A Virtual Reality System Using the Concentric Mosaic: Construction, Rendering, and Data Compression

Heung-Yeung Shum, *Senior Member, IEEE*, King-To Ng, *Member, IEEE*, and Shing-Chow Chan, *Member, IEEE*

***Abstract*—This paper proposes a new image-based rendering (IBR) technique called "concentric mosaic" for virtual reality applications. IBR using the plenoptic function is an efficient technique for rendering new views of a scene from a collection of sample images previously captured. It provides much better image quality and lower computational requirement for rendering than conventional three-dimensional (3-D) model-building approaches. The concentric mosaic is a 3-D plenoptic function with viewpoints constrained on a plane. Compared with other more sophisticated four-dimensional plenoptic functions such as the light field and the lumigraph, the file size of a concentric mosaic is much smaller. In contrast to a panorama, the concentric mosaic allows users to move freely in a circular region and observe significant parallax and lighting changes without recovering the geometric and photometric scene models. The rendering of concentric mosaics is very efficient, and involves the reordering and interpolating of previously captured slit images in the concentric mosaic. It typically consists of hundreds of high-resolution images which consume a significant amount of storage and bandwidth for transmission. An MPEG-like compression algorithm is therefore proposed in this paper taking into account the access patterns and redundancy of the mosaic images. The compression algorithms of two equivalent representations of the concentric mosaic, namely the multiperspective panoramas and the normal setup sequence, are investigated. A multiresolution representation of concentric mosaics using a nonlinear filter bank is also proposed.**

***Index Terms*—Concentric mosaic, data compression, image-based rendering (IBR), MPEG-2, video coding, virtual reality (VR).**

I. Introduction

IMAGES and videos are effective means to convey information of objects, the environment or scenes. To provide users with a better experience in virtual walkthrough applications, virtual reality (VR) techniques are becoming more and more important [16], [21]. Applications of VR are numerous, including interactive multi-player games, flight simulators, medical simulation, and visualization tools, just to name a few. VR can broadly be classified as immersive and nonimmersive, according to the types of user's interaction. Immersive VR systems typically employ head-mounted or stereo displays and data gloves to convince users that they are interacting with an artificial world. Interactions with virtual objects via data glove and similar devices are allowed for manipulating the virtual worlds. The convincing illusion can further be enhanced by acoustic and other technologies. Nonimmersive VR systems typically attempt to recreate a virtual environment as convincingly as possible using conventional input and display devices. Examples of such applications include mouse-controlled navigation through a three-dimensional (3-D) environment on a graphics monitor, stereo viewing from the monitor via stereo glasses, and many others. Apple's QuickTime VR [2], for example, creates an environmental map using a panorama and displays a novel view at any angle around a given point on a graphics monitor. Panoramic viewing belongs to a class of techniques called image-based rendering (IBR) which uses real photographs to recreate reality as convincingly as possible. Instead of building and rendering a complete 3-D model of the environment, a collection of images is used to render the scene while supporting virtual camera motion. Though explicit models allow us to add other graphical elements such as new light sources and shadows to the scene, building such models from images or videos can be very slow and tedious. Using laser-based scanners is a good solution for model building because they provide useful information such as registered depth and colored texture maps. However, this approach is limited to small sized objects and is fairly expensive.

IBR, on the other hand, is an excellent alternative if we just want to re-render images at a collection of viewpoints. It provides superior image quality than 3-D model building for very complicated real world scenes and requires much less computational power for rendering, regardless of the scene complexity. Central to IBR is the plenoptic function [1], which describes all the radiant energy that is perceived by the observer at any point in space and time. The panorama mentioned earlier is a special case of the plenoptic function with two dimensions. In its most general form, the plenoptic function is a seven-dimensional function and it is possible to reconstruct novel views at different positions and time from its samples, provided that the sampling rate is sufficiently high [31], [32]. Due to its high dimensional nature, data reduction or compression of the plenoptic function is essential to IBR systems. One possible way is to reduce the dimension of the plenoptic function to be captured by restricting the degrees of viewing freedom in space. The light field of Levoy and Hanrahan [5] and the lumigraph of Gortler *et al.* [4] are simplified plenoptic

Manuscript received June 26, 2000; revised April 1, 2003. This work was supported in part by the AoE in information technology of the Hong Kong Research Grant Council. This paper was presented in part at ACM SIGGRAPH'99, Los Angeles, CA, August 1999, and IEEE ICIP2000, Kobe, Japan, September 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsuhan Chen.

H. Y. Shum is with Microsoft Research Asia, Beijing, China (e-mail: hshum@microsoft.com).

K. T. Ng and S.-C. Chan are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: ktng@eee.hku.hk; scchan@eee.hku.hk).

Digital Object Identifier 10.1109/TMM.2004.840591

1520-9210/$20.00 © 2005 IEEE

function with four dimensions. Another method is to exploit the redundancy of the plenoptic function using waveform coding or other model-based techniques [22]–[30].

In this paper, we report an image-based rendering system for VR or virtual walkthrough applications using a novel 3-D plenoptic function called "concentric mosaic." The concentric mosaic is a set of slit images created by capturing a vertical slit image when a camera is moving around a set of concentric circles, Fig. 3. It is parameterized by three parameters: the radius of the concentric circles, the rotation angle and the vertical elevation. Compared with the light field and the lumigraph, the concentric mosaic has a much smaller file size because it is a 3-D plenoptic function. In contrast to the panorama in which the viewpoint is fixed, the concentric mosaic allows users to move freely in a circular region and observe significant parallax and lighting changes without recovering the geometric and photometric scene models. The capturing and rendering of the concentric mosaic are very efficient. The entire concentric mosaic can be constructed by rotating an off-centered camera on a rotary table and the whole process only takes about 10 min to complete. Furthermore, the rendering of the concentric mosaic amounts to reordering and interpolating of previously captured slit images in the concentric mosaic, which can be performed in real-time. The entire concentric mosaic consists of hundreds of high-resolution images (called mosaic images in this paper) which requires a considerable amount of storage and transmission bandwidth. Unlike the coding of conventional images, the compressed data stream of the mosaic images has to support efficient access to the slit images in order to achieve a fast rendering speed. In fact, efficient access to the compressed data is an important consideration in IBR systems. An MPEG-like compression algorithm, which takes into account the access patterns and redundancy of the mosaic images, is therefore proposed [26]. It should be noted that a similar discrete cosine transform (DCT)-based compression algorithm called the reference block coding (RBC) was independently proposed by Zhang *et al.* recently [27]. Interested readers are referred to [28] for more information of this coder and a related work on a wavelet-based coder [30]. This paper is organized as follows. In Section II, the plenoptic function and its various simplifications are briefly reviewed. The advantages of the concentric mosaic as well as its limitations are then compared with other representations of the plenoptic function. The principle of the concentric mosaic, its construction and representation, and rendering are then described in Section III. Two concentric mosaic examples of real-world scenes and their rendering results are given. Section IV is devoted to the compression of concentric mosaics, where the problem of random access is addressed. Several techniques for reducing the spatial redundancy of the concentric mosaic and their coding results are presented. Furthermore, a novel decimation and nonlinear interpolation scheme for the multiresolution decomposition and progressive transmission of the concentric mosaic is proposed in Section V. Finally, we summarize our works in Section VI, the conclusion.

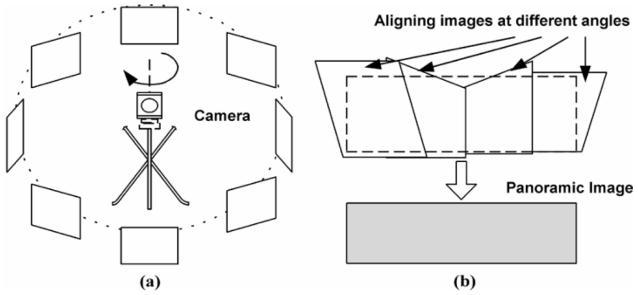

Fig. 1. (a) Capturing of images to construct a panorama. (b) Mapping of the images onto the cylinder to generate the panoramic image.

