Texture Plus Depth Video Coding Using Camera Global Motion Information

Fei Cheng, Tammam Tillo, *Senior Member, IEEE*, Jimin Xiao, *Member, IEEE*, and Byeungwoo Jeon, *Senior Member, IEEE*

***Abstract*—In video coding, traditional motion estimation methods work well for videos with camera translational motion, but their efficiency drops for other motions, such as rotational and dolly motions. In this paper, a motion-information-based three-dimensional (3D) video coding method is proposed for texture plus depth 3D video. The synchronized global motion information of the camera is obtained to assist the encoder improve its rate-distortion performance by projecting the temporal neighboring texture and depth frames into the position of the current frame, using the depth and camera motion information. Then, the projected frames are added into the reference buffer list as virtual reference frames. As these virtual reference frames could be more similar to the current to-be-encoded frame than the conventional reference frames, the required bits to represent the residual will be reduced. The experimental results demonstrate that the proposed scheme enhances the coding performance for all camera motion types and for various scene settings and resolutions using H.264 and HEVC standards, respectively. With the computer graphic sequences, for H.264, the average gain of texture and depth coding are up to 2 dB and 1 dB, respectively. For HEVC and HD resolution sequences, the gain of texture coding reaches 0.4 dB. For realistic sequences, up to 0.5 dB gain (H.264) is achieved for the texture video, while up to 0.7 dB gain is achieved for the depth sequences.**

***Index Terms*—Three-dimensional (3D) video coding, global motion, H.264, HEVC, HD, texture plus depth, temporal projection, virtual reference frame.**

I. Introduction

MANY digital media and mobile manufacturers are investing substantially in 3D technologies, including glassless 3D screen, virtual reality (VR), and 3D cameras. Among such technologies, the autostereoscopic 3D display [1] enables viewing 3D content from different angles without the use of special headgear or glasses. Meanwhile, stereo cameras and depth sensors are employed for mobile devices and home entertainment systems. For example, some mobile phones have two cameras for image enhancement and depth-of-field, while depth sensors are used in Microsoft Kinect and Intel RealSense [2] for 3D video capturing and object detection. With the multiple viewpoints and depth data of virtual reality and 3D videos, the required bit rate increases significantly.

To reduce the redundancy between different viewpoints of a 3D video, besides the commonly used conventional temporal prediction, inter-view prediction [3] is also exploited in the Multi-view Video Coding (MVC) [4] extension of the Advanced Video Coding standard [5]. Though MVC has enormously improved the compression performance of multi-view video, it still requires large bit rate proportional to the number of views [4]. Furthermore, conventional multi-view camera systems need to be set accurately, which puts constraints on the post-processing stage and consequently limits potential applications.

A depth map, which represents the distance from the objects in the scene to the capturing camera, together with its aligned texture, are exploited to describe 3D objects and scenes. Here the Multi-view Video plus Depth (MVD) format is a promising way to represent 3D video content, and recent extensions supporting this format have been introduced [6], [7], [8]. With the MVD format, only a small number of texture views associated with their depth views are required to represent 3D video, and the amount of bitrate allocated to texture and depth views could be properly tuned [9], [10]. At the decoder or display side, Depth-Image-Based Rendering (DIBR) [11], [12] is used to project additional viewpoint video.

Since the texture and its corresponding depth map describe the content and distance of the same scene respectively. Their correlation could be exploited by an encoder to reduce the redundancy. In [13], a motion information sharing mode for efficient depth coding is proposed to enhance the rate-distortion (RD) performance. Whereas in [14], this objective has been achieved by taking into account the Motion Vector (MV) of texture, besides RD performance enhancement, it reduces the Motion Estimation (ME) time for depth map encoding. In this paper, the addition of z direction motion estimation for depth map coding is also been proposed. However, only the information from texture is used to assist the depth map encoding, and the coding performance of texture is not improved.

In [15], View Synthesis Prediction (VSP) is incorporated to improve the prediction in MVC. In this work, the spatial

Manuscript received November 16, 2016; revised February 27, 2017; accepted April 18, 2017. Date of publication May 3, 2017; date of current version October 13, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61210006, Grant 60972085, and Grant 61501379, in part by the Jiangsu Science and Technology Programme under BK20150375, and in part by the MSIP, Korea, under the GITRC support program (IITP-2017-2015-0-00742) supervised by the IITP. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Balakrishnan Prabhakaran.(*Corresponding author: Fei Cheng.*)

F. Cheng, T. Tillo, and J. Xiao are with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: fei.cheng@xjtlu.edu.cn; tammam.tillo@xjtlu.edu.cn; jimin.xiao@xjtlu.edu.cn).

B. Jeon is with the School of Electronic and Electrical Engineering, Sungkyunkwan University, Seoul 03063, South Korea (e-mail: bjeon@skku.edu).

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/TMM.2017.2700622

synthesis views are exploited as reference frames in order to reduce the difference between the current encoded view and reference frames. In [16], some optimized VSP designs are adopted in 3D extensions of H.264/AVC and High Efficiency Video Coding (HEVC). In [17], a low-complexity adaptive view synthesis optimization scheme in the HEVC-based 3D video coding standard is proposed. In VSP, spatial neighboring views are used to generate virtual frames, which are used as reference frames to encode the current view, whereas in our scheme, we use temporal neighboring frames to generate virtual frames, which are used as reference frames to reduce the redundancy caused by camera movement. To fulfill this objective, camera global motion information is exploited. Note here both *view* and *frame* include texture and depth.

In many video capturing scenarios, the camera is not static, so the global motion generally leads to a long ME time, or even to ME "failure", which increases the percentage of Intra-block coding. A great deal of research have been devoted to the use of global motion, but these works only rely on estimating global motion information from texture data [18], [19]. Here it is worth mentioning that a considerable amount of time is needed to determine the global motion from texture, furthermore, this process is prone to errors.

With the advance of sensor technology, and the widespread diffusion of these sensors in video capturing platforms, the physical motion of a camera can be obtained directly from some of these sensors. It is worth highlighting that the accelerometer and the gyroscope are widely integrated into various kinds of smart hand-held devices, such as smart phones, tablet PCs and some cameras. In [20], accelerometers are used together with a digital compass to obtain camera motion to assist video coding. But, as depth information is not used, the motion of objects with different depths cannot be estimated and compensated for properly. In [21], a physical motion information based depth map sequence coding, using frame skipping strategy, was proposed. Some of the insignificant depth frames which can be projected from neighboring frames are skipped at the encoder side, and then they are projected again from corresponding frames at the decode side. Even so, this method does not achieve effective performance for scenes with many moving objects and it cannot be used for texture sequences.

In this paper, we propose a novel Motion information based Texture plus Depth map 3D video Coding (MTDC) scheme. The synchronously sampled global motion information of a camera and its depth map is exploited to improve the coding performance of both texture and depth sequences. The image projection technique and depth re-quantization method are used in the preprocessing stage to reduce the redundancy between temporal neighboring frames. In the video coding stage, the temporal projected virtual frames are exploited as reference frames by using a reference buffer reallocation strategy.

This paper provides multiple contributions to the field. Firstly, the camera global motion is measured using motion sensors. Therefore, it is less time consuming and more reliable than the conventional global motion estimation method.

Secondly, the depth information is used in a novel way to improve the coding performance of both the texture and depth

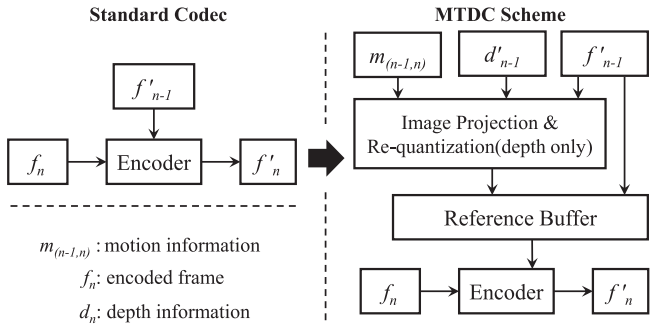

Fig. 1. Different from the traditional codec, the virtual frame obtained from the previous frame is exploited for encoding the current frame in MTDC scheme.

frames, which extends the function of the depth map. In fact, conventionally the depth map is only used by DIBR in the 3D video coding method. In this paper, the depth map is exploited to project one or more temporal neighboring texture and depth frames to the position of the current frame, which can be more similar to the current frame to enhance coding efficiency.

Therefore, the virtual projected image is derived from its temporal neighbor (in other words the temporal domain) instead of its neighboring views (which we will refer to it as the spatial domain neighbor), which is a key contribution of this paper. Thus, the view transformation between a neighboring frame and the current frame can be obtained from the camera global motion and camera parameters (such as view angle, zoom-level and resolution).

Thirdly, the projected virtual frames are added in the reference buffer before encoding the current frame as shown in Fig. 1. Since the added reference frame could be more similar to the current frame, it enhances motion estimation and reduces residual energy. Hence, the coding performance is improved. Note that a new reference frame is added into the reference buffer instead of replacing an existing reference frame. Therefore, even if the depth and motion information are not very accurate, the proposed scheme could still guarantee at least the original performance. Consequently, this simple but efficient modification makes the proposed method suitable for various hybrid video coding schemes, which represents the third contribution of this paper.

