A Review of Image-based Rendering Techniques

Heung-Yeung Shum and Sing Bing Kang
Microsoft Research
{hshum, sbkang}@microsoft.com

Abstract

In this paper, we survey the techniques for image-based rendering. Unlike traditional 3D computer graphics in which 3D geometry of the scene is known, image-based rendering techniques render novel views directly from input images. Previous image-based rendering 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 methods. The continuum between images and geometry used in image-based rendering techniques suggests that image-based rendering with traditional 3D graphics can be united in a joint image and geometry space.

Keywords: Image-based rendering, survey.

1 Introduction

Image-based modeling and rendering techniques have recently received much attention as a powerful alternative to traditional geometry-based techniques for image synthesis. Instead of geometric primitives, a collection of sample images are used to render novel views. Previous work on image-based rendering (IBR) reveals a continuum of image-based representations [22, 15] based on the tradeoff between how many input images are needed and how much is known about the scene geometry.

For didactic purposes, we classify the various rendering techniques (and their associated representations) into three categories, namely rendering with no geometry, rendering with implicit geometry, and rendering with explicit geometry. These categories, depicted in Figure 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 image-based rendering system with depth maps, such as 3D warping [25], and layered-depth images (LDI) [38], LDI tree [5], etc., the model consists of a set of images of a scene and their associated depth maps. 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 3D locations and re-projecting them onto a new picture. For many synthetic environments or objects, it is not difficult to keep the depth information during the rendering process. However, obtaining depth information from real images is hard even for the state-of-art vision algorithms.

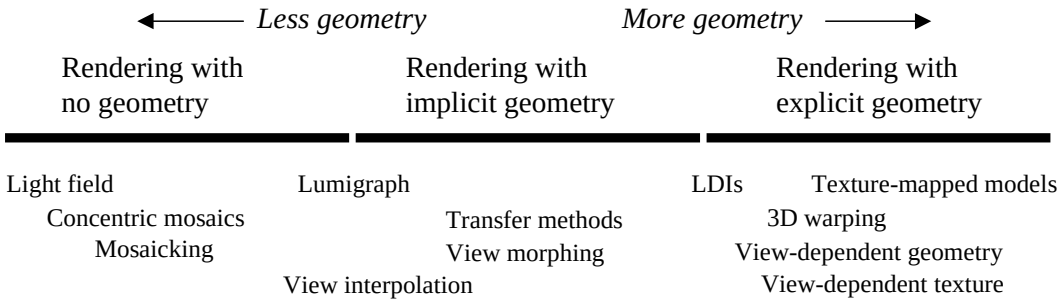


Figure 1: Categories used in this paper, with representative members.

Some image-based rendering systems do not require explicit geometric models. Rather, they require feature (such as points) correspondence between images. For example, view interpolation [6] generates novel views by interpolating optical flow between corresponding points. On the other hand, view morphing [37] generates 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 [23] generates a new image of a scene by appropriately filtering and interpolating a pre-acquired set of samples. Lumigraph [12] is similar to light field rendering but it applies approximated geometry to compensate for non-uniform sampling in order to improve rendering performance. Unlike light field and lumigraph where cameras are placed on a two-dimensional grid, the concentric mosaics representation [39] reduces the amount of data by capturing a sequence of images along a circle path. Light field rendering, however, has 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.

How many images are necessary for anti-aliased rendering? This sampling question needs to be answered by every image-based rendering system. Sampling analysis in image-based rendering, however, is a difficult problem because it involves the unraveling relationship among three elements: the depth and texture information of the scene, the number of sample images, and the rendering resolution. The answer to the sampling analysis provides design principles for image-based rendering systems, in terms of trade-off between the images and the geometry information needed.

The remainder of this paper is organized as follows. Three categories of image-based rendering systems, with no, implicit, and explicit geometric information respectively, are presented in Sections 2, 3, and 4. The issue of trade-off between images and geometric information needed for image-based rendering is discussed in Section 5. We also discuss compact representation and efficient rendering techniques in Section 6, and provide concluding remarks in Section 7.

2 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.