II. Plenoptic Function and IBR

The name *plenoptic function* was first used by Adelson and Bergen [1] to describe all the radiant energy that can be perceived by the observer at any point (V_x, V_y, V_z) in space and time t. At each point, we can also select any of the viewable rays by choosing an azimuth and elevation angle (θ, ϕ) as well as a band of wavelengths, λ. For a dynamic scene, there is an additional time variable t. This gives rise to the following seven-dimensional plenoptic function:

$$p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t). \tag{1}$$

The plenoptic function can also be viewed as a *scene representation* or as *time-varying environment maps* in computer graphics terminology [6], [2]. In principle, one can reconstruct the view at any point in space and time from the plenoptic function by substituting the appropriate values of $(V_x, V_y, V_z), (\theta, \phi)$, and t into (1). In practice, only samples of the plenoptic function are available, and the reconstruction problem, which is usually referred to as *image-based rendering*, can be stated as follows [6]: *Given a set of discrete samples (complete or incomplete) from the plenoptic function, the goal of image-based rendering is to generate a continuous representation of that function.* As mentioned earlier, image-based rendering using the plenoptic function is an efficient and simple technique for rendering photo-realistic images at different viewpoints. Due to the difficulties in *capturing* and *storing* the plenoptic function, various simplifications have been advocated [2]–[10], [17]. The simplest two-dimensional plenoptic function is the *panorama* (cylindrical [2] or spherical [8]). A panorama can be constructed by first taking a set of images at different angles along a given axis—see Fig. 1(a). The images are then re-projected onto a cylinder to create the *panoramic image*—see Fig. 1(b). During rendering, part of the panoramic image is re-projected onto the screen to produce the view at a given angle. *QuickTime VR* was one of the first VR systems using this concept [2]. Chen and Williams' view interpolation [17] and McMillan and Bishop's plenoptic modeling [6] also used a similar technique.

The *lumigraph* of Gortler *et al.* [4] extends the work of QuickTime VR and plenoptic modeling by further developing the idea of capturing the plenoptic function in a region of the environment. If air is assumed to be transparent, then the radiance along a ray through empty space remains constant. By considering the light leaving the convex hull of a bounded object, it is only necessary to represent the value of the plenoptic function along some surfaces, say a cube, that surrounds the object. Furthermore, by limiting that surface to be a cube, one can determine the radiance along any ray in any direction at any point in space, by tracing backward along that ray through empty space to the surface of the cube. Thus, the five-dimensional (5-D) plenoptic function (ignoring wavelength and time) can be reduced to four dimensions. Independently, Levoy and Hanrahan [5] proposed a similar four-dimensionial (4-D) plenoptic function called the light field. In light field and lumigraph rendering, images are taken at certain points in a two-dimensional (2-D) plane. The points usually lie on a regular rectangular grid. At each grid point, an image of the object or scene is taken. Using this *2-D array of images*, it is possible to render different views of the object or scene at different viewing angles. The *lumigraph* differs from the light field in that an *approximate geometric model* of the object, in addition to the 2-D array of images, can be used to improve the quality of the reconstruction at *lower sampling densities*. It also makes no assumptions about reflectance properties as in previous flow-based methods [4].

The concentric mosaic described in this paper is a 3-D plenoptic function obtained by constraining the viewpoints on a plane. Compared with the light field and the lumigraph, the concentric mosaic has a much smaller file size because it is a 3-D plenoptic function. In contrast to the panorama in which the viewpoint is fixed, the concentric mosaic allows the user to move freely in a circular region and observe significant parallax and lighting changes. The rendering and compression of the concentric mosaic are also very efficient. During rendering, the selected slit images are retrieved from the compressed data stream and decoded to render the novel view. A random access mechanism has to be incorporated for the efficient access of the compressed slit images. These are further elaborated on in Sections III and IV. Furthermore, higher dimensional image-based representations such as the light field and the lumigraph require random access at the *pixel level*, which is considerably more complicated. Because of the use of entropy coding, it is very time-consuming to retrieve and decode a pixel from the compressed data if there is no such provision for efficient random access. The efficient compression, rendering and transmission of higher dimensional image-based representations are thus active areas of recent research [22]–[30].

The concentric mosaic does have some limitations. First of all, concentric mosaics do not capture any vertical parallax because all the camera views are constrained to a horizontal planar region, and only a slit image is taken at each viewpoint. However, as shown by our real-time rendering experiments, the concentric mosaic can still provide the viewers with a strong sense of 3-D perception, even with horizontal parallax only. Perhaps it is due to the fact that our eyes remain relatively planar as we move around so that mainly horizontal parallax is observed. Another limitation is the vertical distortion due to the lack of depth information. More details are given later in Section III-B.

III. Concentric Mosaic

A. Representation and Construction

The concentric mosaic is a set of manifold mosaics constructed from slit images taken by cameras rotating on concentric circles. Unlike panoramas, a manifold mosaic [7] (or multiperspective panorama [9], multiple-centers-of-projection image [19], or pushbroom camera [18]) is composed of images taken by a camera at different viewpoints (or by a set of cameras). In practice, these viewpoints are taken along a continuous path.

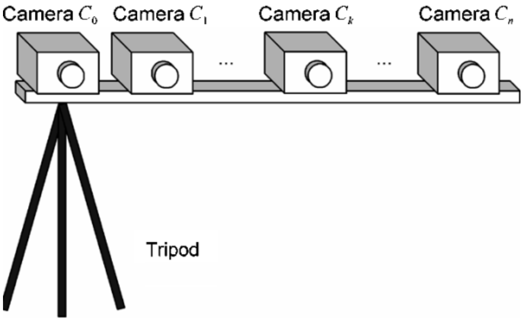

Fig. 2. Setup for constructing the concentric mosaic.

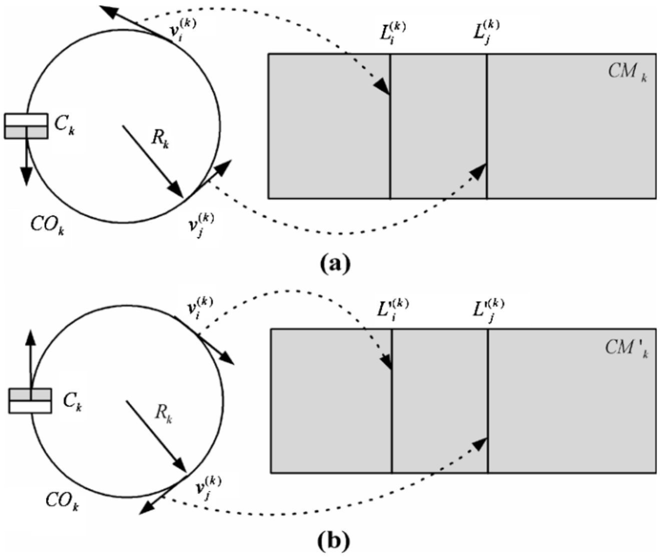

Fig. 3. Construction of a mosaic image.

