Uniform Sampling for Image-based Rendering Shiny Objects

Minglun Gong[1] and Yee-Hong Yang[2]

[1] Department of Math and Computer Science, Laurentian University, Sudbury, ON, Canada
[2] Department of Computing Science, University of Alberta, Edmonton, AB, Canada

Abstract

A novel sampling technique is presented, which can evenly sample all rays that pass through a rectangular surface. Different view directions are uniformly sampled based on the shape of a spiral. Projection positions are then sampled adaptively according to the view direction to provide a constant sampling rate. The experimental results show that our approach can faithfully represent shiny real objects with a reasonable number of samples.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1 Introduction

Image-based rendering (IBR) has gained in popularity in the last decade[7]. Pioneering work in this area has demonstrated the capability of handling the view-dependent appearance of shiny real objects. According to how the rays are parameterized, these approaches can be classified into image-space approaches[1,3,4] and object-space approaches[2,8].

Object-space approaches define the parameterization scheme locally on the surface of the object. The rendering cost depends on the complexity of the geometric model, but the rendering process can be accelerated using graphic hardware. Image-space approaches use a fixed parameterization scheme to sample the 4D ray space and therefore the rendering cost of these approaches is independent of the model complexity.

The approach presented in this paper is an image-space approach. Figure 2 illustrates in 2D how existing image-space approaches sample the appearance of a box-shaped object. Two-plane parameterizations are used in both the light field[4] and the lumigraph[3] approaches. However, in both approaches, if we set the image planes according to the size of the object, some rays (as shown in the figure) that pass through the object will not be sampled. To cover all the samples, the image planes have to be enlarged, which will then introduce useless samples, i.e., rays that do not intersect with the object. In the uniformly sampled light field approach[1], a sphere is used to tightly enclose the object and all rays that pass through the sphere are sampled uniformly. However, for objects with a large length-to-width ratio, using spheres to enclose these objects and then sample the spheres will also generate many useless samples.

For many real world objects, using six or even more rectangles to enclose them can give a tighter bound than using a sphere. Hence, the number of samples required will decrease, if we only sample the rays that pass through these rectangles. Using rectangle as primitive also gives users more flexibility since they can choose to represent only the selected side of the object or of the environment.

Figure 1: *Rendering results with two shiny objects.*

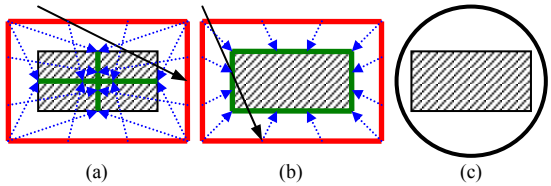


Figure 2: *Sampling used in (a) light field (b) lumigraph (c) uniformly sampled light field. Green lines show the image planes and red lines show the camera plane.*

To sample a given ray space, previous approach[6] shows that great performance can be achieved by sampling the view directions selectively based on their contributions. However, it seems difficult to apply this approach to a complex real object since the view-dependent effects may be visible from all view directions due to the complex lighting and curvature.

The motivation of this paper is therefore to propose a novel sampling technique to uniformly sample rays that pass through a rectangular surface. The key idea is to adaptively sample the image plane according to its projection size. We demonstrate that the new approach can fully capture the view-dependent appearances of shiny objects with a reasonable number of samples.

2 New Sampling Scheme for Rectangular Surfaces

To sample rays that pass through a rectangular surface, we first evenly sample all possible view directions, which are distributed on the hemisphere that covers the rectangle. We then sample the projected positions adaptively according to the view direction to provide a constant sampling rate.

2.1 Uniform Sampling of View Directions

Several previous approaches[1,8] sample different view directions based on polyhedron subdivision. While this technique works well with the adaptive sampling technique to be presented, a new spiral-based sampling approach is used here to facilitate theoretically analyzing the number of samples needed. In addition, the spiral-based sampling approach also has the advantage of providing an arbitrary sampling rate, while the polyhedron subdivision approach increases the number of samples by a factor of four for each subdivision process.

As shown in Figure 3, to sample view directions, we use a spiral curve to cover the upper hemisphere first. Points that are spaced at equal distance along the curve are then used to sample the spiral itself. The parametric function of the spiral can be defined as:

$$\begin{cases} x(\alpha) = \sin(r\alpha)\cos(\alpha) \\ y(\alpha) = \sin(r\alpha)\sin(\alpha) \\ z(\alpha) = \cos(r\alpha) \end{cases}$$

where α is the spiral parameter and r is the parameter that controls the density of the spiral curve.