2.1 Plenoptic modeling

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

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

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 [44] that light source directions can be incorporated into the plenoptic function for illumination control. By dropping out two variables, time t (therefore static environment) and light wavelength λ (hence fixed lighting condition), McMillan and Bishop [28] introduced plenoptic modeling with the 5D complete plenoptic function,

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

The simplest plenoptic function is a 2D panorama (cylindrical [7] or spherical [43]) when the viewpoint is fixed,

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

And a regular image (with a limited field of view) can be regarded as an incomplete plenoptic sample at a fixed viewpoint.

Image-based rendering, therefore, becomes one of constructing a continuous representation of the plenoptic function from observed discrete samples (complete or incomplete). How to sample the plenoptic function and how to reconstruct a continuous function from discrete samples are important research topics. For example, the samples used in [28] are cylindrical panoramas. Disparity of each pixel in stereo pairs of cylindrical panoramas is computed and used for generating new plenoptic function samples. Similar work on regular stereo pairs can be found in [20].

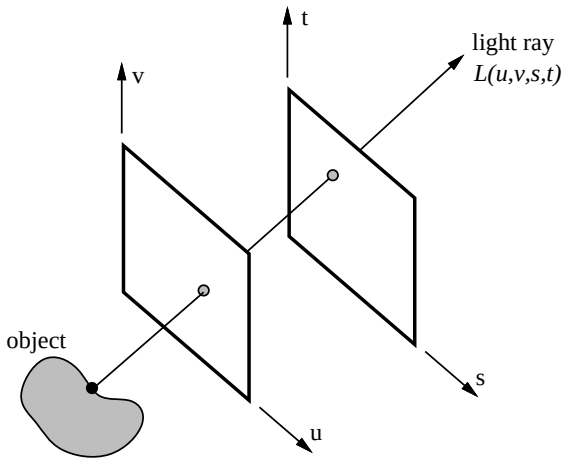


Figure 2: Representation of a light field.

Dimension	Year	Viewing space	Name
7	1991	free	Plenoptic function
5	1995	free	Plenoptic modeling
4	1996	bounding box	Lightfield/Lumigraph
3	1999	bounding plane	Concentric mosaics
2	1994	fixed point	Cylindrical/Spherical panorama

Figure 3: A taxonomy of plenoptic functions.

2.2 Light field and lumigraph

It was observed in both light-field rendering [23] and lumigraph [12] systems that as long as we stay outside the convex hull (or simply a bounding box) of an object,[1] we can simplify the 5D complete plenoptic function to a 4D lightfield plenoptic function,

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

where (u, v) and (s, t) parameterize two parallel planes of the bounding box, as shown in Figure 2. To have a complete description of the plenoptic function for the bounding box, six sets of such two-planes are needed. More restricted versions of lumigraph have also been developed by Sloan *et al*. [41] and Katayama *et al* [19]. The camera motion is restricted to a straight line.

In the light field system, a capturing rig is designed to obtain uniformly sampled images. To reduce aliasing effect, the light field is pre-filtered before rendering. A vector quantization scheme is used to reduce the amount of data used in light field rendering, yet 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 is therefore required. Geometric information is used to guide the choices of the basis functions. Because of the use of geometric information, sampling density can be reduced.

2.3 Concentric mosaics

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 5D plenoptic function. As soon as we stay in a convex hull (or conversely viewing from a convex hull) free of occluders, we have a 4D lightfield. If we do not move at all, we have a 2D panorama. An interesting 3D parameterization of the plenoptic function, called Concentric Mosaics [39] , is proposed by Shum and He where the camera motion is constrained along concentric circles on a plane. A taxonomy of plenoptic functions is shown in Figure 3.

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

Figure 4: Rendering a lobby: rebinned concentric mosaic (a) at the rotation center; (b) at the outermost circle; (c) at the outermost circle but looking at the opposite direction of (b); (d) parallax change between the plant and the poster.

By constraining camera motion to planar concentric circles, concentric mosaics can be created by compositing slit images taken at different locations of each circle. Concentric mosaics index all input image rays naturally in 3 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. Concentric mosaics have good space and computational efficiency. Compared with a lightfield or lumigraph, concentric mosaics have much smaller file size because only a 3D plenoptic function is constructed.