In order to test the proposed scheme, different Computer Graphics (CG) sequences in addition to real scene sequences were used. The CG sequences were generated with different scenes, different camera motions and different video resolutions. The proposed method was tested using the H.264/AVC and HEVC standards, respectively. The experimental results using CG sequences demonstrate that the proposed scheme can improve the coding performance of texture sequences and depth sequences in comparison to the original encoder. For H.264 standard, the average PSNR gain of both texture and depth sequences are 2 dB and 1 dB, respectively, while for HEVC standard, the PSNR gain of High Definition (HD, 1920 $\times$ 1080) sequences achieves around 0.4 dB. On the other hand, a hardware prototype has been developed to acquire texture plus depth sequences and their corresponding camera motions to test the proposed approach with real scenes. The results obtained for these sequences

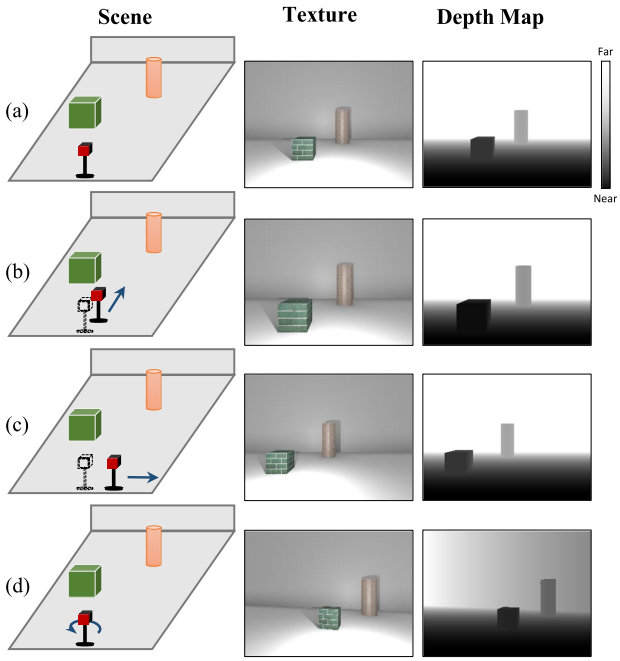

Fig. 2. Impact of different types of camera global motion on texture frame and depth frame: (a) initial frame, (b) dolly motion, (c) tracking motion, and (d) panning motion.

demonstrates that the proposed scheme can improve the coding performance in comparison to H.264/AVC, with PSNR gains of up to 0.5 dB for texture videos and of 0.7 dB for depth videos.

The proposed method is not limited to H.264/AVC or HEVC. In fact, it could be used with H.264/AVC based ATM [22] and HEVC based HTM [23]. Given the potential benefit of using depth data in many areas such as panoramas and VR, it is reasonable to assume that depth data will be more commonly used in future applications, this consequently will benefit texture video coding through the use of the proposed approach.

Compared to our preliminary work [24], in this paper, the camera motion based coding scheme is extended from the texture sequence to depth map sequence, meanwhile, there are more extensive evaluations including high definition sequences, CG sequences and HEVC standard. The rest of this paper is organized as follows. In Section II, the details of the proposed scheme are presented. After this, the experimental methods of the proposed scheme and its test results are presented in Section III. Finally, Section V concludes this work.

II. Proposed Method and Background Explanation

In many video capturing scenarios, the camera is not static and the image content changes with the camera's global motion. According to the imaging principle, the impact of the camera motion on texture and depth images can be pictorially presented as Fig. 2. In this example, a green cube and an orange cylinder are captured by a camera.

Dolly: With a dolly motion, the camera moves forwards, consequently, the sizes of the two objects get scaled up by different factors. The size of the cylinder gets larger as the camera gets closer to it. In addition to the size change, the distance between

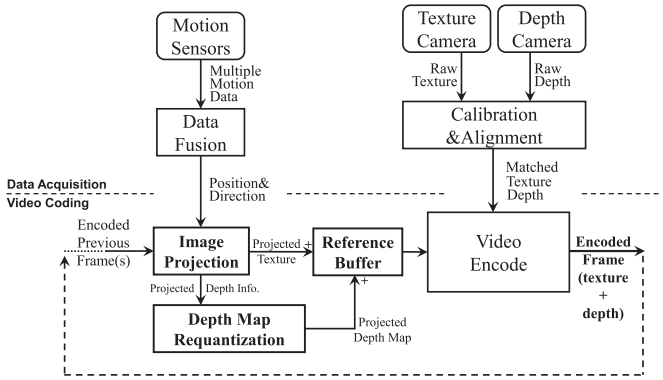

Fig. 3. Overview of the proposed motion information-based texture plus depth video coding scheme.

each object and camera becomes shorter; hence the gray level of depth map also changes. For example, the gray level of the cylinder gets darker (the darker a point is, the closer it is to the camera).

Tracking: For a camera with a tracking motion, the position of each object is shifted, while the relative distance between them changes, since the relative displacement of the cylinder is smaller than that of the cube.

Panning: With a panning camera, the position of each object gets shifted, meanwhile the shape of each object changes in general. For instance, the shape of the projected cube surface is changed from a rectangle into a trapezoid. Such distortion will increase the complexity of the motion estimation at the video encoder side. Meanwhile, the gray level of the depth map is hugely changed by the camera's panning motion. For example, it is worth noticing that the planar wall in the initial frame [i.e. Fig. 2(a)] gets represented by a gradual change of gray level after a panning motion, when this wall becomes nonparallel to the camera's image plane, which might lead to the failure of the temporal inter-predication mechanism.

In video coding, the conventional motion estimation method works well for videos with camera translation in a panning motion, but its efficiency drops for other motions. In the MTDC scheme, since the camera motion information is known, the temporal previous texture and depth images are projected to the current position of the camera and then used as reference frames, which are fairly similar to the current images. The overview of the MTDC scheme and the details of each procedure are introduced as follows.

A. Overview of the Proposed Scheme

The block diagram of the proposed motion information based texture plus depth video coding scheme is shown in Fig. 3. The proposed scheme includes data acquisition, processing and video coding. The data acquisition and processing are employed to obtain and process the proper video sequence and motion data, while the video coding part encodes the texture and depth sequences exploiting the motion information. In this paper, we focus on the single view case for simplicity. Nevertheless, it can be extended to multi-view texture plus depth format.

The main aim of the proposed scheme is to exploit the motion sensors available on most smart portable devices to enhance and assist the 3D video coding. Given the potential benefit of such technology, we assume a scenario where the smart devices' manufacturers exploit the proposed paradigm to enhance the video coding performance on their devices. In this case, the cameras' and sensors' parameters and their accurate configuration in addition to the intrinsic parameters of the camera should be available. Meanwhile, to improve an existing 3D camera by using the proposed method, some calibration would be needed to obtain the intrinsic parameters of the camera [25].

As the texture camera and depth camera might not be located at the same position, the texture image and depth map need to be aligned. Furthermore, the geometric distortion, caused by the optical system, must be compensated [25] to improve the quality of the projected (i.e. virtual) image. To obtain the camera motion information, motion sensors are used together with data fusion algorithms [26].

After obtaining and processing the video sequence and motion information, the texture sequence and depth map sequence are encoded. In this paper, an "implementation-friendly" method is proposed in which virtual reference frames are added into the reference buffer. The virtual reference frames are projected from temporally previous frames based on the camera's motion from the temporally previous position to the current position. Ideally, static objects and the background of the projected previous frame will be very similar to those of the current frame. The differences between the virtual frame and current frame are due to the moving objects, newly appearing objects, dis-occluded background objects, and illumination changes. Because of the added virtual reference frames, there is a high probability that the ME can find a more accurate motion vector (MV) and reduce residual energy.

Texture plus depth 3D video and the corresponding global motion information are sent to the receiver to guarantee proper decoding of the video. The camera motion information for each frame is totally represented by 12 bytes. So the extra bit rate is 2.4 kbps which has a slim impact on the used bandwidth. For example, the average bit rate of an HD resolution video is more than 2500 kbps, therefore, the proposed approach overhead is less than 0.1 %.

In the MTDC method, the image projection, depth map requantization and reference buffer reallocation are introduced in detail in the following sections.

B. Texture Frame Projection and Depth Map Requantization

Each pixel in the texture image needs to be converted from a 2D point into 3D coordinate space. Therefore, the 2D image becomes a 3D point cloud. First, the 3D coordinate system is defined as shown in Fig. 4. The x-axis and the y-axis define a plane surface parallel to the image plane, while the z-axis represents the depth. Let $\mathbf{P} = [w, h]$, where w and h represent the horizontal and vertical coordinate of a pixel in the image, respectively, while d is the corresponding depth. The 3D homogeneous coordinate of a pixel can be projected by

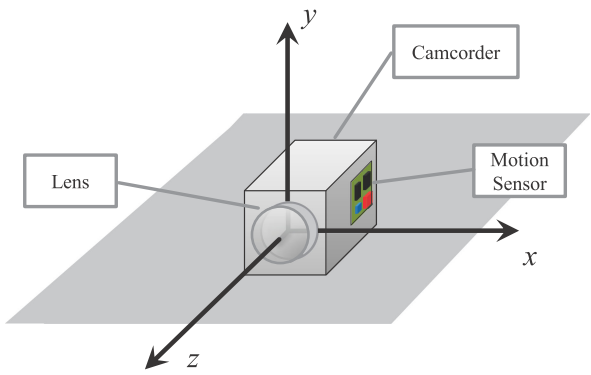

Fig. 4. 3D coordinate system definition for the camera.

$$\mathbf{C} = [x, y, z, 1] = \left[K\left(\frac{W}{2} - w\right)d, K\left(\frac{H}{2} - h\right)d, d, 1\right] \tag{1}$$

where W and H are horizontal and vertical resolution, in pixels, of the image respectively. K is the intrinsic parameter of the cameras, which is represented as

$$K = \frac{\tan(\phi_w)}{W} = \frac{\tan(\phi_h)}{H} \tag{2}$$

where ϕ_w and ϕ_h are the horizontal and vertical viewing angles respectively.

The 4×4 projective transformation matrix is represented by $\mathbf{T}$, which describes the translation and rotation from the previous view to the current view. In this paper, the 3D rotation is represented by *z-x-z* Eular angles, which are α, β and γ [27]. Accordingly, the 3D rotation can be factored into the following matrices:

$$\mathbf{T}_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\mathbf{T}_\beta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{T}_\gamma = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The 3D translation movements are represented as d_x, d_y and d_z

$$\mathbf{T}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_x & d_y & d_z & 1 \end{bmatrix}. \tag{6}$$

Therefore, the global camera motion can be represented as

$$\mathbf{T} = \mathbf{T}_\gamma \times \mathbf{T}_\beta \times \mathbf{T}_\alpha \times \mathbf{T}_t. \tag{7}$$

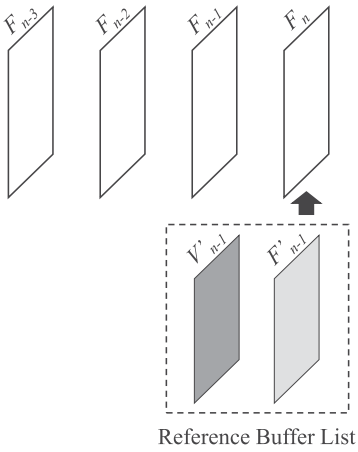

Fig. 5. Example of the reallocation of reference buffer list with V'_{n-1} representing the virtual frame projected from F'_{n-1}.

