# Novel Bitrate Saving and Fast Coding for Depth Videos in 3D-HEVC

Kuo-Liang Chung, *Senior Member, IEEE*, Yong-Huai Huang, *Member, IEEE*, Chien-Hsiung Lin, and Jian-Ping Fang

***Abstract*— With the advance of coding and network technology, the multiview video (MVV) system has received growing attention in the 3-D TV market since it can provide consumers with more realistic scenes from different viewpoints. In the MVV system, each video sequence consists of one color-map video and one depth video to be encoded and the depth-image-based rendering technique is often used to create/synthesize virtual views for different viewpoints. To attain the same quality of virtual views, encoding depth videos in a fast and bitrate-saving manner is crucial. Based on the depth no-synthesis-error model developed by Zhao *et al*., this paper proposes a new intra-/inter-prediction scheme first and then a fast quadtree structure determination scheme for encoding depth videos in 3-D High Efficiency Video Coding (3D-HEVC). The first proposed scheme has bitrate-saving merit because for each coding unit, each depth value is modified to approach the predicted intra-/inter-depth value as closely as possible, but without causing any synthesis errors in the resultant virtual views. The second proposed scheme has the merit of low computational cost because it can terminate the quadtree structure determination for coding tree units as early as possible. Based on four typical test video sequences, the empirical results demonstrate that on average, our proposed encoding method for depth videos has 16.3% bitrate and 13.8% encoding-time improvement ratios when compared with the traditional method in 3D-HEVC test model, while preserving the quality of the rendered virtual views. In addition, our proposed method outperforms that of Lee *et al*. in terms of the required bitrate and encoding time.**

***Index Terms*— 3-D High Efficiency Video Coding (3D-HEVC), depth no-synthesis-error (D-NOSE) model, depth video coding, depth-image-based rendering (DIBR), encoding time, multiview video (MVV) system, quadtree structure decision, quality of rendered view.**

## I. Introduction

3-D TV systems [1] have received growing attention in the consumer electronics market since they provide viewers with 3-D scenes via the stereoscopic video sequences [2]. In some instances, e.g., watching sport, the audience may hope to watch 3-D scenes from different viewpoints. Multiview video (MVV) techniques have thus been developed to satisfy these demands. In general, there are two kinds of video formats used in MVV systems, namely, the MVV format [3] and the MVV plus depth (MVD) format [4].

Compared with encoding MVV sequences, encoding MVD video sequences require less storage and transmission bandwidth because only a few real views, each containing one color-map video sequence and the corresponding depth video sequence, need to be encoded. Each color-map video sequence is acquired using the color camera, while the corresponding depth video sequence is directly acquired by the depth sensor or by a depth estimation method [5], [6]. On the decoder side, the virtual views can be synthesized by the depth-image-based rendering (DIBR) technique [4]. Due to its merits of practicality and effectiveness, the MVD format has increasingly become popular in the 3-D TV broadcasting market.

When encoding MVD video sequences for 3-D TV broadcasting, the three metrics, bitrate, quality, and encoding time, are commonly used for performance comparison. In the JPEG and H.264 [7] environments, several efficient methods have been developed for coding depth maps in the MVD video sequences and they can be roughly classified into five categories. The first category encodes depth maps by referring the information from the corresponding color maps since the color and the depth maps of an MVD video sequence are usually highly correlated with each other [8]–[12]. Maitre *et al*. [8] and Daribo *et al*. [9] proposed to reuse the motion vectors found in the color map for the corresponding depth map so as to reduce the bitrate requirement. In [10] and [11], based on the encoding modes used in the color map, two fast mode decision methods for encoding the depth maps were proposed. Instead of encoding the depth maps, Temel *et al*. [12] encoded the depth cues, which were extracted from the depth maps by referring the luminance, chrominance, texture, and motion information from the color maps, to achieve a bitrate-saving effect and utilized the decoded depth cues to reconstruct the depth maps for synthesizing the virtual views. The second category provides the new encoding modes for edge regions to reduce the synthesis errors caused by the compressed depth values [13], [14]. Liu *et al*. [13] proposed a sparse dyadic mode to effectively extract the edge information in each depth block, resulting in a low bitrate and high rendering quality. Chen *et al*. [14] proposed an efficient intra-coding method, which can locate the depth blocks containing the object boundaries, and then 29 modified prediction modes were used to reduce the intra-prediction errors. The third category improves the

Manuscript received September 4, 2014; revised January 23, 2015, March 22, 2015, and June 6, 2015; accepted August 3, 2015. Date of publication August 26, 2015; date of current version September 30, 2016. The work of K.-L. Chung was supported by the Ministry of Science and Technology, China, under Contract NSC102-2221-E-011-055-MY3. The work of Y.-H. Huang was supported by the Ministry of Science and Technology, China, under Contract MOST103-2221-E228-004. This paper was recommended by Associate Editor F. Wu. *(Corresponding author: Yong-Huai Huang.)*

K.-L. Chung, C.-H. Lin, and J.-P. Fang are with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10672, Taiwan.

Y.-H. Huang is with the Department of Electronic Engineering, Jinwen University of Science and Technology, New Taipei City 23154, Taiwan (e-mail: yonghuai@ms28.hinet.net).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2473296

performance of depth map coding by considering the correlation between different views in the MVD video sequences [15], [16]. Lee *et al*. [15] exploited the temporal and inter-view correlation in the color maps to skip regular coding of similar blocks in the corresponding depth maps so as to reduce the coding time and bitrate requirements without degrading the decoded depth map quality. Li *et al*. [16] proposed the joint view filtering method to enhance inter-view consistency of the depth maps in the MVD video sequences. The fourth category preprocesses the depth maps to reduce the bitrate requirement in encoding depth maps [17]–[21]. De Silva [17] proposed a bitrate-saving method by smoothing the depth maps prior to depth map coding. In addition to the smoothing preprocess, some methods [18]–[21] downsampled depth maps at the encoder side to the end of bitrate saving, and then a special upsampling filter was applied after decoding to reconstruct the depth edge information. The fifth category takes the synthesis errors into account to preserve the quality of the synthesized virtual view [22]–[24]. Merkle *et al*. [22] studied the influence of geometry distortions resulting from the coding artifacts of H.264/MVC and a platelet-based coding algorithm. Cheung *et al*. [23] introduced the concept of the do not care region to manipulate the depth value within it without causing severe synthesis errors and then applied the proposed range to obtain sparse depth representations in the transform domain, leading to a coding gain for depth maps. Zhao *et al*. [24] proposed a new bitrate-saving method by smoothing the depth value of each pixel for all intra prediction via Gaussian filtering. In their method, all the smoothed depth values are forced to fall within the allowable range, which is determined by the depth no-synthesis-error (D-NOSE) model, without causing any synthesis errors.

Although the previous methods for encoding depth maps in H.264 have bitrate-saving and/or fast encoding merits, the coding framework of H.264 limits their bitrate and quality performance. With the release of the next-generation video coding standard, High Efficiency Video Coding (HEVC) [25], the HEVC video coder provides better compression performance than the H.264/AVC coder. To extend the coding scope of HEVC from 2-D to 3-D videos, the HEVC extension for 3-D video coding, named 3D-HEVC, was proposed to compress the MVD video sequences [26]–[28]. In 3D-HEVC, the color maps are encoded using three coding strategies, namely, disparity-compensated prediction, inter-view motion prediction, and inter-view residual prediction. For encoding depth maps, the other three coding strategies, namely, the new intra-coding modes, modified motion compensation, and motion parameter inheritance, are utilized. Further, the synthesis errors are considered in the rate–distortion (RD) optimization process to improve the quality of the virtual views. Accordingly, by employing the above coding strategies in 3D-HEVC, the MVD video sequences can be encoded with a high compression ratio. However, these coding strategies involve several time-consuming steps, leading to high encoding-time complexity. To improve the practicability of 3D-HEVC, how to reduce the encoding time without harming the quality and bitrate performance, or even how to reduce both the encoding time and bitrate without degrading the quality, is rather desired. To this end, several previous methods [15], [18]–[21] have provided effective coding strategies to improve the required encoding time and/or bitrate in H.264, but they may not be suitable for 3D-HEVC. For instance, the effectiveness of Lee *et al*.'s method [15] may be weakened because the used inter-view correlations are considered in 3D-HEVC; the downsampled depth maps suggested in [18]–[21] can reduce the encoding complexity, but some important depth information may be lost during the downsampling process, which may result in severe synthesis errors. For compressing depth maps using 3D-HEVC, this paper aims to develop new and efficient coding strategies to reduce both the encoding time and the bitrate without causing synthesis errors.

In this paper, we propose a novel coding method for compressing depth videos of MVD sequences in 3D-HEVC. The proposed coding method consists of two new D-NOSE-based schemes, the bitrate-saving intra-/inter-prediction scheme and the fast quadtree structure determination scheme. Let us briefly describe the difference between our proposed first scheme and the previous method in [24]. Zhao *et al*. [24] smoothed the depth value of each pixel for all intra predictions via Gaussian filtering until all the smoothed depth values fell within the allowable range. However, they did not propose a method for inter prediction. In our first proposed scheme, we modify each depth value such that within the allowable range, all the modified depth values can approach the intra- and inter-predicted depth values, respectively, as closely as possible, leading to a clear bitrate-saving effect. Besides the bitrate-saving effect contributed in the first proposed scheme, our second proposed scheme, the fast quadtree structure determination scheme contributes the merit of low encoding time. Based on four typical test video sequences, our proposed improved coding method for depth videos outperforms the traditional method in 3D-HEVC test model (3D-HTM) and Lee *et al*.'s method [15] in terms of the bitrate and encoding-time metrics.

