Survey of Image-Based Representations and Compression Techniques

Heung-Yeung Shum, *Senior Member, IEEE*, Sing Bing Kang, *Senior Member, IEEE*, and Shing-Chow Chan, *Member, IEEE*

Abstract—**In this paper, we survey the techniques for image-based rendering (IBR) and for compressing image-based representations. Unlike traditional three-dimensional (3-D) computer graphics, in which 3-D geometry of the scene is known, IBR techniques render novel views directly from input images. IBR techniques can be classified into three categories according to how much geometric information is used: rendering without geometry, rendering with implicit geometry (i.e., correspondence), and rendering with explicit geometry (either with approximate or accurate geometry). We discuss the characteristics of these categories and their representative techniques.**

IBR techniques demonstrate a surprising diverse range in their extent of use of images and geometry in representing 3-D scenes. We explore the issues in trading off the use of images and geometry by revisiting plenoptic-sampling analysis and the notions of view dependency and geometric proxies. Finally, we highlight compression techniques specifically designed for image-based representations. Such compression techniques are important in making IBR techniques practical.

Index Terms—**Image-based modeling, image-based rendering (IBR), image-based representations, survey.**

I. Introduction

IMAGE-BASED modeling and image-based rendering (IBR) techniques have received much attention as a powerful alternative to traditional geometry-based techniques for image synthesis. These techniques use images rather than geometry as primitives for rendering novel views. Previous surveys related to IBR have suggested characterizing a technique based on how image centric or geometry centric it is. This has resulted in the image-geometry continuum of image-based representations [33], [41].

For didactic purposes, we classify the various rendering techniques (and their associated representations) into three categories, namely: 1) rendering with no geometry; 2) rendering with implicit geometry; and 3) rendering with explicit geometry. These categories, depicted in Fig. 1, should actually be viewed as a continuum rather than absolute discrete ones since there are techniques that defy strict categorization.

At one end of the rendering spectrum, traditional texture mapping relies on very accurate geometric models, but only a few images. In an IBR system with depth maps, such as three-dimensional (3-D) warping [53], and layered-depth images (LDIs) [76], LDI trees [11], etc., the model consists of a set of images of a scene and their associated depth maps. The surface light field [88] is another geometry-based IBR representation that uses images and Cyberware scanned range data. When depth is available for every point in an image, the image can be rendered from any nearby point of view by projecting the pixels of the image to their proper 3-D locations and re-projecting them onto a new picture. For many synthetic environments or objects, depth is available. However, obtaining depth information from real images is hard even with state-of-art vision algorithms.

Some IBR systems do not require explicit geometric models. Rather, they require feature correspondence between images. For example, view interpolation techniques [12] generate novel views by interpolating optical flow between corresponding points. On the other hand, view morphing [75] results in in-between camera matrices along the line of two original camera centers based on point correspondences. Computer vision techniques are usually used to generate such correspondences.

At the other extreme, light-field rendering uses many images, but does not require any geometric information or correspondence. Light-field rendering [43] produces a new image of a scene by appropriately filtering and interpolating a pre-acquired set of samples. The Lumigraph [22] is similar to light-field rendering, but it uses approximated geometry to compensate for nonuniform sampling in order to improve rendering performance. Unlike light field and Lumigraph where cameras are placed on a two-dimensional (2-D) grid, the concentric mosaics (CMs) representation [77] reduces the amount of data by capturing a sequence of images along a circle path. In addition, it uses a very primitive form of a geometric impostor, whose radial distance is a function of the panning angle. (A geometric impostor is basically a 3-D shape used in IBR techniques to improve appearance prediction by depth correction. It is also known as geometric proxy.)

Since light-field rendering does not rely on any geometric impostors, it has a tendency to rely on oversampling to counter undesirable aliasing effects in output display. Oversampling means more intensive data acquisition, more storage, and higher redundancy.

What is the minimum number of images necessary to enable antialiased rendering? This fundamental issue needs to be addressed so as to avoid undersampling or unnecessary sampling. Sampling analysis in IBR, however, is a difficult problem because it involves unraveling the relationship among three elements, i.e., the depth and texture information of the scene, the

Manuscript received November 1, 2001; revised June 27, 2003.

H.-Y. Shum is with Microsoft Research Asia, Beijing 100080, China.

S. B. Kang is with Microsoft Research, Redmond, WA 98052-8300 USA (e-mail: sbkang@microsoft.com).

S.-C. Chan is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong.

Digital Object Identifier 10.1109/TCSVT.2003.817360

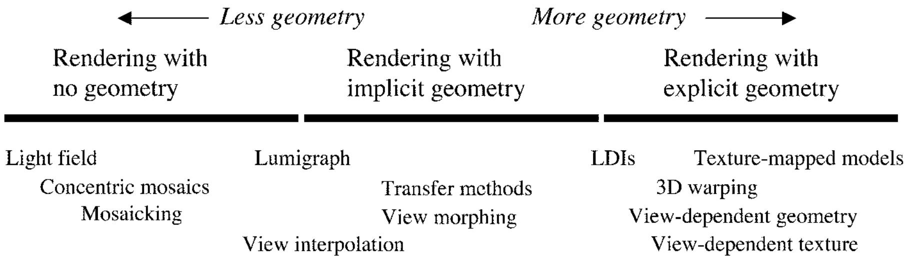

Fig. 1. Categories used in this paper, with representative members.

number of sample images, and the rendering resolution. Chai *et al.* showed in their plenoptic-sampling analysis [9] that the minimum sampling rate is determined by the depth variation of the scene. In addition, they showed that there is a tradeoff between the number of sample images and the amount of geometry (in the form of per-pixel depth) for antialiased rendering.

Since image-based representations are typically image intensive, compression becomes an important practical issue. Compression work has been traditionally carried out in the image and video communities, and many algorithms have been proposed to achieve high compression ratios. Image-based representations tend to have more local coherence than regular video because the captured appearance is that of the same static scene. However, they also have a significantly more complicated structure than regular video because the neighborhood of image samples may not be along a single axis (time axis only for regular video). For example, the Lumigraph is four-dimensional (4-D), and it uses a geometric impostor. Image-based representations also have special requirements of random access and selective decoding for fast rendering. As Sections II–VII will reveal, geometry has been used as a means for encoding coherency and compressing image-based representations.

The remainder of this paper is organized as follows. Three categories of IBR systems, with no, implicit, and explicit geometric information are, respectively, presented in Sections II–IV. The tradeoffs between the use of geometry and images for IBR are weighted in Section V. The issue of compression for IBR, with examples of light fields and CMs, is discussed in Section VI. We also discuss compact representation and efficient rendering techniques in Section VII, and provide concluding remarks in Section VIII.

II. Rendering With no Geometry

In this section, we describe representative techniques for rendering with unknown scene geometry. These techniques rely on the characterization of the plenoptic function.

A. Plenoptic Modeling

The original seven-dimensional (7-D) plenoptic function [1] is defined as the intensity of light rays passing through the camera center at every 3-D location (V_x, V_y, V_z) at every possible angle (θ, ϕ), for every wavelength λ, at every time t, i.e.,

$$P_7 = P(V_x, V_y, V_z, \theta, \phi, \lambda, t). \tag{1}$$

TABLE I
TAXONOMY OF PLENOPTIC FUNCTIONS

Dim.	Year	View space	Name
7	1991	free	Plenoptic function
5	1995	free	Plenoptic modeling
4	1996	bounding box	Lightfield/ Lumigraph
3	1999	bounding circle	Concentric Mosaics
2	1994	fixed point	Cylindrical/Spherical panorama

Adelson and Bergen [1] considered one of the tasks of early vision as extracting a compact and useful description of the plenoptic function's local properties (e.g., low-order derivatives). It has also been shown by Wong *et al.* [87] that light source directions can be incorporated into the plenoptic function for illumination control. By removing two variables, time t (therefore, static environment) and light wavelength λ, McMillan and Bishop [57] introduced the notion of plenoptic modeling with the five-dimensional (5-D) complete plenoptic function

$$P_5 = P(V_x, V_y, V_z, \theta, \phi). \tag{2}$$

The simplest plenoptic function is a 2-D panorama (cylindrical [13] or spherical [84]) when the viewpoint is fixed as follows:

$$P_2 = P(\theta, \phi). \tag{3}$$

A regular rectilinear image with a limited field-of-view can be regarded as an incomplete plenoptic sample at a fixed viewpoint.

IBR can be viewed as a set of techniques to reconstruct a continuous representation of the plenoptic function from observed discrete samples. The issues of sampling the plenoptic function and reconstructing a continuous function from discrete samples are important research topics in IBR. As a preview, a taxonomy of plenoptic functions is shown in Table I.

The cylindrical panoramas used in [57] are 2-D samples of the plenoptic function in two viewing directions. The two viewing directions for each panorama are panning and tilting about its center. This restriction can be relaxed if geometric information about the scene is known. In [57], stereo techniques are applied on multiple cylindrical panoramas in order to extract disparity (or inverse depth) distributions. These distributions can then be used to predict appearance (i.e., plenoptic function) at arbitrary locations. Similar work on regular stereo pairs can be found in

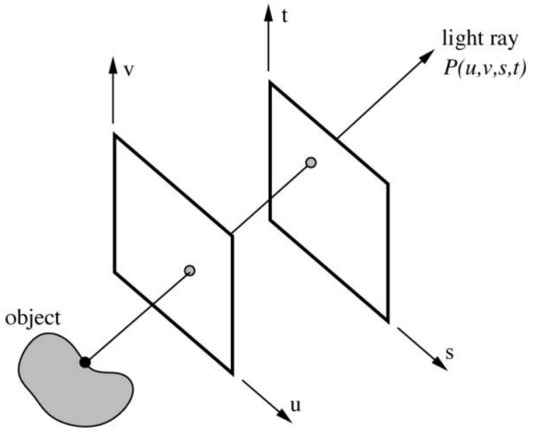

Fig. 2. Representation of a light field.

[39], where correspondences constrained along epipolar geometry are directly used for view transfer.

B. Light Field and Lumigraph

It was observed in both light-field rendering [43] and Lumigraph [22] systems that as long as we stay outside the convex hull (or simply a bounding box) of an object,[1] we can simplify the 5-D complete plenoptic function to a 4-D light-field plenoptic function

$$P_4 = P(u, v, s, t) \tag{4}$$

where (u, v) and (s, t) are parameters of two planes of the bounding box, as shown in Fig. 2. Note that these two planes need not be parallel. There is also an implicit and important assumption that the strength of a light ray does not change along its path. For a complete description of the plenoptic function for the bounding box, six sets of such two-planes would be needed. More restricted versions of Lumigraph have also been developed by Sloan *et al.* [81] and Katayama *et al.* [37]. Here, the camera motion is restricted to a straight line.

The principles of light-field rendering and Lumigraph are the same, except that the Lumigraph has the additional (approximate) object geometry for better compression and appearance prediction. In the light-field system, a capturing rig is designed to obtain uniformly sampled images. To reduce the aliasing effect, the light field is pre-filtered before rendering. A vector quantization (VQ) scheme is used to reduce the amount of data used in light-field rendering, while achieving random access and selective decoding. On the other hand, the Lumigraph can be constructed from a set of images taken from arbitrarily placed viewpoints. A re-binning process (in this case, resampling to a regular grid using a hierarchical interpolation scheme) is, therefore, required. Geometric information is used to guide the choices of the basis functions. Due to the use of geometric information, the sampling density can be reduced. Note that we place the Lumigraph in the category of “no geometry” because it is primarily image based, with geometry playing a secondary (optional) role.

The $P_4 = P(u, v, s, t)$ two-plane parameterization is just one of many for light fields. Other types of light fields include spherical or isotropic light fields [7], [24], sphere-plane light fields [7], and hemispherically arranged light fields with geometry [51]. The issue of uniformly sampling the light field was investigated by Camahort [6]. He introduced an isotropic parameterization that he calls the direction-and-point parameterization (DPP), and showed that while no parameterization is view independent, only the DPP introduces a single bias.

Buehler *et al.* [5] extended the light-field concept through a technique that uses geometric proxies (if available), handles unstructured input, and blends textures based on relative angular position, resolution, and field-of-view. They achieve real-time rendering by interpolating the blending field using a sparse set of locations.

