

Article

Efficient Software HEVC to AVS2 Transcoding

Yucong Chen ¹, Yun Zhou ² and Jiangtao Wen ^{1,*}

¹ Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China; chenyucong702@gmail.com

² Academy of Broadcasting Science, State Administration of Press, Publication, Radio, Film and Television of The People's Republic of China (SAPPRFT), Beijing 100045, China; zhouyun@abs.ac.cn

* Correspondence: jtwen@tsinghua.edu.cn; Tel.: +86-10-62784141

Academic Editor: Khalid Sayood

Received: 29 July 2016; Accepted: 14 September 2016; Published: 19 September 2016

Abstract: The second generation of Audio and Video coding Standard (AVS) is developed by the IEEE 1857 Working Group under project 1857.4 and was standardized in 2016 by the AVS Working Group of China as the new broadcasting standard AVS2. High Efficient Video Coding (HEVC) is the newest global video coding standard announced in 2013. More and more codings are migrating from H.264/AVC to HEVC because of its higher compression performance. In this paper, we propose an efficient HEVC to AVS2 transcoding algorithm, which applies a multi-stage decoding information utilization framework to maximize the usage of the decoding information in the transcoding process. The proposed algorithm achieves $11\times$ – $17\times$ speed gains over the AVS2 reference software RD 14.0 with a modest BD-rate loss of 9.6%–16.6%.

Keywords: video transcoding; high efficient video coding (HEVC); AVS2

1. Introduction

Aiming to establish generic technical standards for the compression, decoding, processing, and representation of digital audio-video content, the Audio and Video coding Standard (AVS) Working Group was formed in China in March 2002 [1]. By the year 2012, the AVS Working Group had published a series of video coding standards, which formed the first version of AVS standards. In AVS1, there are six profiles defined to satisfy different requirements of various applications including Main, Surveillance Baseline, Enhanced, Portable, Surveillance and Broadcasting. Also, the AVS standards are being recognized by the global society. Based on the members of the IEEE standards Association from the AVS Work Group, the IEEE 1857 Working Group was founded in 2012 to globalize the AVS standards as the IEEE 1857 standards [2]. So far, the IEEE 1857 Working Group has finished three parts of the IEEE 1857 standards.

The second generation of AVS, AVS2, has been under development since 2012 [3–5]. At the end of 2015, the final draft of AVS2 was released and the standardization process is running now. Targeting next generation Ultra High Definition TV (UHDTV), AVS2 adopts larger partition sizes and other coding tools to improve coding performance, especially on high resolution videos. Compared to its predecessor AVS1, AVS2 achieves up to 50% bitrate savings and provides more features on surveillance video coding [6].

High Efficiency Video Coding (HEVC) is the latest global standard on video coding. It was developed by the Joint Collaborative Team on Video Coding (JCT-VT) and was standardized in 2013 [7]. HEVC was designed to double the compression ratios of its predecessor H.264/AVC with a higher computational complexity. After several years of developments, mature encoding and decoding solutions are emerging, accelerating the upgrade of the video coding standards of video contents from the legacy standards such as H.264/AVC. With the increasing needs of ultra-high resolution videos, it can be foreseen that HEVC will become the most important video coding standard in the near future.

Compared with HEVC, AVS2 shows comparable coding performance with HEVC on normal videos, while for surveillance videos AVS2 outperforms HEVC by 32.1% due to its additional background coding for surveillance videos [6]. Until now, there have been few well-optimized AVS2 software encoders, so transcoding from HEVC to AVS2 is a fast way to generate AVS2 video contents. Also, HEVC and AVS2 share the same quad-tree partitioning structure and the block processing flows of both are similar, making it easier and more efficient to utilize the HEVC decoding information in the AVS2 encoding process.

In this paper, we propose a fast HEVC to AVS2 transcoding algorithm that utilizes the partition sizes, prediction modes, motion vectors (MVs), reference pictures, intra-prediction modes and other information extracted from the input HEVC bitstream to expedite the AVS2 encoding process and achieves efficient HEVC to AVS2 transcoding. Inspired by the fast H.264/AVC to HEVC transcoding in [8,9], we adopt the multi-stage decoding information utilization framework in [9] and apply in the HEVC to AVS2 transcoding scenario to maximize the usage of the decoding information. Experiments show that our proposed algorithm can achieve up to $17\times$ speed gains over the RD 14.0 AVS2 reference software with an acceptable coding performance loss.

The remainder of this paper is organized as follows. Section 2 reviews some related work. Section 3 compares the similarities and differences between HEVC and AVS2 standards. Section 4 introduces the fast HEVC to AVS2 transcoding algorithm based on the multi-stage decoding information utilization framework. Section 5 shows the experimental results on HEVC test sequences and we discuss them in Section 6. Section 7 presents the materials used and Section 8 concludes this paper.