Most importantly, concentric mosaics are very easy to capture. Capturing concentric mosaics is as easy as capturing a traditional panorama except that concentric mosaics require more images. By simply spinning an off-centered camera on a rotary table, we can construct concentric mosaics for a real scene in 10 minutes. Like panoramas, concentric mosaics do not require the difficult modeling process of recovering geometric and photometric scene models. Yet concentric mosaics provide a much richer user experience by allowing the user to move freely in a circular region and observe significant parallax and lighting changes. The ease of capturing makes concentric mosaics very attractive and useful for many virtual reality applications.

Rendering of a lobby scene from captured concentric mosaics is shown in Figure 4. A rebinned concentric mosaic at the rotation center is shown in Figure 4(a), while two rebinned concentric mosaics taken at exactly opposite directions are shown in Figure 4(b) and (c), respectively. It has also been shown in [32] that such two mosaics taken from a single rotating camera can simulate a stereo panorama. In Figure 4(d), strong parallax can be seen between the plant and the poster in the rendered images.

2.4 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 cylidrical images becomes a simple

Figure 5: Tessellated spherical panorama covering the north pole (constructed from 54 images).

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., [24, 42, 7, 28, 43] among others. When the camera motion is very small, it is possible to put together only small stripes from registered images, i.e., slit images (e.g., [46, 33]), to form a large panoramic mosaic. Capturing panoramas is even easier if omnidirectional cameras (e.g., [30, 29]) or fisheye lens [45] are used.

Szeliski and Shum [43] 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 (e.g., 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 (*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 [40] warps each image based on the results of pairwise local image registrations. Combining both global and local alignment significantly improves the quality of our image mosaics, thereby enabling the creation of full view panoramic mosaics with hand-held cameras.

A tessellated spherical map of the full view panorama is shown in Figure 5. Three panoramic image sequences of a building lobby were taken with the camera on a tripod tilted at three different angles (with 22 images for the middle sequence, 22 images for the upper sequence, and 10 images for the top sequence). The camera motion covers more than two thirds of the viewing sphere, including the top.

3 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; 3D 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.

3.1 View interpolation

From two input images, given dense optical flow between them, Chen and Williams' view interpolation method [6] can reconstruct arbitrary viewpoints. 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 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 degrees.

Establishing flow fields for view interpolation can be difficult, in particular 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.

3.2 View morphing

From two input images, Seitz and Dyer's view morphing technique [37] 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 [36] extends this framework to camera motion in a plane. He assumes, however, that the camera parameters are known.

3.3 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 [21] 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 [11] 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 trifocal 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 computation. This idea has been used to generate novel views from either two or three reference images [2].

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 Figure 6.

4 Rendering with explicit geometry

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

4.1 3D warping

When the depth information is available for every point in one or more images, 3D warping techniques (e.g., [27]) can be used to render nearly viewpoints. An image can be rendered from any nearby point of view by projecting the pixels of the original image to their proper 3D locations and re-projecting them onto the new picture. The most significant problem in 3D warping is how to deal with holes generated in the warped image. Holes are due to the difference of sampling

