Signal Processing: *Image Communication* 29 (2014) 951–961

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Fast encoder decision for texture coding in 3D-HEVC

Na Zhang [a,*], Debin Zhao [a], Yi-Wen Chen [b], Jian-Liang Lin [b], Wen Gao [a]

[a] *Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[b] *MediaTek, No. 1, Dusing Rd. 1, Hsinchu 30078, Taiwan*

ARTICLE INFO

Article history:
Received 6 March 2014
Received in revised form
6 June 2014
Accepted 6 June 2014
Available online 13 June 2014

Keywords:
3D video coding
Early merge mode decision
Early CU splitting termination
Inter-view correlation
3D-HEVC

ABSTRACT

As a 3D extension of the High Efficiency Video Coding (HEVC) standard, 3D-HEVC is developed to improve the coding efficiency of multi-view video. However, the improvement of the coding efficiency is obtained at the expense of a computational complexity increase. How to relieve the computational burden of the encoder is becoming a critical problem in applications. In this paper, a fast encoder decision algorithm to encode the dependent texture views is proposed, where two strategies to accelerate encoder decision by exploiting inter-view correlations are utilized. The first one is an early merge mode decision algorithm, and the second one is an early CU splitting termination algorithm. Experimental results show that the proposed algorithm can achieve 47.1% encoding time saving with overall 0.1% BD-rate reduction compared to HTM (3D-HEVC test model) version 7 under the common test condition (CTC). Both of the two strategies have been adopted into the 3D-HEVC reference software and enabled as a default encoding process under CTC.

1. Introduction

In recent years, 3D video is attracting more and more attention since users can experience realistic 3D scenes and select viewpoints interactively. Different 3D video formats or representations for delivering 3D video in applications have been designed. Texture-only formats such as conventional stereo video (CSV) and multi-view video (MVV) were first investigated. However, the bit-rate for transmission of 3D video with the texture-only format increases approximately linearly with the number of coded views. Thus, it is not feasible for transmission of a multitude of video views suitable for multi-view displays. After that, 3D video formats with few texture views and associated depth information which are also termed as MVD formats were investigated. The depth view is much easier to encode than the texture view. Based on the coded texture and depth maps, all necessary views for any 3D display can be generated, e.g., by means of depth-image-based rendering (DIBR) techniques [1].

However, for the MVD format, the multiple video sequences captured at different views and their associated depth maps still require a large storage space and a high transmission bandwidth. Therefore, an efficient 3D video compression standard is needed. As a 3D extension of the High Efficiency Video Coding (HEVC) [2], 3D-HEVC [3] with the primary focus on efficient compression of multi-view video is being developed by the Joint Collaborative Team on 3D Video Coding (JCT-3V), the joint working group of MPEG and ITU-T VCEG.

Since all cameras capture the same scene simultaneously from different viewpoints, multi-view video data contains plenty of inter-view redundancies. Block based disparity-compensated prediction (DCP) is employed to exploit inter-view correlations. In addition to DCP, 3D-HEVC also integrates

---

[*] Corresponding author.
E-mail addresses: hitnzhang@hit.edu.cn (N. Zhang), dbzhao@hit.edu.cn (D. Zhao), yiwen.chen@mediatek.com (Y.-W. Chen), jl.lin@mediatek.com (J.-L. Lin), wgao@jdl.ac.cn (W. Gao).

http://dx.doi.org/10.1016/j.image.2014.06.003
0923-5965/© 2014 Elsevier B.V. All rights reserved.

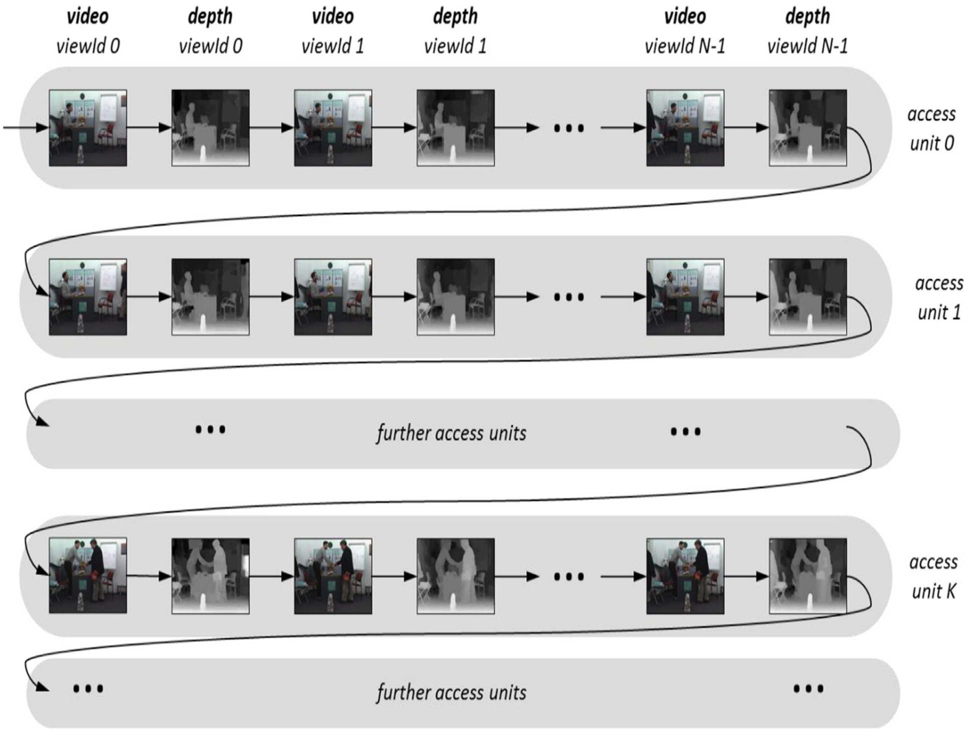

Fig. 1. Access unit structure and coding order of view components [16].

some new coding tools, such as inter-view motion prediction, inter-view residual prediction, and backward view synthesis prediction (BVSP) for texture coding; new intra coding modes and motion parameter inheritance for depth coding [4]. Moreover, a disparity vector (DV) derivation process is critical in 3D video coding for inter-view motion prediction, inter-view residual prediction, BVSP, and any other tool which needs to indicate a correspondence between inter-view pictures. In the 3D-HEVC test model version 7 (HTM-7.0), the DVs used for these coding tools are derived using either the neighboring block disparity vector (NBDV) scheme [5] or the depth oriented neighboring block disparity vector (DoNBDV) scheme [6]. NBDV derives a disparity vector for the current CU by utilizing the disparity information of several spatial and temporal neighbor blocks. DoNBDV uses NBDV to retrieve a virtual depth block in the reference view, from which a refined DV is derived by converting the maximum depth value in the virtual depth block into a disparity value, resulting in better inter-view motion prediction.

