IMPROVED METHODS FOR OBJECT-BASED CODING OF PLENOPTIC VIDEOS

Qing Wu, Shing-Chow Chan
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
Email: qingwu@eee.hku.hk, scchan@eee.hku.hk

Heung-Yeung Shum*
* Microsoft Research, Asia
Beijing, P.R.China
Email: hshum@microsoft.com

ABSTRACT

Plenoptic videos (PVs) are a class of dynamic image-based representations, where the videos are taken at regularly spaced locations along a line. To yield the better rendering quality in scenes with large depth variations and support the functionalities at the object level for rendering, an object-based coding scheme is employed for the coding of PVs. Upon this object-based coding framework, the paper studies the improved coding methods for the texture and depth coding to achieve better compression efficiency. Experimental results show that considerable improvements in texture coding performance are obtained for both synthetic and real scenes. The improved depth coding quality is also illustrated.

1. INTRODUCTION

Image-based rendering (IBR) is a promising technology for photo-realistic rendering of scenes and objects from a collection of densely sampled images or videos. Since the data size of image-based representations is usually very large, especially for dynamic scenes, efficient methods for capturing, storing and transmission of image-based representation are active areas of research [1]. Central to IBR is the plenoptic function [2], which is a 7-dimensional (7D) function. Two important 4D static representations are the Light fields [3] or lumigraphs [4] (lumigraph differs from light fields in using additional depth information), where densely sampled images taken on a 2D camera plane are used to render novel views or intermediate views. To further avoid the large dimensionality and the excessive hardware cost in capturing dynamic representations, a class of dynamic image-based representations called the simplified dynamic light field (SDLF) or plenoptic videos is proposed in [5, 6]. The plenoptic video is also a 4D plenoptic function and it is obtained by constraining the users' viewpoints along a line (or line segments) instead of a 2D plane in static light fields. Despite this simplification employed, plenoptic videos can still provide a continuum of viewpoints, significant parallax and lighting changes along line segments joining the camera arrays. Plenoptic videos are also closely related to multiview video sequences [7, 8, 1]. However, plenoptic videos usually rely on denser sampling in regular geometric configurations to improve the rendering quality. In addition, the random access to individual pixels in the compressed data stream, so-called the random access problem in IBR, becomes very important in real-time rendering.

In our previous work [9], we develop an object-based compression system for plenoptic videos in order to facilitate its rendering, transmission and storage. The main advantages of using the object-based representation are: 1) by properly segmenting IBR into objects at different depths, it has been shown that the rendering quality in large environment can be significantly improved [10]; 2) by coding the plenoptic video at the object level, desirable functionalities such as scalability of contents, error resilience, and interactivity with individual IBR objects (including random access at the object level), etc, can be achieved. The first one is particularly important because, according to plenoptic sampling [11], scenes with large depth variations will require extremely high sampling rate to overcome the rendering artifacts such as ghosting and blurring around depth discontinuities. Such high sampling is usually impractical. Using layers with different depth values, the adverse effect of depth discontinuities can be effectively mitigated during rendering. The object-based coding scheme shares many useful concepts with the MPEG-4 standard. However, in plenoptic videos or in general IBR, additional geometry information such as the depth map considered here has to be incorporated to facilitate rendering of the IBR objects. In addition, this coding scheme exploits both the temporal and spatial redundancy among the video streams in the plenoptic video for better compression efficiency. In the generic structure of our object-based codec, different VOs in the scene may be encoded separately. Each VO contains the VOPs of a certain object in the plenoptic video stream, which are described by its texture, shape and depth information. The scene and VO/VOP descriptors for the plenoptic video are also encoded and multiplexed together with the VOPs, which are used to compose the video scenes at the decoder for display or rendering.

In this paper, we propose the improved coding methods for the texture coding and depth coding to achieve better compression efficiency. Firstly, by means of depth information, an initial global motion vector (MV) is obtained to facilitate the motion estimation process for spatial prediction. Secondly, direct prediction mode and four MVs for a MB, two useful tools provided by MPEG-4, are developed and applied in the motion prediction for the texture coding of PVs. Finally, the independent motion prediction is applied for the depth coding to improve its coding quality. Experimental results show that considerable improvements in coding performance can be obtained by using improved methods.