The above equation defines a spiral that starts from the top of the hemisphere. Any given point on this spiral defines a direction, with the yaw angle to be α and the pitch angle to be $r\alpha$. The pitch between adjacent lines of the spiral is $2r\pi$. The length of the spiral can be computed using the following equation:

$$L(\alpha) = \int_0^{\alpha} \sqrt{(x'(\theta))^2 + (y'(\theta))^2 + (z'(\theta))^2}\, d\theta = E\left(r\alpha, -r^{-2}\right)$$

where $E(\phi,m)$ is the elliptic integral of the second kind:

$$E(\phi, m) = \int_0^{\phi} \sqrt{1 - m\sin^2(\theta)}\, d\theta = \int_0^{\sin(\theta)} \frac{\sqrt{1-mt^2}}{\sqrt{1-t^2}}\, dt$$

To cover the whole hemisphere, the parameter α should be in the range $[0, \pi/(2r)]$. To keep the sampling rate along the spiral the same as that across the spiral, the spiral curve is sampled with a set of points that are spaced at a distance of $2r\pi$. Hence, the total number of samples needed is:

$$n(r) = \left\lfloor \frac{L(\pi/2r)}{2r\pi} \right\rfloor + 1 = \left\lfloor \frac{E\left(\pi/2, -r^{-2}\right)}{2r\pi} \right\rfloor + 1$$

The parameter α for any given sample f can be calculated by solving the equation: $L(\alpha)=f\times 2r\pi$, where $0 \le f < n(r)$. In practice, after the sampling rate r is decided, we pre-compute and tabulate the mapping relation between sample f and parameter α. This introduces little memory overhead since the number of entries in the table is normally small.

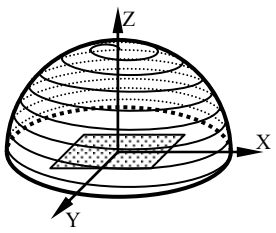


Figure 3: *Use a spiral to sample view directions uniformly.*

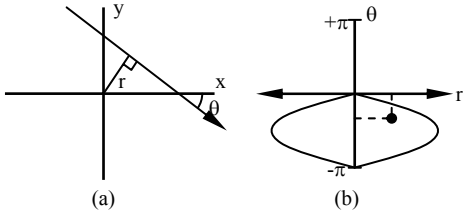


Figure 4: *A ray under (a) the Cartesian space (b) the corresponding line space. In the line space, each ray is represented by a point and each set of rays by a region.*[4]

By construction, when r is small, i.e. the sampling is dense enough, the above sampling scheme gives an equal number of samples per unit solid angle.

2.2 Adaptive Sampling of Projection Positions

We now need to consider how to sample the positions of the rays evenly. We first illustrate the problem in 2D. Assume that we want to sample the set of rays that pass through a line segment on the X-axis. As shown in Figure 4(b), the corresponding region in the line space for this set of rays is the region enclosed by two sine curves. Therefore, under an ideal sampling scheme, the corresponding points of the sampling rays should populate this region uniformly. The two-plane parameterization, which uses a fixed number of samples for different directions, will over-sample the areas that are closer to $\theta=0$ and $\theta=-\pi$. To avoid this problem, we should determine the number of samples needed adaptively according to the length of the segment's projection along different directions.

The extension to 3D is straightforward. To uniformly sample the set of rays that pass through a rectangle on the X-Y plane, we should adjust the number of samples needed according to the area of the rectangle's projection on the plane that is perpendicular to the view direction. To do so, assuming that the rectangle is sampled using $M\times N$ pixels in the perpendicular direction, for any given view direction (x,y,z), the number of pixels we used is $S_xM\times S_yN$, where S_x and S_y are the scale factors in the x- and y-directions, respectively. They are defined as:

$$\begin{cases} S_x = \sqrt{\dfrac{z}{\sqrt{z^2 + x^2y^2}}}\sqrt{1-x^2} \\ S_y = \sqrt{\dfrac{z}{\sqrt{z^2 + x^2y^2}}}\sqrt{1-y^2} \end{cases}$$

The scale factors are defined to provide the following two properties:

- The number of samples used for a given direction is in proportion to the projection size of the rectangle along the direction, i.e.:

$$S_x \times S_y = \frac{z}{\sqrt{z^2 + x^2 y^2}} \sqrt{1 - x^2} \sqrt{1 - y^2} = z$$

- The number of samples along the *x*- or *y*-direction does not change if the projection is perpendicular to the *x*- or *y*-axis, i.e.:

$$S_x = 1 \Leftrightarrow x = 0 \quad S_y = 1 \Leftrightarrow y = 0$$

2.3 Comparison with the Two-Plane Parameterization

In Figure 2 we show that when configuring the light slabs as suggested by the light field and the lumigraph approaches, we have to enlarge the image planes, which will then introduce useless samples. This problem can be solved using a different configuration, i.e., placing the camera plane at infinity. A single light slab can then sample rays that pass the corresponding image plane from the entire hemisphere of directions, and hence, there is no need to enlarge the camera plane. Since such a light slab samples the same ray space as our approach does, here we compare the numbers of samples needed by these two approaches.