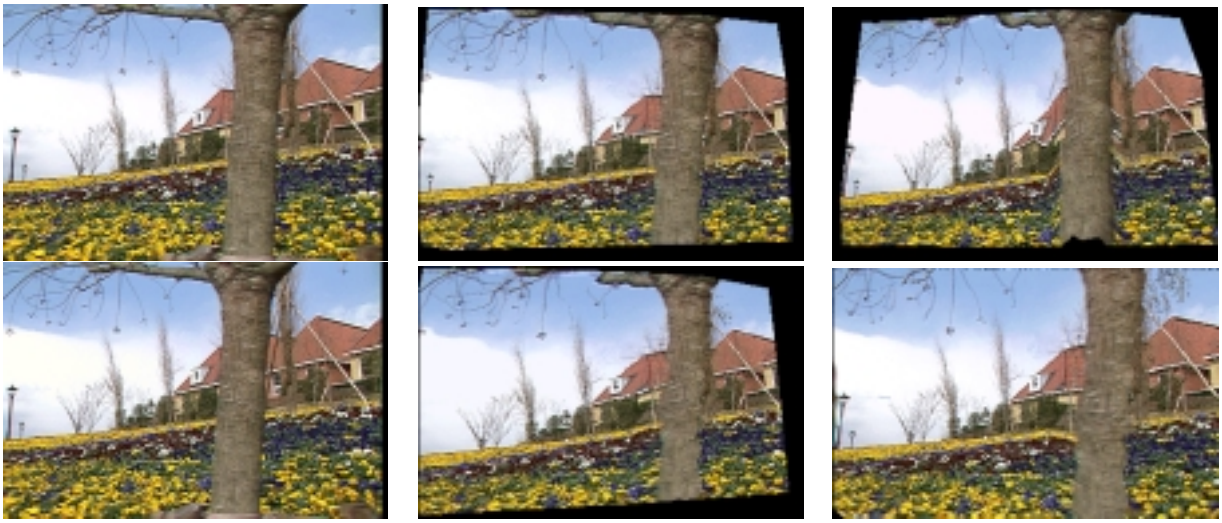


Figure 6: Example of visualizing using the trilinear tensor: The left-most two images are the reference images, with the rest synthesized at arbitrary viewpoints.

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 splat a pixel in the input image to several pixels size in the output image.

4.1.1 Relief texture

To improve the rendering speed of 3D 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 technique proposed by Oliveira and Bishop [31]. Similar factoring approach has been proposed by Szeliski in a two-step algorithm [38] where the depth is first forward warped before the pixel is backward mapped onto the output image.

4.1.2 Multiple-center-of-projection images

The 3D warping techniques can be applied not only to the traditional perspective images, but also multi-perspective images as well. For example, Rademacher and Bishop [35] proposed to render novel views by warping multiple-center-of-projection images, or MCOP images.

4.2 Layered depth images

To deal with the disocclusion artifacts in 3D warping, Shade et al. proposed Layered Depth Image, or LDI [38], to store not only what is visible in the input image, but also what is behind the visible surface. In 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 LDI has the simplicity of warping a single image, it does not consider the issue of sampling rate or how densely should the LDI be. Chang *et al*. [5] 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 with 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.

4.3 View-dependent texture maps

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

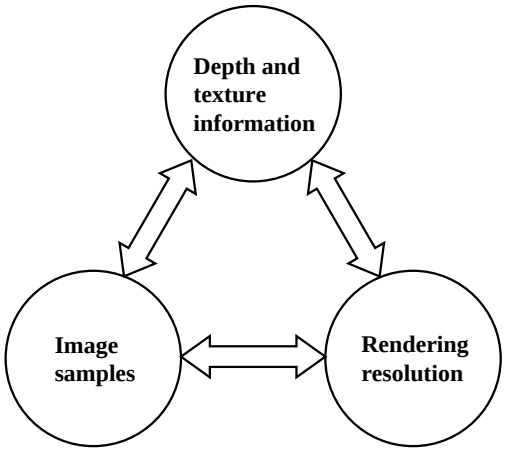


Figure 7: Plenoptic sampling. Quantitative analysis of the relationships among three key elements: depth and texture information, number of input images, and rendering resolution.

To obtain these visual effects of a reconstructed architectural environment, Debevec *et al*. [9] used view-dependent texture mapping to render new views, by warping and compositing several input images of an environment. A three-step view-dependent texture mapping method was also proposed later by Debevec *et al*. [8] to further reduce the computational cost and to have smoother blending. This method employs visibility preprocessing, polygon-view maps, and projective texture mapping.

5 Trade-off 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. So, 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.

5.1 Plenoptic sampling analysis

Many image-based rendering systems, especially light field rendering [23, 12, 39], 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 image-based rendering is a difficult problem because it involves the unraveling relationship among three elements: the depth and texture information of the scene, the number of sample images, and the rendering resolution, as shown in Figure 7.