2. Related Work

There are quite a few research works on video transcoding between different coding standards, which is defined as heterogeneous transcoding in [10]. In [11], the authors discuss some key issues in generic video transcoding process. The authors of [12–14] provide some thoughts on transcoding among MPEG-2, MPEG-4 and H.264.

After the HEVC standard was finalized, there were more explorations on transcoding from H.264/AVC to HEVC. Zhang et al. [15] proposed a solution where the number of candidates for the coding unit (CU) and prediction unit (PU) partition sizes was reduced for the intra-pictures, while for the inter-pictures, a Power-spectrum based Rate-distortion Optimization model-based power spectrum was used to estimate the best CU split tree from a reduced set of PU partition candidates according to the MV information in the input H.264/AVC bitstream. Peixoto et al. [16] proposed a transcoding architecture based on their previous work in [17]. They proposed two algorithms for mapping modes from H.264/AVC to HEVC, namely, dynamic thresholding of a single H.264/AVC coding parameter and context modeling using linear discriminant functions to determine the outgoing HEVC partitions. The model parameters for the two algorithms were computed using the beginning frames of a sequence. They achieved a $2.5\times$ – $3.0\times$ speed gain with a BD-rate [18] loss between 2.95% and 4.42% by the proposed PT-LDF method. Diaz-Honrubia et al. [19] proposed an H.264/AVC to HEVC transcoder based on a statistical NB classifier, which decides on the most appropriate quadtree level in the HEVC encoding process. Their algorithms achieved a speedup of $2.31\times$ on average with a BD-rate penalty of 3.4%. Mora et al. [20] also proposed an H.264/AVC to HEVC transcoder based on quadtree limitation, where the fusion map generated by the motion similarity of decoded H.264/AVC blocks is used to limit the quadtree of HEVC coded frames. They achieved 63% time saving with only a 1.4% bitrate increase.

Because the standardization process of AVS2 is not finished yet, there is little research work on this fairly new standard. According to the inheritances between H.264/AVC and HEVC and between AVS1 and AVS2, works on H.264/AVC to AVS transcoding can be a reference. Wang et al. [21] proposed a fast transcoding algorithm from H.264/AVC bitstream to AVS bitstream. They used a QP mapping method and a reciprocal SAD weighted method on intra mode selection and inter MV estimation.

Their algorithms achieved 50% time saving in intra prediction with ignorable coding performance loss and 40% time saving in inter prediction with minor coding performance loss.

The algorithm we proposed in the paper is inspired by the multi-stage decoding information utilization in our previous work [9]. In [9], we presented a four-level transcoder framework, namely GOP-level Task Distribution, High-level Parallel Processing, Mid-level H.264/AVC Information Utilization, and Low-level SIMD and Assembly Acceleration, and implemented on a distributed multi-core processors system, achieving 720 p at 30 Hz H.264/AVC to HEVC transcoding in real time. The Mid-level H.264/AVC Information Utilization is based on the multi-stage decoding information utilization framework. In this framework, the decoding information is grouped into different layers and the utilization process is divided into different stages, in which decoding information in particular layer will be utilized only when the corresponding information in the upper layer has been used in the previous stage. Thus, the decoding information can be used as much as possible to maximize the efficiency of the transcoding. In this paper, we apply this framework on HEVC to AVS2 transcoding, and, as the experiments show, the framework helps the transcoder achieve a high speed gain efficiency.

3. Comparison between HEVC and AVS2 Standards

Though coming from different series of coding standards, HEVC and AVS2 are quite similar in many aspects in terms of their coding tools, which provides some convenient ways of transcoding HEVC bitstream to AVS2 bitstream. One of the important differences, which needs to be pointed out, is that in AVS2 a new frame type called F frame (Forward-predicted frame) is proposed and several new prediction techniques are added for these new types of frames.

3.1. Partition Sizes

To improve compression performances on high resolution videos, both of HEVC and AVS2 replace the fixed size macroblock partitioning methods with the new quad-tree partitioning structure at a range from 4×4 to 64×64 .

For intra predicted blocks, both HEVC and AVS2 support square $2N \times 2N$ and $N \times N$ partitioning from 4×4 to 64×64 , while in AVS2 a new Short Distance Intra Prediction (SDIP) technique is added to support Coding Units (CUs) parted into four horizontal or vertical strips, that is, $2N \times 0.5N$ and $0.5N \times 2N$. SDIP is more adaptive to the image content especially in an edge area, but calculating the SDIP modes increases the computational complexity, so it is not applied to 64×64 CU.

