PAPER *Special Section on Cyberworlds*

A Method for Fast Rendering of Caustics from Refraction by Transparent Objects

Kei IWASAKI†a), ***Nonmember***, **Fujiichi YOSHIMOTO**†, ***Member***, **Yoshinori DOBASHI**††, ***Nonmember***, ***and*** **Tomoyuki NISHITA**†††, ***Member***

SUMMARY Caustics are patterns of light focused by reflective or refractive objects. Because of their visually fascinating patterns, several methods have been developed to render caustics. We propose a method for the quick rendering of caustics formed by refracted and converged light through transparent objects. First, in the preprocess, we calculate sampling rays incident on each vertex of the object, and trace the rays until they leave the object taking refraction into account. The position and direction of each ray that finally transmits the transparent object are obtained and stored in a lookup table. Next, in the rendering process, when the object is illuminated, the positions and directions of the rays leaving the object are calculated using the lookup table. This makes it possible to render refractive caustics due to transparent objects at interactive frame rates, allowing us to change the light position and direction, and translate and rotate the object.

key words: *caustics, global illumination, real-time rendering, graphics hardware*

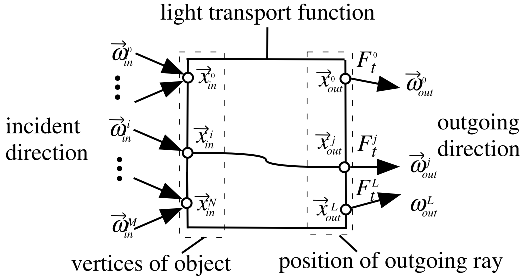


Fig. 1 Light transport function T.

1. Introduction

Realistic image synthesis is one of the most important research fields in computer graphics. Many methods have been developed to achieve this goal. To create realistic images of a scene in which transparent objects are included, it is desirable to render the caustics formed by refracted light from transparent objects. To render refractive caustics, we have to trace the incident light refracted and transmitted through the object and to calculate the illumination distribution of the surface that is illuminated by the refracted rays which finally leave the transparent object. Ray tracing method has been employed for this computation and this is a time-consuming process. However, there are some applications concerning cyber worlds, such as scene simulation, media art, and video games that require real-time rendering of transparent objects, including caustics. We propose an efficient two-pass method to accelerate this rendering.

For each vertex of the transparent object, we relate the direction of the incident light $\vec{\omega}_{in}$ to the position $\vec{x}_{out}$ and direction $\vec{\omega}_{out}$ of the ray that is the refracted and transmitted light from $\vec{\omega}_{in}$ at each vertex through the transparent object, and the total Fresnel transmittance $F_t(\lambda)$ that is the ratio between the energy of the incident light and the outgoing ray $\vec{\omega}_{out}$ (see Fig. 1).

We define the following relationship which we call the light transport function:

$$T(\vec{x}_{in}, \vec{\omega}_{in}) = (\vec{x}_{out}, \vec{\omega}_{out}, F_t(\lambda)), \tag{1}$$

In the proposed method, we calculate the light transport function for the hemisphere of incoming sampled directions at each vertex (see Fig. 3) and store it in a table which we call the *light transport table*. Then in the rendering pass, we use the light transport table to calculate the illumination distribution of caustics on the surface. Our method has the following features.

- Refractive caustics due to transparent objects can be rendered by using the light transport function.
- The light transport function allows the rapid rendering of refractive caustics from objects under rigid transformations with the light changing direction.
- Refraction of the viewing ray through the transparent object also can be calculated by using the light transport function.

In the following, related work on the rendering of caustics is discussed in Sect. 2. In Sect. 3, an overview of our method is presented. In Sect. 4, the precalculation process for computing the light transport table is described. The rendering process for refractive caustics due to transparent objects is mentioned in Sect. 5. In Sect. 6, the method to compress the light transport table is discussed. Finally, the results and conclusion are presented in Sects. 7 and 8, respectively.

Manuscript received September 14, 2004.
Manuscript revised November 30, 2004.

†The authors are with the Faculty of Systems Engineering, Wakayama University, Wakayama-shi, 640–8510 Japan.
††The author is with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060–0814 Japan.
†††The author is with the Graduate School of Frontier Science, the University of Tokyo, Kashiwa-shi, 277–8510 Japan.
a) E-mail: iwasaki@sys.wakayama-u.ac.jp

DOI: 10.1093/ietisy/e88–d.5.904

2. Related Work

In this section, we first review related work on rendering caustics, and then discuss related work on light fields, which are used to perform similar tasks to the light transport functions.