The paper is organized as follows. The review of object-based coding scheme for PVs is presented in section II. Section III introduces the improved methods for the texture and depth coding under the object-based coding framework. Experimental results are shown in section IV and finally, conclusions are given in section V.

2. OBJECT-BASED CODING SCHEME FOR PLENOPTIC VIDEOS

For a review of our object-based coding scheme proposed in [9], Fig. 5 shows the texture coding of an IBR object in a PV. The basic idea is a generalization of our previous frame-based method in [6], except that now video objects with arbitrary shapes rather than images with fixed size are encoded. Only three video object (VO) streams are shown for simplicity, and we call them a group of video object field (GOVOF). In each VO stream, we have a view of the IBR object, which we shall refer to as the video object plane (VOP). There are two types of VO streams associated with each dynamic IBR object: main and secondary video object streams. Main VO streams are encoded similar to the MPEG-4 algorithm, which can be decoded without reference to other VO streams. For better performance, we also allow bi-directional prediction. To provide random access to individual VOP, we follow the structure of Group of VOP (GVOP) of MPEG-4 and employ it in the main stream. A GVOP contains an I-VOP and possibly P-VOPs and/or B-VOPs between this I-VOP and the following I-VOPs. I-VOPs are coded using intra-frame coding to provide random access point, while P-VOPs are coded with reference to previous I/P-VOPs, and B-VOPs are coded by performing bi-directional prediction. The VOPs

0-7803-9266-3/05/$20.00 ©2005 IEEE.

captured at the same time instant as the I-VOP in a main stream constitute an I-VOP field. Similarly, we can define the corresponding P- and B-VOP fields. A secondary I-VOP is encoded using a disparity-compensated prediction, or called "spatial prediction", from the reference I-VOP in the same I-VOP field. The secondary P/B-VOPs also employ spatial prediction from their adjacent P/B-VOPs in the main stream for better performance.

Apart from texture pictures and binary alpha maps for shape coding, each VO may also contain depth maps and grayscale alpha maps for rendering the image-based objects. The latter is useful in matting VOs during VO composition and rendering. The binary alpha map describes the shape of an object and it is encoded using context-based arithmetic encoding (CAE) algorithm. Both grayscale alpha-maps and depth maps are coded in the same way as the luminance component of texture.

3. IMPROVED CODING METHODS

3.1 Initial Global MV

The disparity between a secondary stream and main stream might be very large. To avoid the huge computational overhead and reduce the mismatching probability of spatial prediction resulted from large search range due to the big disparity, we use an initial global MV to compensate the disparity. The initial global MV is used to preliminarily find out the central point of search area in the reference VOP of main stream. Then the spatial prediction is more efficiently performed to seek the local MV in a reasonable search range. The final MV for that MB is the summation of two MVs.

Depth maps of the VOPs can be exploited to accurately get the initial global MVs. Using the pinhole camera model depicted in Fig. 4, where c_m and c_s are focal centers of cameras respectively for the main stream and the secondary stream, then the disparity length d for a visible point of the object is given by:

$$d = d_s - d_m = f \cdot B / Z \tag{1}$$

d_s and d_m are the distances from the image point to the focal centers of the main and secondary streams, respectively. f is the focal length of the camera, B is the baseline distance between two cameras, and Z is the depth value for that visible point at the object surface. Calculating the mean depth value of all the points in the object surface, Z_a, to substitute Z in Eq. (1), the initial global MV for a secondary VOP, $V_g = (v_x, v_y)$, can be finally given by:

$$v_x = f \cdot B / Z_a, \qquad v_y = 0 \tag{2}$$

Relative to the whole scene of a frame, a VO usually has a flatter surface, thus the initial global object MV is precise enough to seek the local MV.

3.2 Direct Motion Prediction Mode And Its Extensions

Following MPEG-4, our object-based coding scheme employs a new motion prediction mode known as direct motion prediction mode [12], briefly called *direct mode*, for the texture coding. It uses direct bi-directional motion compensation derived by exploiting the MVs of the MBs in previously coded P-VOPs and scaling them to yield forward and backward MVs for the MBs of B-VOPs. The extensions of the direct mode are developed to achieve higher compression efficiency by exploiting the temporal and spatial correlation between a secondary stream and the main stream.

