Fast Inter Prediction for H.264 to HEVC Transcoding

Zong-Yi Chen, Chi-Teng Tseng, and Pao-Chi Chang[1]

Abstract. Owing to the high compression efficiency improvement of High Efficiency Video Coding (HEVC) standard, it is expected as a successor to H.264. The widely usage of H.264 today and the expected future adoption of HEVC raises a new demand for H.264 to HEVC transcoding. An efficient transcoding way is necessary. In this paper, the proposed pixel domain transcoding architecture reuses the coding information such as motion vector, prediction mode and residual coefficient extracted from the H.264 decoder. An adaptive search range algorithm for motion estimation and several CU/PU pruning strategies for mode decision are proposed to speed up the encoder. The experimental results show that our proposed transcoding scheme can maintain the coding performance and save about 40% encoding time.

Keywords: H.264, HEVC, transcoding, mode decision, adaptive search range.

1 Introduction

With the improvement of video coding technology, storage capacity, display resolutions, and CPU processing capabilities, the applications of multimedia systems become wider and more popular. Therefore, how to efficiently provide video contents to users under different constraints is very important. Video transcoding is one of the efficient solutions to this problem. The transcoding technique provides video adaption in terms of bitrate reduction, resolution reduction, and format conversion.

H.264 video compression standard has achieved great success in real life applications. It has been widely used for video streaming and video storage. Even Bluray Disc (BD) adopts H.264 as its specifications. However, the next generation of video compression standard HEVC is completed in 2013. Under the same subjective quality, HEVC can save about 50% bitrate compared with H.264 [1]. Thus,

[1] Pao-Chi Chang (✉)
Department of Communication Engineering, National Central University, Jhongli, Taiwan
e-mail: pcchang@ce.ncu.edu.tw

HEVC can be expected as a successor to H.264 and the demand for transcoding from H.264 to HEVC arises.

The basic concept of video transcoding is to convert a compressed video bitstream to another representation. Video transcoding can be categorized into homogeneous transcoding and heterogeneous transcoding. Homogeneous video transcoding converts original video to the same video standard but with different bitrates, resolutions, or frame rates. Heterogeneous video transcoding converts a video to another video standard. In this work, we focus on the heterogeneous transcoding from H.264 to HEVC. The simplest transcoder is constructed by cascading a decoder which is for the input bitstream with an encoder which outputs the target bitstream. This architecture is usually referred to cascaded pixel domain transcoder (CPDT) [2]. In general, a large amount of encoding time is consumed on motion estimation (ME) and mode decision (MD) in a video encoder. However, a full encoding process including ME and MD is performed in CPDT architecture. An efficient way to reduce the complexity during transcoding is required. A typical transcoder from H.264 to HEVC with fast encoding schemes is shown in Fig. 1.

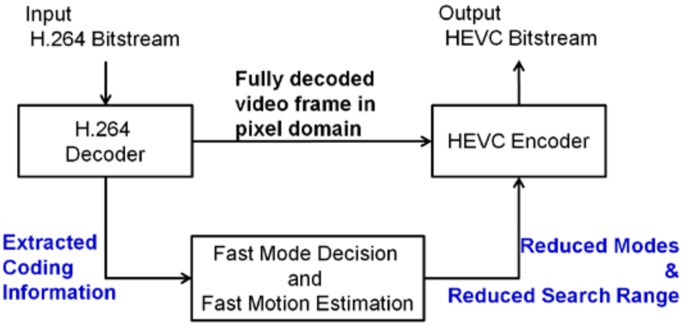

Input
H.264 Bitstream
Output
HEVC Bitstream
Fully decoded video frame in pixel domain
H.264 Decoder
HEVC Encoder
Extracted Coding Information
Fast Mode Decision and Fast Motion Estimation
Reduced Modes & Reduced Search Range

Fig. 1 Pixel domain H.264 to HEVC transcoder

Many works had been proposed [3]-[5] by reusing the decoder information such as motion vector (MV) or mode information to reduce the transcoding complexity for previous standards. Most of the works are developed in the case of the same basic coding block size, i.e. 16x16, between standards. However, the structure and size of prediction block in HEVC are much different from H.264. The decoded information mapping should be refined to adapt to HEVC. A complexity-scalable [6] and a fast algorithm based on power spectrum rate-distortion optimization [7] are proposed for H.264 to HEVC transcoder. In this work, we propose an adaptive search range selection algorithm and a CU/PU candidates reduction method for H.264 to HEVC transcoding.