The rest of this paper is organized as follows. In Section II, we briefly review the MVV system with MVD format and the used 3D-HEVC standard. In Section III, we present the proposed new bitrate-saving and fast method for compressing depth videos. In Section IV, some experiments are carried out to illustrate the bitrate and execution-time merits of the proposed method. In Section V, the conclusion is drawn.

## II. Multiview Video System With MVD Format and the Used 3D-HEVC

This section consists of two subsections. In the first subsection, we briefly introduce the MVV system with MVD format, and explain how the DIBR technique can be used to synthesize the virtual views using the color and depth maps in the MVD video sequence. In the second subsection, we briefly introduce the used 3D-HEVC for compressing MVD video sequences, and point out the bitrate and encoding-time issues when encoding depth videos.

### A. Multiview Video System With MVD Format

To reduce the storage space and transmission bandwidth required in the MVV system, it is suggested that the

MVD format uses fewer cameras to capture real views, and then the virtual views can be synthesized with the captured real views using the DIBR technique [4]. Eventually, the real views and the generated virtual views are used together for 3-D display.

Let the $M$ real views, which are generated by $M$ cameras, be denoted by $v_0$, $v_1$, ..., and $v_{M-1}$. For simplicity, the real view is also called the view. For each view, suppose there are $N$ time slots for both the color and depth maps. Let $f_{m,n}$ and $g_{m,n}$ denote the color map and the corresponding depth map, respectively, captured at the $n$th time slot of the $m$th view where $0 \le n \le N-1$ and $0 \le m \le M-1$. In addition, the color and depth values of $f_{m,n}$ and $g_{m,n}$ at position $(x, y)$, $0 \le x \le W-1$ and $0 \le y \le H-1$, are denoted by $f_{m,n}(x, y)$ and $g_{m,m}(x, y)$, respectively. In [29], the quantized depth value $g_{m,n}$ is obtained by

$$\begin{aligned} & g_{m,n}(x, y) \\ & \quad = Q(z_{m,n}(x, y)) \\ & \quad = \left\lfloor 255 \times \frac{Z_{\text{near}}}{z_{m,n}(x, y)} \times \frac{Z_{\text{far}} - z_{m,n}(x, y)}{Z_{\text{far}} - Z_{\text{near}}} + 0.5 \right\rfloor \end{aligned} \quad (1)$$

where $z_{m,n}(x, y)$ denotes the original depth value, $Z_{\text{near}}$ and $Z_{\text{far}}$ denote the nearest and farthest depth values, and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to $x$. Here, the values 0 and 255 of the quantized depth value represent, respectively, the farthest and nearest distances from the $m$th camera.

Without losing generality, we assume that the $M$ cameras are aligned horizontally such that the vertical disparity between the real view and the neighboring virtual view can be ignored in the warping step of the DIBR technique. We now explain how to synthesize the virtual view between two adjacent views $v_m$ and $v_{m+1}$. As shown in (2), each color pixel $f_{m,n}(x, y)$ in the $n$th color map of $v_m$ is warped to the position $(x', y')$ of the virtual view where

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + d^h_{m,n}(x, y)) \\ y \end{pmatrix} \quad (2)$$

the horizontal disparity $d^h_{m,n}(x, y)$ is defined by

$$d^h_{m,n}(x, y) = \frac{f\ell}{Q^{-1}(g_{m,n}(x, y))} \quad (3)$$

and the warped color pixel is denoted by $f^r_{m,n}(x', y')$.

In (3), $f$ is the focal length of the camera and $\ell$ denotes the baseline length between the real view and the virtual view. The dequantized depth value of $g_{m,n}(x, y)$, $Q^{-1}(g_{m,n}(x, y))$, is defined by

$$Q^{-1}(g_{m,n}(x, y)) = \frac{1}{\frac{g_{m,n}(x,y)}{255}\left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}\right) + \frac{1}{Z_{\text{far}}}}. \quad (4)$$

Since mapping $f_{m,n}(x, y)$ to $f^r_{m,n}(x', y')$ is not a one-to-one function, the above warping process often generates holes. The hole filling process [30], [31] can be used to interpolate the holes.

### *B. Used 3D-HEVC*

In 3D-HEVC [26]–[28], the $M$ views of one MVD video sequence are classified into two types: 1) the independent view $v_0$ and 2) the dependent views containing the other $M-1$ views. The color and depth maps in the independent view are first encoded by HEVC and the depth map coder, respectively, without referring to the dependent views, $v_1$, $v_2$, ..., and $v_{M-1}$.

Considering the redundancy among adjacent views at the same time instance, the color and depth maps in the dependent views could be encoded using the inter-view prediction to improve the quality and bitrate performance. In the 3D-HEVC standard, only the decoding process is defined, so we need an encoder to implement our proposed encoding method for depth videos. Here, we adopt the encoding process defined in the 3D-HTM and follow the provided example implementation to implement the needed encoder such that the encoded bitstreams can be handled by the 3D-HEVC standard-compliant decoder.

In 3D-HEVC, each of the color and depth maps is partitioned into a set of $64 \times 64$ blocks, each of which is called a coding tree unit (CTU). For encoding each CTU, the 3D-HTM adopts the minimal RD cost criterion to accomplish the quadtree partition process for each CTU. According to the complete quadtree representation, each CTU is partitioned into a set of blocks where each block is called the coding unit (CU). The root node of the quadtree is a $64 \times 64$ CU, $C^0_{64\times64}$, which can be partitioned into four $32 \times 32$ CUs, $C^0_{32\times32}$, $C^1_{32\times32}$, ..., and $C^3_{32\times32}$. These CUs can be further partitioned into sixteen $16 \times 16$ CUs, $C^0_{16\times16}$, $C^1_{16\times16}$, ..., and $C^{15}_{16\times16}$. Finally, the sixteen CUs are partitioned into 64 $8 \times 8$ CUs, $C^0_{8\times8}$, $C^1_{8\times8}$, ..., and $C^{63}_{8\times8}$. Each CU can be encoded using intra or inter prediction. The intra prediction reduces the spatial redundancy, while the inter prediction reduces the temporal redundancy or the redundancy between two adjacent views. Both the prediction schemes partition each CU into one or more prediction units (PUs), and then, based on the minimal RD cost criterion, either intra or inter prediction is selected as the best prediction mode. Further, the bottom-up tree merging process is adopted to determine the quadtree structure for encoding the CTU with minimal RD cost.

In the above description of the quadtree partition, it is clear that for encoding depth videos, designing a new intra-/inter-prediction scheme with minimal prediction errors would lead to a bitrate-saving effect, and designing a faster quadtree structure determination scheme would result in an encoding-time reduction effect. In the following section, we propose a novel intra-/inter-prediction scheme and a new fast quadtree structure determination scheme, but without causing any synthesis errors in the resultant virtual views.

## III. Proposed Effective Prediction Scheme and Quadtree Structure Determination for Coding Depth Maps in 3D-HEVC

This section consists of two subsections. In the first subsection, based on the minimization of prediction errors, a bitrate-saving intra-/inter-prediction scheme is presented for coding depth maps in 3D-HEVC. In the second subsection, we propose a fast quadtree structure determination scheme

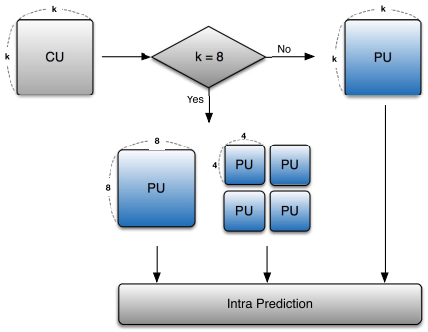



Fig. 1. PUs used in intra prediction.

for early termination of the quadtree-based partition process in the depth map coding.

### A. Proposed Bitrate-Saving Intra-/Inter-Prediction Scheme

*1) For Intra Prediction:* As shown in Fig. 1, for performing the intra prediction on one $8 \times 8$ CU, $C^i_{8\times 8}$, the 3D-HEVC treats $C^i_{8\times 8}$ as an $8 \times 8$ PU denoted by $P^{i,0}_{8\times 8}$ or treats it as four $4 \times 4$ PUs, $P^{i,0}_{4\times 4}$, $P^{i,1}_{4\times 4}$, $P^{i,2}_{4\times 4}$, and $P^{i,3}_{4\times 4}$. However, the 3D-HEVC treats the $64 \times 64$, $32 \times 32$, and $16 \times 16$ CUs as the same-sized PUs, $P^{i,0}_{k\times k}$, $k \in \{64, 32, 16\}$. In intra prediction, there are in total 39 intra-prediction modes used to predict the depth values for each PU, namely, the 35 directional interpolation modes and the four depth modeling modes. It is known that the fewer prediction errors there are, the less the bitrate requirement is. In what follows, a bitrate-saving intra-prediction scheme based on prediction error minimization is presented.

