3D Rendering - Techniques and Challenges

Vishal Verma[#1], Ekta Walia[*2]

[#]Department of Computer Science, M L N College, Yamuna Nagar, Haryana, INDIA

[1]me_vishaal@hotmail.com

[*]IT Department, Maharishi Markandeshwar University, Mullana, Ambala, INDIA

[2]wekta@yahoo.com

Abstract—**Computer generated images and animations are getting more and more common. They are used in many different contexts such as movies, mobiles, medical visualization, architectural visualization and CAD. Advanced ways of describing surface and light source properties are important to ensure that artists are able to create realistic and stylish looking images. Even when using advanced rendering algorithms such as ray tracing, time required for shading may contribute towards a large part of the image creation time. Therefore both performance and flexibility is important in a rendering system. This paper gives a comparative study of various 3D Rendering techniques and their challenges in a complete and systematic manner.**

I. INTRODUCTION

In the real world, light sources emit photons that normally travel in straight lines until they interact with a surface or a volume. When a photon encounters a surface, it may be absorbed, reflected, or transmitted. Some of these photons may hit the retina of an observer where they are converted into a signal that is then processed by the brain, thus forming an image. Similarly, photons may be caught by the sensor of a camera. In either case, the image is a 2D representation of the environment.

The formation of an image as a result of photons interacting with a 3D environment may be simulated on the computer. The environment is then replaced by a 3D geometric model and the interaction of light with this model is simulated with one of a large number of available algorithms. The process of image synthesis by simulating light behavior is called rendering.

II. GEOMETRY BASED RENDERING ALGORITHMS

In geometry based rendering the illumination of a scene has to be simulated by applying a shading model. As hardware systems provided more and more computing power, those models became more sophisticated.

Gouraud shading [1] is a very simple technique that linearly interpolates color intensities calculated at the vertices of a rendered polygon across the interior of the polygon. Phong introduced a more accurate model [2] that is able to simulate specular highlights. He also proposed to interpolate normals instead of intensities on rendered polygons, thus enabling more accurate evaluations of the actual shading model. Many fast methods [3] [4] have also been proposed that approximate the quality of Phong Shading. All of these models are local in the sense that they fail to model global illumination effects such as reflection. A comparative study of local illumination methods in terms of speed and visual quality is done by Walia and Singh [5].

There is a second class of illumination models that can be applied to polygonal scenes, the so called global illumination models. Unlike the local methods, these methods are able to simulate the inter-reflections between surfaces. Diffuse inter-reflections can be simulated by the radiosity method [6], and specular reflections are handled by recursive ray-tracing techniques [7]. Many more advanced global illumination models [8] are also available. However they are computationally too complex to be used for real time image synthesis on available hardware.

The major problems with Geometry Based Rendering are:

- No Guarantee for the rightness of the models.
- A lot of computation time is needed.
- Rendering algorithms are complex and therefore call for special hardware if interactive speeds are needed.
- Even if special hardware is used, the performance of the system is hard to measure since the rendering time is highly dependent on the scene complexity.

3. IMAGE BASED RENDERING ALGORITHMS

Traditionally, a description of the 3D scene being rendered is provided by a detailed and complex model of the scene. To avoid the expense of modeling a complicated scene, it is sometimes more convenient to photograph a scene from different viewpoints. To create images for novel viewpoints that were not photographed, an interpolation scheme may be applied. Rendering using images as a modeling primitive is called image-based rendering.

Computer graphics researchers recently have turned to image-based rendering due to following reasons:

- Close to photo realism.
- Rendering time is decoupled from scene complexity.
- Images are used as input.
- Exploits coherence.
- Pre-calculation of scene data/ images.

Instead of constructing a scene with millions of polygons, in Image Based Rendering the scene is represented by a collection of photographs along with a greatly simplified geometric model. This simple representation allows traditional light transport simulations to be replaced with basic image-processing routines that combine multiple images together to produce never-before-seen images from new vantage points.

