S.C. Chan, Heung-Yeung Shum, and King-To Ng

Image-Based Rendering and Synthesis

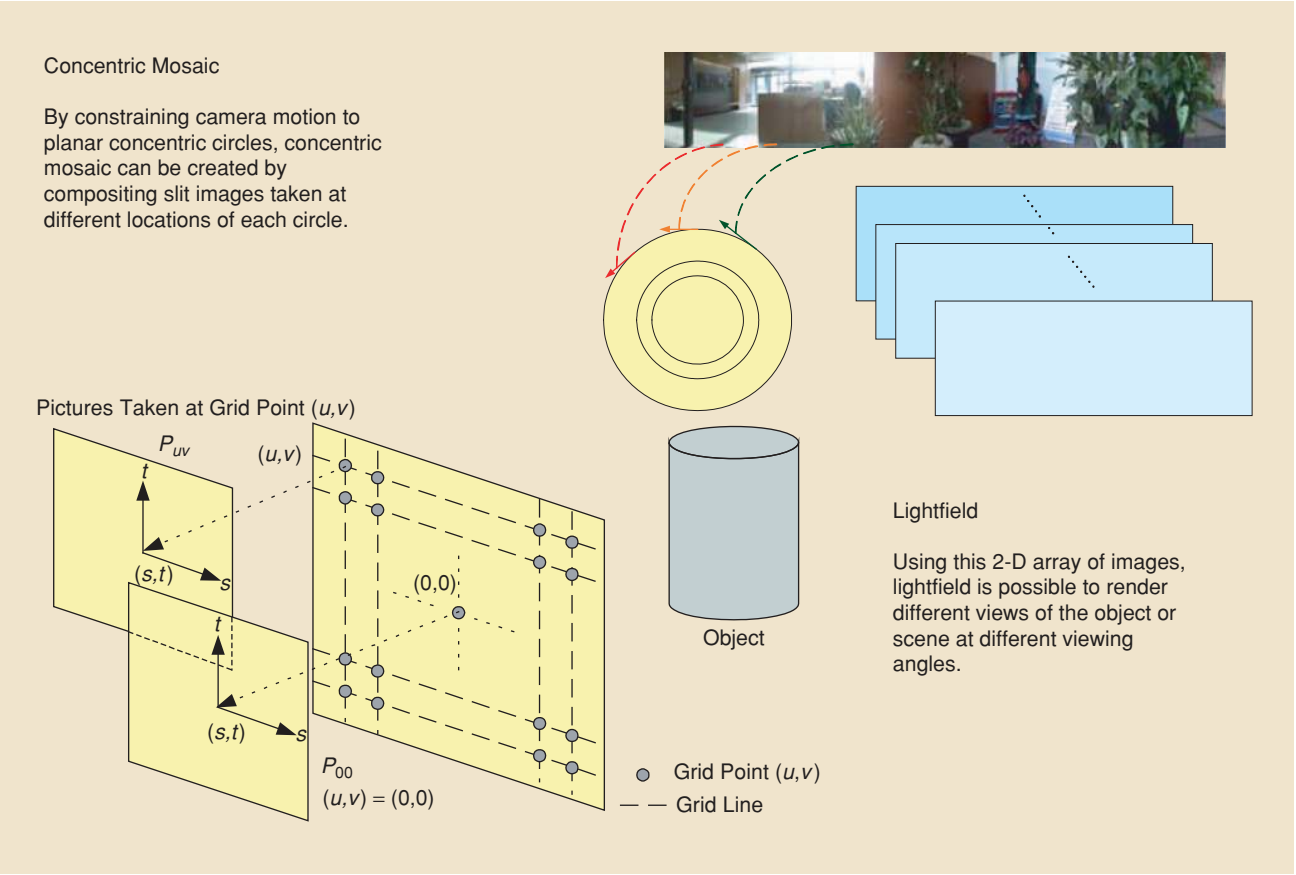
Technological advances and challenges

Multiview imaging (MVI) has attracted considerable attention recently due to its increasingly wide range of applications and the decreasing cost of digital cameras. This opens up many new and interesting research topics and applications, such as virtual view synthesis for three-dimensional (3-D) television (3DTV) and entertainment, high-performance imaging, video processing and analysis for surveillance, distance learning, industry inspection, etc.

One of the most important applications in MVI is the development of advanced immersive viewing or visualization systems using, for instance, 3DTV. With the introduction of multiview TVs, it is expected that a new age of 3DTV systems will arrive in the near future. To realize these goals, however, there are still many new and challenging issues to be addressed. In particular, multiview systems normally require a large amount of storage and are rather difficult to construct. Of more importance still, the various cameras in the camera array usually have very different characteristics and positions, which makes the synthesis of virtual view (multiview synthesis) difficult. While data compression issues are generally known in the signal processing

Digital Object Identifier 10.1109/MSP.2007.905702

1053-5888/07/$25.00©2007IEEE



[FIG1] **Concentric mosaic and lightfield.**

community, other essential techniques such as camera calibration, object segmentation, rendering, and relighting are subjects of intense research in the graphics and vision community. Therefore, experts from signal processing, computer graphics, and vision must be called upon to adequately address these multidisciplinary research issues.

Image-based rendering (IBR) refers to a collection of techniques and representations that allow 3-D scenes and objects to be visualized in a realistic way without full 3-D model reconstruction. IBR uses images as the primary substrate. The potential for photorealistic visualization has tremendous appeal, and it has been receiving increasing attention over the years. Applications such as video games, virtual travel, and E-commerce stand to benefit from this technology. This article serves as a tutorial introduction and brief review of this important technology. We first start with the classification, principles, and key research issues of IBR. We then describe an object-based IBR system to illustrate the techniques involved and its potential application in view synthesis and processing. Stereo matching, which is an important technique for depth estimation and view synthesis, is briefly explained and some of the top-ranked methods are highlighted. Finally, the challenging problem of interactive IBR is explained. Possible solutions and some state-of-the-art systems are also reviewed.

IBR

CLASSIFICATIONS

In IBR [1]–[12], [62], [63], new views of scenes are reconstructed from a collection of densely sampled images or videos. Examples include the well-known panoramas [6], lightfields [8], and variants [9]–[12], concentric mosaics [7], etc. (see Figure 1 for a brief summary of two of these representations). The reconstruction problem (i.e., rendering) is treated as a multidimensional sampling problem, where new views are generated from densely sampled images and depth maps instead of building accurate 3-D models of the scenes. Depending on the functionality required, there is a spectrum of IBR, as shown in Figure 2. The technologies differ from each other in the amount of geometry information of the scenes/objects being used.

At one end of the spectrum, like traditional texture mapping, we have very accurate geometric models of the scenes and objects, for instance, generated by animation techniques, but only a few images are required to generate the textures. Given the 3-D models and the lighting conditions, novel views can be rendered using conventional graphic techniques. Moreover, interactive rendering with movable objects and light sources can be supported using advanced graphics hardware.

At the other extreme, lightfield [8] or lumigraph [9] rendering relies on dense sampling (by capturing more image/videos)

with no or very little geometry information for rendering without recovering the exact 3-D models. An important advantage of the latter is its superior image quality, compared with 3-D model building for complicated real-world scenes. Another important advantage is that it requires considerably less computational resources for rendering regardless of the scene complexity, because most of the quantities involved are precomputed or recorded. This has attracted considerable attention in the computer graphics community recently in developing fast and efficient rendering algorithms for real-time relighting and soft-shadow generation [19]–[22].