In HEVC, the encoder examines all the coding modes (up to 20 different modes) for each coding unit (CU) and evaluates a rate-distortion (RD) cost for each mode. Besides the existing coding tools in HEVC, the HTM encoder also examines the RD performance of additional coding tools, which results in a significant increase of the computational burden at the encoder.

To reduce the computational complexity of the encoding process of HEVC, several fast algorithms [7–9] have been proposed for coding a single view. There have been some fast algorithms [10–13] exploiting inter-view correlations to reduce unnecessary computations at the encoder for multi-view video coding (MVC), an extension of H.264/AVC [14].

As far as we know, there are no other fast algorithm exploiting inter-view correlations designed for the dependent texture views in 3D-HEVC. In this paper, to further relieve the computation complexity of the 3D-HEVC encoder, a fast encoder decision algorithm for texture coding of the dependent view is proposed, which utilizes two strategies to accelerate encoder decisions [15]. The first one is to make an early merge mode decision, and the second one is to make an early CU splitting termination.

The rest of this paper is organized as follows. Section 2 presents an overview of the encoding process in 3D-HEVC and reviews some fast algorithms proposed for the 3D video coding. Section 3 analyzes inter-view correlations in the coding mode and CU splitting depth level. Based on this analysis, a detailed description of the proposed fast encoder decision algorithm is presented in Section 4. Experimental results are provided in Section 5. Section 6 concludes this paper.

2. Background

This section firstly gives a brief overview of the system structure, coding structure, and encoding process in 3D-HEVC. Then the additional coding tools added to 3D-HEVC and merge mode in 3D-HEVC are described. Lastly, the existing fast encoding methods for the 3D video coding are reviewed.

2.1. System structure of 3D-HEVC

The system structure of 3D-HEVC is described as follows. The video pictures and depth maps are coded by access units as illustrated in Fig. 1 [16]. An access unit includes all video pictures and depth maps at the same time instant.

The video picture and depth map corresponding to a particular camera position are indicated by a view identifier (viewId in Fig. 1). The view identifier is also used for specifying the coding order. The view with view identifier equal to 0 is also referred to as the base view or the independent view and is coded independently of the other views using a conventional HEVC video coder. The other views are referred to as dependent views and they can be coded with additional coding tools in 3D-HEVC.

2.2. Coding structure of 3D-HEVC

Identical to that in HEVC, the coding structure in 3D-HEVC includes three basic units: coding unit (CU), prediction unit (PU), and transform unit (TU).

A picture is divided into a set of coding tree units (CTUs). The CTU is equivalent to a macroblock in H.264/AVC. The CU is represented as the leaf node of a quadtree partitioning of the CTU. It is a basic unit with a square shape which is associated with a prediction mode: intra, inter, or SKIP. A CTU may contain only one CU or may be split into four smaller CUs, and each CU could be recursively split into smaller CUs until the predefined splitting limitation is reached.

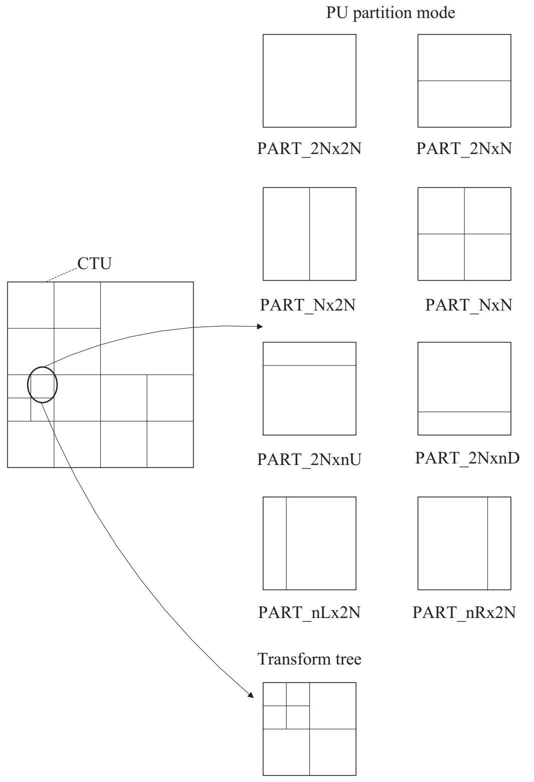

Fig. 2. Quadtree structure of a CTU and TU and possible PU partition modes.

A PU is a basic unit for prediction and has its root at the CU level. The shape of a PU is not necessarily square. Each CU may contain one, two, or four PUs according to the partition mode. The eight partition modes that can be used for an inter-coded CU are illustrated in Fig. 2. Only the PART_2Nx2N and PART_NxN partition modes are used for an intra-coded CU. For both inter-coded CU and intra-coded CU, the partition mode PART_NxN can be allowed only when the corresponding CU size is equal to the minimum CU size.

A TU is another basic unit with a square shape for transform and quantization. Multiple TUs within a CU form a quadtree structure called Residual QuadTree (RQT).

2.3. Encoding process in 3D-HEVC

In HTM, the encoder tests all the coding modes (up to 20 different modes, i.e., inter/merge/skip 2N × 2N, inter/merge 2N × N, inter/merge N × 2N, inter/merge N × N, inter/merge 2N × nU, inter/merge 2N × nD, inter/merge nL × 2N, inter/merge nR × 2N, intra 2N × 2N, intra N × N, and intra PCM) for each CU and selects the mode with the least RD cost. Furthermore, each CU could be recursively split into four sub-CUs and the coding mode of each sub-CU is again determined by examining the RD cost of all the coding modes. Whether the CU should be further split or not is also decided by comparing the RD cost of the CU to the summation of the RD costs of the four sub-CUs. The motion estimation (ME) and the computation of the RD cost for each CU are the most computationally intensive parts.

2.4. Additional coding tools added to 3D-HEVC

The independent view, which is also referred to as the base view, is coded by a conventional HEVC codec. For dependent views, additional tools exploiting inter-view correlations have been integrated into 3D-HEVC. The additionally integrated tools are described in the following.

