Object-Based Coding for Plenoptic Videos

King-To Ng, *Member, IEEE,* Qing Wu, Shing-Chow Chan, *Member, IEEE,* and Heung-Yeung Shum, *Fellow, IEEE*

***Abstract*—A new object-based coding system for a class of dynamic image-based representations called plenoptic videos (PVs) is proposed. PVs are simplified dynamic light fields, where the videos are taken at regularly spaced locations along line segments instead of a 2-D plane. In the proposed object-based approach, objects at different depth values are segmented to improve the rendering quality. By encoding PVs at the object level, desirable functionalities such as scalability of contents, error resilience, and interactivity with an individual image-based rendering (IBR) object can be achieved. Besides supporting the coding of texture and binary shape maps for IBR objects with arbitrary shapes, the proposed system also supports the coding of grayscale alpha maps as well as depth maps (geometry information) to respectively facilitate the matting and rendering of the IBR objects. Both temporal and spatial redundancies among the streams in the PV are exploited to improve the coding performance, while avoiding excessive complexity in selective decoding of PVs to support fast rendering speed. Advanced spatial/temporal prediction methods such as global disparity-compensated prediction, as well as direct prediction and its extensions are developed. The bit allocation and rate control scheme employing a new convex optimization-based approach are also introduced. Experimental results show that considerable improvements in coding performance are obtained for both synthetic and real scenes, while supporting the stated object-based functionalities.**

***Index Terms*—IBR object coding, image-based rending, MPEG-4, object-based coding, plenoptic videos.**

I. Introduction

IMAGE-BASED rendering (IBR) is a promising approach for the photo-realistic rendering of scenes and objects at a continuum of viewpoint from a collection of densely sampled images. It is an important technology for constructing immersive, 3-D or multiview TVs. Since the data size associated with image-based representations is usually very large, especially in the case of dynamic scenes, efficient methods for its capturing, storage and transmission are active areas of research [1]. Different image-based representations and camera arrangements or sampling geometries have been proposed to simplify the capturing process and storage requirements. Interested readers are referred to [2], [3] for more details. In previous works [4] and [5], the authors have developed a multiple camera system for capturing a class of dynamic image-based representation called "plenoptic videos" (PVs). It is a simplified light field for dynamic environments where a linear array of video cameras is used to simplify the hardware requirement. Furthermore, this simplified and regular camera geometry allows a continuum of virtual views to be synthesized along the line segments joining the video cameras. Using a parallel processing-based system, high-quality rendering of dynamic image-based representations using off-the-shelf equipment were obtained. Moreover, its potential applications in visualization and immersive television systems were demonstrated. The plenoptic videos are also closely related to multiview video sequences [6]–[12]. But the former usually employs vision and graphic techniques and specific camera geometry in order to perform image/video synthesis at nearby locations to support more flexible viewing freedom. Efficient compression approaches play a key role in the practical realization, storage and transmission of dynamic image-based representations. Previous studies of multiview video coding usually focus on frame-based approached or the compression of the video data [19], [37]–[43]. However, to simplify the rendering or synthesis of novel or virtual views, it is advantageous to include additional information such as depth maps and other useful information with the videos. In frame-based plenoptic video systems [4], [16], multiple video streams and their associated depth maps are transmitted. For fast rendering speed, the depth map may need to be triangulated into meshes so as to make use of high-speed graphical hardware. To avoid real-time triangulation, the mesh information may need to be transmitted. In other words, there is a tradeoff between the amount of auxiliary information transmitted and the rendering complexity.

This paper proposes an object-based coding approach for plenoptic videos in order to facilitate their rendering, storage and transmission. The main advantages of using the object-based representation are as follows.

1) By properly segmenting data into objects at different depths, it has been shown that the rendering quality in large environment can be significantly improved [13]–[15].
2) By coding plenoptic videos at the object level, desirable functionalities such as scalability of contents,

Manuscript received July 3, 2008; revised July 4, 2009. First version published January 29, 2010; current version published April 2, 2010. This work was supported in part by the Hong Kong Research Grant Council (RGC) and the Innovation and Technology Fund (ITF). Parts of this paper were presented at the IEEE International Conference on Image Processing, 2005 [57], and the International Symposium of Intelligent Signal Processing and Communication Systems, 2005 [58]. This paper was recommended by Associate Editor, L. Onural.

K.-T. Ng is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong (e-mail: ktng@eee.hku.hk).

Q. Wu and S.-C. Chan are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong (e-mail: qingwu@eee.hku.hk; scchan@eee.hku.hk).

H.-Y. Shum is with the Microsoft Corporation, Redmond, WA 98052-6399 USA (e-mail: hshum@microsoft.com).

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

Digital Object Identifier 10.1109/TCSVT.2010.2041820

1051-8215/$26.00 © 2010 IEEE

error resilience, and interactivity with individual objects (including random access at the object level), etc., can be achieved. For instance, the compressed objects can be transmitted at different rates and composited to different plenoptic videos at the receiver.

For convenience, we shall refer to these objects as having the IBR objects. The first advantage is the consequence of plenoptic sampling [13] which says 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. An 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 by matting and inpainting techniques [15], [31]. This idea has been demonstrated in the "pop-up light fields" [14], where excellent rendering quality can be achieved if the light field is properly segmented into layers of different depth values. Basically, objects are rendered layer by layer and special attention is given to object boundaries by using matting and possible holes are filled by inpainting techniques.

In the proposed object-based coding system for PVs, in additional to the video texture, shape, grayscale alpha map (for matting) and depth information of each IBR object are also coded. This coding scheme may be viewed as a generalization of our previous frame-based compression technique [16] for PVs, except that now arbitrarily shaped video objects rather than images with fixed size are encoded to achieve the advantages mentioned above. In practice, it is possible to extract the layers from depth maps and texture at the decoder side for frame-based systems, at the expense of increased complexity of the decoders. By transmitting object shapes and matting information, these expensive operations and possibly complicated mesh information can be avoided. Another possibility is to transmit the texture and shape information of each IBR object to the receiver. Using the segmentation information and the coded textures, depth map can be estimated as in the transmitter with the same transmission bandwidth. This is very attractive but it also requires very high computational complexity at the decoder and a reliable depth estimation algorithm. Since the proposed coder addresses the coding issues of this important information, it is applicable to the various alternatives described above.

The proposed coding scheme shares many useful concepts with the MPEG-4 video coding standard [17], [18] so as to provide various object-based functionalities. The major difference between the two coding schemes, however, is that: the IBR objects in PVs (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. Consequently, multiple video streams in a PV can be encoded into user-defined IBR objects, and flexibly reconstructed at the decoder for display or rendering at either the object level or frame level. Moreover, due to the presence of multiple video streams in a PV, instead of a single stream in MPEG-4, the proposed coding scheme can exploit both the temporal and spatial redundancies among video streams to achieve better compression efficiency.

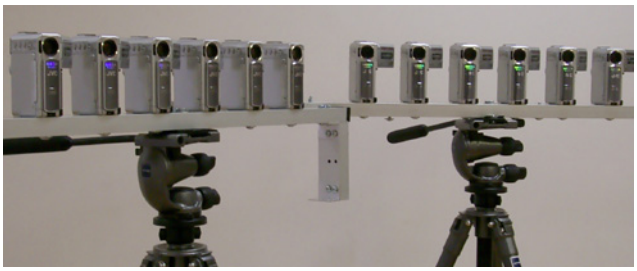

Fig. 1. Configuration of our multiple video camera system.

While spatial and temporal predictions have been employed previously in coding stereoscopic and multiview videos, the proposed coder has been designed to avoid too much dependency in these prediction modes in order to reduce decoding complexity. By properly designing the inter prediction modes and frame structures, novel views can be rendered by selectively decoding adjacent videos containing this view without having to decode the entire multiple videos as in other approaches (please refer to the discussion at the end of Section IV-A for more details). Therefore, the so-called random access problem [1] is satisfactorily addressed. In addition, advanced prediction modes such as global disparity-compensated prediction [1] and direct prediction are employed to improve coding efficiency. These temporal and spatial prediction techniques are useful to object as well as frame-based systems [5]. A convex optimization-based rate control algorithm and a new rate-distortion model for the various information are also introduced to perform bit allocation in the proposed PV coding system. To demonstrate the principle and effectiveness of the proposed system, a multiple video camera system was built. Experimental results show that considerable improvement in coding performance is obtained for both synthetic and real scenes, while supporting the stated object-based functionalities.

The rest of this paper is organized as follows. Section II briefly reviews the concept of plenoptic videos and the corresponding capturing systems. The proposed object-based coding system is introduced in Section III. Several key components of the coding system, including the texture coding, shape/depth coding and rate control, are given in Sections III to VI, respectively. Experimental results are presented in Section VII to evaluate the performance and effectiveness of the various coding techniques. Finally, conclusions are given in Section VIII.