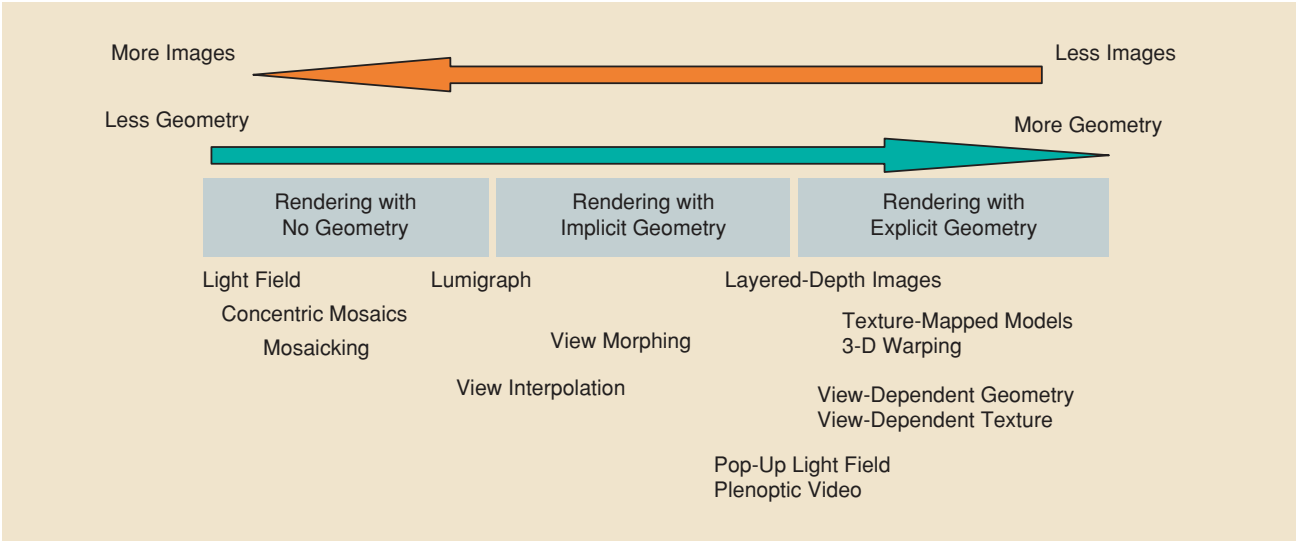
Broadly speaking, image-based representations can be classified according to the geometry information used into three main categories: 1) representations with no geometry, 2) representations with implicit geometry, and 3) representations with explicit geometry. Two-dimensional (2-D) panoramas [6], 3-D concentric mosaics [7], five-dimensional (5-D) McMillan and Bishop's plenoptic modeling [5], four-dimensional (4-D) ray-space representation [62], [63], and lightfields [8]/lumigraph [9] belong to the first category, while layered-based or object-based representations using depth maps [10], [11], [23] fall into the third. Conventional 3-D computer graphics models and other more sophisticated representations [25]–[27] belong to the last category. Another classification is based on the concept of plenoptic function, which is related to the dimensionality of the image-based representations. A recent survey of these representations can be found in [4].

Since capturing a 3-D model in real time is still a very difficult problem, lightfield- or lumigraph-based dynamic IBR representations with little amount of geometry information have received considerable attention in immersive TV (also called 3-D or multiview TV) applications. In particular, excellent rendering quality has been demonstrated using the pop-up lightfield [10], the object-based approach [23], and the layered-based rendering approach [11]. Later sections will be devoted to these representations, which use approximate/incomplete geometry in the form of depth maps. Segmentation, matting, and depth estimation techniques that are crucial to these approaches will also be illustrated.

On the other hand, since 3-D models of the objects and scenes are unavailable, user interaction is limited to the change of viewpoints and sometimes limited amount of relighting. In contrast, more user interaction such as real-time relighting and soft-shadow computation has been found to be feasible using IBR concepts and the associated 3-D models using precomputed radiance transfer (PRT) [21] and precomputed shadow fields [22]. This opens up a new opportunity for very fast interactive visualization/graphic systems with low complexity. If approximate geometry of objects in a scene can be recovered, then interactive editing and relighting of real scenes are in principle feasible. This has important applications in computer games, scientific visualization, and relighting of IBR objects in future generations of IBR systems. 3-D reconstruction of video objects such as human body and fast rendering algorithms are two key problems in supporting these innovative multimedia applications. A brief review of these problems will be given later in the "Interactive IBR" section, where a few state-of-the-art systems will be used as illustration. In the following, we shall briefly outline the principle and major research issues in this exciting area of research.

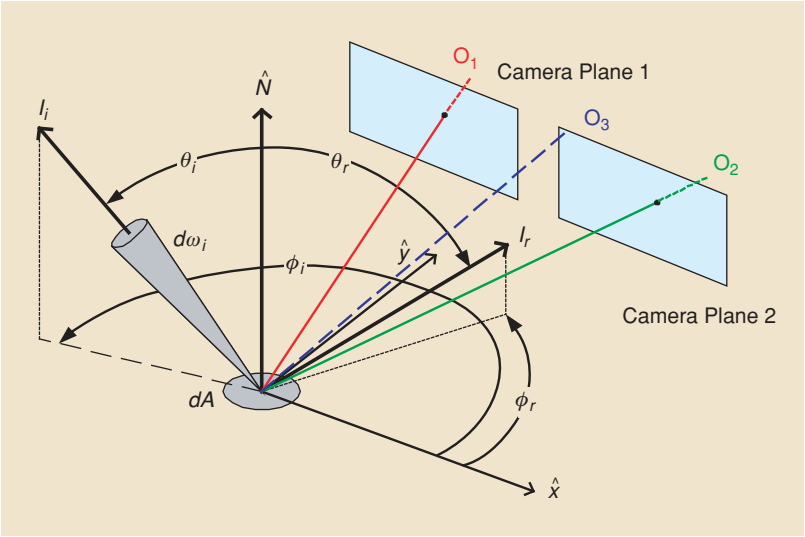
PRINCIPLE AND MAJOR RESEARCH ISSUES

Central to IBR is the concept of the seven-dimensional (7-D) plenoptic function, $P_7 = (V_x, V_y, V_z, \theta, \phi, \lambda, \tau)$ [1], which describes all the radiant energy that is perceived at any 3-D viewing point (V_x, V_y, V_z), from every possible angle (θ, ϕ) for every wavelength λ and at any time τ. Based on this function, theoretically, novel views at different positions and time can be reconstructed from its samples, provided that the sample rate is sufficiently high. Because of the multidimensional nature of image-based representations and scene geometry, much research has been devoted to the efficient capturing, sampling, rendering, and compression of IBR.



[FIG2] Spectrum of IBR representations.

To appreciate the principle of IBR, let us consider the image formation process as shown in Figure 3. As shown in the figure, incident light with a radiance of I_i impinges on a small surface of an object with a normal $\hat{N}$ and an infinitesimal small area dA. The irradiance of the incident light, which is the incident flux per unit surface area, is $E_i = I_i(\cos\theta_i)d\omega_i$, where θ_i is the angle between I_i and $\hat{N}$ and $d\omega_i$ is a small solid angle in the direction of I_i. Let the reflected radiance at an angle θ_r be I_r. The ratio I_r/E_i is known as the bidirectional reflectivity, $\rho(\theta_i, \theta_r)$, which is a property of the object.



[FIG3] **Image formation process.**

For a given lighting and geometry, a camera will capture rays from the objects and record them as pixels on its image plane, such as camera planes 1 and 2 in Figure 3. The area dA will be imaged by different cameras with different reflected radiance I_i, which is a function of the spherical coordinate (θ_r, ϕ_r). If I_i is bandlimited, then we can recover it from a sufficient number of samples in (θ_r, ϕ_r). If we know the exact geometry and lighting, we can estimate the bidirectional reflectivity. This technique has been used to model tree leaves [28] and other physical materials using laser scanners and controlled lighting. Once the geometry and surface property are known, we can render them using conventional graphics techniques or more advanced techniques such as shadow lightfields to be discussed in the "Interactive IBR" section. If we can estimate the geometry of the scene, say in form of depth maps, then we can use the samples captured by the cameras (for instance, O_1 and O_2 in Figure 3) to interpolate the value for that in O_3. In the simplest case where we do not have an accurate geometry like a reasonably accurate depth map, we can approximate the surface of the object by a plane parallel to a 2-D camera plane as in lightfields. The combined effect of geometry, lighting, and surface property give rise to a function I_r of seven dimensions, which is the plenoptic function. It is apparent that significantly more samples are required to recover the plenoptic function, if the geometry is unknown. One approach is to restrict the viewing freedom of the users so that the dimensionality of the representations can be reduced. For example, lightfields [8] or lumigraphs [9] are elegant 4-D IBR representations, where images on a 2-D camera plane are taken (Figure 1). Other IBR representations include the 2-D panorama [6], McMillan and Bishop's plenoptic modeling [5], 4-D ray-space representation [62], [63], the 3-D concentric mosaics [7], etc. Comprehensive reviews of the problems of capturing, sampling, and compression of IBR are available in [2] and [3]. A more recent update can be found in [4].

