The geometry-image representation tradeoff for rendering

Sing Bing Kang, Richard Szeliski, and P. Anandan
Vision Technology Group
Microsoft Research
Microsoft Corporation
Redmond, WA 98052, USA

(Submitted to ICIP2000)

Abstract

It is generally recognized that 3-D models are compact representations for rendering. While pure image-based rendering techniques are capable of producing highly photorealistic outputs, the size of the input "model" is usually very large. The important issues in trading off geometry versus images include compactness of representation, photorealism of reconstructed views, and speed of rendering. In this paper, we describe our past work in modeling and rendering, and articulate lessons learnt. We then delineate our vision of an ideal rendering system that is capable of using spatially varying representations to produce an optimal mix of high compression and realistic rendering.

Keywords*: 3-D modeling, image-based rendering, geometry versus images.*

1 Introduction

Most commercial graphics systems and research focus on explicit 3-D modeling. This is a very traditional area, and the 3-D modeling and rendering issues have been mostly dealt with successfully. It is generally true that 3-D models are compact representations for rendering, especially when fast specialized hardware graphics accelerators are readily available. Better photorealism is possible with the more sophisticated 3-D modeling systems such as 3D StudioMax, though at a loss in rendering speed.

Pure image-based rendering techniques (relevant surveys include [8, 12]) index images directly, and their outputs can be highly photorealistic. The important issues in trading off the use of geometric representations versus images include compactness of representation, photorealism of reconstructed views, speed of rendering, and cost.

In this paper, we review our work in extracting geometry and image-based representations from images, and summarize important lessons from our work. We also describe our concept of an ideal rendering system that is capable of using spatially varying representations to produce an optimal mix of high compression and realistic rendering.

2 Review of relevant work

There has been a lot of prior work on object and scene representation for visualization. This work can be classified according to how geometry or image intensive the underlying representation is. On the image end of the spectrum, there are systems with constrained motion (either pure rotation or translation), such as mosaics from many images [27], stitched fisheye images [30], or view morphing [21]. While this kind of representation is very compact, the degrees of freedom in virtual navigation are very limited. More recent developments in the form of light-field rendering [13] and its model-assisted cousin the Lumigraph [7] provide an enhanced range of virtual motion, but at the expense of a large input database. The concentric mosaic representation [23] is more compact, but loses a degree of freedom in motion and depth effect. The important issue of sampling and artifacts in the context of light field rendering has been examined by several researchers [2, 3, 14].

Work that is primarily image-based with some implicit geometry includes interpolation of color images with single [16] or multiple [4] depth maps, multiple source rendering [9], multiple centers of projection [18], and layers [22, 1]. The compactness of these representations range from high (for flat layers) to low (with explicitly assigned depth at every pixel).

There is plenty of work on 3-D modeling, but we mention only a few as (somewhat biased) representatives in

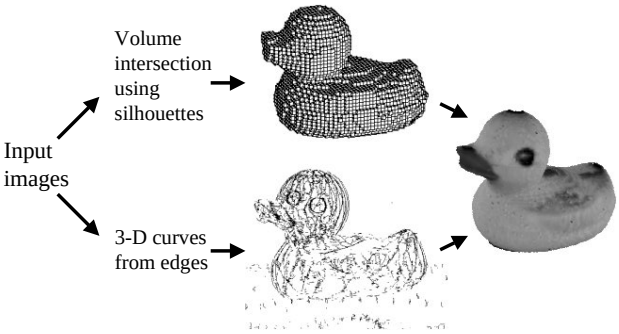

Figure 1: Two possible ways of modeling an object: using silhouettes, and using edges.

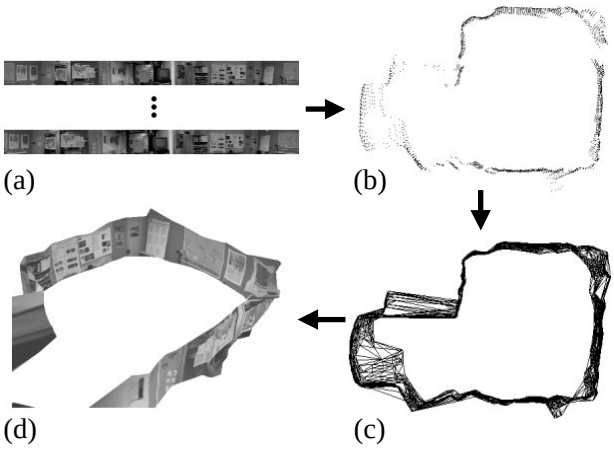

Figure 2: 3-D modeling from multiple cylindrical panoramic images: (a) sequence of input panoramic images; (b) top view of recovered 3-D points; (c) top view of 3-D mesh; and (d) oblique view of texture-mapped model.

this area. Automatic modeling from images can take the form of reconstruction of an octree model from silhouettes [26], 3-D curves from contours [29], or simply from 3-D points using panoramic images [10]. Examplese of 3-D construction systems that require some degree of interactivity include Facade, which uses manual placement of 3-D primitives [5], and Shum *et al.*'s system that makes use of 2-D lines drawn on panoramic mosaics [24].