Fig. 2 shows a possible system for capturing the concentric mosaic. A number of cameras $C_k, k = 0, \ldots, n$ are mounted on a rotating horizontal beam that is supported by a tripod. It is assumed that each camera C_k is a slit camera so that only a vertical line of an image is taken at a certain viewpoint, such as $v_j^{(k)}$ in Fig. 3. To capture the concentric mosaic, the beam is rotating slowly so that each camera C_k is moving continuously along the circle CO$_k$ with a radius R_k. Consider the viewpoint $v_j^{(k)}$ on the circle CO$_k$, the ray that is tangential to the circle CO$_k$ at $v_j^{(k)}$ is captured. Alternatively, we can capture the ray in the opposite viewing direction as shown in Fig. 3(b) using for example another camera facing the opposite direction. If we put together such slit images at different rotation angles along the circle, a mosaic image, as shown in Fig. 3, is formed. For simplicity, we only consider the mosaic images CM, instead of CM'. It can be seen that the entire concentric mosaic consists of $n+1$ mosaic images CM$_k$ captured by the $n+1$ cameras. Each of them consists of slit images taken at different rotation angles by the camera C_k. As shown in Fig. 4, the entire concentric mosaic is conveniently indexed by radius, rotation angle and vertical elevation. If the number of mosaic images, $n + 1$, is large enough, any such rays in free space can be reconstructed

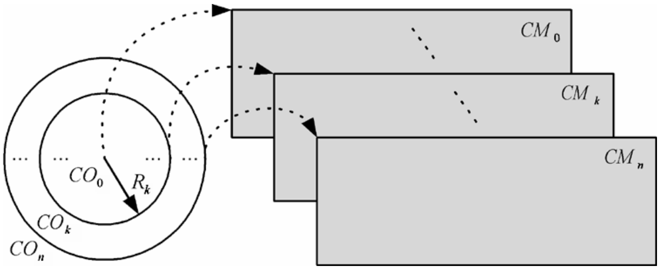

Fig. 4. Entire concentric mosaic.

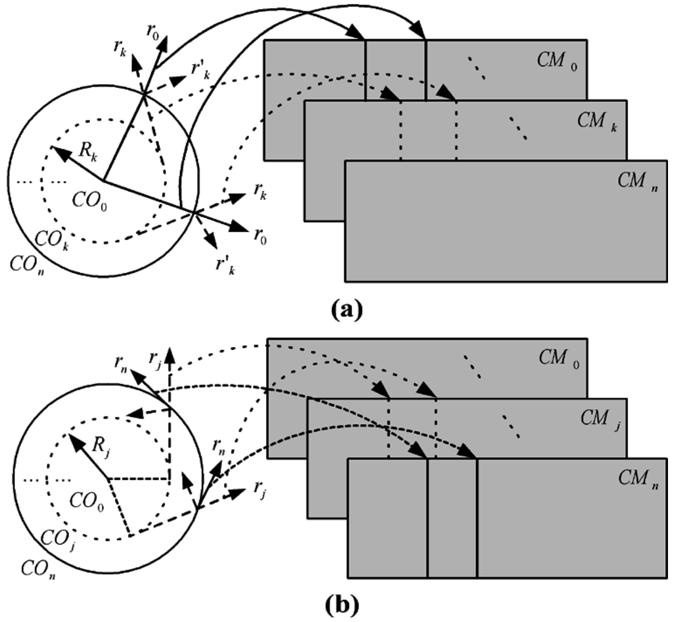

Fig. 5. Construction of the concentric mosaic from one circle: camera along: (a) normal direction and (b) tangential direction.

either from the mosaic images or from their interpolation. It is therefore feasible to reconstruct or render a novel view at any position in the same horizontal plane as the cameras inside the circular free space.

Instead of using many cameras, as shown in Fig. 2, a much simpler capturing method is to use a single off-centered camera that rotates along a circle. The camera can be placed for example on a rotary table with known rotation. At each rotation angle, instead of a slit image, a regular image with multiple vertical lines (depending on the horizontal field of view (FOV) of the camera) is captured. Two possible setups are shown in Fig. 5, one along the normal direction (normal setup), and the other along the tangential direction (tangential setup). For the normal setup shown in Fig. 5(a), no matter where the camera is located on the circle CO$_n$, the same indexed ray captured on the image plane (e.g., the left-most line r_k for CM$_k$ and r'_k for CM'$_k$) is always tangential to the same inner circle CO$_k$. Putting together the same vertical lines in the image planes, the mosaic image CM$_k$ of some inner circle CO$_k$ is constructed. The normal setup covers all the inner circles, from CO$_0$ to CO$_k$ in this case. Similarly, the tangential setup covers all the outer circles between CO$_j$ to CO$_n$. For example, putting the middle vertical lines together, we obtain the mosaic images CM$_0$ and CM$_n$, respectively, from the normal and tangential setups. This capturing method is very simple because only one circular motion is necessary. The resulting visible (or movable) region is, however, significantly limited by the camera's horizontal FOV. Indeed, for the normal setup, the maximum radius of the visible inner circle CO$_k$ is given by $R_k = R_n \sin(h_{\rm FOV}/2)$, where $h_{\rm FOV}$ is the horizontal FOV of the camera. In the following section, we shall consider the rendering of a novel view from such mosaic images.

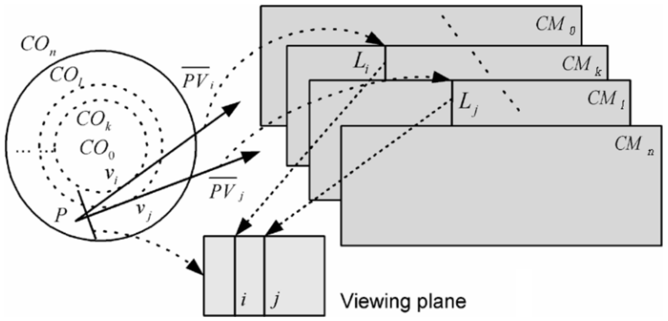

Fig. 6. Rendering a novel view with the concentric mosaic.

B. Rendering of a Novel View

As mentioned earlier, it is possible to render rays of any vertical viewing plane using a sufficient number of mosaic images. It is because the rays are either captured in the mosaic images or they can be interpolated from adjacent lines in the mosaic images. Consider the rendering of a novel view at a point P with a polar coordinate (R, θ) measured from the center of the concentric mosaic, CO$_0$, as shown in Fig. 6. The ray $\overline{PV}_i$ is not captured at the novel view point P. Since the circular region is in free space, we can use the ray previously captured at point v_i in the mosaic image CM$_k$ [4], [5]. Similarly, the ray $\overline{PV}_j$ can be retrieved from the point v_j in the mosaic image CM$_l$. Therefore, the novel view at P can be completely constructed by properly retrieving the slit images from the mosaic images. The rendering is therefore very efficient. In practice, however, only a small subset of the rays is stored in the mosaic images. Those rays that are not recorded in the mosaic images have to be approximated from adjacent rays that were previously captured in the mosaic images.

By rendering with the slit images from different locations, however, we are making an implicit assumption that the depth is at infinity. This approximation causes vertical distortion in rendered images. Several strategies can be used to alleviate this problem [10]. By using constant depth approximation, the depth information requirement can be reduced to a constant for all the captured vertical lines. This approach is essentially a cylindrical environment map that dynamically updates its content based on the location of the novel view. There are vertical distortions for objects whose depth is different from the assumed depth. In our system, the user can interactively adjust the assumed depth with a slider control to correct the distortion for the object of interest.

From the above discussion, it is noted that the concentric mosaic does not capture any vertical parallax, because all the camera views are constrained to be in a horizontal planar region, and only a slit image is taken at each viewpoint. Fortunately, rendering experiments show that the concentric mosaic

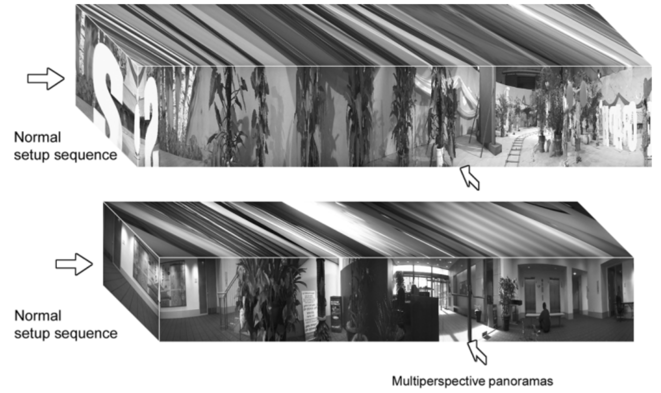

Fig. 7. Data volume of the concentric mosaics: (top) *Kids* and (bottom) *Lobby*.

