

# Caustic Effect of 3D Transparent Object Using Backward and Forward Ray Tracing

Liza Setyaning Pertiwi, Endang Triwahyuni, Rena Marhaini Simbolon

Department of Mathematics, University of North Sumatera, Indonesia

**Abstract:** This research makes a software which can produce reflective and transparent object using backward and forward ray tracing.. Reflective object will modeled can reflect another object surrounding it. And transparent object will modeled can produce caustic effect, that's rays which refract in one area. So that area will appear brighter than area surround it. The transparent objects in this research are ball, cube, tube, cube. The result that using software Delphi 7 dan OpenGL is showing up caustic effect of transparent object. Using backward and forward ray tracing in this research need average rendering time for two transparent objects (balls) 39717.25 ms and average rendering time for one transparent object (tube) is 16041.75 ms.

**Keywords:** transparent, backward ray tracing, forward ray tracing.

## 1. Introduction

Today computer graphics is used in many aspects, rendering method for making realistic pictures advances. One of rendering method that often uses is ray tracing. By ray tracing method, rendering picture result is according to the original because it shows up the reflecting effect such as ambient, diffuse dan specular.

Ray tracing method is distinguished as forward ray tracing and backward ray tracing. Forward ray tracing is browsing light which is emitted from light source, and backward ray tracing is browsing light which emitted from camera or eye. Forward ray tracing method can results the picture using caustic effect but it needs long time for rendering because of so many lights emission from light source and not all of the lights hit object or forward to camera. Because of that, it needs backward ray tracing. The software for visualization is OpenGL and Delphi 7.

## 2. Reflection and Refraction of Light

### 2.1 Reflection of Light

Rays coming on an object's surface will be reflected at an angle equal to the normal in the subject field. Overview of beam coming and reflected ray can be seen in Figure 1. In simple terms it can be said with equal angle between  $\vec{L}$  and  $\vec{N}$ , the angle between the  $\vec{N}$  and  $\vec{R}$ .

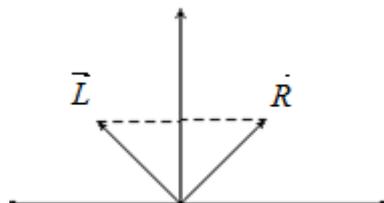


Figure 1: Formation Reflected Ray

To find the direction of the reflected beam can be formulated (Whitted, 1998) as in equation 1.

$$\vec{R} = 2\vec{N}(\vec{N} \cdot \vec{L}) - \vec{L} \quad (1)$$

$\vec{R}$  = sinar pantul,  
 With,  $\vec{R}$  = reflected light,  
 $\vec{N}$  = normal,  
 $\vec{L}$  = negation of light coming.

### 2.2 Refraction of Light

Rays on the transparent objects will be refracted. Results of refraction depends on the density of the material object. Refracted ray is calculated by reference to the normal vector of the object as shown in Figure 2. The beam b from the medium one, the speed is c1 and forming the angle of incidence  $\theta_1$  of the normal line, on and pass an object with two medium, the speed of light in the medium two are c2 and refraction angle formed between the reflected normal light is  $\theta_2$ , the Snell's law (Halliday, 1960) is expressed in the following equation:

$$\frac{\sin(\theta_1)}{c_1} = \frac{\sin(\theta_2)}{c_2} \quad (2)$$

c is 1:n so the term can be writtens

$$\sin(\theta_1) * n_1 = \sin(\theta_2) * n_2 \quad (3)$$

with,  $\theta_1$  and  $\theta_2$  = angle of incidence and angle of reflection,  
 $c_1$  and  $c_2$  = speed of light in the medium of origin and destination,  
 $n_1$  and  $n_2$  = the refractive index of the medium of origin and destination ray.

To find the direction of the rays that have been refracted (d), as shown in Figure 2 is obtained by the equation as the following equation:

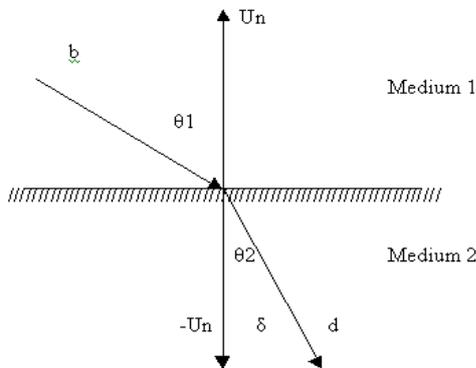
$$d = \delta c + (1 - \delta)(-U_n) \quad (4)$$

$$\delta = \frac{1}{\sqrt{\frac{c_1^2}{c_2^2} * \|c\|^2 - \|c + U_n\|^2}} \quad (5)$$

$$c = \frac{b}{\|b \cdot U_n\|} \quad (6)$$

with, d = reflected ray,

$\delta$  = the distance between the light refracted by the normal line,  
 $U_n$  = normal field,  
 $c_1$  = speed of the medium 1,  
 $c_2$  = on medium speed 2, and  
 $b$  = light that comes on an object surface.



