

The Rayset and Its Applications

Minglun Gong

Yee-Hong Yang

Department of Computer Science
University of Saskatchewan

Abstract

In this paper, a novel concept, rayset, is proposed. A new image based representation, object centered concentric mosaics (OCCM), is derived based on this concept. The rayset is a parametric function, which consists of two mapping relations. The first mapping relation maps from a parameter space to the ray space. The second one maps from the parameter space to the attribute space. We show that different image-based scene representations can all be cast as different kinds of raysets, and image-based rendering approaches can be regarded as attempts to sample and reconstruct the scene using different raysets. A collection of OCCM is a 3-D rayset, which can be used to represent an object. The storage size for OCCM is about the same as an animation sequence generated with a virtual camera rotated around the object. However, comparing with such an animation sequence, OCCM provide a much richer experience because the user can move back and forth freely in the scene and observe the changes in parallax.

Keywords: Image-based rendering, Rayset, Plenoptic function.

1 Introduction

Many approaches have been proposed in the image-based rendering area. Based on the kind of scenes that can be represented, these approaches can be classified into the following categories:

- Represent an environment [5, 6, 13, 16, 17, 28]
- Represent a layer of an environment [22, 27]
- Represent an object [9, 19, 23, 24, 30]
- Represent part of an object [2, 8, 10, 20, 25]
- Represent both environments and objects [3, 11, 12, 14, 15, 26, 29]

McMillan and Bishop claim that “all image-based rendering approaches can be cast as attempts to reconstruct the plenoptic function from a sampled set of that function” [18]. However, their framework does not include texture-mapping based approaches [2, 8, 20].

Chai et al. [4] analyze how many samples of the plenoptic function are needed to reconstruct a new scene. Nonetheless, their derivation is based on the sampling scheme used in light field rendering only, and they do not try to classify different sampling schemes. In addition, the question of what is an image [7] has not yet been answered. Conventionally, an image is considered as a 2-D RGB valued function. Recently, depth images and multiple-center-of-projection images have been proposed for image-based rendering. The question is: Are they still images? In addition, 4-D light field [14] or lumigraph [11] and 3-D concentric mosaics [28] are also introduced. So, what are their relations with the conventional images?

In this paper, a novel concept, rayset, which is more general than the concept of an image, is proposed. Based on the sampling schemes used, we classify different image-based scene representations as different kinds of raysets. Different image-based rendering approaches, therefore, can be regarded as attempts to sample and reconstruct the scene using different raysets.

Based on the rayset concept, we propose a novel image-based representation — object centered concentric mosaics (OCCM for short). The OCCM is a 3-D rayset, which can be used to represent an object. The storage size of OCCM is about the same as a sequence of an animation generated with a virtual camera rotated around the object. However, comparing with such an animation sequence, OCCM provides a much richer experience because it allows a user to move back and forth freely in the scene and to observe the changes in parallax.

The rest of the paper is organized as follows. First, in section 2, the definition of rayset is given, and based on it, several commonly known scene representation techniques are classified. Then, in section 3, the novel object centered concentric mosaic is discussed. The paper concludes in section 4.

2 Rayset

The plenoptic function [1] defines the intensity of ray at direction (\mathbf{q}, \mathbf{f}) , from point (x, y, z) in space, at time t , and with a wavelength I . For a static scene, and consider only the visible light domain, the original plenoptic function can be reduced to five dimensions, which maps any ray in space to a color vector [18]:

$$\mathbf{C} = P(x, y, z, \mathbf{q}, \mathbf{f}) \quad (1)$$

The plenoptic model is an explicit function that depicts the scene. We can also try to use a parametric function to describe the scene. This inspires the definition of a rayset.

2.1 Definition of Rayset

A rayset is a parametric function, which is defined as:

$$\begin{cases} (x, y, z, \mathbf{q}, \mathbf{f}) = \mathbf{S}(\mathbf{U}) \\ \mathbf{T} = \mathbf{A}(\mathbf{U}) \end{cases} \quad (2)$$

where \mathbf{U} is the parameter space, \mathbf{T} the attribute space¹, $\mathbf{S}(\mathbf{U})$ the support function, which maps from the parameter space to the 5-D ray space, and $\mathbf{A}(\mathbf{U})$ the attribute function, which maps from the parameter space to the attribute space. The dimension of the parameter space is called the dimension of the rayset.

Different image-based rendering approaches can be cast as attempts to sample and reconstruct the scene using different raysets. In the rayset generation stage, the attribute function needs to be constructed. Normally, this is fulfilled through sampling and enumerating the parameter space. For each given set of parameters, the support function is used to find the corresponding testing ray, which starts from the viewpoint. The attributes of testing rays are obtained through either rendering a synthetic model, taking pictures, or resampling existing images.

There are essentially two different ways to render a rayset, which are the forward and the backward approaches. In the backward approaches, for a given testing ray, parameters are computed through a reverse process based on the support function. The attributes of the ray can then be found directly using the attribute function. In the forward approaches, all the rays defined in the rayset are enumerated. Each ray is reprojected to the desired view and the attributes of the ray are used to color its corresponding ray in the desired view.

It is noteworthy that the direct inverse function of

¹ The attribute space can be one of the color spaces, such as RGB. It may also be of higher dimensions and carry information of other attributes such as transparency, depth, surface normal, etc.

the support function may not exist since the parameter space normally has fewer dimensions than the ray space. The reverse process is accomplished under some assumptions. For example, in light field rendering, it is assumed that the space is transparent and that the region we consider is free of occluders. Therefore, rays that start from different locations along the direction of the ray are considered the same.