II. Plenoptic Videos and Capturing System

The plenoptic videos considered in this paper is a simplified dynamic light fields [4], [5], with viewpoints being constrained to line segments instead of a 2-D plane. Light fields [20] and lumigraphs [21] are image-based representations obtained by recording images on a camera plane so as to render novel views of the scene around this plane. For dynamic light fields, the number of cameras required on a 2-D plane is usually very

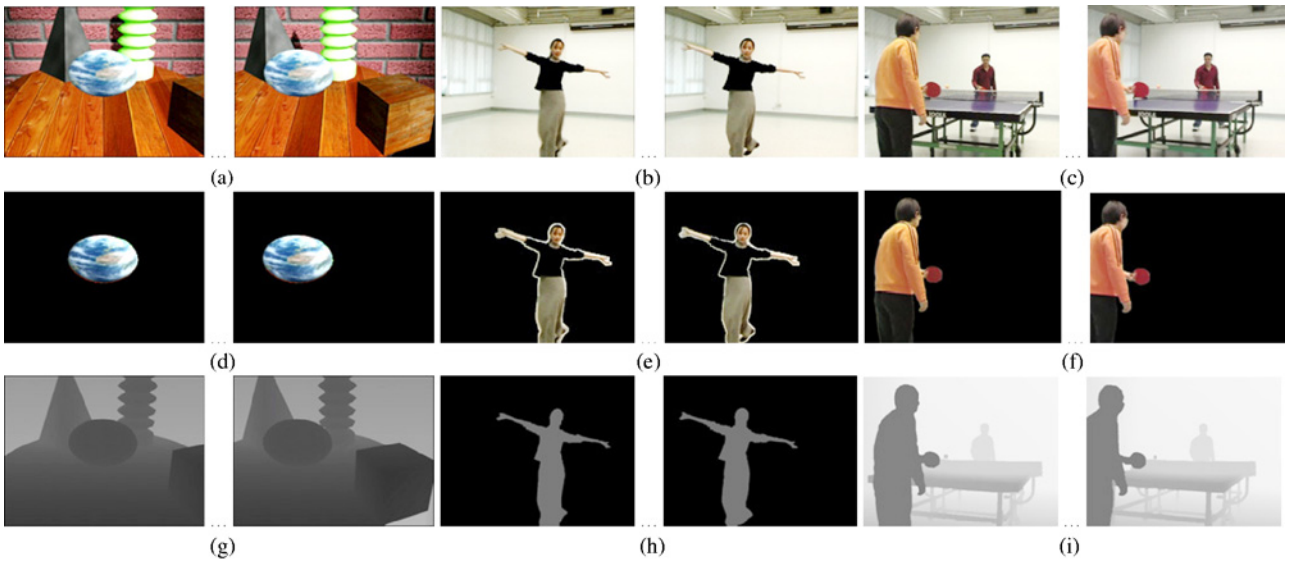

Fig. 2. Top: Snapshots of (a) synthetic PV *Table*, (b) real-scene PVs *Dance*, (c) *Pingpong*. Middle: IBR objects (d) *Ball*, (e) *Dancer*, (f) *Female-Player* extracted. Bottom: depth maps of (g) synthetic PV *Table*, (h) IBR object *Dancer*, (i) real-scene PV *Pingpong*.

large. To avoid such a large dimensionality and the excessive hardware cost, plenoptic videos [5] only consider viewpoints around line segments, since significant parallax and lighting changes along the horizontal direction can still be observed. As mentioned earlier, the plenoptic videos are also very similar to multiview video sequences [6]–[12]. However, plenoptic videos usually rely on denser sampling in regular geometric configurations to improve the rendering quality and provide a continuum of viewpoints around the camera positions.

Previous attempts to generalize image-based representations to dynamic scenes are mostly based on 2-D panoramas. These include the QuickTime virtual reality (VR) [22] and panoramic videos [23]. More recently, there were attempts to construct light field video systems for different applications and characteristics [24]–[27]. In [5], the authors also constructed a capturing system for plenoptic videos, which consists of an array of eight SONY CCX-Z11 charge coupled device (CCD) cameras. For a recent survey of IBR and related issues, see [2] and [3].

Fig. 1 shows the camera system constructed to capture the plenoptic videos. This system consists of two linear arrays of cameras, each hosting six JVC DR-DVP9ah video cameras. The spacing between successive cameras in the two linear arrays is 15 cm and the angle between the arrays can be flexibly adjusted. More arrays may be added to the system to form longer segments, or placed on a plane. Because the videos are recorded on tapes, the system is also more portable for capturing outdoor dynamic scenes. Along each linear camera array, a 4-D simplified dynamic light field is captured. Multiple linear arrays allow users to have more viewing freedom in sport events and other live performances. Other configurations can also be used. The cameras are calibrated with the method in [28] using a large reference grid. Using the extracted intrinsic and extrinsic parameters of the cameras, videos of the cameras are rectified for further processing. Fig. 2 shows several snapshots from three plenoptic videos

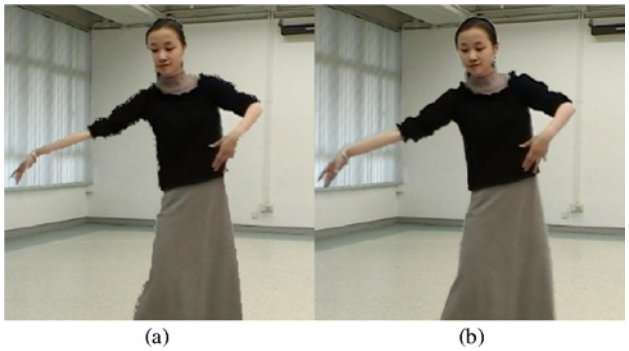

Fig. 3. Rendering samples from the PV *Dance* obtained, respectively, by using: (a) depth map alone, and (b) all the auxiliary information, including depth map, binary shape, and alpha map.

and the IBR objects segmented from the scenes. The left side of Fig. 2 is a synthetic sequence called *Table*, while the middle and right side show two real-scene PVs called *Dance* and *Pingpong*. The *Ball*, the *Dancer* and the *Female-Player* in the scenes are segmented to form IBR objects as shown in Fig. 2(d)–(f). A semi-automatic segmentation method called "lazy snapping" [29] is used to perform the segmentation. The depth maps of the synthetic PVs *Table* and *Pingpong* and the depth maps of the IBR object *Dancer* are also shown. The depth map and shape information of these IBR objects will be used for virtual view synthesis at the decoder (please see [15] for more details). The texture, depth and shape information of an IBR object in a video stream, e.g., the *Dancer* above, form a video object (VO). An instant of a VO at a certain time and camera view is called a video object plane (VOP). Due to space limitation, snapshots for only two streams are shown in Fig. 2, despite that seven and six streams are involved in the PVs of *Table* and *Dance*, respectively. More details of the segmentation and rendering algorithms are discussed in [30], [31].

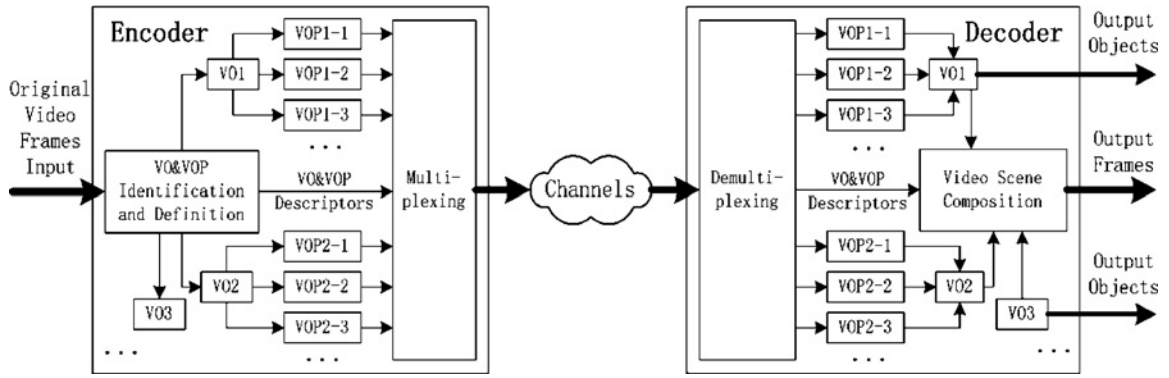

Fig. 4. Generic codec structure of the proposed object-based coding system for plenoptic videos.

III. Object-Based Coding System for Plenoptic Videos

As mentioned earlier, auxiliary information such as depth maps and shape information are extremely important to reduce the rendering artifacts. Efficient methods for coding this information are thus highly desirable. Fig. 3(a) and (b) shows the rendering results from the PV *Dance* obtained respectively by using the depth map alone and all the information above using the rendering techniques proposed in [15]. It can be seen that much better renderings especially at object boundaries can be achieved by incorporating the shape information (including alpha map) and depth map. For fast rendering speed, it is preferable to incorporate (or transmit) this additional information into the compressed bitstream (to the receiver).

In the object-based approach, scenes in a plenoptic video are segmented into multiple IBR objects. After the IBR objects are identified, they can be extracted (e.g., the IBR objects *Ball* from the PV *Table* in Fig. 2) by segmentation. Later on, 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 may be given different numbers of bits (and different amounts of channel coding) and hence different reconstruction qualities (error resilience). They may also be transmitted at different frame rates to temporally achieve object scalability. In frame-based systems [5], an entire PV forms a single IBR object. It offers simple implementation, but it cannot enjoy the object-based functionalities mentioned above. Furthermore, processing of the depth map may be needed to improve the rendering quality especially around object boundaries. Nevertheless, most of the prediction techniques to be introduced later in the next section are also applicable to frame-based systems [5].

Fig. 4 shows the structure of the proposed object-based coding system. It shares many useful concepts with the MPEG-4 video object coding. An IBR object includes the VOPs distributed among all the streams 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 possible motion. The VO/VOP descriptors (e.g., VO/VOP identity (id), height/width, and other syntax information) for the plenoptic video, which are used to compose the video scenes for the plenoptic video at the decoder, are multiplexed together with the VOPs. Via the channels (e.g., the channels inside networks), the encoded bitstream is transmitted to the decoder, which then demultiplexs and reconstructs the VOPs for display, rendering, or other processing. The decoded VOPs can also be composed into a frame for presentation, or further processed by other applications.

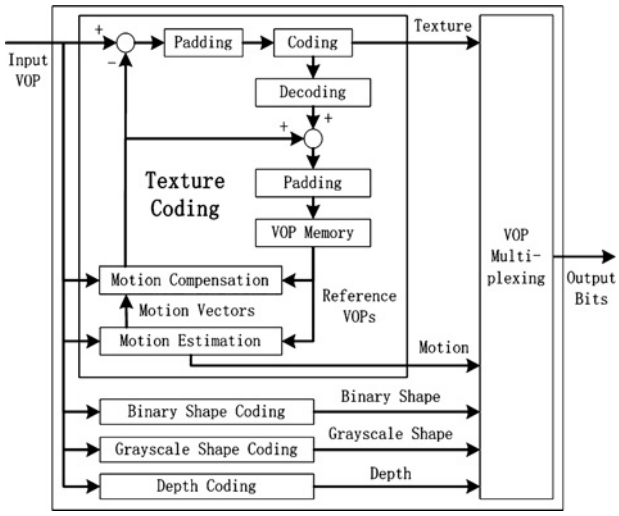

Fig. 5. Diagram of coding a VOP of an IBR object.