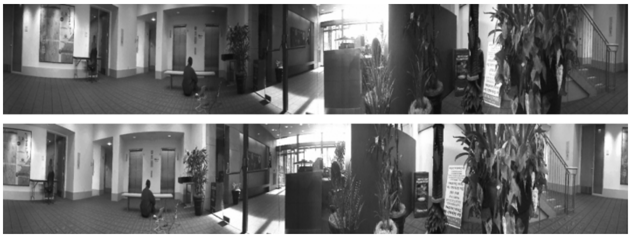

Fig. 8. Two mosaic images of the concentric mosaic *Lobby*.

can still provide the viewers with a strong sense of 3-D perception, even with horizontal parallax only.

C. Experimental Results

In this section, the details of capturing concentric mosaics from real-world scenes are described. Some of the mosaic images from two concentric mosaics called *Lobby* and *Kids* are given together with the rendered results.

In our experiments, we have used a Sony Mini DV digital video camera. A Parker 5-in rotary table is used to slowly rotate the camera along an off-centered circle and to provide accurate rotation parameters. In cases where accurate rotation parameters are not available, vision techniques such as camera calibration and motion estimation could be used to recover the rotation parameters. The reasons that we rotate the camera are: 1) to get sufficient samples along the angular direction for the concentric mosaic and 2) to avoid motion blur during the capture. The capturing process of the concentric mosaic *Lobby* takes about 90 s and a total of 1350 image frames are recorded. The resolution of each digitized frame is 320×240. The images are then rebinned to construct the entire concentric mosaic as described in Section III-A. It took only about 10 min to setup, capture and digitize a complete video sequence needed for constructing the concentric mosaic. Fig. 7 shows the data volume of the concentric mosaic *Lobby* and another concentric mosaic *Kids*, where the normal setup sequences of the concentric mosaics are packed together. The total number of mosaic images in *Lobby* is 160. It is because out of the 320 lines, half of them correspond to the mosaic images in opposite direction CM'—see Fig. 5(a). The resolution of *Kids* is slightly higher and its normal setup sequence consists of 1462 images, each having a resolution of 352×288. Fig. 8 shows two mosaic images from the concentric mosaic *Lobby*. Strong parallax can be seen between the plant and the poster. Lighting changes caused by sunshine can be observed near the ceiling. The dramatic lighting change is partly due to the automatic exposure control of the camera during capturing. Fig. 9 shows two novel views rendered from the concentric mosaic *Lobby*. Again, it can be seen that strong parallax is observed between the plant and the poster.

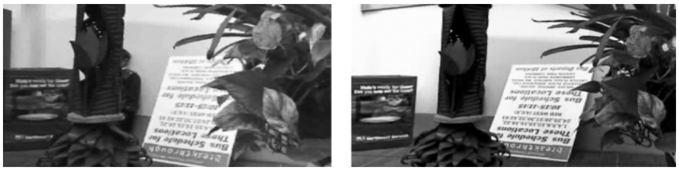

Fig. 9. Two rendered novel views from the concentric mosaic *Lobby*.

IV. Compression of the Concentric Mosaic

A. Random Access Problem

Since the concentric mosaic has large spatial resolution, it has to be compressed for digital storage and transmission. For instance, the entire concentric mosaic *Lobby* consists of 1350 (320×240) video frames which requires 300 MB of storage without compression. It is natural to apply standard image compression techniques like transform coding, vector quantization and wavelet transform to compress these images. Most of these compression algorithms employ entropy coding, such as Huffman or arithmetic coding, to achieve a better compression ratio. Therefore, these symbols after compression are of variable size. This, unfortunately, complicates the rendering of the concentric mosaic. In fact, it is very time consuming to retrieve the slit images if the bitstream does not support any mechanism for randomly or efficiently accessing the compressed slit images. This is also the reason why Levoy and Hanrahan [5] have proposed to use vector quantization (VQ) [12] with fixed index size in solving the compression problem in light field rendering. The fixed size of the VQ index greatly simplifies the retrieval of the pixel data from different light field images to reconstruct the novel view. It also makes real-time decoding possible by utilizing the simple table look-up nature of VQ decoders.

The concentric mosaic basically consists of a series of images, which are retrieved column by column to reconstruct a novel view. The access pattern of the concentric mosaic, in contrast to the light field and lumigraph, is relatively regular. This considerably simplifies the random access problem mentioned earlier. In fact, an entire image line instead of individual pixels is retrieved at the same time to reconstruct the novel view. Therefore, it becomes feasible to generate pointers for indexing the starting locations of each line in the compressed data. On the other hand, the storage overhead due to the pointers is still at an acceptable level. Moreover, several adjacent lines can be grouped together to further reduce the storage overhead. This opens up new possibilities of using more advanced coding techniques for compressing the concentric mosaic. Of course, it requires more computational power to cope with the increased complexities of the decoder. But, we believe that, with proper choice of the coding algorithms and the processing power of nowadays computers, this is a promising and indeed possible direction, as we shall see later in Section IV-B.

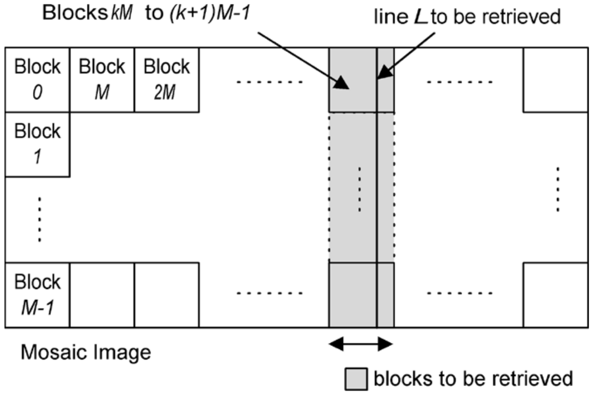

Fig. 10. Accessing a line L in a mosaic image.

Apart from this practical issue, successive mosaic images CM_k's also exhibit significant amount of redundancy, similar to that of conventional video sequences. Therefore, motion compensation techniques can be used to reduce such redundancy making the representation more compact. Using an MPEG-like codec, we are able to improve significantly the performance in compressing the mosaic images. The details are described later in Section IV-C.

B. Pointers Structure

As an illustration, let us consider the mosaic image in Fig. 10. For simplicity, we assume that the image is compressed by some block-based techniques, such as transform coding using the DCT [11]. The reasons for choosing a DCT-based codec are: 1) its reasonably good performance at a medium to high bit rate and 2) the availability of efficient software and hardware implementations. Other coding schemes can also be used after appropriate modifications, as suggested in the following, to achieve fast decoding. In Fig. 10, the image is divided into nonoverlapping blocks of size 8×8 (or 16×16 if MPEG-2 algorithm is used). Here, the blocks are scanned vertically so that pixel data of each vertical line are contained in a group of consecutive blocks. In order to retrieve the pixel data of line L, the compressed data of blocks kM to $(k+1)M-1$ have to be located and decoded. Due to the use of variable length coding mentioned earlier, each compressed group of blocks (GOBs) is of variable length. Locating the required data by searching the headers of the blocks can be very time consuming, especially for real-time rendering. To overcome this problem, a set of pointers to the starting locations of the vertical GOB in the compressed data is first determined and stored in an array prior to rendering. Alternatively, the pointers can be embedded in the compressed bit streams to avoid creating the pointer arrays when new mosaic images are loaded into the memory at the expense of slightly lower compression ratio. During rendering, the compressed data for the required GOB can be located very quickly. Fig. 11 illustrates how the pointers can be used to locate the required compressed data. In the next section, a coding scheme similar to the MPEG-2 standard [14] is used to reduce the spatial redundancy of the mosaic images.