The rest of this paper is organized as follows. Section 2 briefly introduces HEVC. Section 3 and section 4 respectively present the proposed fast ME and MD algorithms in detail. Simulation results are shown in section 5 and conclusions are drawn in section 6.

2 Overview of HEVC

In HEVC, the encoder is composed of coding unit (CU), prediction unit (PU), and transform unit (TU). A CU is the basic unit of region partition for inter/intra coding and it is always square. The CU allows recursively splitting a block into four equally sized blocks as a quadtree. In order to improve the flexibility of predictive coding for both small and high resolution videos, CU can take the size from 8x8 up to 64x64. Each CU contains one to four PUs and the size of PU is limited to that of CU. PU is the basic unit for prediction, and prediction information such as motion vector is signaled on a PU basis. For inter coded CUs, four symmetric PU partitions (2Nx2N, Nx2N, 2NxN, and NxN) and four asymmetric partitions (2NxnD, 2NxnU, nLx2N, and nRx2N) are available. The quadtree structure is also adopted for the TU from 4x4 to 32x32.

While encoding a CU, the best PU mode is determined by successively evaluating the rate-distortion (RD) cost of each inter mode and intra mode. HEVC encoder performs a recursive traversal on the CU quadtree. When the encoder examines all candidate modes on current depth, the current depth will be split to sub CU until minimum CU size. The exhaustive search process of CU/PU achieves the best coding efficiency but costs lots of computational complexity. We reduce the PU candidate modes and stop the CU splitting by analyzing the information extracted from H.264 decoder.

HEVC test model (HM 6.1) adopts two ME algorithms, i.e. full search and EPZS (Enhanced Predictive Zonal Search). In HM 6.1, both full search and EPZS apply a fixed search range while encoding. Although the early termination condition in EPZS considerably reduces the computational complexity of ME, we can further reduce the encoding time by reducing the search range properly. In this work, MV information from H.264 decoder is refined and utilized to reduce the search range during ME process in HEVC encoder.

3 Proposed Fast Motion Estimation Algorithm

With the same ME accuracy (1/4 pixel) between H.264 and HEVC, the H.264 encoded MVs are reused. First we have to map the MVs from H.264 to HEVC. In this work, we let the centroid of block partition as the origin of each H.264 MV. This implies that different positions of centroid represent different modes. The MV with nearest distance from H.264 MV to the centroid of candidate PU is taken as the reference MV (*refMV*) in this paper. The nearest MV is determined by two-dimensional Euclidean distance. If multiple identical distances are found, the top-left one is selected. However, the candidate PU size in HEVC can be larger than an MB (16x16). For PU modes in CU depth 0 and 1, i.e. CU with sizes 64x64 and 32x32, more than one MB is mapped to a PU mode. To reduce the inappropriateness of MV mapping, the H.264 MVs are merged in this case. The median of the covered H.264 MVs is adopted as the reference MV when a PU is larger than 16x16. The derivation of *refMV* is summarized in (1).

$$refMV = \begin{cases} \text{Median of covered H.264 MVs, PU>16}\times\text{16} \\ \text{Nearest H.264 MV, otherwise} \end{cases} \tag{1}$$

To observe the accuracy of *refMV*, the distribution of distance between *refMV* and the HEVC encoded MV is shown in Table 1. Table 1 shows that *refMV* is close to HEVC MV in most sequences. But large difference still exists in some cases. Since the *refMV* can not be directly reused, we prefer to use it to decide the proper search range (SR).