For inter predicted blocks, both HEVC and AVS2 share exactly the same set of partition sizes, including square $2N \times 2N$, rectangular $2N \times N$ and $N \times 2N$, and Asymmetric Motion Prediction (AMP) modes $2N \times nU$, $2N \times nD$, $nL \times 2N$, $nR \times 2N$, where $N = 4, 8, 16, 32$ and AMP modes only exist in CUs larger than 8×8 .

3.2. Intra Prediction

Besides the introduction of SDIP modes, the intra prediction modes in AVS2 are also different from those in HEVC. There are 35 intra modes in HEVC including DC mode, Planar mode, and 33 angular modes, while there are 33 modes in AVS2 including DC mode, Plane mode, Bilinear mode and 30 angular modes. In terms of the angular modes, the degree difference between adjacent angular modes is 5.625 degree in HEVC and 7.5 degree in AVS2, which means HEVC has a smaller partitioning on angular intra modes.

3.3. Inter Prediction

Both HEVC and AVS2 have made some improvements to extend P frames in previous standards because P frames are usually refer to and are closely relevant to coding performance. In HEVC, reference lists are extended to include both forward and backward reference pictures, making it possible for B frames to use two of forward reference pictures or backward reference pictures, or one of each list. Thus, P frames can be replaced by B frames using two forward reference pictures to get

compression gains. In AVS2, F frames are introduced and a temporal Dual Hypothesis Prediction (DHP) technique is designed for these frames with a signaled MV for a main reference picture and an implicit MV for another reference picture calculated according the time distance between two reference pictures. Besides, a spatial Directional Multi-Hypothesis (DMH) technique is introduced to average two near blocks oppositely located around the best MV point in Motion Compensation (MC) process.

For conventional B frames, AVS2 adds a new Symmetric prediction mode along with Forward, Backward, and Bidirectional modes. The Symmetric mode is similar to the Dual mode of DHP, but the MV of the forward reference picture is signaled with the MV of the backward reference picture calculated from the time distance between two reference pictures.

For Direct/Skip modes, HEVC uses the Merge mechanism which chooses a candidate from those derived from the neighbor blocks and uses a flag to indicate whether it is Skip or not. In AVS2, corresponding to the new prediction modes, Weighted, Complementary and Multi-Hypothesis Direct/Skip modes are added besides the conventional ones. Since the MV derivation mechanisms are not so helpful during transcoding, the details of both Direct/Skip modes will not be discussed here, and we only take their MVs and reference pictures into consideration during the transcoding process.

3.4. Other Coding Tools

Apart from the code tools mentioned above, there are some other differences in coding tools between HEVC and AVS2. In an in-loop filter, AVS2 adds an Adaptive Loop Filter (ALF) after the Deblock Filter (DBF) and Sample Adaptive Offset (SAO). HEVC uses Discrete Cosine Transform (DCT) along with 4×4 Discrete Sine Transform (DST), while AVS2 uses Integer Transform (IT) along with 64×64 Logical Transform (LOT). For entropy coding methods, HEVC uses Content Adaptive Binary Arithmetic Coding (CABAC) while AVS2 uses Adaptive Entropy Coding (AEC). In addition, AVS2 adds special coding tools for background frame coding. During the transcoding process we will keep these coding tools work as though they are in the encoding process.

Table 1 summarizes the differences in coding tools between HEVC and AVS2.

Table 1. Comparisons on Coding Tools between High Efficiency Video Coding (HEVC) and Audio and Video coding Standard (AVS2).

| | HEVC | AVS2 |
|------------------------|--|--|
| Frame Types | I, P, B | I, P, B, F |
| Intra Partition Sizes | Square 4×4 – 64×64 | Square 4×4 – 64×64 $2N \times 0.5N$, $0.5N \times 2N$ ($N = 8, 16$) |
| Inter Partition Sizes | $2N \times 2N$, $2N \times N$, $N \times 2N$ ($N = 4, 8, 16, 32$) $2N \times nU$, $2N \times nD$, $nL \times 2N$, $nR \times 2N$ ($N = 8, 16, 32$) | $2N \times 2N$, $2N \times N$, $N \times 2N$ ($N = 4, 8, 16, 32$) $2N \times nU$, $2N \times nD$, $nL \times 2N$, $nR \times 2N$ ($N = 8, 16, 32$) |
| Intra Prediction Modes | Total 35 modes: DC, Planar 33 angular modes | Total 33 modes: DC, Plane, Bilinear 30 angular modes |
| Inter Prediction Modes | L0, L1, Bi Merge/Skip | Fw, Bw, Bi, Dual, Sym Direct/Skip |
| Other Coding Tools | In-loop Filter: DBF + SAO Transform: DCT + 4×4 DST Entropy Coding: CABAC | In-loop Filter: DBF + SAO + ALF Transform: IT + 64×64 LOT Entropy Coding: AES Background frame coding |

4. The Proposed Algorithms

We proposed a fast H.264/AVC to HEVC transcoding algorithm based on multi-stage decoding information utilization framework in [9]. In this framework, H.264/AVC decoding information, including partition size, prediction mode, reference pictures and MVs, is grouped into different layers and the utilization process is divided into several stages. Only after the processing of the information in the upper layer in the previous stage has been completed, can the corresponding information in the lower layer be processed in the next stage.