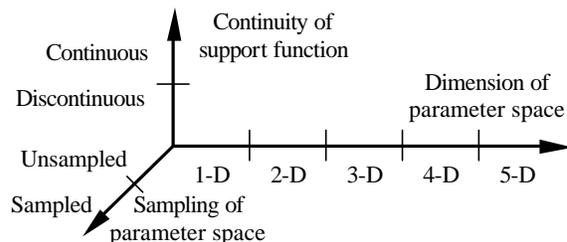


Figure 1: Classification of raysets.

As shown in Figure 1, raysets can be classified according to the dimension of the parameter space. Contingent on whether or not the support function is continuous, raysets can be classified into continuous raysets and discontinuous raysets. Furthermore, they can be classified into sampled raysets and unsampled raysets depending on whether or not the parameter space is discretized.

Most of the image-based rendering techniques employ some kinds of 2-D raysets. Many of these approaches use the simplest planar projection images. In section 2.2, three special planar projection images, which are perspective images, parallel images, and object images, are discussed first. Other kinds of 2-D raysets are covered in section 2.3. High-dimensional raysets are discussed in section 2.4.

2.2 Perspective, Parallel, and Object Images

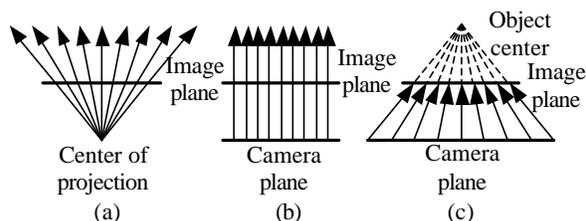


Figure 2: (a) perspective image (b) parallel image (c) object image.

A perspective image is an image obtained using a planar pinhole camera. As shown in Figure 2(a), all the rays defined in a perspective image start from the center of projection, a.k.a. the viewpoint. Based on the planar pinhole camera model, the support function of the corresponding rayset representation can be defined as:

$$\begin{cases} \mathbf{S}_x(j,k) \equiv \mathbf{C}_x \\ \mathbf{S}_y(j,k) \equiv \mathbf{C}_y \\ \mathbf{S}_z(j,k) \equiv \mathbf{C}_z \\ \mathbf{S}_q(j,k) = \arctan2(\mathbf{d}_y + js_y + kt_y, \mathbf{d}_x + js_x + kt_x) \\ \mathbf{S}_f(j,k) = \arcsin(\mathbf{d}_z + js_z + kt_z / \|\mathbf{d} + js + kt\|) \end{cases} \quad (3)$$

where \mathbf{C}_x , \mathbf{C}_y , and \mathbf{C}_z are the x-, y-, and z-coordinates of the center of the projection, \mathbf{d} the viewing direction, \mathbf{s} and \mathbf{t} the basis vectors of the image plane.

A perspective image represents a scene that is visible from a given viewpoint. It is the most commonly used representation in image-based rendering. Figure 3 shows the relationship among different representations that are based on perspective images.

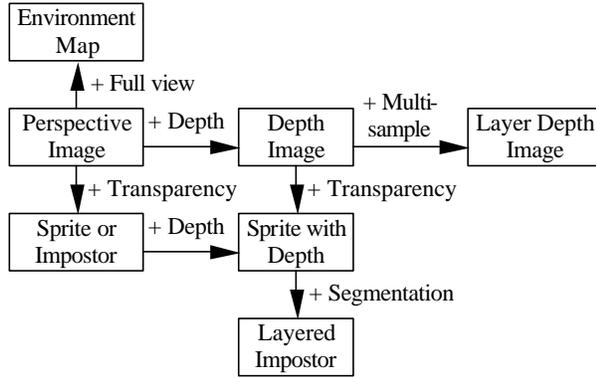


Figure 3: Representations based on perspective images.

Images acquired using a scanner are parallel images. As shown in Figure 2(b), all rays defined in a parallel image have the same direction, which is the viewing direction. Different rays start from different locations on a plane, which we refer to as the camera plane. Assuming that the parametric equation of the camera plane is $\mathbf{F}(x,y)$, then the support function of corresponding rayset representation can be defined as:

$$\begin{cases} \mathbf{S}_x(m,n) = \mathbf{F}_x(m,n) \\ \mathbf{S}_y(m,n) = \mathbf{F}_y(m,n) \\ \mathbf{S}_z(m,n) = \mathbf{F}_z(m,n) \\ \mathbf{S}_q(m,n) \equiv \mathbf{q}_0 \\ \mathbf{S}_f(m,n) \equiv \mathbf{f}_0 \end{cases} \quad (4)$$

where $(\mathbf{q}_0, \mathbf{f}_0)$ specifies the viewing direction, \mathbf{F}_x , \mathbf{F}_y , and \mathbf{F}_z the x-, y-, and z-components of function \mathbf{F} .

A parallel image represents the appearance of a scene viewed from an infinite distance. It samples the scene more evenly, while a perspective image has bias to objects that are closer to the viewpoint. Several representations used in image-based rendering that are derived from parallel images are shown in Figure 3.

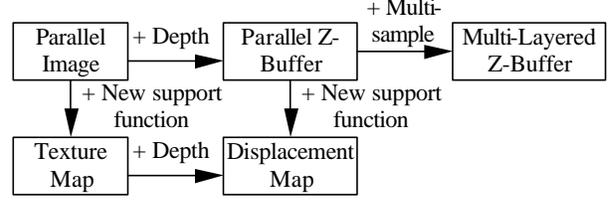


Figure 4: Representations based on parallel images.

It is noteworthy that for texture mapping, the support function used in the rayset rendering stage might be different from the one used in the rayset generation stage. That is, the support function is altered without changing the attribute function so that the desired distortion effect can be generated.