C. CMs

Obviously, the more constraints we have on the camera location (V_x, V_y, V_z), the simpler the plenoptic function becomes. If we want to capture all viewpoints, we need a complete 5-D plenoptic function. As soon as we stay in a convex hull (or, conversely, viewing from a convex hull) free of occluders, we have a 4-D light field. If we do not translate at all, we have a 2-D panorama. An interesting 3-D parameterization of the plenoptic function, called CMs [77], was proposed by Shum and He; here, the sampling camera motion is constrained along concentric circles on a plane.

By constraining camera motion to planar concentric circles, CMs can be created by compositing slit images taken at different locations of each circle. CMs index all input image rays naturally in three parameters: radius, rotation angle, and vertical elevation. Novel views are rendered by combining the appropriate captured rays in an efficient manner at rendering time. Although vertical distortions exist in the rendered images, they can be alleviated by depth correction. CMs have good space and computational efficiency. Compared with a light field or Lumigraph, CMs have much smaller file size because only a 3-D plenoptic function is constructed.

Most importantly, CMs are very easy to capture. Capturing CMs is as easy as capturing a traditional panorama, except that CMs require more images. By simply spinning an off-centered camera on a rotary table, Shum and He [77] were able to construct CMs for a real scene in approximately 10 min. Like panoramas, CMs do not require the difficult modeling process of recovering geometric and photometric scene models. Yet CMs provide a much richer user experience by allowing the user to move freely in a circular region and observe significant parallax and lighting changes. (Parallax refers to the apparent relative change in object location within a scene due to a change in the camera viewpoint.) The ease of capturing makes CMs very attractive for many virtual reality applications.

Rendering of a lobby scene from captured CMs is shown in Fig. 3. A rebinned CM at the rotation center is shown in Fig. 3(a), while two rebinned CMs taken at exactly opposite directions are shown in Fig. 3(b) and (c), respectively. It has also been shown in [67] that such two mosaics taken from a single rotating camera can simulate a stereo panorama. In Fig. 3(d), strong parallax can be seen between the plant and poster in the

[1] The reverse is also true if camera views are restricted inside a convex hull.

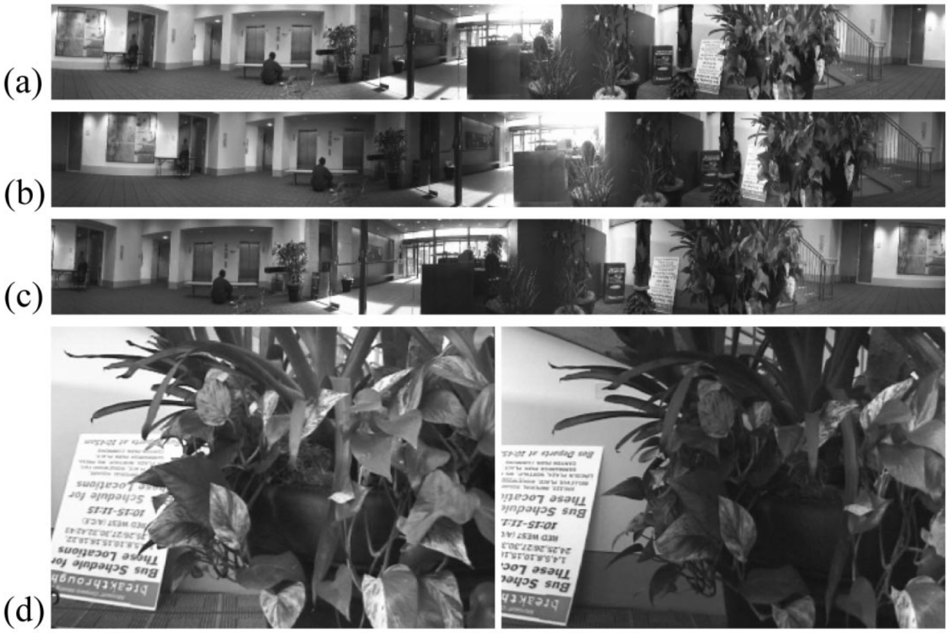

Fig. 3. Rendering a lobby. Rebinned CM: (a) at the rotation center, (b) at the outermost circle, and (c) at the outermost circle, but looking at the opposite direction of (b). (d) Parallax change between the plant and poster.

rendered images. More specifically, in the left image, the poster is partially obscured by the plant, while the poster and plant do not visually overlap in the right image. This is a significant visual cue that the camera viewpoint has shifted.

D. Image Mosaicing

A complete plenoptic function at a fixed viewpoint can be constructed from incomplete samples. Specifically, a panoramic mosaic is constructed by registering multiple regular images. For example, if the camera focal length is known and fixed, one can project each image to its cylindrical map and the relationship between the cylindrical images becomes a simple translation. For arbitrary camera rotation, one can first register the images by recovering the camera movement before converting to a final cylindrical/spherical map.

Many systems have been built to construct cylindrical and spherical panoramas by stitching multiple images together, e.g., [13], [52], [57], [83], and [84], among others. When the camera motion is very small, it is possible to put together only small strips from registered images, i.e., slit images (e.g., [68] and [98]) to form a large panoramic mosaic. Capturing panoramas is even easier if omnidirectional cameras (e.g., [60] and [61]) or fisheye lens [91] are used.

Szeliski and Shum [84] presented a complete system for constructing *panoramic image mosaics* from sequences of images. Their mosaic representation associates a transformation matrix with each input image rather than explicitly projecting all of the images onto a common surface, such as a cylinder. In particular, to construct a full view panorama, a *rotational mosaic* representation associates a rotation matrix (and optionally a focal length) with each input image. A *patch-based alignment* algorithm is developed to quickly align two images given motion models. Techniques for estimating and refining camera focal lengths are also presented.

In order to reduce accumulated registration errors, global alignment through block adjustment is applied to the whole sequence of images, which results in an optimally registered image mosaic. To compensate for small amounts of motion parallax introduced by translations of the camera and other unmodeled distortions, a local alignment (*deghosting*) technique [80] warps each image based on the results of pairwise local image registrations. Combining both global and local alignment significantly improves the quality of image mosaics, thereby enabling the creation of full view panoramic mosaics with handheld cameras.

A tessellated spherical map of the full view panorama is shown in Fig. 4. Three panoramic image sequences of a building lobby were taken with the camera on a tripod tilted at three different angles. 22 images were taken for the middle sequence, 22 images for the upper sequence, and ten images for the top sequence. The camera motion covers more than two-thirds of the viewing sphere, including the top.

Apart from blending images to directly produce wider fields of view, one can use the multiple images to generate higher resolution panoramas as well (e.g., using maximum-likelihood algorithms [25] or learnt image models [8]).

III. Rendering With Implicit Geometry

There is a class of techniques that relies on positional correspondences across a small number of images to render new views. This class has the term *implicit* to express the fact that geometry is not directly available; 3-D information is computed only using the usual projection calculations. New views are computed based on direct manipulation of these positional correspondences, which are usually point features.

The approaches under this class are view interpolation, view morphing, and transfer methods. View interpolation uses general dense optic flow to directly generate intermediate views.

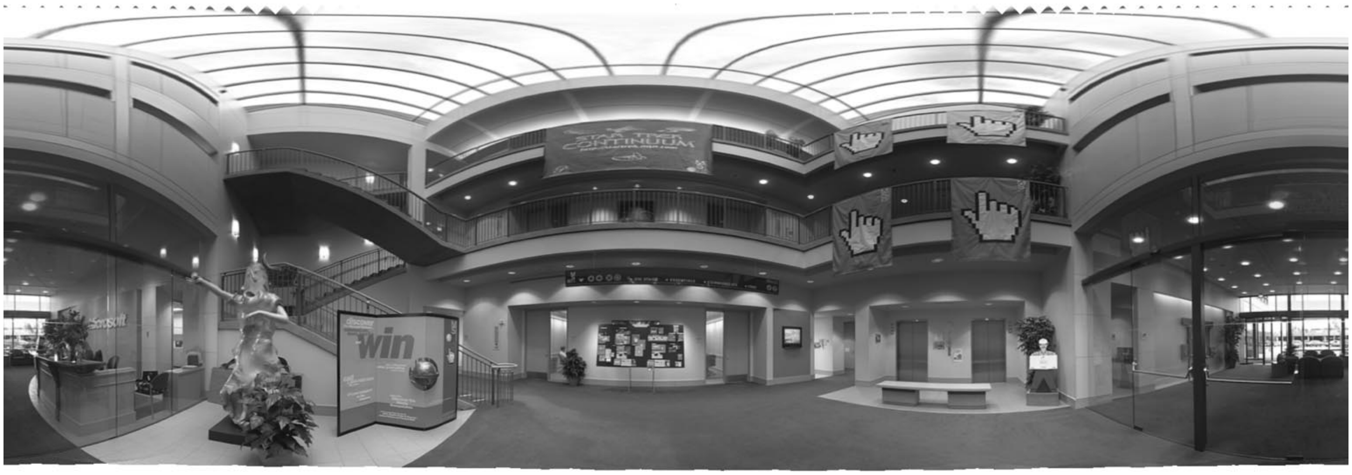

Fig. 4. Tessellated spherical panorama covering the north pole (constructed from 54 images).

The intermediate view may not necessarily be geometrically correct. View morphing is a specialized version of view interpolation, except that the interpolated views are always geometrically correct. The geometric correctness is ensured because of the linear camera motion. Transfer methods are also produce geometrically correct views, except that the camera viewpoints can be arbitrarily positioned.

A. View Interpolation

Chen and Williams' view interpolation method [12] is capable of reconstructing arbitrary viewpoints given two input images and dense optical flow between them. This method works well when two input views are close by so that visibility ambiguity does not pose a serious problem. Otherwise, flow fields have to be constrained so as to prevent foldovers. In addition, when two views are far apart, the overlapping parts of two images may become too small. Chen and Williams' approach works particularly well when all the input images share a common gaze direction, and the output images are restricted to have a gaze angle less than 90°.

Establishing flow fields for view interpolation can be difficult, particularly for real images. Computer vision techniques such as feature correspondence or stereo must be employed. For synthetic images, flow fields can be obtained from the known depth values.

B. View Morphing

From two input images, Seitz and Dyer's view morphing technique [75] reconstructs any viewpoint on the line linking two optical centers of the original cameras. Intermediate views are exactly linear combinations of two views only if the camera motion associated with the intermediate views are perpendicular to the camera viewing direction. If the two input images are not parallel, a pre-warp stage can be employed to rectify two input images so that corresponding scan lines are parallel. Accordingly, a post-warp stage can be used to un-rectify the intermediate images. Scharstein [74] extends this framework to camera motion in a plane. He assumes, however, that the camera parameters are known.

In a more recent work, Aliaga and Carlbom [2] describe an interactive virtual walkthrough system that uses a large network of omnidirectional images taken within a 2-D plane. To construct a view, the system uses the closest set of images, warps them using precomputed corresponding features, and blends the results.

C. Transfer Methods

Transfer methods (a term used within the photogrammetric community) are characterized by the use of a relatively small number of images with the application of geometric constraints (either recovered at some stage or known *a priori*) to reproject image pixels appropriately at a given virtual camera viewpoint. The geometric constraints can be of the form of known depth values at each pixel, *epipolar constraints* between pairs of images, or *trifocal/trilinear tensors* that link correspondences between triplets of images. The view interpolation and view morphing methods above are actually specific instances of transfer methods.

Laveau and Faugeras [40] use a collection of images called reference views and the principle of the fundamental matrix to produce virtual views. The new viewpoint, which is chosen by interactively choosing the positions of four control image points, is computed using a reverse-mapping or raytracing process. For every pixel in the new target image, a search is performed to locate the pair of image correspondences in two reference views. The search is facilitated by using the epipolar constraints and the computed dense correspondences (also known as image disparities) between the two reference views.

Note that if the camera is only weakly calibrated, the recovered viewpoint will be that of a projective structure (see [20] for more details). This is because there is a class of 3-D projections and structures that will result in exactly the same reference images. Since angles and areas are not preserved, the resulting viewpoint may appear warped. Knowing the internal parameters of the camera removes this problem.

