An object-based compression system for a class of dynamic image-based representations

Qing Wu*[a], King-To Ng*[a], Shing-Chow Chan*[a], Heung-Yeung Shum[#b]
[a]Dept. of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, HK;
[b]Microsoft Research, Asia Beijing, P.R.China.

ABSTRACT

This paper proposes a new object-based compression system for a class of dynamic image-based representations called plenoptic videos (PVs). PVs are simplified dynamic light fields, where the videos are taken at regularly spaced locations along line segments instead of a 2-D plane. The proposed system employs an object-based approach, where objects at different depth values are segmented to improve the rendering quality as in the pop-up light fields. Furthermore, by coding the plenoptic video at the object level, desirable functionalities such as scalability of contents, error resilience, and interactivity with individual IBR objects can be achieved. Besides supporting the coding of the texture and binary shape maps for IBR objects with arbitrary shapes, the proposed system also supports the coding of gray-scale alpha maps as well as geometry information in the form of depth maps to respectively facilitate the matting and rendering of the IBR objects. To improve the coding performance, the proposed compression system exploits both the temporal redundancy and spatial redundancy among the video object streams in the PV by employing disparity-compensated prediction or spatial prediction in its texture, shape and depth coding processes. To demonstrate the principle and effectiveness of the proposed system, a multiple video camera system was built and experimental results show that considerable improvements in coding performance are obtained for both synthetic scene and real scene, while supporting the stated object-based functionalities.

Keywords: Image-based rending (IBR), dynamic image-based representations, object-based compression, MPEG-4, plenoptic videos, IBR objects

1. INTRODUCTION

IBR is a promising approach for the photo-realistic rendering of scenes and objects from a collection of densely sampled images. Since the data size associated with the image-based representations is usually very large, especially in the case of dynamic scenes, efficient methods for its capturing, storage and transmission are active fields of research[1]. Different image-based representations have been proposed to simplify the capturing process and storage requirements. For a recent survey of IBR, readers are referred to [1] for more details. In a previous work[2], the authors have developed a multiple cameras system for capturing a class of dynamic image-based representation called "plenoptic videos" (PVs). The PV is a simplified light field for dynamic environments so that users' viewpoints can be selected on the camera plane of a linear video camera array to simplify the hardware requirement and capturing process. Using a parallel processing-based system, high-quality rendering of dynamic image-based representations by means of off-the-shelf equipment were obtained. And also, its potential applications in visualization and immersive television systems were demonstrated. The plenoptic videos are also closely related to multiview video sequences[3, 4, 1]. However, plenoptic videos usually rely on denser sampling in regular geometric configurations in order to improve the rendering quality. Furthermore, the random access to individual pixels inside the compressed data stream, so-called the random access problem in IBR, becomes a very important issue in real-time rendering.

Undoubtedly, efficient compression approaches play a key role in storing and compression of dynamic image-based representations due to their tremendous data sizes. In this paper, we study the object-based compression for plenoptic videos 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

* qingwu@eee.hku.hk; ktng@eee.hku.hk; scchan@eee.hku.hk;
# hshum@microsoft.com

Visual Communications and Image Processing 2005, edited by Shipeng Li, Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proc. of SPIE Vol. 5960 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.632666

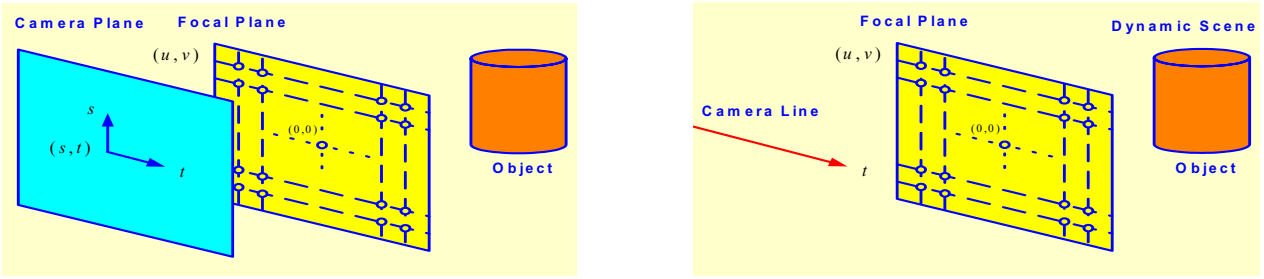
quality in large environment can be significantly improved[5]; 2) by coding plenoptic videos at the object level, desirable functionalities such as scalability of contents, error resilience, and interactivity with individual IBR objects (including random access at the object level), etc, can be achieved. For instance, a compressed image-based object can be transmitted at a different rate and composed to different plenoptic videos at the receiver, as will be shown in Fig. 3. The first advantage, on the other hand, is the consequence of the plenoptic sampling[6], which reveals that the spectral support of a light field is dependent on the depth values of the objects in the scene, when there are no occlusions or depth discontinuities. However, scenes with large depth variations will require extremely high sampling rate to overcome the rendering artifacts such as ghosting and blurring around depth discontinuities. One effective approach to yield better rendering results is to segment the scene into depth layers so that the adverse effect of depth discontinuities can be mitigated. The idea has been demonstrated in the "pop-up light fields" [5], where excellent rendering quality could be achieved if the light field is properly segmented into layers of different depth values.