Fig. 5 shows the block diagram of coding a VOP belonging to 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 based on discrete cosine transform (DCT) with motion estimation and compensation. Like the MPEG-4 standard, those blocks at the boundary of the VOP need to be padded into complete blocks using appropriate padding methods before being coded. The binary shape mask of the VOP is encoded using context-based arithmetic encoding algorithm[32]. The grayscale shape information, which is also called the alpha map and is defined by an eight-bit number, is useful in matting VOs during VO composition and rendering at the decoder. It is coded through the alpha channels in the same way as the luminance signal of texture. The depth map, a type of geometrical information

to facilitate the rendering, is encoded independently as a so-called "depth channel" (it is actually a data space reserved in the output bitstream for storing the encoded depth maps) in the proposed 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, coding of VOPs, along with the bit allocation and rate control for the coding system.

IV. Texture Coding

It is known that adjacent light field images appear to be shifted relative to each other due to disparities among them. Therefore, it is advantageous to employ both temporal prediction and spatial prediction to improve the coding efficiency. Spatial prediction is also referred to as disparity compensated prediction, which has been used in coding of static light fields [33]–[35], stereoscopic images [6], [36] and multiview images/videos [7], [12], [37]–[43].

A. Basic Coding Methodology

Fig. 6 illustrates the basic methodology for coding the texture of an IBR object in a plenoptic video. It employs prediction in both temporal and spatial directions. For simplicity, only three VO streams are shown. In each VO stream, we have a view of the IBR object, which we refer to as the VOP as mentioned previously. There are two types of VO streams associated with each dynamic IBR object: main VO stream and secondary VO stream. The 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 bidirectional prediction for the B-VOPs. To provide random access to individual VOP, we adopt the Group of VOPs (GOVOP) structure of MPEG-4 in the main VO stream. A GOVOP contains an I-VOP and possibly P-VOPs and/or B-VOPs between this I-VOP and the subsequent 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. 6. 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, which contain respectively the P- and B-VOPs of the main VO stream. A VOP from the secondary stream in an I-VOP field is encoded using disparity-compensation 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. The disparity is the displacement of pixels, resulting from the geometry of the objects and the relative positions of the objects and the viewing cameras.

Disparity-compensated prediction has been used in the coding of static light fields [33]–[35]. Therefore, the coding algorithm considered here can be viewed as their generalization to the dynamic IBR object context. Similarly, apart from

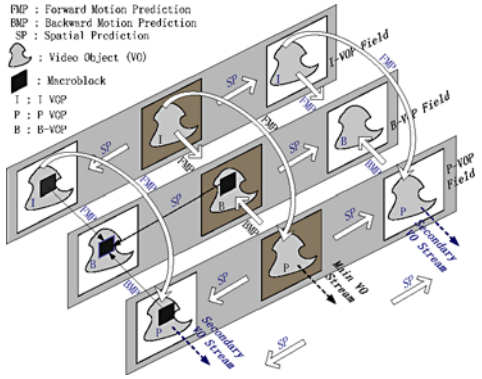

Fig. 6. Basic coding method employing both temporal and spatial predictions for the texture coding of an IBR object in a PV.

using temporal prediction in the same stream, the P/B-VOPs in a secondary steam also employ spatial prediction from their adjacent P/B-VOPs in the main stream for better performance. Compared to the macroblocks (MBs) of a B-VOP in the main stream, whose reference MBs may be obtained by interpolating the forward I/P-VOP and backward I/P-VOP, the MBs of a B-VOP in a secondary stream may have their reference MBs interpolated from three directions. It should be mentioned 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 with the aid of padding or using shape-adaptive DCT. The concept of GOVOP in the main stream can be extended to the VOP fields covering all the streams, which will be called a group of VOP fields (GOVOPF), to provide random access points in a PV. By extending the concept of GOVOPF, group of frame fields can be collected to form group of frame fields (GOFF).

Many other coding schemes for multiview sequences [6]–[12], [40]–[42] have also been proposed to improve coding efficiencies by exploiting the temporal and spatial similarity between the video streams. A review of multiview video compression based on spatial and temporal similarities between multiview video streams is available in [42]. Three important examples are the GOGOP prediction structure in [11], the sequential view prediction structure proposed in [12], and the structure with hierarchical B pictures in [42]. In sequential view prediction, the video sequence from the first camera is encoded using temporal prediction. Then, each frame in the second video sequence is predicted using the corresponding frame in the first sequence and temporal prediction from other frames in the second sequence. Similar sequential prediction operations are performed for the remaining video sequences. The structure with hierarchical B pictures in [42] is similar to the sequential view prediction structure in the spatial direction (inter-view direction), while P frames between two I frames are all replaced with B frames in the temporal direction to achieve further coding efficiency.

As mentioned in [1]–[5], random access capability is an important consideration in IBR systems. In light fields, appropriate image pixels at adjacent light field images are retrieved to render the corresponding novel views. If image pixels are predicted sequentially as in [12], selective decoding of these pixels may be extremely complicated and computationally

expensive due to the interdependence of prediction used and entropy coding [1]–[5]. In the proposed encoder, the prediction structure as illustrated in Fig. 6 is carefully designed to facilitate the selective transmission and decoding of the PVs, while achieving a reasonably good coding performance. Similarly, in the transmission of PVs, it may be difficult to transmit all the PV data to target users due to its large data size. Fortunately, in selective decoding and transmission of PVs, we only need to decode and transmit the two adjacent video streams in order to render the view in between. The proposed prediction structure in Fig. 6 appears to be more advantageous and efficient in supporting such selective decoding or transmission of PVs compared to other prediction structure mentioned above. This is because each secondary stream is coded independently with reference to the main stream only. Hence, in the worst case where two VO streams used for rendering are secondary streams, at most three streams need to be decoded (the main stream followed by two secondary streams). This reduces considerably the decoding complexity. Furthermore, in selective transmission of PVs, only the two decoded VO streams used for rendering need to be transcoded and transmitted to the user. On the other hand, in the worst case, the sequential view prediction structure in [12] will require the decoding of all the video streams in order to select and transmit the two relevant streams for rendering. This is the main difference between our coding scheme with other previous works in multiview coding.

B. Advanced Spatial/Temporal Prediction Methods

In this section, two prediction methods to achieve higher coding efficiency for texture, depth and alpha images will be introduced.

1) *Initial Global Motion Vector:* Disparity-compensation prediction can be used to improve the coding efficiency of multiview sequences. Another related prediction method is to employ epipolar geometry between adjacent camera views to facilitate inter-viewpoint prediction [8], [9]. In principle, if the true depth value of an image pixel is known and the surface is Lambertian, its positions in adjacent calibrated cameras are known. This simplifies considerably the prediction of the depth intensity and alpha value of pixels in adjacent video streams. However, depth maps are usually subject to errors and object surfaces may not be Lambertian. General motion estimation-based compensation methods usually give considerably lower prediction errors and hence bits for coding the residuals. To reduce the number of displacement vector or motion vector (MV), we employ a global MV to account for the mean depth of an object. This also helps to reduce the computational overhead and estimation error resulting from large search ranges. Basically, an initial global MV is first estimated from the search area in the reference VOP in the main stream. A local displacement of MV within a reasonable search range around the global MV is then estimated to minimize a certain distortion measure for each MB to be encoded. The final MV for that MB is the sum of the initial global MV and the local MV.

Since the global MV is closely related to the mean depth or disparity of the object, depth maps of the VOPs can be exploited to estimate these initial global MV. Using the pinhole camera model [44] depicted in Fig. 7, where c_m and c_s are optical centers of cameras for the main and the secondary streams, respectively, then the disparity length d for a visible point **P** is given by

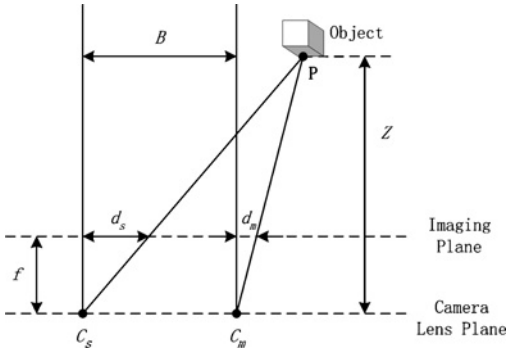

Fig. 7. Disparity calculation for a point of an object using the pinhole camera model.

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

where d_s and d_m are the distances as shown in the Fig. 7, f is the focal length of the camera, B is the baseline distance between two cameras, and Z is the depth value for **P** at the object surface. Using Z_a, the mean depth value of all points on the object surface, for Z in (1), the initial global MV for a VOP in a secondary steam, can be estimated as $V_g = (v_x, v_y)$ where $v_x = f \cdot B/Z_a$ and $v_y = 0$.

Here, we assume that all images have been rectified and thus $v_y = 0$. If not, a global MV can be similarly estimated. Since the disparity within a VO is usually small, the local MV can be found efficiently and accurately within a much narrower searching range after incorporating the initial global object MV. The final MV is used as a common MV for coding texture, depth and alpha maps.