2.1 Rendering Caustics

Several methods have been developed to render caustics. Arvo rendered caustics by using the backward ray tracing method [1]. Shinya et al. developed the grid-pencil tracing method to display caustics, dispersion and extremely bright spots [17]. Heckbert rendered caustics by using adaptive radiosity textures [5], and Mitchell et al. proposed a rendering method for caustics due to curved reflectors [15]. Jensen et al. displayed caustics on non-lambertian surfaces and refractive caustics due to a glass ball by using the photon map method [9], [10]. Brière rendered reflective and refractive caustics by adaptively subdividing light beams [2]. Although realistic images that include caustics can be rendered by using these methods, the computational costs are expensive. Wald et al. proposed a method for rapid rendering with global illumination of images with caustics [20]. This method, however, requires a cluster of dual processor machines. Recently, Gunther et al. presented a real-time rendering method for caustics by using distributed photon mapping [4]. Wyman et al. calculated the irradiance of the caustics photon map and rendered caustics interactively [24]. Both of these methods require several CPUs to render caustics at interactive frame rates. Therefore, these methods are much slower than our method that can achieve the interactive frame rates on a standard PC with only one CPU.

There are several methods that treat the rendering of caustics due to water surfaces [7], [8], [11], [13], [19], [22]. These methods basically take into account only a single refraction of light at the water surface.

Recently, fast rendering methods for caustics using graphics hardware have been developed. Purcell et al. proposed a photon map method that runs on graphics hardware [16]. This method, however, does not achieve real-time rendering. Sloan et al. presented precomputed radiance transfer to achieve real-time rendering of interreflections, soft shadows and caustics [18]. Wand et al. proposed a hardware acceleration method for rendering reflective caustics due to specular objects [21]. These methods consider the rendering of reflective caustics from specular objects and do not deal with refractive caustics from transparent objects.

2.2 Light Field

Recently image-based rendering for the rapid creation of realistic images has emerged. Levoy and Hanrahan proposed a new technique called the light field [12]. The light field represents the reflected light leaving an object at any point and in any direction. The idea of the light field is similar to that of the light transport function. The light fields are created by using pre-acquired imagery from different viewpoints. Gortler et al. developed the Lumigraph [3]. With this, they extended the light field in order to reduce the blur due to resampling by considering the geometric information of the object. Heidrich et al. proposed light field rendering for reflected and refracted light [6]. Wood et al. rendered surface light fields at interactive frame rates [23]. Recently, Masselus et al. developed incident light fields for the relighting of objects [14]. Though the light field rendering is very efficient, the viewpoint or the direction of the light is fixed. Moreover, translation and rotation of the object are not allowed in light field rendering.

The proposed method in this work permits rapid rendering of refractive caustics from moving and rotating objects with changes both in the direction of the light and the viewpoint.

3. Overview

To render refractive caustics due to transparent objects, we have to trace the incident light refracted and transmitted through objects and to calculate the position and direction of rays finally leaving the objects. In that case the illumination distribution of the object surface illuminated by the transmitted light from transparent objects can be calculated. However, ray tracing is a time-consuming task since it requires tests for the intersection between the transparent object surface and the ray. Furthermore, calculation of the illumination distribution is also a complicated task since the refracted light is either converging and diverging. To address this, we propose a rapid two-pass rendering method.

In the first process (the preprocess), we calculate the light transport function that relates the direction of the incident light $\vec{\omega}_{in}$ at vertex $\vec{x}_{in}$, to the outgoing direction $\vec{\omega}_{out}$ from position $\vec{x}_{out}$. To calculate the light transport function for each vertex, we generate sampling rays. The directions of the sampling rays are determined by discretizing the hemisphere over the vertex. We trace the sampling rays until they leave the object. Then the outgoing position, direction and the total transmittance for each sampling ray are obtained and stored in the light transport table.

In the second process (the real-time process), the incident direction of the light at each vertex is computed first. Then the incident direction is used to compute its outgoing direction and position after passing through the transparent object. These are obtained by simply referring to the light transport table using the incident light direction at each vertex. As shown in Fig. 2, by sweeping the refracted outgoing rays calculated using the light transport table, illumination volumes are created. Then the illumination distribution of the caustic is calculated by using the illumination volumes. The areas of intersection between the illumination volumes and the object surface are calculated and the intensities of these are accumulated to render refractive caustics due to the transparent object. The intensity L_P at point P of the in-

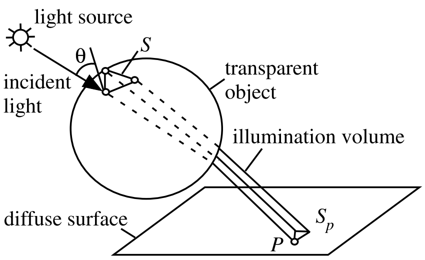


Fig. 2 Illumination volume.