3.2.1. Direct prediction mode

The MBs coded using the direct mode are called direct MBs. They can exist within the B-VOPs preceding a P-VOP in the main stream coded by MPEG-4 (Fig. 1 (a)) and the secondary B-VOPs (Fig. 1 (c)). We now take the former case as an example to depict how to code a direct MB using direct mode, as shown in Fig. 1 (a).

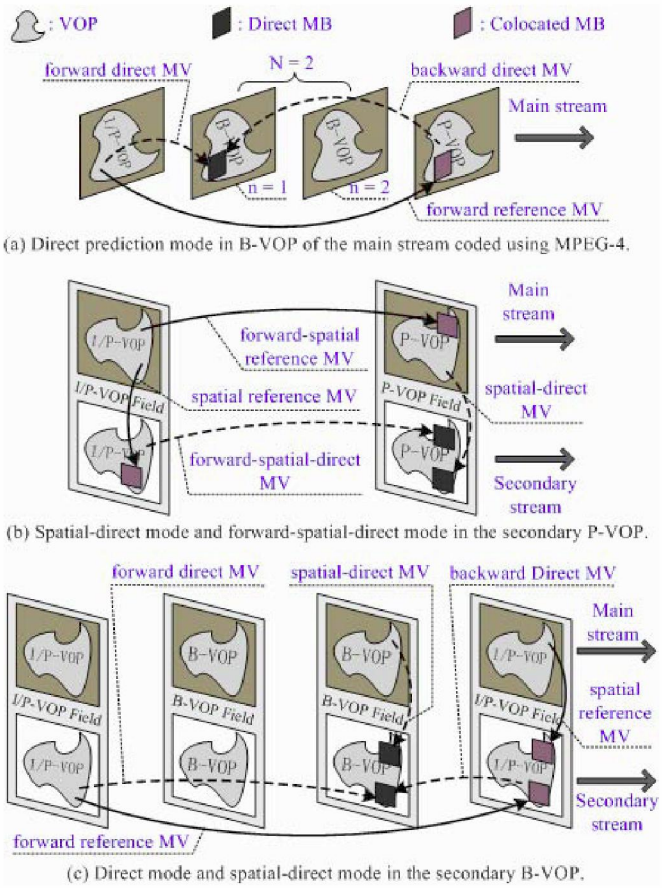


Fig. 1. Direct motion prediction mode and its extensions

Assuming the VO moves continuously, the colocated MBs in the reference VOPs have the same motion tracks as a direct MB to be coded. Then bi-directional MVs for a direct MB can be directly obtained from the available MV (forward reference MV) of the colocated MB in the previously coded P-VOP. The bi-directional compensation is implemented later on for that direct MB relying on two obtained MVs. The direct MVs for a direct MB can be produced by the following equations:

$$V_f = n/(N+1) \cdot V_{ref}, \qquad V_b = n/(N+1) \cdot V_{ref} \tag{3}$$

where n is the index value of a B-VOP in a set of successive B-VOPs between the nearest preceding I-VOP/P-VOP and the following P-VOP, N is the total B-VOP number of the set, V_f is the forward direct MV of direct MB, V_b is the backward direct MV of the direct MB, and V_{ref} is the forward reference MV. To make the prediction more accurately, we employ four MVs for the forward reference MV, each corresponding to an 8×8 block in the colocated MB. Those corresponding direct MVs can be still straightforwardly produced using Eq. (3), and then adjusted by adding a common small delta MV. The value of the delta MV can be found out by testing whether the final resulting direct MVs can achieve the minimum value of sum of absolute difference (SAD) between the MB to be coded and two reference MBs. Since the initial direct MVs can be obtained by Eq. (3) at the decoder, only the common small delta MV needs to be encoded, instead of encoding all the final direct MVs. Thus, the compression efficiency will be improved when using the direct mode, especially in case of applying four MVs at the same time.

3.2.2. Spatial-direct prediction mode

As depicted in Fig. 1 (b), the spatial-direct MBs in a secondary P-VOP are coded using the spatial-direct motion prediction mode. The spatial reference MVs come from the colocated MBs of the nearest preceding I/P-VOP in the same secondary stream. Another case of spatial-direct mode is shown in Fig. 1 (c), where the spatial-direct MBs are in the secondary B-VOP. Its spatial reference MVs come from the colocated MBs of the nearest future I/P-VOP in the same secondary stream. In both the cases, the colocated MBs are coded using the motion prediction with reference to the I/P-VOPs of the main stream in corresponding I/P-VOP fields. The spatial-direct mode is based on the fact that the disparity between the secondary VOP and its corresponding VOP in the main stream usually varies little within a short time interval. The spatial-direct MV only includes the spatial MV V_s, which is directly equal to the reference MV V_{ref}. The initial direct MVs are also adjusted by adding a common small delta MV. The benefit of applying the spatial-direct mode for a MB in a secondary P-VOP/B-VOP is the same as a direct MB introduced previously, especially in case of applying four MVs for a reference MV at the same time.