2) *Direct Prediction and Its Extensions:* As in MPEG-4 and H.264, our object-based coding scheme also employs a new motion prediction mode known as direct motion prediction mode [17], or *direct mode* in short, for texture coding. It employs direct bidirectional motion prediction and compensation derived from the MV of a previously coded MB in a P-VOP to encode a MB in a B-VOP. By properly scaling this MV, we can form a pair of forward and backward MVs for that MB in the B-VOP to be encoded. The final motion prediction is calculated by interpolating the forward and backward reference MBs based on the forward and backward MVs so obtained. This direct mode is extended in the proposed PV coding scheme to achieve a higher compression efficiency by exploiting further temporal and spatial correlations between secondary streams and the main stream. Some other direct modes such as the 2-D direct mode presented in [12] have also been proposed for multiview video coding.

a) *Direct prediction mode:* The coding of a MB coded using the direct mode and called direct MB (MB_d) is illustrated in Fig. 8(a). They can exist in the B-VOPs preceding a P-VOP in the main stream [Fig. 8(a)] and the B-VOPs in the secondary steams [Fig. 8(c)]. The coding of a MB assuming

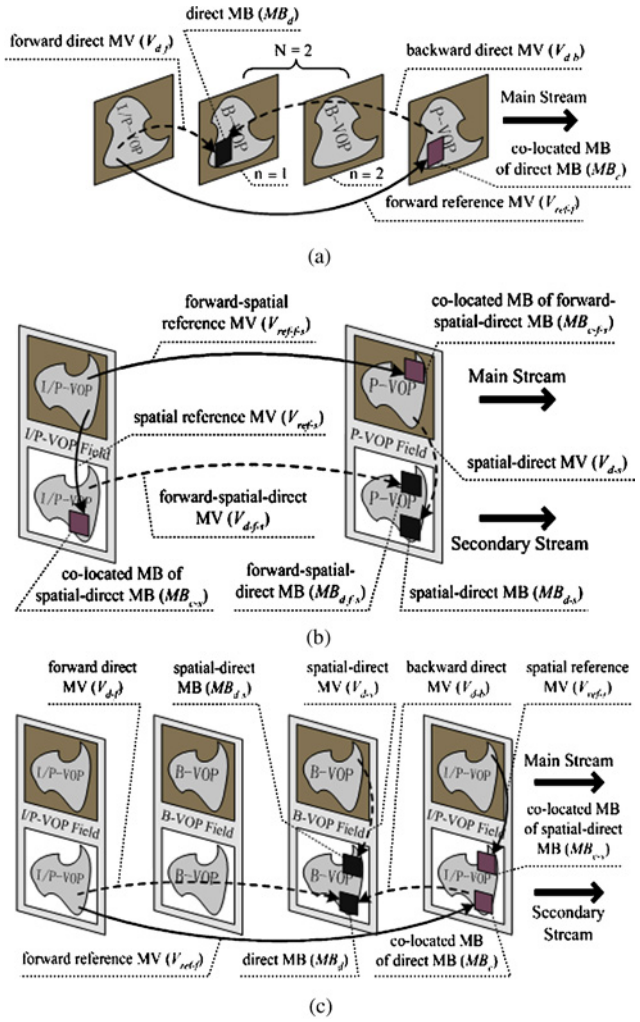

Fig. 8. Direct prediction mode and its extensions. (a) Direct prediction mode in B-VOPs of the main stream coded using MPEG-4. (b) Spatial-direct mode and forward-spatial-direct mode in P-VOPs of a secondary stream. (c) Direct mode and spatial-direct mode in the B-VOPs of a secondary stream.

the VO moves constantly, the co-located MB (MB_c) in the reference VOP will have a similar MV as a direct MB in the B-VOP to be coded. Then the pair of bidirectional MVs for a direct MB, namely forward direct MV ($V_{d\text{-}f}$) and backward direct MV ($V_{d\text{-}b}$) in Fig. 8(a), can be estimated from the available forward reference MV ($V_{ref\text{-}f}$) of the co-located MB (MB_c) in the previously coded P-VOP. Specifically, the forward direct MV ($V_{d\text{-}f}$) and the backward direct MV ($V_{d\text{-}b}$) can be obtained by scaling the forward reference MV ($V_{ref\text{-}f}$) as follows:

$$V_{d-f} = n \cdot V_{ref-f}/(N+1), \quad V_{d-b} = n \cdot V_{ref-f}/(N+1) \quad (2)$$

where n is the index value of a B-VOP in a set of consecutive B-VOPs between the nearest preceding I/P-VOP and the following P-VOP, N is the total number of B-VOPs in that set. To make the prediction more accurate, the method of four MVs (each corresponding to an 8×8 block in the co-located MB) in MPEG-4 is adopted for the forward reference MV. The corresponding direct MVs can still be obtained by using (2) first, and then adjusted by adding a common small delta MV. The best value of the delta MV can be found by testing whether the final resulting direct MVs can achieve the minimum value of the sum of absolute difference between the MB to be coded and the reference MB (the reference MB is obtained by interpolating the forward and backward reference MBs). Since the initial pair of direct MVs can be obtained by (2) at the decoder, only the common small delta MV needs to be encoded. The use of direct mode improves considerably the compression efficiency, especially in case of using the method of four MVs.

b) *Spatial-direct prediction mode:* Fig. 8(b) illustrates the spatial-direct motion prediction mode for coding a spatial-direct MB ($MB_{d\text{-}s}$) in a P-VOP of a secondary stream. The spatial reference MV ($V_{ref\text{-}s}$) comes from the co-located MB ($MB_{c\text{-}s}$) of the nearest preceding I/P-VOP in the same secondary stream. Fig. 8(c) illustrates another case of spatial-direct mode for coding a spatial-direct MB ($MB_{d\text{-}s}$) in a B-VOP of a secondary stream. The spatial reference MV ($V_{ref\text{-}s}$) comes from the co-located MB of the nearest future I/P-VOP in the same secondary stream. In both cases, the co-located MB is coded using the motion prediction with reference to the I/P-VOP of the main stream in the same I/P-VOP field. Basically, the spatial-direct prediction mode utilizes the fact that the disparity between a VOP in a secondary stream and its corresponding VOP in the main stream (within the same VOP field) usually varies slightly during a short time interval. The spatial-direct MV ($V_{d\text{-}s}$) is directly equal to the reference MV ($V_{ref\text{-}s}$). A common small delta MV is also incorporated to this initial direct MV. The benefit of applying the spatial-direct mode for the MBs in the P/B-VOPs of secondary streams is similar to the direct MBs introduced previously, especially when applying the method of four MVs.

c) *Forward-spatial-direct prediction mode:* Since all VOPs in the same VOP-field are captured at the same time instance, the VOs of secondary streams have a similar motion as those in the main stream. Based on this observation, another extension to the direct mode, the forward-spatial-direct mode, is proposed for coding a forward-spatial-direct MB ($MB_{d\text{-}f\text{-}s}$) in the P-VOPs from secondary streams, as shown in Fig. 8(b). The forward-spatial-direct MV ($V_{d\text{-}f\text{-}s}$) is equal to the forward-spatial reference MV ($V_{ref\text{-}f\text{-}s}$) from the co-located MB ($MB_{c\text{-}f\text{-}s}$) of a P-VOP in the main stream. A common delta MV is still needed for this type of direct mode.

It is noted that, besides the direct, spatial-direct and forward-spatial-direct modes, other prediction modes such as intra, forward, backward, spatial, interpolated and triple-interpolated (only for B-VOPs of secondary streams) modes involved in the basic coding method for texture coding are also employed at the same time. The one with the least sum of absolute difference between a MB to be encoded and the predicted MB/MBs from the reference VOP/VOPs will be selected.

C. Entropy Coding for MB Prediction Modes

In the main stream of a PV, the I-VOPs only employ intra mode while P-VOPs may employ either forward or intra mode. In B-VOPs, four prediction modes, i.e., forward, backward,

interpolated and direct modes, may be employed. Variable length coding in the form of Huffman code is employed for different prediction modes. The Huffman codeword table, which involves the entries of all MB prediction modes in the B-VOPs, is created in MPEG-4 according to the occurrence probabilities of prediction modes. The occurrence probabilities in decreasing order for different prediction modes in MPEG-4 are given as follows:

$$direct > interpolated > forward > backward.$$

In the P-VOPs of secondary streams, there may be six MB prediction modes: intra, forward, spatial, interpolated, spatial-direct and forward-spatial-direct modes. B-VOPs may have seven MB prediction modes, i.e., forward, backward, special, interpolated, triple-interpolated, direct and spatial-direct modes. In our experimental results on various testing video streams, including both computer-synthesized and the real ones, the occurrence probabilities for all MB prediction modes in the P-VOPs of secondary streams are as follows:

$$spatial\text{-}direct > forward\text{-}spatial\text{-}direct > spatial > forward > interpolated > intra.$$

For B-VOPs in secondary streams, the occurrence probabilities, in decreasing order, are

$$spatial\text{-}direct > direct > spatial > interpolated > forward > backward > triple\text{-}interpolated.$$

According to the above statistical results, we can specifically design the corresponding Huffman codeword tables for the P-VOPs and B-VOPs in secondary streams so that a better coding efficiency can be obtained. However, employing more MB prediction modes will also increase the overhead used in selecting the prediction modes, since one more prediction mode will result in one more entry in the Huffman codeword table. Fortunately, experimental results show that the extra overhead of the new prediction modes is paid off by the better prediction.

V. Shape and Depth Coding

A. Shape Coding

Shape information is an important component in object-based coding. As in MPEG-4, there are two types of shape information for VOPs in the proposed coder: binary shape information and grayscale shape information. The former provides the binary shape mask co-located with the luminance picture of the VOP, and is used to indicate whether a pixel belongs to that VOP or not. The latter one, also called alpha map, provides transparency levels for the pixels in a VOP, which are useful in matting VOs during composition and rendering. In general, binary shape information can be coded using context-based arithmetic encoding (CAE) algorithm [32]. Similar to the luminance signal of texture, grayscale shape information is coded via the alpha channel with the same MVs obtained from the luminance signal. Here, shape coding mainly refers to the coding of binary shape information. CAE can be used in two different modes: Intra-CAE and Inter-CAE. Intra-CAE codes shape information in intra mode, without using motion prediction, and therefore it is mainly used for I-VOPs in the main stream. In contrast, inter-CAE codes a binary shape mask by employing the motion prediction with reference to the nearest shape mask encoded, and therefore it is used in other types of VOPs except I-VOPs. For example, for a B-VOP in the main stream, inter-CAE selects a shape mask from the nearest preceding I-VOP/P-VOP or future I-VOP/P-VOP as the reference in order to perform shape motion prediction and compensation.