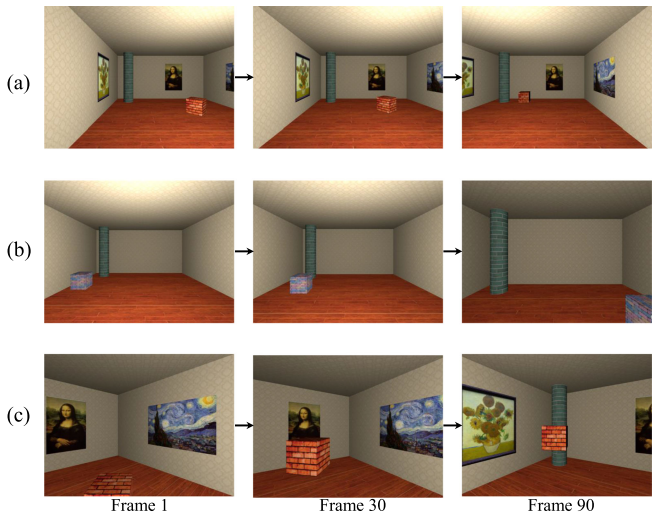

Fig. 6. Different motion types of simple CG scenes used for VGA resolution experiments, including (a) tracking, (b) dolly, and (c) panning (Frame 1, 30, and 90).

The new coordinate of a pixel on the virtual frame can be obtained using

$$\mathbf{C}_v = \mathbf{C} \times \mathbf{T} = [x_v, y_v, z_v, 1]. \tag{8}$$

The 3D coordinate of each pixel in the virtual view has to be now inversely converted to a 2D coordinate in the image using the following equation:

$$\mathbf{P}_v = [w_v, h_v] = \left[\frac{W}{2} - \frac{x_v}{K z_v}, \frac{H}{2} - \frac{y_v}{K z_v}\right]. \tag{9}$$

Then, each pixel on the previous frame can be transformed to a new 2D coordinate in the projected image. However, some of them might end up being located outside of the image and some of the obtained coordinates might not be integer, which could lead to holes in the virtual reference frame. Therefore, an interpolation algorithm is utilized to fill holes and smooth the virtual image. In this paper, the Bicubic interpolation algorithm is used [28]. Finally, the projected image is represented as $\mathbf{V}$.

As the depth z of each pixel in (1) has been quantized to an integer n in the depth map frame, in order to get the real corresponding depth z, it has to be de-quantized first. Moreover, as the global motion of camera might change the depth of the projected pixels, therefore, the new depth z_v will be used to generate the depth map, which will be quantized using the same quantization method.

C. Reallocation of Reference Buffer List

Fig. 5 shows one example of the proposed reference buffer reallocation scheme. In this example, the previous frame F'_{n-1} is used in addition to the virtual frame V'_{n-1} which has been projected from F'_{n-1}. It is worth mentioning that more memory size is required to store the extra virtual frame. Nevertheless, this has slim impact on the system's resources, for example for HD video this is around 2073.6 kB.

It is worth mentioning that some popular codecs (such as HEVC and H.264/AVC) use quarter-pixel motion estimation and motion prediction. Their reference buffer list stores one or more up-sampled reconstructed frames. Therefore, before inserting the virtual reference frames into the reference buffer list, they have to be up-sampled using the same algorithm used by the video encoder.

III. Experimental Method and Results

To the best of our knowledge, there are no suitable standard sequences where the global motion information is obtained during the recording stage. For example, [29] provided some sequences by obtaining information from a laser-scanner, but the depth map is not very well-aligned with the RGB texture and the resolution of depth map is much lower than texture image. In [30], the depth is obtained from Kinect, which provides a low accuracy measurement of distance and suffers from many holes. Consequently, to test the proposed approach, some sequences[1] with synchronously sampled global motion information were generated using computer graphics (CG) technology and a customized hardware platform, respectively.

A. CG Sequences-Based Experiments and Results

Computer graphics technology produces accurate texture and depth video, which is suitable for generating test sequences. Thus, Blender 2.76[2] is used to produce the CG sequences including aligned texture and depth.

Different typical camera motions were emulated to produce different video sequences for experiments, including tracking (right translation), dolly (forward and backward translation) and panning (anticlockwise rotation) as shown in Fig. 6 (VGA resolution, Frame 1, 30 and 90 with normal speed) and Fig. 7 (HD resolution).

In the CG video texture experiments, the image resolution is VGA (640 × 480) @ 25 fps for both H.264/AVC and

[1]All the CG and real scene sequences are available for downloading at http://www.mmtlab.com/mtdc.ashx.

[2][Online]. Available: http://download.blender.org/release/

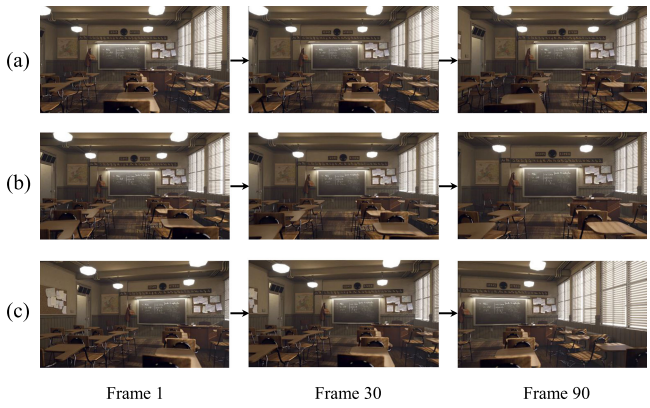

Fig. 7. Different motion types of detailed CG scenes used for HD resolution experiments, including (a) tracking, (b) dolly, and (c) panning (Frame 1, 30, and 90).

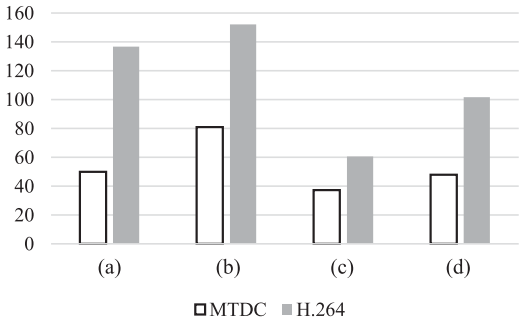

Fig. 8. Average number of intra-blocks per P-frame of texture sequences for different speeds of motion using the MTDC scheme and standard H.264/AVC for different motion types: (a) dolly (forward), (b) dolly (backward), (c) tracking, and (d) panning.

H.265/HEVC based tests, while it is HD (1920 $\times$ 1080) for H.264/HEVC based tests, respectively. The size of the image sensor is 32 mm $\times$ 24 mm, while the focal lengths are 18 mm for VGA sequences and 25 mm for HD sequences. To evaluate the proposed MTDC scheme for different coding standards, H.264/AVC JM reference code 14.1 and H.265/HEVC HM reference code 16.9 were modified to add virtual frames to the reference buffer list. Each virtual frame V'_{n-1} is projected from the temporally previous frame f'_{n-1} using the motion and depth information. For each type of camera motion, different speeds were tested, which are 1x, 3x and 5x of a nominal motion speed. For tracking and dolly motion, the average speed is around 2m per second, while for the panning motion, the average normal rotation rate is 18 degrees per second. To span a reasonable range of bit rate, the Quantization Parameter (QP) is set from 20 to 34. The depth maps used for texture tests are encoded with QP from 10 to 24.

The proposed method makes the reference frame(s) more similar to the current to-be-encoded frame, which increases the percentage of P blocks and reduces the number of I blocks. Fig. 8 demonstrates the comparison of the average I block numbers per P-frame of sequences of different speeds for different types of motion using the proposed MTDC method and standard H.264/AVC. The proposed MTDC method observably decreases

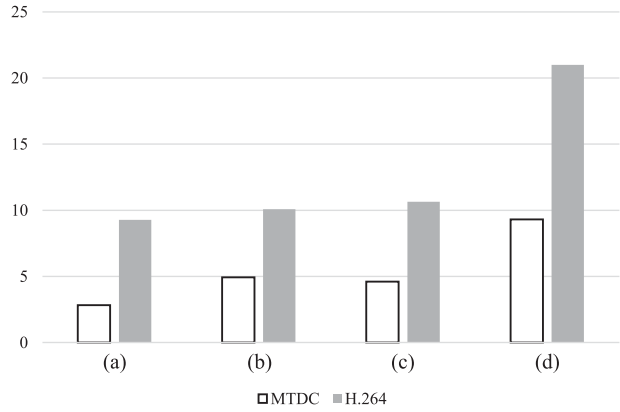

Fig. 9. Average motion vector length of texture sequences for different speeds motion using the MTDC scheme and standard H.264/AVC for different motion types: (a) dolly (forward), (b) dolly (backward), (c) tracking, and (d) panning.

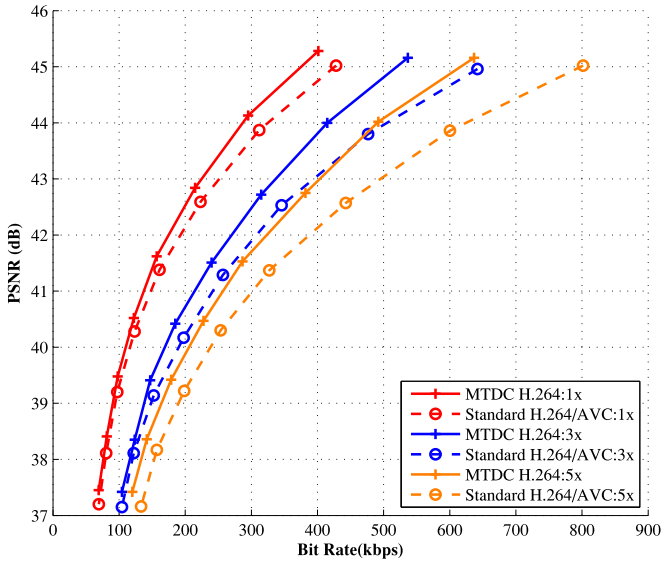

Fig. 10. PSNR versus bit rate of texture sequence for the H.264/AVC based MTDC scheme and standard H.264/AVC for translation motion under different camera speeds (1x/3x/5x).

the number of I blocks, which is positive for reducing the total bit rate. Meanwhile, the proposed MTDC method reduces the average length of the motion vectors for each sequence, as shown in Fig. 9.

