Rendering with Concentric Mosaics

Heung-Yeung Shum
Microsoft Research, China

Li-Wei He
Microsoft Research

Abstract

This paper presents a novel *3D* plenoptic function, which we call *concentric mosaics*. We constrain camera motion to planar concentric circles, and create concentric mosaics using a manifold mosaic for each circle (i.e., composing slit images taken at different locations). 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. Like panoramas, concentric mosaics do not require recovering geometric and photometric scene models. Moreover, 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. Compared with a Lightfield or Lumigraph, concentric mosaics have much smaller file size because only a 3D plenoptic function is constructed. Concentric mosaics have good space and computational efficiency, and are very easy to capture. This paper discusses a complete working system from capturing, construction, compression, to rendering of concentric mosaics from synthetic and real environments.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation - Viewing Algorithms.

Additional Keywords: plenoptic functions, virtual environments, image-based rendering.

1 Introduction

The traditional rendering approach to computer graphics starts by constructing a complete geometric and photometric model. Recently, image-based modeling and rendering techniques have been proposed to recover geometric [DTM96, FSL+98, BTZ96] and photometric [SWI97, YM98] models from a collection of photographs. Constructing such models is, in general, time-consuming, and often difficult or even impossible for real environments. Having explicit models makes it simple to add other graphical elements (such as new lights, shadows) later. On the other hand, if we only want to re-render images at a collection of viewpoints, all we need is a complete plenoptic function [AB91], which describes the irradiance perceived from the observer's viewpoints. Other image-based rendering techniques, e.g., view interpolation [CW93] and view morphing [SD96], have also been proposed lately to bypass the difficult and laborious modeling process.

SIGGRAPH 99, Los Angeles, CA USA

Dimension	Viewing space	Name	Year
7	free	plenoptic function	1991
5	free	plenoptic modeling	1995
4	inside a 3D box	Lightfield/Lumigraph	1996
3	inside a 2D circle	concentric mosaics	1999
2	at a fixed point	panorama	1994

Figure 1: A taxonomy of plenoptic functions.

Indeed, a large thread of image-based rendering work has been based on the plenoptic function. The original 7D plenoptic function was defined as the intensity of light rays passing through the camera center at every location, at every possible viewing angle, for every wavelength and at any time. It has been shown that light source directions can also be incorporated into the plenoptic function for illumination control [WHON97]. By ignoring time and wavelength, McMillan and Bishop [MB95] defined plenoptic modeling as generating a continuous 5D plenoptic function from a set of discrete samples. The Lumigraph [GGSC96] and Lightfield [LH96] systems presented a clever 4D parameterization of the plenoptic function if the scene (or conversely the camera view) can be constrained to a bounding box. If the viewpoint is fixed and only the viewing directions and camera zoom can be altered, the plenoptic function simply becomes a 2D panorama (cylindrical [Che95] or spherical [SS97]).

The dimensionality of each of the various plenoptic functions depends on the space of viewpoints (or images) it represents. A comparison of different plenoptic functions is shown in Figure 1. To represent all possible viewpoints without artifacts, we need to have a complete plenoptic function (5D ignoring time and wavelength) if a complete geometric and photometric model is not used. As long as we stay outside the convex hull of an object, we reduce the dimensionality of the plenoptic function from 5 to 4, thus having a Lumigraph or Lightfield. If we stay at a given point, we can further reduce the dimensionality to only 2 (e.g., a spherical map). The question addressed in this paper is: "if we stay in/near some given region/path, as opposed to a given point (panorama) or a bounding box (Lumigraph/Lightfield), what is a good representation for both space and computational efficiency?".[Coh97]

This paper presents such a good representation, a novel *3D* plenoptic function that we call *concentric mosaics*. We constrain camera motion to planar concentric circles, and create concentric mosaics by composing slit images taken at different locations along each circle. Concentric mosaics index all input image rays naturally in 3 parameters: radius, rotation angle and vertical elevation. Compared with a Lightfield or Lumigraph, concentric mosaics have much smaller file size because only a 3D plenoptic function is constructed. Unlike panoramas in which the viewpoint is fixed, concentric mosaics allow the user to move freely in a circular region and observe significant parallax and lighting changes without recovering the geometric and photometric scene models. Novel views are rendered efficiently by combining appropriate rays that are captured and stored in concentric mosaics. However, rendering with concentric mosaics without depth correction causes vertical distortions in the rendered images, because off-the-plane rays can not be synthesized correctly without knowing corrected depth information.

The usefulness of a plenoptic function also depends on how easily it can be captured. Panoramas have become popular because they are easy to construct. Many previous systems have been built to construct cylindrical and spherical panoramas by stitching multiple images together. [MP94, Sze96, Che95, MB95, SS97] When camera motion is very small, it is possible to put together only small stripes from registered images, i.e., slit images [ZT90, PH97], to form a large panoramic mosaic. Capturing panoramas is even easier if omnidirectional cameras [Nay97], mirrors [Nal96], or fisheye lenses are used.

It is, however, very difficult to construct a continuous 5D complete plenoptic function [MB95, KS96] because it requires solving the difficult feature correspondence problem. The 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 [LF94]. To capture a Lightfield/Lumigraph, precise camera poses have to be known (e.g., using a camera rig [LH96]) or recovered (e.g., using camera calibration [GGSC96]). Walk-throughs of a real scene using Lightfield/Lumigraph have not yet been fully demonstrated because capturing an inside-looking-out Lumigraph or Lightfield is difficult.