For the shape coding of a VOP in a secondary stream, it is possible to select the reference shape mask from either a VOP in this secondary stream or another in the main stream at the same time instance (i.e., in the same VOP field as illustrated in Fig. 6). In general, the shapes of the VOPs in secondary streams are very similar to the VOPs in the main stream, 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 moving 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 VII.

B. Depth Coding

Being treated as monochrome video, the depth map of a VO can be coded in a similar manner as the luminance signal and alpha map of a VO. Following the definition of "alpha channels" as the coded alpha map data within the bitstream, the coded information of the depth map is called the "depth channel." Since the statistical property of depth map data is typically different from that of luminance and color components [40], [42], certain preprocessings are needed for coding depth map to improve the coding efficiency. Two preprocessing steps are employed here. Firstly, since the dynamic range of the depth values can be quite large, it is advantageous to scale it appropriately before being coded. Secondly, for a large object, its depth values may 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 [45] is applied to the depth map. A usual companding method is to calculate the reciprocal of a depth pixel value Z. Hence, the companded value $Z^{'}$ is given by $Z^{'} = 1/Z$. Taking into account the scaling and companding operations mentioned above, the preprocessed value 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 possible value of the companded depth map, which also corresponds to $Z_{\min}$, the minimum

possible depth value of the VOPs, and $S_{\max}$ is the maximum scaling value. 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 the original depth values, the resulting depth map is then encoded using temporal/spatial prediction, similar to the corresponding luminance signal of the texture and alpha maps.

VI. Bit Allocation and Rate Control

Bit allocation and rate control play very crucial roles in controlling the output bit rate as well as the video quality of a video encoder. In the proposed object-based PV coder, rate control has to be performed at the frame level, object level and MB level. Many approaches have been proposed for bit allocation and rate control at different levels [46]–[51]. We shall base our study on the frame-level bit allocation scheme in [42], the object-level bit allocation approach in [47], and the MB-level rate control algorithm in [48], because of their good performance and simplicity.

The proposed algorithm is based on an improved rate-distortion (*R-D*) model for describing the experimental *R-D* behavior of typical videos. The convexity and monotonicity of this model are then exploited to formulate the bit allocation problem at both the frame and object levels as a convex optimization problem. It can then be solved using standard convex programming methods such as the interior-point methods. The target bits assigned to each VOP (or frame in frame-based systems) are further distributed by the MB-level rate control algorithm [48] to each MB.

A. Rate-Distortion Model

Various empirical *R-D* models have been developed [49]–[55] for DCT-based image/video coding. The proposed *R-D* model has the following form:

$$D = \delta \cdot \exp(-\alpha \cdot R^{\beta}) \tag{3}$$

where D and R are the distortion measure and average bits per pixel, δ is the variance of a *data source* (e.g., a frame, a VOP, or a group of MBs) before encoding, and α and β are the model parameters. When $\beta = 1$, it reduces to the ρ-domain *R-D* model presented in [54]. From our simulation results, the model in [54] works well in a relatively narrow range of R, but it is not so accurate for a larger range of R. We found that when the intermediate variable $1 - \rho$ inside the distortion model in [54] is generalized to $(1 - \rho)^{\beta}$, the resulting distortion model in (3) can give more accurate result in describing experimental *R-D* curves in typical videos. We found that β is always smaller than 1.0, and it mostly varies in a very narrow range centered at 0.7, i.e., approximately $0.6 \leq \beta \leq 0.8$. R is the bits per pixel after excluding the header bits for the syntax, MVs and VOP shapes. The mean square error is used as the distortion measure D.

B. Frame-Level Bit Allocation

For a PV using a GOFF structure, the first step of the frame-level rate control algorithm is to assign a target bits for the current GOFF. Let N_G be the number of frames in the current GOFF, then the target bits T_G is given by

$$T_G = N_G \cdot (t_r / f_r) \tag{4}$$

where t_r is the target bit rate, and f_r is the desired frame rate of the PV (equivalent to the frame field rate of a PV).

Assume that there are M streams in the PV, and N types of frames (e.g., I-, P- or B-) in each stream. A data source here is defined as a frame type from each stream in the PV, and there are totally $M \times N$ data sources in the PV at the frame level. Let $C = \{C_{m,n} | 1 \leq m \leq M,\ 1 \leq n \leq N\}$ denote these data sources, where $C_{m,n}$ represent the n-th frame type of the m-th stream in the PV. Similar to TM5 [49], we also assume that all of the frames with the same type in a stream (i.e., all of the frames belonging to a data source) have the same complexity measure, which is measured by the variance δ. Based on the *R-D* model (4), the bits required and the corresponding distortion for a data source $C_{m,n}$ is estimated as follows:

$$D_{m,n} = \delta_{m,n} \cdot \exp(-\alpha_{m,n} \cdot {R_{m,n}}^{\beta_{m,n}}). \tag{5}$$

The optimal bit allocation for the data sources in the current GOFF can thus be formulated as

$$\begin{aligned} &\min_{R_{m,n}} \sum_{m=1}^{M} \sum_{n=1}^{N} X_{m,n} \cdot S \cdot w_{m,n} \cdot \widehat{D}_{m,n} \\ &= \min_{R_{m,n}} \sum_{m=1}^{M} \sum_{n=1}^{N} X_{m,n} \cdot S \cdot w_{m,n} \cdot \delta_{m,n} \cdot \\ &\exp(-\alpha_{m,n} \cdot R_{m,n}^{\beta_{m,n}}) \end{aligned} \tag{6}$$

subject to

$$\sum_{m=1}^{M} \sum_{n=1}^{N} X_{m,n} \cdot S \cdot R_{m,n} = T_G - \sum_{m=1}^{M} \sum_{n=1}^{N} X_{m,n} \cdot H_{m,n} \tag{7}$$

$$L_{m,n} \leq R_{m,n}, m = 1, \ldots, M, n = 1, \ldots, N \tag{8}$$

where

- $\widehat{D}_{m,n}$ is the actual distortion of the data source $C_{m,n}$;
- S is the number of pixels in a frame (same for all frames);
- $X_{m,n}$ is the remaining number of frames of the data source $C_{m,n}$ in the current GOFF;
- T_G is the remaining number of target bits in the current GOFF;
- $H_{m,n}$ is the estimated header bits for a frame of the data source $C_{m,n}$;
- $w_{m,n}$ is the importance weight for the data source $C_{m,n}$;
- $L_{m,n}$ is the lower bound of the allocated bits required to guarantee a minimum coding picture quality for the data source $C_{m,n}$ (should be a nonnegative value).

If the coupling between the data sources is not taken into account, $\widehat{D}_{m,n}$ can be approximated by (5). Because $\alpha_{m,n}$ and $\beta_{m,n}$ in (5) are both positive and $\beta_{m,n} \leq 1.0$, we can see that (6) is a nonlinear function. But, most importantly, it is a convex function, and also monotonically decreasing

with respect to $R_{m,n}$. Since the objective function (6) is a positive linear combination of (5) for all of the data sources $\{C_{m,n}|1 \le m \le M,\ 1 \le n \le N\}$, it is also convex and monotonic. Hence, the above optimization problem is a convex programming problem. The optimal solution, if it exists, can be readily obtained using one of the standard convex programming methods [56], e.g., the barrier method. Since the number of variables involved is not very large (around 20 or less), the complexity in solving this problem is relatively low. Let $\{R^*_{m,n}|1 \le m \le M,\ 1 \le n \le N\}$ be the optimal solution of the problem in (6)–(8) in bits per pixel. The final number of bits (excluding the header bits) assigned to $\{C_{m,n}|1 \le m \le M,\ 1 \le n \le N\}$, $\{B_{m,n}|1 \le m \le M,\ 1 \le n \le N\}$, is obtained by rounding the estimated final bits $(S \cdot R^*_{m,n})$ to the nearest integer as: $\{B_{m,n} = Round(S \cdot R^*_{m,n})|1 \le m \le M,\ 1 \le n \le N\}$. The assigned bits $B_{m,n}$ will be further allocated among the VOPs within the current encoding frame using an object-level rate control algorithm to be presented in the next sub-section.

Because of the convexity and monotonicity of (3), it can be shown that giving a higher importance weight $w_{m,n}$ to a data source $C_{m,n}$ shall result in more target bits being assigned to it. In the main stream, since I- and P-frames may be used as the reference pictures for successive P/B-frames to be encoded, the importance weight for I/P-frames should be given a relatively higher value (e.g., 1.2 for I-frames, 1.3 for P-frames, and 1.0 for B-frames) so as to reflect the nature of inter-frame dependency. Similarly, in the secondary stream, the importance weights of the frame types can be set slightly lower than the corresponding frame types in the main stream (e.g., 1.0 for secondary I-frames, 1.1 for secondary P-frames, and 0.9 for secondary B-frames) since the formers are encoded with reference to the latters. As a result, the overhead bits for encoding the frames using motion estimation can be considerably reduced.

After a frame of the data source $C_{m,n}$ is encoded with actually used bits $B'_{m,n}$, all the relevant parameters such as T_G, $X_{m,n}$, $H_{m,n}$, $\delta_{m,n}$, $\alpha_{m,n}$ and $\beta_{m,n}$ in (6)–(8) need to be determined or updated. T_G is updated as $T_G - B'_{m,n}$, while $X_{m,n}$ is updated as $X_{m,n} - 1$ (in case $X_{m,n} - 1 > 0$). Linear regression is employed to estimate/update $H_{m,n}$, $\delta_{m,n}$, $\alpha_{m,n}$ and $\beta_{m,n}$, based on the encoding results of the latest encoded frames of the data source $C_{m,n}$ inside a sliding window of size W equal to 20. Due to page limitation, the estimation procedure is omitted here and interested readers are referred to [58] for more details.