If a trilinear tensor, which is a $3 \times 3 \times 3$ matrix, is known for a set of three images, then given a pair of point correspondences in two of these images, a third corresponding point can be directly computed in the third image without resorting to any projection

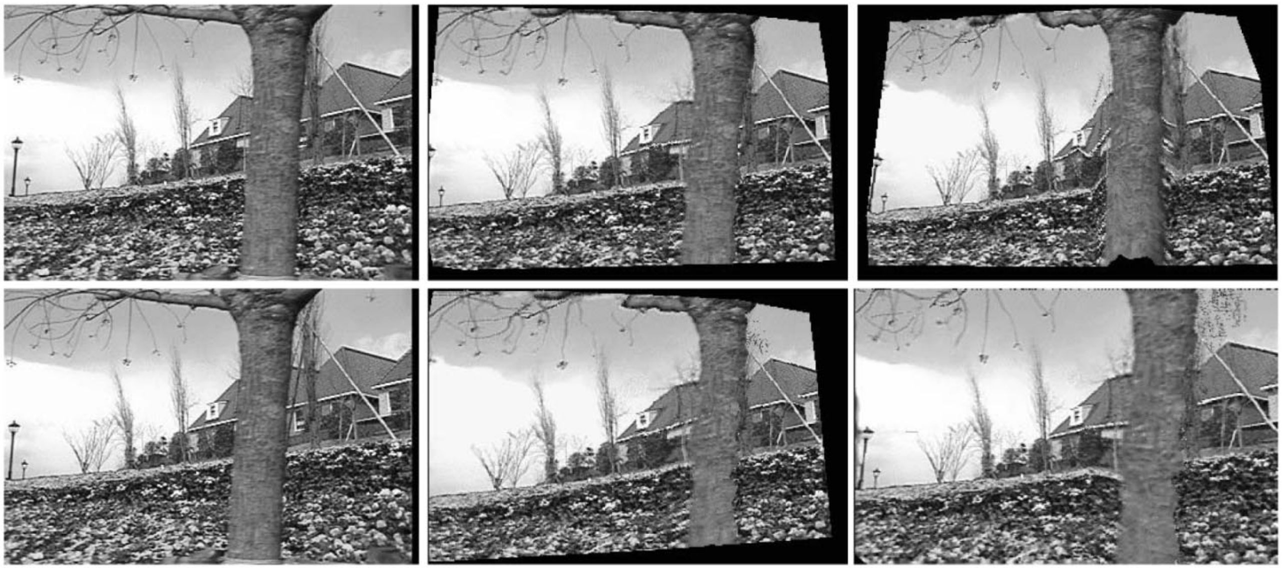

Fig. 5. Example of visualizing using the trilinear tensor. The left-most two images are the reference images, with the rest synthesized at arbitrary viewpoints.

computation. This idea has been used to generate novel views from either two or three reference images [3].

The idea of generating novel views from two or three reference images is rather straightforward. First, the "reference" trilinear tensor is computed from the point correspondences between the reference images. In the case of only two reference images, one of the images is replicated and regarded as the "third" image. If the camera intrinsic parameters are known, then a new trilinear tensor can be computed from the known pose change with respect to the third camera location. The new view can subsequently be generated using the point correspondences from the first two images and the new trilinear tensor. A set of novel views created using this approach can be seen in Fig. 5.

IV. Rendering With Explicit Geometry

In this class of techniques, the representation has direct 3-D information encoded in it, either in the form of depth along known lines of sight, or 3-D coordinates. The more traditional 3-D texture-mapped model belongs to this category (not described here since its rendering uses the conventional graphics pipeline).

In this category, we have 3-D warping, layered depth image (LDI) rendering, and view-dependent texture mapping. 3-D warping is applied to depth per-pixel representations such as sprites. LDIs are extensions of depth per-pixel representations since they can encode multiple depths along a given ray. View-dependent texture mapping refers to mapping multiple texture maps to the same 3-D surface and averaging their colors based on the current viewpoint relative to the sampled viewpoints.

A. 3-D Warping

When the depth information is available for every point in one or more images, 3-D warping techniques (e.g., [56]) can be used to render nearly all viewpoints. An image can be rendered from any nearby point of view by projecting the pixels of the original image to their proper 3-D locations and re-projecting them onto the new picture. The most significant problem in 3-D warping is how to deal with holes generated in the warped image. Holes are due to the difference of sampling resolution between the input and output images, and the disocclusion where part of the scene is seen by the output image, but not by the input images. To fill in holes, the most commonly used method is to map a pixel in the input image to several pixels size in the output image. This process is called *splatting*.

1) Relief Texture: To improve the rendering speed of 3-D warping, the warping process can be factored into a relatively simple pre-warping step and a traditional texture-mapping step. The texture-mapping step can be performed by standard graphics hardware. This is the idea behind relief texture, a rendering technique proposed by Oliveira and Bishop [66]. A similar factoring approach has been proposed by Shade *et al.* in a two-step algorithm [76], where the depth is first forward warped before the pixel is backward mapped onto the output image.

2) Multiple-Center-of-Projection (MCOP) Images: The 3-D warping techniques can be applied not only to the traditional perspective images, but also multiperspective images as well. For example, Rademacher and Bishop [72] proposed to render novel views by warping MCOP images.

B. LDI Rendering

To deal with the disocclusion artifacts in 3-D warping, Shade *et al.* proposed LDI [76] to store not only what is visible in the input image, but also what is behind the visible surface. In their paper, the LDI is constructed either using stereo on a sequence of images with known camera motion (to extract multiple overlapping layers) or directly from synthetic environments with known geometries. In an LDI, each pixel in the input image contains a list of depth and color values where the ray from the pixel intersects with the environment.

Though an LDI has the simplicity of warping a single image, it does not consider the issue of sampling density. Chang *et al.* [11] proposed LDI trees so that the sampling rates of the reference images are preserved by adaptively selecting an LDI in the LDI tree for each pixel. While rendering the LDI tree, only the level of LDI tree that is the comparable to the sampling rate of the output image need to be traversed.

C. View-Dependent Texture Mapping

Texture maps are widely used in computer graphics for generating photo-realistic environments. Texture-mapped models can be created using a computer-aided design (CAD) modeler for a synthetic environment. For real environments, these models can be generated using a 3-D scanner or applying computer vision techniques to captured images. Unfortunately, vision techniques are not robust enough to recover accurate 3-D models. In addition, it is difficult to capture visual effects such as highlights, reflections, and transparency using a single texture-mapped model.

To obtain these visual effects of a reconstructed architectural environment, Debevec *et al.* [16] used view-dependent texture mapping to render new views by warping and compositing several input images of an environment. This is the same as conventional texture mapping, except that multiple textures from different sampled viewpoints are warped to the same surface and averaged, with weights computed based on proximity of the current viewpoint to the sampled viewpoints. A three-step view-dependent texture-mapping method was also proposed later by Debevec *et al.* [15] to further reduce the computational cost and to have smoother blending. This method employs visibility preprocessing, polygon-view maps, and projective texture mapping. More recently, Buehler *et al.* [5] apply a more principled way of blending textures based on relative angular position, resolution, and field-of-view.

V. Tradeoff Between Images and Geometry

Rendering with no geometry is expensive in terms of acquiring and storing the database. On the other hand, using explicit geometry, while more compact, may compromise output visual quality. Thus, an important question is, what is the right mix of image sampling size and quality of geometric information required to satisfy a mix of quality, compactness, and speed? Part of that question may be answered by analyzing the nature of plenoptic sampling.

A. Plenoptic-Sampling Analysis

Many IBR systems, especially light-field rendering [22], [43], [77], have a tendency to rely on oversampling to counter undesirable aliasing effects in output display. Oversampling means more intensive data acquisition, more storage, and more redundancy. Sampling analysis in IBR is a difficult problem because it involves unraveling the relationship among three tightly related elements: the depth and texture information of the scene, the number of sample images, and the rendering resolution, as shown in Fig. 6. The presence of nonrigid effects (such as highlights, inter-reflection, and translucency) significantly complicates this analysis, and is typically ignored. Nonrigid effects would very likely result in higher image sampling requirements than those predicted by analyses that ignore such effects.

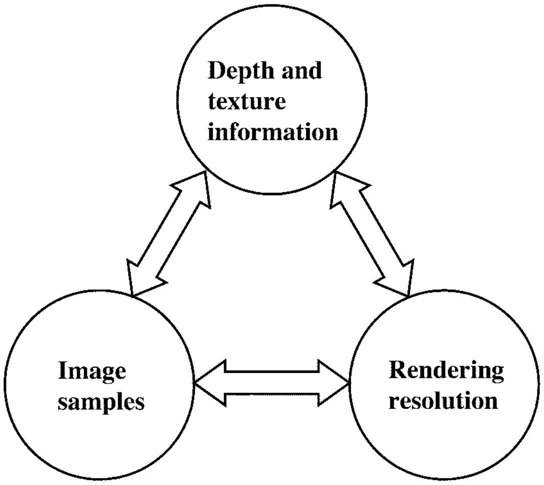

Fig. 6. Plenoptic sampling. Quantitative analysis of the relationships among three key elements: depth and texture information, number of input images, and rendering resolution.

Chai *et al.* [9] recently studied the issue of *plenoptic sampling*. More specifically, they were interested in determining the number of image samples (e.g., from a 4-D light field) and the amount of geometric and textural information needed to generate a continuous representation of the plenoptic function. The following two problems are studied under plenoptic sampling: 1) finding the minimum sampling rate for light-field rendering and 2) finding the minimum sampling curve in the joint image and geometry space.

Chai *et al.* formulate the question of sampling analysis as a high-dimensional signal-processing problem. Rather than attempting to obtain a closed-form general solution to the 4-D light-field spectral analysis, they only analyze the bounds of the spectral support of the light-field signals. A key observation in this paper is that the spectral support of a light-field signal is bounded by only the minimum and maximum depths, irrespective of how complicated the spectral support might be because of depth variations in the scene. Given the minimum and maximum depths, a reconstruction filter with an optimal and constant depth can be designed to achieve antialiased light-field rendering.

The minimum sampling rate of light-field rendering is obtained by compacting the replicas of the spectral support of the sampled light field within the smallest interval after the optimal filter is applied. How small the interval can be depends on the design of the optimal filter. More depth information results in tighter bounds of the spectral support, thus a smaller number of images. Plenoptic sampling in the joint image and geometry space determines the minimum sampling curve, which quantitatively describes the relationship between the number of images and the information on scene geometry under a given rendering resolution. This minimal sampling curve can serve as one of the design principles for IBR systems. Furthermore, it bridges the gap between IBR and traditional geometry-based rendering.

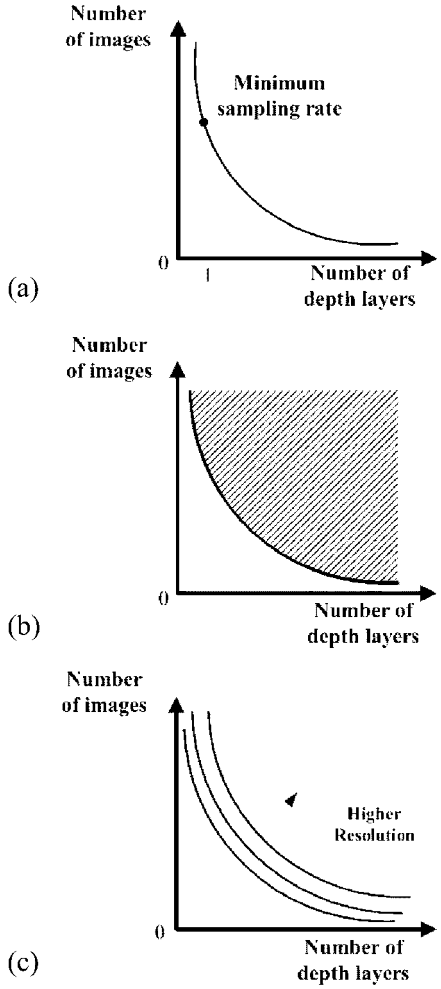

Fig. 7. Minimum sampling. (a) The minimum sampling rate in image space. (b) The minimum sampling curve in the joint image and geometry space. (c) Minimum sampling curves at different rendering resolutions.