3.2.3. Forward-spatial-direct prediction mode

Since all VOPs in a same VOP-Field are captured at the same time instance, the VOs of secondary streams have the motions similar to those of main stream. Based on this fact, another extension to the direct mode, the forward-spatial-direct mode depicted in Fig. 1 (b), is developed to code some MBs in the secondary P-VOPs. The reference MVs come from the colocated MBs of the P-VOP in the main stream, thereby called forward-spatial reference MVs. Like spatial-direct MVs, the forward-spatial-direct MV only contains the forward MV V_f, which equal to V_{ref}, the reference MV. A common delta MV is still needed. Applying forward-spatial-direct mode can also gain a great benefit due to the same reasons as those in the direct mode and spatial-direct mode.

3.3 Improved Method For The Depth Coding

Usually depth maps can be coded well in the same way as the texture luminance component using the common MVs. However, in a dynamic lighting environment, the MVs for the texture might be very different from those for depth maps, since the light can rarely influence the depth values. To address this problem, independent motion prediction is also incorporated to find out the independent MVs for the depth maps. A better coding result is selected from one using common MVs or one using independent MVs, respectively. The independent MVs for the depth maps would result in an extra overhead, however, it would be paid off by the better prediction for depth maps in such case as a dynamic lighting environment.

4. EXPERIMENTAL RESULTS

For the sake of fairness, we still use two plenoptic videos (PVs) same as used in [9] to evaluate the performance improvements achieved by the improved methods. The synthetic PVs (called "synthesis") and the real-scene PVs (called "dancer") respectively have the resolution of 320×240 pixels and 720×576 pixels both in 24-bit RGB format, and with the amount of 240 frames. The real-scene PVs were captured by our multiple video cameras system, as shown in Fig. 2, which consists of two linear arrays of cameras each hosting 6 JVC DR-DVP9$_{AH}$ video cameras. The distance between two adjacent cameras is 15 cm, and the angle between the arrays can be flexibly adjusted. The depth maps are generated with 16 bits per pixel. Fig. 3 shows a few snapshots of two PVs and two IBR objects objects extracted from them – the rotating and moving ball and the dancer. A semi-automatic segmentation method called "lazy snapping" [13] is used to perform the segmentation for real-scene IBR object "dancer". Due to the space limitation, snapshots for only

Fig. 2. Configuration of our multiple video cameras system.

3 streams are shown in Fig. 3, despite the "ball" and "dancer" contain 9 and 6 streams, respectively. Fig. 7 shows the combined coding results with respect to PSNR in texture and shape coding for IBR objects "ball" and "dance" at different bit rates achieved by using VM [14] rate control algorithm. The frame rates used for the PVs are 24 frames per second. For illustration, a Group of VOPs (GVOP) structure consisting of 12 VOPs (1 I-VOP, 3 P-VOPs and 8 B-VOPs) is employed. The curves denoted by "MPEG-4" represent the results for the individual streams using MPEG-4 algorithm without spatial prediction. The curves marked by "SP-3" and "SP-5" respectively indicate the coding results using the coding methods described in [9] (reviewed in section II and we call them as "basic methods") with 3 and 5 VO streams within a GOVOF. The curves marked by "Improved-SP-3" and "Improved-SP-5" denote the results using the proposed improved coding methods. For the synthetic IBR object "ball", it can be seen that the PSNR performance achieved by improved methods is much better (1 dB) than that using basic methods, though both have considerable improvements (5 and 4 dB) over applying MPEG-4 in coding individual VO streams. The performance improvements for the real-scene IBR object "dancer" using improved methods are also better than using basic methods, though both look less significant compared with the synthetic sequence. This is mainly due to the slight position errors introduced by imperfect camera calibration, which destroys somewhat the correlation between the video streams.