C. Object-Level Bit Allocation

The goal of the object-level rate control is to optimally distribute the available target bits, which is obtained by the frame-level rate control algorithm introduced above, to multiple VOPs within a frame such that the overall distortion of that frame is minimized. To this end, the convex optimization-based approach is still employed by converting the object-level rate control problem into a convex optimization problem. Compared to the frame-level bit allocation scheme, the data sources are VOPs instead of frames. The detailed description of this approach is omitted here due to page limitation. Interested readers may refer to [47] for more details.

The region of interest (ROI) functionality at the object level, which is a typical object-based functionality for interactivity with individual object, can be readily realized by assigning higher importance weights to the interested VOPs. This shall result in more bits being assigned to them relative to other less-interested VOPs. For example, in the real-scene PV *Dance* shown in Fig. 2, because the dancer is moving in a static environment, it is natural and reasonable to assign more bits to this important foreground VO than the background VO. In this way, the picture quality of the decoded VO *Dancer* would be much better and thereby significantly improving its rendering quality.

VII. Experimental Results

In this section, experimental results are provided to evaluate the performance of the proposed object-based coding scheme for PVs. Since there is no prior work on object-based coding for PV or light fields, we shall base our comparison with the MPEG-4 video object coding. To achieve selective transmission and decoding and object-based functionalities, it is expected that the bit rate in the object-based coding will be slightly increased over their conventional frame-based counterparts in [5], [16] and other multiview coding approaches. A synthetic PV and two real-scene PVs are used for evaluation. The synthetic PV *Table* with a resolution of 320×240 pixels and 24-bit RGB components per pixel is produced by the 3-D Studio Max software. The real-scene PVs *Dance* and *Pingpong* have a resolution of 720×576 pixels in 24-bit RGB format. It was captured by our multiple video cameras system shown in Fig. 1. The corresponding depth maps are generated (or estimated in the real PV case) with 16 bits per pixel. A few snapshots of the PVs and the IBR objects extracted from them—the *Ball*, the *Dancer*, and the *Female-Player*, respectively, are shown in Fig. 2. Both the PVs of *Table* and *Dance* have 200 frames/VOPs in each stream. The frame rates used for the PVs are 24 frames/s. For illustration, a group of VOPs (GOVOP) structure consisting of 12 VOPs (one I-VOP, three P-VOPs and eight B-VOPs) is employed.

A. Performance of the Proposed Rate Control Algorithm

The performance of the convex optimization-based rate control algorithm is evaluated first. The algorithm is implemented at both the frame and object levels, which is then compared with the integration of conventional frame-level rate control algorithm TM5 and object-level rate control algorithm VM18. The PV *Dance* is selected for testing. Two video streams (streams 2 and 3) of the PV are encoded, where stream 2 is encoded as the main stream and stream 3 is encoded as the secondary stream. Table I summarizes the performances of the proposed rate control scheme and the conventional algorithms. The PSNRs are average values of 200 frames/VOPs of the PV. It can be seen that the proposed rate control scheme outperforms the conventional algorithms by around 0.5 dB in PSNR at a bit rate of 1.7×10^6 bit/s for this pair of video

TABLE I

PERFORMANCE COMPARISON OF RATE CONTROL ALGORITHMS WITH AVERAGE PV BIT RATE 1.7×10^6 BPS/STREAM

		PSNR (db)	
Rate Control Algorithm	IBR Object	Main Stream	Secondary Stream
TM5 plus VM18 (*)	Entire video	38.18	37.58
	VO1 (*Dancer*)	34.81	35.01
	VO2 (*Hall*)	39.04	37.68
Convex optimization-based (#)	Entire video	38.67	38.03
	VO1 (*Dancer*)	35.41	35.55
	VO2 (*Hall*)	39.35	37.94
PSNR gain of (#) over (*)	Entire video	+0.49	+0.45
	VO1 (*Dancer*)	+0.60	+0.54
	VO2 (*Hall*)	+0.31	+0.26

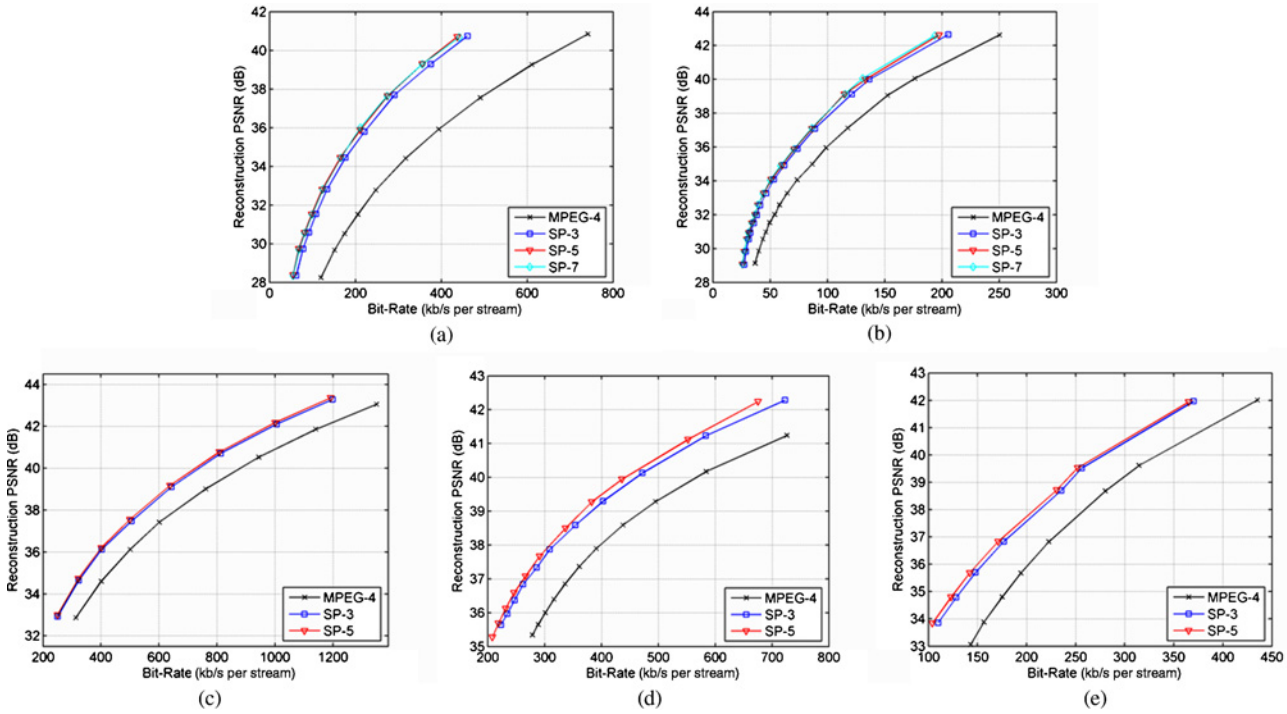

Fig. 9. Coding results using the basic coding method for: (a) Synthetic IBR object *Ball*. (b) *Hose*. (c) Real-scene IBR object *Dancer*. (d) *Female-Player*. (e) *Male-Player*. "SP-3," "SP-5," and "SP-7" represent the coding results with 3, 5, and 7 VO streams in a PV, respectively.

TABLE II

COMPARISON OF CODING PERFORMANCE FOR VARIOUS FRAME TYPES UNDER DIFFERENT CODING SCHEMES

			PSNR (dB) under different Coding Schemes						
IBR Object Name	Bit Rate (kbit/s)	VOP Type	MPEG-4 (*)	SP-3	Advanced-SP-3 (#)	Gain of (#) Over (*)	SP-5	Advanced-SP-5 (∼)	Gain of Over (∼) Over (*)
Ball	300	Entire VO stream	33.9	38.1	39.1	5.2	38.0	39.2	5.3
		I	34.2	38.3	39.5	5.3	38.2	39.5	5.3
		P	34.0	38.2	39.3	5.3	38.2	39.2	5.2
		B	33.7	37.9	38.8	5.1	37.9	39.0	5.3
Dancer	800	Entire VO stream	39.5	40.4	41.2	1.7	40.5	41.0	1.7
		I	39.8	40.7	41.6	1.8	40.6	41.4	1.6
		P	39.7	40.8	41.4	1.7	40.5	41.3	1.6
		B	39.2	39.9	40.9	1.7	40.0	41.0	1.8

streams. Similar experimental results can also be obtained at other bit rates and for other PVs, which are omitted here due to page limitation. The improvement obtained is mainly attributed to the optimal bit allocation among the frames and VOPs by employing the improved *R-D* model. The convex optimization in the proposed algorithm was carried out using the Lindo API [59]. In our simulations, the encoding time of the video sequence is 0.372 s/frame on a Pentium 4 3.0 GHz personal computer. The optimization routines consume about 7.13% of the encoding time.