Although most of these representations do not employ much geometric information, the renderings are of very high quality. Motivated by the potential of IBR, researchers start to look for more general dynamic representations [11]–[18] as well as methods to reduce the number of samples or cameras required. Early attempts, called panoramic videos [18], are mostly based on 2-D panoramas. More recently, there were attempts to construct lightfield video systems for different applications and characteristics. These include the Stanford multicamera array [13], the 3-D rendering system of Naemura et al. [14], the (8×8) lightfield camera of Yang et al. [15], and (8×6) self-reconfigurable camera array of Zhang and Chen [17]. The Stanford array consists of more than 100 cameras and is intended for large-environment applications. It uses low-cost CMOS sensors and dedicated hardware for real-time compression. The systems in [14] and [15] consist of, respectively, 16 and 64 cameras and are intended for real-time rendering applications. In [17], a large self-reconfigurable camera array of 48 (8×6) cameras was built. Given the virtual view point, the cameras move on a set of rails to perform active rearranged capturing to improve the rendering quality. In [12], a simplified lightfield for dynamic environments (SDLF) was proposed where videos at regularly placed locations along a series of line segments were captured. The main motivation is to reduce the large dimensionality and excessive hardware cost in capturing dynamic representations. Because of the close relationship between the SDLF and traditional videos, it is also referred to as "plenoptic videos." Despite the simplification employed, plenoptic videos can still provide a continuum of viewpoints, significant parallax, and lighting changes.

A difficult problem in rendering lightfields and plenoptic videos is the excessive artifacts due to depth discontinuity. If the scene is free of occlusions, then the concept of plenoptic sampling [29] can be applied to determine the sampling rate in the camera plane. Unfortunately, because of depth discontinuities around object boundaries, the sampling rate is usually insufficient (due to the limited number of cameras and their physical separations). Significant rendering artifacts due to occlusion will result. Recently, representations with depth maps and object segmentation information [10], [11], [16], [23] have received great attention because they help to reduce rendering artifacts at depth discontinuities in large environmental modeling. Reliable methods to compute the depth/segmentation

information and how they are incorporated into the rendering systems are subjects of intense focus.

A pioneer in this area is the work of Goldlücke et al. [16], who used a subset of the Stanford multicamera array for acquiring and rendering dynamic scenes. The cameras were calibrated so that depth maps can be used to warp sample views to new views. The disparities of a given view were computed by averaging the disparities computed between this image and those from all other cameras. Due to image noise and blocking artifacts from the MPEG-encoded input images, it is difficult to produce depth maps that are correct to within a pixel at the boundaries.

In [11], an eight-camera video capturing system was constructed. High-resolution FireWire PtGrey color cameras are used to capture $1,024 \times 768$ video at 15 frames/s. Two "concentrator" units built by PtGrey are used to synchronize all the cameras and stream the uncompressed videos to a bank of hard disks through fiber optic cables. Inspired by layered depth images and sprites with depth [25], a two-layer representation—foreground and background—with depth, color, and matting information was employed. Stereo algorithm is used to compute the dense depth map for each image and detect depth continuities. Boundary strips are created around depth discontinuities. Bayesian matting [30] is used to estimate the depths, colors, and opacities (alpha values) of the foreground and background within these strips. Since the cameras are arranged along a one-dimensional (1-D) arc, during rendering, the two reference views nearest to the novel view are chosen, warped, and combined for view synthesis. In the stereo algorithm for finding the depth maps, images at the same time instant are smoothed using a variant of anisotropic diffusion and then segmented using a variant of the mean shift-based color segmentation algorithm. The mean depth of each segment is first computed using the centrally located camera as the global coordinate frame. The segment matching error function is computed in a similar manner as the disparity space image (DSI). Finally, locally computed disparities are averaged or smoothed, subject to projection consistency across images.

In [23] and [24], an object-based approach to plenoptic videos was proposed, where the plenoptic video sequences are segmented into IBR objects, each with its image sequence, depth map, and other relevant information such as shape information. By incorporating the depth and segmentation information, renderings with very good quality are obtained. Furthermore, desirable functionalities such as scalability of contents, error resilience, and interactivity with individual IBR objects (including random access at the object level) can also be supported. As a result, IBR objects can be processed, rendered, compressed, and transmitted separately. To give the reader an idea of these operations, more details of the object-based system in [23] and [24] will be briefly described below as an illustration.

[FIG4] **Snapshots of the plenoptic video *Dance*.**

THE OBJECT-BASED APPROACH

SYSTEM OVERVIEW

In [23] and [24], a plenoptic video system used to capture dynamic scenes was constructed. This system consists of two linear arrays of calibrated cameras, each hosting six JVC DR-DVP9ah video cameras. More arrays can be connected together to form longer segments. Because the videos are recorded on tapes, the system is also suitable for outdoor dynamic scenes. The use of multiple linear arrays allows the user to have more viewing freedom in sport events and other live performances. After capturing, the video data stored on the tapes can be transmitted to computers through a FireWire interface. Figure 4 shows snapshots of plenoptic videos, titled *Dance*, captured by this system. This real-scene plenoptic video has a resolution of 720×576 pixels in 24-b RGB format.

It is assumed that each image pixel in a lightfield has a color as well as a depth value. This representation, compared with using a global depth map, is less sensitive to errors in camera position and depth maps encountered in practical multicamera systems. Though plenoptic sampling suggests that dense sampling of image-based representation can tolerate depth variation within the segments by interpolating the plenoptic function, a very-high-resolution depth map is usually unavailable. Efficient methods are required to reduce the rendering artifacts at depth discontinuities. It was found in [10] that by properly segmenting the videos into image-based or layered objects at different depths and rendering them separately, the rendering quality in a large environment can be considerably improved. From the segmented objects, approximate depth information for each IBR object can be estimated to render new views at different viewpoints. Due to possible segmentation errors around boundaries and finite sampling at depth discontinuities, natural matting should be adopted to improve the rendering quality at object boundaries when mixing IBR objects. By using the estimated alpha map and texture, it is also convenient to composite the image-based objects onto the background of the original or other plenoptic videos. The important issues of segmentation and matting are further elaborated upon in the next section. This object-based approach not only improves the rendering quality, but also provides object-based functionalities in coding and other processing applications. In particular, the IBR objects were encoded individually by an MPEG-4-like object-based coding scheme [31] that includes additional information such as depth maps and alpha maps to facilitate rendering. The processing issue is treated later in the "Object-Based Plenoptic Video Processing" section.