The D-NOSE model [24] indicates that we can change the quantized depth value $g_{m,n}(x, y)$ to a value within the allowable range $R$ without degrading the synthesized virtual view. As described in Section II-A, the virtual view can be synthesized using the warping process, and hence the allowable range $R$ of the quantized depth value $g_{m,n}(x, y)$ is determined by the horizontal disparity $d^h_{m,n}(x, y)$ in (3) and the used rounding and precision strategies. Since the warped pixel position could be represented by different precisions, the $\lambda$-rounding ($0 < \lambda \leq 1$) with precision $1/P$, where $P = 1$, denotes the integer precision and $P > 1$ denotes the subpixel precision, is used to represent the horizontal disparity $d^h_{m,n}(x, y)$. The horizontal disparity representation [24] of the depth is given by

$$\tilde{d}^h_{m,n}(x, y) = \frac{\lceil (d^h_{m,n}(x, y)) - \lambda) \times P \rceil}{P} \tag{5}$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to $x$. Since the disparity representation $\tilde{d}^h_{m,n}(x, y)$ is not a one-to-one mapping, there may exist different depth values which generate the same disparity representation of the depth. Therefore, there exists a unique range $R = [\ell(g_{m,n}(x, y)), u(g_{m,n}(x, y))]$ with

$$\ell(g_{m,n}(x, y)) = \min \left\{ v \left| \frac{\left\lceil \left( \frac{f\ell}{Q^{-1}(v)} - \lambda \right) \times P \right\rceil}{P} = \tilde{d}^h_{m,n}(x, y) \right. \right\} \tag{6}$$

$$u(g_{m,n}(x, y) = \max \left\{ v \left| \frac{\left\lceil \left( \frac{f\ell}{Q^{-1}(v)} - \lambda \right) \times P \right\rceil}{P} = \tilde{d}^h_{m,n}(x, y) \right. \right\} \tag{7}$$

and it has been shown that for each quantized depth value $g_{m,n}(x, y)$, varying $g_{m,n}(x, y)$ in the allowable range $R$ will not cause any synthesis errors in the virtual view. According to the range $R$ derived from the D-NOSE model, we have developed an intra-prediction scheme that minimizes the prediction errors to achieve the bitrate-saving purpose during the depth map coding.

For each $k \times k$ PU, $P^{i,j}_{k\times k}$, $j = 0$ for $k \in \{62, 32, 16, 8\}$ and $j \in \{0, 1, 2, 3\}$ for $k = 4$, the quantized depth value $g_{m,n}(x, y)$ can be predicted using one of the 39 intra-prediction modes, and we let $\mathrm{PD}^{p,k\times k}_{\mathrm{intra}}(g_{m,n}(x, y))$ denote the predicted depth value of $g_{m,n}(x, y)$ using the $p$th intra-prediction mode, $0 \leq p \leq 38$. The proposed intra-prediction scheme aims to modify each $g_{m,n}(x, y)$ to the one within the allowable range $R$ such that the modified depth value can approach $\mathrm{PD}^{p,k\times k}_{\mathrm{intra}}(g_{m,n}(x, y))$ with minimal prediction error; in the meantime it will not cause any synthesis errors in the virtual view. Consequently, we modify $g_{m,n}(x, y)$ by

$$g_{m,n}(x, y) = \arg \min_{g' \in R} \left| g' - \mathrm{PD}^{p,k\times k}_{\mathrm{intra}}(g_{m,n}(x, y)) \right|. \tag{8}$$

In order to obtain the best modified depth value by (8) in constant time, a lookup table is built up in advance to store the allowable lower bound $\ell(g_{m,n}(x, y))$ and upper bound $u(g_{m,n}(x, y))$ of $g_{m,n}(x, y)$, as shown in (6) and (7), for $0 \leq g_{m,n}(x, y) \leq 255$. After examining the 39 intra-prediction modes using the proposed prediction scheme in (8) for each PU, the resultant one with the minimal RD cost is selected as the best intra-prediction mode. Let $\mathrm{RD}_{\mathrm{intra}}(P^{i,j}_{k\times k})$ denote the RD cost using the best intra-prediction mode to predict $P^{i,j}_{k\times k}$. The RD cost of the corresponding CU, $C^i_{k\times k}$, is calculated by

$$\mathrm{RD}_{\mathrm{intra}}\left(C^i_{k\times k}\right) = \begin{cases} \min\left(\mathrm{RD}_{\mathrm{intra}}\left(P^i_{8\times 8}\right), \sum_{j=0}^{3} \mathrm{RD}_{\mathrm{intra}}\left(P^{i,j}_{4\times 4}\right)\right), & k = 8 \\ \mathrm{RD}_{\mathrm{intra}}\left(P^{i,0}_{k\times k}\right), & \text{otherwise.} \end{cases} \tag{9}$$

The intra-prediction result will be compared with the inter-prediction result, which is described in Section III-A2, and then based on the comparative result, the best mode is finally determined.

We now take a real $4 \times 4$ depth block example as a PU to show the bitrate-saving effect of the proposed intra-prediction scheme. As shown in Fig. 2(a), we have a $4 \times 4$ current depth block associated with 17 reference neighboring pixels. Fig. 2(b) and (c) shows, respectively, the predicted depth block after performing the vertical prediction mode on the current block and the prediction error between Fig. 2(a) and (b). Fig. 2(d) and (e) gives, respectively, the lower and the upper bounds of the depth values in the depth block by (6) and (7) with the nearest depth value $Z_{\mathrm{near}} = 2228.75$, the farthest depth values $Z_{\mathrm{far}} = 156\,012.21$, the baseline length $\ell = 38.66$,

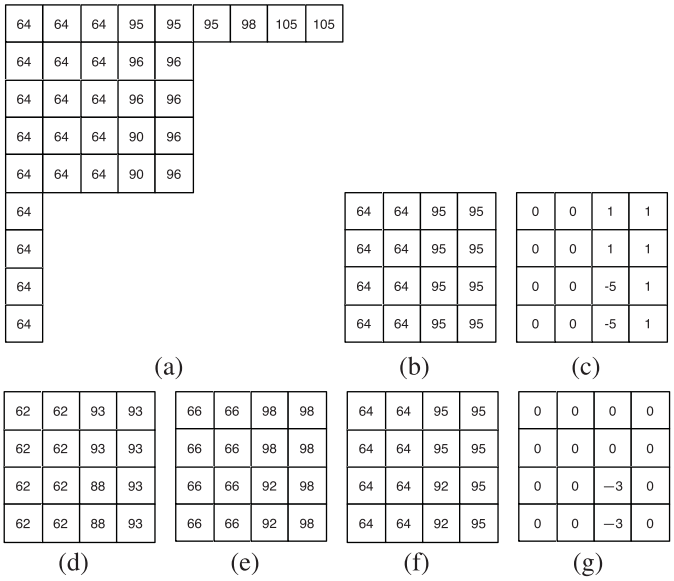



Fig. 2. $4 \times 4$ depth block example to demonstrate the prediction error reduction merit of our proposed intra-prediction scheme. (a) Current depth block and 17 reference neighboring pixels. (b) Predicted depth block by the vertical prediction mode. (c) Prediction-error block between (a) and (b). (d) Lower-bound block. (e) Upper-bound block. (f) Modified depth block by the proposed scheme. (g) Prediction-error block between (a) and (f).

the focal length $f = 2017.81$, and the rounding parameter $\lambda = 0.5$. For convenience, the lower-bound block and the upper-bound block of Fig. 2(a) are shown as Fig. 2(d) and (e), respectively. Using the lookup table technique mentioned above, the lower-bound block and the upper-bound of the current depth block can be built up in linear time. Fig. 2(f) illustrates the modified depth block using the proposed intra-prediction scheme, and the corresponding prediction-error block is shown in Fig. 2(g). Comparing Fig. 2(g) with Fig. 2(c), the substantial effect on the prediction error reduction of the proposed scheme can be observed.

*2) For Inter Prediction:* Besides the intra prediction, the prediction errors of the inter prediction can also be minimized to achieve the bitrate-saving purpose. For each $C^i_{k\times k}$, $k \in \{64, 32, 16, 8\}$, the 3D-HEVC performs the inter prediction according to the eight partition ways as shown in Fig. 3. In the first partition way, the inter prediction is directly performed on the $k \times k$ PU, $P^{i,0}_{k\times k}$. In the three equal-sized partition ways, each $k \times k$ CU is partitioned into: 1) four $k/2 \times k/2$ PUs, $P^{i,0}_{k/2\times k/2}$, $P^{i,1}_{k/2\times k/2}$, $P^{i,2}_{k/2\times k/2}$, and $P^{i,3}_{k/2\times k/2}$; 2) two $k/2 \times k$ PUs, $P^{i,0}_{k/2\times k}$ and $P^{i,1}_{k/2\times k}$; or 3) two $k\times k/2$ PUs, $P^{i,0}_{k\times k/2}$ and $P^{i,1}_{k\times k/2}$. In the four unequal-sized partition ways, each CU is partitioned into: 1) the $k/4\times k$ left-side PU, $P^{i,0}_{k/4\times k}$, and the $3k/4\times k$ right-side PU, $P^{i,1}_{3k/4\times k}$; 2) the $3k/4\times k$ left-side PU, $P^{i,0}_{3k/4\times k}$, and the $k/4\times k$ right-side PU, $P^{i,1}_{k/4\times k}$; 3) the $k \times k/4$ upper-side PU, $P^{i,0}_{k\times k/4}$, and the $k \times 3k/4$ bottom-side PU, $P^{i,1}_{k\times 3k/4}$; or 4) the $k \times 3k/4$ upper-side PU, $P^{i,0}_{k\times 3k/4}$, and the $k \times k/4$ bottom-side PU, $P^{i,1}_{k\times k/4}$.