Instead of three stages as we used in H.264/AVC to HEVC transcoding, two stages are used in HEVC to AVS2 transcoding as is shown in Figure 1, including (1) partition size and mode decisions; (2a) reference picture decisions and MV estimation; (2b) intra mode decisions.

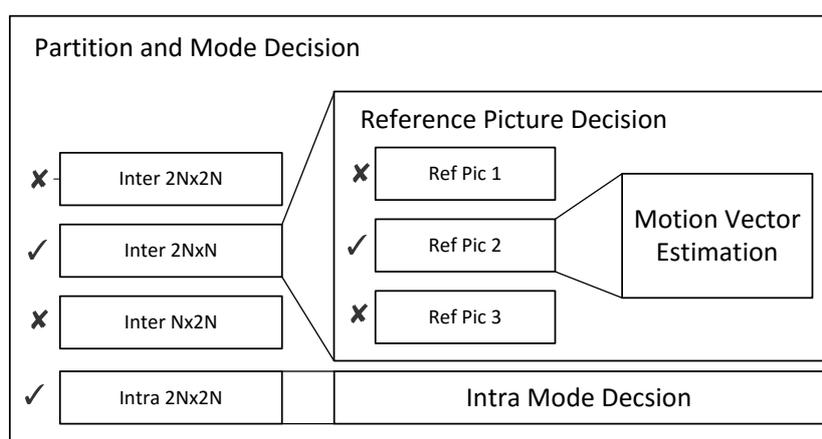


Figure 1. Fast HEVC to AVS2 Transcoding Algorithms Based on Multi-stage Decoding Information Utilization Framework.

4.1. Fast Partition Mode Decision

Although all partition modes from HEVC can be found in AVS2, we cannot just simply map the same partition modes from HEVC to AVS2. For example, if the target bitrates of transcoding are much less than the original bitrate of the input HEVC bitstream, the partition sizes should be larger than those in the HEVC bitstream. To simplify the discussion, we make an assumption that the bitrates of transcoded bitstreams are smaller than the original bitrate of the input HEVC bitstream, based on the fact that larger bitrates of transcoded videos than that of the original input video is less reasonable since there will always be quality loss after transcoding. We divide the Fast Partition Mode Decision stage into three stages according the comparison between the partition depth of the CU in AVS2 and in HEVC.

We first consider the case where the partition depth of CU in AVS2 is smaller than that in HEVC. With the assumption that the bitrates of transcoded bitstreams are smaller, it is unlikely that smaller partitions are used in AVS2 than in HEVC. Thus, we limit the maximum partition depths of CUs in AVS2 to the partition depths of the corresponding CUs in HEVC.

Then we consider the case where the partition depths of CU in AVS2 are larger than that in HEVC. With the same assumption, it is likely that larger partitions are used in AVS2 than in HEVC. In the AVS2 encoding process, the recursive mode examinations of CU are conducted bottom-up, that is, the smallest CUs are checked first to get the best modes and then we check the larger ones. So we always can get the results of mode decisions of the next depth when checking the modes in the current depth, except if we are already in the maximum depth. Thus, we can filter some more possible modes based on the modes used in the next depth. We conduct similar algorithms with the Fast Partition Mode Decision in H.264/AVC to HEVC transcoding described in [9] as follows. Examples for each condition are given in Figure 2.

1. Direct/Skip modes are always checked.
2. Inter $2N \times 2N$ is checked when there exist more than two sub-CUs using Inter $2N \times 2N$ (To simplify the discussions, the Merge $2N \times 2N$ and SKIP modes in HEVC are treated as special cases of the Inter $2N \times 2N$ mode).
3. Inter $2N \times N$ is checked when both the two upper or both the two lower sub-CUs use Inter $2N \times 2N$.
4. Inter $N \times 2N$ is checked when both the two left or both the two right sub-CUs use Inter $2N \times 2N$.
5. Intra $2N \times 2N$ is checked when there exist more than two sub-CUs using Intra $2N \times 2N$.

For AMP modes:

6. Inter $2N \times nU$ is checked when the two lower sub-CUs use Inter $2N \times 2N$ and more than one of the two upper sub-CUs use Inter $2N \times N$.
7. Inter $2N \times nD$ is checked when the two upper sub-CUs use Inter $2N \times 2N$ and more than one of the two lower sub-CUs use Inter $2N \times N$.
8. Inter $nL \times 2N$ is checked when the two right sub-CUs use Inter $2N \times 2N$ and more than one of the two left sub-CUs use Inter $N \times 2N$.
9. Inter $nR \times 2N$ is checked when the two left sub-CUs use Inter $2N \times 2N$ and more than one of the two right sub-CUs use Inter $N \times 2N$.