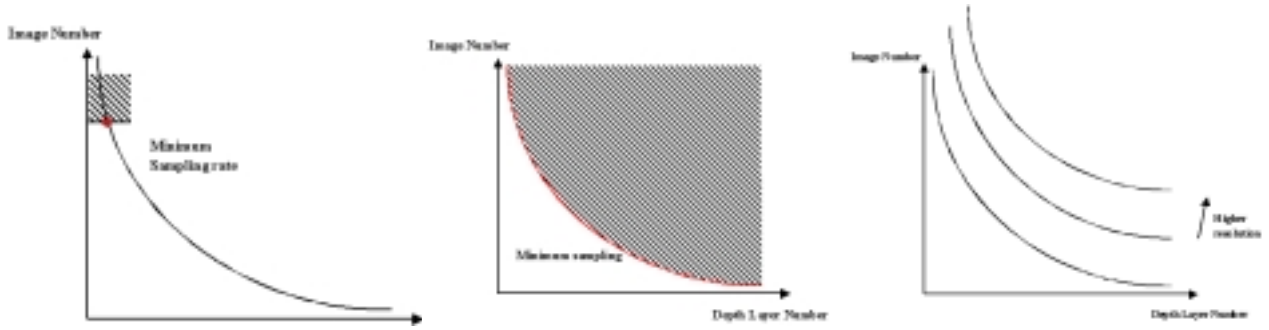
Chai *et al*. [4] recently studied *plenoptic sampling*, or how many images are needed for plenoptic modeling. Plenoptic sampling can be stated as:

> *How many image samples (e.g., from a 4D light field) and how much geometric and textural information are needed to generate a continuous representation of the plenoptic function?*

Specifically, the following two problems are studied under plenoptic sampling:

- Minimum sampling rate for light field rendering;
- 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 $4D$ light field spectral analysis, they only analyze the bounds of the spectral support of the light field signals. A key observation to be presented 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



(a) (b) (c)

Figure 8: 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.

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 anti-aliased 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 serves as the design principles for IBR systems. Furthermore, it bridges the gap between image-based rendering and traditional geometry-based rendering. Minimum sampling rate and minimum sampling curves are illustrated in Figure 8.

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. The following subsections describe some of these techniques.

5.2 Multiple viewpoint rendering

An approach that bridges the notions of the light field or lumigraph and 3D scene geometry is what Halle calls *multiple viewpoint rendering* [13]. Assuming that the 3D scene is completely known, multiple viewpoints can be precomputed at known camera viewpoints and preprocessed to take advantage of the perspective coherence (i.e., the similarity of images of a static scene at different viewpoints). The tool used for such a purpose is the EPI [3] representation. In this case, the EPI is a slice of spatio-perspective space cut parallel to the direction of camera motion.

5.3 View-dependent geometry

Another interesting representation that trades off geometry and images is view-dependent geometry, first used in the context of 3D cartoons [34]. We can potentially extend this idea to represent real or synthetically-generated scenes more compactly. As described in [18], 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 [34], 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 area should be a fertile one for future investigation with potentially significant payoffs.

5.4 Dynamically reparameterized light field

Recently, Isaksen *et al*. [14] 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.

6 Discussion

Image-based rendering is an area that straddles both computer vision and computer graphics. The continuum between images and geometry is evident from the image-based rendering techniques reviewed in this article. However, the emphasis of this article 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.

In this review, image-based 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 image-based rendering techniques have also been proposed by others, such as on the nature of the pixel indexing scheme [15].

There remain many challenges in image-based rendering, including:

1. Efficient representation

   What is very interesting is the trade-off between geometry and images needed to use for anti-aliased image-based rendering. Many image-based rendering systems have made their choices on whether accurate geometry and how much geometric information should be used. Plenoptic sampling provides a theoretical foundation for designing image-based rendering systems.

   Both light field rendering and lumigraph avoided the feature correspondence problem by collecting many light rays with known camera poses. With the help of a specially designed rig, they are capable of generating light fields for objects sitting on a rotary table. Camera calibration with marked features was used in the lumigraph system to recover camera poses. Unfortunately, the resulting light field/lumigraph database is very large even for a small object (therefore small convex hull). Walkthrough of a real scene using lightfield has not yet been fully demonstrated.

   Because of the large amount of data used to represent the 4D function, lightfield compression is necessary. It also makes sense to compress it because of the spatial coherency among all captured images.

2. Rendering performance

   How would one implement the "perfect" rendering engine? One possible would be to utilize 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 [26]. PixelFlow [10] is a high-speed image generation architecture that is based on the techniques of object-parallelism and image composition.

