Efficient MPEG-2 to H.264/AVC Intra Transcoding in Transform-domain

Yeping Su, Jun Xin, Anthony Vetro, Huifang Sun

TR2005-039 May 2005

Abstract

In this paper, we propose a transform-domain MPEG-2 to H.264/AVC intro video transcoder. In the transcoder, the input DCT coefficients are first converted to H.264 transform (HT) coefficients entirely in the transform-domain. Rate-distortion optimized macroblock mode decision is then performed based on the HT coefficients. Finally, the HT coefficients are coded using the selected modes to generate the output H.264 bitstream. The proposed transcoder is equivalent to the conventional pixel-domain implementation in terms of functionality, but its complexity is sifnificantly lower (on average over 20%). To further reduce its computational complexity, we propose a fast mode decision algorithm based on a simple cost function calculated in the HT-domain. This fast algorithm reduces the complexity requirement about 50%, while maintains virtually the same coding efficiency.

IEEE International Symposium on Circuits and Systems (ISCAS)

Efficient MPEG-2 to H.264/AVC Intra Transcoding in Transform-domain

Yeping Su
University of Washington EE Department
Seattle WA 98195, USA
su@ee.washington.edu

Jun Xin, Anthony Vetro, Huifang Sun
Mitsubishi Electric Research Laboratories
Cambridge MA 02139, USA
{jxin, avetro, hsun}@merl.com

***Abstract*—In this paper, we propose a transform-domain MPEG-2 to H.264/AVC intra video transcoder. In the transcoder, the input DCT coefficients are first converted to H.264 transform (HT) coefficients entirely in the transform-domain. Rate-distortion optimized macroblock mode decision is then performed based on the HT coefficients. Finally, the HT coefficients are coded using the selected modes to generate the output H.264 bitstream. The proposed transcoder is equivalent to the conventional pixel-domain implementation in terms of functionality, but its complexity is significantly lower (on average over 20%). To further reduce its computational complexity, we propose a fast mode decision algorithm based on a simple cost function calculated in the HT-domain. This fast algorithm reduces the complexity requirement about 50%, while maintains virtually the same coding efficiency.**

I. INTRODUCTION

The latest video-coding standard, known as H.264 (or MPEG-4 AVC) [1], has been developed collaboratively by the Joint Video Team of ISO/IEC MPEG and ITU-T VCEG. At an equivalent perceptual quality, AVC provides a bit-rate saving of about 50% relative to prior standards such as MPEG-2. Due to its superior compression efficiency, it is expected to have a wide range of applications, including mobile broadcasting and storage. With the deployment of H.264, e.g., for mobile broadcasting, there is a need to convert video in the MPEG-2 format to videos of the H.264 format. This would enable more efficient network transmission and storage.

As shown in Figure 1, a transcoder may first decode an input MPEG video to reconstruct the image pixels, and then encodes them in the H.264 format. This is referred to as a Pixel-Domain Transcoder (PDT). Note that we discuss intra-transcoding only in this paper.

It is well known that transform domain techniques may be simpler since they eliminate the need of inverse transform and transform. However, in the case of MPEG-2 to H.264 transcoding, the transform-domain approach must efficiently solve the following two mismatch problems:

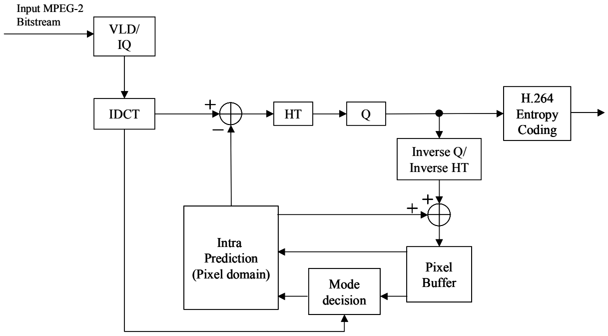


Figure 1. Pixel-domain intra transcoding architecture

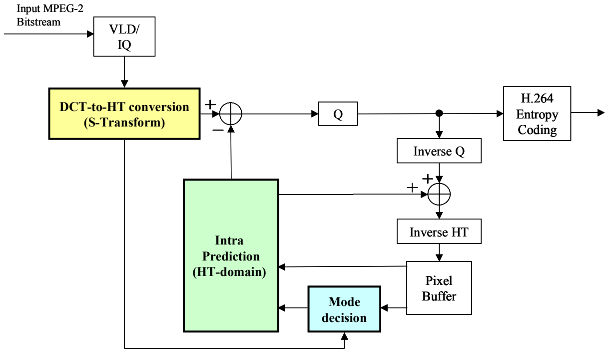


Figure 2. Tranform-domain intra transcoding architecture

- Transform mismatch: an efficient algorithm for DCT-to-HT coefficient conversion. It must be simpler than the trivial concatenation of IDCT-HT.
- Mode mismatch: efficient H.264 mode decision in transform-domain (HT-domain).