Minimum sampling rate and minimum sampling curves are illustrated in Fig. 7. Note that this analysis ignores the effects of both occlusion events and nonrigid motion.

As shown in Fig. 7(a), a minimum sampling rate (i.e., the minimum number of images) can be obtained if only minimum and maximum depths of the scene are known. Fig. 7(b) illustrates that any sampling point above the minimum sampling curve is redundant. Reference [9, Fig. 11] demonstrated that the rendered images with five sampling points (of the number of images and the number of depth layers) above the minimum sampling curve are visually indistinguishable. Such a minimum sampling curve is also related to the rendering resolution, as shown in Fig. 7(c).

Isaksen *et al.* [26] did a similar analysis in frequency domain in the context of their work on dynamically reparameterized light fields. Here, they were concerned about the effect of variable focus and depth-of-field on output quality. Zhang and Chen [94] extended the IBR sampling analysis by proposing a generalized sampling strategy to replace the conventional rectangular sampling in the high dimensional signal space. Their analysis was performed in continuous and discrete spatial domains.

There are a number of techniques that can be applied to reduce the size of the representation; they are usually based on local coherency either in the spatial or temporal domains. Sections V-B–D describe some of these techniques.

B. View-Dependent Geometry

Another interesting representation that trades off geometry and images is view-dependent geometry, first used in the context of 3-D cartoons [71]. We can potentially extend this idea to represent real or synthetically generated scenes more compactly. As described in [36], view-dependent geometry is useful to accommodate the fact that stereo reconstruction errors are less visible during local viewpoint perturbations, but may show dramatic effects over large view changes. In areas where stereo data is inaccurate, they suggest that we may well represent these areas with view-dependent geometry, which comprises a set of geometry extracted at various positions (in [71], this set is manually created).

View-dependent geometry may also be used to capture visual effects such as highlights and transparency, which are likely to be locally coherent in image and viewpoint spaces. This is demonstrated in the work described in [23], in which structure from motion is first automatically computed from input images acquired using a camera following a serpentine path (raster style left to right and top to bottom). The system then generates local depth maps and textures used to produce new views in a manner similar to the Lumigraph [22]. The important issue of automatically determining the minimum amount of local depth maps and textures required has yet to be resolved. This area should be a fertile one for future investigation with potentially significant payoffs.

C. Dynamically Reparameterized Light Field

Recently, Isaksen *et al.* [26] proposed the notion of dynamically reparameterized light fields by adding the ability to vary the apparent focus within a light field using variable aperture and focus ring. Compared with the original light field and Lumigraph, this method can deal with a much larger depth variation in the scene by combining multiple focal planes. Therefore, it is suitable not only for outside-looking-in objects, but also for inside-looking-out environments. When multiple focus planes are used for a scene, a scoring algorithm is used before rendering to determine which focus plane is used during rendering.

While this method does not need to recover actual or approximate geometry of the scene for focusing, it does need to assign which focus plane to be used. The number of focal planes needed is not discussed. This light-field variant exposes another factor that needs to be considered in the tradeoff, i.e., the ability to vary the apparent focus on the scene (the better the focus/defocus effect required, the more image samples needed). It is not currently clear, though, how this need can be quantified in the tradeoff.

D. Geometric Proxies

Many approximated geometric models, or geometric proxies have been proposed in various IBR systems in order to reduce

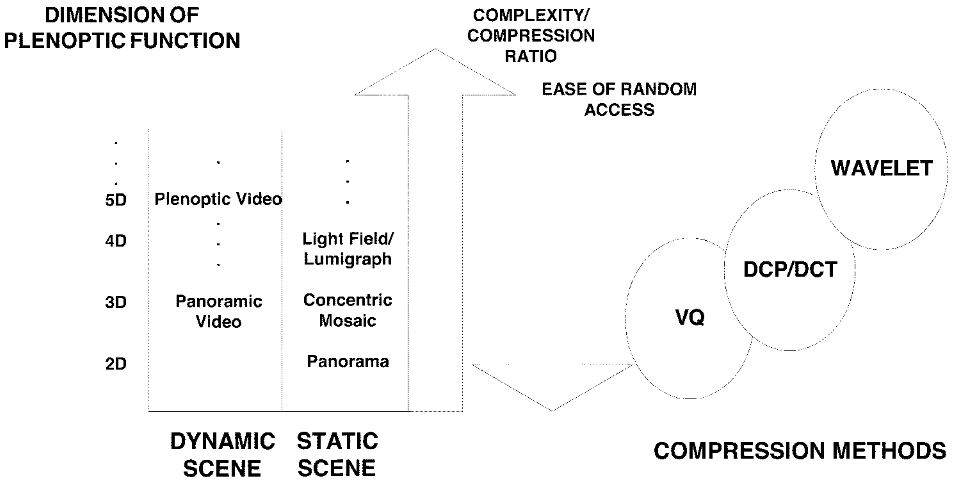

Fig. 8. Comparison of different image-based representation and compression methods in terms of their complexity. The ease of random access increases as the dimension of plenoptic function decreases, while the complexity and potential for compression both increase with the dimension. DCP refers to DCP methods while DCT is the DCT (see Section VI).

the number of images needed for antialiased rendering. Light field, dynamically reparameterized light field, and CMs have used simple planar surfaces. The Lumigraph used an approximated model extracted using "shape-from-silhouette." The unstructured Lumigraph work demonstrated that realistic rendering can be achieved, although the proxies are significantly different from the true models. The image-based visual hull [54] is another geometry proxy that can be constructed and updated in real time.

Acquiring an adequate geometric proxy is, however, difficult when the sampling of light field is very sparse. The geometric proxy, albeit approximate, needs to be continuous because every desired ray must intersect some point on the proxy in order to establish the correspondence between rays. Traditional stereo reconstruction unfortunately cannot provide accurate enough geometric proxies especially at places where occlusion happens. Scam light-field rendering [93] has been recently proposed to build a geometric proxy using only sparse correspondence.

VI. Compression of IBR Representations

Thus far, we have discussed the characteristics of various types of image-based representations, as well as rendering issues. It is clear that image-intensive representations such as light fields, Lumigraphs, and CMs are capable of photo-realistic rendering, but this is achieved at the expense of large storage and transmission bandwidth. To overcome these problems, a significant amount of work has been done on effective compression and transmission of image-based representations. Although image and video compression have been studied extensively and many advanced algorithms and international standards are now available [27]–[31], there are specific important requirements in IBR that need to be addressed.

A. IBR Requirements

First of all, image-intensive representations are usually densely sampled higher dimensional signals (see Table I). Their data sizes are huge, but their samples are highly correlated. Direct application of traditional compression algorithms, however, usually results in sub-optimal performance. Providing random access to the compressed data for real-time rendering is another important and unique problem of IBR compression. Unlike video coding, which supports random access at the picture or group of picture (GOP) level, higher dimensional IBR representations such as 3-D CMs requires random access at the line level, whereas the 4-D light field and Lumigraph require random access at the pixel level. As most existing compression algorithms employ entropy coding (such as Huffman or arithmetic coding) for better compression ratio, the symbols after compression are of variable sizes. It is, therefore, very time consuming to retrieve and decode a single line or pixel from the compressed data if there is no such provision for random access.

In addition, it is often impossible to decode the complete bit stream of a high dimensional representation in main memory for rendering due to its large data sizes. For example, the 3-D CMs of the lobby scene (Fig. 3) require 297 MB of RAM. To overcome this problem, VQ [77], [78] or just-in-time (JIT) decoding [44], [96] is usually used. Only those lines required for rendering are decoded online from the compressed images. Random access mechanisms at the "line level" are, therefore, needed to locate and decode individual compressed line image. These problems are even more pronounced in higher dimensional representations such as the light field and Lumigraph. Consider the 4-D light field of the Buddha statue [43], which consists of 32 × 32 array of images, each having a resolution of 256 × 256 with 24-bit per pixel. The total amount of storage is 192 MB. Decoding the entire light field into the main memory is, thus, prohibitive, especially when the resolution gets increasingly higher. Similar problems exist in the transmission of image-based representations. Techniques to support selective transmission/reception and a scalability data stream are, thus, of paramount importance. A simple comparison of different image-based representations and compression methods in terms of their complexities, compression ratios, and ease of random access is shown in Fig. 8.

B. Different Compression Approaches

In general, there are two approaches to reduce the data size of image-based representations. The first one is to reduce their dimensionality, often by limiting viewpoints or sacrificing some realism. Light fields and CMs are such examples. The second approach is more classical, namely, to exploit the high correlation (i.e., redundancy) within the representation using waveform coding or other model-based techniques. The scene geometry may be used explicitly or implicitly. The second approach can further be classified into three broad categories, which are: 1) pixel-based methods; 2) disparity compensation/prediction (DCP) methods; and 3) model-based/model-aided methods.

In pixel-based methods, the correlation between adjacent image pixels is exploited using traditional techniques such as VQ and transform coding. Very little geometry information, however, is used. In the DCP methods, scene geometry is utilized implicitly by exploiting the disparity of image pixels, resulting in better compression performance. (Disparity refers to the relative displacement of pixels in images taken in adjacent physical locations.) It is somewhat similar to motion of objects in video coding and they have been used in coding stereoscopic and multiview images [4], [42], [45], [59], [64], [65], [70], [82], [92]. Since the disparity is related to the object depth, as well as the viewing geometries, these methods also implicitly use the scene geometry to improve their coding performances. In contrast, model-based/model-aided approaches [50], [51] recover the geometry of the objects or scene in coding the observed images. The models and other information such as prediction residuals [50] or view-dependent texture maps [51] are then encoded. It is clear that an image-geometry tradeoff also exist in IBR compression.

Pixel-based methods are easy to implement and, in some cases, the random access problem is usually less complicated. However, their compression performance is limited compared with the other approaches. The model-based/model-aided methods have the potential to offer higher compression ratios and other functionalities such as model deformation. On the other hand, it requires the acquisition of 3-D models, and the encoding and decoding algorithms are more complicated. Since this paper discusses compression techniques of image-based representations, details on geometry compression [17] and model acquisition (which can be found elsewhere) are omitted.

We first review techniques for encoding IBR representations of static scenes.

C. Compression of Static IBR Representations

We start with compression techniques for CMs since its random access problem is the easiest to illustrate.

1) CMs: As described in Section II-C, CMs are constructed from images captured using a forward-displaced rotating camera. A novel view is reconstructed by retrieving appropriate vertical lines from these images. Compression techniques work well for this representation because the images are highly correlated. Most of these techniques are based on pixel-based or DCP. They also have a special mechanism to support random access at the line level.

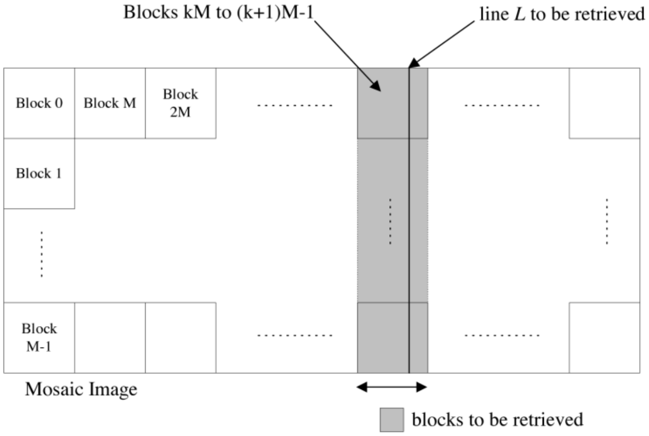

Fig. 9. Accessing a line L in a mosaic image.