OBJECT SEGMENTATION, TRACKING, AND MATTING

OBJECT SEGMENTATION AND TRACKING

As mentioned earlier, objects at large depth differences are segmented into layers and are compressed and rendered separately. In [23] and [24], an initial segmentation of the objects is first obtained using a semi-automatic approach [32]. Tracking techniques are then employed to segment the objects at other video streams and subsequent time instants, using the level-set method [33], [34]. The basic idea is to deform a given curve, surface, or image according to a partial differentiation equation (PDE) and arrive at the desired result as the steady-state solution of this PDE. The problem can also be viewed as minimizing a certain energy function

$$U_I(C) = \int_I F(C, \boldsymbol{x}) dx \tag{1}$$

as a function of a curve or surface C. The subscript indicates that the energy is computed from the given images I. Usually, $F(C, \boldsymbol{x})$ is designed to measure the deviation of the desired curve from C at point $\boldsymbol{x}$. To minimize the functional in (1), the variational approach can be employed to convert it to a PDE. A necessary condition for C to be a local minimum of the functional is $U_I'(C) = 0$. To solve it numerically, we usually start with an initial curve C_0 and let it evolve over a fictitious time variable t according to a PDE, which depends on the derivative $U_I'(C)$ as follows:

$$\frac{\partial C(t)}{\partial t} = U_I'(C(t)). \tag{2}$$

Since the PDE may be singular at certain points, it is usually solved using the level-set method [33], where a curve or surface is represented in "implicit form" such as the zero level sets of a higher-dimensional function. More formally, the time evolution of curves $C(\boldsymbol{x}, t)$ is represented as the level-set of an embedding function $\phi(\boldsymbol{x}, t) : L_c(\boldsymbol{x}, t) := \{(\boldsymbol{x}, t) \in R^3 : \phi(\boldsymbol{x}, t) = c\}$, where c is a given real constant. Equation (2) can be rewritten as a PDE of $\phi(\boldsymbol{x}, t)$, and its time evolution is computed numerically by solving an appropriate discontinued PDE for $\phi(\boldsymbol{x}, t)$ at a sufficiently small time interval. The desired solution is obtained when the PDE converges at sufficiently large value of n. In [23] and [24], the following energy function is used:

$$U_I(C) = \alpha \int_I C_{\text{inside}} dxdy - \beta \int_I C_{\text{outside}} dxdy + \lambda \text{Length}(C), \tag{3}$$

where $C_{\text{inside}}(x, y)$ and $C_{\text{outside}}(x, y)$ are two functions designed, respectively, to control the expansion and contraction of the curve C at location (x, y) and Length (C) measures the length of the curve. If the pixel values inside and outside the curve are assumed to be independent and Gaussian distributed with means c_{in} and c_{out}, respectively, then the PDE can be written as

$$\left.\frac{\partial \phi}{\partial t}\right|_{(x,y)} = \alpha \left(u_{(x,y)} - c_{in}\right)^2 - \beta \left(u_{(x,y)} - c_{out}\right)^2 + \lambda \cdot div\left(\frac{\nabla \phi}{|\nabla \phi|}\right), \tag{4}$$

where α, β and λ are positive parameters, $u_{(x,y)}$ is the value of pixel (x, y), and c_{in} and c_{out} denote, respectively, the driving forces inside and outside the curve C. The third term, which is derived from Length(C), makes the curve smooth and continuous.

There are two different methods for determining c_{in} and c_{out}, namely global-based and local-based methods. The global-based method in [34] utilizes all the pixels to drive the curve C, where c_{in} and c_{out} denote, respectively, the means of all pixels inside and outside the curve C. The global-based method has fast evolution speed and it is less sensitive to noise. However, some fine features along the object's boundary to be tracked may be lost. In contrast, local-based methods use local features of the image to cope with objects having a nonuniform energy distribution. In [23], the global and local methods are combined by further smoothing the local features. In [24], depth information computed using stereo matching is also incorporated in this level, which greatly improves the reliability of this method. A brief introduction to stereo matching will be given later in the "Stereo-Matching Algorithms and Multiview Synthesis" section.

IMAGE-BASED RENDERING REFERS TO A COLLECTION OF TECHNIQUES AND REPRESENTATIONS THAT ALLOW 3-D SCENES AND OBJECTS TO BE VISUALIZED IN A REALISTIC WAY WITHOUT FULL 3-D MODEL RECONSTRUCTION.

OBJECT MATTING

Due to possible segmentation errors around object boundaries and finite sampling at depth discontinuities, it is preferred to calculate soft, instead of a hard, membership functions between image-based objects and the background. In other words, the boundary pixels are assumed to be a linear combination of the corresponding pixels from the foreground and background, $I = \alpha F + (1 - \alpha)B$, where I, F, and B are the pixel's composite, foreground, and background colors, and $0 \leq \alpha \leq 1$ is the pixel's opacity component or the alpha map. Using this model, it is possible to matte a given object with the original at different views and other background.

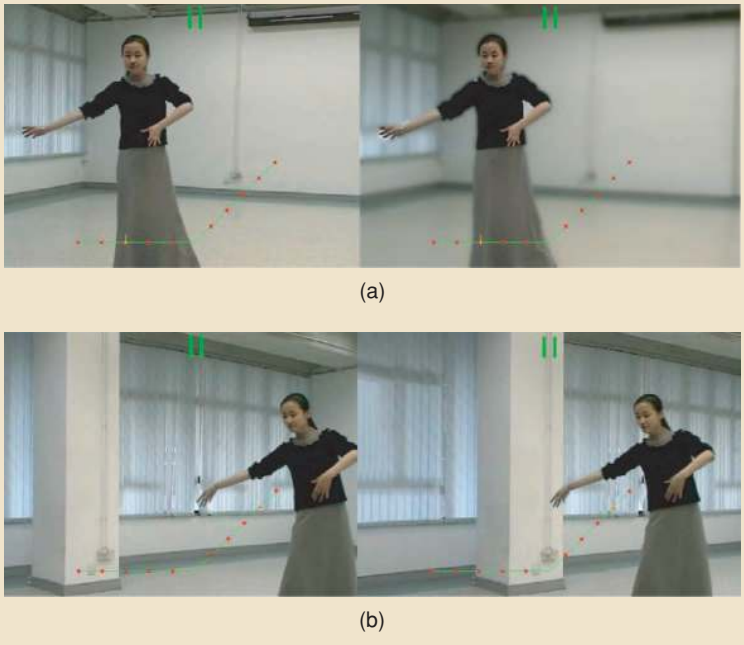
In [23] and [24], the matte was computed using the Bayesian approach [30]. By assuming that $F, B,$ and α to be independent, one gets a set of equations in the estimates of α, F, and B which can be solved iteratively. The initial

values of F and B for each pixel in the conventional method are usually provided by the user through certain user interfaces. In the object-based approach [24], they are obtained from the segmentation and tracking results. Figure 5(a) shows the tracking result obtained in [23] and [24]. For each frame, the initial curve C_0 is the tracking result of the previous frame. The level-set contour evolution is implemented using the narrow-band method. An example depth map and alpha map of the image-based object are illustrated in Figure 5(b).

OBJECT-BASED PLENOPTIC VIDEO PROCESSING

Since images and videos are special cases of the plenoptic function, many conventional image processing algorithms such as coding, segmentation, etc. have similar analogy in IBR. These generalizations are referred to here as plenoptic video processing. In particular, the associated processing operations in the object-based approach are referred to as objected-based plenoptic processing.

[FIG5] (a) Tracking results of our method. (b) Example alpha and depth maps.



[FIG6] The rendered images before and after processing: (a) background defocusing processing and (b) background transformation processing.

The main difference between plenoptic video processing and a single image is the need to ensure the image consistency constraints in multiple views of the same object. Ideally, when a group of pixels of an object in a given image is modified, then the "corresponding" pixels in other images should also be modified consistently. If scene geometry is also available, then the lighting and other physical constraints should also be observed. Due to the difficulty of accurately acquiring scene geometry and other physical parameters, it is unavoidable that the capability of automatic IBR processing be limited in representations with approximate geometry only. In other words, additional prior information must be provided by the users through appropriately designed user interfaces and tools.

In what follows, we briefly describe the generalizations of some commonly used image processing algorithms to the IBR case under the object-based framework. In principle, one should determine the correspondence between image pixels and process them as a whole. Under the object-based framework, similar objects are segmented and grouped together. Therefore, the image consistency may be approximately satisfied by processing the image pixels from the IBR object as a whole or we can make use of the correspondence computed from the depth maps.