For SDIP modes:

10. Intra $2N \times 0.5N$ is checked when both the two upper or both the two lower sub-CUs use Intra $2N \times 2N$.
11. Intra $0.5N \times 2N$ is checked when both the two left or both the two right sub-CUs use Intra $2N \times 2N$.

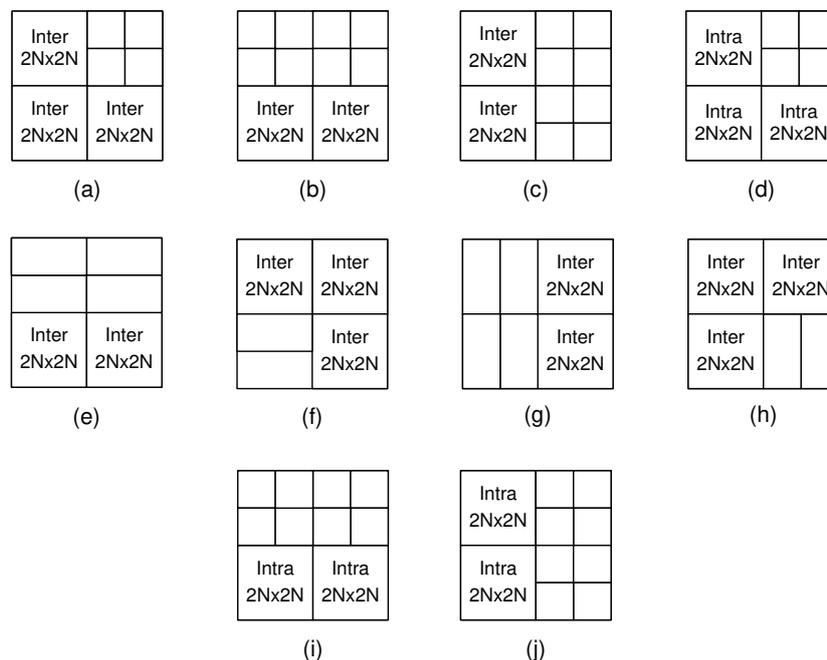


Figure 2. (a–d) Examples for when conditions 2–5 are applied; (e–h) Examples for when conditions of Asymmetric Motion Prediction (AMP) modes are applied; (i,j) Examples for when conditions of Short Distance Intra Prediction (SDIP) modes are applied.

Finally we consider the case where the partition depths of CU in AVS2 and in HEVC are equal. Since the CUs are of the same size, we check the same mode in AVS2 with that in HEVC. If an intra mode is used in the CU in HEVC, we check two more SPID modes in AVS2. Because intra blocks are less possible to be used in B frames, we do not check SPID modes in B frames.

4.2. Fast Reference Picture Decision and Motion Vector Estimation

For B frames, there is only one forward and one backward reference picture in the default settings, so there is no need to make the reference picture decision. For F frames, corresponding to Generalized P and B (GPB) frames in HEVC, there are at most two reference pictures from HEVC. We check all the reference pictures of the HEVC PUs covered by the current block.

In the Motion Estimation (ME) process of AVS2 encoding, the uni-referred forward and backward prediction will be first processed to get the best integer MVs of each reference picture from both lists and then refine them to sub-pixel level. For the multi-referred dual, symmetric and bidirectional prediction, the MV searching will start from the best integer MVs acquired in the forward or backward prediction and then search around it to do sub-pixel refinements. Therefore, we first convert the corresponding HEVC MVs into an integer level and then substitute the best integer MVs of each reference picture in the AVS2 ME process with the converted HEVC MVs to skip the original integer pixel searching. The sub-pixel refinements of multi-referred predictions are kept unchanged.

4.3. Fast Intra Mode Decision

There are total 33 intra modes in AVS2, including DC mode, Plane mode, Bilinear mode and 30 angular modes. Without interpolating the pixels, non-angular modes do not take much time to calculate, so we check all the three non-angular modes. For angular modes, we check four to five angels of AVS2 modes around the angel of HEVC modes. The mapping from HEVC angular modes to AVS2 angular modes can be expressed as follows:

$$Y = \begin{cases} \{y | 30 - \frac{x-2}{4} \times 3 - 2 \leq y \leq 30 - \frac{x-2}{4} \times 3 + 2\} & (x-2) \bmod 4 = 0, \\ \{y | 30 - \lfloor \frac{x-2}{4} \rfloor \times 3 \leq y \leq 30 - \lceil \frac{x-2}{4} \rceil \times 3\} & (x-2) \bmod 4 \neq 0. \end{cases} \quad (1)$$

where x is the index of the HEVC angular mode and Y is the index set of the AVS2 angular modes to be checked.