Capturing concentric mosaics is as easy as capturing conventional panoramas except that we take more images. We put a video camera on a rotary table, simply spin it slowly around an off-centered circle, and capture a continuous video sequence to form concentric mosaics. 3D reconstruction is not needed, nor do feature correspondences have to be computed. To explore a large environment using conventional panoramas, the user may have to choose a large number of panorama sample nodes and jump between nodes.[Che95] By allowing the user to move continuously (within a circular region) to observe motion parallax and lighting changes, concentric mosaics provide a much richer user experience and require fewer nodes to be sampled.

This paper is organized as follows. We introduce the concentric mosaics in Section 2. Rendering with concentric mosaics, including depth correction, is also discussed in Section 2. Construction of concentric mosaics from synthetic and real environments, along with sampling and interpolation issues, is discussed in Section 3. We present experimental results in Section 4. We conclude the paper with a summary and future work.

2 Concentric mosaics

Concentric mosaics are a set of manifold mosaics constructed from slit images taken by cameras rotating on concentric circles. Unlike conventional panoramas, a manifold mosaic[1] [PH97] is composed of images taken by a camera at different viewpoints (or by a set of cameras). In practice, these viewpoints are taken along a continuous path. The composed image is called manifold mosaic because the viewpoints are generally on a continuous surface or curve (a.k.a. manifold). Manifold mosaics have been used in applications such as cel animation [W$^+$97], aerial photogrammetry [GH97], and robot navigation [ZT90].

A possible concentric mosaic capture system setup is shown in Figure 2, where a number of cameras are mounted on a rotating horizontal beam that is supported by a tripod. Each camera is constrained to move continuously and uniformly along its circle. We assume that each camera C_k is a slit camera, i.e., only a vertical line image is taken at the viewpoint v_j. The ray going through the pixel in the slit image that is on the plane is tangent to the circle C_k at v_j. By putting together all lines from C_k at different rotation angles, we obtain a concentric mosaic (manifold mosaic), as shown in the top of Figure 3.

[1] In this paper we use the term manifold mosaic coined by Peleg. There are, however, many other names referring to images taken from different viewpoints, e.g., multiperspective panorama [W$^+$97], multiple centers of projection image [RG98], pushbroom camera [GH97], and pipe mosaic [RPIA98].

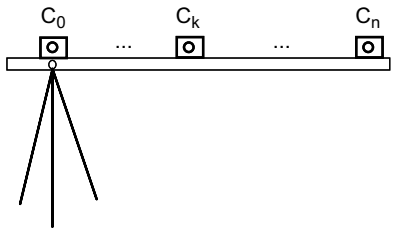


Figure 2: An experimental setup for constructing concentric mosaics.

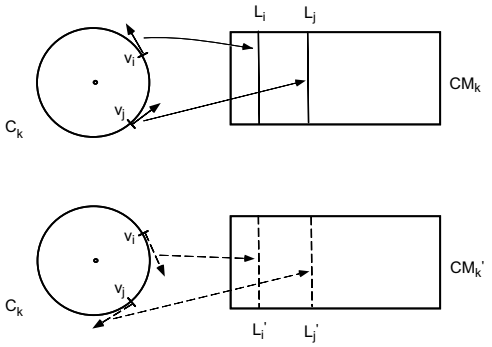


Figure 3: Construction of a concentric mosaic.

As shown in the bottom of Figure 3, for each circle, we would like to capture both viewing directions at each tangent line. Each line $L_j^{'}$ is at the exact opposite direction of L_j. This can be done, for example, with two cameras facing the opposite directions. Therefore, we can capture all the rays going through the circular regions to form a plenoptic function at any point in the region. For clarity, we will only consider concentric mosaic CM from now on. Note that when the camera is located at the rotation center (on the rotation axis), CM_0 is the same as $CM_0^{'}$ but shifted by 180 degrees.

By moving the camera further away from the rotation axis, we obtain a collection of concentric mosaics as illustrated in Figure 4. For example, CM_0 is the mosaic taken when the camera is coincided with the rotation axis. Three examples of mosaic images are shown in Figure 6, each of which corresponds to CM_0, CM_k, CM_n in Figure 4, respectively.

2.1 Rendering a novel view

2.1.1 Rays in the capture plane

Given a collection of concentric mosaics, we can render any novel ray in the capture plane since concentric mosaics have captured most rays in the plane. During the time of rendering, we only need to find out where the rays of the novel view (i.e., a horizontal line image) in the plane are in the previously captured concentric mosaics, or we can bilinearly interpolate them from neighboring mosaics.

This rendering process is illustrated in Figure 5. The ray PV_j is not captured at the novel view point P, but at a different point V_j that is located on the concentric mosaic CM_l. Because the circular region is a free space, the ray captured on the plane at P is the same as that obtained at V_j [LH96, GGSC96]. Similarly, the ray PV_i is captured at V_i in another concentric mosaic CM_k.