Assuming that the surface is sampled using $M \times N$ pixels in the perpendicular direction, the total number of samples required under our approach can be computed by:

$$m = \frac{\int_0^{\pi/2r} MN \times z(\theta) \sqrt{(x'(\theta))^2 + (y'(\theta))^2 + (z'(\theta))^2}\, d\theta}{2r\pi}$$
$$= \frac{M \times N \sqrt{1 + r^2}}{4r^2\pi} + \frac{M \times N}{4\pi} \operatorname{arctanh}\left(\frac{1}{\sqrt{1 + r^2}}\right)$$

With the same sampling density requirement, i.e. the directional space is sampled at $2r\pi$ intervals and the maximum number of samples needed per direction is $M \times N$, the total number of samples required by the two-plane parameterization is:

$$m' = \frac{\pi}{2r\pi} \times \frac{\pi}{2r\pi} \times M \times N = \frac{M \times N}{4r^2}$$

The ratio between the numbers of samples needed by these two parameterization schemes can be calculated by:

$$h = \frac{m'}{m} = \frac{\pi}{\sqrt{1 + r^2} + r^2 \operatorname{arctanh}\left(\frac{1}{\sqrt{1 + r^2}}\right)}$$

The above equation shows that when $r \to 0$, the ratio $h \to \pi$. That is, when the hemisphere is sampled densely enough, the number of samples needed under our approach is $1/\pi$ of the number of samples needed under the two-plane parameterization. In practice, when r is within the range of [1/16, 1/32], the corresponding value of h is within the range of [3.09, 3.13].

3 Experimental Results

A rendering algorithm is implemented using the new sampling scheme. The algorithm also adopts the idea of handling the view-dependent and view-independent appearances of the object separately[5,8]. The new sampling scheme is used for the view-dependent part only. In our rendering algorithm, the view-independent part is rendered using multiple layered-depth images so that the rendering costs for both parts are independent of the object complexity. However, other techniques, such as rendering textured triangle mesh can also be used.

Figure 1 shows an image that our algorithm generates by placing the fish and the elephant captured by Wood et al.[8] in a virtual scene. Due to page limitations, here we only discuss the experiments on the fish dataset in detail. In the experiments, five rectangles are used to model the five sides of the fish, excluding the bottom. According to the aspect ratio of the fish, 270×310 pixels each are used to sample the front and the back, 183×310 each for the left and the right, and 270×183 for the top.

Sampling Rate	Low	High
r	1/16	1/24
# of directions sampled	84	186
Size before compression	20MB	44MB
Size after color compression	6.7MB	14MB
Size after VQ compression	1.7MB	3.5MB

As shown in the table above, two different sampling rates are tested for the view-dependent component. It is noteworthy that the numbers of directions sampled are much less than those used in uniformly sampled light fields[1], where 64K and 245K directions are used. They are also comparable with those used by Lischinski and Rappoport[5], where 66, 258, and 1026 directions are used.

Two compression approaches are also tested. The first one simply compresses the color space from true color to 256 colors. The second applies a 12D vector quantization (VQ). That is, for each rectangle, we generate a codebook with 256 codewords, with each codeword representing the colors of four pixels. The sizes of the datasets before and after compression are also shown in the table.

Please note that, since the view-dependent and view-independent parts are rendered separately, when compressions are applied, only highlights and inter-reflections of the object are affected. Figure 5 shows the rendering results under different compression schemes on an area with both highlights and inter-reflections (on the body of the fish). It suggests that the quality of the image degrades very little after compression.

To evaluate the algorithm, the rendering results are compared with both the original photos and the results reported by Wood et al. To do so, we set the virtual camera to be the same as one of the original photos and also try to match the view used by Wood et al. It is noteworthy that our rendering algorithm uses the resampled rays, and therefore, does not use the original photo directly.

Figure 6 shows the comparison results. It suggests that

using uncompressed high sampling rate, our approach can faithfully reproduce the appearance of the shiny fish. The result is at least comparable with the one generated using uncompressed pointwise faired surface light field, which uses a much larger dataset. We believe that the difference is mainly caused by the new sampling scheme we used. In addition, referring to Figure 5, which shows the tail of the fish at the same view position, we can see that the result generated by our approach using the VQ-compressed dataset preserves the highlights and the inter-reflections better than those generated by surface light field using compressed datasets, even though the datasets used have comparable sizes.