3 Texture-mapped geometry

Texture-mapped geometric models are widely used in computer graphics. Methods for creating such models include using a CAD modeler, using a 3-D digitizer, and applying computer vision techniques to images of real objects or scenes. For the last category, examples include octree modeling from volume intersection using silhouettes [26], shape from contours [29] (Figure 1), and modeling using panoramic images (Figure 2). *[You already mentioned octrees and contours in Section 2. We probably want to eliminate one of these duplicate references. – Rick]*

Unfortunately, computer vision techniques are generally not robust enough to work under all conditions. 3-D recovery is not perfect (see Figure 2(d), for example), and stereo algorithms do not work for textureless regions without prior structural knowledge. In addition, it is very difficult to capture complex visual effects such as highlights, reflections, and transparency using a texture-mapped 3-D model.

Mesh simplification is often used to reduce the complexity of model that affects rendering time. However, this generally results in loss of data; simplification leads to "equal-opportunity" smoothing, since it is difficult to automatically distinguish noise from data.

[Anandan, Rick: any other failure modes/comments you can think of? I can't think of any – Rick]

4 Image-based representations

A large number of image-based rendering systems have been developed in the last few years (see Section 2). Here, we discuss two representative image-based representations, namely sprites with depth and the Lumigraph.

The *sprites with depth* representation [1, 22] (a.k.a. 3-D layered motion models) consists of a collection of overlapping, quasi-planar surfaces with local parallax [11, 20] used to describe their more detailed shape. Each sprite consists of a *matted* color image (i.e., a color image with an alpha channel used to decribe transparency), a plane equation describing its rough orientation, and a depth map encoding the local out-of-plane shape variation. (Alternatively, the parallax could also be represented using a triangular mesh.)

Sprites with depth naturally capture the occlusion effects that occur in complex scenes, e.g., partial visibility, the mixing of foreground and background colors at occluding contours, and transparent mixing of light at reflections [28]. Sprites with depth can also be efficiently rendered using a combination of forward and inverse warping techniques [22]. Figure 3 shows the sprites extracted from a scene simultaneously photographed from a number of different viewpoints. Using a single depth map would be inadequate to capture all of the surfaces in this scene, while

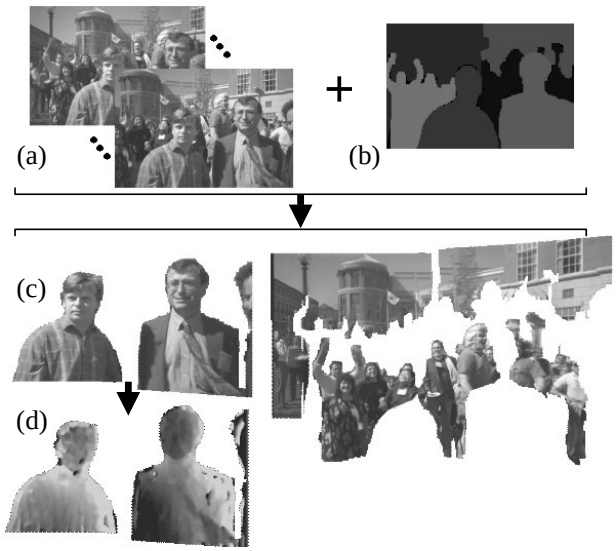

Figure 3: Layered extraction from multiple images: (a) input sequence; (b) manually segmented layers; (c) mosaicked layers; and (d) relative depth (parallax) maps.

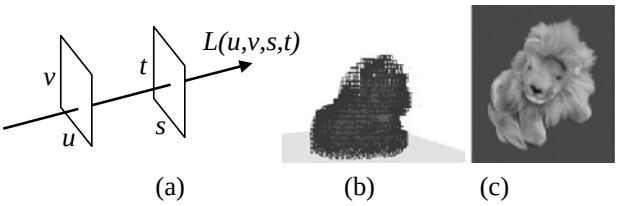

Figure 4: Lumigraph example: (a) sampling space; (b) octree of toy lion (from image sequence); and (c) a novel view (note the detailed fur, which would be very hard to produce with a texture-mapped 3-D model).

building full 3-D models would not be possible because of the limited range of viewpoints.