2.1.2 Rays off the plane

Only a small subset of the rays off the plane is stored in concentric mosaics because a slit image rather than a regular image is captured. This is why concentric mosaics have a much smaller file size than a Lightfield or Lumigraph. We now have to approximate all rays

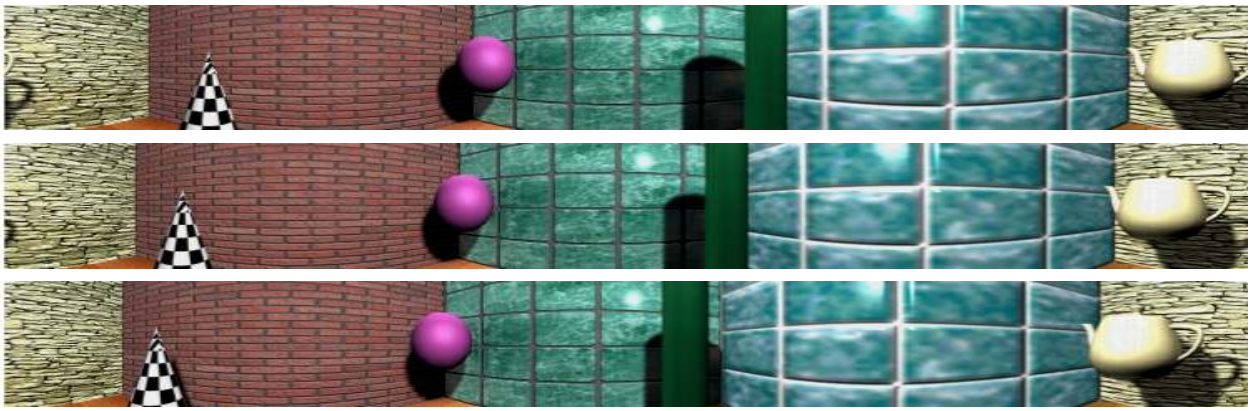


Figure 6: Three examples of concentric mosaics.

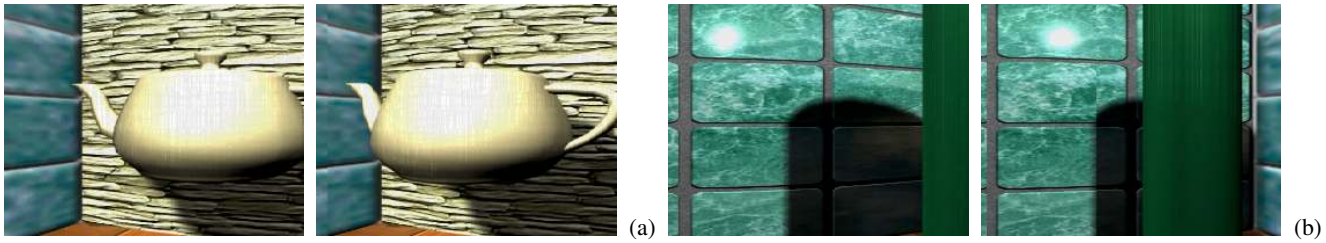


Figure 7: Rendering with concentric mosaics: (a) parallax change; (b) specular highlight and parallax change.

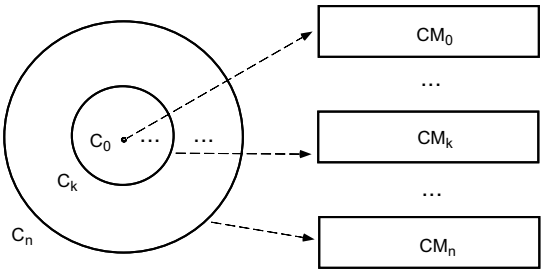


Figure 4: A collection of concentric mosaics.

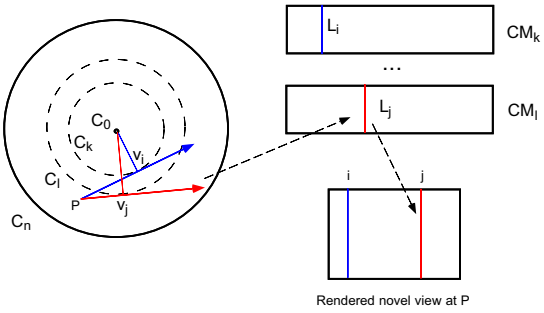


Figure 5: Rendering a novel view with concentric mosaics.

off the plane from only the slit images. As shown in Figure 5, to render a vertical line including rays off the plane at a novel view P, we simply take the entire slit line L_j at position V_j from concentric mosaic CM_l. PV_j is tangent to the concentric circle C_k on the plane.

The rendering is therefore very efficient because the concentric mosaics are indexed properly for efficient retrieval of each line. Details of rendering equations for concentric mosaics are given in Appendix A. Figure 7(a) illustrates the parallax change near the teapot mouth while Figure 7(b) demonstrates the specular highlight movement on the wall presented in rendered views of a synthetic environment. These novel views are rendered with a set of 12 concentric mosaics (three of them are shown in Figure 6).

By taking the whole line from a different location, however, we are making an implicit infinite depth assumption, i.e., objects that the line of rays meets are at infinity. As shown in Figure 8, parallel rays (pixels with same vertical field of view from two viewpoints P_{CM} and P_{new}) only meet at infinity. This approximation will cause vertical distortion in the rendered images.

2.2 Depth correction

To alleviate the above vertical distortion problem, several strategies can be used.

2.2.1 Full perspective correction

If the distances between the camera optical center and the points in the scene are known, full perspective correction based on distances of individual pixels can be used. The pixel values of the captured rays can be warped to the rendered view using optical flow. However, there are two problems with this approach. First, geometries are usually not known for real scenes and recovering geometries from images is known to be a hard vision problem. Second, hole-filling problems still exist. A method similar to layered-depth image [SGHS98] can be used at the cost of storing more pixels (both color and depth) per sampled ray. For synthetic scenes where precise 3D geometries are known, this approach yields correct novel views and

is the preferred approach.