3. 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., [24, 42, 7, 28, 43]). When the camera motion is very small, it is possible to put together only small stripes from registered images, i.e., slit images (e.g., [46, 33]), to form a large panoramic mosaic. Capturing panoramas is even easier if omnidirectional cameras (e.g., [30, 29]) or fisheye lens [45] are used.

   It is, however, very difficult to construct a continuous 5D[2] complete plenoptic function [28, 17] because it requires solving the difficult feature correspondence problem. The QuickTime VR system [7] simply enables the user to discretize the 3D space into a number of sample nodes. The user can only jump between samples.

Image-based rendering can have many interesting applications. Two scenarios, in particular, are worth pursuing:

[2]To date no one has yet shown collection of 7D complete plenoptic functions, even though wandering in a dynamic environment with varying lighting condition is a very interesting problem.

- Large environments.

  Many successful techniques, e.g., light field, concentric mosaics, have restrictions on how much a user can change his viewpoint. For large environment, QuickTime VR is still the most popular system despite the visual discomfort caused by hot-spot jumping between panoramas. This can be alleviated by having multiple panoramic clusters and enabling single DOF transitioning between these clusters [16], but 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.

- Dynamic environments.

  Until now, most of image-based rendering systems have been focused on static environments. With the development of panoramic video systems, it is conceivable that image-based rendering 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).

7 Concluding remarks

We have surveyed recent developments in the area of image-based rendering, 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 3D 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.

Demands on realistic rendering, compactness of representation, speed of rendering, and costs and limitations of computer vision reconstruction techniques force the practical representation to be fall somewhere between the two extremes. It is clear from our survey that IBR and the traditional 3D model-based rendering techniques have complimentary characteristics that can be capitalized. As a result, we believe that it is important that future rendering hardware be customized to handle both the traditional 3D 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*, pages 3–20. MIT Press, Cambridge, MA, 1991.

[2] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, San Juan, Puerto Rico, June 1997.

[3] H. H. Baker and R. C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3(1):33–49, 1989.

[4] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proc. SIGGRAPH*, 2000.

[5] C. Chang, G. Bishop, and A. Lastra. LDI tree: A hierarchical representation for image-based rendering. *Computer Graphics (SIGGRAPH'99)*, pages 291–298, August 1999.

[6] S. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics (SIGGRAPH'93)*, pages 279–288, August 1993.

[7] S. E. Chen. QuickTime VR – an image-based approach to virtual environment navigation. *Computer Graphics (SIGGRAPH'95)*, pages 29–38, August 1995.

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

[9] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics (SIGGRAPH'96)*, pages 11–20, August 1996.

[10] J. Eyles, S. Molnar, J. Poulton, T. Greer, A. Lastra, N. England, and L. Westover. Pixelflow: The realization. In *Siggraph/Eurographics Workshop on Graphics Hardware*, Los Angeles, CA, Sug. 1997.

[11] O. Faugeras. *Three-dimensional computer vision: A geometric viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[12] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Computer Graphics Proceedings, Annual Conference Series*, pages 43–54, Proc. SIGGRAPH'96 (New Orleans), August 1996. ACM SIGGRAPH.

[13] M. Halle. Multiple viewpoint rendering. In *Computer Graphics Proceedings, Annual Conference Series*, pages 243–254, Proc. SIGGRAPH'98 (Orlando), July 1998. ACM SIGGRAPH.

[14] A. Isaksen, L. McMillan, and S. Gortler. Dynamically reparameterized light fields. Technical report, Technical Report MIT-LCS-TR-778, May 1999.

[15] S. B. Kang. A survey of image-based rendering techniques. In *VideoMetrics, SPIE Vol. 3641*, pages 2–16, 1999.