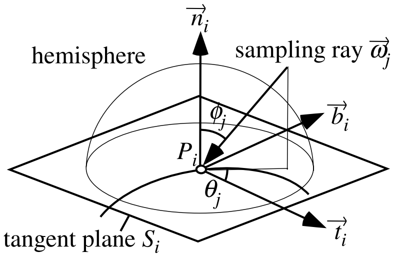


Fig. 3 Local coordinate system at each vertex P_i.

tersection area is calculated by using the following equation (see Fig. 2).

$$L_P(\lambda) = L_{in}(\lambda)\cos\theta F_t(\lambda) F_p \rho_d(\lambda), \qquad (2)$$

where λ is the wavelength, which is sampled for RGB components, $L_{in}(\lambda)\cos\theta$ is the intensity of the incident light onto the triangular mesh of the object surface, and $F_t(\lambda)$ is the total Fresnel transmittance. F_p is the flux ratio and is calculated from the following equation $F_p = S/S_P$, where S_P is the area of the cross section between the illumination volume and the object surface, and S is the area of the triangular mesh on the object surface. $\rho_d(\lambda)$ is the diffuse reflectance of the object.

Refractive caustics are rendered by using the sliced object images proposed by Iwasaki et al. [8]. The combination of the light transport table and the sliced object image achieves rapid rendering of refractive caustics. That is, refractive caustics can be rendered quickly even if the direction of the light changes and the transparent object is translated and rotated since we have calculated the light transport for all directions incident on the object. The proposed method can deal with a parallel light source, a point light source, and a linear light source.

4. Precalculation

In the preprocess, the light transport function is calculated for each vertex and stored in the light transport table. We first calculate the tangent vector $\vec{t}_i$ and the binormal vector $\vec{b}_i$ by using the normal vector $\vec{n}_i$ at the vertex P_i of the transparent object.

The local coordinate system for each vertex P_i is set by using the vectors $\vec{n}_i, \vec{t}_i$ and $\vec{b}_i$ as basis vectors (see Fig. 3). For the direction of the incident light, the sampling rays $\vec{\omega}_j$ are generated over a hemisphere centered at the vertex P_i. The elevation angle between the sampling ray $\vec{\omega}_j$ and the tangent plane S_i is denoted as ϕ_j and the azimuthal angle between the tangent vector $\vec{t}_i$ and the projected vector of $\vec{\omega}_j$ onto the tangent plane S_i is denoted as θ_j (see Fig. 3). The sampling ray $\vec{\omega}_j$ is expressed by the following equation,

$$\begin{aligned}\vec{\omega}_j &= -\cos\phi_j \vec{n}_i - \sin\phi_j \cos\theta_j \vec{t}_i \\ &\quad - \sin\phi_j \sin\theta_j \vec{b}_i. \end{aligned} \qquad (3)$$

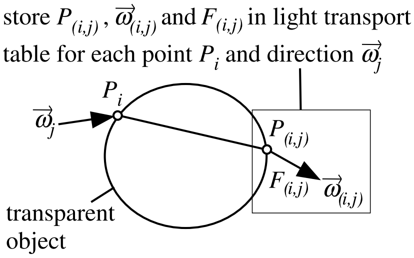

store $P_{(i,j)}$, $\vec{\omega}_{(i,j)}$ and $F_{(i,j)}$ in light transport table for each point P_i and direction $\vec{\omega}_j$
P_i
$\vec{\omega}_j$
$P_{(i,j)}$
$F_{(i,j)}$
$\vec{\omega}_{(i,j)}$
transparent object

Fig. 4 Sampling rays and light transport table.

In the proposed method, the light transport function for each vertex P_i is calculated for each discretized sampling ray $\vec{\omega}_j$. The outgoing position $P_{(i,j)}$ where the incident sampling ray $\vec{\omega}_j$ passing through the vertex P_i finally leaves the object is calculated. At the same time, the direction $\vec{\omega}_{(i,j)}$ of the outgoing ray and the total Fresnel transmittance $F_{(i,j)}$ are also calculated. $P_{(i,j)}$, $\vec{\omega}_{(i,j)}$ and $F_{(i,j)}$ are stored in the light transport table (see Fig. 4). The light transport function is calculated by using the light transport tables for the sampled directions.

For concave transparent objects, some sampling rays are occluded due to the transparent object itself. To compute the sampling rays that are occluded by the object, rays in the opposite direction to the sampling rays are shot. If the ray hits the object, we set a flag that this sampling ray is occluded.