As shown in Figure 2(c), different rays defined in an object image also start from different locations on the camera plane. However, all these rays pass through a point, which we refer to as the object center. Assuming that the parametric equation of the camera plane is $\mathbf{F}(x,y)$, the support function of the corresponding rayset can then be defined as:

$$\begin{cases} \mathbf{S}_x(m,n) = \mathbf{F}_x(m,n) \\ \mathbf{S}_y(m,n) = \mathbf{F}_y(m,n) \\ \mathbf{S}_z(m,n) = \mathbf{F}_z(m,n) \\ \mathbf{S}_q(m,n) = \arctan2(\mathbf{O}_y - \mathbf{F}_y(m,n), \mathbf{O}_x - \mathbf{F}_x(m,n)) \\ \mathbf{S}_f(m,n) = \arcsin(\mathbf{O}_z - \mathbf{F}_z(m,n) / \|\mathbf{O} - \mathbf{F}(m,n)\|) \end{cases} \quad (5)$$

where \mathbf{O}_x , \mathbf{O}_y , and \mathbf{O}_z are the x-, y-, and z-coordinates of the object center.

Object images have been used in image-based rendering research. However, they are sometimes referred to as perspective images, which may cause misunderstanding. For example, in the image-based object approach described in [19], Oliveira and Bishop refer to the object images sharing the same object center as images sharing the same center of projection. Schafler [24] uses the name “back perspective images” to denote object images. Nonetheless, he does not distinguish the difference between the object center and the center of projection.

For clarifications, the differences between an object image and a perspective image are listed below:

- In an object image, all testing rays start from different positions. Therefore, they are actually multiple-center-of-projection images. In a perspective image, however, rays start from the same location, which is the center of projection.
- An object image is normally used to sample the outside of an object. A perspective image is normally used to sample the inside of an environment. Therefore, when only one intersection exists between the testing ray and the scene, object images and

perspective images are sampling different sides of the same surface.

- When multiple intersections exist, a perspective image recodes the attributes of the intersection that is closest to the center of projection. But an object image recodes the attributes of the intersection that is closest to the camera plane, which is often the furthest one from the object center.

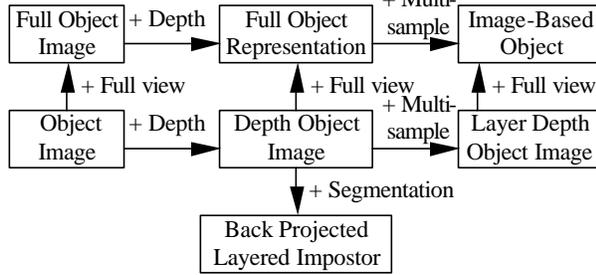


Figure 5: Representations based on object images.

We cannot generate a view using a single object image since all the defined rays start from different locations. However, if we add depth information to an object image, we can reproject it to create new views. As shown in Figure 5, representations based on object images are mainly used to depict an object.

2.3 Other 2-D Raysets

Some image-based rendering approaches employ 2-D representations other than planar projection images, such as cylindrical panorama and images taken by a continuous strip camera. These representations can be regarded as instantiations of 2-D raysets.

A cylindrical panorama can be captured using special devices that rotate around the camera’s optical center. Cylindrical pinhole-camera model is often used to map pixels on a cylindrical panorama to the ray space. To classify cylindrical panorama into 2-D rayset, we simply need to define the support function as follows:

$$\begin{cases} \mathbf{S}_x(u, v) = \mathbf{C}_x \\ \mathbf{S}_y(u, v) = \mathbf{C}_y \\ \mathbf{S}_z(u, v) = \mathbf{C}_z \\ \mathbf{S}_q(u, v) = u \\ \mathbf{S}_f(u, v) = \arctan2(v, f) \end{cases} \quad (6)$$

where, \mathbf{C}_x , \mathbf{C}_y , and \mathbf{C}_z are the x-, y-, and z-coordinates of the center of the projection, f the focus length, which is also equal to the radius of the cylinder.

The concept of the multiple-center-of-projection images [23] is similar to that of the 2-D sampled continuous raysets. Both of them allow different samples to have different viewpoints and both require

the viewing rays to vary continuously across neighboring samples. Nonetheless, they are not exactly the same since the multiple-center-of-projection image defines the mapping between the parameter space and the ray space implicitly using a set of cameras, while 2-D sampled continuous raysets describe the mapping relation explicitly using support functions.

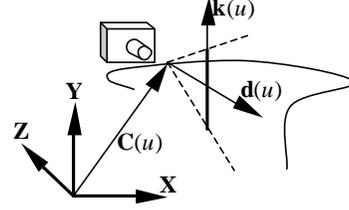


Figure 6: The continuous strip camera model.

Rademacher and Bishop use images taken by a continuous strip camera as examples of multiple-center-of-projection images [23]. Figure 6 shows the continuous strip camera model. Assume that the camera path is defined using a curve, whose function is $\mathbf{C}(u)$, the viewing direction is defined using function $\mathbf{d}(u)$, and the direction of the strip is defined using function $\mathbf{k}(u)$. Then the support function of the corresponding rayset representation can be defined as:

$$\begin{cases} \mathbf{S}_x(u, v) = \mathbf{C}_x(u) \\ \mathbf{S}_y(u, v) = \mathbf{C}_y(u) \\ \mathbf{S}_z(u, v) = \mathbf{C}_z(u) \\ \mathbf{S}_q(u, v) = \arctan2(\mathbf{d}_y(u) + v\mathbf{k}_y(u), \mathbf{d}_x(u) + v\mathbf{k}_x(u)) \\ \mathbf{S}_f(u, v) = \arcsin(\mathbf{d}_z(u) + v\mathbf{k}_z(u) / \|\mathbf{d}(u) + v\mathbf{k}(u)\|) \end{cases} \quad (7)$$

2.4 High-Dimensional Raysets

A 5-D rayset has five parameters. The plenoptic model itself can be expressed as a 5-D rayset. Such a rayset is just a parametric representation of the plenoptic model.