To improve the rendering quality and enable the object-based functionalities mentioned above, we propose an object-based compression system for plenoptic videos to encode the texture, shape, grayscale alpha map (for matting) and depth information of each IBR object together. This compression scheme may be viewed as a generalization of our previous frame-based compression technique[7] for plenoptic videos, except that now arbitrarily shaped video objects rather than images with fixed size are encoded. The proposed coding scheme also shares many useful concepts with the MPEG-4[8] video coding standard. The major difference between the two coding schemes is that: the IBR objects in plenoptic videos, and in general IBR compression, have to incorporate other important information such as additional geometry information in the form of depth maps and alpha maps, etc, to facilitate their renderings. In addition, the proposed compression scheme exploits both the temporal redundancy and spatial redundancy among video streams in plenoptic videos to achieve better compression efficiency. As a result, under the proposed object-based framework, multiple video streams in a plenoptic video can be encoded into user-defined IBR objects, and flexibly reconstructed at the decoder for display and rendering at either the object level or frame level. To demonstrate the principle and effectiveness of the proposed system, a multiple video camera system was built and experimental results show that considerable improvements in coding performance are obtained for both synthetic scene and real scenes, while supporting the stated object-based functionalities.

The rest of the paper is organized as follows. Section 2 briefly reviews the concept of plenoptic function and the plenoptic videos. The proposed object-based compression system is described in Section 3. Experimental results are shown in Section 4 and finally, conclusions are given in Section 5.