1) H.264/AVC-Based Tests of VGA Resolution Texture Sequences: In the camera tracking experiments, the camera is moving from left to right with nonuniform speed. Three groups of speed settings are tested in order to evaluate the RD performance under different speed conditions. In this scene, the wall of the room is covered by a simple wallpaper with some paintings, while the brick cube is moving forward and rotating. Fig. 10 shows the RD performance comparison of the H.264/AVC based MTDC scheme and standard H.264/AVC under this condition. From Table I, the BD-PSNR (with the standard H.264/AVC as an anchor) indicates that the proposed MTDC achieves better performance than the standard H.264/AVC at all speeds. The

TABLE I
EXPERIMENTAL RESULTS OF H.264/AVC-BASED VGA RESOLUTION CG TEXTURE SEQUENCE

Motion Type	1x		3x		5x	
	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)
Tracking	0.35	−8.9	0.56	−14.2	0.76	−19.6
Forward	0.55	−16.2	1.17	−38.4	1.08	−35.5
Backward	0.39	−11.4	0.54	−15.2	0.85	−24.5
Panning	0.34	−9.0	0.48	−12.5	0.79	−20.4

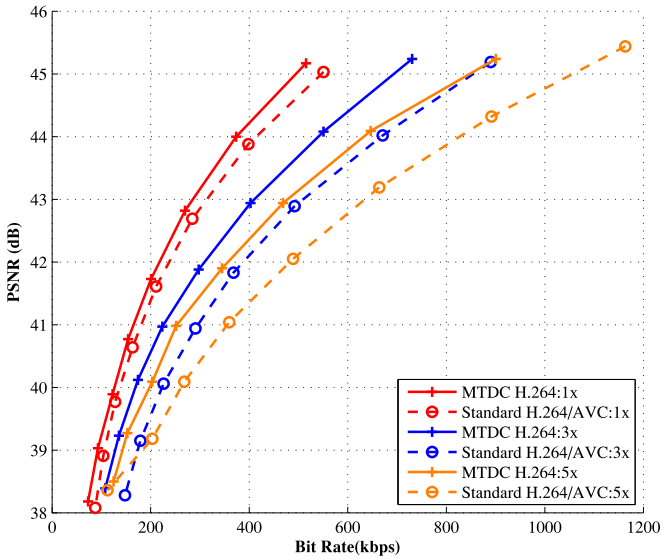

Fig. 11. PSNR versus bit rate of texture sequence for the MTDC scheme and standard H.264/AVC in dolly motion (forward) under different camera speeds (1x/3x/5x).

coding performance gain is up to 0.76 dB, while the BD-Rate is −19.6%.

In the camera dolly motion experiments, the camera is moving forward or backward. Fig. 11 shows the RD performance results of forward motion. From Table I, the average gain for different speeds of forward motion is about 0.93 dB, which achieves better performance than tracking motion, which could be explained from two aspects. In the forward motion type, normally there are no new objects appearing in the scene, while in tracking motion, new objects could enter the captured scene. The other reason is that the H.264/AVC standard encoder cannot deal with the zoom-in or zoom-out effect, while the proposed MTDC scheme is able to address this problem properly. Fig. 12 represents the RD performance curve of backward motion. As there are some new object appearing into the viewed scene, the average gain is lower than for forward motion, nevertheless the gain in the backward motion is around 0.6 dB.

In the camera panning experiments, the camera is rotating anti-clockwise. For panning motion, it is worth noting that the depth map is not necessary for view projection. Fig. 13 shows the RD performance result. The average BD-PSNR is around

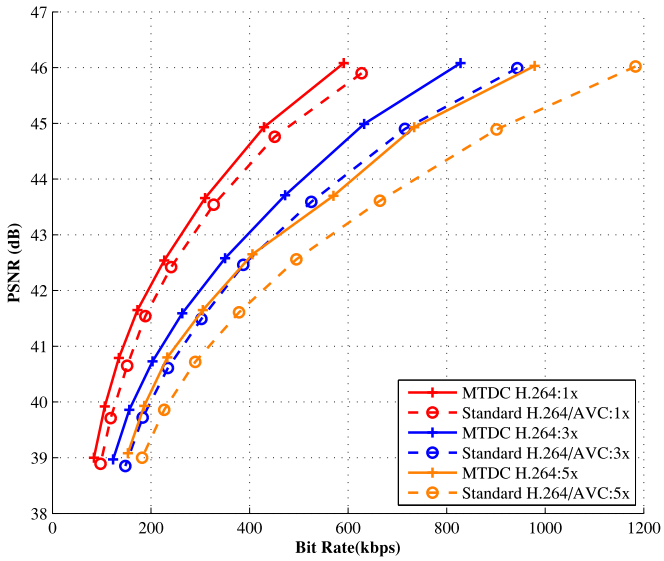

Fig. 12. PSNR versus bit rate of texture sequence for the MTDC scheme and standard H.264/AVC in dolly motion (backward) under different camera speeds (1x/3x/5x).

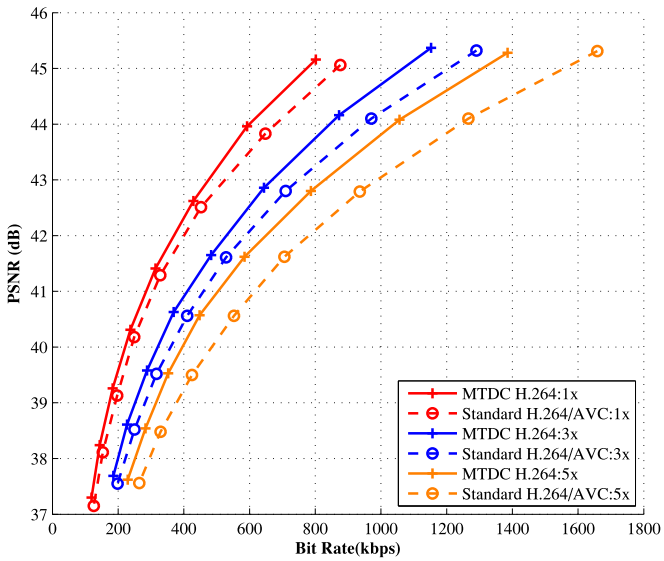

Fig. 13. PSNR versus bit rate of texture sequence for the MTDC scheme and standard H.264/AVC in panning motion (anticlockwise rotation) under different camera speeds (1x/3x/5x).

0.54 dB, while the average BD-Rate is about 14% for the panning motion.

All of the H.264/AVC based experimental results for VGA resolution texture sequences are reported in Table I, which shows that the MTDC outperforms the H.264/AVC standard. The range of BD-PSNR gain is from 0.35 dB to 1.17 dB, while the BD-Rate is from -8.9% to -38.4%. Generally, with an increase in camera speed, the gain of the MTDC scheme also increases. The MTDC scheme achieves the highest performance for the dolly camera motion (forward), and in general the panning motion achieves the lowest gain. The reason is that in the panning motion, some new objects appear into the captured scene. When the speed is unduly fast, the motion estimation might fail due to the huge

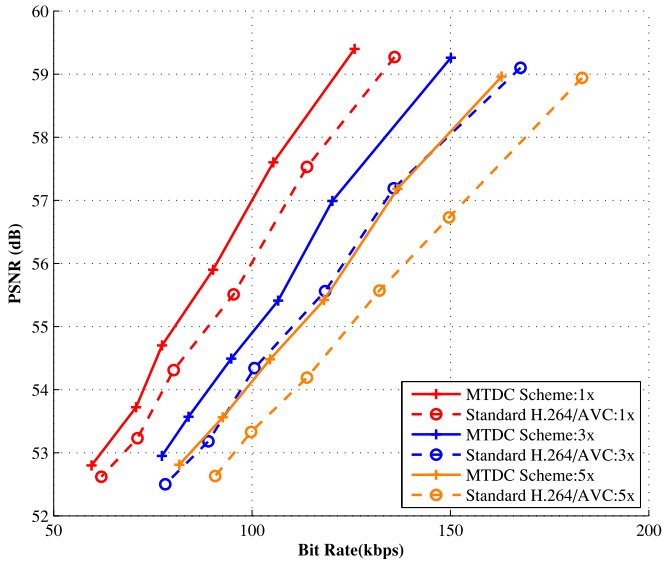

Fig. 14. PSNR versus bit rate of depth sequence for the H.264/AVC-based MTDC scheme and standard H.264/AVC in translation motion under different camera speeds (1x/3x/5x).

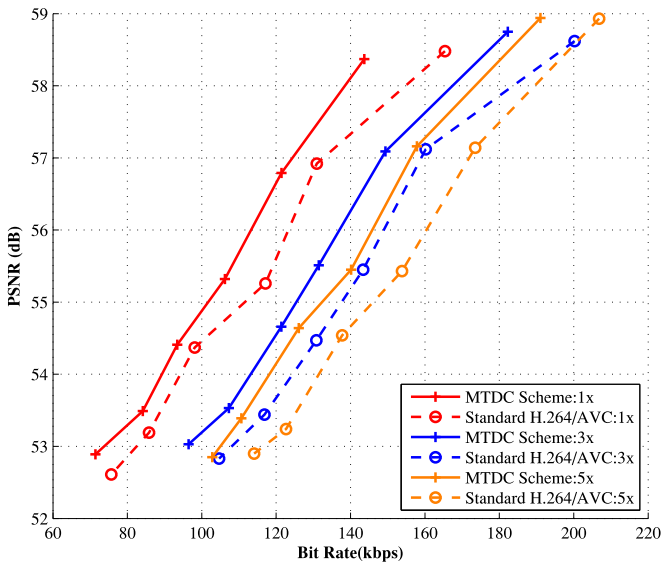

Fig. 15. PSNR versus bit rate of depth sequence for the MTDC scheme and standard H.264/AVC in dolly motion (forward) under different camera speeds (1x/3x/5x).

error in projection, which will reduce the gain of the proposed MTDC scheme.