Since the parameter space is introduced, rayset has much more flexibility than the plenoptic model because it makes generating new representations simpler. Using different support functions, we can change the sampling scheme even though the parameter space is always evenly sampled. For example, in the plenoptic model, the starting positions of rays are sampled uniformly throughout the space. In rayset, we can use the following support function (equation 8) to allow areas closer to the origin to be sampled more densely than areas far away.

$$\begin{cases} \mathbf{S}_x(r, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{f}) = r^2 \cdot \cos(\mathbf{a}) \cdot \cos(\mathbf{b}) \\ \mathbf{S}_y(r, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{f}) = r^2 \cdot \sin(\mathbf{a}) \cdot \cos(\mathbf{b}) \\ \mathbf{S}_z(r, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{f}) = r^2 \cdot \sin(\mathbf{b}) \\ \mathbf{S}_q(r, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{f}) = \mathbf{q} \\ \mathbf{S}_f(r, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{f}) = \mathbf{f} \end{cases} \quad (8)$$

The light field function and the lumigraph function use similar ways to parameterize rays using four parameters. Hence, these two representations can be cast as a special instantiation of 4-D continuous rayset.

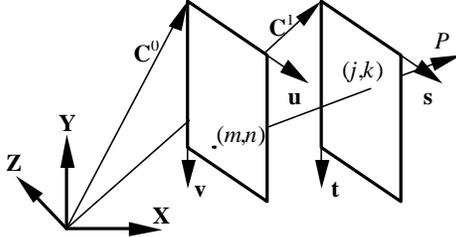


Figure 7: The two planes parameterization used by the light field function.

The light field function and the lumigraph function parameterize rays by their intersections with two planes. Following the notation used in light field rendering, the camera plane is denoted as the UV plane, and the focal plane as the ST plane. As shown in Figure 7, the parameter (m,n,j,k) defines a ray, P , that starts from a point on the UV plane whose local coordinates are (m,n) , and passes through another point on the ST plane whose local coordinates are (j,k) . Suppose the parametric equations of the UV plane and the ST plane are $\mathbf{F}(x,y)$ and $\mathbf{G}(x,y)$, respectively. Then, the support function of the corresponding rayset representation can be defined as:

$$\begin{cases} \mathbf{S}_x(m, n, j, k) = \mathbf{F}_x(m, n) \\ \mathbf{S}_y(m, n, j, k) = \mathbf{F}_y(m, n) \\ \mathbf{S}_z(m, n, j, k) = \mathbf{F}_z(m, n) \\ \mathbf{S}_q(m, n, j, k) = \arctan2 \left(\begin{matrix} \mathbf{G}_y(j, k) - \mathbf{F}_y(m, n) \\ \mathbf{G}_x(j, k) - \mathbf{F}_x(m, n) \end{matrix} \right) \\ \mathbf{S}_f(m, n, j, k) = \arcsin \left(\frac{\mathbf{G}_z(j, k) - \mathbf{F}_z(m, n)}{\|\mathbf{G}(j, k) - \mathbf{F}(m, n)\|} \right) \end{cases} \quad (9)$$

where, \mathbf{F}_x , \mathbf{F}_y , \mathbf{F}_z and \mathbf{G}_x , \mathbf{G}_y , \mathbf{G}_z are the x -, y -, and z -component of function \mathbf{F} and \mathbf{G} , respectively.

From equation 9 we can see that when the parameters m and n are fixed, the function degenerates into the support function of the perspective image. When the parameters j and k are fixed, the function degenerates into the support function of the object image.

Similarly, the spherical light field [12], the uniformly sampled light field [3], and polar parameterized light field [30] are also special kinds of 4-D continuous rayset. The differences between them are the ways the support functions are defined.

Concentric mosaics uses three parameters, radius, rotation angle, and vertical elevation, to parameterize the rays. Therefore, concentric mosaics can be cast as a special kind of 3-D continuous support image.

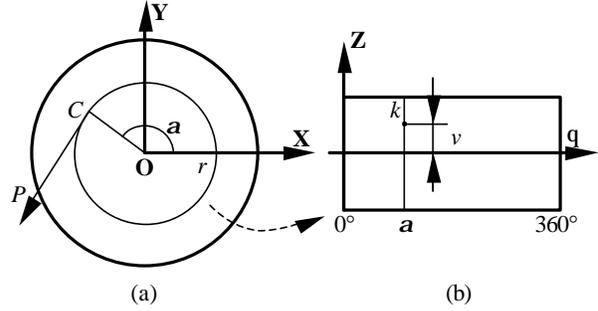


Figure 8: The parameterization used in concentric mosaics.

Assuming that the center of the concentric mosaics is \mathbf{O} , Figure 8(a) shows the projection of concentric mosaics on the X-Y plane and Figure 8(b) shows one concentric mosaic, whose radius is r . The parameter (r, \mathbf{a}, v) defines a ray, P , that starts from a point C on the circle, where the length of OC is r and the angle between X-axis and OC is \mathbf{a} . The direction of ray P is tangent with the concentric mosaic of radius r . The vertical elevation of the projection of ray P on the concentric mosaic is v .

