ACCV2002: The 5th Asian Conference on Computer Vision, 23–25 January 2002, Melbourne, Australia.

Object Representation and Rendering using Inverse Concentric Mosaics

Lifeng Wang , Heung-Yeung Shum
Microsoft Research China
Beijing, 100080, China

Sing Bing Kang
Microsoft Research
Redmond, WA 98052, USA

Abstract

Three of the most basic tasks in computer vision and graphics are capturing, representing, and rendering objects or scenes in as photorealistic a manner as possible. In this paper, we describe our image-based approach to representing and photorealistically rendering objects using what we call Inverse Concentric Mosaics *(ICM). While ICM is fundamentally based on the Concentric Mosaics (CM) [12], ICM permits visualization to be from the outside looking in, which CMs do not allow. Using the ICM, objects can be visualized with continuously and interactively changing viewpoints. Photometric effects can be captured with ICM, thus enhancing photorealism. Finally, the ICM can be very easily acquired and processed. We show examples for synthetic and real objects.*

1 Introduction

Much work in computer vision and graphics has gone into object and scene representation and rendering. Computer vision is far more concerned about image data acquisition, analysis, and representation, while computer graphics is similarly concerned about speed and realism of rendering. The convergence of these two fields is inevitable [6], especially when there is a need for photorealistic rendering of real scenes that conventional graphics cannot deliver.

Over the past few years, image-based rendering (IBR) techniques have been demonstrated to be viable alternatives to the conventional 3D model-based rendering. IBR, which essentially uses images as substrates for rendering, has the advantages of higher potential for photorealism and rendering speeds being independent of scene complexity. A survey on different types of image-based rendering techniques can be found in [5].

1.1 Viewing objects using IBR

There has been a lot of work done on IBR, and we mention only the more relevant approaches here. The reader can refer to [5] for descriptions of other IBR techniques. Two of the more well-known IBR techniques for object visualization are Light Field Rendering [7] and the Lumigraph [3]. Both are 4-parameter subsets of the plenoptic function [1] used to represent ray space, and both adopt a 2-parallel plane configuration. Unfortunately, they require a substantial number of images to be effective. Other IBR techniques attempt to reduce this requirement by adding more geometry to the representation. Examples include:

- Image-based objects [8], in which six Layered Depth Images (LDIs) [11] are used.
- Multiple Center-Of-Projection [10], where central strips with known depth distributions and camera motions are collected over different continuous views as a single pixel plus depth image.
- Point sample rendering [4], where objects are represented as dense point samples. Each pixel has color, normal, and depth information.
- View-based rendering [9], where views are synthesized by combining nearest captured views of texture-mapped 3D models. Each view-dependent 3D model is captured using an accurate range sensor.

All these representations require accurate geometry for optimal quality of view reconstruction. This is less convenient for producing inexpensive off-the-shelf solutions.

1.2 Concentric Mosaics

Concentric Mosaics (CM) [12] has been introduced to visualize a wide set of scenes at a reasonable input size cost and without sacrificing output quality. This is accomplished by further reducing the dimensionality of the plenoptic function to three (namely, horizontal radial distance from a center, rotation angle, and vertical elevation). The acquisition process is simple: images are collected while rotating an outward-looking, forward-displaced camera about a circular path.

Novel views are constructed by rebinning the appropriate captured rays at interactive speeds. The CM is originally intended to visualize wide scenes from the inside looking out. In this paper, we describe a new variant, which we call Inverse Concentric Mosaics (ICM). ICM retains all the

advantages of CM while permitting visualization of objects from the outside looking in.

The differences between CMs and ICMs are as follow:

- *Capture.* For CMs, rays are recorded from the inside looking-out. For ICMs, the camera is fixed while the object of interest is rotated.
- *Depth correction.* CMs use a global depth assumption and ICMs use either global or local geometries.
- *Scene.* Most importantly, CMs represent scenes from inside looking out, while ICMs are more appropriate for visualizing objects from all around.
- *Visibility.* For CMs, the region for panoramic visibility (necessary for visualizing wide scenes) is confined to a disk. For ICMs, where the background is typically not of importance, the region where the virtual camera can be placed to see the object is unlimited. Areas not pre-captured can be artificially filled with the background color used.

1.3 Outline of paper

The remainder of this paper is organized as follows: Section 2 discusses in detail the object-based ICM while Section 3 describes the mechanisms of depth correction and rendering. We show rendering results in Section 4, provide discussions in Section 5, and conclude in Section 6.