There have been many IBR representations invented in the literature. They basically have following three categories [9]:

- Rendering with no geometry
- Rendering with implicit geometry
- Rendering with explicit geometry

A. Rendering with no geometry

We start with representative techniques for rendering with unknown scene geometry. These techniques typically rely on many input images and also on the characterization of the 7D plenoptic function [10]. Common approaches under this class are

- Light field [11]
- Lumigraph [12]
- Concentric mosaics [13]

The lightfield is the radiance density function describing the flow of energy along all rays in 3D space. Since the description of a ray's position and orientation requires four parameters (e.g., two-dimensional positional information and two-dimensional angular information), the radiance is a 4D function. Image, on the other hand, is only two dimensional and lightfield imagery must therefore be captured and represented in 2D form. A variety of techniques have been developed to transform and capture the 4D radiance in a manner compatible with 2D [11] [12].

In Light Field Rendering [11], the light fields are created from large arrays of both rendered and digitized images. The latter are acquired using a video camera mounted on a computer-controlled gantry. Once a light field has been created, new views may be constructed in real time by extracting 2D slices from the 4D light field of a scene in appropriate directions. The Lumigraph [12] is similar to light field rendering [11]. In addition to features of light field rendering, it also allows us to include any geometric knowledge we may capture to improve rendering performance. Unlike the light field and Lumigraph where cameras are placed on a two-dimensional grid, the 3D Concentric Mosaics [13] representation reduces the amount of data by capturing a sequence of images along a circular path.

Challenges: Because such rendering techniques do not rely on any geometric impostors, they have a tendency to rely on oversampling to counter undesirable aliasing effects in output display. Oversampling means more intensive data acquisition, more storage, and higher redundancy.

B. Rendering with implicit geometry

These techniques for rendering rely on positional correspondences (typically 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. Common approaches under this class are

- View Interpolation [14],
- View Morphing [15],
- Joint View Interpolation [16].

View interpolation [14] uses optical flow (i.e. Relative transforms between cameras) to directly generate intermediate views. But the problem with this method is that the intermediate view may not necessarily be geometrically correct. View morphing [15] is a specialized version of view interpolation, except that the interpolated views are always geometrically correct. The geometric correctness is ensured because of the linear camera motion. Computer vision techniques are usually used to generate such correspondences. M. Lhuillier et al. proposed a new method [16] which automatically interpolating two images and tackle two most difficult problems of morphing due to the lack of depth information: pixel matching and visibility handling.

Challenges: Representations that rely on implicit geometry require accurate image registration for high-quality view synthesis.

C. Rendering with explicit geometry

Representations that do not rely on geometry typically require a lot of images for rendering, and representations that rely on implicit geometry require accurate image registration for high-quality view synthesis. IBR representations that use explicit geometry have generally source descriptions. Such descriptions can be the scene geometry, the texture maps, the surface reflection model etc.

1) Scene Geometry as Depth Maps

These approaches use depth maps as scene representation. Depth maps indicate the per-pixel depth values of the reference views. Such a depth map is easily available for synthetic scenes, and can be obtained for real scenes via a range finder. Common approaches under this class are

- 3D Warping [17]
- Relief Texture [18]
- Layered Depth Images (LDI) [19]
- LDI tree [20]

When the depth information is available for every point in one or more images, 3D warping [17] techniques can be used to render nearly viewpoints. 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 [18]. A similar factoring algorithm was performed for the LDI [19], where the depth map is first warped to the output image with visibility check, and colors are pasted afterwards. LDI store a view of the scene from a single input camera view, but with multiple pixels along each line of sight. Though LDI has the simplicity of warping a single image, it does not consider the issue of sampling rate. Chang *et al.* [20] 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.