Based on this parameterization method, the support function of the corresponding rayset representation can be defined as:

$$\begin{cases} \mathbf{S}_x(r, \mathbf{a}, v) = \mathbf{O}_x + r \cdot \cos(\mathbf{a}) \\ \mathbf{S}_y(r, \mathbf{a}, v) = \mathbf{O}_y + r \cdot \sin(\mathbf{a}) \\ \mathbf{S}_z(r, \mathbf{a}, v) \equiv \mathbf{O}_z \\ \mathbf{S}_q(r, \mathbf{a}, v) = \mathbf{a} + \mathbf{p}/2 \\ \mathbf{S}_f(r, \mathbf{a}, v) = \arctan2(v, f) \end{cases} \quad (10)$$

where, \mathbf{O}_x , \mathbf{O}_y , and \mathbf{O}_z are the x -, y -, and z -coordinates of the center of the concentric mosaics, and f is the focal length.

From equation 10 we observe that when the parameters r is equal to zero, the function degenerates into the support function of the cylindrical panorama. When the parameter \mathbf{a} is fixed, the function degenerates into the support function of the semi-parallel image. The term semi-parallel image refers to images, in which different strips are parallel to each other, but within the strips the projection is still perspective [21].

3 Object Centered Concentric Mosaics

In most of the cases, when a user walks through a virtual environment, the height of the viewpoint is fixed. Based on this observation, the concentric mosaics approach constrains the camera motion to a horizontal plane and uses manifold mosaics of different radii to depict the scene. It allows a user to move freely in a circular region and to observe lighting changes and horizontal parallax effects. However, vertical distortions still exist. In addition, this approach cannot be used to represent objects.

Based on the taxonomy we discussed above, we know that if we reverse the sampling direction of a perspective image we will get an object image, which describes the appearance of a point under different viewing direction. Following this idea, we can reverse the sampling direction of concentric mosaics to get, what we call, OCCM, which represents the appearance of an object under different viewing directions.

3.1 Parameterization Scheme

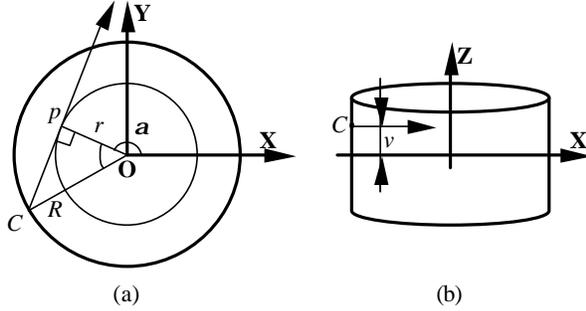


Figure 9: The parameterization used in OCCM.

OCCM uses three parameters, namely, radius, rotation angle, and vertical elevation, to parameterize the rays. However, instead of shooting the rays from planar circles on a plane, all the rays start from a bounding cylinder, which surrounds the object tightly. Moreover, the directions of all testing rays are parallel to the X-Y plane.

Figure 9(a) shows the projection of OCCM on the X-Y plane and Figure 9(b) shows one object centered concentric mosaic. The ray with parameters (r, \mathbf{a}, v) is tangent to a circle of radius r . The angle between the X-axis and Op is \mathbf{a} , where Op is the line connecting the center of the circle and the tangent point. The vertical elevation, v , determines the height of the starting position of the ray.

More concisely, this sampling scheme can be defined using the following support function:

$$\begin{cases} \mathbf{S}_x(r, \mathbf{a}, v) = \mathbf{O}_x + R \cdot \cos(\mathbf{a} + \arccos(r/R)) \\ \mathbf{S}_y(r, \mathbf{a}, v) = \mathbf{O}_y + R \cdot \sin(\mathbf{a} + \arccos(r/R)) \\ \mathbf{S}_z(r, \mathbf{a}, v) = \mathbf{O}_z + v \\ \mathbf{S}_q(r, \mathbf{a}, v) = \mathbf{a} - \mathbf{p}/2 \\ \mathbf{S}_r(r, \mathbf{a}, v) \equiv 0 \end{cases} \quad (11)$$

where, \mathbf{O}_x , \mathbf{O}_y , and \mathbf{O}_z are the x-, y-, and z-coordinates of the geometric center of the bounding cylinder, and R is the radius of the bounding cylinder. The parameter r can be both positive and negative so that both sides of the object can be samples.

3.2 Construction of OCCM

To construct OCCM, we need to sample the continuous support function defined in equation 11. In the original concentric mosaics approach, the angular direction is sampled uniformly. But in the radial direction, the radii are sampled at locations: $0, 1/n^{1/2}, 2/n^{1/2}, \dots, 1$. Using the same way to sample the OCCM will cause the boundary of the object to be over sampled and the center part of the object under sampled. Therefore, in our approach, both the angular direction and the radial direction are uniformly sampled.

The shape of the object to be represented determines the parameters of bounding cylinder. We try to keep the ratio between the number of mosaics and the vertical resolution of the mosaic to be the same as the ratio between the diameter and the height of the bounding cylinder. This restriction helps to achieve similar sampling rate for both the horizontal and the vertical directions. The actual sampling rate needed is determined by the maximum output resolution of the object in novel views.

For each sample ray, both the illumination information and the depth information are sampled. Since the object is surrounded by the bounding cylinder, we know that the depth of intersection must be within the range $[0, (R^2 - r^2)^{1/2}]$. Therefore, the depth is quantified using the following equation:

$$d = \begin{cases} 0 & \text{if no intersection exists} \\ 255 - 254 \times \frac{\text{depth}}{2\sqrt{R^2 - r^2}} & \text{otherwise} \end{cases} \quad (12)$$

Figure 10 shows two examples of OCCM generated for the synthetic Utah teapot model. Figure 10(a) and (b) are the color information and depth information for the object centered concentric mosaic of radius 0, respectively. Figure 10(c) and (d) are those for the object centered concentric mosaic whose radius is equal to half of the bounding cylinder's radius.