2 Object-based ICM

Concentric Mosaics (CM) is a more specific form of manifold mosaics. A manifold mosaic is constructed from vertical slit images taken by a camera at different viewpoints along any continuous path. A concentric mosaic is a special case where this path is circular. All paths of the concentric mosaics have the same center, i.e., all cameras rotating on concentric circles. As a result, the CM is basically an approximated 3D plenoptic function parameterized by the rotation angle, radial distance to rotation axis, and vertical elevation. Our ICM uses the same parameterization.

2.1 Creating the Inverse Concentric Mosaics

Generating an ICM for a synthetic object is easy, since we can control the camera motion through software. We used $3DStudioMax^{TM}$ to prerender an object. The camera is moved along a circular path while oriented towards the object at the center and rendering at predefined intervals. Once all the images are generated, all vertical stripes from each image are rebinned to generate the ICM. For example, if we have 1000 images, each with a resolution of 320×240, then 320 concentric mosaics with dimension 1000×240 are created.

There are a number of ways to capture ICMs of real objects. We use a single fixed camera pointing towards an object that is placed on a turntable (Figure 2). Since there

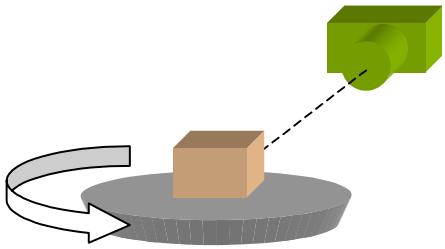


Figure 2. Capturing process for real objects

is an expectation that the entire scene changes with camera viewpoint, we arranged the background to be of uniform color to avoid artifacts during rendering.

The relative object-camera position is known with the assumption of constant rotation speed and frame rate. Again, as with the synthetic object, the ICM can be constructed using the images acquired. Two rebinned concentric mosaics of a real object are shown in Figure 1. Each mosaic is rebinned from 1319 images with resolution of 360×288.

2.2 Generating novel views

A concentric mosaic is composed of rays that are tangent to a cylinder with a common axis. Constructing a novel view is equivalent to computing the rays associated with each constituent pixel. Each ray is retrieved based on its proximity to the concentric mosaics. If the ray does not fall exactly on a concentric mosaic, we can linearly interpolate the closest stored rays on neighboring mosaics by applying some depth compensation technique described later.

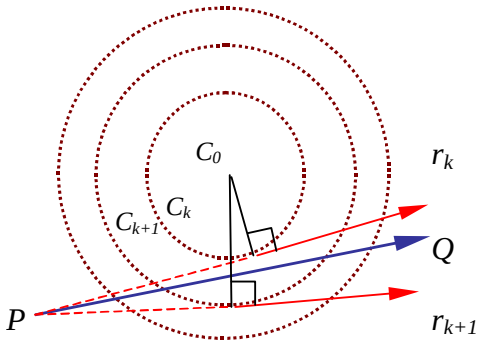


Figure 3. Synthesizing ray PQ associated with novel view P. PQ is interpolated from rays r_k and r_{k+1}, which lie on closest concentric mosaics C_k and C_{k+1}, respectively.

Figure 3 shows how a ray PQ can be extracted from concentric mosaics indicated by $C_0, \ldots, C_{k-1}, C_k$, The radius of concentric mosaic C_k is $r = Rsin(HFOV_k/2)$, where R is the radius of the capture camera path and $HFOV_k$ is the horizontal angle subtended by the kth slit camera.

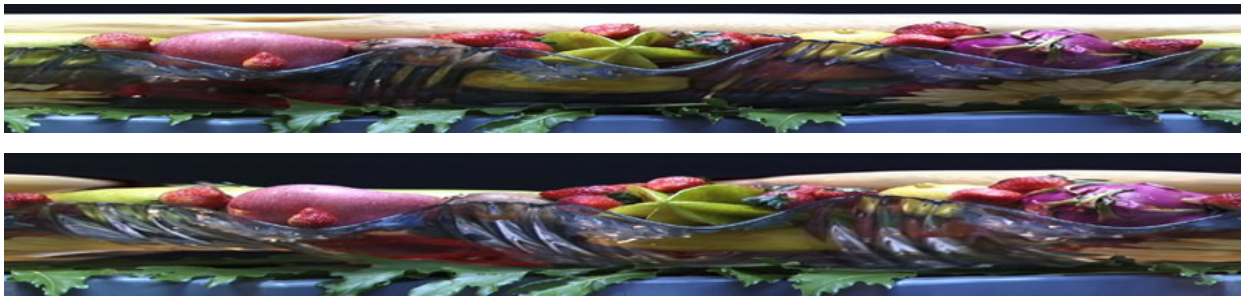


Figure 1. Two rebinned concentric mosaics of a bowl of fruits (see also Figure 7)