Table 1 Distribution of distance between *refMV* and HEVC encoded MV

Sequence Class B	P(*D*=0)	P(*D*≤4)	P(*D*≤8)	P(*D*≤16)	P(*D*≤32)	Sequence Class C	P(*D*=0)	P(*D*≤4)	P(*D*≤8)	P(*D*≤16)	P(*D*≤32)
Kimono	0.04	0.27	0.42	0.63	0.84	Basketball	0.58	0.84	0.87	0.91	0.95
ParkScene	0.12	0.65	0.77	0.88	0.96	BQMall	0.72	0.86	0.92	0.96	0.99
Cactus	0.54	0.78	0.86	0.94	0.98	PartyScene	0.47	0.90	0.94	0.97	0.98
BQTerrace	0.18	0.87	0.91	0.94	0.98	RaceHorse	0.03	0.36	0.47	0.60	0.81
Avg.	0.22	0.64	0.74	0.85	0.94	Avg.	0.45	0.74	0.80	0.86	0.93

The predictor set indicates the start point of ME search, which is called AMVP (Advanced Motion Vector Prediction) in HEVC. The motion vector difference (MVD) is derived from the distance between the best MV and AMVP. If we have any prior knowledge of the MVD value, the best search range can be regarded as MVD as (2). However, we can obtain MVD only after the ME process. But *refMV* is available before HEVC encoding. We define a search range predictor (SRP) as (3) which plays a similar role as MVD to investigate the relation between the best SR and SRP. Where the best SR and SRP in (2) and (3) are calculated by two-dimensional Euclidean distance. The illustration of AMVP, HEVC MV, and *refMV* is shown in Fig. 2.

$$\text{best SR} = \sqrt{(\text{HEVC MV} - \text{AMVP})^2} \tag{2}$$

$$\text{SRP} = \sqrt{(refMV - \text{AMVP})^2} \tag{3}$$

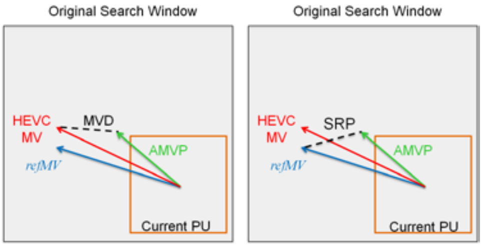


Fig. 2 Relation between AMVP, HEVC MV, and *refMV*

From Fig. 3 we can observe that the absolute difference between the best SR and SRP is small, and it prompts us to take SRP as the search range. However, prediction error occurs when SRP is smaller than the best SR. A constant α is employed to reduce the prediction error of search range. The proposed search range is finally determined by (4). Original SR is the fixed search range in HEVC encoder. Based on the experiments, we set $\alpha = 2$. Figure 4 shows the average prediction error rate has significant reduction when the proposed search range limitation is adopted.

$$\text{Search Range} = \min(\max(\text{SRP}, \alpha), \text{Original SR}) \tag{4}$$

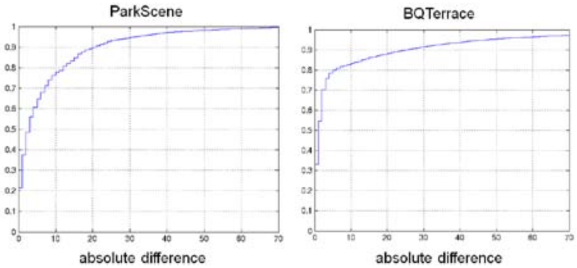


Fig. 3 CDF of difference between the best SR and SRP. The horizontal-axis is the absolute difference between the best SR and SRP, and the vertical-axis is the cumulative distribution

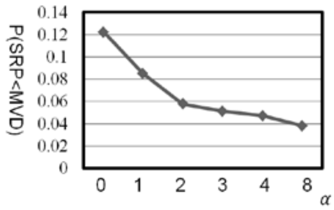


Fig. 4 Average prediction error rate with different α at QP = 24

4 Proposed Fast Mode Decision Algorithm

After carrying out a series of experiments, however, it is not easy to find an obvious mode mapping tendency due to the block partition flexibility provided by HEVC. Thus to develop the mode decision algorithm, the mode information should be combined with other decoded information such as amount of partition, residual coefficient. Because the CU size can be larger than an MB, we discuss the mapping between MB and CU separately. When the CU depth is 0 or 1, the block size is larger than an MB, thus multiple MBs are mapped to a CU. On the contrary, only one MB is mapped when CU depth is 2 or 3. Here we group the modes in H.264 into three types for simplicity, i.e. P16MB (inter 16x16, 16x8, 8x16), SubMB (inter 8x8, 8x4, 4x8, 4x4), and IntraMB (intra 4x4, 16x16).

