RATE-DISTORTION OPTIMIZED TRANSFORM FOR INTRA-FRAME CODING

Xin Zhao[1,2], Li Zhang[3], Siwei Ma[3] and Wen Gao[3]

[1]Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100049, China
[3]Institute of Digital Media, Peking University, Beijing 100871, China
{xzhao, zhanglili, swma, wgao}@jdl.ac.cn

ABSTRACT

In this paper, a novel algorithm is proposed for intra-frame coding, named as rate-distortion optimized transform (RDOT). Unlike existing intra-frame coding schemes where the transform matrices are either fixed or mode dependent, in the proposed algorithm, transform is implemented with multiple candidate transform matrices. With this flexibility, for coding each residual block, the encoder is endowed with the power to select the optimal transform matrix in terms of rate-distortion tradeoff. The proposed algorithm has been implemented in the latest ITU-T VCEG-KTA software. Experimental results show that, over a wide range of test set, the proposed method achieves average 0.43dB coding gain compared with the recent Mode-Dependent Directional Transform (MDDT). The improvement is more significant at high bit-rates, and up to 1dB coding gain can be achieved.

Index Terms—Video coding, Transforms, Karhunen-Loeve transforms.

1. INTRODUCTION

In hybrid video coding scheme, transform is employed to concentrate the energies of residual blocks onto a few coefficients, such that the subsequent processing of coefficient scanning and entropy coding can be efficiently implemented. For the good tradeoff between computational complexity and transform efficiency, discrete cosine transforms (DCTs) have been widely used in several international image/video coding standards, such as the latest H.264/AVC [1]. In H.264/AVC, a new 4×4 transform has been employed with exact integer arithmetic and featured by the low complexity. To further improve the transform efficiency in H.264/AVC, in recent years, a new transform tool named as Mode-Dependent Directional Transform (MDDT) has been proposed in [2, 3] to improve the intra coding in H.264/AVC, and significant coding gain has been achieved.

In MDDT, transforms are obtained from Karhunen-Loève transform (KLT), which is best known as non-separable transform. However, separable directional transforms are used in MDDT for complexity considerations. In the intra coding of H.264/AVC, for an intra prediction (IP) mode indexed by i, the following transform is applied in MDDT for a residual block X of size $N\times N$:

$$F = C_i \cdot X \cdot R_i \,, \qquad (1)$$

where C_i and R_i, both of size $N\times N$, are the column and row transform matrices for the IP mode indexed by i, and F denotes the resulting transform coefficient matrix, which is also of size $N\times N$. To obtain a pair of column and row transform matrices for each IP mode, Singular Value Decomposition (SVD) is used for off-line training. The training process is applied to the training set in both column and row directions, and the derived floating-point basis functions are approximated by fixed-point integers to reduce the computational cost.

Due to the mode- and data-dependent feature, MDDT efficiently improves the intra coding of H.264/AVC. However, in our simulations, it is observed that for the same IP mode, the residuals usually present various statistics. This observation indicates that there is still room to improve the transform efficiency for each IP mode, and it leads us to the idea of using multiple candidate transform matrices. In this paper, a novel algorithm is proposed based on the idea, with a complete implementation scheme for the intra coding.

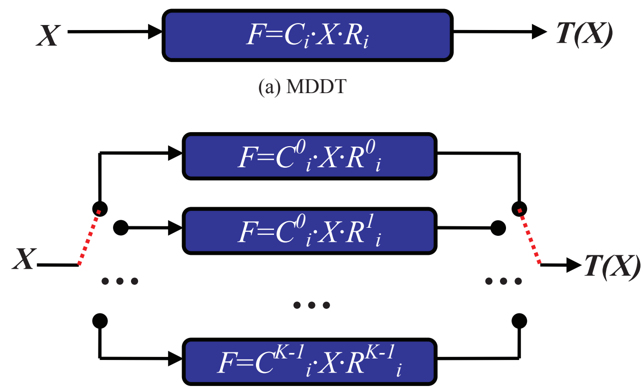


Figure 1 Transform schemes of a residual block *X* for IP mode *i* in (a) MDDT and (b) proposed RDOT.

