual," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'09)*, 2009, pp. 789–792.

[7] M. Karczewicz, "Improved intra coding," ITU-T Video Coding Experts Group, Document VCEG-AF15, April 2007.

[8] Y. Ye and M. Karczewicz, "Improved H.264 Intra coding based on bi-directional Intra prediction, directional transform, and adaptive coefficient scanning," in *Proc. Int. Conf. on Image Processing (ICIP'08)*, San Diego, October 2008, pp. 2116–2119.

[9] I. E.G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*, John Wiley & Sons, Ltd., West Sussex, England, 2003.

[10] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264 / AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, July 2003.

[11] "H.264/AVC reference software," JM 16.1 and JM11.0KTA2.6r1, available at *http://iphome.hhi.de/suehring/tml/*.

[12] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," ITU-T Video Coding Experts Group, Document VCEG-M33, Austin, April 2001.