Figure 2 illustrates our proposed transform-domain transcoder, with the problems highlighted. As will be explained in this paper, both mismatch problems are solved more efficiently in the transform-domain than in the pixel-domain. Note that the intra prediction consumes little computation since it can reuse the result of the mode decision.

In the remainder of this paper, we discuss how these two problems are solved in our proposed novel transform-domain transcoder. We first discuss in Section II the efficient DCT-to-HT conversion. Then in Section III, we discuss the transform-domain mode decision and a fast algorithm for further speedup. Simulation results are provided in Section IV before Section V concludes this paper.

II. EFFICIENT DCT-TO-HT CONVERSION

The DCT-to-HT conversion can be shown to be a 2D linear transform [2], hereinafter referred as S-Transform. Let X denote an 8x8 block of DCT coefficients, the corresponding HT coefficient block Y, consisting of four 4x4 HT blocks, is give by

$$Y = S \times X \times S^T \tag{1}$$

The kernel matrix S is

$$S = \begin{pmatrix} a & b & 0 & -c & 0 & d & 0 & -e \\ 0 & f & g & h & 0 & -i & -j & k \\ 0 & -l & 0 & m & a & n & 0 & -o \\ 0 & p & j & -q & 0 & r & g & s \\ a & -b & 0 & c & 0 & -d & 0 & e \\ 0 & f & -g & h & 0 & -i & j & k \\ 0 & l & 0 & -m & a & -n & 0 & o \\ 0 & p & -j & -q & 0 & r & -g & s \end{pmatrix} \tag{2}$$

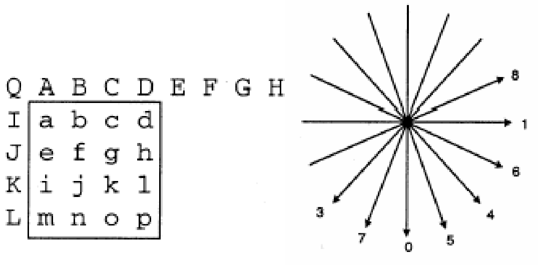
$a = 1.4142$, $b = 1.2815$, $c = 0.45$, $d = 0.3007$, $e = 0.2549$, $f = 0.9236$, $g = 2.2304$, $h = 1.7799$, $i = 0.8638$, $j = 0.1585$, $k = 0.4824$, $l = 0.1056$, $m = 0.7259$, $n = 1.0864$, $o = 0.5308$, $p = 0.1169$, $q = 0.0922$, $r = 1.0379$, $s = 1.975$.

The symmetry of the kernel matrix can be utilized to design fast implementations of the transform. In this work, we use the fast integer algorithm proposed in [2], which saves about 30% of operations compared with IDCT followed by HT. For the proof and more details of the fast algorithms, please refer to [2].

III. TRANSFORM-DOMAIN MODE DECISION

A. Conventional Mode Decision

Let us consider the conventional H.264 pixel-domain mode decision (as in JM76), and in particular, the Rate-Distortion Optimized (RDO) decision for the Intra_4x4 modes. Figure 3(a) illustrates the candidate neighboring pixels “A-Q” used for prediction of current 4x4 block pixels “a-p”. Figure 3(b) illustrates the eight directional prediction modes. In addition, DC prediction (DC_Pred) can also be used.



(a) (b)

Figure 3. (a) Neighboring samples “A-Q” are used for prediction of samples “a-p”. (b) Prediction mode directions (except DC_Pred).