Future work should take into account the use of B-like virtual frames, which will contribute to tracking, backward, and panning motions.

2) H.264/AVC-Based Tests of VGA Resolution Depth Sequences: In this section, the RD performance of the proposed method was evaluated for the compression of depth map sequences. The depth map sequences are the corresponding depth maps of the texture sequences used for the previous VGA texture tests, and they were generated by the uniform quantization method. For each type of camera motion, different camera speeds are tested, which are 1, 3 and 5 times faster than the nominal speed. To span a reasonable range of bit rates, the QP is set from 12 to 24. The experimental coding software is modified from the H.264 based MTDC texture encoder by adding the depth map re-quantization.

Similarly, interpolation is utilized to fill holes and smooth the projected depth frame in order to assure that the projected one is similar to the current encoded depth frame.

In tracking experiments, Fig. 14 shows that the average BD-PSNR gain is about 0.41 dB for the depth scene.

In the dolly motion experiments, Fig. 15 shows that the average gain is 0.76 dB. From the comparison between dolly motion and tracking motion, the MTDC scheme demonstrates better gain for dolly motion than tracking motion for the depth map. Meanwhile, the forward motion might change the near and far points in the scene, which makes the gray level of the same surface change from frame to frame in the depth sequence. The proposed MTDC scheme is able to reduce this impact and consequently improves the coding performance of the depth map.

The coding performance for panning motion is increased by 1.23 dB on average, as shown in Fig. 16, which is better than those for tracking and dolly motion. For the panning motion,

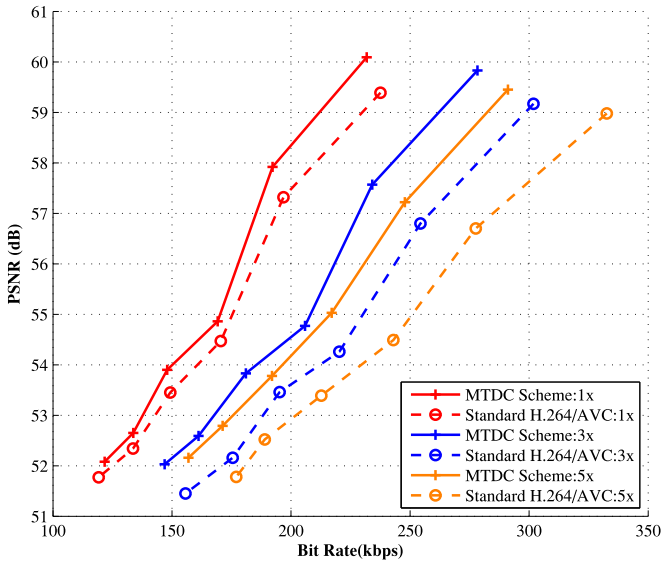

Fig. 16. PSNR versus bit rate of depth sequence for the MTDC scheme and standard H.264/AVC in panning motion under different camera speeds (1x/3x/5x).

the gray level representing the depth of the same objects might change more than it does for the dolly motion. So for example, a surface parallel to the image plane will appear as a uniform gray level surface in the depth image at this position, but when the camera rotates, the surface will not be parallel to the image plane and consequently its projected image will appear as a gradual change of gray level. Indeed, the standard H.264 cannot deal with such a change, while the propose scheme could encode the depth map efficiently.

All of the experimental results for depth sequence are reported in Table II, which demonstrates that the H.264/AVC based MTDC scheme is better than the H.264/AVC standard. The range of BD-PSNR gain is from 0.5 dB to 1.7 dB, while

TABLE II
EXPERIMENTAL RESULTS OF H.264/AVC-BASED VGA RESOLUTION CG DEPTH SEQUENCE

Motion Type	1x		3x		5x	
	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)
Tracking	0.29	−4.1	0.39	−4.7	0.56	−7.6
Forward	0.60	−8.3	0.73	−8.5	0.95	−9.4
Panning	0.57	−7.4	1.27	−12.6	1.59	−16.7

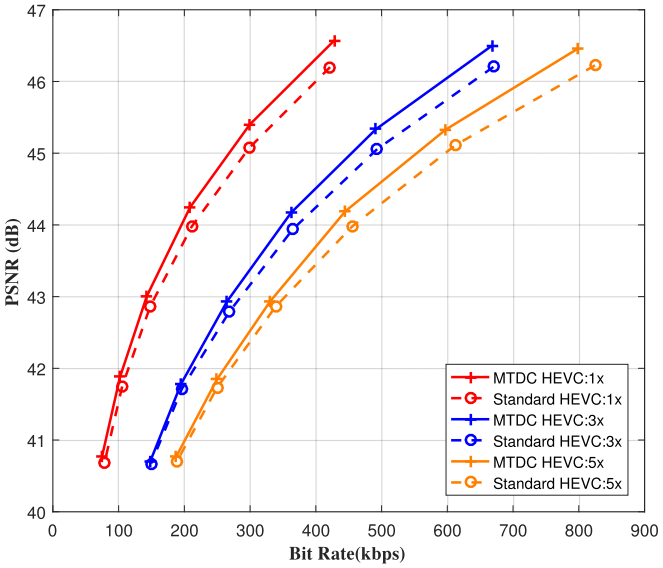

Fig. 17. PSNR versus bit rate of VGA resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in translation motion under different camera speeds (1x/3x/5x).

the BD-Rate increase is from −3.8% to −18%. The MTDC achieves the highest performance for the panning motion.

3) H.265/HEVC-Based Tests of VGA Resolution Sequences: The H.265/HEVC is the current state-of-art video coding standard, which improves video coding performance with respect to H.264/AVC. In this paper, the MTDC scheme is also implemented using H.265/HEVC by HM reference code version 16.9. Firstly, VGA resolution sequences with simple scenes are tested with various types of motion and speed.

In the camera tracking motion tests, Fig. 17 shows that the average gain of the H.265/HEVC based MTDC scheme is around 0.25 dB, which is lower than the one obtained using H.264 for the same sequence. This is because the texture of scene is simple and the efficiency of intra-prediction of HEVC is higher than that of H.264, thus the margin of H.265/HEVC improvement gets reduced. In the camera dolly motion (forward) experiments, the gain is lower than for the tracking motion as shown in Fig. 18. But the average gain is still 0.2 dB. The reason is that while zooming into the objects, less details appear in the image, which is easier for the intra-prediction of HEVC. In the camera backward motion experiments, the new appearing objects impact the

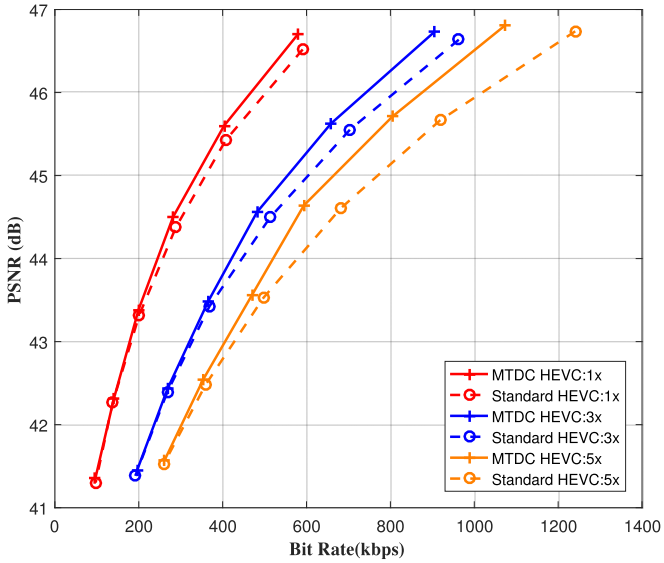

Fig. 18. PSNR versus bit rate of VGA resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in dolly motion (forward) under different camera speeds (1x/3x/5x).

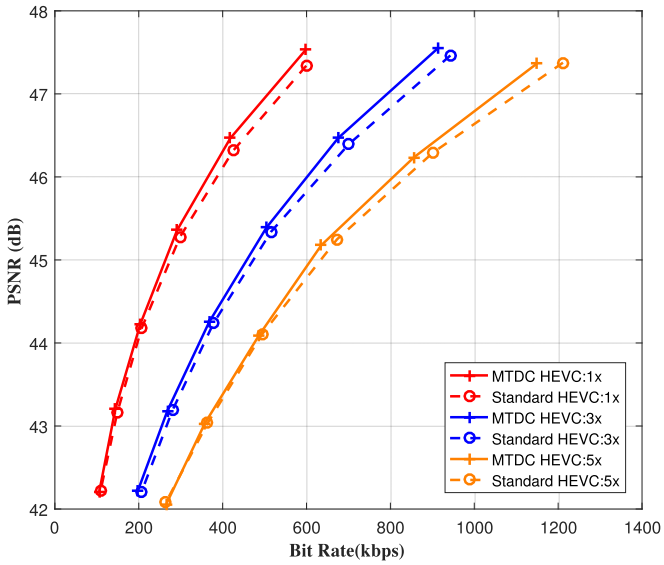

Fig. 19. PSNR versus bit rate of VGA resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in dolly motion (backward) under different camera speeds (1x/3x/5x).

efficiency of the MTDC scheme further. Fig. 19 and Table III show that the average gain of backward motion is about 0.12 dB. The average gain for the panning motion is around 0.11 dB as represented in Fig. 20 and Table III.

All the experimental results for VGA resolution texture sequences for H.265/HEVC are reported in Table III. The average gain is lower than that of H.264 because of the improvement of the intra-prediction in H.265/HEVC. When the scene is very simple, the efficiency of the initial H.265/HEVC is quite high, which reduces the gain of the proposed MTDC scheme.