The remainder of this paper is organized as follows. The proposed RDOT technique is introduced in Section 2, and Section 3 presents the experimental results. Finally, the paper is concluded in Section 4.

2. THE PROPOSED RATE-DISTORTION OPTIMIZED TRANSFORM

In this section, all the technical details including the implementation, training process and coding of side-information for the proposed algorithm are discussed. Finally, we give a brief analysis on the complexity of the proposed algorithm.

2.1. Observations, algorithm and implementation

The two matrices shown below are two residual blocks obtained from an actual coding process,

$$\begin{bmatrix} 21 & -2 & -3 & -4 \\ 21 & -2 & -3 & -4 \\ 26 & -1 & -3 & -4 \\ 22 & -1 & -2 & -4 \end{bmatrix}, \begin{bmatrix} 7 & -16 & -14 & 14 \\ 7 & -22 & -16 & 11 \\ 13 & 1 & 0 & 3 \\ 44 & 40 & 9 & -7 \end{bmatrix}.$$

Both blocks are obtained from the same IP mode indexed by 0, but very different statistics are presented. The left block contains a vertical edge, but the texture of the right block is irregular. This motivating observation demonstrates the limitation of existing MDDT algorithm that even in the same IP mode, one fixed pair of transform matrices can not handle all the residual blocks efficiently. Therefore, we propose to use multiple candidate transform matrices to further improve the transform efficiency.

To illustrate the key differences between MDDT and RDOT, the transform schemes of both algorithms have been shown in Fig. 1. As it is shown in Fig. 1, for each IP mode i, the transform matrices are specified as C_i and R_i in MDDT. However, in the proposed RDOT, for each IP mode i, there are a group of candidate transform matrices C^k_i and R^k_i, where $k=0, 1\ldots, K-1$. And the optimal path, which is denoted by the red dotted line in Fig. 1(b), is selected using rate distortion optimization (RDO) criteria for the actual video coding. Another important difference between MDDT and RDOT lies in the fact that, in RDOT, the indexes of the transform matrices has to be sent to the decoder side, such that the decoding process can be correctly implemented using the right transform matrix without mismatch.

In our simulations, it is observed that for some smooth regions, RDOT becomes unnecessary because the benefits can be covered by the overheads. However, for textured regions, RDOT becomes advantageous due to the great improvements on transform efficiency. Motivated by the above observations, we implement the proposed algorithm as a combination of RDOT and MDDT, that is, besides the original macroblock coding modes I4MB, I8MB and I16MB which employs MDDT, three additional modes I4MB_RDOT, I8MB_RDOT and I16MB_RDOT based on RDOT are added. And the optimal coding mode is selected using the RDO criteria. Furthermore, in each IP mode, we use specified coefficient scanning order obtained by off-line training for each combination of column and row candidate transform matrices.

Input:
1. Training set T, of size $M\times N$, where M denotes the number of training vectors, N=4 for I4MB and N=16 for I16MB.
2. Number of transform matrices K.
3. Number of iterations t.

Output:
K candidate transform matrices.

(0) Initialization:
Randomly label each residual vector with one of the indexes of candidate transform matrices; i=0.

(1) Iteration:
Step I: For each of the K classes of residual vectors, calculate the SVD and record the K transform matrices as B_0^i, B_1^i, …, B_{K-1}^i;
Step II: For each of the training vectors, use B_0^i, B_1^i, …, B_{K-1}^i to redo classification, i=i+1;
If i=t, output B_0^i, B_1^i, …, B_{K-1}^i as the final K candidate transform matrices, else go to Step I.

Figure 2 Training process of the candidate transform matrices.

In the above implementation scheme, there are two important problems need to be handled that influence the performance of RDOT:

1. How to obtain the candidate transform matrices?
2. How to encode the transform matrix indexes?

In subsection 2.2 and 2.3, we will discuss in detail how the above two problems are solved in our implementation.