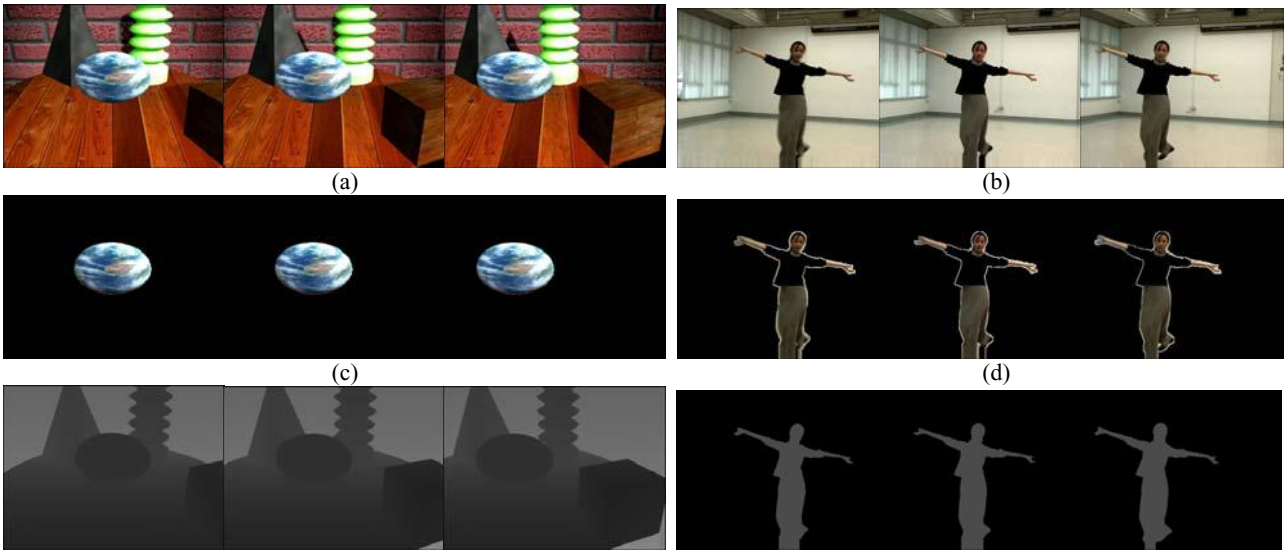
2. THE PLENOPTIC VIDEOS —— A CLASS OF DYNAMIC IMAGE-BASED REPRESENTATIONS

Adelson and Bergen[9] first proposed the 7-dimensional plenoptic function, $P_7 = (V_x, V_y, V_z, \theta, \phi, \lambda, \tau)$, to describe all the radiant energy that is perceived at any 3-D viewing point (V_x, V_y, V_z), from every possible angle (θ, ϕ) for every wavelength λ, and at any time τ in the case of dynamic scenes. Based on this function, theoretically, the novel views at different positions and at different time from its samples can be reconstructed, provided that the sample rate is sufficiently high. Due to its high dimensional nature, data reduction (compression), transmission and efficient rendering of the plenoptic function are essential to IBR systems. One approach is to restrict the viewing freedom of the users so that the dimension of the IBR can be reduced. Light fields[10] or lumigraph[11] (lumigraph differs from light fields in using additional depth information) are two important types of four-dimensional image-based representations, where images on a camera plane are taken to render novel views of the scene. Fig. 1(a) illustrates the principle of light field or lumigraph. For dynamic light fields, the number of videos required to be captured on a 2D plane is usually very large. To avoid such a large dimensionality and the excessive hardware cost, a kind of simplified dynamic light field[2, 7] (SDLF) with viewpoints being constrained along a line instead of a 2D plane, as illustrated in Fig. 1(b), was proposed. Because of the close relationship between the SDLF with traditional videos, we also referred it to as "plenoptic videos". Besides the simplicity of the overall system, significant parallax and lighting changes along the horizontal direction can also be observed plenoptic videos. On the other hand, the given number of cameras can be used to maximize the sampling rate along the horizontal direction and thus reducing the risk of insufficient sampling in a 2D configuration with the same number of cameras and horizontal panning range.



(a) (b)

Fig. 1. (a) 4D static light fields: viewpoints constrained on a 2D plane; (b) 4D simplified dynamic light fields (the plenoptic video) viewpoints constrained along a line in a dynamic environment.



(a) (b)

(c) (d)

(e) (f)

Fig. 2. Snapshots of (Top) (a) synthetic PV "synthesis" and (b) real-scene PV "dancer"; (Middle) the IBR objects (c) "ball" and (d) "dancer" extracted; (Bottom) the depth maps of the synthetic PV (e) "synthesis" and (f) the IBR object "dancer".

Fig. 2 shows several snapshots from two plenoptic videos and IBR objects segmented from the scenes. At the left side of Fig. 2 is a synthetic sequence called "synthesis", while the right side of Fig. 2 is a real-scene PV called "dancer". The "ball" and the "dancer" in the scenes are segmented to form two IBR objects as shown in Figs. 2(c) and (d). A semi-automatic segmentation method called "lazy snapping"[12] is used to perform the segmentation. In the next section, the proposed object-based compression system for the coding of these IBR objects will be introduced.

3. PROPOSED OBJECT-BASED CODING SYSTEM FOR PLENOPTIC VIDEOS

3.1 System overview

Once the IBR objects have been identified, defined, and then extracted from the plenoptic video (e.g., the objects "ball" from the PV "synthesis"), they can be compressed individually to provide functionalities such as scalability of contents, error resilience, and interactivity with individual IBR objects. For example, different IBR objects might be given different numbers of bits (and different amounts of channel coding) and hence different reconstruction qualities (error resilience). They might also be transmitted at different frame rates to achieve object scalability. Fig. 3 shows the generic codec structure of our object-based coding system, which shares much useful concept with the MPEG-4 video object coding. A video object (VO) includes the video object planes (VOPs) distributed in all the streams involved in the plenoptic video, each containing its corresponding binary shape mask, grayscale shape map (alpha map) and depth map. Each VOP is then encoded based on its shape and motion. The scene and VO/VOP descriptors for the plenoptic