TABLE III
EXPERIMENTAL RESULTS OF H.265/HEVC-BASED VGA RESOLUTION CG TEXTURE SEQUENCE

Motion Type	1x		3x		5x	
	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)
Tracking	0.28	−9.0	0.21	−5.8	0.22	−6.0
Forward	0.13	−3.6	0.17	−5.4	0.36	−10.7
Backward	0.12	−3.4	0.13	−3.7	0.10	−2.6
Panning	0.11	−3.2	0.10	−2.1	0.11	−2.7

TABLE IV
EXPERIMENTAL RESULTS OF H.265/HEVC-BASED HD RESOLUTION CG TEXTURE SEQUENCE

Motion Type	1x		3x		5x	
	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)	BD- PSNR (dB)	BD- Rate (%)
Tracking	0.28	−9.6	0.22	−6.9	0.21	−6.1
Forward	0.40	−13.4	0.33	−9.6	0.30	−8.6
Backward	0.37	−12.0	0.30	−8.8	0.26	7.5
Panning	0.23	−8.1	0.22	−6.6	0.20	−5.5

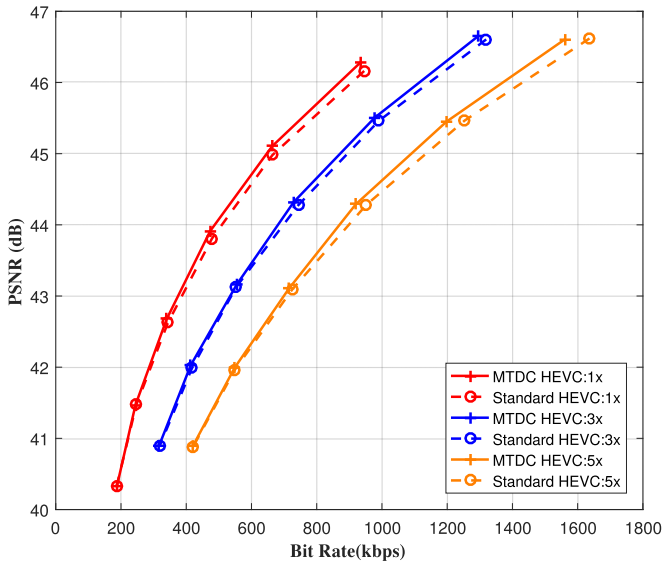

Fig. 20. PSNR versus bit rate of VGA resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in panning motion (anticlockwise rotation) under different camera speeds (1x/3x/5x).

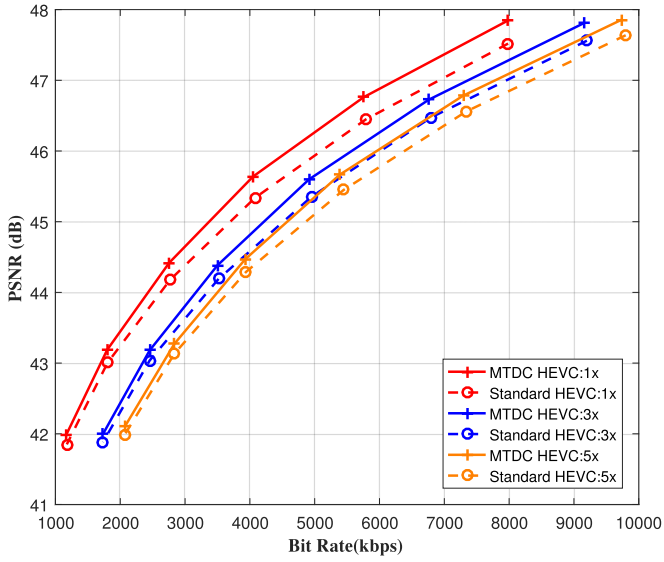

Fig. 21. PSNR versus bit rate of HD resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in translation motion under different camera speeds (1x/3x/5x).

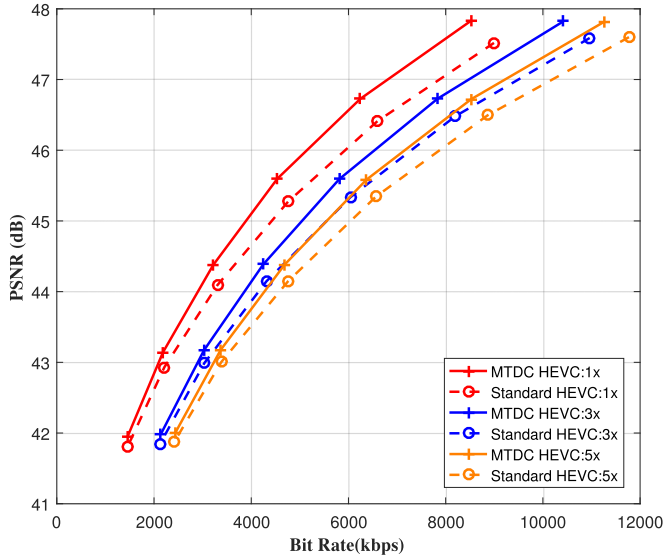

Fig. 22. PSNR versus bit rate of HD resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in dolly motion (forward) under different camera speeds (1x/3x/5x).

Nevertheless, the proposed MTDC scheme can still enhance the performance of H.265/HEVC.

4) H.265/HEVC-Based Tests of HD Resolution Sequences: HD resolution(1920 × 1080) sequences for different motions with many details are used for testing the performance of the H.265/HEVC based MTDC scheme, as shown in Fig. 7. The scene we used is a CG classroom with many desks, chairs, blackboard, clock and windows etc, which is named as "Classroom".[3]

In the camera tracking experiments, Fig. 21 shows the RD performance of the proposed MTDC scheme for H.265/HEVC. From Table IV, the gains of 1x, 3x, 5x are 0.28 db, 0.21 dB and 0.22 dB. Figs. 22 and 23 represent the RD performances for forward and backward motion, respectively. It is possible to note in Table IV that the average gain for forward motion is 0.40 dB, while the gain for backward motion is 0.37 dB, which is better than the gain for the tracking motion. The reason is that MTDC is able to deal with an image zooming in and zooming out. In the panning experiments, the RD performance is shown in Fig. 24. The average gain is around 0.22 dB. The gain is

[3][Online]. Available: https://www.blender.org/download/demo-files/

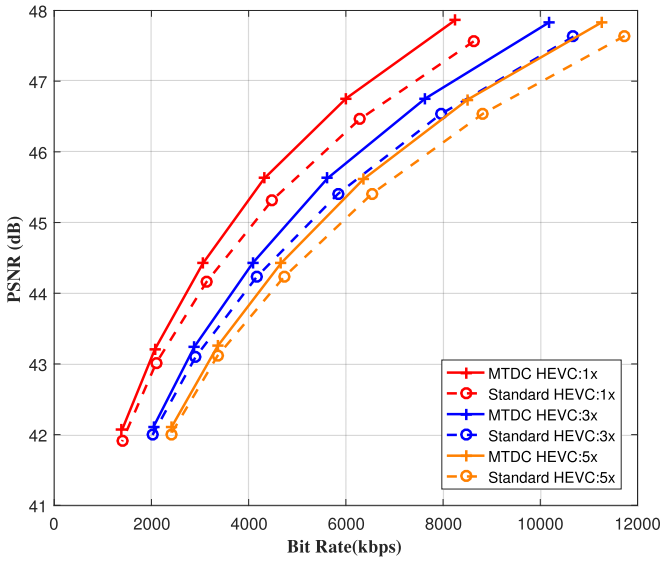

Fig. 23. PSNR versus bit rate of HD resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in dolly motion (backward) under different camera speeds (1x/3x/5x).

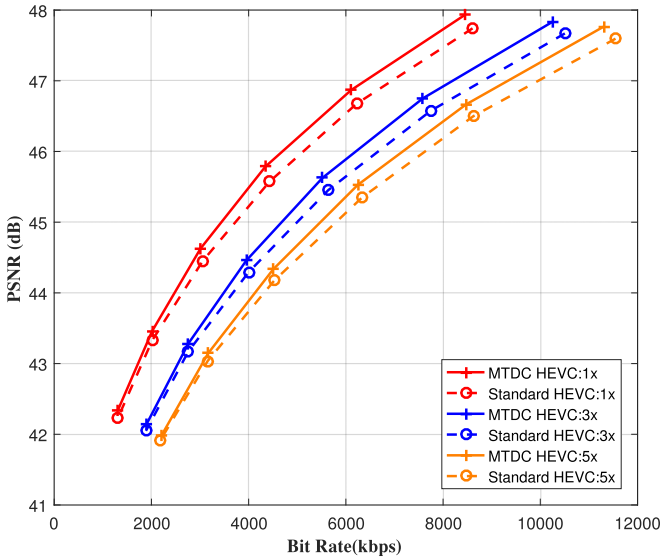

Fig. 24. PSNR versus bit rate of HD resolution texture sequence for the H.265/HEVC-based MTDC scheme and standard H.265/HEVC in panning motion (anticlockwise rotation) under different camera speeds (1x/3x/5x).

lower when the speed increases, this is caused by the big pixel distance of HD resolution.

All the experimental results for HD resolution texture sequences for H.265/HEVC are reported in Table IV. The average gain is higher than that of H.265/HEVC based VGA sequences because the scene is much more complex than the VGA scene. But with the increasing of camera speed, the gain becomes lower. The reason is that the projection distortion becomes bigger when the virtual synthesized view becomes further away from the original image.

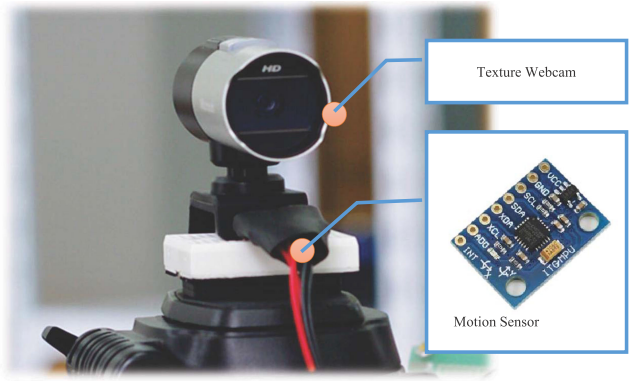

Fig. 25. Customized platform for the rotational motion.