To construct OCCM for real objects is also

relatively simple. From equation 11, we notice that when the parameter \mathbf{a} is fixed, the rayset degenerates into a 2-D rayset, W , in which all rays defined are parallel with each other. Since both parameters r and v are uniformly sampled, the only difference between a sampled rayset W and a discrete parallel image is that in W all the rays start from a cylinder, rather than from a camera plane. Since no intersection should exist outside the bounding cylinder, we can always convert a parallel image into rayset W through adjusting the depth value of each pixel. Therefore, the full OCCM can be converted from a set of depth parallel images, whose horizontal directions evenly sample the angular space.

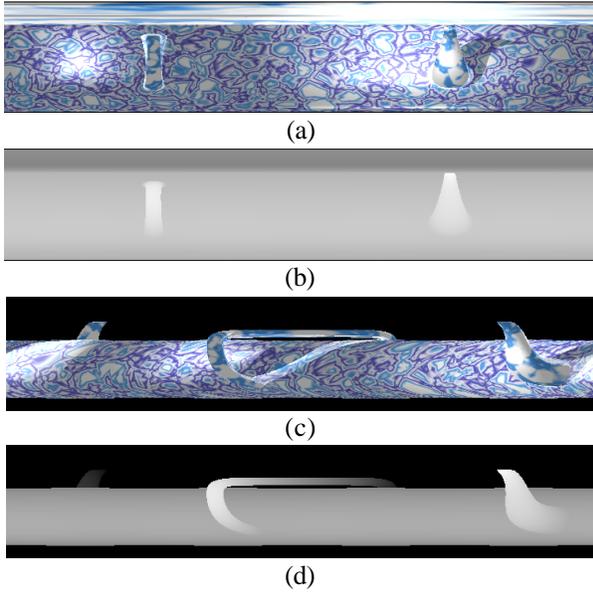


Figure 10: Two OCCM for a teapot model.

The above feature leads to a simple way to construct OCCM for real objects: Put the object on a turntable and rotate the object along a vertical axis. For each angular direction, sample both color and depth information using a fixed parallel projected range scanner. The obtained depth parallel images can then be converted into an object centered concentric mosaic. No resampling process is needed since all we need to do is to reindex the captured pixels.

3.3 Rendering of OCCM

The rendering process contains three steps, which are the backward mapping in the horizontal direction, the forward mapping in the vertical direction, and then, postwarping. The first two steps generate an intermediate image, whose center of projection is the same as the novel view, but its view direction is adjusted to make it parallel to the X-Y plane. In the intermediate image, different pixels in the same column

have the same horizontal viewing direction, i.e. the parameter \mathbf{q} of the corresponding rays are the same. This makes it possible to do the rendering one column at a time.

The backward mapping process is similar to the approaches used in the concentric mosaic approach. Basically, for each column in the intermediate image, we try to find a corresponding column in a pre-sampled concentric mosaic or to bilinearly interpolate between neighboring mosaics. This process is very efficient since the two parameters needed, r and \mathbf{a} , can be calculated directly using the following equations:

$$\begin{cases} r = \sqrt{\|\mathbf{O}' - \mathbf{P}'\|^2 - \frac{\|(\mathbf{O}' - \mathbf{P}') \cdot \mathbf{d}'\|^2}{\|\mathbf{d}'\|^2}} & \text{If } \mathbf{d}'_x (\mathbf{O}'_y - \mathbf{P}'_y) \leq \mathbf{d}'_y (\mathbf{O}'_x - \mathbf{P}'_x) \\ \mathbf{a} = \arctan 2(\mathbf{d}'_y, \mathbf{d}'_x) + p/2 \end{cases}$$

$$\begin{cases} r = -\sqrt{\|\mathbf{O}' - \mathbf{P}'\|^2 - \frac{\|(\mathbf{O}' - \mathbf{P}') \cdot \mathbf{d}'\|^2}{\|\mathbf{d}'\|^2}} & \text{Otherwise} \\ \mathbf{a} = \arctan 2(\mathbf{d}'_y, \mathbf{d}'_x) - p/2 \end{cases}$$

where, $\mathbf{O}' = [\mathbf{O}_x, \mathbf{O}_y]^T$, $\mathbf{P}' = [\mathbf{P}_x, \mathbf{P}_y]^T$, and $\mathbf{d}' = [\mathbf{d}_x, \mathbf{d}_y]^T$ are the projections of the concentric mosaics center, \mathbf{O} , the start position of ray, \mathbf{P} , and the direction of the ray, \mathbf{d} , respectively.

After the corresponding column is found or interpolated, forward mapping is applied within the column. This process takes the position of the novel viewpoint and the depth information as inputs, and warps pixels to new locations to eliminate the vertical distortion. The warping is conducted in the occlusion compatible order [17]. Therefore, no z-buffer is needed. This process is also efficient since only 1-D warping is involved. Details of the 1-D warping process can be found in [20].

Finally, the postwarping process is conducted to apply projective transforms to the intermediate image. The novel view is then produced based on the required viewing direction. Our implementation takes advantage of an efficient scanline algorithm [31]. This process can also be performed through texture mapping and exploit graphics hardware.

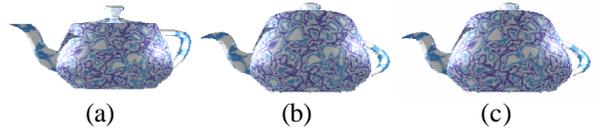


Figure 11: The three steps used in rendering process.

Figure 11 shows the images generated after each step. It shows that the horizontal parallax effect is produced in the backward mapping process (Figure

11a). The vertical parallax effect is introduced in the forward mapping process (Figure 11b). Finally the postwarping process adds the foreshortening effect according to the viewing direction (Figure 11c).