C. Predicting Mosaic Images

As mentioned earlier, successive mosaic images have a significant amount of spatial redundancy. Thus, prediction techniques similar to motion estimation in video coding can be applied to exploit this redundancy. In this paper, a modified MPEG-2 video codec is used to compress the concentric mosaic. In the MPEG-2 algorithm, the I-pictures are coded using DCT-based coding without reference to other image frames. The P-pictures are predicted by motion compensation using the nearest encoded P- or I-pictures as references. The prediction residuals or the original pixels (if motion-compensation fails) are encoded using transform coding. Like I-pictures, P-pictures serve as prediction references for B-pictures and future P-pictures. B-pictures, which offer much higher compression ratio, are coded by bidirectional motion estimation and transform coding. By first transmitting and decoding the reference pictures, it is possible to decode each of the B-pictures in the group of pictures (GOPs) independently so as to provide efficient access to individual pictures. If each of the mosaic images is treated as a video frame, then it is possible to use the MPEG-2 algorithm to compress and decompress the concentric mosaic. In this paper, we investigate the coding of the mosaic images using 1) the multiperspective panoramas and 2) the image sequences obtained in the normal setup [Fig. 5(a)]. Instead of treating each mosaic image as an image frame, the multiperspective panoramas are used to better exploit spatial redundancy. The multiperspective panoramas are obtained by putting together the lines $L_i(\theta)$ at the same horizontal position, i, for successive images captured in the normal setup [see Fig. 12(b)]. This is illustrated in Fig. 7 for the concentric mosaics *Lobby* and *Kids*. Fig. 12(a) and (b) shows how the MPEG-2 algorithm can be used to compress the mosaic images and its

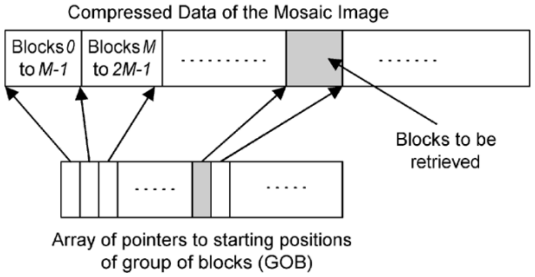
Compressed Data of the Mosaic Image
Blocks 0 to M-1
Blocks M to 2M-1
Blocks to be retrieved
Array of pointers to starting positions of group of blocks (GOB)

Fig. 11. Accessing the required group of blocks using a set of pointers.

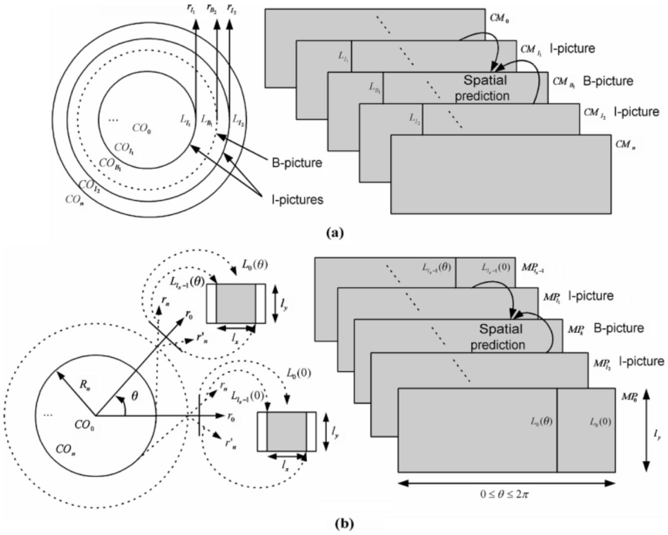

Fig. 12. Spatial prediction of mosaic images: (a) rays captured in the concentric mosaic and spatial prediction and (b) multiperspective panoramas (MP) and spatial prediction.

equivalent multiperspective panoramas, respectively. The main reason for choosing multiperspective panoramas is their larger spatial redundancy between successive panoramas. The correlation between successive mosaic images, on the other hand, is slightly lower because the rays are further apart when the radius of the circles decreases [see lines I_{I_1}, I_{B_1}, and I_{I_2} in Fig. 12(a)].

For an I-picture, the pointer structure mentioned earlier can be used to access the compressed data of a group of blocks (GOB). These pointers can either be embedded in the compressed bit stream or created during decoding. The former slightly decreases the compression ratio while the latter increases the loading time of the mosaic images. If B-pictures are added, for higher compression ratio, the pointer structure would only allow us to efficiently decode the motion vectors and the prediction residuals of that GOB. We still have to retrieve their predictors in the I-pictures. Since the predictors in the I-pictures are in general located in different GOBs, these blocks in the I-pictures also need to be decoded. The situation is even worse if P-pictures are involved because they are, in turns, predicted from previous P-pictures. One solution to this problem is to decode all the I- and P-pictures and save them in memory for later use, or implement a cache to store previously decoded pictures, at the expense of higher memory requirement and complexity. Hence, there is a tradeoff between rendering speed and the amount of compression that can be achieved. Fortunately, we found that using the multiperspective panoramas, the number of I- and P-pictures required can be significantly reduced. Out of the 320 multiperspective panoramas of the concentric mosaic *Lobby*, we only need two I-pictures at the beginning and the end, and seven P-pictures in between. Therefore, there are approximately 39 B-pictures between two P- or I-pictures. The P-pictures are decoded when the concentric mosaic is loaded into the memory so that fast random access can be achieved. Alternatively, we can replace them with I-pictures to reduce the loading time at the expense of slightly lower compression ratio. Another reason for the increased separation between the P- and I-pictures is that we have applied global motion estimation to the whole GOB in the multiperspective panoramas before carrying out the motion estimation for the macroblocks in the MPEG-2 algorithm. This reduces the search range in the motion estimation of the MPEG-2 algorithm and hence the data required to represent the motion vectors. More precisely, the block size used in the global motion estimation is 16×208, which consists of a vertical stripe as shown in Fig. 13. Because of the differences in field of view of the multiperspective panoramas, the upper and lower boundaries of a given multiperspective panorama do not, in general, appear in its previous neighbor, Fig. 13(a). Therefore, only the middle portion is used to estimate the global displacement (motion) vectors for a group of blocks in the "forward prediction," Fig. 13(a). The "backward prediction" in the multiperspective panoramas is done similarly and it is illustrated in Fig. 13(b). The global motion vectors are used as initial positions for carrying out the motion estimation of the macroblocks in the MPEG-2 algorithm. The differential motion vectors of the macroblocks in the vertical slice are coded using the MPEG-2 algorithm while the global motion vectors are stored in the modified GOB headers.

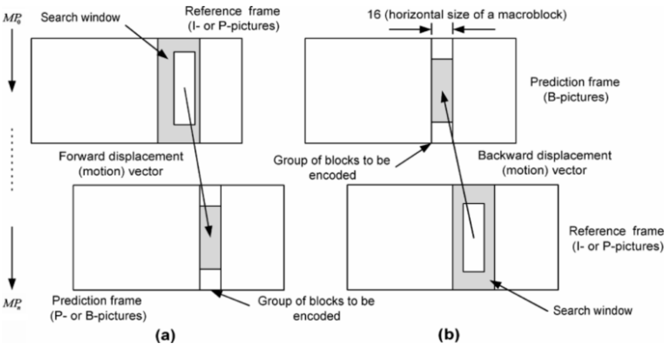

Fig. 13. Global motion estimation in multiperspective panoramas: (a) forward prediction and (b) backward prediction.