In the proposed method, we only take the refraction of the ray into account to calculate the light transport table, and assume that the viewing ray incident onto the transparent object transmits through the object. Although some sampling rays cannot transmit through the transparent object, we store the intersect point between the ray and the object surface and direction of the ray in the light transport table. These rays are not used for rendering of caustics, but used for the calculation of the refracted viewing ray.

5. Rendering Refractive Caustics

This section describes the rendering process. Illumination volumes are created by using the light transport tables and refractive caustics are rendered by using illumination volumes and sliced object images.

5.1 Creation of Illumination Volume

If the incident light $\vec{L}$ is specified, only the lit triangular

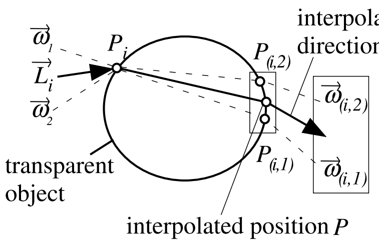


Fig. 5 Interpolation of outgoing ray from object.

meshes are taken into consideration in the following process. The light ray $\vec{L}_i$ in the local coordinate system is calculated for each vertex P^i. In the preprocess, we precompute the position, direction and transmittance of the ray finally leaving the object for a set of sampling rays, and we interpolate to calculate the position, direction and transmission of the ray for the incident light direction $\vec{L}_i$. We find the four closest sampling rays $\vec{\omega}_1, \vec{\omega}_2, \vec{\omega}_3$ and $\vec{\omega}_4$ for the incident light $\vec{L}_i$. We refer to the light transport tables using $\vec{\omega}_j (j = 1, 2, 3, 4)$ and obtain the positions $P_{(i,j)}$, directions $\vec{\omega}_{(i,j)}$ and total transmittances $F_{(i,j)}$. Then we calculate the position P and the total Fresnel transmittance F for the incident light $\vec{L}_i$ by using bilinear interpolation of $P_{(i,j)}$ and $F_{(i,j)}$. The direction $\vec{\omega}$ is calculated by using the quaternion-based vector interpolation (Figure 5 shows the example of the interpolation in 2D). Illumination volumes are created by sweeping the refracted vector $\vec{\omega}$ from P.

In the case of a parallel light source, the incident light $\vec{L}$ is all the same for each vertex P_i in the world coordinate system. Then the direction of the incident light $\vec{L}_i$ in the local coordinate system is calculated for each vertex P_i. For a point light source, the incident light $\vec{L}_i$ is calculated for each vertex P_i in the world coordinate system. Then we calculate the direction of the incident light $\vec{L}_i$ in the local coordinate system. For a linear light source, we replace the linear light souce with a number of point light sources. Then we create illumination volumes for each point light source.

5.2 Rendering Caustics Using Illumination Volumes

The illumination distribution on the diffuse surface is calculated by using sliced object images. The sliced object images are created by setting virtual planes (sample planes) around an object and projecting a part of the object surface between two adjacent sample planes onto the sample plane. Then the illumination distribution of the object surface is approximated by using the illumination distributions of the sample planes.

To calculate the illumination distribution of the sample plane efficiently by using graphics hardware, we create two maps, caustics map and illumination map [8]. The caustics map stores the intensity of the incident illumination distribution on the sample plane. The illumination map represents the reflectance of object in the RGB components. The alpha components of the pixel in the illumination map represents whether the object surface exists or not (see Fig. 6).

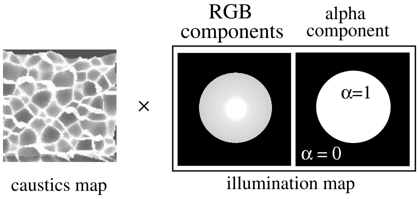


Fig. 6 Caustics map and illumination map.

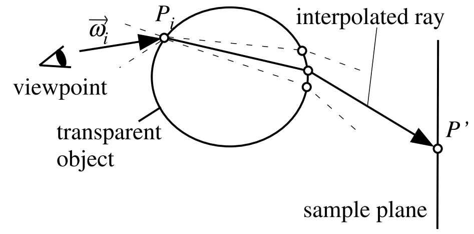


Fig. 7 Intersection calculation between refracted viewing ray and sample plane.

The caustics map is created by drawing the intersection triangles between the illumination volumes and the sample planes, and by accumulating the intensities of the the intersection triangles. To calculate the illumination map, we first set the adjacent sample planes as clipping planes. Then we draw the object surface that exists between two planes by using graphics hardware. The sliced object image is calculated by multiplying the caustics map and the illumination map (see the previous method [8] for more details).

5.3 Rendering Transparent Objects Taking into Account Refraction of the Viewing Ray

To render a transparent object, the refraction of the viewing ray by the transparent object must be taken into account. Let $\vec{\omega}_v$ be the viewing direction from the viewpoint to the vertex P_i (see Fig. 7).