3 Depth correction

There is a problem associated with the CM: only a small subset of the rays off the horizontal plane is stored, since only slit images are captured. As a result, off-plane rays have to be approximated from these slit images, leading to the vertical distortions in the final rendered images.

Depth correction is typically applied to reduce the severity of this problem. One simple way is to assume infinite constant depth.

3.1 Infinite depth correction

Infinite constant depth correction is equivalent to using parallel rays to compute the desired ray in the concentric mosaic. The infinite depth assumption is the worst limiting case used in conjunction with linear interpolation.

Although linear interpolation with infinite depth can produce good image quality, it can result in blurred images. A better way is to assume constant depth.

3.2 Constant depth correction

The analysis of plenoptic sampling [2] has shown that if the sampling rate is higher above a certain bound, we can produce antialiased images without the use of correct geometry. The sampling rate for our experiments is chosen so that it is higher than this bound. We use two depth assumptions to interpolate a desired ray. They are constant cylindrical and planar depth assumptions. Because these two assumptions are more accurate than infinite constant depth assumption, we get better results using these two assumptions. The two constant depth assumptions are shown in Figure 4.

The constant cylindrical depth is a global approximation of the geometry. The depth persists even if the viewpoint changes. On the other hand, the planar depth assumption is essential a provision of view-dependent local geometry. The approximated geometry changes with the position of the novel view. Our rendering experiments show that both global geometry and local geometry assumption are improvements over using infinite depth. This is how our rendering algorithm works:

1. For a given novel view, repeat steps 2-4 for each vertical strip.
2. Find two nearest concentric mosaics and its tangent points with respect to the virtual camera center.
3. For each tangent point, use constant depth assumption to find corresponding rays in concentric mosaic.
4. Linearly interpolate the two rays.

Depth correction can be even more generalized to using arbitrary shapes, but this would require additional geometry information of the object viewed.

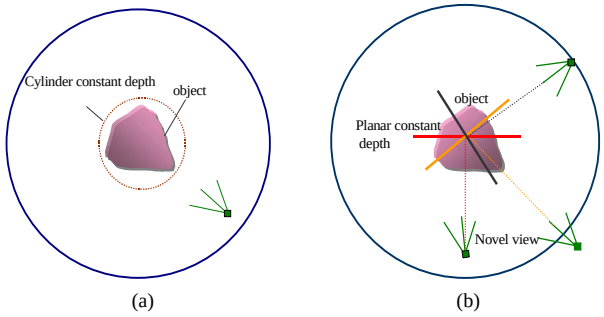


Figure 4. Depth assumption. (a) Constant cylindrical depth, (b) View-dependent constant planar depth.

4 Rendering results

We conducted experiments using synthetic and real objects. All synthetic objects are rendered using $3DStudioMax^{TM}$, while images of the real objects were captured using a Sony DV camera. The images are compressed using vector quantization (VQ). We chose VQ because it facilitates selective decoding and random access, as demonstrated in Light Field compression [7]. Using VQ with a codebook size of 16384, block size of $2 \times 2 \times 2 \times 3$, and image code length of 16 bits, a video of 1000 frames (with resolution of 360×288) was compressed from 306MB down to 22MB.

In addition to file compression, we also used a carefully designed decode buffer system based on the line scheme of CM [12]. For all our rendering experiments, we used a

500MHz Pentium III PC. With the optimizations, we obtained a rendering rate of about 45 fps with linear interpolation for an output resolution of 360×288. For an output resolution of 800×372, the rendering rate dropped down to about 16 fps.

The first set of results is that of a synthetic object. The number of images used is 1200, each with resolution of 320×240. The compressed data size is 22MB. Sample views rendered using our ICM representation are shown in Figure 5. We rendered this object using the constant cylindrical depth assumption. Parallax changes can be clearly seen. The novel views also reflect the lighting changes in the scene (Figure 5(d,e)).