1. To share the previously encoded texture information of reference views, the disparity-compensated prediction (DCP) has been added as an alternative to motion-compensated prediction (MCP) [16].
2. The inter-view motion prediction is employed to predict the motion information for the current block from the previously encoded motion information in the reference views [16].
3. The residual signal of the current block can also be predicted from the residual signal of the corresponding block in the reference views [16].
4. Backward view synthesis prediction (BVSP) is a technique that exploits inter-view redundancies, in which a synthesized signal is used as a reference to predict the current picture [17].
5. For the depth component, among all the above additional tools, only DCP is enabled. However, some new intra prediction modes and the motion parameter inheritance (MPI) mode are added [16].

2.5. *Merge mode in 3D-HEVC*

For inter mode, the motion parameters including the prediction direction, the reference picture index, and the motion vector (MV) need to be transmitted. In order to improve coding efficiency and reduce the bits used to encode the motion parameters, merge mode allows to merge several blocks with similar motion into a single region. As a result, all PUs within this region share the same motion information, thus the motion parameters need to be transmitted only once [18]. Merge mode contributed significantly to the coding efficiency of 3D-HEVC.

Merge mode first constructs the merging candidate list and then selects the motion information (such as MV, reference index) of the candidate with the least RD cost as that of the current block. In HEVC, the merging candidate list is constructed of several spatial candidates derived from spatial neighbor blocks and one temporal candidate derived from the temporal neighbor block. In 3D-HEVC, inter-view motion prediction, BVSP, and MPI are introduced as new merge candidates. These tools further improve the motion prediction ability of merge mode.

2.6. *Fast encoding methods for the 3D video coding*

There have been several fast encoding algorithms for 3D video coding [10–13,19,20]. Shen et al. [10] propose a fast algorithm for adjusting the search strategies for different types of macroblocks (MBs) according to the coding modes and motion vectors of the corresponding MBs in the neighbor views. By exploiting the mode complexity and motion homogeneity between neighbor views, the encoder can perform a precise search according to video content. 85% computational complexity can be saved with an increase of bit-rate of about 1.2% on average.

Shen et al. [11] also propose an early skip mode decision algorithm for MVC by exploring inter-view correlations of the prediction mode distribution. As a result, much of unnecessary computation for ME can be skipped. It can achieve 54% computational complexity saving on average with no significant loss of rate-distortion performance.

A low complexity mode decision algorithm [12] is proposed which studies inter-view correlations apart from spatial and temporal correlations in the coding prediction modes and RD costs. All the prediction modes are first categorized into five motion-activity classes. According to the adaptive threshold and predicted motion vector (PMV) derived utilizing the mode correlation between neighboring views, only one of the five classes will be chosen to identify the optimal mode. It can reduce the computational load by 73% with an increase of the total bit-rate of about 2.2% on average. However, the algorithm needs to additionally store the RD cost of the MBs selecting skip mode as the optimal mode.

A fast mode decision algorithm based on a statistical model is proposed in [13], where the inter-view correlations are also utilized. Considering both prior probabilities and time consuming of all coding modes, a hybrid model is developed utilizing several model selection rules for different requirements depending on the tradeoff between decision accuracy and time saving. It can achieve about 80% encoding time saving with 3% BD-rate increase.

A quadtree limitation and predictive coding (QTLPC) tool for depth coding [19,20] is proposed and adopted in the second JCT-3V meeting, which is based on the principle that a given depth CU cannot be split more than its collocated texture CU and the depth quadtree can therefore be predictively encoded according to the texture quadtree. It can reduce encoding time by 31% while achieving 0.3% BD-rate gain on average for coded and synthesized views.

Different from the previous fast algorithms designed for MVC, the proposed fast encoder decision algorithm combines with the new technologies such as DoNBDV and merge mode existing in 3D-HEVC. Since 3D-HEVC already performs depth quadtree (QT) limitation and prediction from the texture QT [19,20] to save encoding time and reduce bit-rates, the proposed fast algorithm is only applied to dependent texture views.

3. Statistical analysis

In this section, the probability of merge mode selection and CU splitting depth level are first reported and then the inter-view correlations in the coding mode and CU splitting depth level are analyzed. The test sequences are listed in Table 1. All the statistics are performed under the common test conditions of 3D-HEVC [21] and all the frames of each sequence are coded for the analysis. Four quantization parameters (QPs): 25, 30, 35, 40 are tested for the independent texture view. The QP offset for dependent texture views compared to the independent texture view equals 3. The depth QP has a fixed relation to the texture QP as shown in [21]. The statistics results of the whole paper are averaged across the four different QPs and across the two dependent texture views.

3.1. *Merge mode*

Table 2 shows the probability of merge mode selection for the dependent texture views of each test sequence listed in Table 1. From Table 2 it can be seen that the average probability of merge mode selection in dependent texture views is up to 97.1%. Although the probability of merge mode selection in dependent texture views is high, the encoder of 3D-HEVC still needs to examine the RD performance of all inter and intra modes. In most cases,

Table 1
Test sequences.

Test sequence	Resolution	Input views	Frames
Balloons	1024 × 768	1–3–5	300
Kendo	1024 × 768	1–3–5	300
Newspapercc	1024 × 768	2–4–6	300
GhostTownFly	1920 × 1088	9–5–1	250
PoznanHall2	1920 × 1088	7–6–5	200
PoznanStreet	1920 × 1088	5–4–3	250
UndoDancer	1920 × 1088	1–5–9	250
Shark	1920 × 1088	1–5–9	300

Table 2
Probability of merge mode selection for dependent texture views.

Sequence	Merge mode (%)
Balloons	97.3
Kendo	96.5
Newspapercc	96.8
PoznanHall2	96.4
PoznanStreet	97.5
GhostTownFly	97.5
UndoDancer	96.6
Shark	97.6
Average	97.1

Table 3
Accuracy and discovery rate of early merge mode decision by only exploiting inter-view correlations.

Sequence	Prediction accuracy (%)	Discovery rate (%)
Balloons	99.5	62.6
Kendo	99.5	62.9
Newspapercc	99.5	77.9
PoznanHall2	99.4	79.7
PoznanStreet	99.6	83.7
GhostTownFly	99.7	50.5
UndoDancer	99.1	68.8
Shark	99.7	61.3
Average	99.5	68.0

these operations for checking inter and intra modes can be avoided without sacrificing coding performance. Therefore, a fast merge mode decision for CUs is feasible and it is effective for saving encoding time.