2) Scene Geometry as Mesh Model

Mesh model is the most widely used component in model-based rendering. Despite the difficulty to obtain such a model, if it is available in image-based rendering, we should make use of it to improve the rendering quality. Common approaches that use mesh models as scene representation are

- Unstructured Lumigraph [21]
- Spatial-temporal view interpolation [22] [23]

Buchler *et al.* proposed the unstructured Lumigraph rendering [21], where weighted light ray interpolation was used to obtain light rays in the novel view. One concern about the mesh model is that it has a finite resolution. To remove the granular effects in the rendered image due to finite resolution, a model smoothing algorithm was applied during the rendering, which greatly improved the resultant image quality [22] [23].

3) Scene Geometry with Texture Maps

As texture maps are often obtained from real objects, a geometric model with texture mapping can produce very realistic scenes. Common approaches that use texture maps with scene geometry as scene representation are

- View dependent texture map [24] [25]
- Image-based visual hull [26]

Debevec *et al.* proposed view dependent texture mapping (VDTM) [24], in which the reference views are generated from the texture map through a weighting scheme. The weights are determined by the angular deviation from the reference views to the virtual view to be rendered. Later a more efficient implementation of VDTM was proposed in [25], where the per-pixel weight calculation was replaced by a per-polygon search in a pre-computed lookup table. The image-based visual hull (IBVH) algorithm [26] can be considered as another example of VDTM. In IBVH, the scene geometry was reconstructed through an image space visual hull [27] algorithm. Note that VDTM is in fact a special case of the later proposed unstructured Lumigraph rendering [21].

4) Scene Geometry with Reflection Model

Other than the texture map, the appearance of an object is also determined by the interaction of the light sources in the environment and the surface reflection model. Common approaches that use Reflection model with scene geometry as scene representation are

- Reflection space IBR [28]
- Surface light field [29]

In [28], Cabral et al. proposed reflection space image-based rendering. Reflection space IBR records the total reflected radiance for each possible surface direction. The above method assumes that if two surface points share the same surface direction, they have the same reflection pattern. This might not be true due to multiple reasons such as inter reflections. Wood et al. proposed improved surface light field [29] which also considers the concept of inter reflections.

Challenges: Obtaining source descriptions from real images is hard even with state-of-art vision algorithms.

D. Sampling and Compression

Once the IBR representation of the scene has been determined, one may further reduce the data size through sampling and compression [9] [30]. The sampling analysis can tell the minimum number of images / light rays that is necessary to render the scene at a satisfactory quality. Compression, on the other hand, can further remove the redundancy inside and between the captured images. Due to the high redundancy in many IBR representations, an efficient IBR compression algorithm can easily reduce the data size by tens or hundreds of times.

IV. IMPORTANCE OF IBR

Traditionally, virtual reality environments have been generated by rendering a geometrical model of the environment using techniques such as polygon rendering or ray-tracing. In order to get a convincing image, both the geometrical model and the rendering algorithms have to be complex and therefore call for special hardware if interactive speeds are needed. Even if special hardware is used, the performance of the system is hard to measure since the rendering time is highly dependent on the scene complexity. In simulators, for example, it is not acceptable to have a low frame rate when the view is complex, since it introduces a time lag in the control loop where persons play an important role. Creating systems able to perform well in worst cases is expensive.

Image-based rendering is a better approach as by sampling the light distribution in the scene to be rendered, typically by taking photographs from different positions and in different directions, it is able to present new views of the scene. The algorithms used are relatively fast and several commercial implementations for use on ordinary personal computers exist, of which the QuickTime VR system [31] from Apple Computer is the best known today. Image-based systems have a fixed rendering time, independent of the scene complexity, which simplify system construction.

In addition to the above, many other forces *have* contributed to the recent research work in the area of image-based rendering. Among these are:

- Our ability to render models has begun to outpace our capacity to create high-quality models.
- The limited computational capabilities and lack of powerful 3D graphics hardware support in mobile/ hand held devices.
- The availability of inexpensive digital image acquisition hardware.
- Recent trends in computer graphics accelerator architectures.