We calculate the azimuthal angle θ and elevation angle ϕ of $\vec{\omega}_v$ in the local coordinate system at P_i. Then the four closest sampling rays for the viewing direction $\vec{\omega}_v$ are calculated by using θ and ϕ. The position from which this viewing ray $\vec{\omega}_v$ finally leaves the object and the direction of the transmitted viewing ray are calculated by referring to the light transport table.

The intersection point P' between the transmitted viewing ray and the sample plane is calculated. The image of object refracted by the transparent object is rendered by mapping the sliced object images onto the transparent object so that point P' corresponds to point P_i (see Fig. 7).

We set the alpha component of a pixel, corresponds to the object surface, at 1, and otherwise we set it at 0 (see Fig. 6). To determine whether the transmitted viewing ray intersects the object surface, the alpha component of the illumination map is checked. That is, the alpha component of the illumination map is used for masking invisible parts of the objects. These operations are performed using the alpha and stencil tests that are accelerated by using graphics hardware.

6. Data Compression

The size of the light transport table is huge, since the table size is approximately proportional to the multiplication of the number of the vertices of the object and the number of sampling rays. The huge data size is inconvenient for the applications such as video games in the cyber world and web shopping because the limitation of the bandwidth of the network and the memory size. To address this problem, we compress the light transport table.

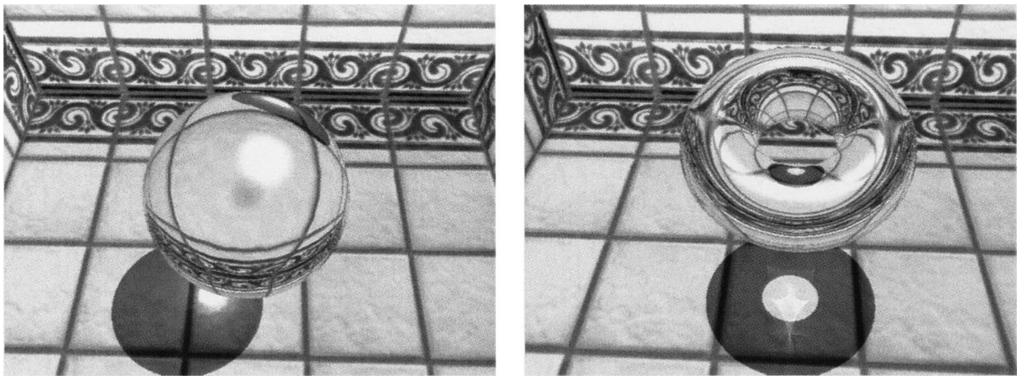
We quantize the position of the outgoing ray. In the proposed method, we quantize the position to 6 bytes (2 bytes for each *xyz* component). We first calculate the bounding box of the transparent object. Let bx_{max} and bx_{min} be the maximum and minimum value of the x component of the bounding box. The x component of the position P, P_x, is quantized to 2 bytes as follows. First, we calculate the following value, $P_q = ((P_x - bx_{min})/(bx_{max} - bx_{min}) \times 2^{16})$. Then we find the nearest integer of P_q and represent the integer as 2 bytes. The y and z components of the position are quantized in the same manner.

We discretize the direction $\vec{\omega}$ of the outgoing ray. The direction $\vec{\omega}$ is represented by the azimuthal angle θ $(0 \leq \theta < 2\pi)$ and the elevation angle ϕ $(0 \leq \phi \leq \pi)$ in spherical coordinate system. Then the θ and ϕ are quantized to 2 bytes, respectively. That is, the direction $\vec{\omega}$ is quantized to 2^{32} directions. The transmittance F_t is also quantized to 2 bytes.

7. Results

Figure 8 shows the examples of refractive caustics due to a glass ball and a torus. In Fig. 8 (a), caustics on the floor can be seen through the transparent glass ball. Refractive caustics due to a glass torus are shown in Fig. 8 (b). Figure 9 shows the examples of refractive caustics due to a glass knot and a glass teapot. This figure indicates that the proposed method can create realistic images of caustics from complex transparent objects. In Fig. 10, we show the examples of caustics under various light conditions. Figure 10 (a) shows the caustics due to the spotlight, and Fig. 10 (b) depicts the result of caustics under three point light sources. Figure 10 (c) shows the example of caustics under a linear light source. In this example, we approximate the linear light source by 8 point light sources. Blurred caustics and shadows due to the linear light source can be seen. Figure 11 shows the examples of refractive caustics due to a colored-glass dolphin and a colored-glass bunny.