Since several coding tools in 3D-HEVC (e.g. inter-view motion prediction, inter-view residual prediction, and BVSP) have greatly improved the coding performance by exploiting inter-view correlations, these correlations must hence be very strong. To see inter-view correlations of the coded merge modes, some statistics are performed. The accuracy of the prediction that current CU is coded as merge mode when the inter-view neighbor blocks are all coded as merge modes and the discovery rate representing the ratio of the target blocks whose inter-view neighbor blocks all coded as merge modes are presented in Table 3. To make a tradeoff between the prediction accuracy and the amount of data access, the number of inter-view neighbor blocks in this paper is set to 5. As shown in Fig. 3, the center block is the corresponding block of the current CU, and the remaining four blocks are the four directly adjacent blocks of the corresponding block in the reference view. The corresponding block in the already coded picture of the reference view is located by compensating with the DV derived by DoNBDV.

It can be observed from Table 3 that when the five inter-view neighbor blocks are all coded in merge modes, the probability of coding the current CU in merge mode (referred to as the prediction accuracy in this paper) equals 99.5%. Moreover, 68% of the exhaustive mode decision can be skipped if an early merge mode decision by only exploiting inter-view correlations is used.

3.2. CU splitting depth

To determine the best coding mode of a CTU, the HTM encoder exhaustively examines all possible CU splitting depth levels ranging from 0 to 3. Table 4 shows the probability of all four splitting depth levels of dependent texture view CUs for each test sequence in Table 1.

From Table 4 it can be seen that the probabilities of splitting depth levels 0, 1, 2, and 3 are 76.2%, 17%, 5.4%, and 1.4%, respectively. The probability of splitting depth levels 0 and 1 is up to 93.2% and the probability of the splitting depth level 3 is only 1.4%. It can be inferred that checking the prediction performance of the last two depth levels is usually unnecessary, especially for the last depth level.

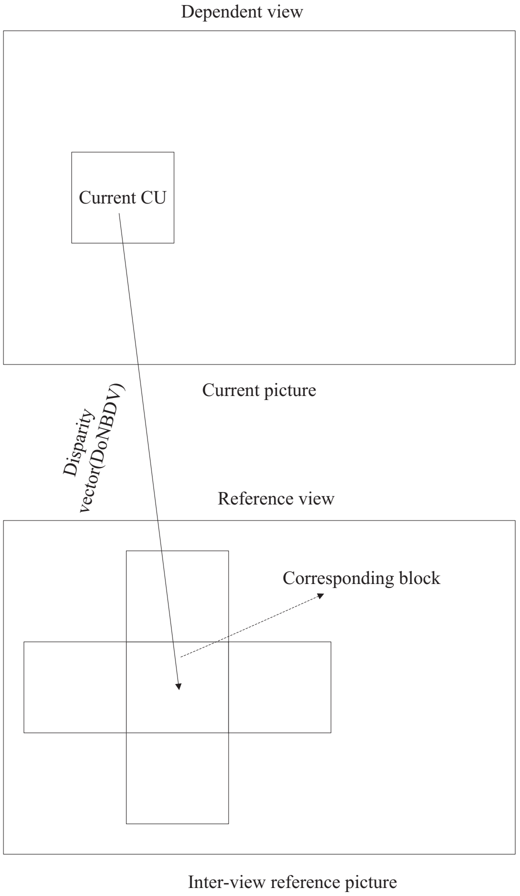

Fig. 3. The five inter-view neighbor blocks of the current CU.

To analyze the correlation between the splitting depth level of the current CU and the splitting depth levels of the inter-view neighbor blocks, Table 5 shows the probability of CUs having a splitting depth level less than or equal to the maximum splitting depth level of the five inter-view neighbor blocks as shown in Fig. 3 (referred to as the prediction accuracy in this paper). For each CU, there is the maximum splitting depth level of its five inter-view neighbor blocks. It can be seen from Table 5 that the average accuracy is 94.7%.

Table 4
Probability of each splitting depth level for dependent texture views.

Sequence	Level 0 (%)	Level 1 (%)	Level 2 (%)	Level 3 (%)
Balloons	68.2	24.1	6.1	1.5
Kendo	73.9	20.5	4.6	1.0
Newspapercc	79.2	14.7	4.8	1.4
PoznanHall2	84.3	12.7	2.8	0.2
PoznanStreet	83.8	11.2	4.0	1.0
GhostTownFly	69.0	20.2	8.6	2.2
UndoDancer	73.5	18.4	6.0	2.1
Shark	75.3	17.9	5.3	1.6
Average	76.2	17.0	5.4	1.4

Table 5
Accuracy of early CU splitting termination by only exploiting inter-view correlations.

Sequence	Prediction accuracy (%)
Balloons	93.1
Kendo	95.4
Newspapercc	96.9
PoznanHall2	92.8
PoznanStreet	96.7
GhostTownFly	89.8
UndoDancer	95.8
Shark	96.9
Average	94.7

Table 6
Encoding time ratio of each depth level for dependent texture views.

Sequence	Level 0T (%)	Level 1T (%)	Level 2T (%)	Level 3T (%)
Balloons	19.5	20.8	24.4	34.7
Kendo	20.4	21.1	24.2	33.9
Newspapercc	18.5	20.3	24.6	36.0
PoznanHall2	19.0	20.1	24.3	36.1
PoznanStreet	18.2	20.2	24.7	36.4
GhostTownFly	19.5	20.0	24.0	36.0
UndoDancer	18.9	20.2	24.4	36.0
Shark	19.3	20.0	24.3	36.0
Average	19.1	20.2	24.4	35.8

Table 6 shows the encoding time ratio of all four splitting depth levels for dependent texture views. It can be seen that 19.1%, 20.2%, 24.4%, and 35.8% of the total encoding time is spent on the splitting depth levels 0, 1, 2, and 3 respectively. The encoding time ratio becomes larger with the increase of the depth level, because the number of CUs needed to be checked in the current depth level is a quarter of that in the next depth level.

Therefore, searching solutions for early determining the best prediction mode of a CU as merge mode and omitting the larger depth level checking in certain cases is likely to be effective for fast encoding.

4. Fast encoder decision for dependent texture views

As observed in Section 3, there are strong inter-view correlations between the coding modes and splitting structures of different views. Accordingly, a fast encoder decision algorithm to encode the dependent texture view by exploiting inter-view correlations is proposed in this section. It utilizes two strategies to accelerate encoder decision. One is to make an early merge mode decision, and the other one is to make an early CU splitting termination. The proposed fast encoding algorithm is shown in Fig. 4.