If the HEVC mode is non-angular, we check all the AVS2 intra modes.

5. Experimental Results

Experiments were conducted with standard HEVC test clips to evaluate the speed and Rate-Distortion (R-D) performances of our proposed transcoding algorithm. Test tests were run on a server with two Intel Xeon 2.4 GHz CPUs and 64 GB RAM running Linux operating system.

The RD-14.0 AVS2 reference software is used in the full decoding-and-encoding transcoding method, which is the baseline compared to our proposed fast transcoding algorithms in the experiments. We use the encoder_ra.cfg default configuration file to configure RD-14.0 with a Random Access frame structure of 7 B frames and 1 F frame and an intra period of 48 frames. Our proposed algorithms are implemented based on RD-14.0 and the same configuration file is used. QPs {35, 40, 45, 50} are used for each test clip.

The input HEVC bitstreams of each test clip are generated using HM-16.0 HEVC reference software with the default Random Access configuration and a QP of 27. Sharing the same HEVC decoding procedure, we only compare the performances on the AVS2 encoding side, where the AVS2 encoder encodes the decoded frames from HEVC decoder into AVS2 bitstreams. To observe the performance changes between each stage of our proposed algorithms, we record stage changes and accumulative changes of all the stages and sub-stages in our proposed algorithms as is shown in Table 2. Figure 3 shows the Rate-Distortion (R-D) performances of the proposed transcoder against the full decoding-and-encoding AVS2 transcoder using RD-14.0.

Table 2. Performance in stages of proposed transcoding algorithms.

| Sequence | Stage | 1-Stage Change | | Accumulative Changes | |
|--|---------|----------------|----------------|----------------------|----------------|
| | | BD-Rate (%) | Speed Gain (×) | BD-Rate (%) | Speed Gain (×) |
| Traffic 2560 × 1600 150 frames | RD-14.0 | 0.00 | 1.00 | 0.00 | 1.00 |
| | I-1 | 0.89 | 2.97 | 0.89 | 2.97 |
| | I-2 | 4.18 | 1.35 | 5.11 | 4.00 |
| | I-3 | 4.35 | 1.92 | 9.73 | 7.66 |
| | II-A | −0.81 | 1.50 | 8.87 | 11.53 |
| ParkScene 1920 × 1080 240 frames | II-B | 0.93 | 1.09 | 9.88 | 12.60 |
| | RD-14.0 | 0.00 | 1.00 | 0.00 | 1.00 |
| | I-1 | 0.59 | 2.65 | 0.59 | 2.65 |
| | I-2 | 3.94 | 1.38 | 4.56 | 3.67 |
| | I-3 | 4.38 | 1.94 | 9.19 | 7.12 |
| Cactus 1920 × 1080 500 frames | II-A | −0.13 | 1.46 | 9.05 | 10.37 |
| | II-B | 0.47 | 1.11 | 9.57 | 11.48 |
| | RD-14.0 | 0.00 | 1.00 | 0.00 | 1.00 |
| | I-1 | 0.93 | 2.86 | 0.93 | 2.86 |
| | I-2 | 4.57 | 1.36 | 5.54 | 3.89 |
| FourPeople 1280 × 720 600 frames | I-3 | 6.76 | 1.94 | 12.74 | 7.54 |
| | II-A | 2.69 | 1.51 | 15.79 | 11.35 |
| | II-B | 0.70 | 1.11 | 16.60 | 12.60 |
| | RD-14.0 | 0.00 | 1.00 | 0.00 | 1.00 |
| | I-1 | 0.80 | 4.23 | 0.80 | 4.23 |
| 1280 × 720 600 frames | I-2 | 4.16 | 1.20 | 4.98 | 5.08 |
| | I-3 | 2.21 | 2.09 | 7.32 | 10.60 |
| | II-A | 2.40 | 1.40 | 9.95 | 14.82 |
| II-B | 1.15 | 1.14 | 11.20 | 16.84 | |