[16] S. B. Kang and P. K. Desikan. Virtual navigation of complex scenes using clusters of cylindrical panoramic images. In *Graphics Interface*, pages 223–232, Vancouver, Canada, June 1998.
[17] S. B. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 364–370, San Francisco, California, June 1996.
[18] S. B. Kang, R. Szeliski, and P. Anandan. The geometry-image representation tradeoff for rendering. In *International Conference on Image Processing*, Vancouver, Canada, Sept. 2000.
[19] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura. A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images. In S. Fisher, J. Merritt, and B. Bolas, editors, *Stereoscopic Displays and Virtual Reality Systems II, Proc. SPIE*, volume 2409, pages 11–20, 1995.
[20] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images and fundamental matrices. Technical Report 2205, INRIA-Sophia Antipolis, February 1994.
[21] S. Laveau and O. D. Faugeras. 3-d scene representation as a collection of images. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, volume A, pages 689–691, Jerusalem, Israel, October 1994. IEEE Computer Society Press.
[22] J. Lengyel. The convergence of graphics and vision. Technical report, IEEE Computer, July 1998.
[23] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics Proceedings, Annual Conference Series*, pages 31–42, Proc. SIGGRAPH'96 (New Orleans), August 1996. ACM SIGGRAPH.
[24] S. Mann and R. W. Picard. Virtual bellows: Constructing high-quality images from video. In *First IEEE International Conference on Image Processing (ICIP-94)*, volume I, pages 363–367, Austin, Texas, November 1994.
[25] W. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. In *Proc. Symposium on I3D Graphics*, pages 7–16, 1997.
[26] D. K. McAllister, L. Nyland, V. Popescu, A. Lastra, and C. McCue. Real-time rendering of real world environments. In *Eurographics Workshop on Rendering*, Granada, Spain, June 1999.
[27] L. McMillan. An image-based approach to three-dimensional computer graphics. Technical report, Ph.D. Dissertation, UNC Computer Science TR97-013, 1999.
[28] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39–46, August 1995.
[29] V. S. Nalwa. A true omnidirecional viewer. Technical report, Bell Laboratories, Holmdel, NJ, USA, February 1996.
[30] S. Nayar. Catadioptric omnidirectional camera. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 482–488, San Juan, Puerto Rico, June 1997.
[31] M. Oliveira and G. Bishop. Relief textures. Technical report, UNC Computer Science TR99-015, March 1999.
[32] S. Peleg and M. Ben-Ezra. Stereo panorama with a single camera. In *Proc. Computer Vision and Pattern Recognition Conf.*, 1999.
[33] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 338–343, San Juan, Puerto Rico, June 1997.
[34] P. Rademacher. View-dependent geometry. *SIGGRAPH*, pages 439–446, Aug. 1999.
[35] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *Computer Graphics Proceedings, Annual Conference Series*, pages 199–206, Proc. SIGGRAPH'98 (Orlando), July 1998. ACM SIGGRAPH.
[36] D. Scharstein. Stereo vision for view synthesis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 852–857, San Francisco, California, June 1996.
[37] S. M. Seitz and C. M. Dyer. View morphing. In *Computer Graphics Proceedings, Annual Conference Series*, pages 21–30, Proc. SIGGRAPH'96 (New Orleans), August 1996. ACM SIGGRAPH.
[38] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Computer Graphics (SIGGRAPH'98) Proceedings*, pages 231–242, Orlando, July 1998. ACM SIGGRAPH.
[39] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *Proc. SIGGRAPH 99*, pages 299–306, 1999.
[40] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 953–958, Bombay, January 1998.
[41] P. P. Sloan, M. F. Cohen, and S. J. Gortler. Time critical lumigraph rendering. In *Symposium on Interactive 3D Graphics*, pages 17–23, Providence, RI, USA, 1997.
[42] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, March 1996.
[43] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and texture-mapped models. *Computer Graphics (SIGGRAPH'97)*, pages 251–258, August 1997.
[44] T. Wong, P. Heng, S. Or, and W. Ng. Image-based rendering with controllable illumination. In *Proceedings of the 8-th Eurographics Workshop on Rendering*, pages 13–22, St. Etienne, France, June 1997.
[45] Y. Xiong and K. Turkowski. Creating image-based VR using a self-calibrating fisheye lens. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 237–243, San Juan, Puerto Rico, June 1997.
[46] J. Y. Zheng and S. Tsuji. Panoramic representation of scenes for route understanding. In *Proc. of the 10th Int. Conf. Pattern Recognition*, pages 161–167, June 1990.