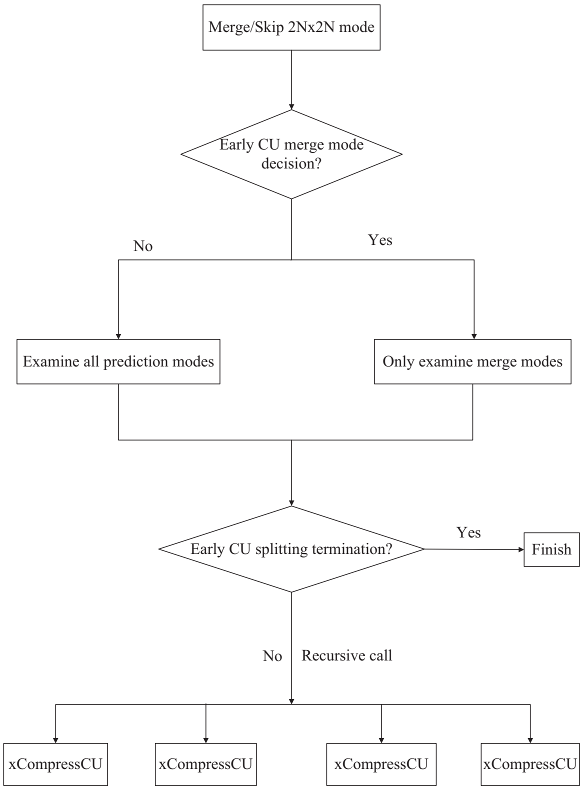

Fig. 4. The proposed fast encoding algorithm.

The details of the proposed fast encoder decision algorithm are presented as follows. In Section 4.1, the strategy of early merge mode decision is described. Section 4.2 presents the strategy of early CU splitting termination.

4.1. Early merge mode decision (EMD)

Early determining the best prediction mode of a CU as merge mode can omit the unnecessary examinations of inter and intra modes. As a result, the most time-consuming part: the motion estimation process, can be skipped. An early merge mode decision (EMD) is proposed, considering both the coding modes of the inter-view neighbor blocks and the temporary coding mode of current CU, to make a more accurate prediction about the best coding mode of current CU. The strategy of EMD includes the following two conditions.

Condition 1. All five inter-view neighbor blocks as shown in Fig. 3 are coded as merge modes.

Condition 2. The RD performance of skip mode is better than 2N × 2N merge mode for the current CU.

Table 7
Accuracy and discovery rate of the proposed early merge mode decision.

Sequence	Prediction accuracy (%)	Discovery rate (%)
Balloons	99.9	62.0
Kendo	100.0	62.1
Newspapercc	99.9	77.2
PoznanHall2	99.9	79.0
PoznanStreet	99.9	83.2
GhostTownFly	99.8	50.5
UndoDancer	99.6	68.1
Shark	99.8	61.2
Average	99.8	67.6

If the above two conditions are true, all prediction units (PUs) within the current CU will only need to examine the merge modes instead of examining all the prediction modes.

To verify the effectiveness of EMD, the prediction accuracy and discovery rate of EMD are provided in Table 7. Here, the prediction accuracy represents the probability that the best prediction mode of current CU is merge mode when the proposed two conditions hold, and the discovery rate represents the ratio of the target blocks which meet the proposed two conditions.

It can be observed from Table 7 that the average prediction accuracy of EMD is 99.8% and 67.6% of the exhaustive mode decision can be skipped due to the proposed early merge mode decision. Compared to the data presented in Table 3, the prediction accuracy increases by 0.3% and the discovery rate decreases by 0.4% due to additionally considering the condition 2.

4.2. *Early CU splitting termination (ECUST)*

Early terminating the CU splitting can avoid unnecessary RD performance checks for the larger depth levels. An early CU splitting termination (ECUST) is proposed, considering both the splitting depth levels of the inter-view neighbor blocks and the best coding mode of current CU, to make a more accurate prediction about the splitting depth level of current CU. The strategy of ECUST includes the following two conditions.

Condition 3. The CU splitting depth level of the current CU is equal to or larger than the maximum depth level of the five inter-view neighbor blocks.

Condition 4. Skip mode is selected as the best prediction mode for the current CU after checking all the possible prediction modes.

If the above two conditions are true, no further CU splitting is needed for the current CU.

To verify the effectiveness of ECUST, the prediction accuracy and discovery rate of ECUST are provided in Table 8. The prediction accuracy represents the probability that the best splitting depth level of current CU is less than or equal to the test depth level when the proposed two conditions hold. Here, the test depth level is equal to or larger than the maximum splitting depth level of the five inter-view neighbor blocks. This is because that in addition to the depth level limitation in Condition 1, if the best prediction mode of the current CU is not skip mode, it needs to be further split. The discovery rate represents the ratio of the target blocks which meet the proposed two conditions. It can be seen from Table 8 that the average prediction accuracy of ECUST is 95.6% and the discovery rate is 99.6%. Compared to the data presented in Table 5, the prediction accuracy increases by 0.9% due to additionally considering the condition 2.

Table 8
Accuracy and discovery rate of the proposed early CU splitting termination.

Sequence	Prediction accuracy (%)	Discovery rate (%)
Balloons	94.7	99.7
Kendo	97.0	99.8
Newspapercc	98.0	99.6
PoznanHall2	93.6	99.9
PoznanStreet	97.2	99.7
GhostTownFly	90.4	99.6
UndoDancer	97.0	99.3
Shark	97.6	99.7
Average	95.6	99.6

Table 9
Probability of maximum test depth level after applying the proposed early CU splitting termination.

Sequence	Test level 0 (%)	Test level 1 (%)	Test level 2 (%)	Test level 3 (%)
Balloons	49.2	30.3	13.9	6.6
Kendo	53.1	27.6	12.8	6.5
Newspapercc	63.0	20.5	10.2	6.3
PoznanHall2	75.9	15.3	6.1	2.6
PoznanStreet	69.1	16.3	8.7	6.0
GhostTownFly	40.9	31.6	18.0	9.5
UndoDancer	52.8	22.9	13.7	10.5
Shark	45.2	29.8	16.3	8.6
Average	55.5	24.3	12.8	7.4

In 3D-HEVC, CU splitting depth level 0, 1, 2, and 3 are needed to be examined for each CU. Table 9 gives the probability of the maximum test splitting depth level after applying the proposed early CU splitting termination to HTM-7.0. It can be easily seen that the proposed method could effectively reduce the test cases. As shown in Table 9, the encoding process of 55.5% of the CUs are terminated at splitting depth level 0, which means these CUs are not further split; only 7.4% of the CUs need to test their RD-costs under all the splitting depth levels: 0, 1, 2, and 3.