BACKGROUND DEFOCUSING

Background defocusing is a special effect where the background is blurred, so that certain objects can be popped up. Figure 6(a) shows the defocusing results, where the background is smoothed using a low-pass filter of size (5×5). It can be seen from Figure 6(a) that the object boundaries are well preserved because the smoothing operation is applied only to the background object, in contrast to conventional smoothing operations which will result in blurring of the entire image.

BACKGROUND TRANSFORMATION

With the help of the alpha map of an object, it is possible to matte a given object to a transformed background to create a special rotation effect. Figure 6(b) shows the original rendered image and another one with the object "dancer" being pasted to a rotated background.

These operations will greatly increase the viewing freedom and support zooming, panning, looking upwards and downwards, etc. Moreover, by synthesizing different views appropriately, we can also generate views of stereo and multiview display for 3-D and multiview TVs. Details of other

operations such as object completion, inpainting enhancement, etc. can be found in [24]. If shadows and other lighting effects are desirable, further postprocessing may be necessary. This is difficult to carry out without the knowledge of the geometry of the scene. Therefore, for interactive rendering and relighting, capturing a rough geometry of the scene is of great importance, and it will be briefly reviewed in the "Interactive IBR" section.

STEREO-MATCHING ALGORITHMS AND MULTIVIEW SYNTHESIS

Stereo matching, which infers 3-D scene geometry from two images, is an important tool in intermediate view synthesis and IBR.

ASSUMPTIONS AND REPRESENTATIONS

Due to the difficulties in handling scenes with specularities, reflective surfaces, or transparency, most stereo-matching algorithms assume that the scene is Lambertian or intensity-invariant from different viewpoints. Moreover, the camera is usually assumed to be calibrated so that more reliable matching can be obtained using the epipolar geometry constraint. For example, in Figure 7(a), a point P is imaged as pixels (x, y) and (x', y') by two cameras with centers O_1 and O_2, respectively. It can be seen that the corresponding pixel in image m of (x, y) in image r lies on the line l_2, which also lies on the plane containing (x, y), O_1, and O_2. If the position, orientation, and other relevant parameters of the two cameras O_1, and O_2 are known through camera calibration, then one can match the intensity of (x, y), $I(x, y)$, against those along line l_2 by assuming that the surface S is Lambertian. Furthermore, the image planes are transformed, a process called rectification, such that pairs of conjugate epipolar lines (l_1, l_2) become collinear and parallel to one of the image axes [Figure 7(b)]. In the rectified images, the problem of computing stereo correspondences is reduced to a 1-D search problem along the horizontal raster lines of the rectified images. Consider a rectified reference image r and another image to be matched m as shown in Figure 7(a). The correspondence between the pixel at (x, y) in r and the pixel (x', y') in m is given by

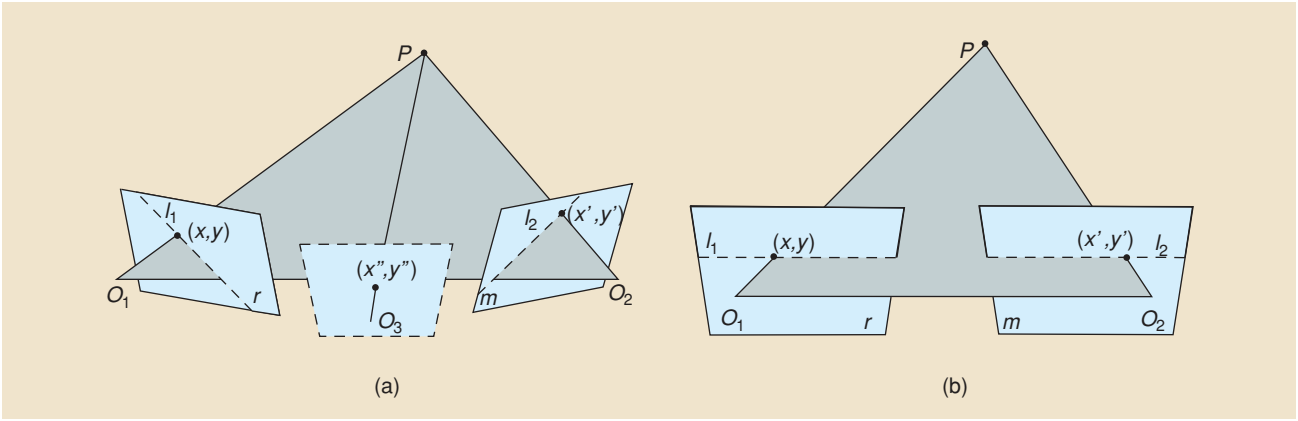
$$x' = x + d(x, y), \quad y' = y, \tag{5}$$

where $d(x, y)$ is the disparity. The univalued map $\{d(x, y) : \forall (x, y)\ in\ r\}$ is called the disparity map of r. Once the disparity map is determined, the image of P at other camera positions such as O_3 in Figure 7(b) can be determined, if it is not occluded and S is Lambertian, i.e., $I_1(x, y) = I_2(x', y') = I_3(x'', y'')$. If the camera is calibrated, the depth value of P and hence the pixel (x, y) can be determined. The collection of depth values for each pixel in an image forms its depth map. The goal of a stereo correspondence or matching algorithm is to estimate a univalued function in the disparity space $D(r) = (x, y, d)$ that best describes the shape of the surfaces in the scene.

STEREO-MATCHING ALGORITHMS

Scharstein and Szeliski [35] gave an extensive survey on stereo algorithms and provided an online evaluation based on the Middlebury Stereo Evaluation (MSE) data set. Since then, many new and novel approaches to stereo-matching algorithms have been developed and evaluated online with the MSE data set. According to Scharstein and Szeliski's taxonomy [35], many existing stereo-matching algorithms perform all or some of the following four steps: 1) matching cost computation, 2) cost (support) aggregation, 3) disparity computation/optimization, and 4) disparity refinement.

From the Lambertian assumption, the desired disparity $d(x, y)$ of pixel (x, y) should minimize the intensity difference $I(x, y) - I(x', y')$. Therefore, it is necessary to compute some matching cost between $I(x, y)$ and $I(\tilde{x}, \tilde{y})$, where $(\tilde{x}, \tilde{y})$ is a possible candidate in the disparity space $D(r)$, such as nearby pixels along l_2 in Figure 7. Various matching costs have been proposed for reliable matching cost computation. To reduce noise, matching costs of similar adjacent pixels are usually aggregated locally (cost aggregation). After that, the disparity map is computed by minimizing some energy or cost function to measure the matching of the two images while preserving certain smoothness constraints of the surfaces (disparity computation/optimization). Finally, the computed disparity map is further refined to detect occluded pixels and infer its depth values from adjacent pixels (disparity refinement).



[FIG7] Stereo matching: (a) epipolar geometry and (b) rectification.

[TABLE 1] SUMMARY OF SOME TOP-RANKED STEREO-MATCHING ALGORITHMS (AD: ABSOLUTE INTENSITY DIFFERENCE, SAD: SUM OF ABSOLUTE DIFFERENCE, WTA: WINNER TAKE ALL, GRAD: GRADIENT-BASED MEASURE).