A variant on sprites with depth is the *Layered Depth Image* (LDI) representation [22], where multiple color/depth values can be stored at each location in a 2-D array. This representation is easier to compute than a layered sprite representation, since it does not require any segmentation of the scene into parts. On the other hand, the rendering algorithm, which uses forward mapping (*splatting*) cannot take advantage of the contiguity between adjacent pixels in the LDI.

The most data-intensive image-based representation is the 4-D lightfield, which captures all rays passing through

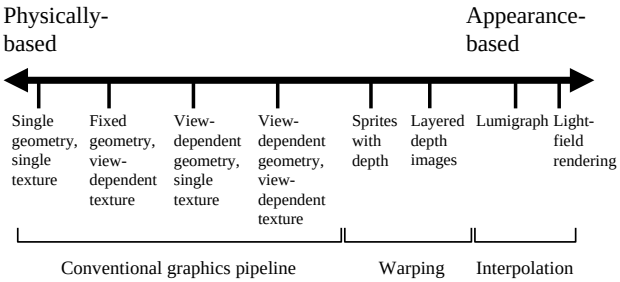

Figure 5: Range of possible spatially-varying representations with their respective dominant rendering mechanisms.

some viewing volume [13]. The Lumigraph [7] uses a similar representation, but also contains a rough 3-D model (see Figure 4) that enables better quality reconstructions using fewer light ray samples. Reduced (e.g., 3-D) Lumigraphs which only sample images along some viewing *path* can use far less data, while still presenting the viewer with a strong sense of parallax, and hence 3-D perception [25, 23].

5 The geometry-image continuum

Based on the complementary sets of characteristics associated with 3-D texture-mapped and image-based models, we can imagine using a continuum of representations to represent a scene. Each representation would be designed to optimize compactness, speed, and visual fidelity; it could change spatially within the same scene. One such range of representations is shown in Figure 5. As can be seen, at the physically-based end of the spectrum, we have the usual single texture-mapped geometry. Along the way, we have variations of geometry models, followed by increasingly image-intensive representations. Note also that the dominant means of rendering changes from the conventional graphics pipeline to warping to interpolation, suggesting that an optimized renderer needs a different specialized hardware to handle these different modes of rendering.

An interesting set of representations that we think is a good bridge from the use of pure geometry to pure images has a view-dependent geometry [17] component. We feel that view-dependent geometry is necessary because stereo algorithms are often prone to errors. In areas where stereo data is inaccurate, we may well represent these areas with view-dependent geometry, which comprises a set of geometry extracted at various positions. For a geometry at

particular viewpoint, despite faulty stereo reconstruction, it is expected that the reconstructed view would still be acceptable for minor virtual camera perturbations. View-dependent geometry may also be used to capture visual effects such as highlights and transparency, where a single geometric representation will fail. *[Anandan: your comments on this?]*

The range of representations to be used for a scene could be determined on a per application basis, i.e., depending on the speed requirement and bandwidth constraint. One possible approach would be to start from the most geometric representation and progressively move to a more and more image-based representation (Figure 5) based on how well the reconstructed view match the expected view, subject to a predefined threshold. The error metric used could be a perceptually-based one as described in [19]. For example, we would expect a blank textureless wall to be represented simply by a plane with possibly view-dependent texture. On the other hand, a plant may be represented by view-dependent geometry with view-dependent texture or a Lumigraph, depending on the desired quality.

6 Discussion

For 2-D image viewing over the Internet, progressive JPEG is a popular format because it allows the user to immediately view the whole image, with increasingly better detail as time progresses. In a similar manner, we can imagine using a form of progressive transmission of representations from pure geometry to more image-based ones for 3-D viewing in web applications. This allows the user to virtually navigate with the normal full degrees of freedom, but with initially impoverished perceptual quality. The quality should increase with time. An interesting alternative is to download high quality images in such a manner as to allow virtual navigation, but with reduced degree of freedom in movement (say, to just rotation). In addition, within each representation, tradeoffs of rendering quality for time can still be used. An example of this is using subsampling or projective textures for the Lumigraph [25].

So far we have discussed the selection criteria from the end-user's or application's point of view. Important considerations exist from the content creator's perspective, such as ease of image acquisition and ease of model reconstruction (both involving expense and achievability). Depending on the level of difficulty of these factors, the selection of models may be limited, regardless of the end-user's or application's requirements. *[I don't understand this last sentence. Remove it? – Rick]*

Another important consideration is the adaptability to the rendering platform. Platforms can vary from PDAs to PCs to supercomputers. In rendering on a PDA, for example, compactness and speed is more important than fidelity of view reconstruction due to its relatively low CPU horsepower and screen resolution. On higher end machines, compactness may be less of an issue.

