IMAGE-BASED RENDERING TECHNIQUES

[1]Sahil Mudgal, [2]Rajat Wason, [3]Pooja Nayak

[1,2,3] *Dronacharya College of Engineering, Gurgaon (India)*

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 and rendering with explicit geometry. it is interesting to note that image-based rendering has evolved to be an area that has generated active interest from both computer vision and computer graphics communities.

I 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.

We classify the various rendering technique into three categories, namely rendering with no geometry, rendering with implicit geometry, and rendering with explicit geometry.

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.

II RENDERING WITH NO GEOMETRY

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 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.,

$$P7 = P(V_x,V_y,V_z,\theta,\varphi,\lambda,t). \tag{1}$$

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 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,

$$P5 = P(V_x, V_y, V_z, \theta, \varphi). \quad (2)$$

The simplest plenoptic function is a 2D panorama when the viewpoint is fixed,

$$P2 = P(\theta, \varphi). \quad (3)$$

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 are cylindrical panoramas. Disparity of each pixel in stereo pairs of cylindrical panoramas is computed and used for generating new plenoptic function samples.

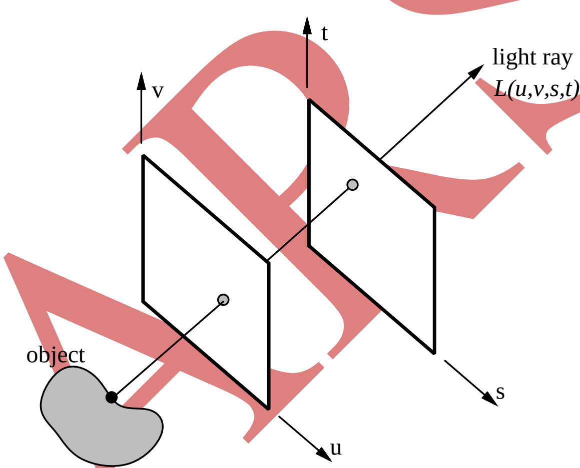

Figure 2: Representation of a light field.

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.

III LIGHT FIELD AND LUMIGRAPH

It was observed in both light-field rendering 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 light field plenoptic function,

$$P4 = P(u,v,s,t), \tag{4}$$

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

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.

3.1 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.

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.

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.

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 translation. For arbitrary camera rotation, one can first register the images by recovering the camera movement, before converting to a final cylindrical/spherical map.

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.

Figure 5: Tessellated Spherical Panorama Covering The North Pole (Constructed From 54 Images).

IV 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.

4.1 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. 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.

4.2 View Morphing

From two input images, Seitz and Dyer's view morphing technique 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.

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.

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.

V 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).

5.1 3D Warping

When the depth information is available for every point in one or more images, 3D warping techniques 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

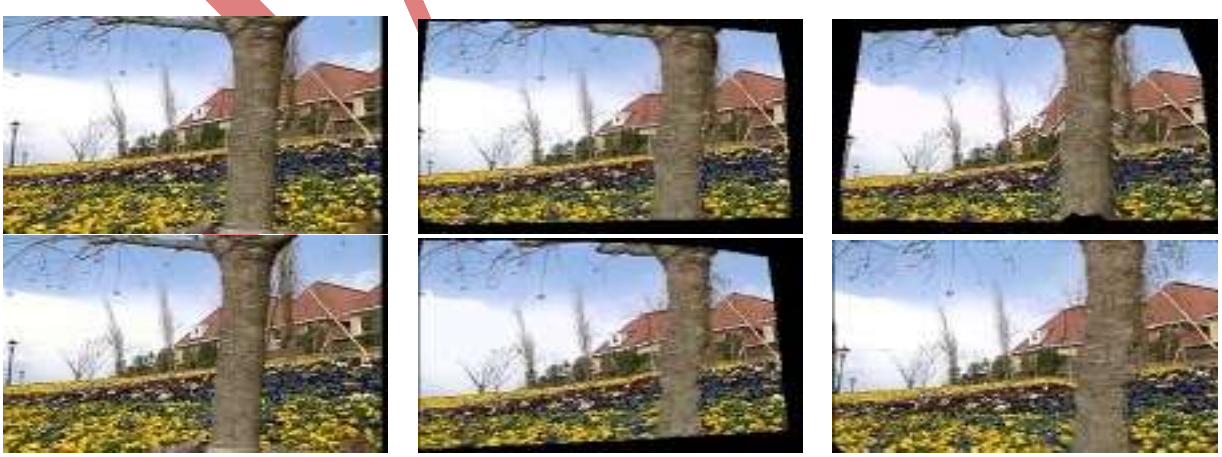

Figure 6: Example of visualizing using the trilinear tensor: The left-most two images are the reference images, with the rest synthesized at arbitrary viewpoints.

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.

5.1.1 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. Similar factoring approach has been proposed by Szeliski in a two-step algorithm where the depth is first forward warped before the pixel is backward mapped onto the output image.

5.1.2 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.

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.

5.2 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

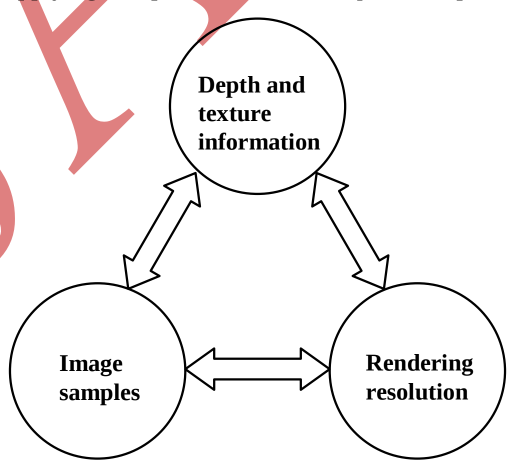

Figure 7: Plenoptic sampling. Quantitative analysis of the relationships among three key elements: depth and texture information, number of input images, and rendering resolution.

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.

To obtain these visual effects of a reconstructed architectural environment 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.

VI 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.

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

6.1 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.

6.2 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 .PixelFlow is a high-speed image generation architecture that is based on the techniques of object-parallelism and image composition.

6.3 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. When the camera motion is very small, it is possible to put together only small stripes from registered images, i.e., slit images to form a large panoramic mosaic. Capturing panoramas is even easier if omnidirectional cameras or fisheye lens are used.

Many successful techniques, e.g., light field, concentric mosaics, have restrictions on how much a user can change his viewpoint. For large environment, QuickTimeVR 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.

Two issues must be studied: sampling (how many images should be captured), and compression (how to reduce data effectively).

VII 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.

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. QuickTimeVR – 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.