5. Experimental results

To verify the performance of the proposed method, both strategies have been implemented in HTM-7.0 and tested strictly in accordance with the common test conditions under the JCT-3V configurations [21]. The test sequences are listed in Table 1. The average PSNRs and bit-rates of the coded texture and depth views are measured. Besides, the average PSNR of the synthesized views between two coded views is also measured. The simulations are carried out under 64bit Linux platform with Xeon

5160 3.0 GHz CPUs. The encoding time of the anchor (HTM-7.0) and the proposed method are compared to show the complexity reduction of the proposed method. The performance of the two proposed strategies for accelerating encoder decision of dependent texture coding in HTM is demonstrated separately as well as jointly.

5.1. Results of early merge mode decision

The BD-rate performance of the proposed early merge mode decision as described in Section 4.1 is given in Table 10. As can be seen, almost no coding loss is introduced.

The encoding and motion estimation (ME) time ratio over the anchor are shown in Table 11. As can be seen from Table 11, the proposed early merge mode decision scheme reduces the average encoding time to 68.2%. Since the proposed method is only applicable to the dependent texture views, the encoding time and ME time ratio for only encoding the dependent texture views are shown in the third and the forth column of Table 11. The average encoding time and ME time ratio over the anchor for encoding the dependent texture views is 49.6% and 31.5%, respectively.

The BD-rate performance and the corresponding encoding time ratio over the anchor for only considering

Table 10
BD-rate performance of the proposed early merge mode decision compared to HTM-7.0.

Sequence	Video 1[a] (%)	Video 2[a] (%)	Video PSNR/ video *bitrate*[b] (%)	Video PSNR/ total *bitrate*[c] (%)	Synth PSNR/ total *bitrate*[d] (%)
Balloons	0.0	−0.1	0.1	0.1	0.0
Kendo	0.1	0.1	0.1	0.1	0.0
Newspapercc	−0.2	0.0	0.0	0.1	0.0
GhostTownFly	0.2	−0.1	0.1	0.1	0.0
PoznanHall2	0.3	0.6	0.5	0.5	0.1
PoznanStreet	−0.3	−0.2	0.1	0.1	0.0
UndoDancer	−0.2	−0.2	0.1	0.1	0.0
Shark	−0.1	−0.2	0.0	0.0	0.0
Average	0.0	0.0	0.1	0.1	0.0

[a] Video 1 & 2: The BD-rate performance considering Y-PSNR of view 1 and 2 (dependent views).

[b] Video PSNR/video bitrate: The BD-rate performance considering Y-PSNR of the coded texture views over the bitrate of texture data.

[c] Video PSNR/total bitrate: The BD-rate performance considering Y-PSNR of the coded texture views over the bitrate of texture data and depth data.

[d] Synth PSNR/total bitrate: The BD-rate performance considering Y-PSNR of the synthesized texture views over the bitrate of texture data and depth data.

Table 11
Run time ratio of the proposed early merge mode decision over HTM-7.0.

Sequence	Enc time (all) (%)	Enc time (dep) (%)	ME time (dep) (%)
Balloons	71.3	53.9	33.5
Kendo	71.2	54.9	36.8
Newspapercc	66.3	45.0	20.5
GhostTownFly	73.5	58.1	47.9
PoznanHall2	61.8	41.8	16.7
PoznanStreet	63.6	42.0	15.1
UndoDancer	67.4	49.5	29.8
Shark	70.2	51.5	38.9
Average	68.2	49.6	31.5

Table 12
Experimental results of only considering Condition 1 of the proposed early merge mode decision compared to HTM-7.0.

Sequence	Video 1 (%)	Video 2 (%)	Video PSNR/ video bitrate (%)	Video PSNR/ total bitrate (%)	Synth PSNR/ total bitrate (%)	Enc time (all) (%)
Balloons	0.2	0.2	0.2	0.2	0.0	71.1
Kendo	0.6	0.6	0.3	0.3	0.1	70.9
Newspapercc	0.4	0.4	0.3	0.3	0.3	65.3
GhostTownFly	0.4	0.1	0.1	0.1	0.1	73.8
PoznanHall2	0.9	2.0	1.0	1.0	0.5	61.5
PoznanStreet	0.2	−0.1	0.2	0.2	0.0	63.3
UndoDancer	0.6	1.7	0.5	0.5	0.5	67.2
Shark	0.4	0.7	0.2	0.2	0.3	69.9
Average	0.5	0.7	0.4	0.4	0.2	67.9

Table 13
BD-rate performance of the proposed early CU splitting termination compared to HTM-7.0.

Sequence	Video 1 (%)	Video 2 (%)	Video PSNR/ video bitrate (%)	Video PSNR/ total bitrate (%)	Synth PSNR/ total bitrate (%)
Balloons	−0.2	−0.3	0.2	0.2	−0.1
Kendo	−0.1	−0.3	0.2	0.3	0.0
Newspapercc	−0.3	−0.5	0.0	0.1	0.1
GhostTownFly	−1.8	−2.1	−0.1	−0.2	−0.3
PoznanHall2	−1.5	−1.3	−0.3	−0.2	−0.4
PoznanStreet	−0.9	−1.0	−0.1	−0.1	−0.2
UndoDancer	−0.8	−1.0	0.0	0.0	−0.1
Shark	−1.4	−1.5	−0.1	−0.1	−0.2
Average	−0.9	−1.0	0.0	0.0	−0.2

Table 14
Run time ratio of the proposed early CU splitting termination over HTM-7.0.

Sequence	Enc time (all) (%)	Enc time (dep) (%)
Balloons	62.3	40.2
Kendo	63.5	42.0
Newspapercc	60.4	35.1
GhostTownFly	64.8	44.3
PoznanHall2	53.4	28.2
PoznanStreet	57.9	32.2
UndoDancer	62.8	41.3
Shark	62.7	43.1
Average	61.0	38.7

Condition 1 of the proposed early merge mode decision as described in Section 4.1 are given in Table 12.

From the data provided in Tables 10–12, considering two conditions of the proposed early merge mode decision can improve the coding performance with almost the same time saving compared to only considering the condition 1. The 0.5% & 0.7% BD-rate loss on dependent views are reduced to 0% when Condition 2 is enabled, since the increase of prediction accuracy with Condition 2 is 0.3% and also the decrease of discovery rate with Condition 2 is 0.4%.

5.2. Results of early CU splitting termination

The BD-rate performance of the proposed early CU splitting termination as described in Section 4.2 is illustrated in Table 13. The proposed early CU splitting termination could even achieve 0.2% BD-rate reduction.

The encoding time ratio for encoding all the input texture and depth views and the encoding time ratio for only encoding the dependent texture views are shown in Table 14. As can be seen from Table 14, the proposed scheme of early CU splitting termination could reduce the average encoding time to 61% for encoding all the input texture and depth views. The average encoding time for only encoding dependent texture views is reduced to 38.7%.