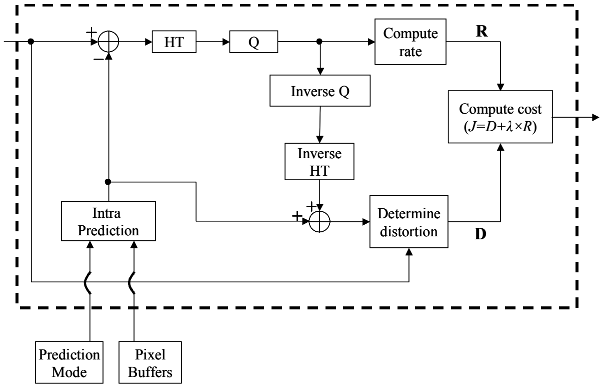


Figure 4. Pixel-domain RD cost computation

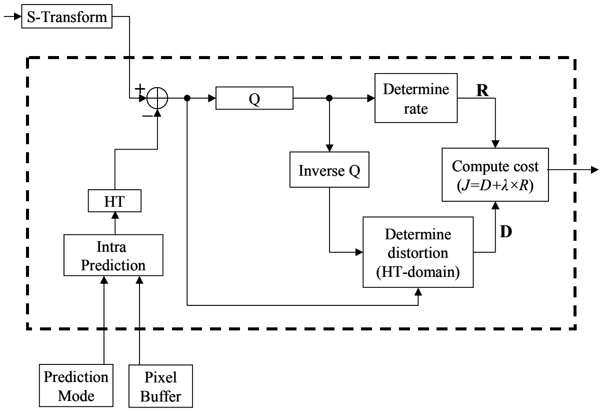


Figure 5. Transform-domain RD cost computation

Consider the rate distortion calculation in a video encoder with *RDO_on*, the conventional calculation of the Lagrange cost for one coding module (in this case for one 4x4 luma block) is shown in Figure 4. The prediction residual is transformed, quantized and entropy encoded to determine the rate, $R(m)$, for a given mode m. Then, inverse quantization, inverse transform are performed and then compensated with the prediction block to get the reconstructed signal. The distortion, denoted $SSD_{REC}(m)$, is computed as the sum of squared distance between the original block, s, and the reconstructed block $\tilde{s}(m)$:

$$SSD_{REC}(m) = \left\| s - \tilde{s}(m) \right\|_2^2 \tag{3}$$

where $\left\| \cdot \right\|_p$ is the Lp-norm. The Lagrange cost is computed using the rate and distortion as follows:

$$Cost_{4x4} = SSD_{REC}(m) + \lambda_{MODE} R(m) \quad (4)$$

where λ_{MODE} is a Lagrange multiplier, which may be calculated as a function of the quantization parameter. The optimal coding mode corresponds to the mode with the minimum cost.

Besides this RDO mode selection, a low-complexity algorithm (i.e., with *RDO_off*) would only calculate the sum of absolute distance of the Hadamard-transformed prediction residual signal:

$$SATD(m) = \| T(s - \hat{s}(m)) \|_1 \quad (5)$$

where $\hat{s}(m)$ is the prediction signal for the mode *m*. In this case, the cost function would then be given by

$$Cost_{4x4} = SATD(m) + \lambda_{MODE} * 4 * (1 - \delta(m = m^*)) \quad (6)$$

where m^* is the most probable mode for the block.

B. Transform-domain Mode Decision

The proposed Transform-domain mode decision calculates the Lagrange cost for each mode according to Figure 5, which is based on our previous work on H.264 encoding [3]. Compared to the pixel-domain approach, the transform-domain implementation has several major differences in terms of computation involved:

- It saves one inverse HT computation for each candidate prediction mode, since the distortion is determined using the reconstructed and original *residual* HT coefficients.
- Instead of operating on the prediction residual pixels, the HT now operates on the prediction signals. HT of some intra-prediction signals is very simple to compute [3]. For example, for DC_Pred mode, there is only one nonzero DC element in the transformed prediction signal $HT(\hat{S}(DC\_Pred))$. Therefore, additional computational saving is achieved.

Note that the distortion is calculated in the HT-domain. HT is not an orthonormal transform; hence, it does not preserve the *L2* norm (energy). However, the distortion can still be calculated with proper coefficient weighting [3].