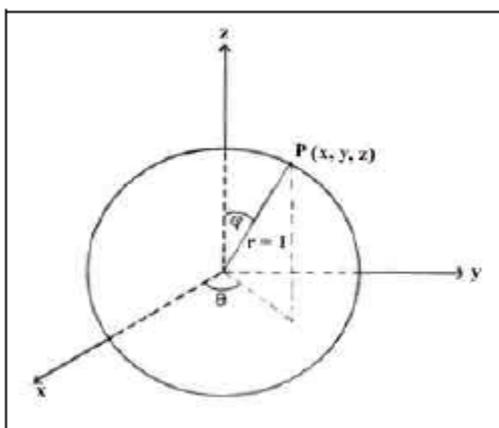
**Figure 2:** Refraction rays of medium 1

### 3. Forward and Backward Ray Tracing

Ray tracing is a method of rendering an image by tracing rays on the object. If the traced rays of an object and the calculated intensity of the color of the object. Ray tracing method is divided into two types, namely forward and backward ray tracing ray tracing. Forward ray tracing is used to obtain the intensity of the beam subjected to many objects. With this method obtained caustic effect of the transparent object. While in backward ray tracing, rays are traced rays on the eye.

#### 3.1 Forward Ray Tracing

On the forward ray tracing, ray traced rays are emitted from the light source. Forward ray tracing for illumination phase, the phase calculation of rays on the object. So that the search was not long then traced rays are limited in number. The light source used in this study is in the form of a point light source. Light source is modeled as a sphere with a radius of one and a center point position of the light source. Sought direction from the center point of the beam across the surface of the ball. This can be seen in Figure 3.



**Figure 3:** Finding the direction of the beam from the light source

As for how to get a point on the surface of the ball used equation 7:

$$P(u, v) = (r * \cos(v) * \cos(u), r * \cos(v) * \sin(u), r * \sin(v)) \quad (7)$$

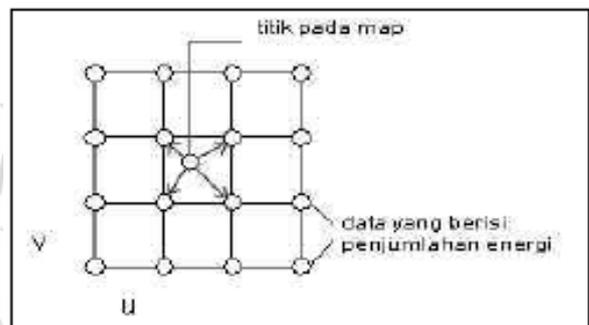
in which;

$r$  = radius of the ball (value of 1)

$u$  = angle in the direction of the horizontal axis is worth between  $0 \leq u < 2\pi$

$v$  = angle in the direction of the vertical axis is worth between  $-\pi/2 \leq v < \pi/2$ .

Mapping (Arvo 1986) storing energy from light in a 2D array. Because the form of a 3D object, then converted first to form  $P(u, v)$  it. After that distribute the energy to the point - the point around, is bilinear, as shown in Figure 4.

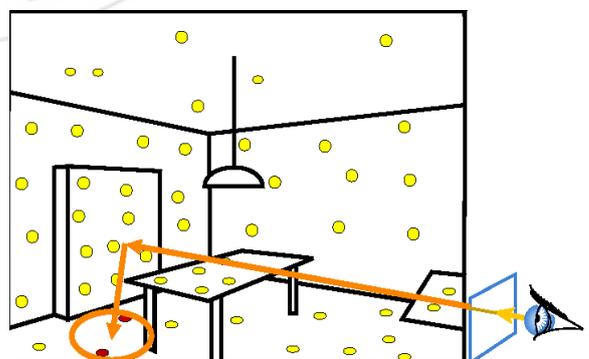


**Figure 4:** Mapping the beam on the object surface

To be distributed to four points surrounding the first distributed to two points on each line parallel to the two end points, then distribute it to two points. Map that stores energy on the surface of an object called the illumination map. Obtained by menbagi energy intensity in illumination map to the area represented (in this study is used widely ball).