The second set of results is that of a synthetic bonfire (Figure 6). This is an example of a *dynamic* object with semi-repetitive motion. Such objects can be efficiently rendered, and with high quality, using our method. Here, 1400 images were used, each with a resolution of 320×240. The compressed data size is 24MB.

The third set of results is that of a real object (fruits in a glass bowl) as shown in Figure 7. We used a Sony DV camera to capture the object. A total of 2638 images were captured, each with a resolution of 360×288. The compressed data size is about 36MB. The cylindrical depth correction technique was used in this case.

5 Discussion

Compared with most other IBR techniques for rendering objects, ICM has the merits of easy capture, fast rendering (even without using graphics accelerators), and tolerable data size. Rendering techniques that use very few input images generally require accurate geometry information. As a result, they tend to rely on additional vision sensors or algorithms to extract such information. ICM does not require geometry information, at the expense of a higher input image count.

More image-intensive representations such as Light Field [7] and Lumigraph [3] can be used to render objects. However, their database size is considerably higher because they are 4D approximations of the plenoptic function. Like the CM, the ICM is comparatively more practical. Both use less data by limiting the viewpoints to within a planar region. The viewpoint restriction is, however, a small price to pay.

Still, the size of an ICM can be large, even after compression. In addition, the vertical distortion can be apparent. By using more accurate depth information and compression methods with selective decoding capability, or by rendering and compressing within the same transform domain, it may be possible to achieve higher quality view reconstructions and higher compression ratios at the same time. This is a subject for future work.

Another important topic for future work is the ability to stream and randomly access ICMs. Enabling such capabilities will have a significant impact on the practical use of ICMs (and CMs) in web-based applications such as e-commerce, virtual tourism, and games.

6 Conclusion

We have proposed a new representation for rendering objects, which we call Inverse Concentric Mosaics (ICM). It inherits all the advantages associated with the Concentric Mosaics (CM), such as easy acquisition, real-time rendering without the need for graphics accelerators, inclusion of photometric effects, and no requirement for exact geometry. Unlike the CM, ICMs permit objects to be viewed from all around. One of the biggest advantages associated with the ICM is that feature correspondence, one of the more difficult problems in computer vision, is not necessary.

References

[1] E. H. Adelson and J. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing, MIT Press*, pages 3–20, 1991.

[2] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenotpic sampling. *Computer Graphics proceeding, Annual Conference Series, ACM SIGGraph'00*, pages 307–318, 2000.

[3] S. Gortler, R. Grzeszczuk, and M. Szeliski, R. Cohen. The lumigraph. *Computer Graphics proceeding, Annual Conference Series, ACM SIGGraph'96*, pages 43–54, 1996.

[4] J. Grossman and W. Dally. Point sample rendering. *Proc. 9th Eurographics Workshop on Rendering*, pages 181–192, 1998.

[5] S. B. Kang. A survey of image-based rendering techniques. *Proc. Videometrics VI, SPIE Int'l Symp on Electronic Imaging: Science and Technology*, 3641:2–16, 1999.

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

[7] M. Levoy and P. Hanrahan. Light field rendering. *Computer Graphics proceeding, Annual Conference Series, ACM SIGGraph'96*, pages 31–42, 1996.