4.1 Only one MB is mapped

In CU depth 3, the block size can be ranged from 4x4 to 8x8, this is the same as the SubMB in H.264. Thus we can intuitively split CU to depth 3 only when the mapped MB is SubMB. But this kind of simple mapping will result in poor RD performance. The residual coefficient of H.264 decoded MB can be used to jointly determine the current CU partition. Large residual coefficient implies that the MB includes details, and the partition of CU should be smaller to guarantee the prediction accuracy. We define a variable *r*, which is the ratio of power of an MB over the average MB power in a frame, to decide the homogeneity of an MB. The value of *r* is set as 1.5 in experience to achieve a good trade-off between RD and time saving. The proposed *criterion 1 for CU depth 3 skipping* is as follows.

- *Criterion 1*: In CU depth 2 and 3, we split CU to depth 3 only when the mapped MB is SubMB or the residual power ratio $r > 1.5$.

The intra mode is selected if inter prediction performs poorly. When the mapped MB is skip mode, this means that inter prediction is good enough. The proposed *criterion 2 for intra PU skipping* is as follows.

- *Criterion 2*: In CU depth 2 and 3, we omit the intra PU modes when the mapped MB is skip mode.

4.2 Multiple MBs are mapped

We investigate the splitting tendency of CU when 16 MBs are mapped. We observe that CU tends to be further split to smaller partition when there are few skip modes or there are many SubMBs in the mapped 16 MBs. The two observations imply that the encoder will rarely select depth 0 and 1 as the best depth. Therefore, we do not have to search all PU mode candidates in these two depths. The proposed *criterion 3 for PU candidates reduction* is as follows.

- *Criterion 3*: In CU depth 0 and 1, only skip mode and inter 2Nx2N mode are examined in RDO when the amount of skip mode is less than 2 or the amount of SubMB is greater than 8 in the mapped 16 MBs.

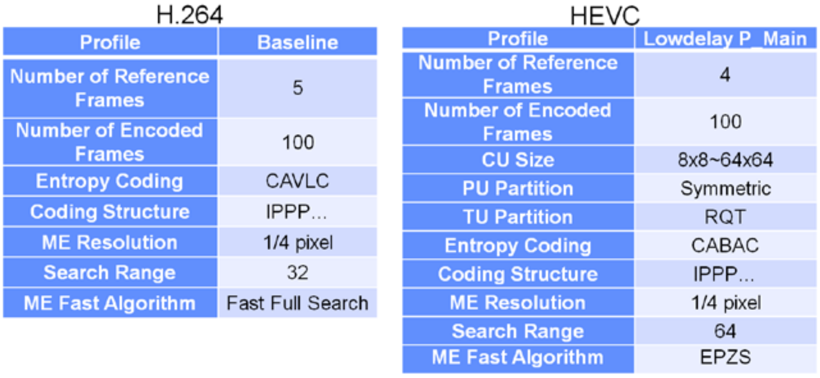
In order to further reduce the complexity of depth 2 and 3, we jointly utilize the partition amount in an MB and the residual coefficient. When the mapped 16 MBs are all zero-blocks, this means that the candidate CU is in a homogeneous region. We can restrict the CU depth to 0 and 1 without sacrificing too much RD performance. Besides, when the partition amount of the 16 mapped MBs is small, the CU tends to be encoded as large partition. The partition amount is 1 in mode 16x16, and it is 2 in mode 16x8, 4 in mode 8x8, and so on. The proposed *criterion 4 for CU candidates reduction* is as follows.

- *Criterion 4*: Only CU depth 0 and 1 is searched when all the 16 mapped MBs are zero-blocks and the partition amount is smaller than or equal to 20.