To render the view-dependent part of the fish with output resolution of 240×320 pixels, our current implementation takes about 0.1 sec on a Pentium 4 2GHz PC with 1GB memory running Windows XP.

4 Conclusions

A new way of sampling view-dependent appearance of shiny real objects is presented in this paper — objects are approximated using rectangular surfaces and rays that pass through these rectangles are uniformly sampled.

Compared with representing objects using six light slabs, the new approach has two advantages: (1) it samples rays that pass through the rectangle from all possible directions, hence we can enclose the object tightly using rectangles to reduce useless samples; (2) the rectangles are sampled adaptively according to their projection sizes so that total number of samples needed are reduced.

In addition, for many objects, using six or even more rectangles to enclose them can give a much tighter bound than using spheres. Hence, fewer useless samples exist in our new approach than in the uniformly sampled light field approach.

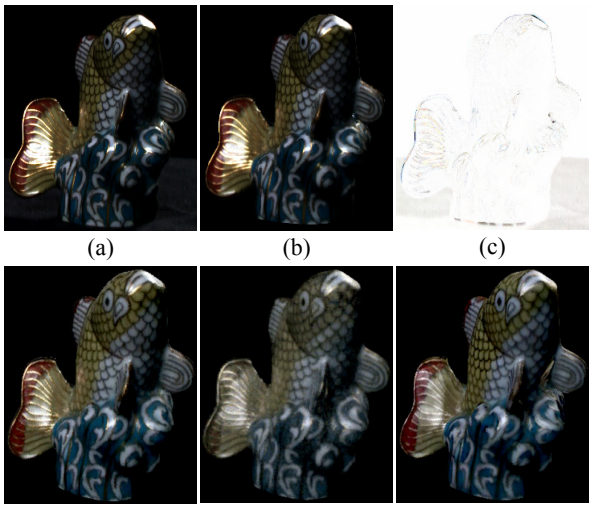
It is noteworthy that there is no redundancy when we use multiple rectangles to approximate an object and then sample these rectangles, as long as these rectangles form a convex solid. This is because even though the datasets for the two adjacent rectangles may contain rays with the same view direction, they do not contain rays with the same projected positions. As a result, using more rectangles may actually decrease the number of samples needed if such a setting can provide a tighter bound for the object.

Besides reducing the number of samples needed, adaptively sampling the projection position also helps to eliminate the aliasing effect. The traditional fixed sampling scheme over-samples the area when the projection direction is close 0 or π. This introduces high frequency information, which will cause aliasing in rendering results.

The two datasets used in surface light field are used in our paper for demonstration. The experimental results suggest that our approach produces acceptable image quality using a small dataset. However, the rendering rate is a little bit slow since our implementation is not optimized. We are currently investigating the use of graphics hardware to accelerate the rendering speed.

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

Figure 5: *Effects of compression: (a) original photo (b) uncompressed dataset (44MB) (c) color compressed dataset (14MB) (d) VQ compressed dataset (3.5MB).*



(a) (b) (c)

(d) (e) (f)

Figure 6: *Results comparison (a) original photo (b) ours using uncompressed high sampling rate dataset (44MB) (c) difference between a and b (d) uncompressed pointwise faired surface light field (177MB) (e) VQ compressed surface light field (8.1MB) (f) surface light field compressed using principal function analysis (2.5MB).*

References

1. E. Camahort, A. Lerios, & D. Fussell. Uniformly sampled light fields. *EGWR*. pp. 117-130. 1998.
2. W.-C. Chen, J.-Y. Bouguet, M. H. Chu, & R. Grzeszczuk. Light field mapping. *SIGGRAPH*. 2002.
3. S. J. Gortler, R. Grzeszczuk, R. Szeliski, & M. F. Cohen. The lumigraph. *SIGGRAPH*. pp. 43-54. 1996.
4. M. Levoy & P. Hanrahan. Light field rendering. *SIGGRAPH*. pp. 31-42. 1996.
5. D. Lischinski & A. Rappoport. Image-based rendering for non-diffuse synthetic scenes. *EGWR*. pp. 301-314. 1998.
6. H. Schirmacher, W. Heidrich, & H.-P. Seidel. Adaptive acquisition of lumigraphs from synthetic scenes. *EUROGRAPHICS*. pp. 151-160. 1999.
7. H.-Y. Shum & S.-B. Kang. Review of image-based rendering techniques. *Visual Communications and Image Processing*. pp. 2-13. 2000.
8. D. Wood, D. Azuma, W. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, & W. Stuetzle. Surface light fields for 3D photography. *SIGGRAPH*. pp. 287-296. 2000.