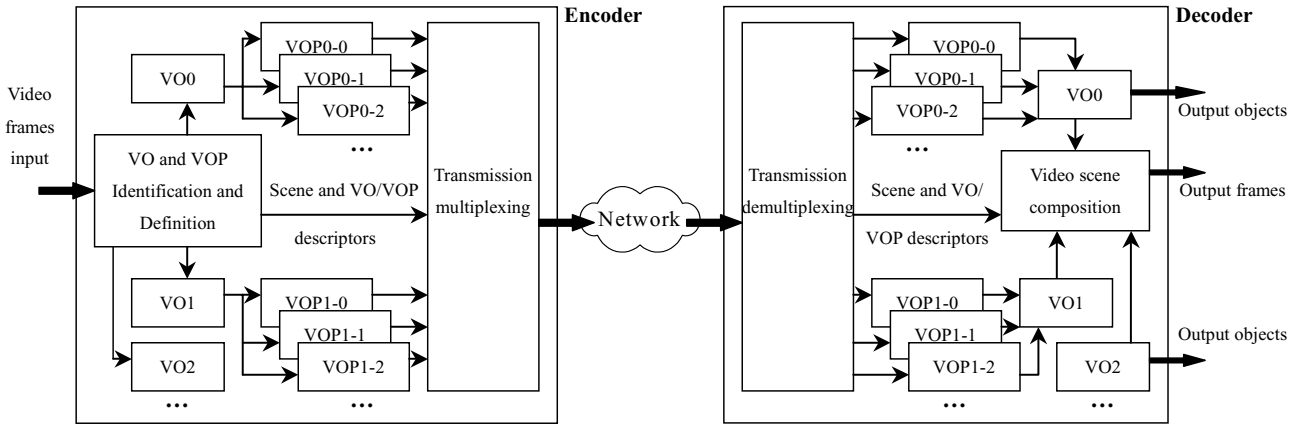


Fig. 3. Generic codec structure of the proposed object-based compression system for the plenoptic video.

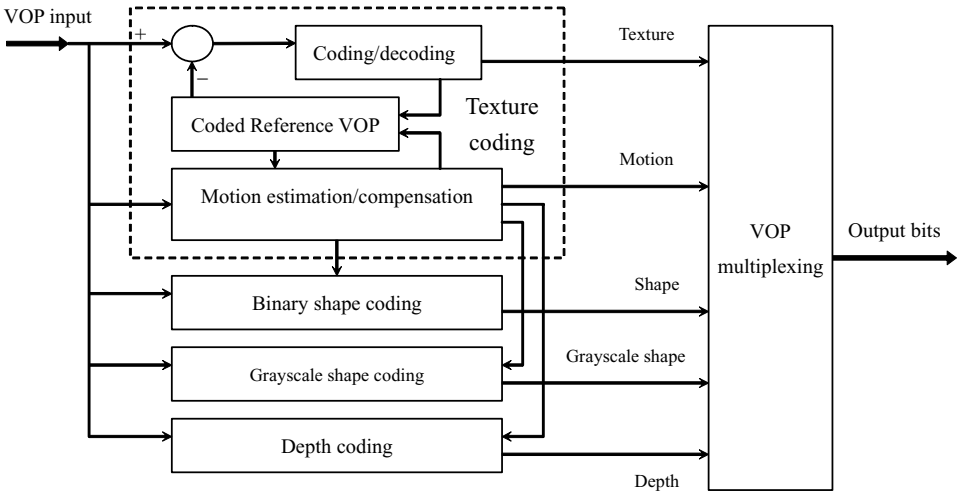


Fig. 4. Diagram of encoding a VOP.

videos are also encoded and multiplexed together with the VOPs, which are used to compose the video scenes at the decoder. Via the channels of the networks, the decoder can demultiplex and decode the VOPs for display or rendering. Of course, the reconstructed VOPs can also be further composed into a frame for presentation and other operations.

Fig. 4 shows the encoder diagram of a VOP in an IBR object. It consists of four major components: texture coding, binary shape coding, grayscale shape coding and depth map coding. Texture coding is performed using Discrete Cosine Transform (DCT) based on motion prediction and compensation. The binary shape mask of the VOP is encoded using context-based arithmetic encoding (CAE) algorithm[13]. Grayscale shape information (alpha map), defined by an eight-bits number, is useful in matting VOs during VO composition and rendering at the decoder. Following MPEG-4, grayscale shape information (alpha map) is coded through alpha channels in the same way as the luminance signal of texture. Depth map, as a type of geometrical information, is encoded independently as a so-called "depth channel" in the object-based coding system. After these four parts are encoded, they are then multiplexed together as an entire encoded VOP. In the following, we will present the details of the texture coding, binary shape coding and depth coding of VOPs.