3.4 Experimental Results

The algorithm is tested using a synthetic model, the Utah teapot. The model is represented using Bezier surfaces with cellular textures mapped onto it. Based on the aspect ratio of the bounding cylinder for the teapot, 299 concentric mosaics are generated, with the vertical resolution set as 138. The angular space is sampled at every half a degree, i.e. the horizontal resolution of the concentric mosaics is set as 720.

The raw data size is about 110MB. Similar to the concentric mosaics approach and the light field approach, vector quantization can be applied to reduce the size. However, in our implementation, we chose to use the raw image directly since it is affordable to load them in the memory.

On our Pentium III 733MHz PC with 256MB RAM running Windows 2000, only 0.015 seconds are needed to generate a frame of size 320×240 (the actual object fills the frame).

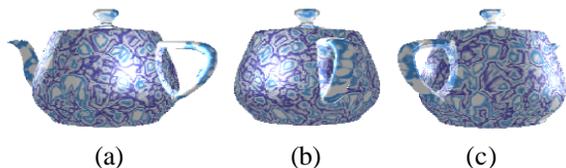


Figure 12: Rendering results for different viewing directions.

Figure 12 shows the rendering results for different viewing directions. It demonstrates both the highlight changes and the reflection changes.



Figure 13: Rendering results for different camera setting.

Figure 13 illustrates the parallax changes for the same viewing direction but under different camera settings. Figure 13(a) is captured using a virtual telescope lens from a larger distance, while Figure 13(b) is captured using a wide-angle lens at a much closer position.

The original concentric mosaics approach only allows the user to move within a planar circular region,

i.e. the user cannot move up and down. Our approach gives the user more freedom in changing the height of the viewing position within the top and bottom boundary of the bounding cylinder. Figure 14 shows the two rendering results with the same horizontal viewing direction but at different heights.

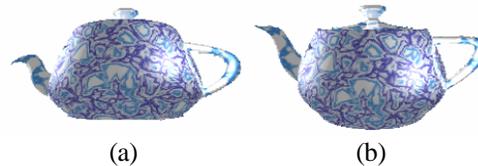


Figure 14: Rendering results for viewpoints at different heights.

4 Conclusion

In this paper, the rayset concept is proposed and a novel image based representation is derived based on this concept.

The rayset is a new concept that can be used to classify existing image-based rendering techniques. Comparing with images, the rayset concept explicitly defines the mapping relation between the parameter space and the ray space. Conventional approaches either ignore this mapping relation or define it implicitly. In the latter case, the mapping relation is usually specified using the type of camera models and some additional parameters. For example, for images taken by a planar pinhole camera, the internal parameters of the camera, such as the focal length and the dimensions of the film, and the external parameters of the camera, such as the center of projection and viewing direction, are used.

Another extension to the conventional images, which is defined on a 2-D parameter space, is that raysets allow the usage of high-dimensional parameter space. Therefore, raysets can be used to generalize approaches like light field, lumigraph, and concentric mosaics.

Comparing with the plenoptic model, rayset separates the mapping from the ray space to the attribute space into two separate mapping relations. The first mapping relation defines how the ray space is sampled, while the second one defines the sampling result. Such a separation introduces flexibilities. For example, keeping the support function constant and changing the attribute function can simulate the effect of changing objects in the scene. On the other hand, changing the support function without altering the attribute function can simulate the effect of distorting the scene.

A texture is essentially a parallel image in the

context of rayset. Then, without altering the attribute mapping, texture mapping can be facilitated by defining a new support function to reorient the rays to conform to the underlying geometric structure of the object. Therefore, under the rayset concept, the approaches that use images as textures are more naturally integrated with other image-based rendering techniques.

A collection of OCCM is a 3-D rayset, which can be used to represent an object. These mosaics are all multiple-center-of-projection images. However, different from multiple-center-of-projection images, all columns in the mosaics are constrained to be vertical. The constraint makes it possible to find the corresponding column in the mosaics directly. This advantage is very important since we cannot afford to reproject all the columns in different mosaics to render a single image.

Comparing with the image-based object approach, OCCM adds one more dimension. Therefore, non-diffuse effect and self-occlusion are captured. In addition, rendering OCCM uses backward mapping plus 1-D warping, which is more efficient than the 3-D image warping used in image-based object.

We can also represent objects in the scene using six relief textures mapped on the faces of a box. Nonetheless, the relief texture approach uses a single image [20]; therefore, it cannot produce non-diffuse effects, such as highlight and reflection.

As a 3-D rayset, OCCM does not capture the appearance of the object on the top and the bottom direction. Therefore, the viewpoint is limited within the height range of the bounding cylinder. However, in many applications, such as to show a new product or to evaluate an architectural model, there is no need to observe the object from the top or the bottom. The OCCM approach is a good choice for these applications since it requires only a reasonable storage and can be rendered in real time.

Acknowledgement

The authors would like to thank financial support provided by NSERC and the University of Saskatchewan. The comments by the reviewers helped to improve the paper, and were much appreciated.