Image Based Rendering approach to visualize real-world or synthetic scenes on mobile devices, has been proposed in [32] [33] [34]. For the mobile devices that are equipped with wireless network, client-server framework with IBR can be utilized to increase the performance.

V. RESULTS AND CONCLUSION

We have surveyed the rendering techniques which have two main classifications: Geometry Based Rendering and Image Based Rendering.

In geometry based rendering techniques, we found that the shading quality obtained from Phong shading Model is better as compared to Gouraud shading Model but it is computationally more expensive. A comparison is shown below in Fig 1.

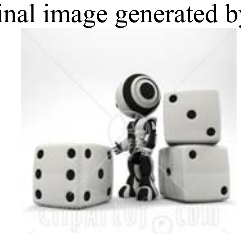
(a) Image Generation using Gouraud Shading

(b) Image Generation using Phong Shading

Fig. 1

In addition these models do not simulate the global illumination effects. The global illumination models are computationally too complex to be used for real time image synthesis on available hardware.

Alternative approach is Image Based Rendering. We found that all the IBR representations originate from the 7D plenoptic function [10], which describes the appearance of the world. As the 7D plenoptic function has too much data to handle, various approaches have been proposed to reduce the data size while still giving the viewer a good browsing experience. Such techniques are widely adopted in the real world. For example, Fig. 2(a) shows the original image. Fig. 2(b) and Fig. 2(c) shows the two views of original image generated by 3D warping.



(a) Original Image

(b) Generated view through 3D Warping

(c) Generated view through 3D Warping

Fig. 2

As compared to Geometry Based Rendering, the rendering process in IBR is usually very fast and can be implemented with software. However, hardware acceleration will be definitely helpful for future high-resolution IBR rendering. As most operations in IBR rendering are simple mathematical operations such as linear interpolation, and most IBR rendering process can be performed in parallel, we expect that such hardware is not difficult to develop and can dramatically increase the rendering speed.

No matter how much the storage and memory increase in the future, sampling and compression are always useful to keep the IBR data at a manageable size. The work on sampling and compression, however, has just started. There are still many problems which remain unsolved, such as the sampling rate when certain source description is available. A high compression ratio in IBR seems to rely heavily on how good the images can be predicted, which depends on, e.g., how good a certain source description can be reconstructed. Joint work between the signal processing community and the computer vision community is highly expected in this regard.

REFERENCES

[1] H. Gouraud. "Continuous Shading of Curved Surfaces." IEEE Transactions on Computers, C-20(6):623–629, June 1971.

[2] B.T. Phong. "Illumination for Computer Generated Pictures." Communications of the ACM, 18(6):311–317, June 1975.

[3] Chandan Singh, Ekta Walia, "Shading by Fast Bi-Quadratic Normal Vector Interpolation", ICGST International Journal on Graphics, Vision and Image Processing, Vol. 5. Issue 9, pp. 49-54, 2005.

[4] Chandan Singh, Ekta Walia, "Fast Hybrid Shading: An Application of Finite Element Methods in 3D Rendering", International Journal of Image and Graphics, Vol. 5, No. 4, pp-789-810, 2005.

[5] Ekta Walia, Chandan Singh. "An Analysis of Linear and Non-Linear Interpolation Techniques for Three-Dimensional Rendering." IEEE Proceedings of the Geometric Modeling and Imaging— New Trends (GMAI'06). 2006.

[6] Cohen, Greenberg, Immel, and Brock. "An efficient radiosity approach for realistic image synthesis." IEEE Computer Graphics and Applications 6, pages 23–35, 1986.

[7] Turner Whitted. "An improved illumination model for shaded display." Communications of the ACM, 23(6):343–349, June 1980.

[8] P. Dutre, P. Bekaert, and K. Bala. "Advanced Global Illumination." AK Peters, Natick, MA, 2003.