2.2. Off-line training

During our experiments, it is found that the performance of RDOT is evidently influenced by the set of candidate transform matrices. Unfortunately, the Singular Value Decomposition (SVD) method used in [2, 3] for off-line training can not be directly employed, because in RDOT, we intend to obtain K candidates for both column and row transform matrices instead of one in MDDT. To resolve this difficult problem, we have designed an iterative approach, as shown in Fig.2, to obtain the K candidate transform matrices for both column and row transforms.

As shown in Fig.2, the iteration process is implemented by two steps. At the first step, the K basis functions are derived by applying SVD for each class of residual vectors, at the second step, the residual vectors are re-classified using the K transform matrices obtained at the first step. In the second step, for each residual vector, the classification rule is to select the optimal transform matrix in terms of transform efficiency, and re-label it with the index of the optimal transform matrix. Note that, there are different criterions on measuring the transform efficiency, and in our simulations, we simply use the magnitude of the principal component, i.e., the first transform coefficient, as the criterion of measuring the transform efficiency. This criterion is based on an observation that larger magnitude of the principal component usually implies faster energy decay in the transform domain. For example, let v=[-1, 6, -33, -80] be a residual vector, and the transform matrix set includes

$$K_0 = \begin{bmatrix} -0.0131 & -0.2244 & 0.7531 & 0.6184 \\ 0.1679 & -0.6912 & 0.3135 & -0.6290 \\ 0.5480 & -0.4992 & -0.5036 & 0.4438 \\ 0.8194 & 0.4719 & 0.2846 & -0.1580 \end{bmatrix}, \quad K_1 = \begin{bmatrix} 0.4026 & 0.5737 & 0.5047 & 0.5041 \\ 0.6027 & 0.4014 & -0.3345 & -0.6031 \\ 0.5249 & -0.3951 & -0.5173 & 0.5484 \\ 0.4464 & -0.5947 & 0.6048 & -0.2852 \end{bmatrix}.$$

By applying the transform on v, i.e., calculating $v \cdot K_0$ and $v \cdot K_1$, we obtain the resulting transform coefficient vectors f_0=[-82.6119, -25.2018, -5.0204, -6.3986] and f_1=[-49.8174, 62.4495, -33.8221, 0.5941]. It is observed that, for the particular residual vector v, K_0 is more efficient than K_1 because $f_0[0]$ is larger than $f_1[0]$. Therefore, in Step II, the residual vector v is re-labeled with the index of K_0, i.e., 0.

In our simulations, for each IP mode, we allow 2, 4 and 4 candidates for both column and row transform matrices in I4MB, I8MB and I16MB, respectively. The numbers of candidate transform matrices are based on the empirical experimental results. The iteration num t in the training process is set as 16, 32 and 64 for I4MB, I8MB and I16MB, respectively. Also an early termination is allowed if the iteration becomes converged, i.e., there is no residual vectors changing the labels in Step II.

2.3. Syntax change and coding of the side information

For each individual macroblock, a one-bit flag is set in the macroblock header to indicate whether this macroblock is coded using RDOT. If the flag is set as 1, i.e., RDOT is used, the transform matrix indexes of every block will be also coded into the macroblock header. In our simulations, context-based adaptive binary arithmetic coding (CABAC) [4] is used for coding the new syntax elements including the flag and transform matrix indexes.

To employ the CABAC entropy coding engine for coding the new syntax elements in RDOT, a proper context model need to be selected. Note that, the optimal transform matrix is directly correlated with the statistics of a residual block, therefore in our implementation, we take advantage of the neighboring transform matrix indexes to the left and on the top of current block as the context model.

2.4. Complexity analysis

The major differences between MDDT and RDOT in the implementation complexity are twofold:

a. Compared with MDDT, in RDOT, both the encoder and decoder need to be allocated with additional memories to store the candidate transform matrices.
b. Compared with MDDT, in RDOT, the encoder has to do additional computations to select the optimal candidate transform matrices for each residual block.