3.2 Texture coding

Because of disparity, adjacent light field images appear to be shifted relative to each other. Therefore, it is advantageous to employ both temporal or spatial predictions (also referred to as disparity compensated prediction (DCP)) to improve the coding efficiency as in traditional stereoscopic image coding[4].

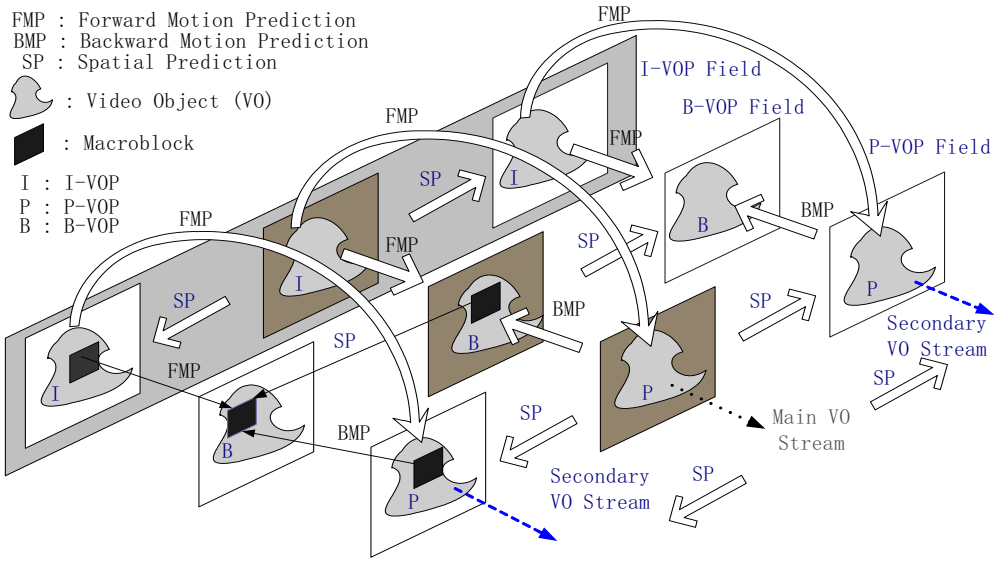


Fig. 5. Proposed method for the texture coding of an IBR object in the plenoptic video.

As depicted in Fig. 5, the proposed method employs predictions in both the temporal and spatial directions for coding the texture of an IBR object in the plenoptic videos. For simplicity only three video object (VO) streams are shown, 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 refer to as the video object plane (VOP) as mentioned previously. There are two types of VO streams associated with each dynamic IBR object: main video object stream and secondary video object stream. Main VO stream is 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 for the B-VOPs. 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 VO 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 without reference to any other VOPs, while P-VOPs are coded by motion predictive coding using previous I- or P-VOPs as references. B-VOPs are coded by a similar method except that forward and backward motion compensations are performed by using nearby I- or P-VOPs as references, which are indicated by the block arrow in Fig. 5. The VOPs captured at the same time instant as the I-VOP in a main stream constitute an I-VOP field. Similarly, we define the P- and B-VOP fields as the VOP field containing respectively the P- and B-VOPs of the main VO stream. A VOP from the secondary stream in the I-VOP field are encoded using disparity-compensated prediction (DCP) or "spatial prediction" from the reference I-VOP in the I-VOP field. It is because adjacent light field images appear to be shifted relative to each other, similar to the effect of linear motion in video coding. As mentioned earlier, the disparity is actually the displacement of pixels, which is related to the objects and viewing geometries. The disparity-compensated prediction has been used in the coding of static light fields. Therefore, the coding algorithm considered here can be viewed as their generalization to the dynamic IBR object context. Similarly, apart from using temporal prediction in the same stream, the secondary P/B-VOPs also employ spatial prediction from their adjacent P/B-VOPs in the main stream for better performance.

It can be seen that employing spatial prediction for coding the secondary VO streams can achieve a better prediction in comparison to the main stream, which only employs temporal prediction, especially for VOs having fast motions. However, introducing spatial prediction also increase the overhead used in selecting the prediction modes, since one more prediction mode will result in one more entry in the codewords table for the entropy coding of MB prediction modes. According to the occurrence probabilities of MB prediction modes, we construct two codewords tables of prediction modes respectively for secondary P-VOPs and secondary B-VOPs, while keeping unchanged the codewords tables for the B-VOPs of main stream. There are no codeword tables needed for the P-VOPs of main steam and the secondary I-VOPs, since they can only have two coding mode selections, that is, either intra mode or predictive mode. Experimental results show that the extra overheads of the new prediction modes is paid off by the better predictions offered by spatial prediction. Finally we need to mention that the blocks which lie within the object are

coded similar to traditional video coders, while blocks at the boundary of an object can either be coded using padding or shape-adaptive DCT.