The BD-rate performance and the corresponding encoding time ratio over the anchor for only considering Condition 1 of the early CU splitting termination as described in Section 4.2 are given in Table 15.

From the data provided in Tables 13–15, considering two conditions of the proposed early CU splitting termination can improve the coding performance with almost the same time saving compared to only considering the condition 1. The 0.6% & 0.5% BD-rate loss on dependent views are converted to −0.9% & −1.0% BD-rate gain when Condition 2 is enabled, since the increase of prediction accuracy with Condition 2 is 0.9% and also the decrease of discovery rate with Condition 2 is 0.4%.

5.3. Results of combination of early merge mode decision and early CU splitting termination

The BD-rate performance of the combination of the proposed early merge mode decision and the proposed early CU splitting termination is illustrated in Table 16. As can be seen from Table 16, the combination of the two proposed strategies could even bring 0.1% BD-rate reduction. The encoding and motion estimation (ME) time ratio over the anchor are shown in Table 17. As can be seen from Table 17, the combination of the two proposed strategies could reduce the average encoding time to 52.9% for encoding all the input texture and depth views. The average encoding time and ME time ratio over the anchor for encoding the dependent texture views is 26% and 20.9%, respectively.

From the data shown in Table 16, it can be seen that about 1% gain is achieved in the dependent views after applying the combination of the two proposed strategies. That gain disappears in the total results (video PSNR/video bitrate and video PSNR/total bitrate). It can be explained as follows:

Firstly, the encoding algorithm for the dependent view in 3D-HEVC basically follows the design in HEVC. However, such a design was proposed to optimize the HEVC encoder, not for the dependent view in 3D-HEVC. For example, since the prediction is much better on the dependent view than on the base view, it is not wise to utilize the same lambda in the D+lambda *R equation on the dependent view as that on the base view. As a result, the RDO criterion on the dependent view is not accurate enough. Therefore, the optimal mode selected by the RDO criterion for the dependent view does not necessarily have the best BD-rate performance. This is how gains are possible on dependent views with encoder shortcuts.

Table 15
Experimental results of only considering Condition 1 of the proposed early CU splitting termination compared to HTM-7.0.

Sequence	Video 1 (%)	Video 2 (%)	Video PSNR/ video bitrate (%)	Video PSNR/ total bitrate (%)	Synth PSNR/ total bitrate (%)	Enc time (all) (%)
Balloons	1.3	1.4	0.8	0.7	0.5	61.3
Kendo	1.5	1.4	0.8	0.8	0.7	61.6
Newspapercc	1.4	1.4	0.7	0.6	0.4	59.0
GhostTownFly	−0.8	−1.2	0.1	0.0	−0.2	64.9
PoznanHall2	0.5	0.1	0.5	0.4	0.2	52.8
PoznanStreet	0.0	0.1	0.2	0.2	0.0	57.3
UndoDancer	0.5	0.7	0.5	0.4	0.1	61.7
Shark	0.1	−0.2	0.2	0.2	0.3	62.0
Average	0.6	0.5	0.5	0.4	0.3	60.1

Table 16
BD-rate performance of the combination of the two proposed strategies compared to HTM-7.0.

Sequence	Video 1 (%)	Video 2 (%)	Video PSNR/ video bitrate (%)	Video PSNR/ total bitrate (%)	Synth PSNR/ total bitrate (%)
Balloons	−0.2	−0.3	0.3	0.4	−0.1
Kendo	−0.1	−0.3	0.3	0.4	0.1
Newspapercc	−0.4	−0.5	0.1	0.2	0.1
GhostTownFly	−1.8	−2.2	−0.1	−0.1	−0.3
PoznanHall2	−1.1	−1.0	0.3	0.3	−0.3
PoznanStreet	−0.9	−1.0	0.0	0.1	−0.2
UndoDancer	−1.0	−1.2	0.1	0.1	−0.1
Shark	−1.4	−1.6	0.0	0.0	−0.1
Average	−0.9	−1.0	0.1	0.2	−0.1

Table 17
Run time ratio of the combination of the two proposed strategies over HTM-7.0.

Sequence	Enc time (all) (%)	Enc time (dep) (%)	ME time (dep) (%)
Balloons	54.8	28.5	21.1
Kendo	56.5	30.7	24.4
Newspapercc	51.9	22.1	13.8
GhostTownFly	57.7	32.9	30.7
PoznanHall2	45.1	15.8	10.0
PoznanStreet	49.0	19.0	10.9
UndoDancer	53.8	28.3	21.6
Shark	54.0	28.9	25.7
Average	52.9	26.0	20.9

Secondly, the proposed method will produce a lower bit-rate with a worse image quality at the same QP for a dependent view. As shown in Table 16, it can improve the BD-rate performance on that dependent view. On the other hand, the dependent views only contribute a minor portion of the total bit-rate but take about two thirds of the average PSNR. The bit-rate reduction may not balance the image quality loss when considering the total results. Thus, the phenomenon that the coding gain achieved in the dependent views disappears in the total results (video PSNR/video bitrate and video PSNR/total bitrate) is reasonable.

Fig. 5(a) and (b) illustrates an example of RD and time saving curves under four different QPs (25, 30, 35, 40) compared to HTM-7.0. As shown in Fig. 5, the proposed

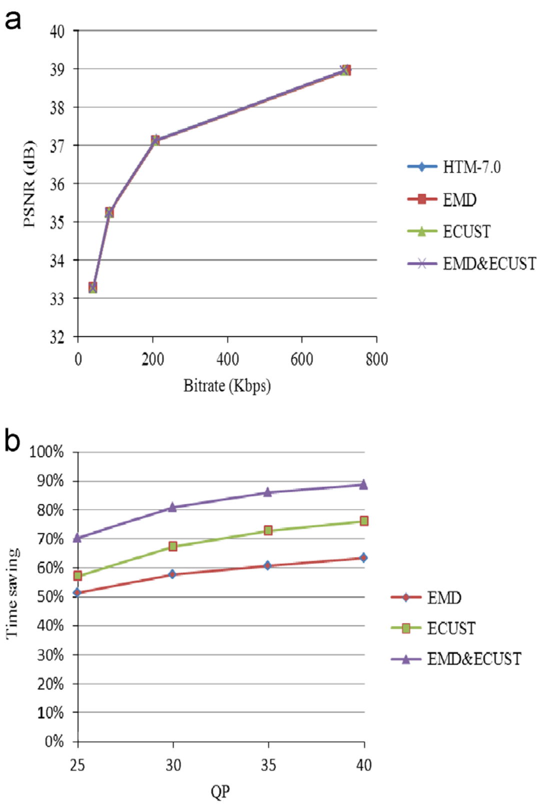

Fig. 5. Experimental results of "*PoznanStreet*" in video 1 under different QP settings (25, 30, 35, 40). (a) RD curves of "*PoznanStreet*" in video 1. (b) Time saving curves of "*PoznanStreet*" in video 1.