B. Performance of the Proposed Coding Scheme

We first present the combined coding results of texture and shape coding by using the basic coding method described in Section IV-B-1. Fig. 9 illustrates the coding performance in terms of PSNR for multiple IBR objects, at different bit rates by employing the rate control algorithms introduced in Section VI. It can be seen that the rate control algorithms are able to encode the PVs to the desired bit rates for transmission purpose. For variable rate applications, appropriate constant quantization stepsize can be used to maintain a certain reconstruction quality. Due to page limitation, we only focus on the rate-constrained case below. The curves denoted by "MPEG-4" represent the results obtained by coding each VO stream using the MPEG-4 algorithm without spatial prediction, while those denoted by "SP-3," "SP-5," and "SP-7" represent the number of VO streams used within a PV, respectively. It can be seen from Fig. 9 that, for the synthetic IBR object *Ball*, there is a considerable PSNR improvement (around 4 dB) with the basic coding method for texture coding in the proposed object-based coding scheme over the direct application of the MPEG-4 to each individual VO stream. 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. In Fig. 9(b), the coding performance of another IBR object *Hose* in the synthetic PV is shown with different bitrates. The performance improvements for the real-scene IBR objects *Dancer*, *Female-Player* and *Male-Player* as shown in Fig. 9(c)–(e), are less significant compared to 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. Fig. 10(a) and (b) shows the coding results for the synthetic IBR object *Ball* and real-scene IBR object *Dancer* by employing the advanced spatial/temporal prediction methods in Section IV-B and those by using the basic coding method alone. The advanced methods include initial global MV, direct prediction and its extensions. To save space, only SP-5 for the synthetic PV and SP-3 for the real-scene PV are provided in Fig. 10(a) and (b), respectively. It can be seen that for both types of IBR objects, the incorporation of the advanced prediction methods can significantly improve the PSNR (around 1 dB) over those using the basic coding method. This suggests that the advanced coding methods can further exploit the spatial correlation among the VO streams in a PV. More detailed comparisons for different VOP types under various coding schemes are provided in Table II. The IBR objects *Ball* and *Dancer* are encoded at the bit rates of 300 and 800 kbit/s, respectively.

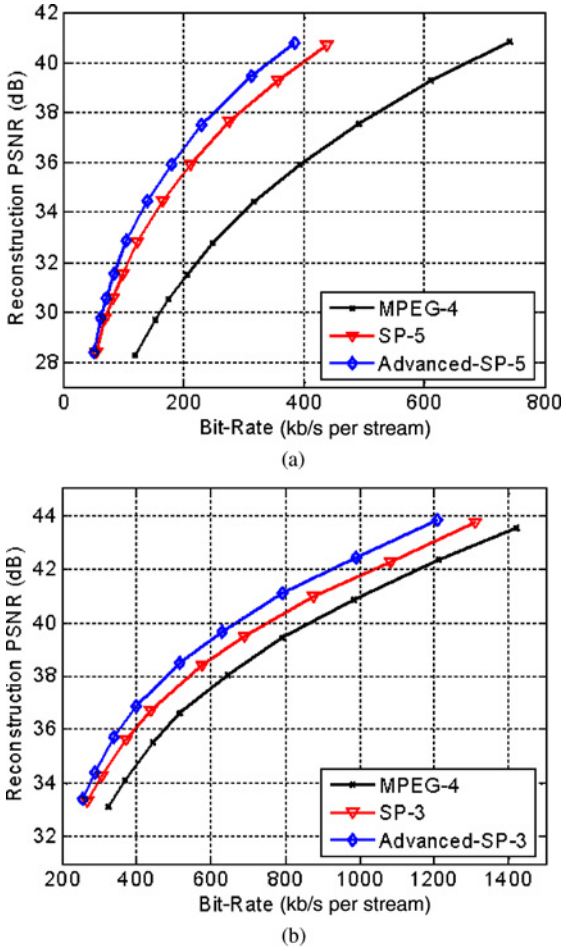

Fig. 10. Coding results comparison between advanced spatial/temporal prediction methods and the basic coding method for: (a) Synthetic IBR object *Ball*. (b) Real-scene IBR object *Dancer*. "SP-3/5" and "Advanced-SP-3/5" represent the coding results with 3/5 VO streams in a PV only using, respectively, the basic coding method and the advanced spatial/temporal prediction methods.

Table III shows a comparison using different prediction models in the shape coding of different VO streams in the synthetic PV *Table*. For simplicity, we only list the results of five streams in the PV, i.e., the main stream (stream 2) along with other four secondary steams. The coding results are obtained by averaging the number of bits per VOP. The object *Ball* has much motion, whereas the object *Pyramid* is static and the object *Hose* moves very slowly. From Table III, we can see that stream 1 and 3 have better shape coding results than stream 0 and 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 are much less than the texture information, the depth map can be coded with a higher compression ratio than the latter. The

TABLE III
COMPARISON OF BINARY SHAPE CODING RESULTS USING DIFFERENT SHAPE PREDICTION MODES (UNIT: BITS/VOP)

Shape Prediction Mode				
	VO Stream Name	Intra-Stream	Inter-Stream	Hybrid
Ball	Stream1/Stream3	409	287	287
	Stream0/Stream4	407	307	307
Hose	Stream1/Stream3	388	351	344
	Stream0/Stream4	405	368	359
Pyramid	Stream1/Stream3	143	356	143
	Stream0/Stream4	148	401	148

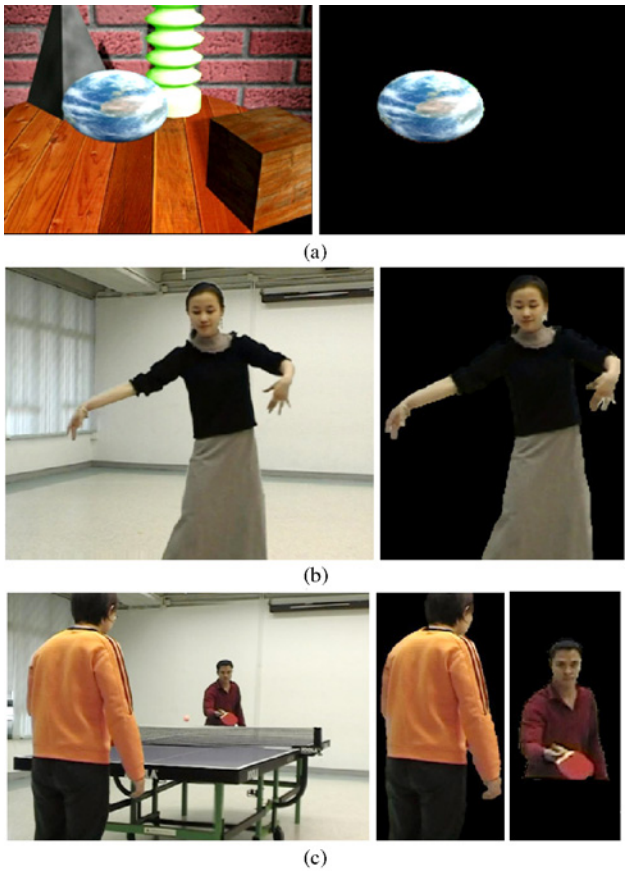

Fig. 11. Typical rendering results for: (a) Synthetic PV *Table* and IBR object *Ball*. (b) Real-scene PV *Dance* and IBR object *Dancer*. (b) Real-scene PV *Pingpong*, and IBR objects *Female-Player* and *Male-Player*.

rendering examples displayed in Fig. 11(a) 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 at both the frame and the object levels from the real-scene PV *Dance* and *Pingpong* are also shown in Fig. 11(b) and (c), respectively.

VIII. CONCLUSION

A new object-based coding system for a class of dynamic image-based representations called plenoptic videos has been presented. By coding the PVs at the object level, the rendering quality for novel views can be significantly improved, especially around object boundaries. Furthermore, object-based functionalities such as scalability of content, interactivity to individual IBR object, etc., can be supported. The coder is designed to avoid excessive inter-dependence between the predictions of images so that selective decoding can be realized to speed up rendering. The texture, binary shape map, alpha map and depth map of an IBR object with arbitrary shape are encoded together by exploiting both temporal and spatial redundancy among the VO streams in a PV, so as to facilitate the matting and rendering of the IBR objects. Advanced spatial/temporal prediction methods were presented to further improve the efficiency for coding texture, depth and alpha maps. Experimental results showed that considerable improvements in coding performance are obtained for both synthetic and real-scene PVs (4-5 dB in PSNR), compared with that using the conventional MPEG-4 algorithm to compress individual streams. The flexibility to manipulate and render individual IBR object and its good coding performance were demonstrated. Most of the proposed techniques are also applicable to frame-based methods.

REFERENCES

[1] H. Y. Shum, S. B. Kang, and S. C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.
[2] H. Y. Shum, S. C. Chan, and S. B. Kang, *Image-Based Rendering*. Berlin, Germany: Springer, 2006.
[3] S. C. Chan, H. Y. Shum, and K. T. Ng, "Image-based rendering and synthesis," *IEEE Signal Process. Mag.*, vol. 24, no. 7, pp. 22–33, Nov. 2007.
[4] 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. IEEE Int. Symp. Circuits Syst.*, vol. 3. May 2004, pp. 905–908.
[5] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan, and H. Y. Shum, "The plenoptic video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1650–1659, Dec. 2005.
[6] M. G. Strintzis and S. Malasiotis, "Object-based coding of stereoscopic and 3-D image sequences: A review," *IEEE Signal Process. Mag.*, vol. 16, no. 3, pp. 14–28, May 1999.
[7] M. E. Lukacs, "Predictive coding of multiviewpoint image sets," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 1986, pp. 521–524.
[8] K. Hata and M. Etoh, "Epipolar geometry estimation and its application to image coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Oct. 1999, pp. 472–476.
[9] J. Lu, H. Cai, J. G. Lou, and J. Li, "An effective epipolar geometry assisted motion-estimation technique for multiview image coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 1089–1092.
[10] C. Zhang and J. Li, "Compression of lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering," in *Proc. IEEE Data Compression Conf.*, Mar. 2000, pp. 253–262.
[11] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Multiview video coding using reference picture selection for free-viewpoint video communication," in *Proc. Int. Picture Coding Symp.*, 2004, pp. 499–502.
[12] *Survey of Algorithms Used for Multiview Video Coding (MVC)*, document N6909.doc, ISO/IEC JTC1/SC29/WG11, Jan. 2005.
[13] J. X. Chai, X. Tong, S. C. Chan, and H. Y. Shum, "Plenoptic sampling," in *Proc. Special Interest Group Graph. Interactive Tech.*, Jul. 2000, pp. 307–318.
[14] 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," *Assoc. Comput. Machinery Trans. Graph.*, vol. 23, no. 2, pp. 143–162, Apr. 2004.
[15] S. C. Chan, Z. F. Gan, K. T. Ng, K. L. Ho, and H. Y. Shum, "An object-based approach to image-based synthesis and processing for 3-D and multiview televisions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 821–831, Jun. 2009.