TABLE I
CODING RESULTS OF THE CONCENTRIC MOSAICS USING MULTIPERSPECTIVE PANORAMAS (Y COMPONENT). C. R.: COMPRESSION RATIO

Lobby **(Q=16)**			***Kids*** **(Q=24)**		
GOP structure	**C.R.**	**PSNR (dB)**	**GOP structure**	**C.R.**	**PSNR (dB)**
I only	24.28	33.67	I only	22.05	28.87
I 5B	43.22	34.52	I 5B	33.67	29.94
I 10B	46.50	34.53	I 10B	35.64	30.02
I 20B	45.03	34.54	I 20B	35.28	30.07
I 39B	42.01	34.50	I 43B	33.14	30.06

We have also studied the encoding of the image sequence obtained from the normal setup [Fig. 5(a)]. This can be viewed as a video sequence with a constant camera panning motion. There is more spatial correlation within the normal setup sequence. The compression ratio when using the normal setup sequence is higher than that of using multiperspective panoramas. The separation of the I-pictures, on the other hand, is much smaller than the former. This is due to the significant amount of uncovered scenes appearing in each image frame which cannot be predicted by bidirectional motion estimation when the I-pictures are set further apart. Therefore, it would require a significant amount of storage to decode all the I-pictures initially and load them into memory for fast rendering. Instead, we selectively decode the required GOBs together with those in the reference frames for their prediction. This means that the rendering speed is slightly slower than using multiperspective panoramas. For efficient rendering, we do not employ P-pictures in our algorithm, due to their inter-dependencies.

D. Experimental Results

The concentric mosaics *Lobby* and *Kids* described in Section III are compressed using the proposed coding algorithms. For simplicity, no rate control algorithm is applied and a uniform quantization scale factor (Q) is used for all the I- and B-pictures.

For the concentric mosaic *Lobby*, the multiperspective panoramas are represented as 320 24-bit true color panoramas, each having a resolution of 1350×240. Table I shows the compression results of the multiperspective panoramas. It can be seen that the coding performance is reasonably good even when the separation between successive I-pictures is increased to 39. Since there are only nine reference I-pictures, they can be

TABLE II
CODING RESULTS OF THE CONCENTRIC MOSAICS USING NORMAL SETUP SEQUENCE (Y COMPONENT)

GOP structure	*Lobby*				*Kids*			
	Q = 16		Q = 24		Q = 24		Q = 32	
	C.R.	PSNR (dB)	C.R.	PSNR (dB)	C.R.	PSNR (dB)	C.R.	PSNR (dB)
I only	30.49	35.15	39.76	33.15	29.48	31.07	36.19	29.67
I 1B	47.04	35.59	63.82	33.60	43.73	31.66	54.97	30.22
I 2B	56.90	35.72	79.25	33.74	50.70	31.79	64.94	30.33
I 3B	60.89	35.70	86.88	33.69	54.22	31.84	70.46	30.36
I 4B	63.92	35.74	92.80	33.74	55.61	31.83	73.09	30.34
I 5B	64.83	35.73	95.38	33.73	55.39	31.80	73.33	30.30
I 6B	65.28	35.72	97.08	33.71	54.24	31.77	72.22	30.25
I 7B	64.68	35.69	96.98	33.65	52.60	31.74	70.40	30.20
I 8B	63.63	35.66	96.11	33.61	50.91	31.71	68.36	30.16
I 9B	62.14	35.62	94.42	33.55	49.03	31.69	65.93	30.13

decoded and stored in the main memory for fast decoding of all the 320 image frames. The memory requirement for these reference I-pictures is only 8.34 MB ($1350 \times 240 \times 3 \times 9$ Bytes). It can also be seen that the proposed global motion compensation technique performs much better than direct application of the MPEG-2 algorithm. A peak signal-to-noise ratio (PSNR) of 34.50 dB can be achieved at a compression ratio of 42.01 (or 0.571 bpp). On the other hand, the compression ratio for the concentric mosaic *Kids* is much lower than that of *Lobby* at a given PSNR. It is because the former contains more details.

The normal setup sequence of the concentric mosaic *Lobby* consists of 1350 (320×240) images with 24-bit true color. Table II shows its compression results with Q equal to 16 and 24. If only I-pictures are used, a PSNR of 35.15 dB can be achieved at a compression ratio of 30.49 (or 0.787 bpp). The performances of using different combinations of I- and B-pictures are also given. It can be seen that using more B-pictures improves the coding performance when the separation between successive I-pictures are less than 6 for Q equal to 16 and 24. When the compression ratio is increased to 65.28 (or 0.368 bpp), using 6 B-pictures between two consecutive I-pictures improves the PSNR to 35.72 dB. This shows that there is a significant amount of inter-frame redundancy in the normal setup sequence. Also shown in Table II are the coding results for the concentric mosaic *Kids* with Q equal to 24 and 32. As a comparison, a compression ratio of about 20 can be achieved using vector quantization and entropy coding [10]. Therefore, it can be concluded that the proposed MPEG-2 based compression algorithms can provide much higher compression ratios than simple vector quantization while preserving the random access capability. The reference block coder (RBC) in [27] achieves a PSNR of 36.1 dB for the concentric mosaic *Lobby* and 31.5 dB for the concentric mosaic *Kids* at a compression ratio of 60 (or 0.4 bpp). It can be seen that the coding performance of the proposed modified MPEG-2 algorithm for the normal setup sequence is comparable to that of the RBC. On the other hand, the wavelet coder [28], [30] outperforms the RBC codec by about 3 dB in PSNR, but its decoding complexity is considerably higher.

TABLE III
RENDERING SPEED COMPARISON ON PENTIUM II 300-MHZ PC
(FRAMES PER SECOND)

Algorithm	352×168	800×372
VQ [10]	40.00	10.53
Multiperspective Panorama (I 39B)	41.67	10.87
Normal Setup Sequence (I 6B)	27.03	9.62

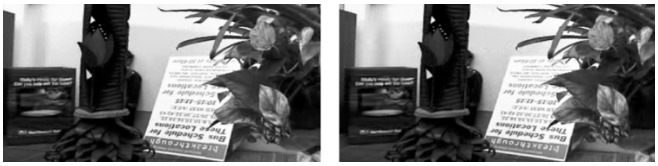

Fig. 14. Rendered views of the concentric mosaic *Lobby* after decompression: (a) using the normal setup sequence (compression ratio: 65.28; mean PSNR: 35.72 dB) and (b) using multiperspective panoramas (compression ratio: 42.01; mean PSNR: 34.50 dB).

Next, we compare the rendering speed of the proposed algorithms. In our experiments, there are, respectively, 39 and 6 B-pictures between successive I-pictures in coding the multiperspective panoramas and the normal setup sequence. Novel views at the center of the concentric circles are continuously rendered at an angular spacing of 0.006 radians up to 1000 different views. The averaged rendering speed of the three algorithms on Pentium II 300-MHz PC with 64-MB memory are shown in Table III. It can be seen that the rendering speed of using multiperspective panoramas is comparable to that of using VQ. The algorithm using the normal setup sequence is about 35% slower than the former, but it can still achieve a rendering speed of 27 frames per second at a resolution of 352×168. Fig. 14(a) and (b) shows the rendered views of the concentric mosaic *Lobby* obtained respectively from the compressed normal setup sequence and the multiperspective panoramas. It shows good quality reconstruction with a compression ratio of 65 and 42, respectively.