5 Experimental Results

In the simulations, the input bitstream is generated from JM 17.2 with QP = 24. The H.264 bitstream is transcoded to HEVC using HM 6.1 with QP = 24, 27, 31, and 35. The simulation environments are listed in Table 2. The proposed algorithms are compared with the original HM6.1. The experimental results of the proposed fast ME and MD algorithms are shown in Table 3, respectively. Both the proposed algorithms achieve slight RD losses with about 15% and 30% time saving. Similar results can be found at other QPs. Table 4 shows the performance of overall scheme. The performance in terms of BDrate and BDPSNR [8] are also shown in Table 4. On average, 40% encoding time can be saved with 1.2% BDrate increase. The good performance demonstrates the effectiveness of the proposed algorithms.

Table 2 Simulation environments



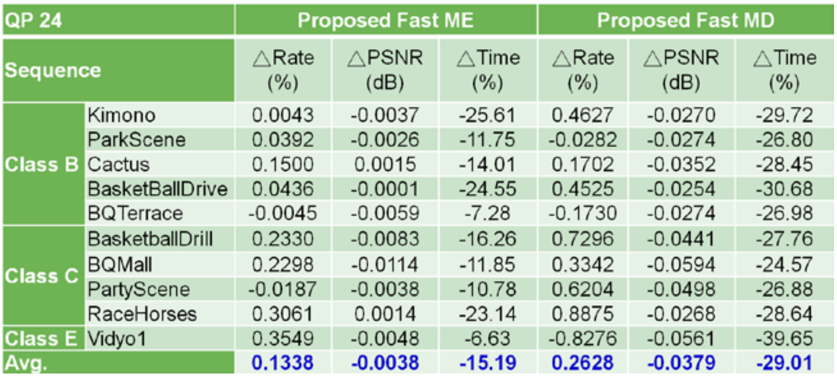
H.264

Profile	Baseline
Number of Reference Frames	5
Number of Encoded Frames	100
Entropy Coding	CAVLC
Coding Structure	IPPP…
ME Resolution	1/4 pixel
Search Range	32
ME Fast Algorithm	Fast Full Search

HEVC

Profile	Lowdelay P_Main
Number of Reference Frames	4
Number of Encoded Frames	100
CU Size	8x8~64x64
PU Partition	Symmetric
TU Partition	RQT
Entropy Coding	CABAC
Coding Structure	IPPP…
ME Resolution	1/4 pixel
Search Range	64
ME Fast Algorithm	EPZS

Table 3 Performance of the proposed fast ME and MD algorithms at QP = 24

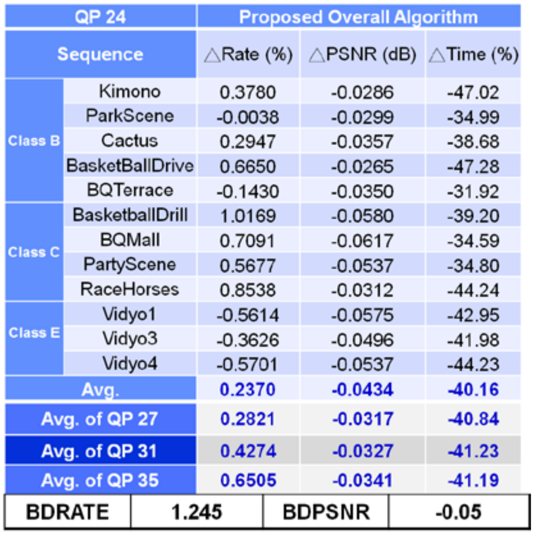


QP 24		Proposed Fast ME			Proposed Fast MD		
Sequence		△Rate (%)	△PSNR (dB)	△Time (%)	△Rate (%)	△PSNR (dB)	△Time (%)
Class B	Kimono	0.0043	-0.0037	-25.61	0.4627	-0.0270	-29.72
	ParkScene	0.0392	-0.0026	-11.75	-0.0282	-0.0274	-26.80
	Cactus	0.1500	0.0015	-14.01	0.1702	-0.0352	-28.45
	BasketBallDrive	0.0436	-0.0001	-24.55	0.4525	-0.0254	-30.68
	BQTerrace	-0.0045	-0.0059	-7.28	-0.1730	-0.0274	-26.98
Class C	BasketballDrill	0.2330	-0.0083	-16.26	0.7296	-0.0441	-27.76
	BQMall	0.2298	-0.0114	-11.85	0.3342	-0.0594	-24.57
	PartyScene	-0.0187	-0.0038	-10.78	0.6204	-0.0498	-26.88
	RaceHorses	0.3061	0.0014	-23.14	0.8875	-0.0268	-28.64
Class E	Vidyo1	0.3549	-0.0048	-6.63	-0.8276	-0.0561	-39.65
Avg.		**0.1338**	**-0.0038**	**-15.19**	**0.2628**	**-0.0379**	**-29.01**