References

- [1] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," In *Computational Models of Visual Processing*, M. S. Landy and A. J. Movshon Ed., The MIT Press, Cambridge, MA, USA, 1991.
- [2] J. F. Blinn, "Simulation of Wrinkled Surfaces," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 286-292, New York, NY, USA, August 23-25, 1978.
- [3] E. Camahort, A. Leros, and D. Fussell, "Uniformly Sampled Light Fields," In: *Rendering Techniques '98: Proceedings of the 9th Eurographics Workshop on Rendering*, European Association for Computer Graphics, pp. 117-130, Vienna, Austria, June 29-July 1, 1998.
- [4] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic Sampling," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 307-318, New Orleans, LA, USA, July 23-27, 2000.
- [5] S. E. Chen, "Quicktime VR - an Image-Based Approach to Virtual Environment Navigation," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 29-38, Los Angeles, CA, USA, August 9-11, 1995.
- [6] S. E. Chen and L. Williams, "View Interpolation for Image Synthesis," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 279-285, Anaheim, CA, USA, August 1-6, 1993.
- [7] M. F. Cohen, M. Levoy, J. Malik, L. McMillan, and S. E. Chen, "Image-Based Rendering: Really New or Deja Vu?" In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 468-470, Los Angeles, CA, USA, August 3-8, 1997.
- [8] R. L. Cook, "Shade Trees," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 223-231, Minneapolis, MN, USA, July 23-27, 1984.
- [9] W. J. Dally, L. McMillan, G. Bishop, and H. Fuchs, "The Delta Tree: An Object-Centered Approach to Image-Based Rendering," Technical Memo 1604, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1996.
- [10] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 11-20, New Orleans, LA, USA, August 4-9, 1996.
- [11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 43-54, New Orleans, LA, USA, August 4-9, 1996.
- [12] I. Ihm, S. Park, and R. K. Lee, "Rendering of Spherical Light Fields," In: *Proceedings of the 5th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, pp. 59-68, Seoul, South Korea, October, 1997.
- [13] S. Laveau and O. D. Faugeras, "3-D Scene Representation as a Collection of Images," In: *Proceedings of the 12th International Conference on Pattern Recognition Vol. 1*, International Association for Pattern Recognition, pp. 689-691, Jerusalem, Israel, October 9-13, 1994.
- [14] M. Levoy and P. Hanrahan, "Light Field Rendering," In: *Computer Graphics Proceedings, Annual Conference*

- Series*, ACM Siggraph, pp. 31-42, New Orleans, LA, USA, August 4-9, 1996.
- [15] D. Lischinski and A. Rappoport, "Image-Based Rendering for Non-Diffuse Synthetic Scenes," In: *Rendering Techniques '98: Proceedings of the 9th Eurographics Workshop on Rendering*, European Association for Computer Graphics, pp. 301-314, Vienna, Austria, June 29-July 1, 1998.
- [16] W. R. Mark, L. McMillan, and G. Bishop, "Post-Rendering 3D Warping," In: *Proceedings of the Symposium on Interactive 3D Graphics*, ACM, pp. 7-16, Providence, RI, USA, April 27-30, 1997.
- [17] L. McMillan and G. Bishop, "Head-Tracked Stereoscopic Display Using Image Warping," In: *Proceedings of SPIE Vol. 2409: Stereoscopic Displays and Virtual Reality Systems II*, International Society for Optical Engineering, pp. 21-30, San Jose, CA, USA, February 7-8, 1995.
- [18] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 39-46, Los Angeles, CA, USA, August 9-11, 1995.
- [19] M. M. Oliveira and G. Bishop, "Image-Based Objects," In: *Proceedings of the Symposium on Interactive 3D Graphics*, ACM, pp. 191-198, Atlanta, GA, USA, April 26-28, 1999.
- [20] M. M. Oliveira, G. Bishop, and D. McAllister, "Relief Texture Mapping," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 359-368, New Orleans, LA, USA, July 23-27, 2000.
- [21] S. Peleg and J. Herman, "Panoramic Mosaics by Manifold Projection," In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, pp. 338-343, San Juan, PR, USA, June 17-19, 1997.
- [22] T. Porter and T. Duff, "Compositing Digital Images," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 253-259, Minneapolis, MN, USA, July 23-27, 1984.
- [23] P. Rademacher and G. Bishop, "Multiple-Center-of-Projection Images," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 199-206, Orlando, FL, USA, July 19-24, 1998.
- [24] G. Schaufler, "Image-Based Object Representation by Layered Impostors," In: *Proceedings of the Symposium on Virtual Reality Software and Technology*, ACM, pp. 99-104, Taipei, Taiwan, November, 1998.
- [25] G. Schaufler & M. Priglinger, "Efficient Displacement Mapping by Image Warping," In: *Rendering Techniques '99: Proceedings of the 10th Eurographics Workshop on Rendering*, pp. 175-186, Granada, Spain, June 21-23, 1999.
- [26] H. Schirmacher, W. Heidrich, & H.-P. Seidel, "High-Quality Interactive Lumigraph Rendering Through Warping," In: *Proceedings of Graphics Interface*, pp. 87-94, Montreal, Quebec, May 15-17, 2000.
- [27] J. Shade, S. J. Gortler, L.-W. He, and R. Szeliski, "Layered Depth Images," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 231-242, Orlando, FL, USA, July 19-24, 1998.
- [28] H.-Y. Shum and L.-W. He, "Rendering with Concentric Mosaics," In: *Computer Graphics Proceedings, Annual Conference Series*, ACM Siggraph, pp. 299-306, Los Angeles, CA, USA, August 8-13, 1999.
- [29] P. P. Sloan, M. F. Cohen, and S. J. Gortler, "Time Critical Lumigraph Rendering," In: *Proceedings of the Symposium on Interactive 3D Graphics*, ACM, pp. 17-23, Providence, RI, USA, April 27-30, 1997.
- [30] G. Tsang, S. Ghali, E. L. Fiume, & A. N. Venetsanopoulos, "A Novel Parameterization of the Light Field," In: *Proceedings of Image and Multidimensional Digital Signal Processing Workshop*, pp. 319-322, Alpbach, Austria, July 12-16, 1998.
- [31] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos, CA, USA, 1990.