Table 18
Statistical results of "*PoznanStreet*" in video 1.

QP	Original (%)	EMD (%)	ECUST (%)			
	Merge mode	Discovery rate	Test level 0	Test level 1	Test level 2	Test level 3
25	96.0	75.8	53.3	21.0	14.0	11.8
30	97.5	82.6	66.5	18.2	9.3	6.0
35	98.1	86.1	75.8	14.1	6.6	3.5
40	98.7	88.5	81.2	11.9	4.9	2.0

approaches (early merge mode decision (EMD), early CU splitting termination (ECUST), and the combination (EMD& ECUST)) can achieve a consistent time saving over a large bitrate range with similar or even a little better RD performance compared to that of the HTM-7.0. Moreover, with the bit-rate decreases, the encoder runtime savings increase. This is because that with the QP increases, the probability of merge mode selection, the probability of only checking merge modes for dependent texture view CUs due to EMD, and the probability of only testing depth level 0 for dependent texture view CUs due to ECUST are all increased, as shown in Table 18.

6. Conclusion

To reduce the encoding time of the 3D-HEVC, a fast encoding algorithm including early merge mode decision and early CU splitting termination by exploiting inter-view correlations and referring the coding information of current CU is proposed. Experimental results show that applying the proposed fast encoding algorithm to the dependent texture views can achieve 47.1% encoding time reduction whereas also bringing 0.1% overall BD-rate saving. The proposed fast encoding algorithm has been adopted into the 3D-HEVC reference software to significantly speed up the encoding process.

Acknowledgments

This work was supported by the National Science Foundation of China (NSFC) under Grants 61272386 and 61390513, the Program for New Century Excellent Talents in University (NCET) of China (NCET-11-0797), and the Fundamental Research Funds for the Central Universities (Grant No. HIT.BRETIII.201221).

References

[1] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, Depth map creation and image based rendering for advanced 3DTV services providing interoperability and scalability, Signal Process.: Image Commun. 22 (February (2)) (2007) 217–234 (special issue 3DTV).
[2] B. Bross, W.-J. Han, G.J. Sullivan, J.-R. Ohm, T. Wiegand, High efficiency video coding (HEVC) text specification draft 8, JCTVC-J1003, Stockholm, July 2012.
[3] G. Tech, K. Wegner, Y. Chen, S. Yea, 3D-HEVC draft text 1, JCT3V-E1001, Vienna, July 2013.
[4] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F.H. Rhee, G. Tech, M. Winken, T. Wiegand, 3D high-efficiency video coding for multi-view video and depth data, IEEE Trans. Image Process. 22 (September (9)) (2013) 3366–3378.
[5] L. Zhang, Y. Chen, M. Karczewicz, Disparity vector derivation for multiview video and 3DV, m24937, Geneva, April 2012.
[6] C.-L. Wu, Y.-L. Chang, Y.-P. Tsai, S. Lei, Depth-oriented neighboring block disparity vector (DoNBDV) with virtual depth retrieval, JCT3V-C0131, Geneva, January 2013.
[7] R.H. Gweon, Y.-L. Lee, J. Lim, Early termination of CU encoding to reduce HEVC complexity, JCTVC-F045, Torino, July 2011.
[8] K. Choi, S.-H. Park, E.S. Jang, Coding tree pruning based CU early termination, JCTVC-F092, Torino, July 2011.
[9] L. Zhao, L. Zhang, S. Ma, D. Zhao, Fast mode decision algorithm for intra prediction in HEVC, in: IEEE Conference on Visual Communications and Image Processing, Taiwan, November 2011.
[10] L. Shen, Z. Liu, T. Yan, Z. Zhang, P. An, View-adaptive motion estimation and disparity estimation for low complexity multiview video coding, IEEE Trans. Circuits Syst. Video Technol. 20 (June (6)) (2010) 925–930.
[11] L. Shen, Z. Liu, T. Yan, Z. Zhang, P. An, Early SKIP mode decision for MVC using inter-view correlation, Signal Process.: Image Commun. 25 (February (2)) (2010) 88–93.
[12] H. Zeng, K.-K. Ma, C. Cai, Fast mode decision for multiview coding using mode correlation, IEEE Trans. Circuits Syst. Video Technol. 21 (November (11)) (2011) 1659–1666.
[13] T. Zhao, S. Kwong, H. Wang, Z. Wang, Z. Pan, C.-C.J. Kuo, Multiview coding mode decision with hybrid optimal stopping model, IEEE Trans. Image Process. 22 (April (4)) (2013) 1598–1609.
[14] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (July (7)) (2003) 560–576.
[15] N. Zhang, Y.-W. Chen, J.-L. Lin, J. An, K. Zhang, S. Lei, S. Ma, D. Zhao, W. Gao, Fast encoder decision for texture coding, JCT3V-E0173, Vienna, July 2013.
[16] L. Zhang, G. Tech, K. Wegner, S. Yea, 3D-HEVC test model 5, JCT3V-E1005, Vienna, July 2013.
[17] D. Tian, F. Zou, A. Vetro, Backward view synthesis prediction using neighbouring blocks, JCT3V-C0152, Geneva, January 2013.
[18] P. Helle, S. Oudin, B. Bross, D. Marpe, M.O. Bici, K. Ugur, J. Jung, G. Clare, T. Wiegand, Block merging for quadtree-based partitioning in HEVC, IEEE Trans. Circuits Syst. Video Technol. 22 (December (12)) (2012) 1720–1731.
[19] J. Jung, E. Mora, Depth quadtree prediction for 3DHTM 4.1, JCT3V-B0068, Shanghai, October 2012.
[20] E. Mora, J. Jung, M. Cagnazzo, B. Pesquet-Popescu, Initialization, limitation and predictive coding of the depth and texture quadtree in 3D-HEVC, IEEE Trans. Circuits Syst. Video Technol. (2013), http://dx.doi.org/10.1109/TCSVT.2013.2283110.
[21] D. Rusanovskyy, K. Müller, A. Vetro, Common test conditions of 3DV core experiments, JCT3V-D1100, Incheon, April 2013.