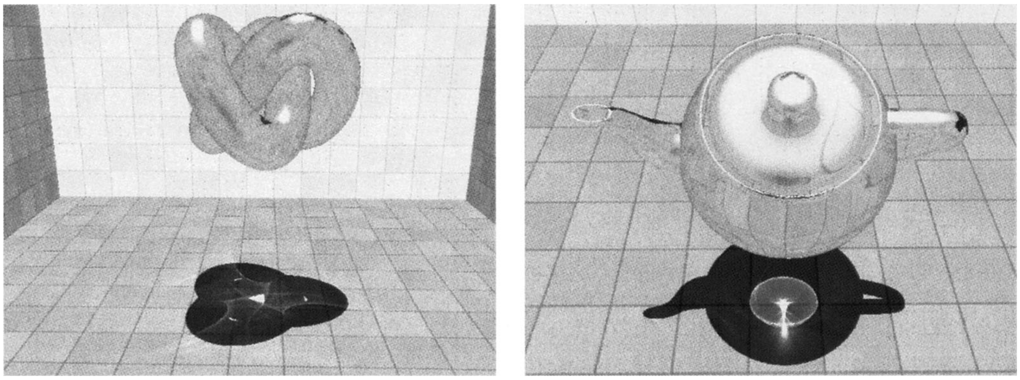
These images were created on a laptop PC (PentiumM



(a) A glass ball

(b) A glass torus

Fig. 8 Refractive caustics due to a glass ball and a torus.



(a) A glass knot

(b) A glass teapot

Fig. 9 Refractive caustics due to a glass knot and a teapot.

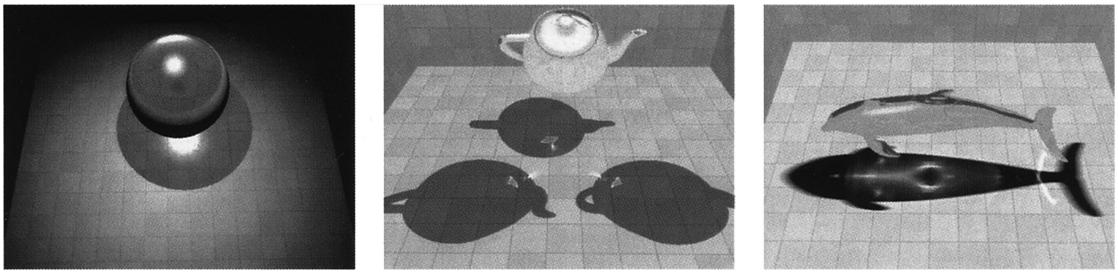

(a) Spot light (b) Three point light sources (c) Linear light source

Fig. 10 Refractive caustics under several light sources.

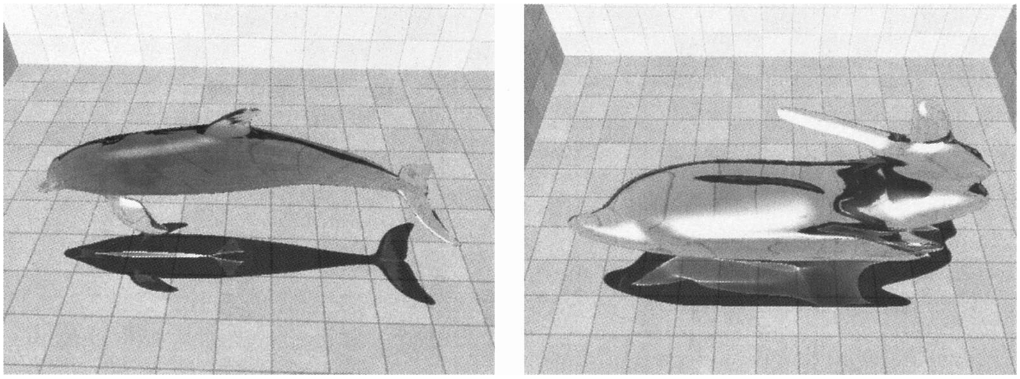
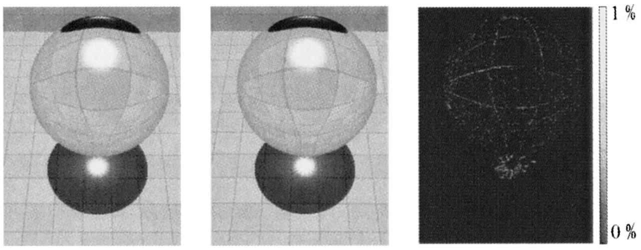

(a) A glass dolphin (b) A glass bunny

Fig. 11 Refractive caustics due to complex colored-glass objects.

Table 1 Rendering frame rates and precomputation times.