3.3 Shape coding

Shape information is an important component in object-based coding. Following MPEG-4, there are two types of shape information: binary shape information and grayscale shape information. The former only provides the binary shape mask collocated with the luminance picture of the VOP, used to indicate whether the pixels belong to that VOP or not. The latter one also called alpha map, provides pixels' transparency levels for a VOP, which are useful in matting VOs during composition and rendering after being decoded. Binary shape information generally can be coded using Context-based Arithmetic Encoding (CAE) algorithm[13]. As discussed in previous subsection, grayscale shape information is coded by using DCT similar to coding luminance signal via the alpha channel. Hence, shape coding mainly refers to coding binary shape information. CAE algorithm includes two types: Intra-CAE and Inter-CAE. Intra-CAE codes shape information in intra mode, without using motion prediction, and therefore mainly used for I-VOP in the main stream. In contrast, inter-CAE makes use of motion prediction from a shape mask reference, and therefore used in other types of VOPs except I-VOP. In the main stream coded by MPEG-4, for a B-VOP, inter-CAE selects a shape mask in the nearest preceding I-VOP/P-VOP or future I-VOP/P-VOP as the reference to do the shape motion prediction and compensation.

For the shape coding of a VOP in a secondary stream, it is possible to select the reference from either a VOP in this secondary stream or another in the main stream at the same time instant. In general, the shapes of the VOP in secondary streams are very similar to the VOP of the main stream at the same GOVOF, because they are captured by two cameras at the same time instance. As a result, selecting the VOP of the main stream as reference usually performs better than selecting the VOP in the same secondary stream. However, if the object is static, or moves very slowly, the shape motion prediction performed in the same secondary stream (i.e., *intra stream mode*), can achieve a better result than that performed between the secondary stream and the main stream (i.e., *inter stream mode*). To achieve a better shape coding result, both modes are incorporated. They are selected by performing the shape coding for each VOP in both modes, and the better one will be chosen. This method is referred to as the *hybrid mode*, and its improvement will be illustrated in section 4.

3.4 Depth coding

To more precisely render the novel views for a VO/video stream in the plenoptic video, depth information of the steams is often exploited. The more accurate the depth information is, the better the rendering quality will be. In MPEG-4, alpha channels are provided to encode a set of grayscale shape information in the same way as the luminance component of textures. As a result, the coded grayscale shape information also forms parts of the final coded bitstream.

The depth map of a VO resembles closely the alpha information of a VO. Hence, it can be coded in a similar way as the alpha map, except for some pre-processing to be described below. Similar to "alpha channel", the data in the final encoded bitstream for storing the encoded depth map are called the "depth channel". We now described the pre-processing of the depth map for better coding performance. Firstly, since the dynamic range of the depth values can be quite large, it is advantageous to scale it appropriately before coding. Secondly, for a large object, its depth values might vary significantly, and the depth pixels with small values are commonly more important since they result in large disparity of image pixels in rendering the VO. To avoid introducing too much distortion in encoding depth pixels with small values after scaling, companding[15] is also applied to the depth map. A usual companding approach is to calculate the reciprocal of a depth pixel value Z, where, the companded value $Z^{'}$ is given by $Z^{'} = 1/Z$. Taking into account the scaling and companding operations mentioned above, the final value of a depth pixel Z_f before being fed to the encoder, is given by:

$$Z_f = \frac{Z^{'}}{Z^{'}_{\max}} \cdot S_{\max} = \frac{1/Z}{1/Z_{\min}} \cdot S_{\max} = \frac{Z_{\min}}{Z} \cdot S_{\max}$$

where $Z^{'}_{\max}$ is the maximum value of the companded depth maps, which also corresponds to $Z_{\min}$, the minimum depth values of the VOPs, and $S_{\max}$ is the maximum scaling values. If 8 bits is used to represent a pixel for encoding, then $S_{\max}$ would be 255. Similarly, for 12 bits, $S_{\max}$ would be 4095.