For each available $k_1 \times k_2$ PU, $P^{i,j}_{k_1\times k_2}$, mentioned in the last paragraph, the inter prediction is performed on the current PU by motion-compensated or disparity-compensated prediction.

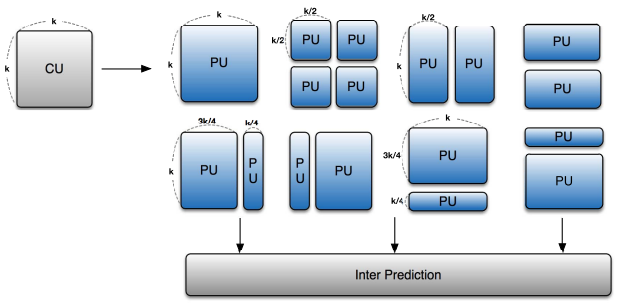



Fig. 3. PUs used in inter prediction.

Motion-compensated prediction predicts depth values for the current PU by referring to the best matched block among these reference depth maps, $g_{m,r_0}$, $g_{m,r_1}, \ldots,$ and $g_{m,r_{L-1}}$, in the current view $v_m$. Disparity-compensated prediction refers to these $m$ reference depth maps, $g_{m-1,n}$, $g_{m-2,n}, \ldots,$ and $g_{0,n}$, at the same instance in the previous $m$ views to find the best matched block of the current PU. Applying the motion-compensated prediction to the PU $P^{i,j}_{k_1\times k_2}$, each depth value $g_{m,n}(x, y)$ in $P^{i,j}_{k_1\times k_2}$ is predicted by $\mathrm{PD}^{0,k_1\times k_2}_{\mathrm{inter}}(g_{m,n}(x, y))$. Similarly, $g_{m,n}(x, y)$ in $P^{i,j}_{k_1\times k_2}$ is predicted by $\mathrm{PD}^{1,k_1\times k_2}_{\mathrm{inter}}(g_{m,n}(x, y))$ when the disparity-compensated prediction is used. For minimizing the inter-prediction error, we modify $g_{m,n}(x, y)$ to the one $g'$ within the allowable range $R$ to approach the predicted depth value $\mathrm{PD}^{p,k_1\times k_p}_{\mathrm{inter}}(g_{m,n}(x, y))$, $p = \{0, 1\}$. Thus, $g_{m,n}(x, y)$ is modified by

$$g_{m,n}(x, y) = \arg\min_{g'\in R} \left|g' - \mathrm{PD}^{p,k_1\times k_2}_{\mathrm{inter}}(g_{m,n}(x, y))\right|. \quad (10)$$

After examining all modes, the best inter-prediction mode of $P^{i,j}_{k_1\times k_2}$ is the one with the minimal RD cost. Let $\mathrm{RD}_{\mathrm{inter}}(P^{i,j}_{k_1\times k_2})$ denote the RD cost using the best inter-prediction mode to predict $P^{i,j}_{k_1\times k_2}$. The RD cost $\mathrm{RD}_{\mathrm{inter}}(C^i_{k\times k})$ is calculated by

$$\mathrm{RD}_{\mathrm{inter}}(C^i_{k\times k}) = \min \begin{Bmatrix} \mathrm{RD}_{\mathrm{inter}}(P^{i,0}_{k\times k}) \\ \sum_{j=0}^{3} \mathrm{RD}_{\mathrm{inter}}(P^{i,j}_{k/2\times k/2}) \\ \sum_{j=0}^{1} \mathrm{RD}_{\mathrm{inter}}(P^{i,j}_{k\times k/2}) \\ \sum_{j=0}^{1} \mathrm{RD}_{\mathrm{inter}}(P^{i,j}_{k/2\times k}) \\ \mathrm{RD}_{\mathrm{inter}}(P^{i,0}_{k/4\times k}) + \mathrm{RD}_{\mathrm{inter}}(P^{i,1}_{3k/4\times k}) \\ \mathrm{RD}_{\mathrm{inter}}(P^{i,0}_{3k/4\times k}) + \mathrm{RD}_{\mathrm{inter}}(P^{i,1}_{k/4\times k}) \\ \mathrm{RD}_{\mathrm{inter}}(P^{i,0}_{k\times k/4}) + \mathrm{RD}_{\mathrm{inter}}(P^{i,1}_{k\times 3k/4}) \\ \mathrm{RD}_{\mathrm{inter}}(P^{i,0}_{k\times 3k/4}) + \mathrm{RD}_{\mathrm{inter}}(P^{i,1}_{k\times k/4}) \end{Bmatrix}. \quad (11)$$

According to the two RD costs obtained by the proposed scheme for inter prediction and intra prediction, $\mathrm{RD}_{\mathrm{intra}}(C^i_{k\times k})$ and $\mathrm{RD}_{\mathrm{inter}}(C^i_{k\times k})$, the final RD cost of $C^i_{k\times k}$ is obtained by

$$\mathrm{RD}(C^i_{k\times k}) = \min\left(\mathrm{RD}_{\mathrm{intra}}(C^i_{k\times k}), \mathrm{RD}_{\mathrm{inter}}(C^i_{k\times k})\right). \quad (12)$$

With the help of the above proposed intra-/inter-prediction scheme, in Section III-B, we propose a fast decision scheme to terminate the quadtree structure determination of CTU decisions as early as possible.

### B. Proposed Fast Quadtree Structure Decision Scheme

To encode each CTU, a full quadtree consisting of different-sized CUs is initially established, and the smallest RD cost of each partitioned CU can be determined by the proposed bitrate-saving intra-/inter-prediction scheme. Then a bottom-up tree merging process is used to decide the quadtree structure with the minimal RD cost. Although performing the top-down quadtree partition process and the bottom-up tree merging process could determine the optimal quadtree structure, it suffers from high computational cost. We now propose the fast quadtree structure determination scheme for depth map coding.

Before partitioning $C^i_{k\times k}$ for $k \in \{64, 32, 16\}$ into four child CUs $C^{4i}_{k/2\times k/2}$, $C^{4i+1}_{k/2\times k/2}$, $C^{4i+2}_{k/2\times k/2}$, and $C^{4i+3}_{k/2\times k/2}$, the proposed quadtree structure decision scheme will check whether a further partition is required. Using the lookup table of $R$, each depth value $g_{m,n}(x, y)$ in $C^i_{k\times k}$ can access its allowable range $R$ $(=[\ell(g_{m,n}(x, y)), u(g_{m,n}(x, y)])$ in constant time. After encoding $C^i_{k\times k}$, we further decode $C^i_{k\times k}$ to obtain the reconstructed CU $C^{r,i}_{k\times k}$ and for each depth value $g^r_{m,n}(x, y)$ in $C^{r,i}_{k\times k}$, we check whether the condition $g^r_{m,n}(x, y) \in R$ is held or not. If all depth pixels in $C^{r,i}_{k\times k}$ satisfy this condition, we stop partitioning $C^i_{k\times k}$ further, achieving the time-saving effect.

Although the proposed quadtree structure determination scheme is fast and synthesis error free, it may lead to a slight bitrate performance degradation. In our experiments, we find that when compared with the optimal quadtree structure determined by the 3D-HTM, the early termination strategy in the proposed scheme may skip some CU partitions, resulting in many zero matrices. Because one zero matrix is encoded merely using an end of block (EOB) symbol, skipping these CU partitions increases the bitrate required for encoding the CTU.

To remedy the above bitrate performance degradation, an improved strategy is presented as follows. Let $\hat{C}^i_{k\times k}$ denote the prediction result of $C^i_{k\times k}$, the prediction error is denoted by $E^i_{k\times k} = C^i_{k\times k} - \hat{C}^i_{k\times k}$, $F(E^i_{k\times k})$ denotes the coefficient matrix after performing the discrete cosine transform or discrete sine transform on $E^i_{k\times k}$, and $Q(F(E^i_{k\times k}))$ denotes the quantized $F(E^i_{k\times k})$. If $Q(F(E^i_{k\times k})) = 0_{k\times k}$, where $0_{k\times k}$ denotes a $k \times k$ zero matrix, is held, we stop splitting the CU further. Because a depth map usually has a large portion of homogeneous regions, many quantized frequency coefficient matrices tend to be zero matrices. Using the proposed improved strategy, the above bitrate performance degradation problem of the proposed quadtree structure determination scheme can be remedied.