2.2.2 Weak perspective approximation

In many real scenes, it is a reasonable approximation that pixels from a vertical line have the same depth. In that case, we can use a weak perspective approximation. Efficient rendering algorithms exist if weak perspective camera model is used for each line. We only need to estimate a depth value for each vertical line and scale the whole line uniformly. Vertical distortions increase with the amount of depth variation in each vertical line and decrease with the distances of the scene objects from the camera.

2.2.3 Constant depth approximation

The depth information requirement can be further reduced to be a constant for all the captured vertical lines. This approach is essentially a cylindrical environment map that dynamically updates its content based on the location of the novel view. Like the weak perspective case, there are vertical distortions for objects whose depth is different from the assumed depth. The Lumigraph system [GGSC96] uses a similar approach to put the (u, v) plane at the object center. In our system, the user can interactively adjust the assumed depth with a slider control to correct the distortion to the object of interest.

2.3 3D plenoptic function

It is now straightforward to show that the concentric mosaics representation is a plenoptic function parameterized by three variables: rotation angle, radius (distance of camera to rotation axis), and vertical elevation or field of view (FOV), as illustrated in Figure 9.

No vertical parallax is captured in concentric mosaics because all the camera views are constrained to a horizontal planar region, and only a slit image is taken at each viewpoint. However, as shown in our real-time rendering experiments and demonstrated years ago by the horizontal parallax only holographic stereograms [Ben83], people still have a strong sense of 3D perception even with only horizontal parallax. Perhaps it is due to the fact that our eyes remain relatively planar and observe mainly horizontal parallax as we move around.

Sloan *et al.*[SCG97] have also derived a 3D parameterization of the Lumigraph by replacing the (s, t) plane with its 1D subset, i.e., constraining the camera motion along a line. A (s, u, v) representation is used where s parameterizes the camera motion (i.e., drop t in the (s, t) plane[2]), and (u, v) parameterizes the plane roughly in the object center. Moving along the line provides parallax in the motion direction. To have a complete coverage of the object, the camera can move on four connected perpendicular lines, i.e., a square. This sampling is not as uniform as concentric mosaics because one has to switch from one line to another. But it has the advantage that straight lines are preserved from 3D Lumigraph to rendered images, therefore has better computational efficiency.

3 Construction of concentric mosaics

3.1 Synthetic scenes

For synthetic environments, we use a renderer (3D Studio Max, www.ktx.com) to render slit images at different concentric circles. These slit images are then put together to construct concentric mosaics. Some examples of concentric mosaics are shown in Figure 6. With the help of a z-buffer plug-in, we are also able to obtain a depth value for each pixel.

How do we sample concentric mosaics in both radial and angular directions to ensure that the user moves uniformly inside a unit circle? It is fair to sample uniformly the angular direction

[2]Here we follow [SCG97] and use the Lumigraph notation. The notation in Lightfield flips (u, v) and (s, t).

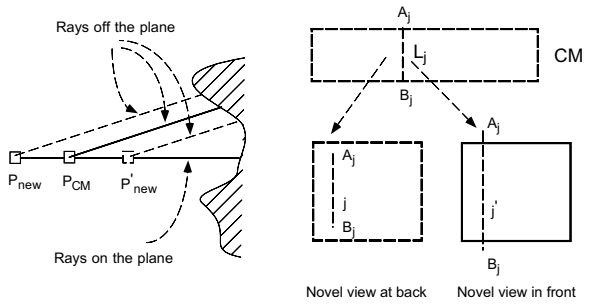


Figure 8: Depth correction with concentric mosaics.

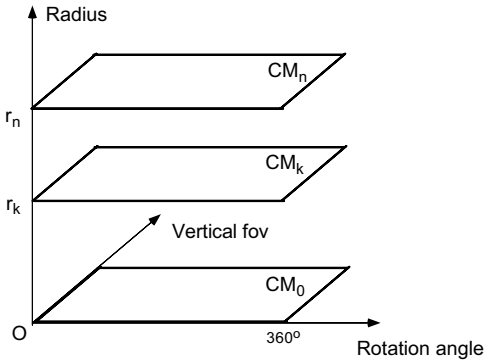


Figure 9: Concentric mosaics represent a 3D plenoptic function.

(e.g., every tenth of a degree). But in the radial direction, the sampled circles should not be at locations $\{0, 1/n, 2/n, ..., 1\}$, but at $\{0, \sqrt{1/n}, \sqrt{2/n}, ..., 1\}$ because the median circle dividing the unit circle into two halves of equal areas is not at $1/2$ but at $\sqrt{1/2}$.

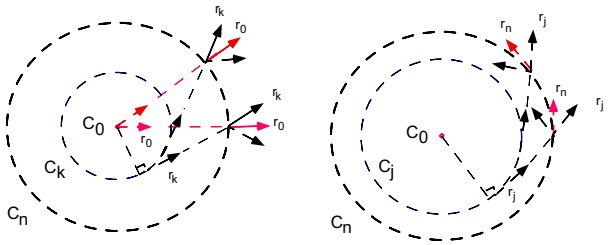
How many samples do we need? In a typical novel view with 36 degrees horizontal FOV and 200 width image, 2000 samples in the angular direction are needed. The more samples taken in the radial direction, the less interpolation is required (therefore better rendering results). Typically, we sample 20 concentric circles in 3000 angular directions, i.e., 20 concentric mosaics with a width of 3000. The width of each slit image is determined according to its horizontal field of view (e.g., 0.12 degree for 3000 samples at each circle).