After companding and scaling of the original depth values, the resulting depth map is then encoded using temporal/spatial prediction, similar to their corresponding luminance signal of the texture and alpha map.

4. EXPERIMENTAL RESULTS

In this section, experimental results are provided to evaluate the performance of our proposed object-based coding scheme for plenoptic videos (PVs). Both a synthetic PV and a real-scene PV are encoded for the evaluation, respectively. The synthetic PV "synthesis" is produced by using the 3D Studio Max software with a resolution of 320×240 pixels and 24-bit RGB components per pixel. The real-scene PV "dancer" has a resolution of 720×576 pixels in 24-bit RGB format. It was captured by our multiple video cameras system, as shown in Fig. 6, 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 corresponding depth maps are generated with 16 bits per pixel. Fig. 2 shows a few snapshots of the two PVs and two IBR objects extracted from them – the "ball" and the "dancer", respectively. The depth maps of the synthetic PV and the depth maps of IBR object "dancer" are also shown. Both the "synthesis" and "dancer" have 240 frames/VOPs in each stream. Due to space limitation, snapshots for only 3 streams are shown in Fig. 2, despite that the "synthesis" and "dancer" contain 9 and 6 streams, respectively.

Figs. 7 and 8 show 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 using MPEG-4-like algorithm without spatial prediction, while those denoted by "SP-3", "SP-5" and "SP-7" represent the coding results using the proposed coding scheme with 3, 5 and 7 VO streams within a GOVOF, respectively. It can be seen from Fig. 7 that, for the synthetic IBR object "ball", there is a considerable improvement in PSNR performance (4 dB) of the proposed object-based coding scheme over the direct application of the MPEG-4 to individual VO streams. The coding performances of SP-5 and SP-7 are slightly better than that of SP-3, while the former two are very close to each other. This is to be expected because when the disparity between two video streams increases, spatial prediction becomes less effective. The performance improvement for the real-scene IBR object "dancer", as shown in Fig. 8, is 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. Therefore, the results for SP-3 and SP-5 are very close to each other.

Table 1 shows a comparison of shape coding results produced in different types of prediction modes for different VO streams extracted from the synthetic PV "synthesis"(the left side of Fig. 2) which consists of five video streams. The coding results are denoted by average number of bits per VOP. Stream 2 is the main stream, and others are the secondary streams. The object "ball" has a lot of motion, whereas the object "pyramid" is static and the object "green hose" moves very slowly. From Table 1, we can see that stream 1 and stream 3 have better shape coding results than stream 0 and stream 4. This is because the disparity of the formers with respect to the main stream is much smaller than those of the latters. It can be seen that the hybrid mode achieves the best performance than using intra or inter stream mode alone.

Since the variations in the depth map within the IBR object is much less than the texture information, the depth map can be coded with a higher compression ratio than the latter. The rendering examples displayed in Fig. 9 are rendered using the reconstructed depth maps, where the average compression ratio of the depth map for the IBR object "ball" is about 500 at a PSNR of 40 dB. Finally, to further demonstrate the object-based functionality of the proposed codec, the renderings from the real-scene PV "dancer" at both the frame and the object levels are also shown in Fig.10.

In closing, we note that the performance of the proposed system can be further improved if more tools of MPEG-4 such as four motion vectors for a MB, direct prediction mode and so forth are incorporated in coding the secondary VO streams. Moreover, it would be valuable to incorporate other advanced coding tools in the new H.264 standard into the proposed compression scheme for better coding performance. These will be studied in our future work.

Table 1: Comparison of binary shape coding results using different shape prediction modes.

VO stream name \ Shape prediction mode		Intra stream mode	Inter stream mode	Hybrid mode
Ball	Stream1/stream3	409 bits/VOP	287 bits/VOP	287 bits/VOP
	Stream0/stream4	407 bits/VOP	307 bits/VOP	307 bits/VOP
Hose	Stream1/stream3	388 bits/VOP	351 bits/VOP	344 bits/VOP
	Stream0/stream4	405 bits/VOP	368 bits/VOP	359 bits/VOP
Pyramid	Stream1/stream3	143 bits/VOP	356 bits/VOP	143 bits/VOP
	Stream0/stream4	148 bits/VOP	401 bits/VOP	148 bits/VOP

Fig. 6. Configuration of our multiple video cameras system.