We now discuss the impact of the proposed fast quadtree structure determination scheme on the bitrate and quality. Let $T^*$ and $T$ denote the quadtree structures determined by the 3D-HTM and the proposed scheme, respectively. Assuming that $T^* \neq T$, since $T^*$ is the RD optimized quadtree structure, we have $\mathrm{RD}_{T^*}(\mathrm{CT}) \leq \mathrm{RD}_T(\mathrm{CT})$ where $\mathrm{RD}_{T^*}(\mathrm{CT})$ and $\mathrm{RD}_T(\mathrm{CT})$ denote, respectively, the RD costs using $T^*$ and $T$ to partition and encode the CTU $CT$. It is known that $T^*$ can result in smaller RD cost. Considering the bitrate term in the above two RD costs, denoted by $R_{T^*}(\mathrm{CT})$ and $R_T(\mathrm{CT})$, we now explain why the condition $R_{T^*}(\mathrm{CT}) \geq R_T(\mathrm{CT})$ is held. First, because the height of $T$ is smaller than that of $T^*$, we can use less bits to encode $T$ than to encode $T^*$. Further, the proposed scheme early terminates the partition process with zero matrices which can minimize the bitrate for encoding the partitioned CUs by EOBs. Considering the distortion term in the RD costs corresponding to $T^*$ and $T$ denoted by $D_{T^*}(\mathrm{CT})$ and $D_T(\mathrm{CT})$, $\mathrm{RD}_{T^*}(\mathrm{CT}) \leq \mathrm{RD}_T(\mathrm{CT})$ and $R_{T^*}(\mathrm{CT}) \geq R_T(\mathrm{CT})$ lead to $D_{T^*}(\mathrm{CT}) \leq D_T(\mathrm{CT})$. In fact, the 3D-HTM calculates the distortion term by combining the distortions of the quantized depth values and the rendered virtual view. Although the proposed scheme incurs larger distortion in quantized depth values, the synthesis error-free property can minimize the distortion in rendered virtual view. From the above discussion, in addition to fast determination of quadtree structure, the proposed scheme may also achieve bitrate-saving effect without causing synthesis errors.

As mentioned previously, the allowable ranges for all 256 possible depth values, $0, 1, 2, \ldots, 255$, have been stored in advance in the lookup table $\mathrm{LUT}_R(i)$ for $0 \leq i \leq 255$. We now present the whole two-stage CTU decision scheme which consists of the CTU partitioning stage and the CTU decision stage. In the CTU partitioning stage, the two procedures, namely, the Partition_CTU and Partition_CU, are used to construct the optimal quadtree structure from the input CTU. Here, the CTU $CT$ and the quadtree $T$ are used as the inputs of the CTU quadtree decision stage.

PROCEDURE Partition_CTU(CT, $\mathrm{LUT}_R$)
1 $T \leftarrow \{\}$.
2 Construct $C^0_{64\times 64}$ from CT.
3 $T \leftarrow T \cup \mathrm{CU\_Partition}(C^0_{64\times 64}, \mathrm{LUT}_R, T)$.
4 **return** $T$.

In line 1 of the above procedure, the quadtree $T$ is initialized to an empty set. In lines 2 and 3, the $64\times 64$ CU is constructed and the procedure Partition_CU is called to construct the quadtree $T$, respectively.

PROCEDURE Partition_CU($C^i_{k\times k}$, $\mathrm{LUT}_R$, $T$)
1 Determine the best prediction mode of $C^i_{k\times k}$ according to $\mathrm{RD}(C^i_{k\times k})$.
2 **if** $k > 8$ **then**
3 Encode $C^i_{k\times k}$ by the determined best prediction mode and decode it to obtain the reconstructed depth block $C^{r,i}_{k\times k}$.
4 **if** $g^r_{m,n}(x, y) \in \mathrm{LUT}_R(g_{m,n}(x, y))$ for all pixels in $C^{r,i}_{k\times k}$ and $Q(F(E^i_{k\times k})) = 0_{k\times k}$ **then**
5 **return** $C^i_{k\times k}$.
6 **else**
7 Partition $C^i_{k\times k}$ into $C^{4i}_{k/2\times k/2}$, $C^{4i+1}_{k/2\times k/2}$, $C^{4i+2}_{k/2\times k/2}$, and $C^{4i+3}_{k/2\times k/2}$.
8 $T \leftarrow T \cup \mathrm{Partition\_CU}(C^{4i}_{k/2\times k/2}, \mathrm{LUT}_R, T)$.
9 $T \leftarrow T \cup \mathrm{Partition\_CU}(C^{4i+1}_{k/2\times k/2}, \mathrm{LUT}_R, T)$.
10 $T \leftarrow T \cup \mathrm{Partition\_CU}(C^{4i+2}_{k/2\times k/2}, \mathrm{LUT}_R, T)$.
11 $T \leftarrow T \cup \mathrm{Partition\_CU}(C^{4i+3}_{k/2\times k/2}, \mathrm{LUT}_R, T)$.
12 **end-if**
13 **else**
14 **return** $C^i_{k\times k}$.
15 **end-if**

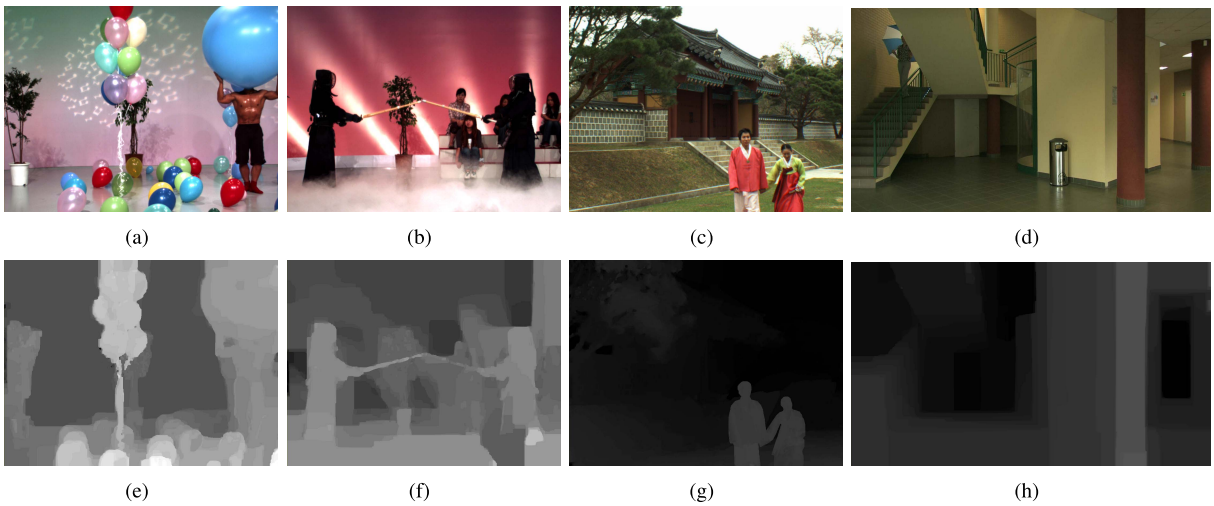



Fig. 4. Color and depth maps of the four test sequences. (a)–(d) First color maps of the first view of the two-view sequences extracted from the *Balloons, Kendo, Lovebird*, and *Poznan Hall* sequences, respectively. (e)–(h) Depth maps for (a)–(d), respectively.

In lines 3–12 of the above procedure, we deal with the partition process for $64 \times 64$ CU, $32 \times 32$ CU, or $16 \times 16$ CU. After encoding and decoding the CU in line 3, the two early termination conditions for CU partitioning are checked in line 4 and if one of them is not held, the four child CUs are further partitioned (see lines 7–11). After constructing the quadtree $T$, the CTU decision stage is used to determine the optimal quadtree structure, and it is realized by the procedure Optimal_Quadtree whose main body is the tree merging process.

PROCEDURE Optimal_Quadtree($T$)
**1** $k \leftarrow 16$.
**2** **while** $k \leq 64$ **do**
**3** **for** each $C^i_{k\times k}$ in $T$ **do**
**4** $C_\ell \leftarrow$ Descendant_Leaf($T$, $C^k_{k\times k}$).
**5** **if** RD($C^i_{k\times k}$) $< \sum_{C\in C_\ell}$ RD($C$) **then**
**6** $C_d \leftarrow$ Descendant($T$, $C^k_{k\times k}$).
**7** $T \leftarrow T \backslash C_d$.
**8** **end-if**
**9** **end-for**
**10** $k \leftarrow k \times 2$.
**11** **end-while**
**12** **return** $T$

According to the bottom-up tree merging approach, as shown in lines 2–11, the CUs in $T$ are examined in the order of the $16 \times 16$ CUs followed by the $32 \times 32$ CUs, and then the $64 \times 64$ CU. For $C^i_{k\times k}$, the procedure Descendant_Leaf in line 4 is used to take the descendants at the leaves of $T$, say $C_\ell$. In line 5, we compare the RD cost of $C^i_{k\times k}$ with the sum of RD costs of CUs in $C_\ell$. When $C^i_{k\times k}$ has a lower RD cost, the tree merging process in lines 6 and 7 is used to remove all descendant CUs from $T$. In line 7, $\backslash$ denotes the removal operator. Continue the above process to remove infeasible CUs until the resultant merged quadtree $T$ with the minimal RD cost has been constructed.