3.2 Real scenes

We considered a number of ways to capture concentric mosaics from a real scene. One design was having many cameras on a beam which rotates, as shown in Figure 2. The problem with this design is that the whole system is bulky and expensive because many cameras are needed. Another design was to use a single camera that shifts to different locations before moving in a new circle. A motorized one-dimensional linear X-stage can be used to control the camera location. However, both designs require synchronization and alignment of multiple mosaics.

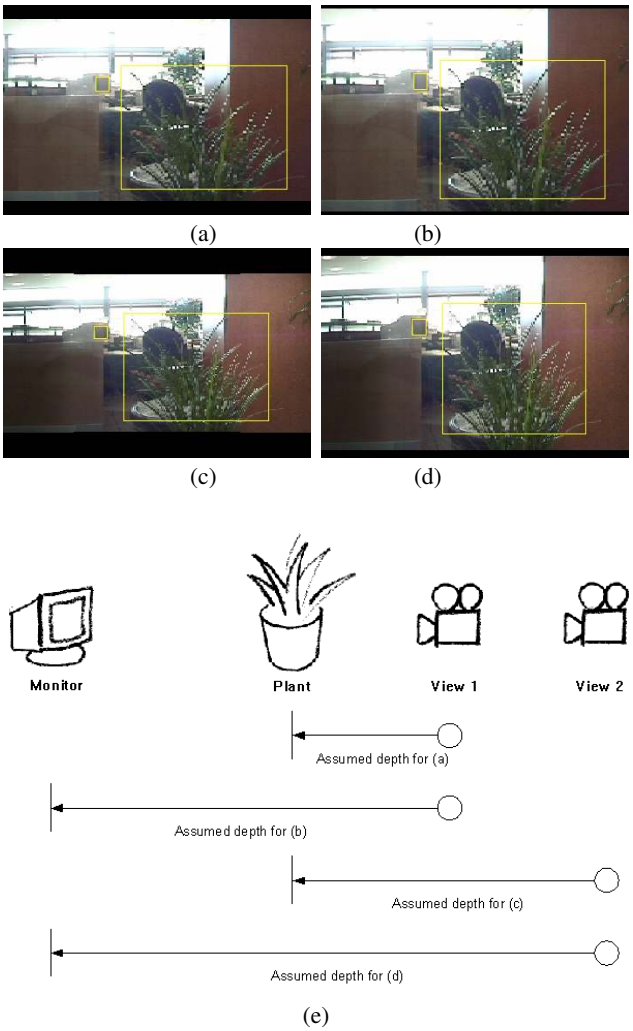
A much simpler design is to use a single off-centered camera that rotates along a circle. The rotation is known with the use of a rotary table. At each rotation angle, instead of a slit line image, a regular image with multiple vertical lines (depending on the horizontal FOV of the image) is captured.[PBE99]

As shown in Figure 10a, no matter at which point along the circle C_n the camera is located, the same indexed ray in the plane (e.g., the left most line r_k) is always tangent to the same circle C_k. A concentric mosaic is formed by putting together the same vertical lines (e.g., the $20th$ vertical scanline) taken at different rotation



(a) (b)

Figure 10: Construction of concentric mosaics from one circle: camera along (a) normal direction; (b) tangential direction.



(a) (b) (c) (d) (e)

Figure 11: Rendering with constant depth correction: (a) and (c) front view and back view with depth correction $4R$; (b) and (d) front view and back view with depth correction $16R$; (e) illustration of camera viewpoints, front view A, back view B. The aspect ratio of the plant is preserved in (a) and (c), but not in (b) and (d).

angles. A sequence of M regular images (taken along a circle) with size $H \times (2W-1)$ can be rebinned into W concentric mosaics with size $M \times H$. It is possible to subsample the concentric mosaics by simply dropping columns in the original images. The rebinning process, then, can be skipped to avoid double sampling.

Figure 10 illustrates two possible setups, one along the normal direction[3], the other along the tangential direction. The normal setup covers the inner circle (from C_0 to C_k), while the tangential setup covers the outer ring (between C_j to C_n). Putting all middle vertical lines together, we obtain the concentric mosaic CM_0 from the normal setup, but the concentric mosaic CM_n from the tangential setup.

Capturing with one circular motion is easy. However, the resulting visible (or movable) region is significantly limited by the camera's horizontal FOV. For the normal setup, the radius of the visible inner circle R_k depends on the horizontal field of view of the camera ($HFOV$). Indeed,

$$R_k = R_n \sin(HFOV/2). \quad (1)$$

Because vertical lines in an input image have different horizontal FOV's (e.g., the line further away from the middle has smaller FOV than the middle line), they have to be resampled to ensure uniform sampling. This resampling process becomes significant when the camera's horizontal FOV is large. More importantly, the average distance between the novel view (to be rendered) and the captured view (to be used for rendering) is much larger if only one circle is used instead of having multiple circles taken. Longer distance causes bigger depth distortion. This is an important difference between 3D concentric mosaics and 4D Lumigraph: for concentric mosaics, not only do we want to find the captured ray on the plane, but we also want to find the ray as close to the novel view as possible.