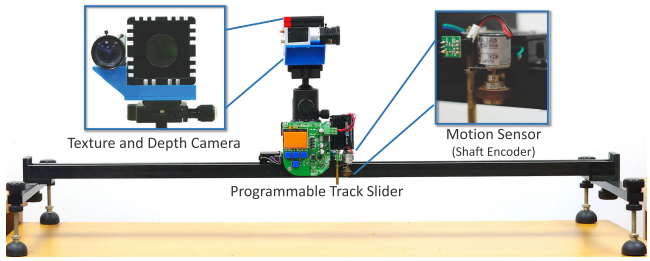

Fig. 26. Customized platform for the translational motion.

B. Real Scene Data Acquisition, Platforms, and Results

Normally, the motion information should be obtained from inertial motion sensors. The current inertial motion sensors which are widely used in smart phones generally integrate accelerometers, gyroscopes and compasses. With the data fusion technology [31], they are able to obtain relatively accurate rotational motion information. However, the measurement of translation motion using inertial sensors is not so accurate, due to the limitation of the sensor technology. In this work, aninertial motion sensor is applied for rotational motion tests, while a distance sensor is employed with a track slider for translational motion tests to provide an upper performance bound.

Fig. 25 presents the platform for the rotational motion test. The motion sensor is InvenSense MPU6050, and a Microsoft HD WebCam is employed as the camera.

Fig. 26 shows the platform for the translational motion test, which generates smooth and accurate translational motion information. A Balser acA640-90gc camera is used to capture texture video, while the depth camera is Mesa Imaging SwissRanger SR4000.

In the rotation experiment, the image resolution is VGA (640×480) @ 25 fps. The video sequence is captured when the camera was rotated clockwise around the y-axis. The average angle of rotation per frame is around 1.3 degrees, while each angle is recorded respectively. To span a reasonable range of bit rates, the QP was set from 20 to 34. Fig. 27 shows an example of this experiment. Frame F'_{n-1} is the previous frame, while the virtual frame V'_{n-1} is the projected frame from F'_{n-1} using the motion information of the camera. By looking at the left edge of each image, it is possible to note that V'_{n-1} is more

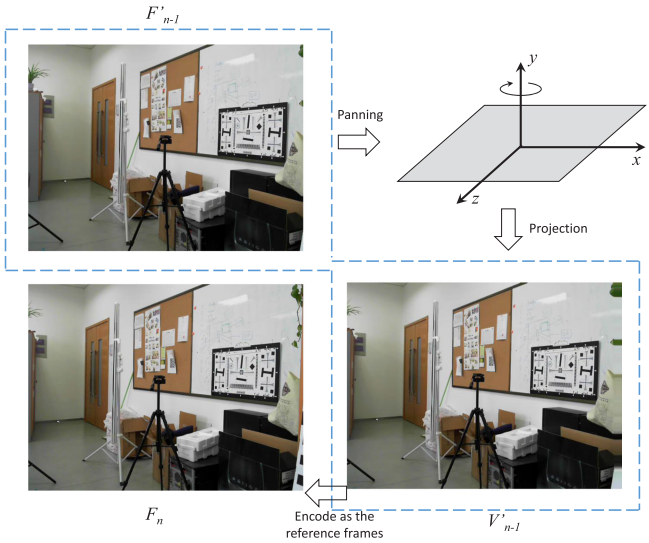

Fig. 27. Example of encoding process of the panning motion experiment. F_n is the current frame; F_{n-1} is previous frame; and V_{n-1} is the virtual reference frame.

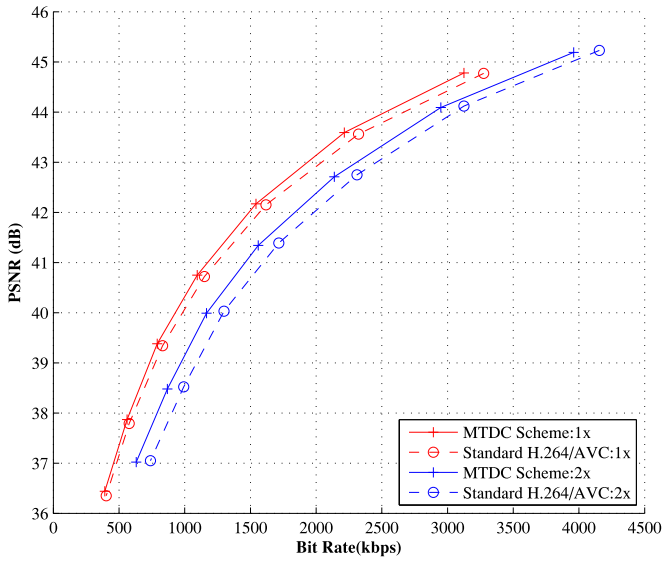

Fig. 28. PSNR versus bit rate for the proposed scheme and standard H.264/AVC in panning motion; the panning angle is around 1.33 (1x) and 2.66 (2x) degree per frame, respectively.

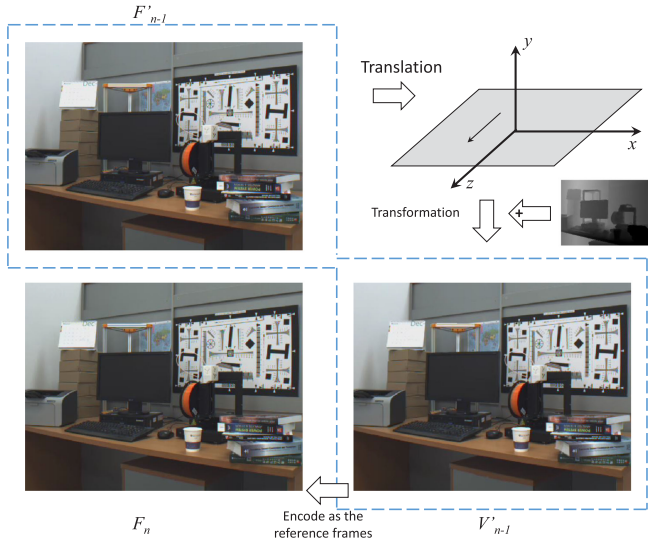

Fig. 29. Example of encoding process of the dolly motion experiment.

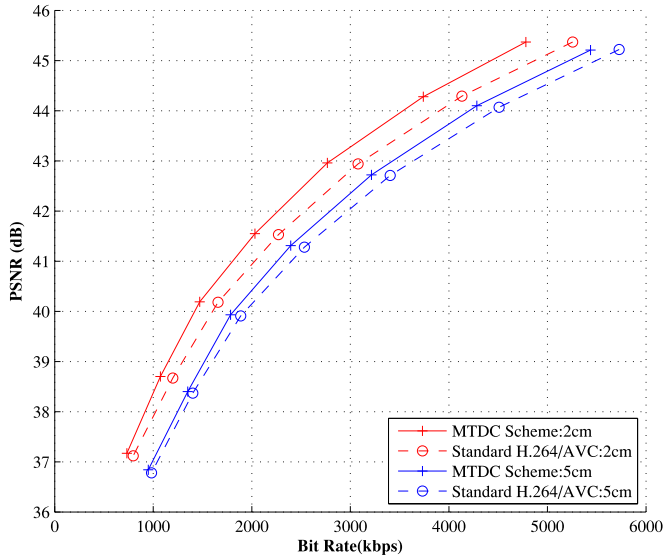

Fig. 30. PSNR versus bit rate of texture sequence for the proposed scheme and standard H.264/AVC in dolly forward motion; the distance traveled is 2 cm and 5 cm per frame, respectively.

similar to the current frame F_n. Fig. 28 presents the PSNR comparison between the proposed MTDC scheme and standard H.264/AVC. The BD-Rate was -5.48%, while the BD-PSNR gain was 0.22 dB.

In the forward dolly motion experiment, the resolution of texture images is VGA (640×480) @ 25 fps, while the resolution of depth images is QCIF (176×144). Therefore, the depth images were projected, up-sampled and cropped in order to have them aligned with the texture images. The encoder settings were the same for the panning test. In Fig. 29, an example of the dolly test is described. The virtual frame V'_{n-1} was projected from the previous frame F'_{n-1} using the depth information and global motion information. It is worth noting that the distance between the first book and the right image edge in F'_{n-1} is bigger than that in F_n, but that distance in V'_{n-1} is closer to that in F_n, which means V'_{n-1} is more similar to F_n. In order to evaluate the impact of different translational speeds, the moving speed is set at 2 cm per frame and then at 5cm per frame. The experimental results are shown in Fig. 30. Comparing with H.264/AVC, the BD-Rate is -11.87%, while the BD-PSNR gain is 0.5 dB for the video with 5cm/frame speed. Although the 5cm/frame test presents a lower coding gain than the 2 cm/frame test, it is still better than standard H.264/AVC, with the BD-Rate being -5.7% and the BD-PSNR being 0.3 dB. The reason is that the real depth map used for this test is up-sampled from QCIF, which is not very accurate. So the larger motion leads to more errors in projection. When the depth map becomes accurate enough,

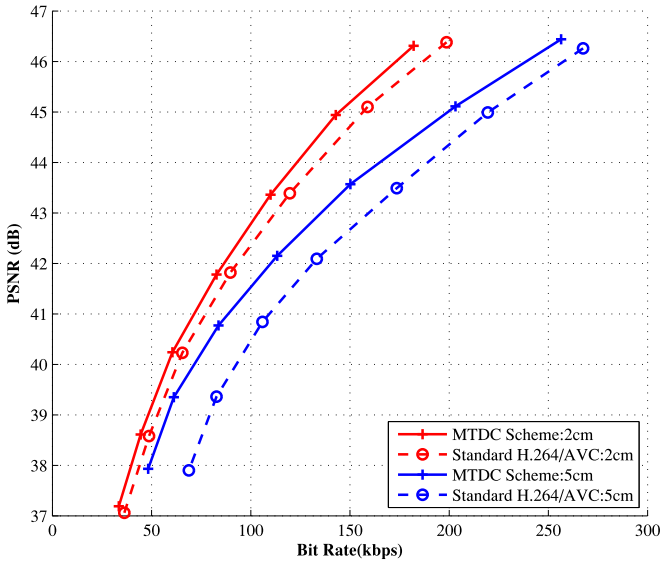

Fig. 31. PSNR versus bit rate of depth sequence for the proposed scheme and standard H.264/AVC in dolly forward motion; the distance is 2 cm and 5 cm per frame, respectively.

the performance of real videos is expected to approach that of CG videos.