## IV. Experimental Results

To demonstrate the effectiveness of the proposed coding method for depth videos, four MVD video sequences, namely, *Balloons, Kendo, Lovebird,* and *Poznan Hall*, were used as the test instances for performance comparison between the proposed improved coding method and the traditional method in 3D-HTM. Furthermore, the depth map coding method proposed by Lee *et al.* [15] was also considered for comparison mainly because it aims to simultaneously reduce the bitrate and save the encoding time, which is the same as the goal of the proposed method, but via adopting a different strategy. Each color/depth map in the first three video sequences, *Balloons, Kendo,* and *Lovebird,* is $1024 \times 768$ in size, and each map in the *Poznan Hall* sequence is of size $1920 \times 1088$. Among the four test MVD video sequences, each sequence is composed of three to twelve views, and each view has 200–300 color and depth maps. Since fully using the above-mentioned four MVD video sequences in our experiments requires huge memory space and encoding time, in order to make our experiments feasible, we extracted 100 color and depth maps from the first and third views in the *Balloons* and *Kendo* sequences, the fourth and sixth views in the *Lovebird* sequence, and the fifth and seventh views in the *Poznan Hall* sequence to form the four experimental two-view sequences. The first color and depth maps of the first view are shown in Fig. 4, as the input of 3D-HTM. The second view in the *Balloons* and *Kendo* sequences, the fifth view in the *Lovebird* sequence, and the sixth view in the *Poznan Hall* sequence are used as the ground truth of the rendered virtual view. All experiments were performed on a computer with an Intel i7-3770 CPU 3.4 GHz and 4-GB RAM. The program developing environment was Visual Studio C++ 2008 implemented on platform 3D-HTM 9.0 [33] with the Microsoft Windows 7 operating system. The virtual views are rendered using the view synthesis reference software (VSRS) [32] and the rendering method in VSRS is also used in 3D-HTM. The group of pictures (GOP) size used in 3D-HTM is set to 8 and its structure is shown in Fig. 5.

The bitrate of the compressed depth videos, the peak signal-to-noise ratio (PSNR) of the rendered virtual views, and the encoding time for compressing the depth videos are the three main performance measures for comparison. With the frame

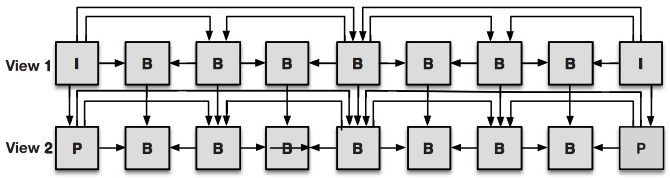



Fig. 5. GOP structure used in our experiments.

rate of $F$ frames per second, the bitrate of the compressed depth videos in the MVD sequence with $M$ real views is defined as

$$\mathrm{BR} = \sum_{i=0}^{M-1} \frac{B_i}{N} \times F \tag{13}$$

where $F$ denotes the frame rate per second, $N$ denotes the number of depth maps in each depth video, and $B_i$ represents the total number of bits used for compressing the depth video for the $i$th view. A low bitrate implies less requirement for storage and network bandwidth, and is preferred. Let $\mathrm{BR}_{\mathrm{Proposed}}$, $\mathrm{BR}_{\mathrm{Lee}}$, and $\mathrm{BR}_{\mathrm{3D\text{-}HTM}}$ denote the bitrate required in the compressed depth video using the proposed improved coding method, the coding method by Lee *et al.* [15], and the traditional method in 3D-HTM, respectively. Taking the traditional method in 3D-HTM as the comparison basis, the bitrate improvement ratio of the coding method $X$ is calculated by

$$\eta_{\mathrm{BR}} = \frac{\mathrm{BR}_{\mathrm{3D\text{-}HTM}} - \mathrm{BR}_X}{\mathrm{BR}_{\mathrm{3D\text{-}HTM}}}. \tag{14}$$

The PSNR of the rendered virtual view is expressed as

$$\mathrm{PSNR} = \frac{1}{N} \sum_{i=0}^{N-1} 10 \log_{10} \frac{255^2}{\mathrm{MSE}_i} \tag{15}$$

with

$$\mathrm{MSE}_i = \frac{1}{\mathrm{WH}} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} \left[ f_i^r(x, y) - f_i^g(x, y) \right]^2 \tag{16}$$

where $f_i^r(x, y)$ and $f_i^g(x, y)$ denote the color values of the $i$th color maps in the rendered virtual view and the corresponding ground truth, respectively. Here, the rendered virtual view is synthesized using the compressed color video rather than the original one. Let $T_{\mathrm{Proposed}}$, $T_{\mathrm{Lee}}$, and $T_{\mathrm{3D\text{-}HTM}}$ denote the encoding time for compressing the depth video using the proposed improved coding method, the coding method by Lee *et al.*, and the traditional method in 3D-HTM, respectively. The encoding-time improvement ratio of the coding method $X$ over the traditional method in 3D-HTM is calculated by

$$\eta_T = \frac{T_{\mathrm{3D\text{-}HTM}} - T_X}{T_{\mathrm{3D\text{-}HTM}}}. \tag{17}$$

The empirical results, against different values of quantization parameter (QP), regarding the bitrate of the compressed depth video, the PSNR of the rendered virtual view, and the encoding time of the depth video are listed in Tables I–III, respectively. Table I gives the bitrates of the compressed depth videos and the relative bitrate improvement ratios of the proposed method and Lee *et al.*'s method over the traditional method in 3D-HTM. Because the D-NOSE-based allowable ranges are somewhat wide in the first three test video sequences, as shown in Table I, the corresponding bitrate reduction effects of the proposed method are rather clear. Considering the four test video sequences, the proposed coding method has a 16.3% average bitrate improvement when compared with the traditional one in 3D-HTM. Meanwhile, Lee *et al.*'s method performs worse than the proposed method, especially in the *Lovebird* video

TABLE I

BITRATE COMPARISON IN KILOBITS PER SECOND OF THE PROPOSED CODING METHOD, LEE *et al.*'S CODING METHOD, AND THE TRADITIONAL METHOD IN 3D-HTM

| | Sequence | Bitrate ($\eta_{BR}$) | | |
|---|---|---|---|---|
| | | 3D-HTM | Lee *et al.*'s | Proposed |
| QP = 18 | Balloons | 3115 | 2834 ( 9.0%) | 2543 (18.4%) |
| | Kendo | 2175 | 2001 ( 8.0%) | 1766 (18.3%) |
| | Lovebird | 1495 | 1471 ( 1.6%) | 1210 (19.1%) |
| | Poznan | 1294 | 1216 ( 6.1%) | 1192 ( 7.9%) |
| | Ave. | 2020 | 1880 ( 6.9%) | 1678 (16.9%) |
| QP = 22 | Balloons | 1636 | 1492 ( 8.8%) | 1353 (17.3%) |
| | Kendo | 1198 | 1086 ( 9.3%) | 971 (18.9%) |
| | Lovebird | 973 | 967 ( 0.5%) | 790 (18.8%) |
| | Poznan | 790 | 739 ( 7.3%) | 710 (10.1%) |
| | Ave. | 1149 | 1070 ( 6.9%) | 956 (16.8%) |
| QP = 26 | Balloons | 846 | 796 ( 5.9%) | 723 (14.5%) |
| | Kendo | 636 | 588 ( 7.5%) | 528 (15.4%) |
| | Lovebird | 644 | 645 (-0.2%) | 516 (19.9%) |
| | Poznan | 477 | 451 ( 5.5%) | 434 ( 9.0%) |
| | Ave. | 651 | 620 ( 4.7%) | 550 (15.5%) |
| QP = 30 | Balloons | 478 | 467 ( 2.4%) | 415 (13.2%) |
| | Kendo | 362 | 339 ( 6.3%) | 307 (15.2%) |
| | Lovebird | 420 | 427 (-1.8%) | 347 (17.4%) |
| | Poznan | 309 | 303 ( 2.1%) | 291 ( 5.8%) |
| | Ave. | 392 | 384 ( 2.1%) | 340 (13.3%) |
| Ave. | | 1053 | 989 ( 6.1%) | 881 (16.3%) |

TABLE II

PSNR COMPARISON IN DECIBELS AMONG THE PROPOSED CODING METHOD, LEE *et al.*'S CODING METHOD, AND THE TRADITIONAL METHOD IN 3D-HTM

| | Sequence | PSNR | | |
|---|---|---|---|---|
| | | 3D-HTM | Lee *et al.*'s | Proposed |
| QP = 18 | Balloons | 35.85 | 35.83 | 35.71 |
| | Kendo | 36.69 | 36.63 | 36.65 |
| | Lovebird | 31.84 | 31.84 | 31.90 |
| | Poznan | 35.97 | 35.97 | 35.94 |
| | Ave. | 35.09 | 35.06 | 35.05 |
| QP = 22 | Balloons | 35.76 | 35.74 | 35.61 |
| | Kendo | 36.62 | 36.53 | 36.58 |
| | Lovebird | 31.84 | 31.84 | 31.89 |
| | Poznan | 35.94 | 35.93 | 35.92 |
| | Ave. | 35.04 | 35.01 | 35.00 |
| QP = 26 | Balloons | 35.56 | 35.54 | 35.44 |
| | Kendo | 36.42 | 36.31 | 36.38 |
| | Lovebird | 31.82 | 31.83 | 31.89 |
| | Poznan | 35.93 | 35.91 | 35.92 |
| | Ave. | 34.93 | 34.90 | 34.91 |
| QP = 30 | Balloons | 35.21 | 35.20 | 35.09 |
| | Kendo | 36.06 | 35.95 | 36.01 |
| | Lovebird | 31.74 | 31.73 | 31.79 |
| | Poznan | 35.87 | 35.86 | 35.86 |
| | Ave. | 34.72 | 34.68 | 34.69 |
| Ave. | | 34.95 | 34.91 | 34.91 |