For the first item, in each IP mode, there are 2, 4 and 4 candidate matrices for both column and row transform in the I4MB, I8MB and I16MB coding modes, respectively. Each component of the candidate matrices is approximated by a one-byte integer. And we totally need additional 13.376KB of memories for storing the candidate transform matrices in both encoder and decoder.

In the proposed RDOT, the computation complexity is increased for the encoder, because the encoder needs to check all the possible combinations of candidate matrices exhaustively and select the optimal one in terms of rate-distortion tradeoff. In the current implementation, we have made some optimization to accelerate the encoding process for the proposed scheme. For the decoder, where the complexity is a very critical issue, RDOT induces nearly no additional computation burden.

3. EXPERIMENTAL RESULTS

The proposed RDOT has been implemented on the latest ITU-T VCEG-KTA software [5] version 2.6r1. Experiments are performed with MDDT on, all coded frames are intra frames, RDO-Q is disabled and CABAC is used for entropy coding. Furthermore, we use two different sets of QP values in the experiments: QP1={22, 27, 32, 37} and QP2={17, 21, 25, 29}, where QP2 indicates a high bit-rate coding configuration. The test set includes 10 sequences with QCIF format and 11 test sequences with CIF format.

Comparisons are made between the original KTA with MDDT turned on and the improved KTA integrated with our proposed RDOT. The Bjontegaard PSNR (BD-PSNR) [6] gains of RDOT compared to MDDT are tabulated in Table 1. From Table 1 it can be seen that over a wide range of test set, our proposed RDOT achieves average 0.43dB coding gain for QP1 and the maximum coding gain achieves 0.74dB. For QP2 which indicates a high bit-rate coding configuration, the coding gain is more significant and average 0.62dB can be reached, with a maximum 1 dB coding gain. The rate-distortion performance for several sequences is also shown in Fig.3. From Fig.3, it is observed that our proposed RDOT outperforms MDDT in the full range of QP set, and the gains become more significant at high bit-rates. For complexity comparisons, the encoding and decoding time for several sequences is tabulated in Table 2. The average encoding time of the proposed scheme is near 3 times larger than MDDT, and the average decoding time of the proposed scheme is close to the decoding time of MDDT.

4. CONCLUSIONS AND FUTURE WORK

In this paper, a novel transform scheme is proposed for intra-frame coding. With the proposed algorithm, for coding each residual block and prediction mode, the encoder selects the optimal transform matrix from multiple candidates in terms of rate-distortion tradeoff. Compared with the MDDT, the proposed algorithm achieves average 0.43dB coding over a wide range of test set. The improvements become more significant for high bit-rate coding, and up to 1 dB coding gain can be achieved.

In MDDT, only the transforms in intra frame coding are considered and improved, and in our future work, RDOT for inter coding will also be investigated. Also the fast candidate transform matrix selection strategy will be explored to accelerate the encoding process.

11. REFERENCES

[1] Iain E. G. Richardson, *H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia*, Chichester: Wiley, 2003.

[2] Y. Ye and M. Karczewicz, "Improved h.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," *Proceedings of the 2008 IEEE International Conference on Image Processing (ICIP)*, pp. 2116-2119, Oct. 2008.

[3] Y. Ye and M. Karczewicz, "Improved intra coding," *ITU-T Q.6/SG16 VCEG*, *VCEG-AG11*, Shenzhen, China, Oct. 2007.

[4] 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.

[5] ITU-T VCEG. version 2.4. *Reference ITU-T VCEG-KTA Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA/

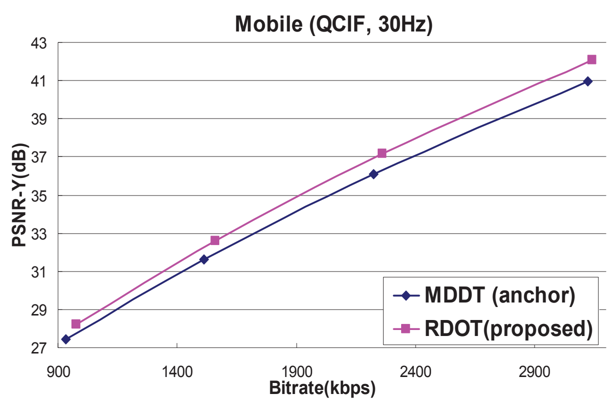
[6] G. Bjontegaard, "Calculation of Average PSNR Differences between RD curves," *ITU-T SG16/Q6*, *13th VCEG Meeting*, *Doc. VCEG-M33*, Austin, Texas, USA, Apr. 2001.