[9] H.Y. Shum, S.B. Kang, and S.C. Chan. "Survey of image-based representations and compression techniques." IEEE Trans. On Circuits and Systems for Video Technology, vol. 13, no. 11, pp. 1020-1037. Nov. 2003.

[10] E. H. Adelson, and J. R. Bergen, "The plenoptic function and the elements of early vision", Computational Models of Visual Processing, Edited by Michael Landy and J. Anthony Movshon. The MIT Press, Cambridge, Mass. Chapter 1. 1991.

[11] M. Levoy and P. Hanrahan, "Light field rendering", Computer Graphics (SIGGRAPH'96), pp. 31-42. August 1996.

[12] S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, "The Lumigraph", Computer Graphics (SIGGRAPH'96), pp. 43-54. August 1996.

[13] H.Y. Shum and L.W. He, "Rendering with concentric mosaics", Computer Graphics (SIGGRAPH'99), pp.299-306. August 1999.

[14] S. E. Chen and L. Williams, "View interpolation for image synthesis", Computer Graphics (SIGGRAPH'93), pp. 279-288. August 1993.

[15] S. M. Seitz and C. M. Dyer, "View morphing", Computer Graphics (SIGGRAPH'96), pp. 21-30. August 1996.

[16] M. Lhuillier and L. Quan. "Image interpolation by joint view triangulation." In IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 139-145, Fort Collins, CO, June 1999.

[17] L. McMillan, "An Image-Based Approach to Three-Dimensional Computer Graphics", Ph.D. Thesis, Department of Computer Science, University of North Carolina at Chapel Hill, 1997.

[18] M. Oliveira and G. Bishop. "Relief textures". Technical report, UNC Computer Science TR99-015, March 1999.

[19] J. Shade, S. Gortler, L. W. He and R. Szeliski, "Layered depth images", In Computer Graphics (SIGGRAPH), pages 231-242, August 1998.

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

[21] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured Lumigraph Rendering", Computer Graphics (SIG-GRAPH'01), pp 425-432. August 2001.

[22] S. Vedula, S. Baker and T. Kanade, "Spatio-Temporal View Interpolation", Proc. of the 13th ACM Eurographics Workshop on Rendering, June, 2002.

[23] S. Vedula, S. Baker, and T. Kanade. "Image-based spatio-temporal modeling and view interpolation of dynamic events." ACM Transaction on Graphics, 24(2):240-261, April 2005.

[24] P. Debevec, C. J. Taylor and J. Malik, "Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach", Computer Graphics (SIGGRAPH'96), pp. 11-20. August 1996.

[25] 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.

[26] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, "Image-based Visual Hulls", Computer Graphics (SIGGRAPH'00), pp. 369-374. July 2000.

[27] A. Laurentini, "The Visual Hull Concept for Silhouette Based Image Understanding." *IEEE PAMI*, Vol. 16, No. 2, pp. 150-162, 1994.

[28] B. Cabral, M. Olano, P. Nemec, "Reflection space image based rendering", Computer Graphics (SIGGRAPH'99), pp. 165-171. August 1999.

[29] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin and W. Stuetzle, "Surface light fields for 3D photography", Computer Graphics (SIGGRAPH'00), pp. 287-296. July 2000.

[30] Cha Zhang. "On Sampling of Image Based Rendering Data". Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University Pittsburgh, PA. June 2004.

[31] S. E. Chen, "QuickTime VR – An Image-Based Approach to Virtual Environment Navigation", Computer Graphics (SIGGRAPH'95), pp. 29-38, August 1995.

[32] Chang C, Ger S. "Enhancing 3D graphics on mobile devices by image-based rendering". Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing, Taiwan, pp. 1105–1111. 2002.

[33] Boukerche A, Jing F, de Araujo R B. "A 3D image-based rendering technique for mobile handheld devices". Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), Buffalo, pp. 325–331. 2006.

[34] E. Pilav, B. Brkic. "Real-time Image Based Rendering Using Limited Resources". CESCG 2008.