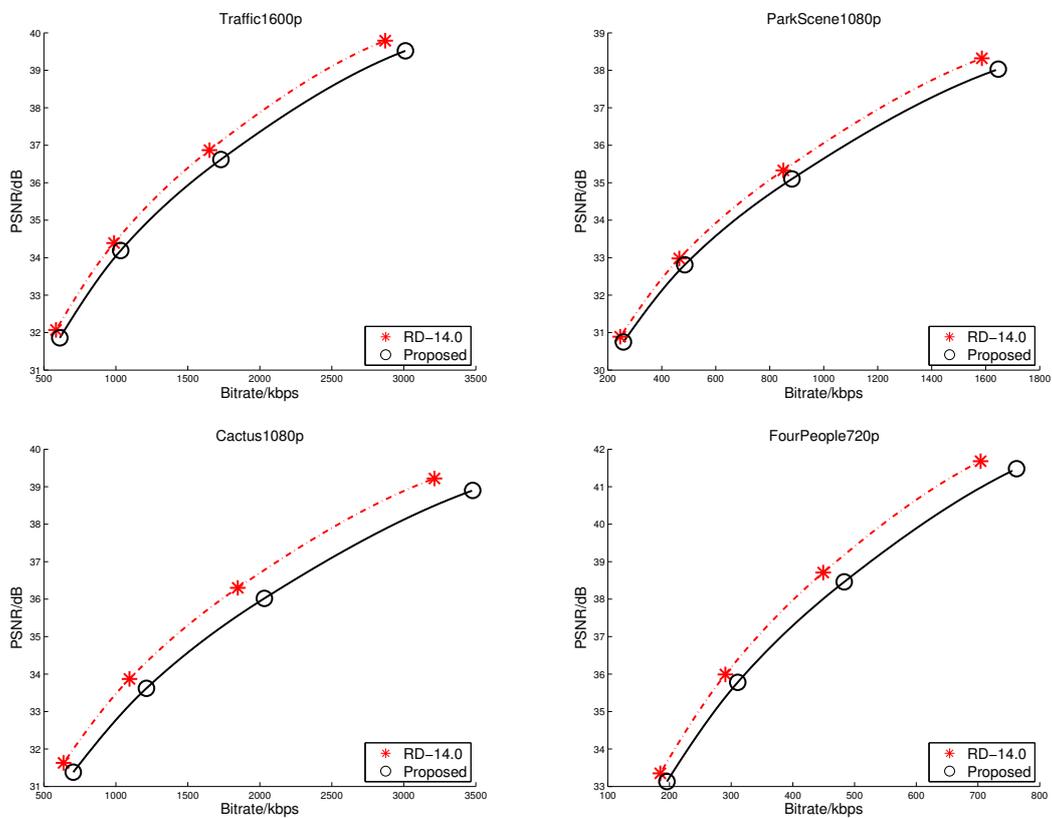


Figure 3. R-D Performance of Proposed Transcoder on HEVC Test Clips.

6. Discussion

Table 2 shows the performance changes of each stage, where I-1, I-2 and I-3 represent the three sub-stages in Fast Partition Size and Mode Decision described in Section, II-A represents the Fast Reference Picture Decision and MV Estimation, and II-B represents the Fast Intra Mode Decision.

For stage I-1, the limitation of maximum partition depth of AVS2 CUs brings $2.7\times$ – $4.2\times$ speed gains with ignorable R-D performance loss. For stage I-2, the filtering of larger partition modes achieves a speed gain around $1.3\times$ with about a 4% BD-Rate loss. For stage I-3, the mapping of partition modes between CUs of same depth brings around $2\times$ speed gains with an average 4.4% RD-Rate loss. For the whole Fast Partition and Mode Decision stage, the algorithms achieve total $7\times$ – $10\times$ speed gains with an average 9.6% R-D performance loss. Although the speed gains achieved are not so significant compared with the corresponding BD-Rate loss, this stage makes it possible for the subsequent stages of the transcoding process to use the rest decoding information more efficiently, leading to a good overall speed gain for the entire transcoding process and an acceptable R-D performance for many applications.

For stage II-A, the Fast Reference Picture Decision and MV Estimation brings a speed gain of around $1.5\times$ with an average 1% BD-Rate loss. For stage II-B, the Fast Intra Mode Decision achieves about $1.1\times$ speed gains and the average BD-Rate loss is within 1%. This stage makes good speedup efficiency since the usages of decoding information in this stage are more reliable after the last stage.

The overall transcoding algorithms achieves an $11\times$ – $17\times$ speed gain with a 9.6%–16.6% R-D performance loss. By the calculation of Speedup Efficiency presented in [9], the efficiencies of the proposed algorithms for the test clips are 117, 110, 70, 141, which are relatively higher than the H.264/AVC to HEVC transcoder in [9].

7. Materials and Methods

The RD-14.0 AVS2 reference software is retrieved from AVS Work Group [22]. The test clips used in the experiments are selected from the HEVC standard test sequences.

8. Conclusions

In this paper, we proposed a fast HEVC to AVS2 transcoding algorithm based on the multi-stage decoding information utilization framework. By utilizing HEVC decoding information in a multi-stage approach, the proposed algorithms achieve a speed gain of $11\times$ – $17\times$ more than the the RD-14.0 AVS2 reference software, with an R-D performance loss of 9.6%–16.6%. This provides an efficient way to transcode existing HEVC video contents into AVS2 video contents in a high processing speed with an acceptable compression performance.