For the real scene, besides the texture video, the coding performance of depth is also tested. The original size (QCIF) depth map from a SR4000 depth camera is used. The dolly (forward) motion is applied as previously, while the camera motion speeds are set at 2 cm per frame and then 5 cm per frame, respectively. In Fig. 31, for the depth video the BD-PSNR is around 0.3 dB for the video with 2 cm/frame speed, while the BD-PSNR is around 0.7 dB for the video with 5 cm/frame speed.

IV. Application and Limitation

This section will discuss and analyze the suitability and practicability of the proposed approach. The proposed scheme could be used for real-time video coding in smart portable devices. The enabling technologies are in fact available in most of today's portable devices. Firstly, motion sensors have been integrated into current mobile devices and their accuracy is continuously increasing with new sensor technology. Secondly, more and more mobile phones are using two or more cameras, and some are even integrating depth cameras on their platforms. Thirdly, the computational capability of mobile devices' CPU's and GPU's are surging.

Although in this paper CG sequences and a track slider were used for proof of concept, future industry platforms could use and integrate the previously mentioned enabling technologies in a cooperative way to aid in video compression.

The main limitations are the additional complexity of the virtual reference frame generation, and the proportion of moving objects in the scene may affect the MTDC's performance. Those limitations are not significant, in fact the total coding time increases by 21% with a non-optimized implementation, in terms of complexity, of the proposed approach. Meanwhile, the proportion of moving objects will influence the RD performance of proposed method. Nevertheless, we should recall that, in general, the proportion of moving objects is not very high in video sequences.

V. Conclusion

This paper has introduced a novel texture plus depth video coding scheme using the global motion information of the camera. The experiments using different coding standards and different types of sequences have demonstrated that the proposed MTDC scheme is able to improve the coding performance of the H.264/HEVC and H.265/HEVC standards, respectively. For the H.264/AVC, the VGA resolution CG texture and depth sequence coding performance are enhanced by 2 dB and 1 dB, respectively, while for the realistic texture video, the performance improvement is around 0.5 dB for texture video and 0.7 dB for depth video. For H.265/HEVC, the MTDC scheme enhances the coding performance using VGA resolution and HD resolution sequences by up to 0.3 dB and 0.4 dB, respectively. It is noted that the accuracy of the depth and motion information affects the performance of realistic video coding. With the further quality improvements in depth cameras and motion sensors, the performance of realistic video will be improved and approach the performance of CG video.

References

[1] P. Benzie *et al.*, "A survey of 3DTV displays: Techniques and technologies," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1647–1658, Nov. 2007.

[2] M. Draelos, Q. Qiu, A. Bronstein, and G. Sapiro, "Intel realsense = Real low cost gaze," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 2520–2524.

[3] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[4] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[5] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H. 264-ISO/IEC 14496-10 AVC, 2005.

[6] M. Hannuksela, Y. Chen, T. Suzuki, J. Ohm, and G. Sullivan, Eds., "AVC Draft Text 8," JCT-3V Doc. JCT3V-F1002, vol. 16, 2013.

[7] Y. Chen, M. M. Hannuksela, T. Suzuki, and S. Hattori, "Overview of the MVC+ D 3D video coding standard," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 679–688, 2014.

[8] Y. Chen, X. Zhao, L. Zhang, and J.-W. Kang, "Multiview and 3D video compression using neighboring block based disparity vectors," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 576–589, Apr. 2016.

[9] F. Shao, G. Jiang, W. Lin, M. Yu, and Q. Dai, "Joint bit allocation and rate control for coding multi-view video plus depth based 3D video," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1843–1854, Dec. 2013.

[10] H. Yuan, S. Kwong, X. Wang, W. Gao, and Y. Zhang, "Rate distortion optimized inter-view frame level bit allocation method for MV-HEVC," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2134–2146, Dec. 2015.

[11] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. IEEE Int. Conf. Image Process.*, Sep.–Oct. 2007, vol. 1., pp. I-201–I-204.

[12] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, 2004.

[13] J. Seo, D. Park, H.-C. Wey, S. Lee, and K. Sohn, "Motion information sharing mode for depth video coding," in *Proc. 3DTV-Conf.: True Vis,—Capture, Transmiss. Display 3D Video*, Jun. 2010, pp. 1–4.

[14] P.-J. Lee and X.-X. Huang, "3D motion estimation algorithm in 3D video coding," in *Proc. Int. Conf. Syst. Sci. Eng.*, Jun. 2011, pp. 338–341.

[15] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Process., Image Commun.*, vol. 24, no. 1, pp. 89–100, 2009.
[16] F. Zou, D. Tian, A. Vetro, H. Sun, O. C. Au, and S. Shimizu, "View synthesis prediction in the 3-D video coding extensions of AVC and HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1696–1708, Oct. 2014.
[17] S. Ma, S. Wang, and W. Gao, "Low complexity adaptive view synthesis optimization in HEVC based 3D video coding," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 266–271, Jan. 2014.
[18] K.-Y. Hsu and S.-Y. Chien, "Hardware architecture design of frame rate up-conversion for high definition videos with global motion estimation and compensation," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2011, pp. 90–95.
[19] A. Abou-Elailah, F. Dufaux, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion of global and local motion estimation for distributed video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 158–172, Jan. 2013.
[20] X. Chen, Z. Zhao, A. Rahmati, Y. Wang, and L. Zhong, "Sensor-Assisted video encoding for mobile devices in real-world environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 3, pp. 335–349, Mar. 2011.
[21] F. Cheng, J. Xiao, T. Tillo, and Y. Zhao, "Global motion information based depth map sequence coding," in *Advances in Multimedia Information Processing–PCM 2015*. New York, NY, USA: Springer, 2015, pp. 721–729.
[22] K. Muller *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.
[23] M. M. Hannuksela *et al.*, "Multiview-video-plus-depth coding based on the advanced video coding standard," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3449–3458, Sep. 2013.
[24] F. Cheng, J. Xiao, and T. Tillo, "3D video coding using motion information and depth map," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jun. 2015, pp. 1–6.
[25] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[26] M. A. Abidi and R. C. Gonzalez, *Data Fusion in Robotics and Machine Intelligence*. New York, NY, USA: Academic, 1992.
[27] J. C. Nearing, *Mathematical Tools for Physics*. New York, NY, USA: Dover, 2003.
[28] C. De Boor, "Bicubic spline interpolation," *J. Math. Phys.*, vol. 41, no. 3, pp. 212–218, 1962.
[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D slam systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst.*, Oct. 2012, pp. 573–580.
[30] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui, "Towards a simulation driven stereo vision system," in *Proc. IEEE 21st Int. Conf. Pattern Recog.*, Nov. 2012, pp. 1038–1042.
[31] X. Yun and E. R. Bachmann, "Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1216–1227, Dec. 2006.

Fei Cheng received the B.Eng. degree in electronic and information engineering from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2011, the M.Sc. degree in sustainable energy technology from the University of Liverpool, Liverpool, U.K., in 2014, and is currently working toward the Ph.D. degree at the University of Liverpool.

His current research interests include 3D video coding, image and video processing, computer vision, and new technology commercialization.

Tammam Tillo (S'03–M'05–SM'12) received the Engineer Diploma in electrical engineering from the University of Damascus, Damascus, Syria, in 1994, and the Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Torino, Italy, in 2005.

In 2004, he served as a visiting research student with Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland. From 2005 to 2008, he worked as a Postdoctoral Researcher with the Image Processing Laboratory, Politecnico di Torino. For few months he was an Invited Research Professor with the Digital Media Laboratory, Sungkyunkwan University, Seoul, South Korea. In 2008, he joined the Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China, where he established the multimedia technology laboratory. From 2010 to 2013, he was the Head of the Department of Electrical and Electronic Engineering, XJTLU University, and from 2012 to 2013, he was the Acting Head of the Department of Computer Science and Software Engineering, XJTLU University. He joined the Free University of Bozen-Bolzano, Bolzano, Italy, in 2017.

Jimin Xiao (M'15) received the B.S. and M.E. degrees in telecommunication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2007, respectively, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2013.

From 2013 to 2014, he was a Senior Researcher with the Department of Signal Processing, Tampere University of Technology, Tampere, Finland, and an External Researcher with the Nokia Research Center, Tampere, Finland. Since 2014, he has been a Faculty Member with the Xi'an Jiaotong-Liverpool University, Suzhou, China. His research interests include image and video processing, computer vision, and deep learning.

Byeungwoo Jeon (S'88–M'95–SM'02) received the B.S. (*Magna Cum Laude*) and M.S. degrees in electronics engineering from the Seoul National University, Seoul, Korea, in 1985 and 1987, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1992.

From 1993 to 1997, he was with the Signal Processing Laboratory, Samsung Electronics, Seoul, Korea, where he worked for research and development of video compression algorithms, design of digital broadcasting satellite receivers, and other MPEG-related research for multimedia applications. Since September 1997, he is with the faculty of the School of Electronic and Electrical Engineering, Sungkyunkwan University (SKKU), Seoul, South Korea, where he is currently a Professor. From March 2004 to February 2006, he served as a Project Manager of Digital TV and Broadcasting with the Korean Ministry of Information and Communications, Seoul, South Korea, where he supervised all digital TV-related R&D in Korea. From January 2015 to December 2016, he was the Dean of the College of Information and Communication Engineering, SKKU. His research interests include multimedia signal processing, video compression, statistical pattern recognition, and remote sensing.

Prof. Jeon is a Member of SPIE, and an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING. He was the recipient of the 2005 IEEK Haedong Paper Award in Signal Processing Society in Korea and the 2012 Special Service Award from the IEEE Broadcast Technology Society.