METHOD	MATCHING COST	AGGREGATION	OPTIMIZATION
KLAUS ET AL. [36]	SAD+GRAD	SQUARE WINDOW (SW)	BP ON PLANE
YANG ET AL. [37]	AD	SW WEIGHTED BY COLOR AND SPATIAL	BP
SUN ET AL. [38]	SD	SW	BP
HIRSCHMULLER [39]	MUTUAL INFORMATION	SW	SEMI-GLOBAL MATCHING
LEI ET AL. [40]	AD	NONE	DP ON REGION TREE
YOON AND KWEON [41]	AD	SW WEIGHTED BY COLOR AND SPATIAL	WTA
KOLMOGOROV AND ZABIH [42]	SD	NONE	GC

Global optimization techniques like graph cut (GC) [11], [42] and belief propagation (BP) [36]–[38] are widely used for the disparity optimization step in top-rank methods. On the other hand, segment-based methods [36]–[41] have also attracted considerable attention due to their good performance on nontexture area. There are also attempts for high-quality real-time stereo with the help of variants of dynamic programming [40] or semiglobal matching techniques [39]. Once the depth maps are obtained, new views can be synthesized by interpolating the pixel values from nearby images. If there are holes in the rendered images, they have to be filled by neighboring pixels using inpainting algorithms [64], [65]. In the object-based approach, pixels from foreground and background objects will be matted together to reduce artifacts. Table 1 gives a summary of some top-ranked stereo-matching algorithms and their corresponding taxonomy according to Scharstein and Szeliski's survey.

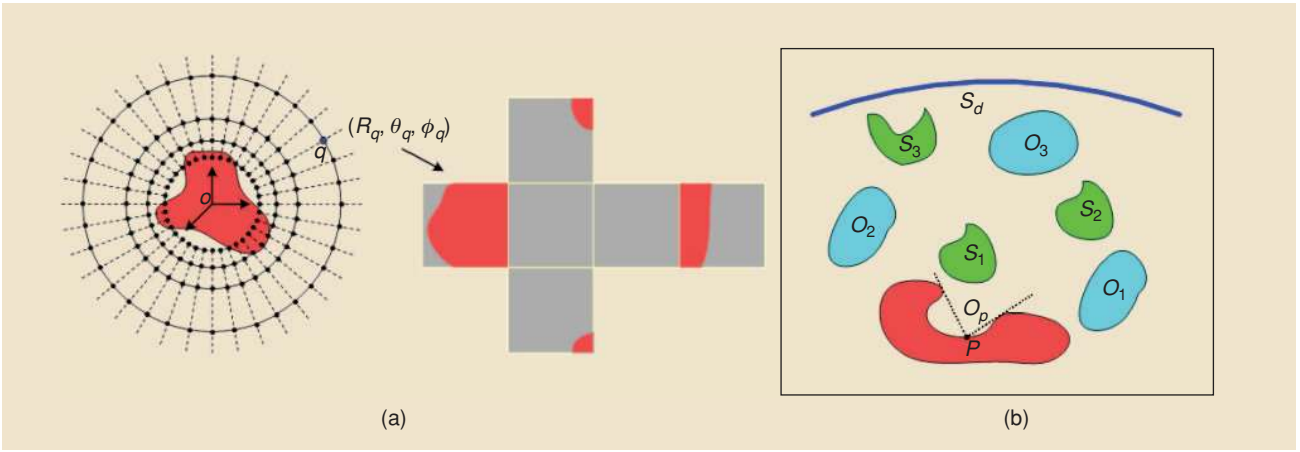
INTERACTIVE IBR

As mentioned earlier, representations with explicit geometry can provide increased functionalities such as relighting and interactive placement of objects. Fast rendering algorithms and methods for capturing static and dynamic 3-D models are two important practical problems. A brief review of these subjects will be given below.

In principle, if the geometry and surface property (BRDF) of an object are available, precomputed shadow fields can be employed to speed up the relighting and soft shadow generation processes. For nonreflective objects that do not change their shapes, the geometry of the object can be acquired by a 3-D laser scanner. The capturing of its surface property is also possible, though involved [43]. While there has recently been considerable progress in relighting human faces using structured lighting [44], the extraction of 3-D dynamic models—such as human beings—using a multiple camera array and computer vision techniques such as object profiles [45] or level-set methods [46] are of great interest in dynamic scenes. Moreover, the panoramic video-based environment maps can be combined with these dynamic 3-D models to achieve fast interactive rendering. These are useful in incorporating relighting and soft shadow capabilities in interactive augmented reality systems.

FAST RENDERING AND COMPRESSION ALGORITHMS

We shall briefly explain how shadow fields of individual entity and the radiance fields of local light sources of the scene can be used for interactive relighting and soft shadow generation. A shadow field is a kind of plenoptic function that describes the shadowing or occluding effects of an individual scene entity at sampled points surrounding it. Figure 8(a) shows the shadow field or object occlusion field (OOF) of an object. At each point q that surrounds the object with spherical coordinates (R_q, θ_q, ϕ_q), we can describe the occlusion or shadowing effect of the object as a 2-D function in the polar coordinates



[FIG8] (a) Sampling of shadow field with the occlusion effect being represented as cube map. (b) Combining shadow fields to obtain the incident radiance distribution. Point p on an object has self-visibility $\mathbf{O}_p$. There are three objects in the scene with OOFs $\mathbf{O}_1$, $\mathbf{O}_2$, $\mathbf{O}_3$, and there are three local sources $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathbf{S}_3$ and a distant lighting $\mathbf{S}_d$.

$(\theta_{\text{in}}, \phi_{\text{in}})$. Alternatively, we can use a cube map to represent this 2-D function, as shown in Figure 8(a). If the shape of the object is time-varying, then the OOF is a six-dimensional (6-D) plenoptic function $P(R_q, \theta_q, \phi_q, \theta_{in}, \phi_{in}, t)$. The same concept can be used to describe the effect of a local light source, which is referred to as a source radiance field (SRF).

The SRF of each light source and OOF of each local object are individually precomputed at sampled locations in its surrounding space, as shown in Figure 8(b). Unlike traditional approaches for soft shadow generation methods [19] and precomputed radiance transfer (PRT)-based methods [20], [21], the precomputed shadow fields approach [22] represents the shadowing effects of a single scene element in an empty space, and thus can be precomputed independent of scene configuration. Another important property of shadow fields is that they can be quickly combined in run time for soft-shadow generation. Therefore, this method is able to handle motion of objects and dynamic local light sources.

More precisely, the incident radiance distribution of a point, say p in Figure 8(b), is computed from the contributions of all the light sources using their SRFs and the objects' OOFs that lie between the light source under consideration and the given point. For example, the contribution from source S_1 is just $\mathrm{S}_1(p)^*\mathrm{O}_\mathrm{p}(p)$, where $\mathrm{S}_1(p)$ is the SRF of S_1 for point p, $\mathrm{O}_\mathrm{p}(p)$ is the OOF for point p due to self-occlusion, and * denote element-wise product of their cube maps. Since the SRF and OOF are precomputed for a given orientation of the light sources or objects, certain coordinate transformations and interpolation are required to compute the corresponding value at the desired direction, say p in this example. Similarly, the contribution from the local light source S_3 is $S_3(p)^*\mathrm{O}_2(p)^*\mathrm{O}_1(p)^*\mathrm{O}_\mathrm{p}(p)$, because objects O_2 and O_1 between them may occlude the rays from S_3 to p. By adding the contributions of each light source in the scene, the final incoming radiance distribution that determines the soft shadow at the point p can be determined. For time-varying light sources—for instance, some kind of video textures—and dynamic 3-D objects, a different set of cube maps will be required at different times and this poses a significant storage problem.

Such cube maps are also frequently used to model environmental maps of distant objects, and they are also referred to as panoramas. For light sources or objects that vary with time, the resulting time-varying cube maps become a panoramic video. Therefore, the compression of SRF and OOFs is closely related to the compression of panoramic videos, except that there is additional correlation between cube maps at adjacent spatial locations. The compression of panoramic videos using a modified MPEG-2 algorithm has been studied previously in [18] and it can be modified to compress these time-varying cube maps. It is expected that vector quantization (VQ) with possible temporal/spatial prediction will provide a fast rendering speed at the expense of lower compression ratio. For the object occlusion field (OOF), the image mainly consists of the alpha values describing the occlusion effect of the object. For some applications, it just assumes binary values, and the shape coding algorithm in MPEG-4 video object coding can be utilized.