Fig. No.	# of vertices	size of light transport table	preproc. time	frame rates
8 (a)	6.4 K	10.6 MB	1.2 hrs	5.1 fps
8 (b)	6.4 K	10.6 MB	1.2 hrs	5.3 fps
9 (a)	13.0 K	21.4 MB	5.1 hrs	2.8 fps
9 (b)	12.8 K	21.1 MB	4.0 hrs	3.0 fps
10 (a)	6.4 K	10.6 MB	1.2 hrs	5.5 fps
10 (b)	12.8 K	21.1 MB	4.0 hrs	1.0 fps
10 (c)	8.9 K	11.6 MB	1.9 hrs	1.3 fps
11 (a)	8.9 K	11.6 MB	1.9 hrs	3.6 fps
11 (b)	5.1 K	8.4 MB	1.0 hrs	7.1 fps



(a) No quantization (b) Quantization (c) Difference image

Fig. 12 Comparison of the image quality.

1.6 GHz, 1 GB main memory) with a nVidia QuadroFX Go700. The image size of these figures is 640 × 480. The number of the sampling rays ω_j for each vertex is 129. Table 1 lists the number of vertices of each model, the size of each light transport table, the calculation time for the preprocess, and the rendering frame rates. Please note that the time for the preprocess is measured on a desktop PC (Pentium4 3.2 GHz), and the time for the rendering is measured on the laptop PC. The number of the sliced object images in all figures except for Fig. 10 (a) is 6. In Fig. 10 (a), the number of the sliced object image is 1. The size of the sliced object images is 512 × 512. The examples shown in this section demonstrate that the proposed method can render transparent objects at an interactive rate.

The computational time of Fig. 8 (a) without the light transport table is 12 seconds. That is, the computational time using the light transport table is about 67 times faster than that without the light transport table. This indicates that the proposed method can render caustics efficiently by using the light transport table. Figure 12 shows the comparison between the image rendered using the light transport table without quantization and the image rendered using the quantized light transport table. Figure 12 (c) shows a difference image of Figs. 12 (a) and (b), where 100% means the difference of the intensity is 255. Figure 12 (c) indicates that the error due to the quantization is less than 1%. The maximum error between the position P of the outgoing ray and the quantized position P_q in Fig. 12 is 0.0032%. The maximum error between the direction $\vec{\omega}$ of the outgoing ray and the quantized direction $\vec{\omega}_q$ is 0.0143%. The light transport table without compression in Fig. 12 is 23.0 MB. On the other hand, the light transport table with compression is

10.6 MB. That is, the compression rate is 45%.

8. Conclusion and Future Work

This paper presented a method for rendering refractive caustics due to transparent objects and rendering images of objects refracted by transparent objects. We proposed a light transport function that describes the light flow through a transparent object. We calculated the light transport function for each vertex in advance and stored it in a light transport table. This makes it possible to very quickly compute the refracted ray that finally leaves an object just by referring to the light transport table. Rapid rendering of refractive caustics can be achieved by using a combination of the light transport table and sliced object images. Since the refraction of the viewing ray also be computed by using the light transport table, an object refracted by a transparent object can be rendered at an interactive rate. We also demonstrated that the proposed method can render refractive caustics under several kinds of light sources such as a parallel light source, point light source, and linear light source.

In future work, we would like to extend the proposed method and apply it to non-rigid transparent objects such as caustics due to water moving in a glass. Since this method takes into account only refraction, reflection of light from transparent objects must also be calculated.

References

[1] J. Arvo, “Backward ray tracing,” Course Note #12 of SIGGRAPH’86, pp.259–263, 1986.

[2] N. Brière and P. Poulin, “Adaptive representation of specular light flux,” Comput. Graph. Forum, vol.20, no.2, pp.149–159, 2001.

[3] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, “The lumigraph,” Proc. SIGGRAPH’96, pp.43–54, 1996.

[4] J. Gunther, I. Wald, and P. Slusallek, “Realtime caustics using distributed photon mapping,” Proc. Eurographics Workshop on Rendering, pp.111–121, 2004.

[5] P.S. Heckbert, “Adaptive radiosity textures for bidirectional ray tracing,” Proc. SIGGRAPH’90, pp.145–154, 1990.

[6] W. Heidrich, H. Lensch, M.F. Cohen, and H.P. Seidel, “Light field techniques for reflections and refractions,” Proc. Eurographics Workshop on Rendering, pp.187–196, 1999.

[7] K. Iwasaki, Y. Dobashi, and T. Nishita, “An efficient method for rendering underwater optical effects using graphics hardware,” Comput. Graph. Forum, vol.21, no.4, pp.701–711, 2002.