Acknowledgments: This work is supported by the State Key Program of National Natural Science of China (Grant No. 61133008).

Author Contributions: Yucong Chen and Jiangtao Wen conceived and designed the experiments; Yun Zhou performed the experiments; Yucong Chen and Yun Zhou analyzed the data; Yun Zhou contributed materials; Yucong Chen wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fan, L.; Ma, S.; Wu, F. Overview of AVS video standard. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04), Taipei, Taiwan, 27–30 June 2004; Volume 1, pp. 423–426.
2. Ma, S.; Wang, S.; Gao, W. Overview of IEEE 1857 video coding standard. In Proceedings of the IEEE International Conference on Image Processing, Melbourne, Australia, 15–18 September 2013; pp. 1500–1504.
3. He, Z.; Yu, L.; Zheng, X.; Ma, S.; He, Y. Framework of AVS2-video coding. In Proceedings of the IEEE International Conference on Image Processing, Melbourne, Australia, 15–18 September 2013; pp. 1515–1519.

4. He, Z.; Yu, L.; Zheng, X.; Ma, S. AVS2-video coding standard—An application-oriented and high performance video coding standard. In Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, China, 14–18 July 2014; pp. 1–5.
5. Ma, S.; Huang, T.; Gao, W. The second generation IEEE 1857 video coding standard. In Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Chengdu, China, 12–15 July 2015; pp. 171–175.
6. Ma, S.; Huang, T.; Reader, C.; Gao, W. AVS2 ? Making video coding smarter [standards in a nutshell]. *IEEE Signal Process. Mag.* **2015**, *32*, 172–183.
7. Sullivan, G.; Ohm, J.; Han, W.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668.
8. Shen, T.; Lu, Y.; Wen, Z.; Zou, L.; Chen, Y.; Wen, J. Ultra Fast H.264/AVC to HEVC Transcoder. In Proceedings of the Data Compression Conference (DCC), Snowbird, UT, USA, 20–22 March 2013; pp. 241–250.
9. Chen, Y.; Wen, Z.; Wen, J.; Tang, M.; Tao, P. Efficient Software H.264/AVC to HEVC Transcoding on Distributed Multicore Processors. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1423–1434.
10. Ahmad, I.; Wei, X.; Sun, Y.; Zhang, Y.Q. Video transcoding: An overview of various techniques and research issues. *IEEE Trans. Multimed.* **2005**, *7*, 793–804.
11. Vetro, A.; Christopoulos, C.; Sun, H. Video transcoding architectures and techniques: An overview. *IEEE Signal Process. Mag.* **2003**, *20*, 18–29.
12. Xin, J.; Sun, M.; Chun, K. Motion re-estimation for MPEG-2 to MPEG-4 simple profile transcoding. In Proceedings of the International Packet Video Workshop, Pittsburgh, PA, USA, 24–26 April 2002.
13. Kalva, H. Issues in H.264/MPEG-2 video transcoding. In Proceedings of the 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 5–8 January 2004; pp. 657–659.
14. Zhou, Z.; Sun, S.; Lei, S.; Sun, M. Motion information and coding mode reuse for MPEG-2 to H.264 transcoding. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2005), Kobe, Japan, 23–26 May 2005; pp. 1230–1233.
15. Zhang, D.; Li, B.; Xu, J.; Li, H. Fast Transcoding from H.264 AVC to High Efficiency Video Coding. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Melbourne, Australia, 9–13 July 2012; pp. 651–656.
16. Peixoto, E.; Shanableh, T.; Izquierdo, E. H.264/AVC to HEVC Video Transcoder Based on Dynamic Thresholding and Content Modeling. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 99–112.
17. Peixoto, E.; Izquierdo, E. A complexity-scalable transcoder from H.264/AVC to the new HEVC codec. In Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012; pp. 737–740.
18. Bjontegaard, G. Improvements of the BD-PSNR model. *ITU-T SG16 Q* **2008**, *6*, 35.
19. Díaz-Honrubia, A.J.; Martínez, J.L.; Cuenca, P.; Gamez, J.A.; Puerta, J.M. Adaptive Fast Quadtree Level Decision Algorithm for H.264 to HEVC Video Transcoding. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 154–168.
20. Mora, E.G.; Cagnazzo, M.; Dufaux, F. AVC to HEVC transcoder based on quadtree limitation. In *Multimedia Tools and Applications*; Springer: Berlin, Germany, 2016; pp. 1–25.
21. Wang, B.; Shi, Y.; Yin, B. Transcoding of H.264 Bitstream to AVS Bitstream. In Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 24–26 September 2009; pp. 1–4.
22. Audio Video Coding Standard Workgroup of China. Available online: <http://www.avs.org.cn> (accessed on 18 September 2016). (In Chinese)