As mentioned earlier, time-varying or dynamic 3-D models require a considerable amount of storage and limit the complexity of a scene that can be handled. While 3-D mesh compression has been extensively studied in the literature, the compression of dynamic 3-D meshes is relatively new [47], [48]. A very good survey and a sophisticated compression algorithm that addressed the dynamic connectivity of 3-D mesh were given in [48]. Although this algorithm is very flexible, its performance will be affected considerably when input mesh structure changes substantially between frames. Improved dynamic mesh compression is thus an important area of research.

CAPTURING OF 3-D DYNAMIC MODELS

APPROACHES

Volumetric representations are a simple and robust method for reconstruction from silhouettes and they have been extensively used [49], [50]. However, to achieve better reconstruction precision, the size of spatial partition must be decreased, which results in a tremendous increase in complexity of the reconstructed 3-D models. Another approximate geometric representation is based on the visual hull [51], which is constructed by casting the visible silhouette information from a collection of input images to 3-D space and intersecting the cast volume. Recently, Matusik et al. [52] proposed a representation called image-based visual hulls (IBVHs) for fast dynamic scene rendering based on the visual hull without explicit geometric or volumetric construction. These approaches, which approximate visual hull with polyhedrons, outperform the volumetric approaches in terms of accuracy and computation complexity, and they maintain comparable robustness for objects with complex topologies. A disadvantage of visual hulls is that concavities that cannot be observed as silhouettes cannot be handled. For objects with smooth surfaces, more precise estimation of surface points can be achieved by a differential analysis on the deformation of the object contours [53], [54]. With sufficient and well-distributed viewpoints, these approaches can give high-quality estimation of the surface points, especially for objects with smooth surfaces. However, few of them offer similar robustness as the volumetric approaches for objects having complicated topologies. Recently, Liang and Wong [55] employed epipolar parameterization in identifying a well-defined local tangent basis to handle fairly complicated shapes.

Apart from using computer vision techniques, the geometry of the nonreflective object that does not change with time can also be captured using 3-D laser scanners. Computer vision techniques, on the other hand, are applicable to dynamic objects and potentially will require a lower hardware cost.

CAPTURING SYSTEMS

The Virtualized Reality project by Kanade et al. [56] was probably the first attempt to capture dynamic 3-D scenes and render them at arbitrary viewpoints. Their first system consists of 51 cameras arranged around a 5-m geodesic dome. The shape of the scene being modeled was extracted by computing stereo

depth maps, at each vantage point, using immediate neighboring cameras and their multibaseline algorithm. In one version, the closest reference view is used for view synthesis with two other nearest views used to fill possible holes. To simplify programming and hardware rendering in a later version, all the depth maps were merged into a single 3-D model using volumetric integration. Temporal information was also incorporated later in [57] to achieve spatial-temporal view interpolation.

Despite the advances in 3-D reconstruction algorithms, reliable computation of 3-D scene models remains difficult. In many situations, only a certain object in the scene—such as the human body—is of interest. Here prior knowledge of the object model can be used to simplify the rendering process or improve the reconstruction quality. Therefore, several model-based systems for human motion capture were proposed. In [58], an actor's movement is captured using multiple video cameras at 15 frames/s and a resolution of 320×240. The cameras are synchronized, calibrated, and arranged in a convergent configuration. A binary image of the silhouettes of the human actor in each image captured is computed by comparing the pixel values with those from the fixed background. The human model consists of 16 articulated body parts, each represented by a triangular mesh, and 17 joints. 35 parameters are used to define the body pose. The model parameters are computed by a nonlinear minimization approach, where the global translation and rotation are estimated first, followed by other parameters such as head and hip joint rotations, etc. Since the geometry extracted is inexact, a consistent texture map is used for rendering.

In the marker-less human motion transfer system of Cheung et al. [59], a more detailed person-specific model is used and both silhouette and color information are employed for motion tracking. The shape of a person is first acquired from a video taken on a rotating turntable using the "shape-from-silhouette across time" (SFSAT) algorithm [60]. The model has 22 degrees of freedom. The joint skeleton is then estimated one element at a time. During motion capture, the shape is recovered using the SFSAT algorithm and is aligned to the human model. The captured motion of one person can be directly transferred to another through the joint motions. A more detailed summary of these and other related systems can be found in [4] and [61].

CONCLUSIONS

A brief review of the technological advances and future challenges of IBR has been presented. This includes the basic principles, key research problems, important techniques, state-of-the-art systems, and possible future extensions of IBR. We hope this article gives readers a grasp of this emerging technology and contributes to the further development of multiview systems, 3DTV, and other immersive viewing applications.

ACKNOWLEDGMENT

This work was supported in part by the Research Grant Council of Hong Kong Special Administrative Region, China.

AUTHORS

S.C. Chan (scchan@eee.hku.hk) received the B.Sc. (Eng.) and Ph.D. degrees from the University of Hong Kong in 1986 and 1992, respectively. He joined the University of Hong Kong in 1994 and is now an associate professor. He was a visiting researcher in Microsoft Corporation, Redmond, USA, and Microsoft, Beijing, China, in 1998 and 1999, respectively. His research interests include fast transform algorithms, filter design and realization, multirate signal processing, and image-based rendering. He is currently a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He was chair of the IEEE Hong Kong Chapter of Signal Processing from 2000–2002.

Heung-Yeung Shum (hshum@microsoft.com) received a doctorate in robotics from the School of Computer Science at Carnegie Mellon University in Pittsburgh, Pennsylvania. He is a corporate vice president at Microsoft. He oversees the research activities at Microsoft Research Asia and the lab's collaborations with universities in Asia Pacific. Recently, Dr. Shum has taken the additional responsibility of driving the long-term and short-term technology investments in search and advertising at Microsoft. He is an IEEE Fellow and an American Computational Machinery (ACM) Fellow. He serves on the editorial board of the *International Journal of Computer Vision* and is a program chair of the International Conference of Computer Vision (ICCV) 2007. He has published more than 100 papers in computer vision, computer graphics, pattern recognition, statistical learning, and robotics. He holds more than 50 U.S. patents.

King-To Ng (ktng@graduate.hku.hk) received the B.Eng. degree in computer engineering from the City University of Hong Kong in 1994 and the M.Phil. and Ph.D. degrees in electrical and electronic engineering from the University of Hong Kong, in 1998 and 2003, respectively. In 2004, he worked as a visiting associate researcher at Microsoft Research Asia, Beijing, China. Currently, he is a postdoctoral fellow in the Department of Electrical an Electronic Engineering, The University of Hong Kong. His research interests include visual communication, image-based rendering, and video broadcast and transmission.

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, pp. 3–20, 1991.

[2] H.Y. Shum, S.B. Kang, and S.C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[3] C. Zhang and T. Chen, "A survey on image-based rendering—Representation, sampling and compression," in *EURASIP Signal Processing: Image Commun.*, vol. 19, no. 1, pp. 1–28, Jan. 2004.

[4] H.Y. Shum, S.C. Chan, and S.B. Kang, *Image-Based Rendering*. New York: Springer-Verlag, 2006.

[5] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1995, pp. 39–46.

[6] R. Szeliski and H.Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1997, pp. 251–258.

[7] H.Y. Shum and L.W. He, "Rendering with concentric mosaics," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1999, pp. 299–306.

[8] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1996, pp. 31–42.

[9] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, "The lumigraph," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1996, pp. 43–54.

[10] H.Y. Shum, J. Sun, S. Yamazaki, Y. Lin, and C.K. Tang, "Pop-up light field: An interactive image-based modeling and rendering system," *ACM Trans. Graphics*, vol. 23, no. 2, pp. 143–162, Apr. 2004.

[11] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 2004, pp. 600–609.

[12] S.C. Chan, K.T. Ng, Z.F. Gan, K.L. Chan, and H.Y. Shum, "The plenoptic videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1650–1659, Dec. 2005.

[13] B. Wilburn, M. Smulski, H.H. Lee, and M. Horowitz, "The light field video camera," in *Proc. SPIE Electronic Imaging: Media Processors'2002*, vol. 4674, Jan. 2002, pp. 29–36.