[8] K. Iwasaki, Y. Dobashi, and T. Nishita, “A fast rendering method for refractive and reflective caustics due to water surfaces,” Comput. Graph. Forum, vol.22, no.3, pp.601–609, 2003.

[9] H.W. Jensen, “Rendering caustics on non-lambertian surfaces,” Proc. Graphics Interface’96, pp.116–121, 1996.

[10] H.W. Jensen and P.H. Christensen, “Efficient simulation of light transport in scenes with participating media using photon maps,” Proc. SIGGRAPH’98, pp.311–320, 1998.

[11] A. Kunimatsu, Y. Watanabe, H. Fujii, T. Saito, K. Hiwada, T. Takahashi, and H. Ueki, “Fast simulation and rendering techniques for fluid objects,” Comput. Graph. Forum, vol.20, no.3, pp.57–66, 2001.

[12] M. Levoy and P. Hanrahan, “Light field rendering,” Proc. SIGGRAPH’96, pp.31–42, 1996.

[13] T. Nishita and E. Nakamae, “Method of displaying optical effects within water using accumulation-buffer,” Proc. SIGGRAPH’94, pp.373–380, 1994.

[14] V. Masselus, P. Peers, P. Dutre, and Y.D. Willems, “Relighting with 4D incident light fields,” Proc. SIGGRAPH 2003, pp.613–620, 2003.

[15] D. Mitchell and P. Hanrahan, “Illumination from curved reflections,” Proc. SIGGRAPH’92, pp.283–291, 1992.

[16] T.J. Purcell, C. Donner, M. Cammarano, H.W. Jensen, and P. Hanrahan, “Photon mapping on programmable graphics hardware,” Proc. Graphics Hardware 2003, pp.41–50, 2003.

[17] M. Shinya, T. Saito, and T. Takahashi, “Rendering techniques for transparent objects,” Proc. Graphics Interface’89, pp.173–181, 1989.

[18] P. Sloan, J. Kautz, and J. Snyder, “Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments,” Proc. SIGGRAPH 2002, pp.527–536, 2002.

[19] C. Trendall and A.J. Stewart, “General calculations using graphics hardware, with application to interactive caustics,” Proc. Eurographics Workshop on Rendering, pp.287–298, 2000.

[20] I. Wald, T. Kollig, C. Benthin, A. Keller, and P. Slusallek, “Interactive global illumination using fast ray tracing,” Proc. Eurographics Workshop on Rendering, pp.15–24, 2002.

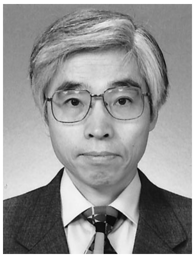
[21] M. Wand and W.Straßer, “Real-time caustics,” Comput. Graph. Forum, vol.22, no.3, pp.610–619, 2003.

[22] M. Watt, “Light-water interaction using backward beam tracing,” Proc. SIGGRAPH’90, pp.377–385, 1990.

[23] D.N. Wood, D.I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D.H. Salsin, and W. Stuetzle, “Surface light field for 3d photography,” Proc. SIGGRAPH 2000, pp.287–296, 2000.

[24] C. Wyman, C. Hansen, and P. Shirley, “Interactive caustics using local precomputed irradiance,” Proc. Pacific Graphics, pp.143–151, 2004.

Kei Iwasaki received the B.S., M.S., and Ph.D. degrees from The University of Tokyo in 1999, 2001, 2004, respectively. He is presently a research associate at the Faculty of Systems Engineering, Wakayama University. His research interests are mainly for computer graphics.



Fujiichi Yoshimoto received the Ph.D. in computer science from Kyoto University, Japan, in 1977. He is presently a professor in the Department of Computer and Communication Sciences, at Wakayama University, Wakayama, Japan. His research interests are in entertainment computing, computer graphics and CAD.

Yoshinori Dobashi received the B.E., M.E., and Ph.D. in Engineering in 1992, 1994, and 1997, respectively, from Hiroshima University. He worked at Hiroshima City University from 1997 to 2000 as a research associate. He is presently an assistant professor at Hokkaido University in the graduate school of engineering, Japan since 2000. His research interests are computer graphics including lighting models.

Tomoyuki Nishita received the B.E., M.E., and Ph.D. degrees from Electrical Engineering from the Hiroshima University, Japan, in 1971, 1973, and 1985, respectively. He worked for Mazda Motor Corp. from 1973 to 1979. He has been a lecturer at the Fukuyama University since 1979, then became an associate professor in 1984, and later became a professor in 1990. He moved to the Department of Information Science of the University of Tokyo as a professor in 1998 and now is a professor at the Department of Complexity Science and Engineering of the University of Tokyo since 1999. His research interests are mainly for computer graphics.