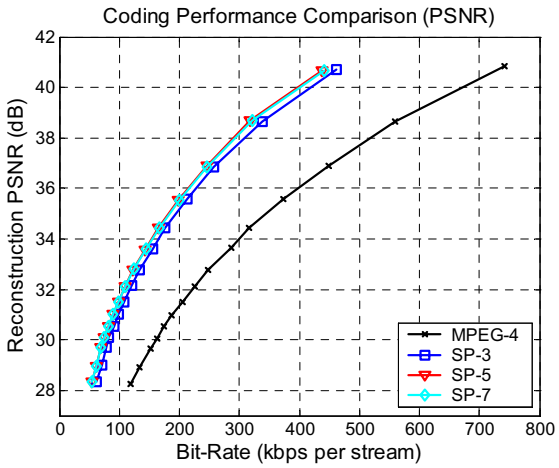


Fig. 7. Object-based coding result for the IBR object "ball".

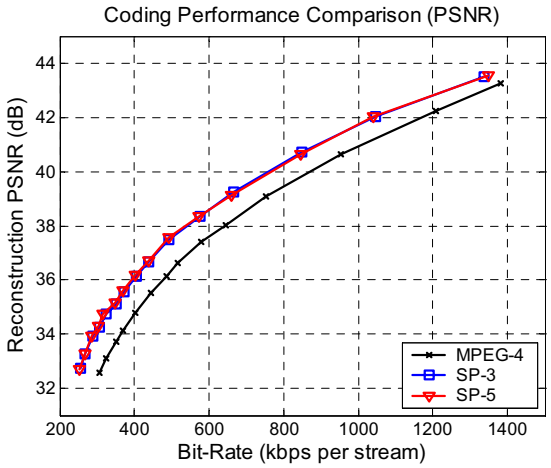


Fig. 8. Object-based coding result for the IBR object "dancer".

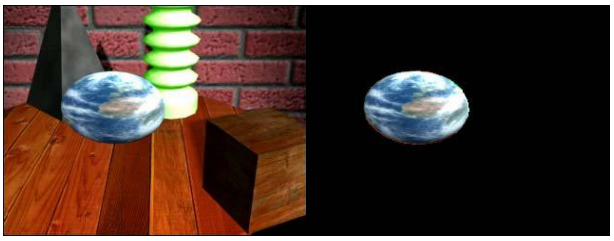


Fig. 9. Typical rendering results for the IBR object "ball".

Fig. 10. Typical rendering results for the IBR object "dancer".

5. CONCLUSION

In this paper, a new object-based compression system for a class of dynamic image-based representations called plenoptic videos is introduced. It enables flexible object-based functionalities and provides support for improving the rendering quality. It also exploits both the temporal and spatial redundancy among VO streams in the videos to achieve higher compression efficiency in texture coding, binary shape coding, grayscale shape (alpha map) coding and depth coding. Experimental results show that considerable improvements in coding performance are obtained for both synthetic and real scenes. The flexibility to manipulate and render individual IBR objects and its coding performance are also demonstrated.

REFERENCES

1. H.Y. Shum, S.B. Kang and S.C. Chan, "Survey of Image-Based Representations and Compression Techniques," in *IEEE Trans. Circuits and System for Video Technology*, vol. 13, pp. 1020-1037, Nov. 2003.
2. 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.
3. 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.
4. M. E. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proc. of IEEE ICASSP'86*, pp. 521–524, 1986.
5. 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.
6. J. X. Chai, X. Tong, S.C. Chan and H.Y. Shum, "Plenoptic sampling," in *Proc. of SIGGRAPH'00*, pp. 307–318, July 2000.
7. 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.
8. ITU-T Recommendation ISO/IEC 14496-2:2001, "Information Technology- Coding of audio-visual objects -- Part 2: Visual".
9. 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.
10. M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. of SIGGRAPH'96*, pp. 31-42, Aug. 1996.
11. S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, "The lumigraph," in *Proc.of SIGGRAPH'96*, pp. 43-54, Aug. 1996.
12. Y. Li, J. Sun, C. K. Tang and H. Y. Shum, "Lazy snapping," in *Proc. of SIGGRAPH'04*, pp.303-308, 2004.
13. F. Bossen and T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation," in *Proc. of IEEE ICASSP'97*, vol. 4, pp. 3129-3132, Munich, Germany, Apr., 1997.
14. MPEG-4 video verification model v18.0, ISO/IECJTC1/SC19/ WG11 Coding of Moving Pictures and Audio N3908, Pisa, Jan. 2001.
15. N. S. Jayant and P. Noll: *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, N.J., 1984.

.