TABLE III
ENCODING-TIME COMPARISON IN SECONDS OF THE PROPOSED CODING METHOD, LEE *et al.*'s CODING METHOD, AND THE TRADITIONAL METHOD IN 3D-HTM

| | Sequence | Encoding Time ($\eta_T$) | | |
|---|---|---|---|---|
| | | 3D-HTM | Lee *et al.*'s | Proposed |
| QP = 18 | Balloons | 2583 | 2362 ( 8.5%) | 2142 (17.1%) |
| | Kendo | 2539 | 2342 ( 7.7%) | 2187 (13.9%) |
| | Lovebird | 1633 | 1568 ( 4.0%) | 1346 (17.6%) |
| | Poznan | 5353 | 4812 (10.1%) | 4948 ( 7.6%) |
| | Ave. | 3027 | 2771 ( 8.4%) | 2656 (12.3%) |
| QP = 22 | Balloons | 1958 | 1830 ( 6.5%) | 1637 (16.4%) |
| | Kendo | 2026 | 1892 ( 6.6%) | 1731 (14.6%) |
| | Lovebird | 1431 | 1389 ( 2.9%) | 1153 (19.4%) |
| | Poznan | 4003 | 3670 ( 8.3%) | 3582 (10.5%) |
| | Ave. | 2355 | 2195 ( 6.8%) | 2026 (14.0%) |
| QP = 26 | Balloons | 1566 | 1483 ( 5.3%) | 1304 (16.7%) |
| | Kendo | 1697 | 1589 ( 6.4%) | 1468 (13.5%) |
| | Lovebird | 1290 | 1242 ( 3.7%) | 997 (22.7%) |
| | Poznan | 3271 | 3013 ( 7.9%) | 2919 (10.8%) |
| | Ave. | 1956 | 1832 ( 6.3%) | 1672 (14.5%) |
| QP = 30 | Balloons | 1305 | 1244 ( 4.7%) | 1102 (15.6%) |
| | Kendo | 1464 | 1371 ( 6.4%) | 1286 (12.2%) |
| | Lovebird | 1186 | 1121 ( 5.5%) | 914 (22.9%) |
| | Poznan | 3079 | 2807 ( 8.8%) | 2651 (13.9%) |
| | Ave. | 1759 | 1636 ( 7.0%) | 1488 (15.4%) |
| Ave. | | 2274 | 2109 ( 7.3%) | 1960 (13.8%) |

sequence, and only delivers a 6.1% bitrate improvement on average when compared with the traditional one in 3D-HTM, indicating that our proposed method is more robust and possesses clear and satisfactory bitrate-saving merit.

Table II gives the PSNR values of the four rendered virtual views using the three concerned methods. From Table II, the average PSNR degradation of the proposed method and Lee *et al.*'s method over the traditional one in 3D-HTM is only 0.04 dB. In fact, when viewing the 3-D synthesized scenes, such a quality degradation hardly causes any visual difference. This very small PSNR deviation in the proposed method occurs because, for the proposed intra-/inter-prediction scheme, a very small portion of the decompressed depth values may fall outside the allowable range after quantization at the encoder side and dequantization at the decoder side. The reason for the PSNR degradation of Lee *et al.*'s method is mainly because skipping the regular encoding of depth blocks and inheriting the encoding results from the similar colocated depth blocks in the temporal or inter-view reference frames cannot necessarily guarantee that no synthesis errors will happen.

According to the average bitrate shown in Table I and the average PSNR shown in Table II, we adopt the RD curve, which depicts the average PSNR against the average bitrate for the four different QPs, to illustrate the overall performance of the proposed method, Lee *et al.*'s method, and the traditional method in 3D-HTM. From the resultant RD curve depicted in Fig. 6, the proposed method for compressing the depth videos of the MVD sequences in 3D-HTM outperforms both Lee *et al.*'s method and the traditional one in 3D-HTM in an average sense.

Table III gives the encoding time required for compressing the depth videos and the relative encoding-time improvement ratios of the proposed method and Lee *et al.*'s method over

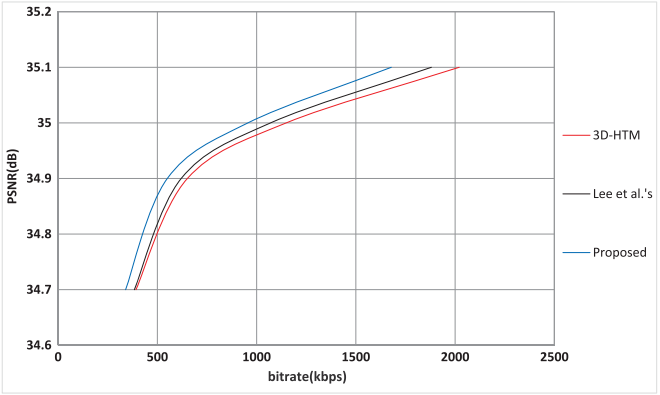



Fig. 6. RD curves of average PSNR against average bitrate over the four concerned MVD video sequences for the proposed method, Lee *et al.*'s method and the traditional method in 3D-HTM.

the traditional method in 3D-HTM. Since a lookup table of the D-NOSE-based allowable ranges is built up in advance so as to speed up the two proposed schemes and the proposed quadtree structure determination scheme has the merit of reducing the computational complexity, the proposed method demonstrates less encoding time required for compressing the depth videos and has a 13.8% encoding-time improvement on average when compared with the traditional one in 3D-HTM. Especially, compressing the first three test video sequences by the proposed method can result in a higher encoding-time reduction effect mainly because the D-NOSE-based allowable ranges are somewhat wide in the corresponding depth maps. In contrast, Lee *et al.*'s method only delivers a 7.3% average encoding-time improvement when compared with the traditional one in 3D-HTM, implying that the proposed method, which has 13.8% encoding-time improvement, is more effective for improving the encoding time of 3D-HTM.

Finally, we discuss the individual performance of the proposed first and second schemes, i.e., the proposed bitrate-saving prediction scheme and the proposed fast quadtree structure determination scheme. To evaluate the performance of the first scheme, we disabled the second scheme and used the quadtree partition process of 3D-HTM to determine the optimal quadtree partition of each CTU. For convenience, we denote the above scheme as Proposed-I. Similarly, to evaluate the performance of the second scheme, the first scheme was disabled and the intra/inter prediction of the 3D-HTM was used to encode each CU; this scheme is denoted as the Proposed-II. Table IV shows the performance comparisons of the average PSNR, bitrate, and encoding time for Proposed-I, Proposed-II, and the traditional one in 3D-HTM. From Table IV, both Proposed-I and Proposed-II can preserve the quality of the rendered virtual view. When compared with the traditional one in 3D-HTM, Proposed-I delivers a 9.2% average bitrate improvement and nearly a 0% encoding-time overhead, while Proposed-II provides not only a 16.2% average encoding-time improvement but also a 4.2% bitrate improvement. The bitrate improvement achieved by Proposed-II confirms the discussion about the bitrate-saving effect of the proposed fast quadtree structure determination scheme in Section III-B. Note that in Table IV, the sum of the

TABLE IV
PERFORMANCE COMPARISON OF THE AVERAGE PSNR, BITRATE, AND ENCODING-TIME PERFORMANCE OF PROPOSED-I, PROPOSED-II, AND THE TRADITIONAL ONE IN 3D-HTM

| | 3D-HTM | Proposed-I | Proposed-II |
|---|---|---|---|
| PSNR | 34.95 | 34.90 | 34.94 |
| Bitrate ($\eta_{BR}$) | 1053 | 956 (9.2%) | 1009 (4.2%) |
| Encoding Time ($\eta_T$) | 2274 | 2263 (0.5%) | 1906 (16.2%) |

average bitrate improvements of Proposed-I and Proposed-II is 13.4% (=9.2% + 4.2%), which is kind of different from the average bitrate improvement of the proposed whole method, i.e., 16.3%, as shown in Table I. The reason for this discrepancy in Tables I and IV is because disabling either the proposed first scheme or the proposed second scheme results in different quadtree structures from using both schemes together, such as the resultant RD cost and quadtree structure. Based on the same reason, the sum of average encoding-time improvement of Proposed-I and Proposed-II, i.e., 16.7% (=0.5% + 16.2%), is also kind of different from the average encoding-time improvement of the proposed whole method, i.e., 13.8%, as shown in Table III.

In summary, the experimental results shown in Tables I–IV confirm that for compressing depth videos of the MVD sequences in 3D-HTM, the proposed coding method has not only a clear bitrate-saving effect but also the merits of reducing the computational cost and preserving the quality of the rendered virtual view.

## V. CONCLUSION

Based on the color and depth videos of the MVV system with MVD format, the virtual views with arbitrary viewpoints can be synthesized using the DIBR technique. With the same quality of synthesized virtual views, compressing depth videos in a fast and bitrate-saving manner is crucial. We have presented the proposed bitrate-saving and low computational coding method for improving the depth video coding in 3D-HTM. The proposed coding method makes two main contributions.