A better capturing device is to use a few regular cameras, aligned as shown in Figure 2, with the tangential setup as shown in Figure 10b. Each camera can cover a ring of visible regions.

4 Results for real scenes

In our experiments of rendering with concentric mosaics for real environments, we have used a digital video camera (Sony Mini DV digital video camera) with a resolution of 320 by 240. A rotary table (Parker $5''$ table and Parker 6104 Indexdrive controller) is used to slowly rotate the camera along an off-centered circle and to provide accurate rotation parameters. If we do not control the rotation precisely, vision techniques such as camera calibration and motion estimation could be used to recover the rotation parameters. There are two reasons why we rotate the camera slowly. The first is to get enough samples along the angular direction. The second is to avoid motion blur. With each full circle motion of about 90 seconds, a total of 1351 frames is recorded. It took a total of only 10 minutes to set up, capture and digitize a complete sequence of video needed for constructing concentric mosaics.

Renderings of a lobby scene from captured concentric mosaics are shown in Figure 14. A rebinned concentric mosaic at the rotation center is shown in Figure 14(a), while two rebinned concentric mosaics taken from exactly opposite directions at the outermost circle are shown in Figures 14(b) and (c), respectively. It has been shown in [PBE99] that such two mosaics taken from a single rotating camera can simulate a stereo panorama.

Rendered images are shown in Figures 14(d)(e)(f). In Figure 14(d), a child can be observed in one view but not the other. In Figure 14(e), strong parallax can be seen between the plant and the poster. In Figure 14(f), lighting changes caused by sunshine can be observed near the ceiling. The dramatic lighting change is partly due to the fact that we have used automatic exposure for the camera during capturing.

[3] In general, the middle ray on the plane is set along the normal direction.

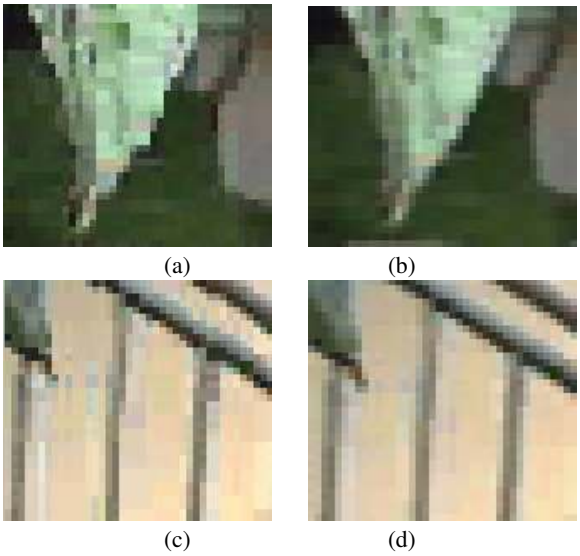


Figure 12: Comparison between point sampling (a)(c) and bilinear sampling (b)(d): blown-up images from rendering the lobby scene in Figure 14.

Constant depth correction is used in the real scene examples. Currently it is the user's responsibility to provide the depth estimation for the region of exploration. If the depth is estimated correctly, we would expect the aspect ratios[4] of objects remain the same after depth correction regardless of the distance between the object and the camera. This is indeed the case. In the lobby scene shown in Figure 14, two views rendered with different depth corrections are shown in Figure 11. The bottom images (c) and (d) are taken at the viewpoint 2, and the top images (a) and (b) at the viewpoint 1. The left images (a) and (c) are depth-corrected using a constant depth at the plant, while the images on the right (b) and (d) at the monitor. Figure 11e shows two viewpoints 1 (in front) and 2 (at back) which point to the same direction. The aspect ratios of the plant using depth correction at the plant are almost identical while they are different using depth correction at the monitor. The reverse is true for the computer monitor.

The input video sequences can be compressed well because of the significant spatial adjacency between frames. We used vector quantization since it is a good choice for selective decoding, as demonstrated in Lightfield compression [LH96]. Using vector quantization, the original video of 1351 frames (of size 320x240) can be compressed to 25M from 415M bytes. Entropy coding can be applied to further reduce the file size to 16M. On a Pentium II PC running at 300 MHz, using the compressed file, we achieved a frame rate of 20 frames per second for rendering with concentric mosaics with bilinear interpolation. Results are shown in Figure 14. Rendering using point sampling is more than twice as fast as using bilinear interpolation, but has lower visual quality, as shown in Figure 12.

We have also applied a standard MPEG4 codec to our video sequences and achieved very high compression ratios. For example, with only 1 I-frame (the first image) and 56k bit rate, the lobby video can be compressed to 640k, which is a compression ratio of 648. At such high compression ratio, block artifacts become visible even though motion parallax and lighting changes are still observed.

[4] Aspect ratio is a good approximation for frontal parallel objects and can be easily computed by identifying the boundary of the object in the image. Ideally, the cross ratio of four co-linear points [Fau93] should be computed to verify if the perspective projection is indeed preserved.

5 Conclusion and future work

In this paper we have presented a novel *3D* plenoptic function, which we call *concentric mosaics*. We constrain camera motion to planar concentric circles, and create concentric mosaics by composing 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 continuously in a circular region and observe significant parallax and lighting changes. The ease of capture makes concentric mosaics very attractive for many virtual reality applications.