Let $\tilde{s} = s + \tilde{e}$ denote the reconstructed signal, where $\tilde{e}$ is the reconstructed residual signal. The pixel-domain distortion, $SSD_{REC}(m)$, is given by (3). The same distortion as can be computed in the HT-domain as [3]:

$$D_{HT}(m) = \left\| (E - \tilde{E}(m) \otimes W_2) \otimes W_1 \right\|_2^2 \quad (7)$$

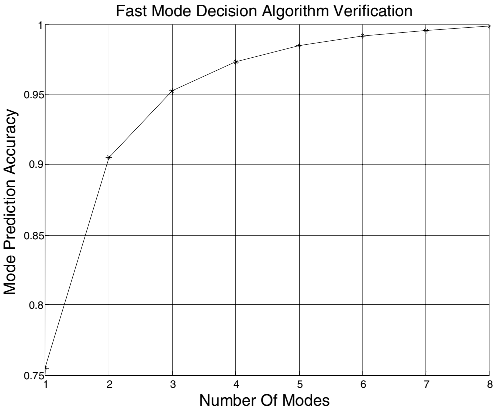


Figure 6. Number of Test Modes vs. Accuracy.

$$W_1 = \begin{bmatrix} \frac{1}{4} & \frac{1}{\sqrt{40}} & \frac{1}{4} & \frac{1}{\sqrt{40}} \\ \frac{1}{\sqrt{40}} & \frac{1}{10} & \frac{1}{\sqrt{40}} & \frac{1}{10} \\ \frac{1}{4} & \frac{1}{\sqrt{40}} & \frac{1}{16} & \frac{1}{\sqrt{40}} \\ \frac{1}{\sqrt{40}} & \frac{1}{10} & \frac{1}{\sqrt{40}} & \frac{1}{10} \end{bmatrix} \quad W_2 = \frac{1}{64} \begin{bmatrix} 16 & 20 & 16 & 20 \\ 20 & 25 & 20 & 25 \\ 16 & 20 & 16 & 20 \\ 20 & 25 & 20 & 25 \end{bmatrix}$$

where E and $\tilde{E}$ are HT transformed residual signal and reconstructed residual signal. Note that there are two scaling factor matrixes, W_1 and W_2, to compensate for the different norms of the HT, the inverse HT and the H.264 quantization design. The $\otimes$ operator represents a scalar multiplication or element-wise multiplication of two matrices. For further details, please refer to [3].

C. Fast Intra Mode Decision

When operating at high-complexity mode, the H.264 coder utilizes Lagrange coder control to optimize mode decisions in the rate-distortion sense. At low-complexity mode, the SATD cost in (6) requires much simpler computation, but the coding performance is not as good because the cost function is only an approximation of the actual RD cost given by (4). The proposed fast intra mode decision algorithm is based on the following observation: although choosing the mode with the smallest SATD value often misses the best mode in the RD sense, the best mode usually contains smaller SATD cost. In other words, the mode rankings according to the two cost functions are highly correlated.

The basic idea is to rank all candidate modes using the easy-to-compute SATD costs, and then evaluate Lagrange RD costs only for the few best modes decided by the ranking. Based on the input HT coefficients of prediction residual signal, the algorithm is described in the following.

First, we compute the HT-domain $cost_1$ for all candidate modes based on normalized HT-domain residual coefficients:

$$cost_1(m) = \left\| (S - \hat{S}(m)) \otimes W_1 \right\|_1 + \lambda_{\mathrm{MODE}} * 4 * (1 - \delta(m = m^*)) \tag{8}$$

Then, we sort the modes according to $cost_1$ in ascending order, putting the first k smallest modes in the test set $\boldsymbol{T}$. Next, we add *DC_Pred* into $\boldsymbol{T}$ if it is not in the set already. For the modes in $\bar{\boldsymbol{T}}$, compute

$$cost_2(m) = \left\| (E - \tilde{E}(m) \otimes W_2) \otimes W_1 \right\|_2^2 + \lambda_{MODE} * R(m) \tag{9}$$

We finally select the best mode according to $cost_2(m)$.

Note that in calculating (6), instead of using Hadamard transform, the distortion SATD is defined as the SAD of HT coefficients since they are already available in the transform-domain transcoder. The parameter k controls the complexity-quality trade-off. To verify the correlations between rankings using $cost_1$ and $cost_2$, a simple experiment is performed. We collect the two costs for all luma 4x4 blocks in the first frame of all 5 CIF test sequences coded with QP=28, and then count the percentage of times when the best mode according to $cost_2$ is in the test set $\boldsymbol{T}$. This is called the mode prediction accuracy. The results are plotted in Figure 6 as k vs. accuracy. The strong correlation between the two costs is evident in the high accuracies shown. In this work, k is set to be 3.