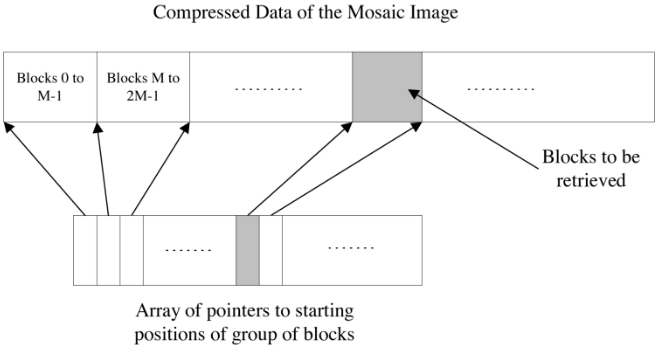
Compressed Data of the Mosaic Image
Blocks 0 to M-1
Blocks M to 2M-1
Blocks to be retrieved
Array of pointers to starting positions of group of blocks

Fig. 10. Accessing the required group of blocks using a set of pointers.

Pixel-based methods: In the original work on CMs [77], VQ with a fixed vector size is used to simplify the random access problem. The compression ratio reported was 12 : 1. (Levoy and Hanrahan [43] were the first to employ VQ to overcome the random access problem in light fields). The fixed size of the VQ index allows quick access to the required pixel data from the compressed light field or CMs for rendering. It also makes real-time decoding possible because VQ decoding involves only simple table look up. A compression ratio of 6 : 1 to 23 : 1 was reported in [43] for light fields at good reconstruction quality. However, the compression ratio of simple VQ is rather limited; it will also be unable to cope with future generations of image-based representations with extensive synthetic, as well as real-world scenes. The solution is to combine the pixel-based method with the DCP method.

DCP methods: In [78] and [79], Shum *et al.* proposed an MPEG-like algorithm to support random access of CMs at the line level. This is illustrated in Fig. 9, where a CM is encoded by a block-based technique such as the discrete cosine transform (DCT). Other coding schemes such as the wavelet transform can also be used with appropriate modifications.

The blocks (of size 16×16) are scanned vertically so that a set of vertical lines is completely contained in a group of consecutive blocks. In order to retrieve a vertical line L, the compressed data of macroblocks kM to $(k+1)M-1$ have to be decoded. A set of pointers to the starting locations of each vertical group of macroblocks is used to provide line-level random access, as shown in Fig. 10. These pointers can either be determined or stored in an array prior to rendering, or they can be embedded in the compressed bit streams. The last option avoids the creation of the pointer array each time when a new set of CMs is loaded into the memory, but this is accomplished at the expense of a slightly lower compression ratio. The 4-D light field faces the same problem of having to encode additional data bits to enable more efficient access to pixel data.

To further improve the compression efficiency, the DCP technique is applied to the sequence of mosaic images [77]. Mosaic images at regular intervals are chosen as I-pictures and coded independently, while images in-between are coded as B-pictures. P-pictures are not used due to their inter-dependencies, which complicates real-time rendering. The pointers structure is used to index the vertical group of blocks in the I- and B-pictures. For the lobby sequence, good quality reconstruction can be achieved at a compression ratio of 65 using six B-pictures between two consecutive I-pictures, and real-time rendering can be achieved on a Pentium II 300 desktop PC.

A similar MPEG-like algorithm, called the reference block coder (RBC), was proposed in [95]. The mosaic images are classified as anchor (A) and predicted (P) frames. A-frames are independently encoded in a similar manner as the I-pictures in MPEG-2, while the P-frames are encoded using DCP with reference to the surrounding A-frames. The P-frame in RBC differs from the P-pictures of MPEG-2 in that it refers only to the A-frames to facilitate random access. In addition, a two-level hierarchical table is embedded in RBC for indexed bit stream access. The compression ratio is slightly better than direct application of MPEG-2 after taking into account the regular panning nature of the image sequence. An interesting feature of RBC is the extensively used of data caches to reuse previously decoded macroblocks, which improves rendering speed. The rendering system is able to run smoothly on a Pentium II 300 desktop PC. The RBC was also the first algorithm that enabled the online streaming of CMs [97].

The application of wavelet transform to the compression of CMs was studied in [46], [89], and [90]. Potential advantages of wavelet transform are its higher coding performance and ability to provide resolution and quality scalabilities. Direct 3-D wavelet transform coding [40], however, yields a performance only comparable to that of MPEG-2. By using a smart rebinning approach to align successive images in a CM, the wavelet-based approach produces very encouraging results, which outperforms the MPEG-2 based algorithm by 3.7 dB on average. The success of the rebinning method is due to its ability to exploit the redundancy of multiple mosaic images arising from the disparity of image pixels.

The rendering operation is, however, complicated by the long filter support of the wavelet transform (compared with block transforms). In fact, decoding a given pixel involves decoding other adjacent pixels. To overcome this problem, the progressive inverse wavelet synthesis (PIWS) method [89] only performs the necessary inverse calculations to reconstruct the coefficient used in the current view. With extensive cache usage, PIWS was able to perform real-time rendering. A multiresolution subband coder using nonlinear filter bank [62] has also been proposed to overcome the long filter support of wavelet transform for progressive transmission.

2) Light Fields and Lumigraphs: The light field and Lumigraph sample the plenoptic function in a 2-D plane and generate a 2-D array of images of the scene. Since adjacent light-field images appear to be shifted relative to each other, there is considerably redundancy in the 4-D data set. In additional to conventional pixel- and disparity-based methods, a number of model-based/model-aided algorithms that explicitly explore the scene geometry were proposed.

Pixel-based methods: Earlier approaches on light-field or Lumigraph compression were mostly based on conventional pixel-based methods. The original work of Levoy and Hanrahan [43] used VQ to provide random access in light fields; DCT coding [58] and wavelet coding [38], [69] were subsequently used. More recently, DCP and model-based/model-aided methods were proposed to achieve a higher compression ratio for storage and transmission.

DCP methods: Disparity compensated prediction, as with CMs, can be applied to predict one light-field image from the others. This is illustrated in Fig. 11, where the array of light-field images is divided into I- and P-pictures. The P-pictures can be predicted by disparity compensation from the nearest encoded I-pictures, which are evenly distributed. An example is the V-coder described in [47] and [49], which is based on the H.263 video-coding algorithm. Like conventional video coder, the P-images are divided into 16×16 blocks. Eight different coding modes are incorporated to efficiently exploit the characteristic of the light field. Mode selection was determined using a rate-constrained approach and was solved using the method of Lagrange multipliers. Prior to rendering, the I-images are decoded and kept in local memory to provide instantaneous access to a low-resolution version of the light field. However, rendering speed may be adversely affected if the compressed light field is decoded online. This is because random access of light rays (pixel) is not available.

Recently, Tong and Gray [85] combined disparity compensation prediction (DCP) and VQ (HDCP) and proposed a hierarchical light-field coder. The 2-D array of light-field images is divided into layers, with the lowest layer being vector quantized without any prediction. Images in higher layers are predicted from images in the lower layers using DCP. The prediction residuals are again vector quantized and different coding modes are incorporated to improve coding efficiency. To facilitate random access, the residuals and disparities are not entropy encoded. Moreover, the predictive coded images are divided into regions, and each is associated with a 4-B offset to support random access. Significantly better compression rates were obtained for the “Buddha,” “Dragon,” and “Lion” light fields, compared with using simple tree-structure VQ (TSVQ).

The *D-coder*, which was also proposed in [48] and [49], relies on disparity compensation of light-field images. The four corner images in the image array are first encoded as I-images. Their disparity maps are then estimated and Huffman coded. From the encoded corner images and their disparity maps, the center image, and then the images midway between any two corner-images, are predicted. The residuals, if any, are DCT coded. These nine encoded images are then used to divide the image array into four quadrants, each of which is recursively encoded. Due to the hierarchical nature of the D-coder, the decoding of the image pixels is very time consuming. This slows down the rendering speed if the compressed data is decoded online.

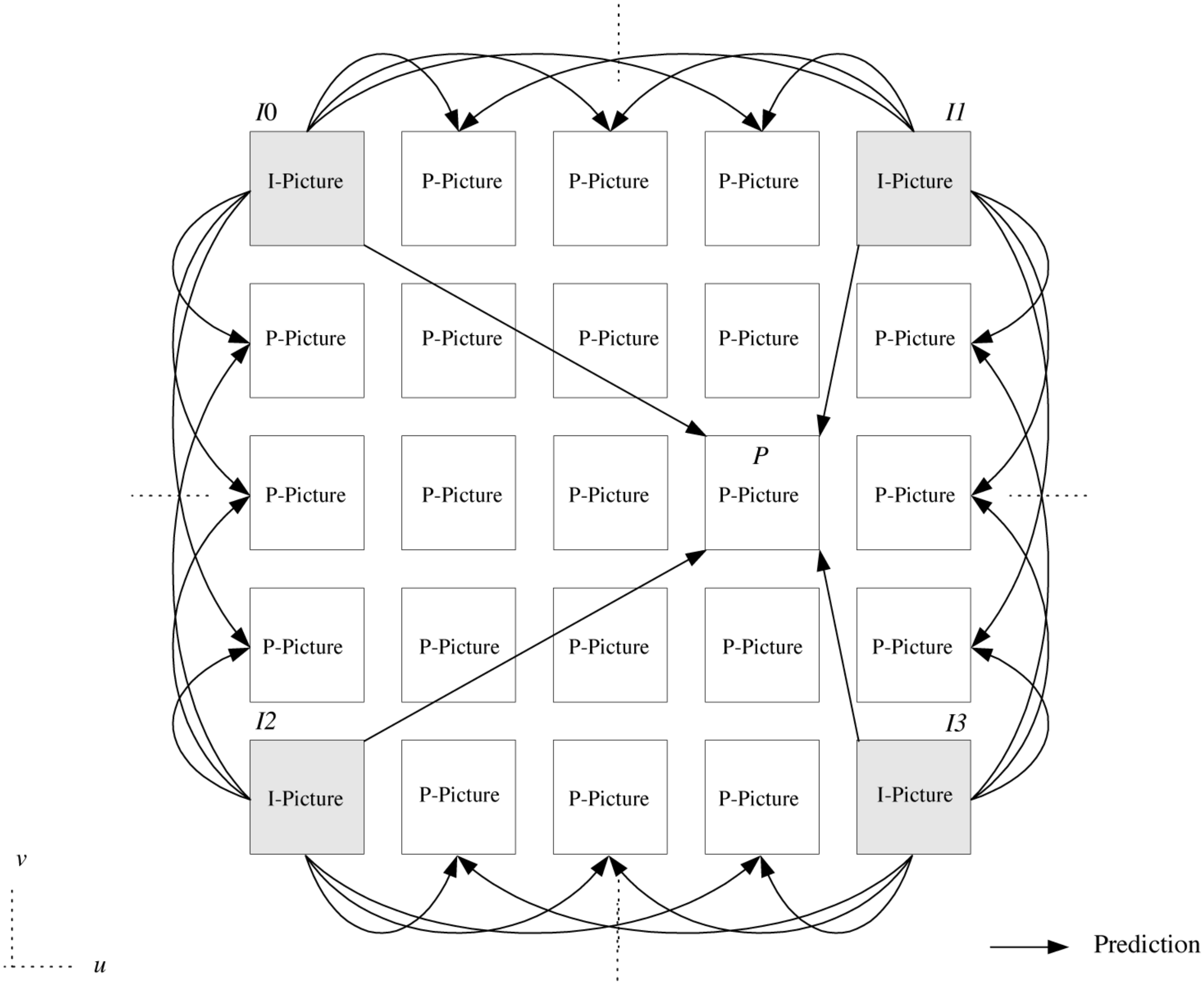

Fig. 11. DCP in light-field compression.

Zhang and Li [96] have also extended the reference block coding to the encoding of Lumigraph using multiple reference frame (MRF) prediction. Disparity compensation is applied to the 2-D light-field array instead of the one-dimensional (1-D) image sequence in CMs. As with I-images in [49] and [85], certain images in the light-field array are chosen as the anchor frames (A frames), which serve as references for predicting the remaining P images. A two-level index table is incorporated into the bit stream for quick access to individual picture and macroblocks. Like CMs, this reduces the compression ratio. At a compression ratio of 100 : 1, the overhead incurred is 10%. The overhead increases to 30% when the compression ratio reaches 160 : 1. A caching scheme is also incorporated to speedup the rendering.