It is important, however, to pay attention to vertical distortions while using concentric mosaics. For example, one should capture small field of view environments because vertical distortions increase with larger field of view. Vertical distortions also become more apparent as the user moves backward and forward because significant parallax change occurs. On the other hand, parallax change caused by lateral moves is significantly less, and can therefore be better compensated.

While vertical distortions can be reduced with constant depth correction demonstrated in our experiments, they still exist. The next step in our approach will be to reconstruct 3D depth from the input images using vision techniques such as structure from motion or stereo. Stereo from concentric mosaics is of particular interest because of the special camera configuration and its resulting epipolar geometry. We are currently working on this problem. We are also working on moving from a set of concentric mosaics to another so that the region of motion can be enlarged.

Another important area of future work is the compression of concentric mosaics. Although currently satisfactory, vector quantization will become less and less acceptable as we increase image size and capture more sets of concentric mosaics. Higher compression ratios can be achieved with motion compensated compression techniques. For example, we achieved a very high compression ratio using an MPEG4 codec to compress concentric mosaics. But selectively decoding the compressed MPEG bit stream (depending on the viewing direction) is difficult because of the inter-frame and intra-frame dependency. Better compression algorithms for selective decoding enable us to effectively explore a large environment through the use of multiple sets of concentric mosaics.

6 Acknowledgements

This work was inspired by Shmuel Peleg's talk on manifold mosaic at Microsoft Research in 1997 and Michael Cohen's idea to constrain the camera motion. The authors benefited from discussion with Qifa Ke, Anandan, Steve Seitz, Michael Cohen, Rick Szeliski, Jim Kajiya, Mike Sinclair, Peter-Pike Sloan and Steve Gortler. Timely help from Dan Robbins on using 3D Studio Max, Dave Thiel on making Siggraph submission video, Hong-Hui Sun and Min-Sheng Wu on generating pictures in Figure 14 is very much appreciated.

References

[AB91] 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.

[Ben83] S. A. Benton. Survey of holographic stereograms. In *Proc. SPIE Vol. 367 Int. Soc. Eng.*, pages 15–19, 1983.

[BTZ96] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Fourth European Conference on Computer Vision (ECCV'96)*, volume 2, pages 683–695, Cambridge, England, April 1996. Springer-Verlag.

[Che95] S. E. Chen. QuickTime VR – an image-based approach to virtual environment navigation. *Computer Graphics (SIGGRAPH'95)*, pages 29–38, August 1995.

[Coh97] M. F. Cohen. Personal email communication, September 1997.

[CW93] S. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics (SIGGRAPH'93)*, pages 279–288, August 1993.

[DTM96] 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.

[Fau93] O. Faugeras. *Three-dimensional computer vision: A geometric viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[FSL+98] O. D. Faugeras, Laveau S., Robert L., Csurka G., and Zeller C. 3-D reconstruction of urban scenes from sequences of images. *Computer Vision and Image Understanding*, 69(3):292–309, March 1998.

[GGSC96] 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.

[GH97] R. Gupta and R.I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, September 1997.

[KS96] 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.

[LF94] 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.

[LH96] 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.

[MB95] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39–46, August 1995.

[MP94] 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.

[Nal96] V. S. Nalwa. A true omnidirecional viewer. Technical report, Bell Laboratories, Holmdel, NJ, USA, February 1996.

[Nay97] 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.

[PBE99] S. Peleg and M. Ben-Ezra. Stereo panorama with a single camera. In *Proc. Computer Vision and Pattern Recognition Conf.*, 1999.

[PH97] 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.

[RG98] P. Rademacher and Bishop G. Multiple-center-of-projection images. In *Computer Graphics Proceedings, Annual Conference Series*, pages 199–206, Proc. SIGGRAPH'98 (Orlando), July 1998. ACM SIGGRAPH.

[RPIA98] B. Rousso, S. Peleg, Finci I., and Rav-Acha A. Universal mosaicing using pipe projection. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 945–952, Bombay, January 1998.

[SCG97] 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.

[SD96] 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.

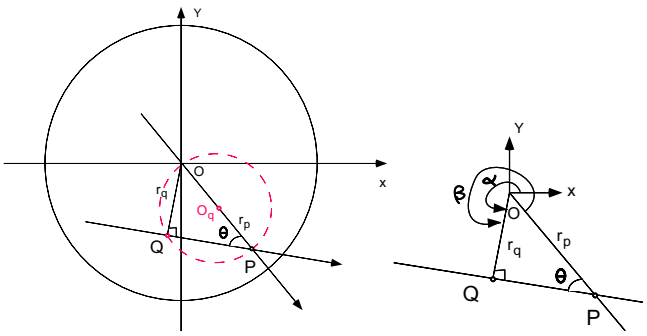


Figure 13: Rendering with concentric mosaics.

[SGHS98] 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.

[SS97] 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.

[SWI97] Y. Sato, M. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 379–387, Proc. SIGGRAPH'97 (Los Angeles), August 1997. ACM SIGGRAPH.

[Sze96] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.

[W+97] D. N. Wood et al. Multiperspective panoramas for cel animation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 243–250, Proc. SIGGRAPH'97 (Los Angeles), August 1997. ACM SIGGRAPH.

[WHON97] 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.

[YM98] Y. Yu and J. Malik. Recovering photometric properties of architectural scenes from photographs. *Computer Graphics (SIGGRAPH'96)*, pages 207–218, July 1998.

[ZT90] 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.