Table 1 Coding gain of RDOT compared with MDDT for intra

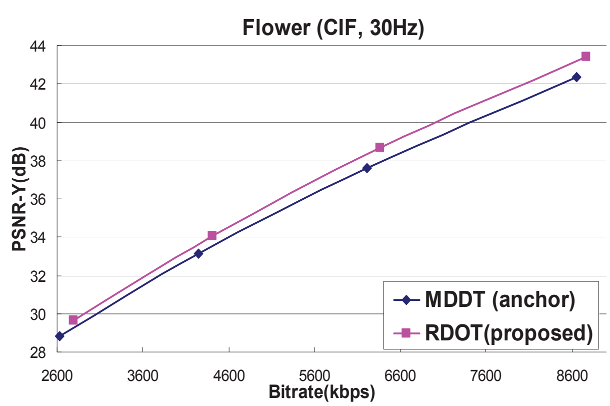
Sequence		ΔBPSNR (dB) QP1={22,27,32,37}	ΔBPSNR (dB) QP2={17,21,25,29}
QCIF	Bus	0.4593	0.6491
	Football	0.4229	0.5174
	Tempete	0.4909	0.7595
	Coastguard	0.4292	0.6104
	Container	0.4521	0.6496
	Hall	0.3679	0.6772
	Paris	0.4913	0.7050
	Mobile	0.7350	1.0030
	Silent	0.2411	0.4079
	Foreman	0.2632	0.4495
CIF	Flower	0.6028	0.8357
	Mobile	0.7012	0.9096
	Paris	0.3895	0.5653
	Stefan	0.5181	0.7374
	Bus	0.4650	0.5902
	Coastguard	0.3642	0.4652
	Container	0.4918	0.5733
	Hall	0.3352	0.5631
	Foreman	0.2387	0.4238
	Tempete	0.3035	0.4968
	News	0.3109	0.4544
AVERAGE		**0.4321**	**0.6211**

Table 2 Comparisons of encoding and decoding time between MDDT and the proposed scheme for intra coding

Sequence	QP	Encoding Time (sec) Anchor	Encoding Time (sec) Proposed	Decoding Time (sec) Anchor	Decoding Time (sec) Proposed
Bus 150 frames QCIF	22	113.732	330.217	1.672	1.828
	27	92.984	286.718	1.422	1.563
	32	78.532	234.279	1.250	1.390
	37	67.028	191.852	1.078	1.250
Hall 300 frames QCIF	22	175.894	521.040	2.672	3.125
	27	153.248	439.195	2.469	2.843
	32	134.482	379.561	2.281	2.640
	37	121.092	330.829	2.125	2.453
Paris 300 frames CIF	22	875.737	2513.684	12.312	13.609
	27	748.516	2161.259	10.953	11.953
	32	648.253	1845.605	9.765	11.094
	37	567.428	1563.952	9.500	10.265
News 300 frames CIF	22	675.037	1912.247	9.781	10.969
	27	595.176	1656.637	9.109	10.140
	32	540.946	1468.922	8.688	9.594
	37	498.729	1314.510	8.047	9.250
Average T_{pro}/T_{anc}	**22**	**2.89**		**1.12**	
	27	**2.91**		**1.11**	
	32	**2.84**		**1.13**	
	37	**2.75**		**1.14**	



(a) Mobile (QCIF, 30fps, QP={22, 27, 32, 37}, All Intra)



(b) Flower (CIF, 30fps, QP={22, 27, 32, 37}, All Intra)

Figure 3 RD curves for intra coding using MDDT and proposed scheme, (a) Mobile (QCIF, 30fps), (b) Flower (CIF, 30fps).