[14] T. Naemura, J. Tago and H. Harashima, "Real-time video-based modeling and rendering of 3D scenes," *IEEE Comput. Graphics Applicat.*, vol. 22, no. 2, pp. 66–73, Mar.–Apr. 2002.

[15] J.C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera," in *Proc. Eurographics Workshop on Rendering*, 2002, pp. 77–86.

[16] B. Goldlücke, M. Magnor, and B. Wilburn, "Hardware-accelerated dynamic light field rendering," in *Proc. VMV'2002,* pp. 455–462.

[17] C. Zhang and T. Chen, "Active rearranged capturing of image-based rendering scenes-Theory and practice," in *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 520–531, Apr. 2007.

[18] K.T. Ng, S.C. Chan, and H.Y. Shum, "The data compression and transmission aspects of panoramic videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 82–95, Jan. 2005.

[19] M. Agrawala, R. Ramamoorthi, A. Heirich, and L. Moll, "Efficient image-based methods for rendering soft shadows," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, 2000, pp. 372–384.

[20] R. Ng, R. Ramamoorthi, and P. Hanrahan, "Triple product wavelet integrals for all-frequency relighting," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, 2004, pp. 477–487.

[21] P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low- frequency lighting environment," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, 2002, pp. 527–536.

[22] K. Zhou, Y. Hu, S. Lin, B. Guo, and H.Y. Shum, "Precomputed shadow fields for dynamic scenes," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, 2005, pp. 1196–1201.

[23] Z.F. Gan, S.C. Chan, K.T. Ng, and H.Y. Shum, "An object-based approach to plenoptic videos," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2005, pp. 3435–3438.

[24] S.C. Chan, Z.F. Gan, K.T. Ng, and H.Y. Shum, "An object-based approach to a class of dynamic image-based representations," submitted for publication.

[25] J. Shade, S. Gortler, L.W. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Orlando, FL, July 1998, pp. 231–242.

[26] C. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1999, pp. 291–298.

[27] P.E. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," in *Proc. Eurographics Workshop on Rendering*, 1998, pp. 150–116.

[28] L. Wang, W. Wang, J. Dorsey, X. Yang, B. Guo, and H.Y. Shum, "Real-time rendering of plant leaves," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, July 2005, pp. 712–719.

[29] J.X. Chai, X. Tong, S.C. Chan and H.Y. Shum, "Plenoptic sampling," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, July 2000, pp. 307–318.

[30] Y.Y. Chuang, B. Curless, D.H. Salesin, and R. Szelishi, "A Bayesian approach to digital matting," in *Proc. IEEE Conf. CVPR*, vol. 5, 2001, pp. 264–271.

[31] Q. Wu, K.T. Ng, S.C. Chan and H.Y. Shum, "On object-based compression for a class of dynamic image-based representations," in *Proc. IEEE Int. Conf. Image Processing*, Italy, vol. 3, Sept. 11–14, 2005, pp. 405–408.

[32] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum, "Lazy snapping," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, 2004, pp. 303–308.

[33] S.J. Osher and J.A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Compute. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.

[34] T.F. Chan and L.A. Vese, "Active contours without edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[35] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Computer Vision*, vol. 47, no. 1/2/3, pp. 7–42, 2002 [Online]. Available: http://www.middlebury.edu/stereo/

[36] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. ICPR*, vol. 3, 2006, pp. 15–18.

[37] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," in *Proc. IEEE Conf. CVPR*, June 2006, vol. 2, pp. 2347–2354.

[38] J. Sun, Y. Li, S.B. Kang, and H.Y. Shum, "Symmetric stereo matching for occlusion handling," in *Proc. IEEE Conf. CVPR*, vol. 2, 2005, pp. 399–406.

[39] H. Hirschmueller, "Stereo vision in structured environments by consistent semi-global matching," in *Proc. IEEE Conf. CVPR*, 2006, pp. 2386–2393.

[40] C. Lei, J. Selzer, and Y. Yang, "Region-tree based stereo using dynamic programming optimization" in *Proc. IEEE Conf. CVPR*, 2006, pp. 2378–2385.

[41] K.J. Yoon and I.S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 4, pp. 650–656, 2006.

[42] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. ICCV, 2001*, vol. 2, pp. 508–515.

[43] S. Rusinkiewicz, "Survey of BRDF representation for computer graphics," 1997 [Online]. Available: http://www.cs.princeton.edu/~smr/cs348c-97/surveypaper.html

[44] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec, "Performance relighting and reflectance transformation with time-multiplexed illumination," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, 2005, pp. 756–764.

[45] K.Y.K. Wong and R. Cipolla, "Reconstruction of sculpture from its profiles with unknown camera positions," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 381–389, Mar. 2004.

[46] S. Osher and N. Paragios, *Geometric Level Set Methods in Imaging, Vision, and Graphics.* New York: Springer-Verlag, 2003.

[47] J.E. Lengyel, "Compression of time-dependent geometry," in *ACM Symp. Interactive 3D Graphics*, pp. 89–96, 1999.

[48] S. Gupta, K. Sengupta, and A. Kassim, "Registration and partitioning-based compression of 3-D dynamic data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1144–1155, 2003.

[49] W. Martin and J. Aggarwal, "Volumetric descriptions of objects from multiple views," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, no. 2, pp. 150–158, 1983.

[50] B. García and B. Brunet, "3D reconstruction with projective octrees and epipolar geometry," in *Int. Conf. Computer Vision*, pp. 1067–1072, Jan. 1998.

[51] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 2, pp. 150–162, Feb. 1994.

[52] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, "Image-based visual hulls," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, July 2000, pp. 369–374.

[53] R. Cipolla and A. Blake, "Surface shape from the deformation of apparent contours," *Int. J. Computer Vision*, vol. 9, no. 2, pp. 83–112, 1992.

[54] R. Vaillant and O. Faugeras, "Using extremal boundaries for 3-D object modeling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 157–173, Feb. 1992.

[55] C. Liang and K.Y. Wong, "Robust recovery of shapes with unknown topology from dual space," *IEEE Trans. Pattern Anal. Machine Intell.*, to be published.

[56] T. Kanade, P. W. Rander, and P. J. Narayanan, "Virtualized reality: constructing virtual worlds from real scenes," *IEEE Multimedia, Immersive Telepresence*, vol. 4, no. 1, pp. 34–47, Jan. 1997.

[57] S. Vedula, S. Baker, and T. Kanade, "Image-based spatio-temporal modeling and view interpolation of dynamic events," *ACM Trans. Graphics*, vol. 24, no. 2, pp. 240–261, Apr. 2005.

[58] J. Carranza, C. Theobolt, M.A. Magnor, and H.P. Seidel, "Free-viewpoint video of human actors," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, July 2003, pp. 569–577.

[59] G. Cheung, S. Baker, J. Hodgins, and T. Kanade, "Markerless human motion transfer," in *Proc. 2nd Int. Symp. 3D Data Processing Visualization Transmission*, Thessaloniki, Greece, Sept. 2004, pp. 373–378.

[60] G. Cheung, S. Baker, and T. Kanade, "Visual hull alignment and refinement across time: A 3D reconstruction algorithm combining shape-from-silhouette with stereo," in *Proc. IEEE Conf. CVPR*, vol. 2, June 2003, pp. 375–382.

[61] M. Magnor, *Video-Based Rendering.* Wellesley, MA: A.K. Peters, 2005.

[62] T. Fujii, "A basic study on integrated 3-D visual communication," Ph.D dissertation, The University of Tokyo, 1994 (in Japanese).

[63] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3D visual communication," in *Proc. Picture Coding Symp. '96*, Mar. 1996, pp. 447–451.

[64] M. Bertalmino, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. SIGGRAPH ACM Trans. Graphics*, 2000, pp. 417–424.

[65] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. IEEE Conf. CVPR*, 2003, vol. 2, pp. 721–728. SP