Model-based/model-aided methods: It has been shown that 3-D scene geometry can improve coding efficiency and rendering quality considerably [14], [88]. The model-based coding (also known as texture-based coding) proposed in [51] makes use of the scene geometry to convert the images from a spherical light field to view-dependent texture maps. These maps exhibit greater inter-map correlation than the original images and are more effectively encoded using a modified set partitioning in hierarchical trees (SPIHTs) 4-D wavelet codec. On the other hand, model-aided predictive coding [50] makes use of geometry information to morph and predict new views from already encoded images. The prediction residuals are encoded using DCT-based coding. Like the hierarchical light-field coder in [85], a decimated version of the spherical light-field array are encoded as intra- or I-pictures, and they serve as references for predicting images at the next layer. By arranging the images in a hierarchical manner, a multiresolution representation of the image data is obtained which facilitates progressive rendering and decoding. Both algorithms encode the geometry of the objects using the embedded mesh coding (EMC) in which the vertex coordinates and mesh connectivity are jointly encoded to provide better scalability and improved performance. Experimental results showed that the model-aided approach is more robust to variations of the geometric models.

D. Compression of Dynamic IBR Representations

The image-based representations discussed thus far are associated with static scenes. There is a significant amount of work in stereoscopic video coding, which are mostly based on disparity compensation [65]. However, the compression and transmission

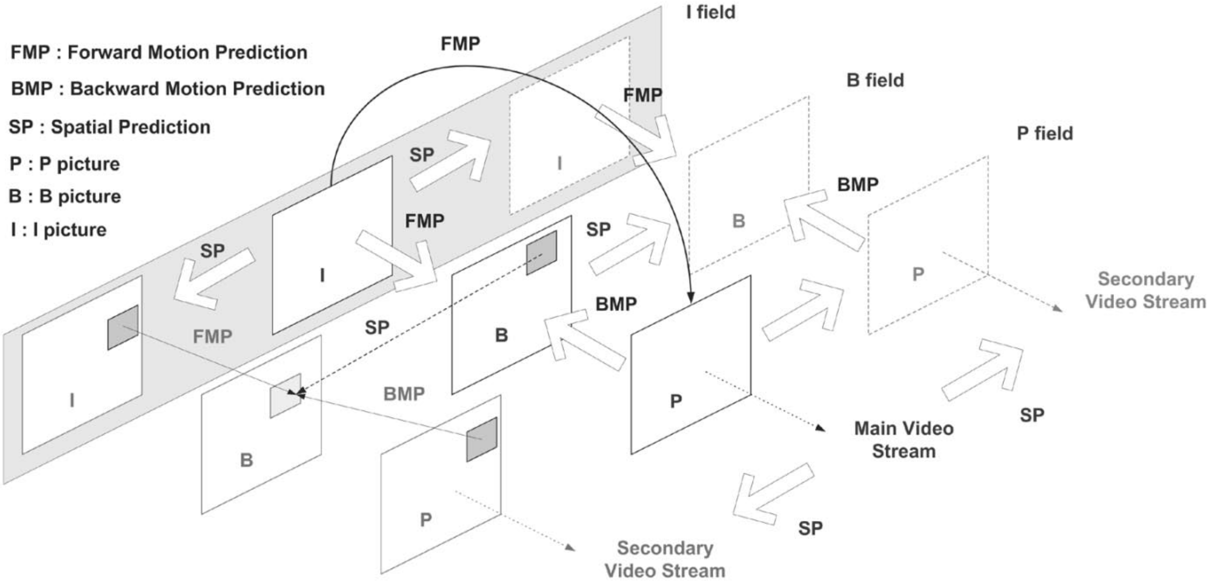

Fig. 12. Compression of 4-D SDLF.

of general dynamic IBR representations are not well studied. This is largely attributed to the logistical difficulties in capturing and transmitting of dynamic representations, which inherently involves huge amounts of data. Nevertheless, the ability of image-based techniques in creating photorealistic images of real scenes has stimulated a lot of interest in constructing sensor systems for capturing dynamic environments from multiple viewpoints. Examples are the Stanford University, Stanford, CA, Multicamera Project[2] and the Carnegie–Mellon University, Pittsburgh, PA, Virtualized Reality Project [32]. The goal of the Multicamera Project is to build an array of 128 video cameras using low-cost CMOS camera, inexpensive lens, and other processing and compression hardware. A prototype system with six cameras was reported [86]. The Virtualized Reality Project uses a set of synchronized cameras, and allows the viewer to virtually fly around and watch the event from new positions. This is made possible by reconstructing 3-D (octree) models at every frame offline.

More recently, Chan *et al.* [10] proposed a disparity and motion compensated compression algorithm for the simplified dynamic light fields (SDLFs), where videos were taken at regularly spaced locations along a line. This is illustrated in Fig. 12 for three videos sequences, called a group of field (GOF). A modified MPEG-2 video compression algorithm is used to provide random access to individual pictures. There are two types of video streams in the SDLF: main and secondary video streams. Main video streams are encoded using the MPEG-2 algorithm, which can be decoded without reference to other video streams. The light-field images captured at the same time instants as the I-pictures in a main stream constitute an I-field. Similarly, P- and B-fields are defined as the light-field images containing respectively the P- and B-pictures of the main video stream. Pictures from the secondary stream in the I-field are encoded using disparity or spatial prediction from the reference I-picture in the I-field. Pictures from the secondary stream in a P-field are predicted using spatial prediction from adjacent P-picture in the main stream, and the forward motion compensation from the reference I- or P- fields in the same secondary stream. Pictures from the secondary stream in B-field are predicted using spatial prediction and forward/backward motion compensation. To address the random access problem, pointers were embedded into the compressed data stream. Simulation results using an 16-camera synthetic SDLF showed an improvement of 2 dB in peak signal-to-noise ratio (PSNR) for the disparity/motion compensation scheme over direct application of MPEG2 algorithm to individual streams.

Another interesting 3-D dynamic IBR representation with a much lower data requirement is the panoramic video or time-varying environment map [13]. A panoramic video is a sequence of panoramas created at different time instants, which can be used to capture dynamic scenes at a stationary location or in general along a path with a 360° field-of-view. The resolution of a panoramic video may be large, which would pose a number of problems for transmission, digital storage, and rendering. For example, a 2048×768 panoramic video at 25 frames/s would require 112.5 MB/s of digital storage or transmission bandwidth.

In [63], each panoramic video frame is divided into tiles of smaller size to support selective decoding. As shown in Fig. 13, one frame of the panoramic video “Cafeteria” is divided into six smaller tiles of the same size. A panoramic video is thus partitioned into six separate subvideos, each of which can be compressed by the MPEG-2 algorithm. In virtual walkthrough applications, the appropriate portion in these tiles will be directly rendered to emulate virtual camera panning and zooming. If the whole panorama has a 360° field-of-view, then the maximum viewing angle of each tile will be 60°. Taking into account the possibility of overlapping, at most, two adjacent tiles have to be decoded simultaneously to support a user’s view of 60°. To handle the tile switching when the user changes the viewing angle, a random access mechanism, as shown in Fig. 14, was incorporated into the compressed data stream to facilitate fast

[2] [Online]. Available: http://graphics.stanford.edu/projects/array/

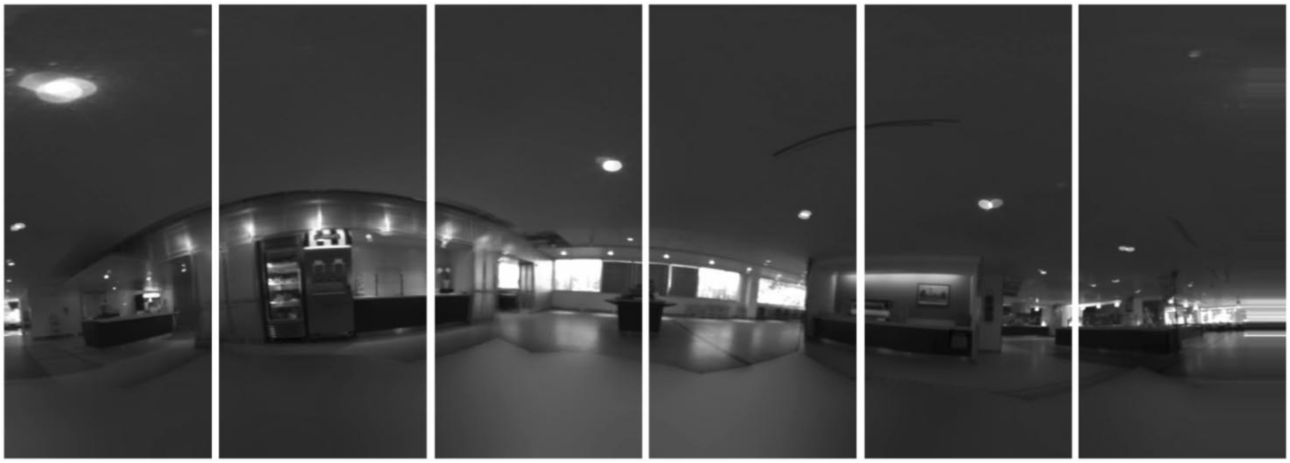

Fig. 13. Frame 8 of the panoramic video sequence "cafeteria."

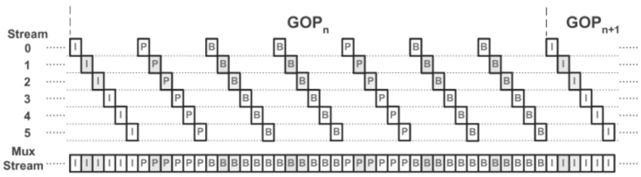

Fig. 14. Multiplexing of the tiles (streams) in the MPEG2 compressed panoramic video.

tile seeking. Here, each tile is encoded by the MPEG-2 standard with a GOP consisting of one I-picture, two P-pictures, and four B-pictures. If a tile switching is required during the decoding of the P- and B-pictures inside a GOP, it can only begin in the next GOP because the I-pictures of the new tiles in the current GOP might not be available. Therefore, the separation of the I-pictures should not be very large. In the current example, the maximum delay during tile switching is 0.28 s, assuming a frame rate of 25 frames/s. Interested readers are referred to [63] for other applications of panoramic video and issues of transmission over high-speed network.

E. Future Directions and Challenges

Table II summarizes the various IBR compression methods described earlier. Despite the significant progress achieved in IBR compression over the last few years, many research problems still remain. We envision that the data compression and transmission of the various image-based representations described in this paper and related representations (such as the compression of LDIs [18]) will continue to be important issues in IBR research. For example, the integration of model-based coding with traditional video coding approaches for light field compression [21], [50], [51] is an interesting area of research.

Methods for capturing, compression, and transmission of dynamic IBR functions have not been well explored yet. The panoramic video, as discussed earlier, is a 3-D dynamic image-based representation that is relatively simple to manipulate. As a result, this representation will be easier to use in a commercial setting. Dynamic generalizations of the light field and Lumigraph, which we called the plenoptic video, will likely involve scores of synchronized videos for them to be effective and compelling. It would be very challenging to efficiently compress and transmit them.

We predict that future virtual reality and gaming systems will rely heavily on image-based representations to render photo-realistic real-world scenes. Realistic-looking synthetic scenes that are expensive to render may be prerendered instead and stored as image-based representations in such systems as well. However, before such systems become a reality, the high level of interactivity associated with 3-D gaming will have to be enabled. This is a challenging and interesting topic that will need to be adequately addressed.

In addition, the amount of digital data associated with future IBR representations will become so large that selective decoding, reception, and streaming techniques for transmission will play a major role in their processing. This again calls for sophisticated random access methodology to retrieve these components with wide range of characteristics.

VII. Discussion

IBR is an area that straddles both computer vision and computer graphics. The continuum between images and geometry is evident from the IBR techniques reviewed in this paper. However, the emphasis of this paper is more on the aspect of rendering and not so much on image-based modeling. Other important topics such as lighting and animation are also not treated here.

Due to the large amount of data used to represent the 4-D function, light-field compression is necessary to make it practical. This is possible because of the high spatial coherency among all the input images. Some of the challenges in IBR compression such as rendering directly from compressed streams and producing more efficient scalable and embedded representations are briefly mentioned in Section VI-E.