1) The proposed bitrate-saving intra-/inter-prediction scheme can modify each depth value in the depth videos such that the modified depth value falls within the allowable range to minimize the intra- and inter-prediction errors, so that it has no synthesis errors.
2) The proposed fast optimal-CTU-quadtree decision scheme can terminate the quadtree-based partition process of CTU as early as possible.

Although a very small portion of the decompressed depth values may deviate from the allowable ranges after quantization, the average PSNR degradation of the rendered virtual view is only 0.04 dB empirically. On average, the proposed coding method for compressing depth videos has 16.3% bitrate and 13.8% encoding-time improvement ratios when compared with the traditional method in 3D-HTM. Consequently, the proposed coding method for compressing depth videos achieves better bitrate and encoding-time performance than Lee *et al.*'s method and the traditional method in 3D-HTM while preserving the quality of the rendered virtual view. Furthermore, since the proposed coding method still incurs some quality deviation due to the fact that a very small portion of the modified depth values may fall outside the allowable range after dequantization, further extension work is planned to take the quantization distortion into account so as to solve this problem.

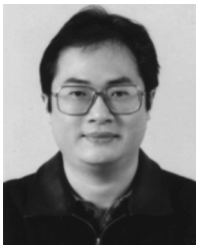
## ACKNOWLEDGMENT

The authors would like to thank C.-W. Yu and C. Harrington for their help in programming and proofreading, respectively.

## REFERENCES

[1] C. Fehn *et al.*, "An evolutionary and optimised approach on 3D-TV," in *Proc. Int. Broadcast Conf.*, Sep. 2002, pp. 357–365.
[2] T. Naemura, M. Kaneko, and H. Harashima, "Compression and representation of 3-D images," *IEICE Trans. Inf. Syst.*, vol. E82–D, no. 3, pp. 558–565, Mar. 1999.
[3] R.-S. Wang and Y. Wang, "Multiview video sequence analysis, compression, and virtual viewpoint synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 397–410, Apr. 2000.
[4] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, May 2004.
[5] Y.-L. Chang, C.-Y. Fang, L.-F. Ding, S. Y. Chen, and L.-G. Chen, "Depth map generation for 2D-to-3D conversion by short-term motion assisted color segmentation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1958–1961.
[6] Y.-C. Wang, C.-P. Tung, and P.-C. Chung, "Efficient disparity estimation using hierarchical bilateral disparity structure based graph cut algorithm with a foreground boundary refinement mechanism," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 5, pp. 784–801, May 2013.
[7] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC), Joint Video Team of ISO/IEC and ITU-T*, document JVT-G050, Mar. 2003.
[8] M. Maitre, Y. Shinagawa, and M. N. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 946–957, Jun. 2008.
[9] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bitrate allocation for 3D video-plus-depth coding," *EURASIP J. Appl. Signal Process.*, vol. 2009, pp. 1–13, Jan. 2009.
[10] G. Cernigliaro, M. Naccari, F. Jaureguizar, J. Cabrera, E. Pereira, and N. Garcia, "A new fast motion estimation and mode decision algorithm for H.264 depth maps encoding in free viewpoint TV," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1013–1016.
[11] Q. Zhang, P. An, Y. Zhang, L. Shen, and Z. Zhang, "Low complexity multiview video plus depth coding," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1857–1865, Nov. 2011.
[12] D. Temel, M. Aabed, M. Solh, and G. AlRegib, "Efficient streaming of stereoscopic depth-based 3D videos," *Proc. SPIE*, vol. 8666, pp. 86660I-1–86660I-10, Feb. 2013.
[13] S. Liu, P. Lai, D. Tian, C. Gomila, and C. W. Chen, "Sparse dyadic mode for depth map compression," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 3421–3424.
[14] L. Chen, M. M. Hannuksela, and H. Li, "Intra coding for depth maps using adaptive boundary location," in *Proc. Vis. Commun. Image Process.*, Nov. 2012, pp. 1–6.
[15] J. Y. Lee, H.-C. Wey, and D.-S. Park, "A fast and efficient multi-view depth image coding method based on temporal and inter-view correlations of texture images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 12, pp. 1859–1868, Dec. 2011.
[16] R. Li, D. Rusanovskyy, M. M. Hannuksela, and H. Li, "Joint view filtering for multiview depth map sequences," in *Proc. IEEE Int. Conf. Image Process.*, Sep./Oct. 2012, pp. 1329–1332.
[17] D. V. S. X. De Silva, E. Ekmekcioglu, W. A. C. Fernando, and S. T. Worrall, "Display dependent preprocessing of depth maps based on just noticeable depth difference modeling," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 2, pp. 335–351, Apr. 2011.
[18] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3-D video," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 747–750, Sep. 2009.

[19] E. Ekmekcioglu, M. Mrak, S. Worrall, and A. Kondoz, "Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering," in *Proc. 16th IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 733–736.
[20] M. O. Wildeboer, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Depth up-sampling for depth coding using view information," in *Proc. 3DTV Conf.*, May 2011, pp. 1–4.
[21] V.-A. Nguyen, D. Min, and M. N. Do, "Efficient techniques for depth video compression using weighted mode filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 189–202, Feb. 2013.
[22] P. Merkle *et al.*, "The effects of multiview depth video compression on multiview rendering," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, Jan. 2009.
[23] G. Cheung, A. Kubota, and A. Ortega, "Sparse representation of depth maps for efficient transform coding," in *Proc. Picture Coding Symp.*, Dec. 2010, pp. 298–301.
[24] Y. Zhao, C. Zhu, Z. Chen, and L. Yu, "Depth no-synthesis-error model for view synthesis in 3-D video," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2221–2228, Aug. 2011.
[25] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, and T. Wiegand, *Working Draft 4 of High-Efficiency Video Coding JCTVC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, document JCTVC-F803:WD4, Jul. 2011.
[26] K. Müller *et al.*, "3D High-Efficiency Video Coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.
[27] G. Tech, K. Wegner, Y. Chen, and S. Yea, *3D-HEVC Draft Text 2 Joint Collaborative Team 3D Video Coding Extension Development (JCT-3V)* document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Oct. 2013.
[28] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of High Efficiency Video Coding (HEVC)," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.
[29] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3d video," *Proc. SPIE*, vol. 7443, pp. 74430T-1–74430T-11, Sep. 2009.
[30] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, *Reference Softwares for Depth Estimation and View Synthesis ISO/IEC JTC1/SC29/WG11*, document M15377, Apr. 2008.
[31] M. Solh and G. AlRegib, "Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 5, pp. 495–504, Sep. 2012.
[32] M. Tanimoto, T. Fujii, and K. Suzuki, *View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS2.0)*, document ISO/IEC JTC 1/SC 29/WG 11 M16090, Feb. 2009.
[33] (Nov. 2013). *3D-HEVC Test Model (3D-HTM) Version 9.0*. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-9.0/



**Kuo-Liang Chung** (M'91–SM'01) received the B.S., M.S., and Ph.D. degrees from National Taiwan University, Taipei, Taiwan, in 1982, 1984, and 1990, respectively.

He was a Research Assistant with Institute of Information Science, Academia Sinica, Taipei, from 1986 to 1987, after two years of obligatory military service from 1984 to 1986. He was a Visiting Scholar with University of Washington, Seattle, WA, USA, in 1999. From 2003 to 2006, he was the Chair of the Department of Computer Science and Information Engineering with National Taiwan University of Science and Technology (NTUST), Taipei. He has been with the Department of Computer Science and Information Engineering, NTUST, since 1995, where he is currently a Professor. Since 2009, he has been a University Chair Professor with NTUST. His research interests include video compression, multiview video coding, data hiding, camera image processing, and shape analysis.

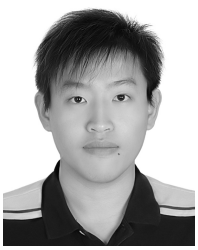
Dr. Chung is a fellow of the Institute of Engineering and Technology. He was a recipient of the Distinguished Research Award and the Distinguished Research Project Award from the National Science Council of Taiwan from 2004 to 2007 and 2009 to 2012, the best paper award from the Image Processing and Pattern Recognition Society of Taiwan in 2007, and the Distinguished Teaching Award of NTUST in 2009. He was a Managing Editor of *Journal of Chinese Institute of Engineers* from 1996 to 1998. In 2000, he was the Program Co-Chair of the Conference on Computer Vision, Graphics, and Image Processing, Taiwan. He is an Associate Editor of *Journal of Visual Communication and Image Representation*.

**Yong-Huai Huang** (M'11) received the B.S. degree in information management from Aletheia University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University of Science and Technology, Taipei.

He is an Associate Professor with the Department of Electronic Engineering, Jinwen University of Science and Technology, New Taipei City, Taiwan. His research interests include image processing and compression, and algorithms.

**Chien-Hsiung Lin** received the B.S., M.S., and Ph.D. degrees in information management from National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, in 2003, 2005, and 2011, respectively.

He is a Post-Doctoral Researcher with the Department of Computer Science and Information Engineering, NTUST. His research interests include digital camera image processing, video compression, multiview video coding, data hiding, statistical analysis, and stochastic simulation.



**Jian-Ping Fang** received the B.S. degree in electronics engineering from National Yunlin University of Science and Technology, Yunlin, Taiwan, and the M.S. degree in computer science and information engineering from National Taiwan University of Science and Technology, Taipei, Taiwan.

His research interests include image processing and compression, and algorithms.