Determining the optimal set of representations to use for a scene is a complex and very likely time-consuming task. As a result, offline processing to precompute types of representation for rendering will be necessary.

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 [15]. PixelFlow [6] is a high-speed image generation architecture that is based on the techniques of object-parallelism and image composition.

[Anandan, Rick: anything you'd like to add?]

7 Conclusions

We have described various 3-D reconstruction and image-based techniques, and outlined their characteristics. Generally, while 3-D texture-mapped models are compact, their construction is not error-free, and complex visual effects such as highlights and transparency cannot be easily replicated. On the other hand, more image-based representations such as the Lumigraph are capable of producing photorealistic views, but at the expense of a large database and high cost of image acquisition. Based on these two sets of complementary (and conflicting) characteristics, we propose a set of representations that an optimal renderer may use in combination to produce acceptable levels of compactness, photorealism, and speed.

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Conf. on Computer Vision and Pattern Recognition*, pages 434–441, Santa Barbara, CA, June 1998.

[2] E. Camahort and D. Fussell. A geometric study of light field representations. Technical Report TR99-35, Dept. of Computer Sciences, The Univ. of Texas at Austin, Austin, TX, 1999.

[3] J. Chai and H.-Y. Shum. Plenoptic sampling. *Computer Graphics (SIGGRAPH'00)*, page (to appear), Aug. 2000.

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

[5] 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, Aug. 1996.

[6] 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.

[7] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. *Computer Graphics (SIGGRAPH'96)*, pages 43–54, Aug. 1996.

[8] S. B. Kang. A survey of image-based rendering techniques. In *Videometrics VI (SPIE Int'l Symp. on Electronic Imaging: Science and Technology)*, volume 3641, pages 2–16, San Jose, CA, Jan. 1999.

[9] S. B. Kang and H. Q. Dinh. Multi-layered image-based rendering. In *Graphics Interface*, pages 98–106, 1999.

[10] S. B. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. *Int'l J. of Computer Vision*, 25(2):167–183, Nov. 1996.

[11] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, volume A, pages 685–688, Jerusalem, Israel, October 1994. IEEE Computer Society Press.

[12] J. Lengyel. The convergence of graphics and vision. *IEEE Computer*, 1:46–53, 1998.

[13] M. Levoy and P. Hanrahan. Light field rendering. *Computer Graphics (SIGGRAPH'96)*, pages 31–42, Aug. 1996.

[14] Z. Lin and H.-Y. Shum. On the number of samples needed in light field rendering with constant-depth assumption. In *Conf. on Computer Vision and Pattern Recognition*, page (to appear), Hilton Head Island, NC, June 2000.

[15] 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.

[16] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39–46, Aug. 1995.

[17] P. Rademacher. View-dependent geometry. *Computer Graphics (SIGGRAPH'99)*, pages 439–446, Aug. 1999.

[18] P. Rademacher and G. Bishop. Multiple-center-of-projection images. *Computer Graphics (SIGGRAPH'98)*, pages 199–206, July 1998.

[19] M. Ramasubramaniam, S. N. Pattanaik, and D. P. Greenberg. A perceptually based physical error metric for realistic image synthesis. *Computer Graphics (SIGGRAPH'99)*, pages 73–82, August 1999.

[20] H. S. Sawhney. 3D geometry from planar parallax. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 929–934, Seattle, Washington, June 1994. IEEE Computer Society.

[21] S. M. Seitz and C. R. Dyer. View morphing. *Computer Graphics (SIGGRAPH'96)*, pages 21–30, Aug. 1996.

[22] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. *Computer Graphics (SIGGRAPH'98)*, pages 231–242, July 1998.

[23] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. *Computer Graphics (SIGGRAPH'99)*, pages 299–306, August 1999.

[24] H.Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3D models from panoramic mosaics. In *Conference on Computer Vision and Pattern Recognition*, pages 427–433, Santa Barbara, CA, June 1998.

[25] P.-P. Sloan, M. F. Cohen, and S. J. Gortler. Time critical Lumigraph rendering. In *Symp. on Interactive 3D Graphics*, pages 17–23, Providence, RI, Apr. 1997.

[26] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.

[27] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, Mar. 1996.

[28] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, Hilton Head Island, June 2000.

[29] R. Szeliski and R. Weiss. Robust shape recovery from occluding contours using a linear smoother. *Int'l J. of Computer Vision*, 32(1):27–44, June 1998.

[30] Y. Xiong and K. Turkowski. Creating image-based VR using a self-calibrating fisheye lens. In *Conf. on Computer Vision and Pattern Recognition*, pages 237–243, San Juan, Puerto Rico, June 1997.