[16] 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. IEEE Int. Conf. Acoustics Speech Signal Process.*, vol. 3. Apr. 2003, pp. 653–656.
[17] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, document ISO/IEC 14496-2.doc, ISO/IEC, 2001.
[18] *Coding of Moving Pictures and Audio*, MPEG-4 Video Verification Model v18.0, document N3908.doc, ISO/IEC JTC1/SC29/WG11, Jan. 2001.
[19] W. Yang, Y. Lu, F. Wu, J. Cai, K. N. Ngan, and S. Li., "4-D wavelet-based multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 11, pp. 1385–1396, Nov. 2006.
[20] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Special Interest Group Graph. Interactive Tech.*, Aug. 1996, pp. 31–42.
[21] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. Special Interest Group Graph. Interactive Tech.*, Aug. 1996, pp. 43–54.
[22] S. E. Chen, "QuickTime VR—An image-based approach to virtual environment navigation," in *Proc. Special Interest Group Graph. Interactive Tech.*, 1995, pp. 29–38.
[23] K. T. Ng, S. C. Chan, H. Y. Shum, and S. B. Kang, "On the data compression and transmission aspects of panoramic video," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Oct. 2001, pp. 105–108.
[24] B. Wilburn, M. Smulski, K. Lee, and M. Horowitz, "The light field video camera," in *Proc. Soc. Photographic Instrum. Engineers Electron. Imaging: Media Process.*, vol. 4674. Jan. 2002, pp. 29–36.
[25] B. Goldlücke, M. Magnor, and B. Wilburn, "Hardware-accelerated dynamic light field rendering," in *Proc. Vision Modeling Vis.*, 2002, pp. 455–462.
[26] T. Naemura, J. Tago, and H. Harashima, "Real-time video-based modeling and rendering of 3-D scenes," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 66–73, Mar.–Apr. 2002.
[27] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera," in *Proc. Eurograph. Workshop Rendering*, 2002, pp. 77–86.
[28] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[29] Y. Li, J. Sun, C. K. Tang, and H. Y. Shum, "Lazy snapping," in *Proc. Special Interest Group Graph. Interactive Tech.*, 2004, pp. 303–308.
[30] Z. F. Gan, S. C. Chan, K. T. Ng, and H. Y. Shum, "Object tracking for a class of dynamic image-based representations," in *Proc. Soc. Photograph. Instrum. Engineers Visual Commun. Image Process.*, Jul. 2005, pp. 1267–1274.
[31] Z. F. Gan, S. C. Chan, and H. Y. Shum, "Object tracking and matting for a class of dynamic image-based representations," in *Proc. IEEE Adv. Video Signal-Based Surveillance*, Sep. 2005, pp. 81–86.
[32] F. Bossen and T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 4. Apr. 1997, pp. 3129–3132.
[33] J. Li, H. Y. Shum, and Y. Q. Zhang, "On the compression of image based rendering scene: A comparison among block, reference and wavelet coders," *Int. J. Image Graph.*, vol. 1, no. 1, pp. 45–61, 2001.
[34] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 338–343, Apr. 2000.
[35] X. Tong and R. M. Gray, "Coding of multiview images for immersive viewing," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, vol. 4. Jun. 2000, pp. 1879–1882.
[36] J. R. Ohm, "Stereo/multiview encoding using the MPEG family of standards," in *Proc. Electron. Imag.*, Jan. 1999, pp. 242–253.
[37] A. Puri, R. V. Kollarits, and B. G. Haskell, "Basics of stereoscopic video, new compression results with MPEG-2 and a proposal for MPEG-4," *J. Signal Process.: Image Commun.*, vol. 10, nos. 1–3, pp. 201–234, 1997.
[38] T. Naemura, M. Kaneko, and H. Harashima, "Compression and representation of 3-D images," *Institute Electron. Inform. Commun. Engineers Trans. Inf. Syst.*, vol. E82-D, no. 3, pp. 558–567, 1999.
[39] J. R. Ohm, "Encoding and reconstruction of multiview video objects: Looking at data compression in the context of the MPEG-4 multimedia standard," *IEEE Signal Process. Mag.*, vol. 16, no. 3, pp. 47–54, May 1999.
[40] A. Smolic, K. Müeller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1606–1621, Oct. 2007.
[41] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol., Special Issue 3DTV Multiview Video Coding*, vol. 17, no. 11, pp. 1461–1473, Oct. 2007.
[42] K. Müller, P. Merkle, and T. Wiegand, "Compressing time-varying visual content," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 58–67, Nov. 2007.
[43] M. Flierl and B. Girod, "Multiview video compression," *IEEE Signal Process. Mag.*, vol. 24, no. 7, pp. 66–76, Nov. 2007.
[44] Y. Wang, J. Ostermann, and Y. Q. Zhang, *Video Process. Commun.*, Englewood Cliffs, NJ: Prentice-Hall, 2002.
[45] N. S. Jayant and P. Noll, *Digit. Coding Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.
[46] Q. Wu, S. C. Chan, and H. Y. Shum, "A convex optimization-based frame-level rate control algorithm for motion compensated hybrid DCT/DPCM video coding," in *Proc. IEEE Int. Conf. Image Process*, Oct. 2006, pp. 2469–2472.
[47] Q. Wu, S. C. Chan, and H. Y. Shum, "A convex optimization based object-level rate control algorithm for MPEG-4 video object coding," in *Proc. Asia Pacific Conf. Circuits Syst.*, Dec. 2006, pp. 785–788.
[48] Q. Wu and S. C. Chan, "An improved Macroblock level rate control algorithm for MPEG-4 video object coding," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, Dec. 2005, pp. 261–264.
[49] *Coding of Moving Pictures and Associated Audio*, Test Model 5, ISO/IEC JTC1/SC29/WG11, MPEG, 1994.
[50] T. Chiang and Y. Q. Zhang, "A new rate control scheme using quadratic rate-distortion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 246–250, Feb. 1997.
[51] J. I. Ronda, M. Eckert, F. Jaureguizar, and N. Garcia, "Rate control and bit allocation for MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1243–1258, Dec. 1999.
[52] J. R. Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172–185, Feb. 1999.
[53] H. M. Hang and J. J. Chen, "Source model for transform video coder and its application—Part I: Fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 287–298, Apr. 1997.
[54] Z. He and S. K. Mitra, "Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 840–849, Oct. 2002.
[55] Z. Chen and K. N. Ngan, "Linear rate-distortion models for binary shape in MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 869–873, Jun. 2004.
[56] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.
[57] Q. Wu, K. T. Ng, S. C. Chan, and H. Y. Shum, "On object-based compression for a class of dynamic image-based representations," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, pp. 405–408.
[58] Q. Wu, S. C. Chan, and H. Y. Shum, "Improved methods for object-based coding of plenoptic videos," in *Proc. Int. Symp. Intell. Signal Process Commun. Syst.*, Dec. 2005, pp. 481–484.
[59] *Lindo API User Manual*, Lindo Systems, Inc. [Online]. Available: http://www.lindo.com.

King-To Ng (S'96–M'03) received the B.Eng. degree in computer engineering from the City University of Hong Kong, Hong Kong, China, in 1994, and the M. Phil. and Ph.D. degrees in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 1998 and 2003, respectively.

In 2004, he was with Microsoft Research Asia, Beijing, China, as a Visiting Associate Researcher. Currently, he is with the Department of Electrical and Electronic Engineering, University of Hong Kong, as a Postdoctoral Fellow. His research interests include visual communication, image-based rendering, and video broadcast and transmission.

Qing Wu received the B.E. and M.E. degrees in electrical and information engineering from Huazhong University of Science and Technology, Wuhan, Hubei, China, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2007.

Currently, he is a Research Associate with the Department of Electrical and Electronic Engineering, University of Hong Kong. His research interests include image-based rendering, video coding/transmission, and rate control.

S. C. Chan (S'87–M'92) received the B.S. (Eng.) and Ph.D. degrees from the University of Hong Kong, Pokfulam, Hong Kong, in 1986 and 1992, respectively.

Since 1994, he has been with the University of Hong Kong and currently is an Associate Professor. He was a Visiting Researcher with Microsoft Corporation, Redmond, WA and Microsoft, Beijing, China, in 1998 and 1999, respectively. His research interests include fast transform algorithms, filter design and realization, multirate signal processing, and image-based rendering.

Dr. Chan is currently an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS and the *Journal of Very Large Scale Integration Signal Processing and Video Technology*, and a Member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He has been the Chairman of the IEEE Hong Kong Chapter of Signal Processing from 2000 to 2002.

Heung-Yeung Shum (SM'01–F'06) received the Ph.D. degree in robotics from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Currently, he is the Corporate Vice President responsible for search product development with Microsoft Corporation, Redmond, WA. Previously, he oversaw research activities at Microsoft Research Asia, Beijing, China, as well as the lab's collaborations with universities in the Asia Pacific region. He was responsible for the Internet Services Research Center, an applied research organization dedicated to long-term and short-term technology investments in search and advertising at Microsoft, and he was a Researcher with Microsoft Research in 1996, based in Redmond. He moved to Beijing, China as one of the founding members of Microsoft Research, China (later renamed Microsoft Research Asia). There he began a nine-year tenure as a Research Manager, subsequently moving on to become Assistant Managing Director, Managing Director of Microsoft Research Asia, Distinguished Engineer, and Corporate Vice President. He has published more than 100 papers about computer vision, computer graphics, pattern recognition, statistical learning, and robotics. He holds more than 50 U.S. patents.

Dr. Shum is a Fellow of the Association for Computing Machinery.