Fig. 6 (a) and (b) shows the texture image and the depth map of a VOP coming from a secondary VO stream in the synthetic PV "ball" with a dynamic lighting. The VO within the PV may be considered as the background of the VO "ball", consisting of a slowly moving object (the "hose") and other 4 static objects. Fig. 7 (c) and (d) shows 2 reconstructed depth maps of that VOP coded at the same compression ratio of 850 using common and independent MVs, respectively. We can see that the one using independent MVs is much better than that using common MVs, especially around the object boundaries.

5. CONCLUSION

Based on an object-based coding framework for plenoptic videos, the improved methods for the texture and depth coding are presented. By exploiting depth information and developing several tools provided by MPEG-4, the improved methods for the texture coding are proposed to achieve higher coding efficiency. The independent motion prediction applied for the depth coding is also introduced. Experimental results show that the considerable improvements in coding performance can be achieved by using the improved methods.

REFERENCES

[1] H.Y. Shum, S.B. Kang and S.C. Chan, "Survey of Image-Based Representations and Compression Techniques," *IEEE Trans. Circuits and System for Video Technology*, vol. 13, pp. 1020-1037, Nov. 2003.

[2] E. H. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, pp. 3-20, MIT Press, Cambridge, MA, 1991.

[3] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Of SIGGRAPH'96*, pp. 31-42, Aug. 1996.

[4] S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH'96*, pp. 43-54, Aug. 1996.
[5] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan and H. Y. Shum, "The plenoptic videos: capturing, rendering and compression," in *Proc. of IEEE ISCAS'04*, vol. 3, pp. 905-908, May 23-26, 2004.
[6] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan and H.Y. Shum, "The Compression of Simplified Dynamic Light Fields," in *Proc. of IEEE ICASSP'03*, vol. 3, pp. 653-656, Hong Kong, Apr. 2003.
[7] M. G. Strintzis and S. Malasiotis, "Object-based coding of stereoscopic and 3D image sequences: A review," *IEEE Signal Processing Mag.*, vol. 16, pp. 14–28, May 1999.
[8] M. E. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proc. of IEEE ICASSP'86*, pp. 521–524, 1986.
[9] Q. Wu, K.T. Ng, S.C. Chan, and H.Y. Shum, "On Object-based Compression For A Class Of Dynamic Image-Based Representations", to appear in *Proc. ICIP'2005*.
[10] H. Y. Shum, J. Sun, S. Yamazaki, Y. Li and C. K. Tang, "Pop-up Light Field: An Interactive Image-based Modeling and Rendering System," *ACM Trans. on Graphics,* vol. 23, issue 2, pp. 143 -162, April 2004.
[11] J.X. Chai, X. Tong, S.C. Chan and H.Y. Shum, "Plenoptic sampling," in *Proc. of SIGGRAPH'00*, pp. 307–318, July 2000.
[12] ITU-T Recommendation ISO/IEC 14496-2:2001, "Information Technology- Coding of audio-visual objects -- Part 2: Visual".
[13] Y. Li, J. Sun, C. K. Tang and H. Y. Shum, "Lazy snapping," in *Proc.in SIGGRAPH'04*, pp.303-308, 2004.
[14] MPEG-4 video verification model v18.0, ISO/IEC JTC1/SC29/ WG11 Coding of Moving Pictures and Audio N3908, Pisa, Jan. 2001.

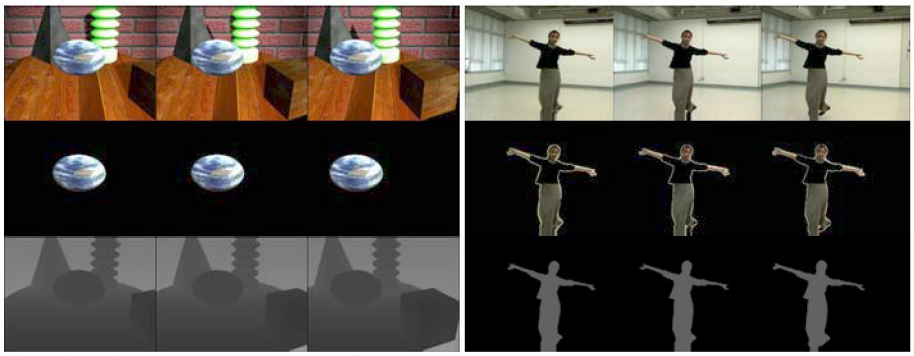


Fig. 3. Snapshots of (Top) synthetic PV "synthesis" and real-scene PV "dancer"; (Middle) the IBR objects "ball" and "dancer"; (Bottom) the depth maps of the synthetic PVs "ball" and the IBR object "dancer".