In this review, IBR techniques are divided based on how much geometric information has been used, i.e., whether the method uses explicit geometry (e.g., LDI), implicit geometry or correspondence (e.g., view interpolation), or no geometry at all (e.g., light field). Other methods of dividing IBR techniques have also been proposed by others, such as on the nature of the pixel indexing scheme [33].

TABLE II
SUMMARY OF IBR COMPRESSION TECHNIQUES (NOTE: DCP = DISPARITY COMPENSATION PREDICTION, VQ = VECTOR QUANTIZATION, MRFP = MULTIPLE REFERENCE FRAME PREDICTION, MB = MACROBLOCK)

STATIC SCENES (Concentric Mosaics: Random access at line level)

References	Method	Random access	Compress. ratio	Remarks
[77]	VQ	Simple	Low	Simple, fast, rendering
[38], [69]	Haar wavelet decomp. and thresholding	Tree of wavelet coeffs. to speedup data access	Moderate	Scalable bit stream
[78]	Modified MPEG-2	Pointer to line of MB	High	Real-time rendering
RBC [44], [95]	Modified MPEG-2	Simple (pointer)	High	Real-time cache to enhance speed
[46], [89], [90]	Wavelet	Complicated	High	Less rendering speed. Real-time with cache enhancement [89]. Good compression efficiency using smart rebinning [90].

STATIC SCENES (Light Field: Random access at pixel level)

References	Method	Random access	Compress. ratio	Remarks
[43]	VQ	Simple	Low	Simple, fast rendering
HDCP [85]	DCP, VQ	Pointer to regions	High	Simple, fast rendering
V-coder [47], [49]	DCP, DCT-based coding	Complicated	High	
D-coder [48], [49]	HDCP using disparity map, DCT-based coding	Complicated	Slightly inferior to V-coder	Possible to interpolate intermediate missing picture
RBC [44], [96]	Modified MPEG-2, MRFP	2-level index table	High	Real-time rendering. Also for Lumigraph compression. Significant overheads of indexing at high compression ratio.
Model-aided coder (MAC) [50]	Approx. 3D geometry model and DCP	Complicated	Better than MBC	
Model-based coder (MBC) [51]	Use scene geometry to convert images to texture maps, which are coded using a modified SPIHT algorithm [73]	Random access to arbitrary texture segments	High	Supports progressive decoding. Graphics hardware can be used to accelerate rendering.

DYNAMIC SCENES (Panoramic Video: Random access at tile level)

Reference	Method	Random access	Compress. ratio	Remarks
[63]	Modified MPEG-2	B-pictures in MPEG-2, pointers to tiles	High	Panoramas divided into tiles for selective decoding and reception

DYNAMIC SCENES (Simplified Dynamic Light Field: Random access at line level)

Reference	Method	Random access	Compress. ratio	Remarks
[10]	Modified MPEG-2, DCP	Pointers to line of MB	High	

A. Challenges

There remain many challenges in IBR, including the following.

a) Efficient representation: What is very interesting is the tradeoff between geometry and images needed to use for antialiased IBR. The design choices for many IBR systems were made based on the availability of accurate geometry. Plenoptic sampling provides a theoretical foundation for designing IBR systems.

Both light-field rendering and Lumigraph avoid the feature correspondence problem by collecting many images with known camera poses. Due to the size of the database (even after compression), virtual walkthroughs of a real scene using light fields have not yet been fully demonstrated.

b) Rendering performance: How would one implement the "perfect" rendering engine? One possibility would be to adapt current hardware accelerators to produce, say, an approximate version of an LDI or a lumigraph by replacing it with view-dependent texture-mapped sprites. The alternative is to design new hardware accelerators that can handle both conventional rendering and IBR. An example in this direction is the use of PixelFlow to render image-based models [55]. Pixelflow [19] is a high-speed image-generation architecture that is based on the techniques of object-parallelism and image compositions.

c) Capturing: Panoramas are relatively not difficult to construct. Many previous systems have been built to construct cylindrical and spherical panoramas by stitching multiple images together (e.g., [13], [52], [57], [83], and [84]). When the camera motion is very small, it is possible to put together only small strips from registered images, i.e., slit images (e.g., [68] and [98]) to form a large panoramic mosaic. Capturing panoramas is even easier if omnidirectional cameras (e.g., [61] and [60]) or fisheye lens [91] are used.

It is, however, very difficult to construct a continuous 5-D complete plenoptic function [35], [57] because it requires solving the difficult feature correspondence problem. To date, no one has yet shown a collection of 7-D complete plenoptic functions (authoring a dynamic environment with time-varying lighting conditions is a very interesting problem).

B. Two Scenarios

IBR can have many interesting applications. Two scenarios, in particular, are worth pursuing:

a) Large environments: Many successful techniques, e.g., light field, CMS, have restrictions on how much a user can change his viewpoint. QuickTime VR [13] is still popular for showcasing large environments despite the visual discomfort caused by jumping between panoramas. While this can be alleviated by having multiple panoramic clusters and enabling single degree of freedom (DOF) transitioning between these clusters [34], the range of virtual motion is nevertheless still restricted. To move around in a large environment, one has to combine image-based techniques with geometry-based models in order to avoid excessive amount of data required.

b) Dynamic environments: Until now, most of IBR systems have been focused on static environments. With the development of panoramic video systems, it is conceivable that IBR can be applied to dynamic environments as well. Two issues must be studied: sampling (how many images should be captured), and compression (how to reduce data effectively).

VIII. Concluding Remarks

We have surveyed recent developments in the area of IBR and, in particular, categorized them based on the extent of use of geometric information in rendering. Geometry is used as a means of compressing representations for rendering, with the limit being a single 3-D model with a single static texture. While the purely image-based representations have the advantage of photorealistic rendering, they come with the high costs of data acquisition and storage requirements. We have also surveyed development in compression techniques for image-based representations, with examples of CMs and light fields.

Demands on realistic rendering, compactness of representation, speed of rendering, and costs and limitations of computer vision reconstruction techniques force the practical representation to fall somewhere between the two extremes. It is clear from our survey that IBR and the traditional 3-D model-based rendering techniques have complimentary characteristics that can be capitalized. As a result, we believe that it is important that future graphics rendering hardware and video technology be customized to handle both the traditional 3-D model-based rendering as well as IBR.

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] D. G. Aliaga and I. Carlbom, "Plenoptic stitching: A scalable method for reconstructing 3D interactive walkthroughs," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 2001, pp. 443–450.

[3] S. Avidan and A. Shashua, "Novel view synthesis in tensor space," in *Computer Vision and Pattern Recognition Conf.*, San Juan, PR, June 1997, pp. 1034–1040.

[4] H. Aydinoglu and M. H. Hayes, "Stereo image coding: A projection approach," in *IEEE Int. Image Processing Conf.*, vol. 8, 1998, pp. 506–516.

[5] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 2001, pp. 425–432.

[6] E. Camahort, "4D light-field modeling and rendering," Univ. Texas at Austin, Austin, TX, Tech. Rep. TR01-52, 2001.

[7] E. Camahort, A. Lerios, and D. Fussell, "Uniformly sampled light fields," in *9th Eurographics Rendering Workshop*, Vienna, Austria, June–July 1998, pp. 117–130.

[8] D. Capel and A. Zisserman, "Super-resolution from multiple views using learnt image models," in *Computer Vision and Pattern Recognition Conf.*, vol. 2, Kauai, HI, Dec. 2001, pp. 627–634.

[9] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic sampling," in *Proc. ACM Annu. Computer Graphics Conf.*, July 2000, pp. 307–318.

[10] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan, and H.-Y. Shum, "The data compression of simplified dynamic light fields," presented at the IEEE Int. Acoustics, Speech, and Signal Processing Conf., Hong Kong, Apr. 2003.

[11] C. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 1999, pp. 291–298.

[12] S. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 1993, pp. 279–288.

[13] S. E. Chen, "QuickTime VR-an image-based approach to virtual environment navigation," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 1995, pp. 29–38.

[14] W.-C. Chen, J. V. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," *ACM Trans. Graph.*, pp. 447–456, July 2002.

[15] P. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," in *Proc. 9th Eurographics Rendering Workshop*, 1998, pp. 105–116.

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

[17] M. Deering, "Geometry compression," in *Proc. SIGGRAPH'99*, Aug. 1995, pp. 13–20.

[18] J. Duan and J. Li, "Compression of the layered depth image," in *IEEE Data Compression Conf.*, Mar. 2001, pp. 331–340.

[19] J. Eyles, S. Molnar, J. Poulton, T. Greer, A. Lastra, N. England, and L. Westover, "Pixelflow: The realization," presented at the SIGGRAPH Eurographics Graphics Hardware Workshop, Los Angeles, CA, Aug. 1997.

[20] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric ViewPoint*. Cambridge, MA: MIT Press, 1993.

[21] B. Girod, P. Eisert, M. Magnor, U. Steinback, and T. Wiegand, "3-D image models and compression synthetic hybrid or natural fit?," in *IEEE Int. Image Processing Conf.*, Kobe, Japan, Oct. 1999, pp. 525–529.

[22] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. ACM Annu. Computer Graphics Conf.*, New Orleans, LA, Aug. 1996, pp. 43–54.

[23] B. Heigl, R. Koch, M. Pollefeys, S. Denzler, and L. Van Goal, "Plenoptic modeling and rendering from image sequences taken by hand-held camera," in *Die Deutsche Arbeitsgemeinschaft für Mustererkennung*, 1999, pp. 94–101.

[24] I. Ihm, S. Park, and R. Lee, "Rendering of spherical light fields," in *Pacific Graphics*, Seoul, Korea, Oct. 1997, pp. 59–68.

[25] M. lrani and S. Peleg, "Improving resolution by image registration," *Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.

[26] A. Isaksen, L. McMillan, and S. Gortler, "Dynamically reparameterized light fields," in *Proc. ACM Annu. Computer Graphics Conf.*, July 2000, pp. 297–306.

[27] *Digital Compression and Coding of Continuous-Tone Still Images*, ISO/IEC Int. Standard DIS 10 918.

[28] *Overview of the MPEG-4 Standard*, ISO/IEC JTC I/SC 29/WG II, 1999.

[29] *JPEG 2000 Image Coding System: Core Coding System [WG 1 N 1646]*, ISO/IEC JTC 1/SC 29/WG1, ISO/IEC FCD 15 444-1, 2000.

[30] *Generic Coding of Moving Pictures and Associated Audto Information: Video*, ITU-T Rec. H.262/ISO/IEC 1381 8-2, 1994.

[31] *Video Coding for Low Bit Rate Communication*, ITU-T Rec. H.263, 1998.

[32] T. Kanade, P. Rander, and P. J. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE Multimedia*, vol. 4, pp. 34–47, Jan.–Mar. 1997.

[33] S. B. Kang, "A survey of image-based rendering techniques," *VideoMetrics*, vol. SPIE 3641, pp. 2–16, 1999.

[34] S. B. Kang and P. K. Desikan, "Virtual navigation of complex scenes using clusters of cylindrical panoramic images," in *Graphics Interface*, Vancouver, BC, Canada, June 1998, pp. 223–232.

[35] S. B. Kang and R. Szeliski, "3-D scene data recovery using omni- directional multibaseline stereo," in *IEEE Computer Vision and Pattern Recognition Soc. Conf.*, San Francisco, CA, June 1996, pp. 364–370.

[36] S. B. Kang, R. Szeliski, and P. Anandan, "The geometry-image representation tradeoff for rendering," in *Int. Image Processing Conf.*, vol. 2, Vancouver, BC, Canada, Sept. 2000, pp. 13–16.

[37] A. Katayama, K. Tanaka, T. Oshino, H. Tamura, and S. Fisher, "A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images," in *Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems II*, S. Merritt and B Balsa, Eds., 1995, vol. 2409, pp. 11–20.

[38] P. Lalonde and A. Fournier, "Interactive rendering of wavelet projected light fields," in *Graphics Interface*, Kingston, ON, Canada, June 1999, pp. 107–114.

[39] S. Laveau and O. Faugeras, "3-D scene representation as a collection of images and fundamental matrices," INRIA-Sophia Antipolis, Sophia Antipolis, Italy, Tech. Rep. 2205, 1994.