V. MULTIRESOLUTION REPRESENTATION OF THE CONCENTRIC MOSAIC

Multiresolution representation is an effective method for image data compression and progressive transmission [13]. It is usually performed by decomposing the images into different

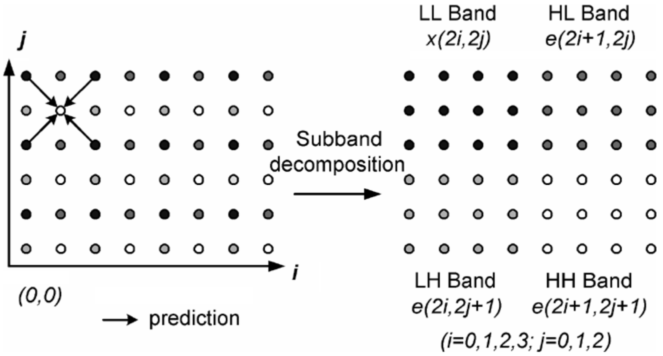

Fig. 15. One level of decomposition using the nonlinear filter bank. Samples marked with • are used to predict the samples marked where ○, • and •. $e(\cdot,\cdot)$ are the prediction residuals.

resolutions using perfect reconstruction (PR) filter banks. The biorthogonal $9/7$ wavelet transform [15] is commonly used because of its good coding performance. In this paper, we study the application of a nonlinear PR filter bank to provide a multiresolution representation for concentric mosaics. This is motivated by the nonlinear filter bank proposed in [20] for image compression. It has the advantages of very low implementation complexity and reduced blocking artifacts.

Fig. 15 shows one level of decomposition using the nonlinear filter bank in [20]. The low-low (LL) band is simply obtained by downsampling the original image by a factor of two both horizontally and vertically, and it consists of pixels at positions $(2i, 2j)$. The HH band consists of prediction residuals, $e(2i+1, 2j+1)$, when the pixels at positions $(2i+1, 2j+1)$ are predicted from its four quantized neighbors at positions $(2i, 2j), (2i+2, 2j), (2i, 2j+2)$, and $(2i+2, 2j+2)$. Similarly, the HL and LH bands consist of the prediction residuals when the pixels at $(2i+1, 2j)$ and $(2i, 2j+1)$ are predicted from their four nearest quantized neighbors, respectively. Different prediction schemes can be used. The one proposed in [20] is particularly simple which involves only four comparsions and two additions per pixel. More precisely, given the four quantized samples, the minimum and maximum values of the samples are first discarded. The predictor is then computed as the mean of the remaining two elements. A similar decomposition can be applied to the LL band so that a multiresolution decomposition of the image like an octave wavelet filter bank can be created. Another important advantage of using this nonlinear filter banks is that the reconstruction of a given pixel at a given resolution only requires its four adjacent neighbors. This property greatly simplifies the rendering of concentric mosaics because only very few additional samples are needed to reconstruct a given pixel. Furthermore, the decomposition, as we have seen previously, does not require any multiplications or floating-point numbers.

Such decomposition can also be utilized to improve the visual quality of the encoded images at very high compression ratios and to provide progressive transmission of concentric mosaics. At very low bit rate, the high-frequency subbands can simply be discarded in order to reduce the amount of information to be encoded and those overheads (such as headers) associated with commonly used video coding standards. Therefore, it is possible to make better use of the available bits so that a reasonable image quality can be achieved at a very low bit rate.

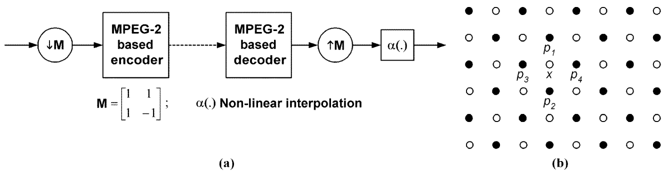
↓M
MPEG-2 based encoder
MPEG-2 based decoder
↑M
α(.)
$M = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$;
α(.) Non-linear interpolation
(a)
p_1
p_3
x
p_4
p_2
(b)

Fig. 16. (a) Simple decimation and nonlinear interpolation system. (b) Quincunx subsampling lattice. (Samples marked with • are used to interpolate the samples marked with ○).

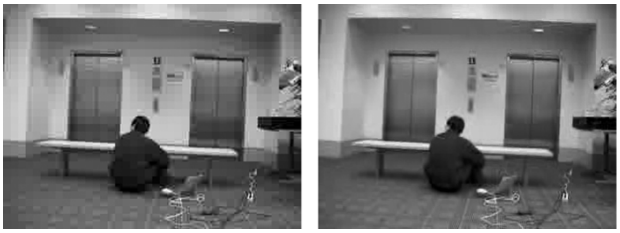

Fig. 17. Decompressed B-pictures using: (left) the modified MPEG-2 algorithm (compression ratio: 175.12; mean PSNR: 30.78 dB) and (right) the proposed subsampling compression scheme (compression ratio: 171.26; mean PSNR: 30.27 dB).

For simplicity, our nonlinear filter bank uses a quincunx subsampling lattice and it is applied to decimate the I-pictures in the normal setup sequence by a factor of two. The subsampled data is then compressed as I-pictures using the modified MPEG-2-based encoder as shown in Section IV [Fig. 16(a)]. To achieve a high compression ratio, the coefficients in the higher frequencies subband are simply discarded. The amount of data is therefore reduced by a factor of two. The reconstructed image is then interpolated using the simple median filter based reconstruction scheme [20] mentioned earlier to obtain the reference images for predicting the B-pictures in the modified MPEG-2 algorithm. The interpolation process is shown in Fig. 16(b), where the four immediate neighbors from the quincunx subsampled pixels $\{p_1, p_2, p_3, p_4\}$ are used to interpolate sample x using the operator $\alpha(\cdot)$. The minimum and maximum values of $\{p_1, p_2, p_3, p_4\}$ are discarded. The output is equal to the mean of the remaining two elements. This operation, though very simple, is able to enhance image edges using the subsampled image data. It can also be viewed as a post-processing operation for improving the visual quality of the reconstructed I-pictures. An alternative configuration is to decimate all the images in the sequence by a factor of two using the quincunx subsampling lattice. The resulting image sequence is then encoded using the modified MPEG-2 algorithm. It was found that the previous approach yields a higher PSNR than this configuration. This is largely due to the improved performance of the motion estimation of the former over the latter, which employs decimation. On the other hand, the second configuration yields a multiresolution decomposition of the image sequence and is more appropriate for progressive transmission. For simplicity, we only consider the first approach. To evaluate the effectiveness of the subsampling scheme, we first encode the normal setup sequence as I-pictures only. It was found that the mean PSNR of the modified MPEG-2 algorithm decrease rapidly at a high compression ratio. For compression ratios larger than 60, the subsampling scheme outperforms the modified MPEG-2 algorithm. At a compression of 70, the PSNR improvement is about 1 dB. In

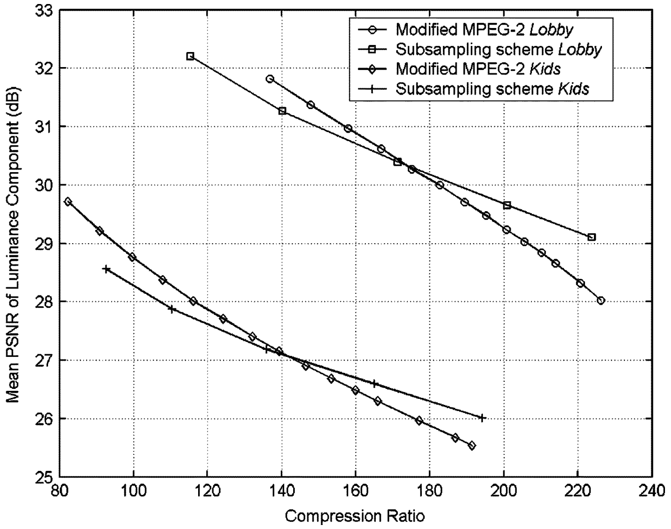

Fig. 18. Coding performances of the proposed subsampling compression scheme and the modified MPEG-2 algorithm (normal setup sequence).