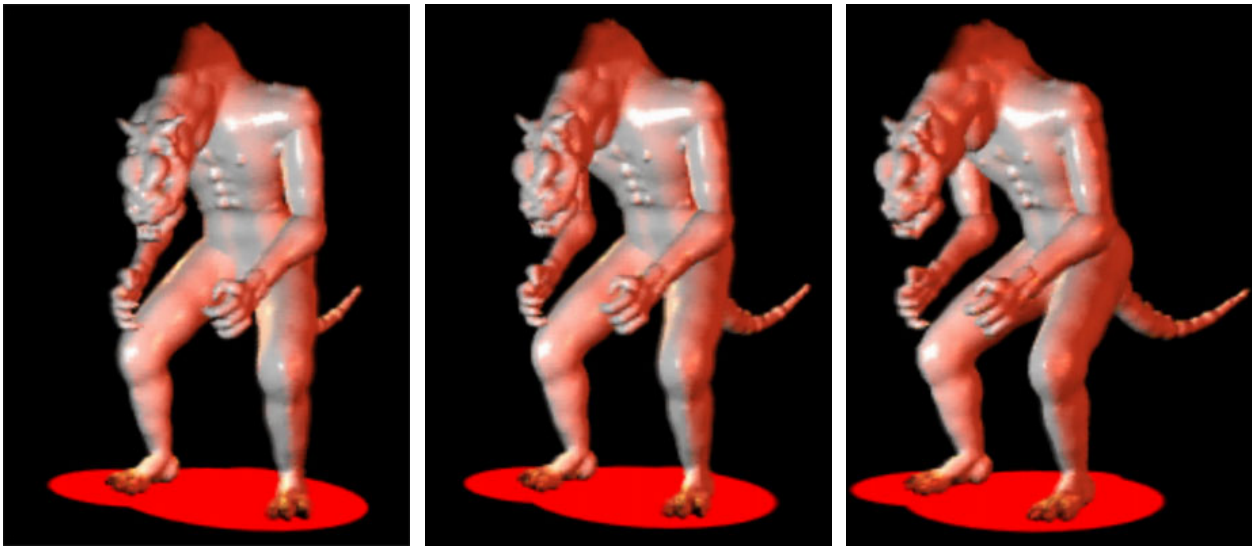
[8] N. Oliveira, M. Manuel, and G. Bishop. Image-based objects. *Proc. ACM Symp. on Interactive 3D Graphics*, pages 191–198, 1999.

[9] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. *Proc. 8th Eurographics Workshop on Rendering*, pages 23–34, 1997.

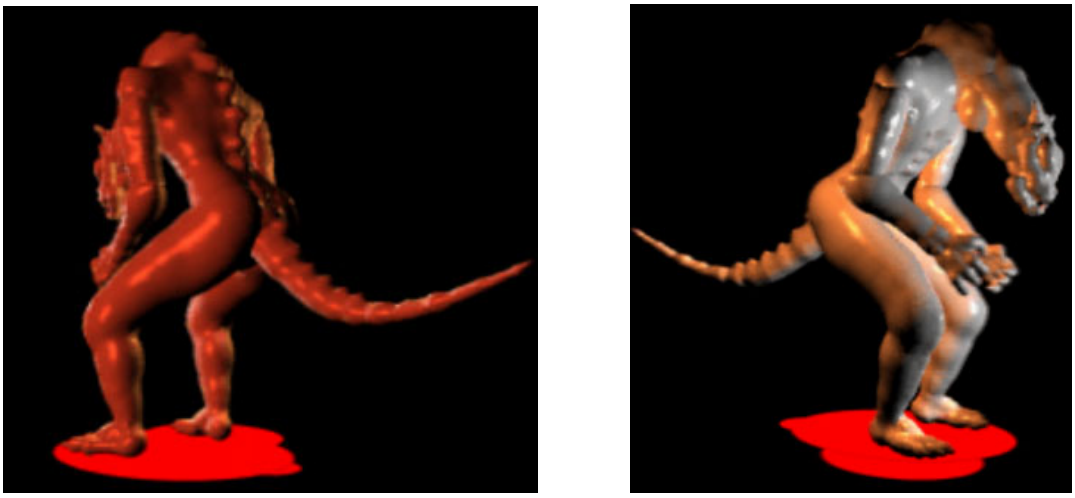
[10] P. Rademacher and G. Bishop. Multiple-center-of-projection images. *Computer Graphics proceeding, Annual Conference Series, ACM SIGGraph'98*, pages 199–206, 1998.

[11] J. Shade, S. Gortler, L.-Y. He, and R. Szeliski. Layered depth images. *Computer Graphics proceeding, Annual Conference Series, ACM SIGGraph'98*, pages 231–242, 1999.

[12] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. *Computer Graphics proceeding, Annual Conference Series, ACM SIGGraph'99*, pages 299–306, 1999.



(a) (b) (c)



(d) (e)

Figure 5. Rendered views of a synthetic object. (a-c) are three translated views. (d) and (e) are 2 rotated views. Note the significant parallax changes. The photometric variations have also been captured (note the lighting changes between (d) and (e)).

Figure 6. A bonfire rendered using ICM. Note that as the view is changed, the flame appears to change in shape. This is because the interpolated rays originate from different frames (hence different timeframes). This 3D dynamism is difficult to render in real-time in traditional graphics.

Figure 7. Three novel views of the real bowl of fruits. The upper two images are rotated views, while the bottom image is the result of moving the virtual camera closer. Note the highlight changes.