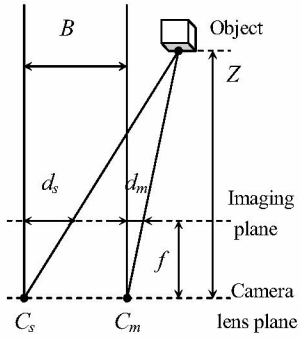


Fig. 4. Disparity calculation for a point of an object using pinhole camera model.

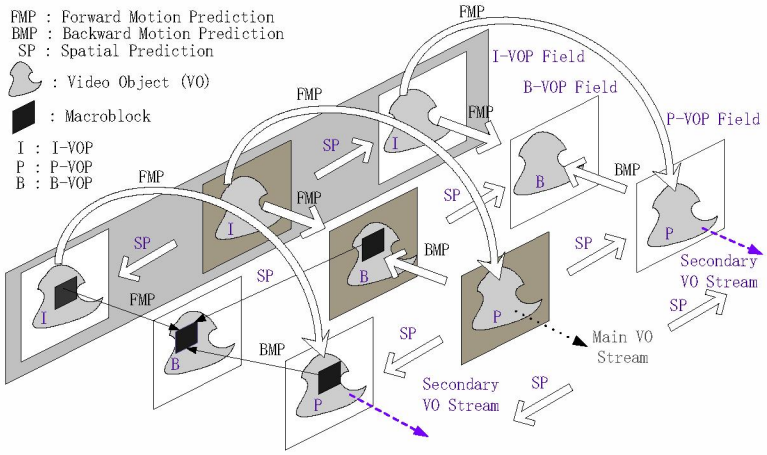
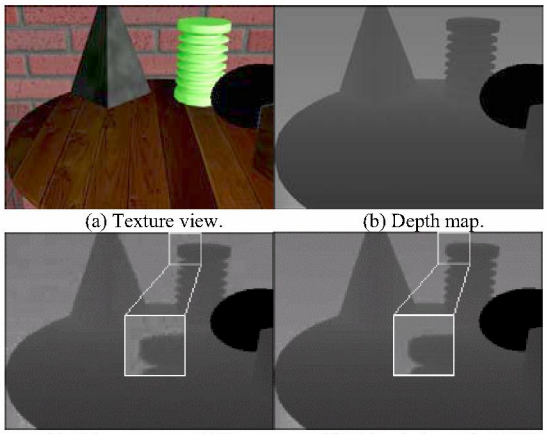
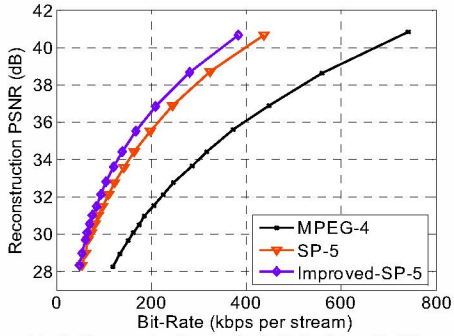


Fig. 5. Object-based coding scheme for the plenoptic video using basic temporal and spatial predictions.

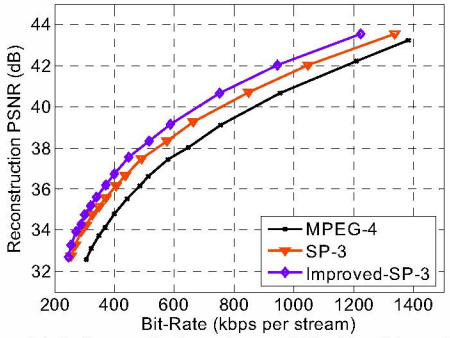


(a) Texture view. (b) Depth map.
(c) Using common MVs. (d) Using independent MVs

Fig. 6. An example of depth coding results for a VOP in a dynamic lighting environment



(a) Coding results for synthetic IBR object "ball"



(b) Coding results for real-scene IBR object "dancer"

Fig .7 Coding performance comparison of using improved methods and basic methods for synthetic and real-scene IBR objects