6 Conclusions

This paper proposes a fast inter prediction algorithm including fast motion estimation and mode decision for H.264 to HEVC transcoding. In ME, the search range predictor (SRP) which is close to MVD in HEVC is proposed to determine the search range adaptively. In MD, coding information including skip mode, SubMB,

Table 4 Performance of the proposed overall algorithm



QP 24		Proposed Overall Algorithm		
Sequence		△Rate (%)	△PSNR (dB)	△Time (%)
Class B	Kimono	0.3780	-0.0286	-47.02
	ParkScene	-0.0038	-0.0299	-34.99
	Cactus	0.2947	-0.0357	-38.68
	BasketBallDrive	0.6650	-0.0265	-47.28
	BQTerrace	-0.1430	-0.0350	-31.92
Class C	BasketballDrill	1.0169	-0.0580	-39.20
	BQMall	0.7091	-0.0617	-34.59
	PartyScene	0.5677	-0.0537	-34.80
	RaceHorses	0.8538	-0.0312	-44.24
Class E	Vidyo1	-0.5614	-0.0575	-42.95
	Vidyo3	-0.3626	-0.0496	-41.98
	Vidyo4	-0.5701	-0.0537	-44.23
Avg.		0.2370	-0.0434	-40.16
Avg. of QP 27		0.2821	-0.0317	-40.84
Avg. of QP 31		0.4274	-0.0327	-41.23
Avg. of QP 35		0.6505	-0.0341	-41.19
BDRATE	1.245	BDPSNR	-0.05	

partition amount, and residual coefficient are employed. Several criteria are proposed to stop the CU splitting or reduce the PU candidates. These strategies can achieve about 15% ME and 30% MD time saving, respectively. Simulation results show that the proposed overall scheme can save up to 47% encoding time with negligible bitrate increase.

7 References

1. JCTVC (2012). HM6: High Efficiency Video Coding (HEVC) Test Model 6 Encoder Description. JCTVC-H1002.
2. Vetro, A., Christopoulos, C., & H. Sun (2003). Video Transcoding Architectures and Techniques: An Overview. *IEEE Signal Processing Magazine*, 18-29.
3. Ahmad, I., Wei, X., Sun, Y., & Zhang, Y.-Q. (2005). Video Transcoding: An Overview of Various Techniques and Research Issues. *IEEE Trans. Circuits Syst. Video Technol., 7 (5)*, 793-804.
4. Garrido-Cantos, R., Cock, J. D., Martínez, J. L., Leuven, S. V., & Cuenca, P. (2011). Motion-based Temporal Transcoding from H.264 AVC-to-SVC in Baseline Profile. *IEEE Trans. Consum. Electron., 57 (1)*, 239-246.
5. Lu, X., Tourapis, A. M., Yin, P., & Boyce, J. (2005). Fast Mode Decision and Motion Estimation for H.264 with a Focus on MPEG-2/H.264 Transcoding. *IEEE Int. Symp. Circuits Syst.*, 1246-1249.
6. Peixoto, E., & Izquierdo, E. (2012). A Complexity-Scalable Transcoder From H.264/AVC to the New HEVC Codec. *IEEE ICIP*, 737-740.
7. Zhang, D., Li, B., Xu, J., & Li, H. (2012). Fast Transcoding From H.264/AVC To High Efficiency Video Coding. *IEEE ICME*, 651-656.
8. Bjontegaard, G. (2008). Improvements of the BD-PSNR Model. ITU-T SG16/Q6, 35th VCEG Meeting, Berlin, Germany, VCEG-AI11.