Fig. 18, the overall performance of the subsampling scheme is compared to the modified MPEG-2 algorithm. The subsampling scheme outperforms the modified MPEG-2 algorithm when the compression ratio is larger than 175 for the concentric mosaic *Lobby* and 140 for the concentric mosaic *Kids*. To compare the visual quality of the two methods, Fig. 17 shows typical decompressed B-pictures for the concentric mosaic *Lobby*. Although the mean PSNR of the two methods are close to each other, it is observed that the visual quality of the subsampling scheme is better. There are less blocking artifacts and the edges are sharper. There are also less significant ringing artifacts frequently encountered in linear interpolation with long interpolation filters. At very low bit rate, however, there are insufficient bits for encoding the DCT coefficients. Artifacts, which appear as saw-tooth-like distortion at diagonal edges, may occur. Interested readers are referred to [29] for further application of this nonlinear filter bank to the scalable coding and transmission of concentric mosaics. Finally, it should be noted that other post-processing and coding schemes can also be employed to further improve the image quality.

VI. Conclusion

A new image-based rendering technique called "concentric mosaic" for VR applications is presented. The concentric mosaic is a 3-D plenoptic function with viewpoints constrained on a plane. Concentric mosaics allow the user to move freely in a circular region and observe significant parallax and lighting changes without recovering the geometric and photometric scene models. More importantly, concentric mosaics are very easy to capture. By simply spinning an off-centered camera on a rotary table, an entire concentric mosaic can be captured in about 10 min. The rendering of concentric mosaics is also very efficient, which amounts to reordering and interpolating of previously captured slit images in the concentric mosaic. Detailed examples are given to illustrate the usefulness of the proposed method. An MPEG-like compression algorithm is also presented for the compression of concentric mosaics. It explores the spatial redundancy of the mosaic images while satisfying the random access requirement of the slit images. Experimental results show that real-time reconstruction of novel views with good image quality can be achieved at 27–40 frames per second on a Pentium II 300-MHz PC with 64-MB memory. Using the modified MPEG-2 algorithm and a nonlinear filter bank, a multiresolution representation of concentric mosaics is also proposed. It can support the progressive transmission of concentric mosaics and improve the visual quality of the reconstructed images at high compression ratio.

Acknowledgment

The authors would like to thank the anonymous reviewers for carefully reading the manscript and their useful suggestions which greatly improve the presentation of this paper.

References

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

[2] S. E. Chen, "QuickTime VR—An image-based approach to virtual environment navigation," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'95)*, Aug. 1995, pp. 29–38.

[3] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry—and image-based approach," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'96)*, Aug. 1996, pp. 11–20.

[4] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'96)*, Aug. 1996, pp. 43–54.

[5] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'96)*, Aug. 1996, pp. 31–42.

[6] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. Proc. Annu. Conf. Computer Graphics (SIGGRAPH'95)*, Aug. 1995, pp. 39–46.

[7] S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'97)*, Jun. 1997, pp. 338–343.

[8] R. Szeliski and H. Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'97)*, Aug. 1997, pp. 251–258.

[9] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin, "Multiperspective panoramas for cel animation," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'97)*, Aug. 1997, pp. 243–250.

[10] H. Y. Shum and L. W. He, "Rendering with concentric mosaics," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'99)*, Aug. 1999, pp. 299–306.

[11] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Application to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[12] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1991.

[13] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[14] *Generic Coding of Moving Pictures and Associated Audio Information: Video*, ITU-T Recommendation H.262 ISO/IEC 13818-2, Nov. 1994.

[15] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[16] C. Machover and S. E. Tice, "Virtual reality," *IEEE Comput. Graph. Appl.*, vol. 14, no. 1, pp. 15–16, Jan. 1994.

[17] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'93)*, Aug. 1993, pp. 279–288.

[18] R. Gupta and R. Hartley, "Linear pushbroom cameras," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 9, pp. 963–975, Sep. 1997.

[19] P. Rademacher and G. Bishop, "Multiple-center-of-projection images," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'98)*, Jul. 1998, pp. 199–206.

[20] R. L. de Queiroz and D. A. F. Florencio, "A nonlinear filter bank for image coding," in *Proc. IEEE 38th Midwest Symp. Circuits and Systems*, vol. 2, 1996, pp. 1115–1118.

[21] M. Goebel, M. Hirose, and L. Rosenblum, "Today's VR," *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 22–24, Nov.–Dec. 2001.

[22] I. Peter and W. Strasser, "The wavelet stream—Progressive transmission of compressed light field data," in *Proc. IEEE Visualization'99*, Oct. 1999, pp. 69–72.

[23] 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.

[24] X. Tong and R. M. Gray, "Coding of multi-view images for immersive viewing," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, Jun. 2000, pp. 1879–1882.

[25] 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.

[26] H. Y. Shum, K. T. Ng, and S. C. Chan, "Virtual reality using the concentric mosaic: Construction, rendering, and data compression," in *Proc. IEEE Int. Conf. Image Processing (ICIP2000)*, vol. 3, Sep. 2000, pp. 644–647.

[27] C. Zhang and J. Li, "Compression and rendering of concentric mosaics with reference block codec (RBC)," in *Proc. SPIE Visual Communication and Image Processing*, vol. 4067, Jun. 2000.

[28] 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 and Graphics*, vol. 1, no. 1, pp. 45–61, 2001.

[29] K. T. Ng, S. C. Chan, and H. Y. Shum, "Scalable coding and progressive transmission of concentric mosaic using nonlinear filter banks," in *Proc. IEEE Int. Conf. Image Processing (ICIP2001)*, vol. 2, Oct. 2001, pp. 113–116.

[30] J. Li, H. Y. Shum, and Y. Q. Zhang, "On the compression of image based rendering scene," in *Proc. IEEE Int. Conf. Image Processing (ICIP2000)*, vol. 2, Sep. 2000, pp. 21–24.

[31] S. C. Chan and H. Y. Shum, "A spectral analysis for light field rendering," in *Proc. IEEE Int. Conf. Image Processing (ICIP2000)*, vol. 2, Sep. 2000, pp. 25–28.

[32] J. X. Chai, X. Tong, S. C. Chan, and H. Y. Shum, "Plenoptic sampling," in *Proc. Annu. Conf. Computer Graphics (SIGGRAPH'00)*, Jul. 2000, pp. 307–318.

Heung-Yeung Shum (SM'01) received the Ph.D. degree in robotics from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, in 1996.

He worked as a Researcher for three years in the vision technology group at Microsoft Research, Redmond, WA. In 1999, he moved to Microsoft Research Asia, Beijing, China, where he is currently the Managing Director. His research interests include computer vision, computer graphics, human computer interaction, action, pattern recognition, statistical learning and robotics.

Dr. Shum is on the editorial boards for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (CSVT), IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI), *International Journal of Computer Vision* (IJCV), and *Graphical Models*. He is the General Co-Chair of Tenth International Conference on Computer Vision (ICCV 2005, Beijing).

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

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

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

He joined City Polytechnic of Hong Kong in 1990 as an Assistant Lecturer and later as a University Lecturer. Since 1994, he has been with the Department of Electrical and Electronic Engineering, the University of Hong Kong, where he is currently an Associate Professor. He was a Visiting Researcher at Microsoft Corporation, Redmond, WA, in 1998 and at Microsoft China in 1999. His research interests include fast transform algorithms, filter design and realization, multirate signal processing, communications signal processing, and image-based rendering.

Dr. Chan is currently a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He was Chairman of the IEEE Hong Kong Chapter of Signal Processing from 2000 to 2002.