A Rendering equations for concentric mosaics

As shown in Figure 13, given the center of concentric mosaics O, a viewpoint P, and its viewing direction θ, where is the point Q on the concentric mosaics?

The polar coordinates (r_p, θ_p) of the viewpoint P are defined as

$$r_p = \sqrt{(P_x - O_x)^2 + (P_y - O_y)^2} \quad (2)$$

$$\theta_p = atan2(P_y - O_y, P_x - O_x) \quad (3)$$

and the polar coordinates (r_q, β) of the point Q become

$$r_q = r_p |\sin(\theta)| \quad (4)$$

$$\beta = \theta + \theta_p - \pi/2 \quad (5)$$

where $\alpha = \theta - \pi/2$ and $\alpha - \beta = -\theta_p$.

It is apparent that as θ increases from 0 to 180 degrees, the trajectory of Q is a circle centered at the middle of OP (O_q in Figure 13) and going through O and P. Indeed,

$$Q_x - O_{qx} = -\frac{r_p}{2}\cos(2\theta + \theta_p) \quad (6)$$

$$Q_y - O_{qy} = -\frac{r_p}{2}\sin(2\theta + \theta_p). \quad (7)$$

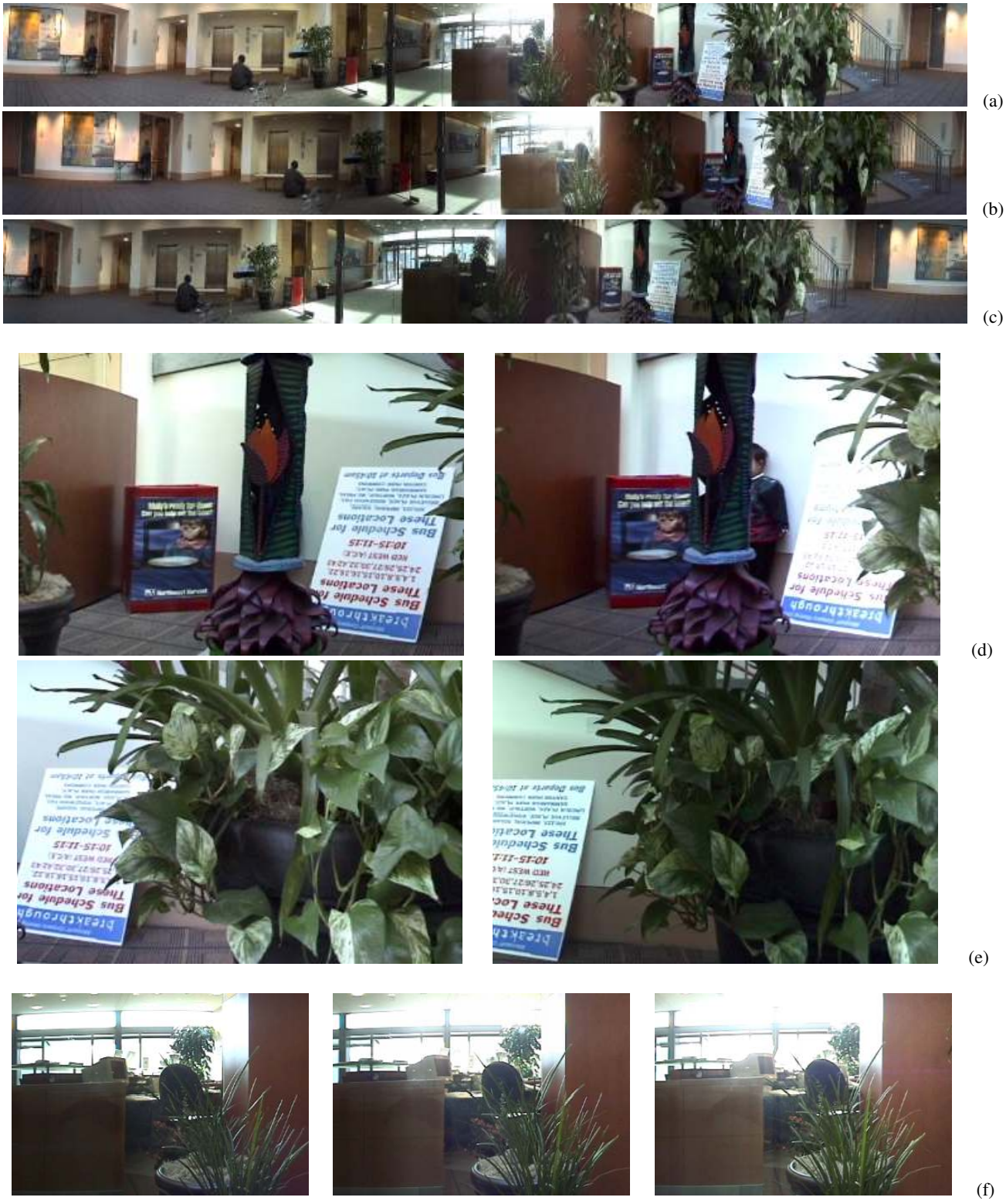


Figure 14: Rendering a lobby: rebinned concentric mosaic (a) at the rotation center; (b) at the outermost circle; (c) at the outermost circle but looking at the opposite direction of (b); (d) a child is occluded in one view but not in the other; (e) parallax change between the plant and the poster; (f) lighting changes near the ceiling caused by sunshine.