IV. SIMULATION RESULTS

The proposed transcoder is implemented based on MSSG MPEG-2 software codec [4] and H.264 JM7.6 reference code [5]. All simulations are performed using Windows XP, Intel P4 CPU, Intel C++ Compiler 7.0, 30 frames/s CIF test sequences (100 frames each), RDO on, and UVLC entropy coding. The sequences are all intra-encoded at 6Mbps.

Three transcoders are evaluated: Pixel Domain Transcoder (PDT), Transform Domain Transcoder (TDT), and Transform Domain Transcoder with Fast intra mode decision (TDT-F). The performance is measured using PSNR between the output H.264 video and the corresponding raw video. The computational complexity is measured using the runtime of the entire transcoder.

Tables 1-2 show the RD performances of the three transcoders. Table 3 shows the complexity reductions of TDT/TDT-F over PDT. TDT achieves virtually the same RD performance as PDT. The computational saving of TDT over PDT is typically more than 20%. When the fast intra mode decision algorithm is used, TDT-F saves around 50% of the computation, with a negligible PSNR degradation of less than 0.1dB for all test cases.

V. CONCLUSIONS

We proposed a transform-domain MPEG-2 to H.264 intra video transcoder. The transform-domain architecture is equivalent to the conventional pixel-domain implementation in terms of functionality, but it has significantly lower complexity. We achieved the complexity reduction by taking advantage of direct DCT-to-HT coefficient conversion and transform-domain mode decision. We also presented a fast intra mode decision algorithm utilizing transform-domain features that can further reduce its computational complexity. Simulations show that we can achieve significant computational reduction without sacrificing video quality.

TABLE I. RD PERFORMANCE COMPARISONS WITH QP=30

		Akiyo	**Foreman**	**Container**	**Mobile**	**Stefan**
PDT	*Bit-rate*	1253.20	1659.48	2213.60	5608.50	3807.39
	PSNR	38.63	35.84	34.77	29.31	33.26
TDT	*Bit-rate*	1258.04	1654.16	2207.19	5593.96	3795.78
	PSNR	38.59	35.83	34.75	29.30	33.25
TDT-F	*Bit-rate*	1257.72	1656.48	2208.46	5587.70	3789.77
	PSNR	38.59	35.82	34.72	29.26	33.19

(Bit-rate unit: kbps, PSNR unit: dB.)

TABLE II. RD PERFORMANCE COMPARISONS WITH QP=45

		Akiyo	**Foreman**	**Container**	**Mobile**	**Stefan**
PDT	*Bit-rate*	495.55	558.81	601.09	1324.62	909.97
	PSNR	28.00	27.02	25.57	21.97	23.35
TDT	*Bit-rate*	495.23	559.60	601.07	1323.52	910.60
	PSNR	27.93	27.04	25.54	21.95	23.35
TDT-F	*Bit-rate*	496.14	561.50	602.77	1327.36	914.96
	PSNR	27.83	27.03	25.57	21.96	23.37

(Bit-rate unit: kbps, PSNR unit: dB.)

TABLE III. COMPLEXITY REDUCTIONS (%) RELATIVE TO PDT

		Akiyo	**Foreman**	**Container**	**Mobile**	**Stefan**
TDT	*QP=30*	22.3%	22.5%	19.3%	17.9%	20.4%
	QP=45	24.0%	24.6%	19.1%	23.1%	22.5%
TDT-F	*QP=30*	49.8%	49.8%	49.9%	50.9%	51.5%
	QP=45	51.7%	51.5%	49.4%	48.4%	49.9%

REFERENCES

[1] ITU-T Rec. H.264 | ISO/IEC 14496-10, "Advanced Video Coding," 2003.

[2] J. Xin, A. Vetro, and H. Sun, "Converting DCT Coefficients to H.264/AVC Transform Coefficients," Pacific-rim Conference on Multimedia (PCM) 2004.

[3] J. Xin, A. Vetro, and H. Sun, "Efficient Macroblock Coding Mode Decision for H.264/AVC Video Coding," Picture Coding Symposium (PCS) 2004.

[4] MPEG-2 video codec v12, available online at http://www.mpeg.org/MPEG/MSSG.

[5] H.264/AVC Reference Software JM7.6, available online at http://bs.hhi.de/~suehring/tml/download/.