[40] S. Laveau and O. D. Faugeras, "3-D scene representation as a collection of images," in *12th Pattern Recognition Int. Conf.*, vol. A, Jerusalem, Israel, Oct. 1994, pp. 689–691.

[41] J. Lengyel, "The convergence of graphics and vision," IEEE Computer Soc., Los Alamitos, CA, Tech. Rep., July 1998.

[42] H. W. Leung and T. Chen, "Compression with mosaic prediction for image-based rendering applications," presented at the IEEE Int. Multimedia Expo. Conf., New York, NY, July 2000.

[43] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. ACM Annu. Computer Graphics Conf.*, New Orleans, LA, Aug. 1996, pp. 31–42.

[44] J. Li, H.-Y. Shum, and Y. Q. Zhang, "On the compression of image based rendering scene," *IEEE Int. Image Processing Conf.*, vol. 2, pp. 21–24, Sept. 2000.

[45] M. E. Lukacs, "Predictive coding of multi-viewpoint image sets," in *IEEE Int. Acoustics, Speech, and Signal Processing Conf.*, 1986, pp. 521–524.

[46] L. Luo, Y. Wu, J. Li, and Y. Q. Zhang, "Compression of concentric mosaic scenery with alignment and 3D wavelet transform," presented at the SPIE Image and Video Communications and Processing Conf., San Jose, CA, Jan. 2000.

[47] M. Magnor and B. Girod, "Adaptive block-based light field coding," in *3rd Int. Synthetic and Natural Hybrid Coding and Three-Dimensional Imaging Workshop*, Santorini, Greece, Sept. 1999, pp. 140–143.

[48] ——, "Hierarchical coding of light fields with disparity maps," presented at the IEEE Int. Image Processing Conf., Kobe, Japan, Oct. 1999.

[49] ——, "Data compression for light-field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 338–343, Apr. 2000.

[50] ——, "Model-aided coding of multi-viewpoint image data," in *IEEE Int. Image Processing Conf.*, vol. 2, Vancouver, BC, Canada, Sept. 2000, pp. 919–922.

[51] ——, "Model-based coding of multi-viewpoint imagery," in *Proc. SPIE Visual Communication and Image Processing Conf.*, vol. 4067, Perth, Australia, June 2000, pp. 14–22.

[52] S. Mann and R. W. Picard, "Virtual bellows: Constructing high-quality images from video," in *1st IEEE Int. Image Processing Conf.*, vol. I, Austin, TX, Nov. 1994, pp. 363–367.

[53] W. Mark, L. McMiIlan, and G. Bishop, "Post-rendering 3D warping," in *Proc. I3D Graphics Symp.*, 1997, pp. 7–16.

[54] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, "Image-based visual hulls," in *Proc. ACM Annu. Computer Graphics Conf.*, July 2000, pp. 369–374.

[55] D. K. McAllister, L. Nyland, V. Popescu, A. Lastra, and C. McCue, "Real-time rendering of real world environments," presented at the Eurographics Rendering Workshop, Granada, Spain, June 1999.

[56] L. McMillan, "An image-based approach to three-dimensional computer graphics," Ph.D. dissertation, Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, 1999.

[57] L. McMiIlan and U. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. ACM Annu. Computer Graphics Conf.*, Aug. 1995, pp. 39–46.

[58] G. Miller, S. Rabin, and D. Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," in *Eurographics Rendering Workshop*, Oct. 1998, pp. 281–292.

[59] T. Naemua, M. Kaneko, and H. Harashima, "Compression and representation of 3-D images," *IEICE Trans. Inform. Syst.*, vol. E82-D, no. 3, pp. 558–567, 1999.

[60] V. S. Nalwa, "A true omnidirecional viewer," Bell Labs., Holmdel, NJ, Tech. Rep., 1996.

[61] S. Nayar, "Catadioptric omnidirectional camera," in *IEEE Computer Soc. Computer Vision and Pattern Recognition Conf.*, San Juan, PR, June 1997, pp. 482–488.

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

[63] K. T. Ng, S. C. Chan, H.-Y. Shum, and S. B. Kang, "On the data compression and transmission aspects of panoramic video," in *IEEE Int.Image Processing Conf.*, vol. 2, Thessaloniki, Greece, Oct. 2001, pp. 105–108.

[64] 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 Processing Mag.*, vol. 16, pp. 47–54, May 1999.

[65] ——, "Stereo/multiview encoding using the MPEG family of standards," presented at the Electron. Imaging, San Diego, CA, Jan. 1999.

[66] M. Oliveira and G. Bishop, "Relief textures," Univ. North Carolina, Chapel Hill, NC, Comput. Sci. Tech. Rep. TR99-015, Mar. 1999.

[67] S. Peleg and M. Ben-Ezra, "Stereo panorama with a single camera," presented at the Computer Vision and Pattern Recognition Conf., 1999.

[68] S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," in *IEEE Computer Soc. Computer Vision and Pattern Recognition Conf.*, San Juan, PR, June 1997, pp. 338–343.

[69] I. Peter and W. Strasser, "The wavelet stream: Interactive multi resolution light field rendering," in *Eurographics Rendering Workshop*, June 2001, pp. 262–273.

[70] 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," *Signal Processing*, vol. 10, pp. 201–234, 1997.

[71] P. Rademacher, "View-dependent geometry," in *SIGGRAPH'99*, Aug. 1999, pp. 439–446.

[72] P. Rademacher and G. Bishop, "Multiple-center-of-projection images," in *Proc. ACM Annu. Computer Graphics Conf.*, Orlando, FL, July 1998, pp. 199–206.

[73] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[74] D. Scharstein, "Stereo vision for view synthesis," in *IEEE Computer Soc. Computer Vision and Pattern Recognition Conf.*, San Francisco, CA, June 1996, pp. 852–857.

[75] S. M. Seitz and C. M. Dyer, "View morphing," in *Proc. ACM Annu. Computer Graphics Conf.*, New Orleans, LA, Aug. 1996, pp. 21–30.

[76] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, "Layered depth images," in *Proc. ACM Annu. Computer Graphics Conf.*, Orlando, FL, July 1998, pp. 231–242.

[77] H.-Y. Shum and L.-W. He, "Rendering with concentric mosaics," in *ACM SIGGRAPH'99*, Los Angeles, CA, Aug. 1999, pp. 299–306.

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

[79] ——, "A virtual reality system using the concentric mosaic: Construction, rendering, and data compression," *IEEE Trans. Multimedia*, 2003, to be published.

[80] H.-Y. Shum and R. Szeliski, "Construction and refinement of panoramic mosaics with global and local alignment," in *6th Int. Computer Vision Conf.*, Bombay, India, Jan. 1998, pp. 953–958.

[81] P. P. Sloan, M. F. Cohen, and S. J. Gortler, "Time critical lumigraph rendering," in *Interactive 3D Graphics Symp.*, Providence, RI, 1997, pp. 17–23.

[82] M. G. Strintzis and S. Malasiotis, "Object-based coding of stereoscopic and 3D image sequences: A review," *IEEE Signal Processing Mag.*, vol. 16, pp. 14–28, May 1999.

[83] R. Szeliski, "Video mosaics for virtual environments," *IEEE Comput. Graph. Appl.*, vol. 16, pp. 22–30, Mar. 1996.

[84] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and texture-mapped models," in *Proc. ACM Computer Graphics Conf.*, Aug. 1997, pp. 251–258.

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

[86] B. Wilburn, M. Smulski, H. H. K. Lee, and M. Horowitz, "The light field video camera," *SPIE Electronic Imaging: Media Processors*, vol. 4674, pp. 1–8, 2002.

[87] T. Wong, P. Heng, S. Or, and W. Ng, "Image-based rendering with controllable illumination," in *Proc. 8th Eurographics Rendering Workshop*, St. Etienne, France, June 1997, pp. 13–22.

[88] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3D photography," in *Proc. SIGGRAPH'00*, July 2000, pp. 287–296.

[89] Y. Wu, L. Luo, J. Li, and Y. Q. Zhang, "Rendering of 3D wavelet compressed concentric mosaic scenery with progressive inverse wavelet synthesis (PIWS)," in *Proc. SPIE Visual Commun. Image Processing*, vol. 4067, Perth, Australia, June 2000.

[90] Y. Wu, C. Zhang, J. Li, and J. Xu, "Smart rebinning for compression of the concentric mosaics," in *ACM Int. Multimedia Conf.*, Los Angeles, CA, Oct. 2000, pp. 201–209.

[91] Y. Xiong and K. Turkowski, "Creating image-based VR using a self-calibrating flsheye lens," in *IEEE Computer Soc. Computer Vision and Pattern Recognition Conf.*, San Juan, PR, June 1997, pp. 237–243.

[92] H. Yamaguchi, Y. Tatehira, K. Akiyama, and Y. Kobayashi, "Stereoscopic images disparity for predictive coding," in *IEEE Int. Acoustics, Speech, Signal Processing Conf.*, 1989, pp. 1976–1979.

[93] J. Yu, L. McMillan, and S. Gortler, "Scam light field rendering," in *Proc. Pacific Graphics'02*, 2002, pp. 137–144.

[94] C. Zhang and T. Chen, "Generalized plenoptic sampling," Carnegie–Mellon Univ., Pittsburgh, PA, Tech. Rep. AMPOI-06, Sept. 2001.

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

[96] ——, "Compression of lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering," in *IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 254–263.

[97] ——, "Interactive browsing of 3D environment over the internet," in *Proc. SPIE Visual Communication and Image Processing Conf.*, vol. 4310, San Jose, CA, Jan. 2001.

[98] J. Y. Zheng and S. Tsuji, "Panoramic representation of scenes for route understanding," in *Proc. 10th Int. Pattern Recognition Conf.*, June 1990, pp. 161–167.

Heung-Yeung Shum (S'90–M'90–SM'01) received the Ph.D. degree in robotics from the School of Computer Science, Carnegie–Mellon University, Pittsburgh, PA, in 1996.

For three years he was a Researcher with the Vision Technology Group, Microsoft Research, Redmond, WA. In 1999, he moved to Microsoft Research Asia, Beijing, China, where he is currently a Senior Researcher and the Assistant Managing Director. He has authored or coauthored 12 SIGGRAPH papers in the areas of IBR, texture, and appearance modeling, and motion synthesis. He has also authored or coauthored over 40 journal papers and approximately 100 conference papers. His research interests include computer vision, computer graphics, human–computer interaction, pattern recognition, statistical learning, and robotics.

Dr. Shum has been on several vision and graphics committees, including the SIGGRAPH Papers Committee, the Computer Vision and Pattern Recognition (CVPR) Area Chair Committee and the International Conference on Computer Vision (ICCV) Area Chair Committee. He is the general co-chair of the 10th ICCV, Beijing, China, 2005.

Sing Bing Kang (S'85–M'89–SM'01) received the B.Eng. and M.Eng. degrees in electrical engineering from the National University of Singapore (NUS), Singapore, in 1987 and 1990, respectively, and the M.Sc. and Ph.D. degrees in robotics from Carnegie–Mellon University (CMU), Pittsburgh, PA, in 1992 and 1994, respectively. His doctoral research concerned enabling robot systems to observe, recognize, and replicate grasping tasks.

From 1995 to 1999, he was a Member of Research Staff with the Cambridge Research Laboratory, Compaq, where he was involved with 3-D modeling from image sequences and IBR. He is currently a Researcher with Microsoft Research, Redmond, WA, where he is involved with environment modeling from images.

Dr. Kang was the recipient of the 1991 IEEE Computer Society Outstanding Paper Award for his paper on the "Complex Extended Gaussian Image" presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). He was also the recipient of the 1997 King-Sun Fu Memorial Best Transaction Paper Award for his IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION paper "Toward Automatic Robot Instruction From Perception—Mapping Human Grasps to Manipulator Grasps."

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

In 1990, he joined the City Polytechnic of Hong Kong, as an Assistant Lecturer and then became a University Lecturer. Since 1994, he has been with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, where he is currently an Associate Professor. In 1998 and 1999, he was a Visiting Researcher with the Microsoft Research, Redmond, WA, and Microsoft China, respectively. His research interests include fast transform algorithms, filter design and realization, multirate signal processing, communications signal processing, and IBR.

Dr. Chan is 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.