#### 3.2 Backward Ray Tracing

Once the forward ray tracing process is complete, then proceed with the second phase by using backward ray tracing algorithm. Picture of photon mapping in the backward calculation ray tracing can be seen in Figure 5.



**Figure 5:** Backward ray tracing with photon mapping

The data structure used to store the beam is kd-tree data structure (k-dimensional tree) to get the render time. Photon data is broken down by position, the division was based on one axis (X, Y or Z) (Tin-Tin et al, 2005). The model of the effect of the bias shown in the following equation

$$c = \frac{1}{r^2} \sum_{i=1}^M (\max(\bar{n} \cdot \bar{l}_i, 0) * k_d * color + \max(\bar{n} \cdot \bar{h}_i, 0)^a * k_s * color) * energy_i \quad (8)$$

with,

$c$  = the effect of the bias light collection at the point  $P$ ,

$r$  = radius of the sphere  $S$ ,

$M$  = number of photons that are found in the sphere  $S$ ,

$\bar{n}$  = Normal vector at point  $P$ ,

$\bar{l}_i$  = Negation photon direction  $i$ ,

$\bar{h}_i$  = Vector of the direction of refraction with sightings

photon direction  $i$ ,

$a$  = exponent specular sharpness,

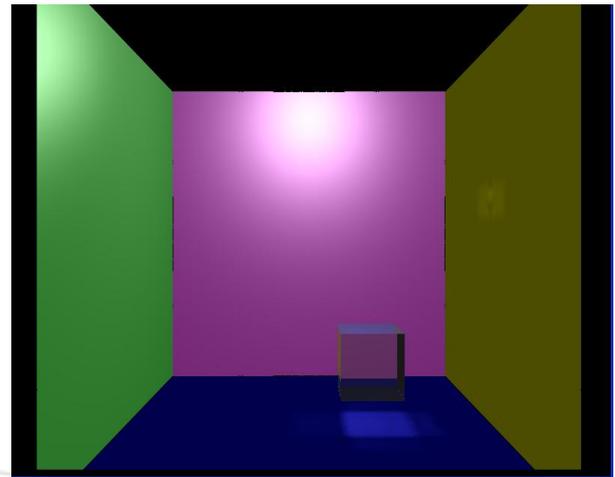
$kd$  = coefficient of diffuse,

$ks$  = coefficient of specular,

$color$  = color of the surface at the point  $P$ ,

$energy_i$  = photon energy level  $i$ , and

$\max(a, b)$  = back to the maximum value of  $a$  and  $b$  (Cheah, 1996).



**Figure 6:** Refracted ray effect with 1 cube

While the algorithm is used as follows:

- 1) for each light source
- 2) find the direction of the beam
- 3) looking for a hit-ray point with all objects
- 4) If the object crashing
- 5) If on the reflective object
- 6) find light reflections
- 7) The plot of the beam refracted parameters
- 8) If on the transparent object
- 9) looking refractive beam
- 10) The plot of the refractive beam parameters

### 3.3 Implementation

The trial of the software that was created by using Delphi 7 and OpenGL aims to determine whether it is appropriate output. For testing, the data - the data is organized as follows:

- Position Camera (0,300,900), view (0,300,0)
- Position Camera (0,300,900), view (0,300,0)
- The color combination of pink background set with 0.3, 0.3 and blue green 0.3, yellow 1.0
- Lamp with ambient coefficient (0.5, 0.5, 0.5), diffuse and specular coefficient (0.8 0.8 0.8). Lights are put in the position (0, 19, 0). Energies 20.
- Flat Field with normal (0, 1, 0), the distance from the plane  $y = -5$  0. For ambient coefficient 0.7, 0.2 diffuse, specular 0.8, reflectivity and transparency 0 0,
- The cube with the center point (2, 1, -1), finger - finger 4. ambient coefficient 0.3, 0.9 diffuse, specular 0.2, the reflectivity of 0.2, 0.7 transparency and refractive index 1:25. The cube is not textured and colored (0, 1, 0).
- Scale folder (the square footage of the real world is represented by each unit area of the illumination map) on a horizontal plane is 2.

With the data - the data mentioned above, it is obtained as shown in figure 6.

### 4. Conclusion

- 1) Results of modeling the caustic effects of light depends on the intensity of the light source, the position and the refractive index of the object.
- 2) Effect of kd-tree data structure on which the software will be built is accelerate rendering time for storing photons on more than one leaf.

### References

- [1] Arvo, J. 1986. Backward Ray Tracing. Apollo Computer, Inc. Chelmsfort, MA, <http://graphics.stanford.edu/courses/cs348b-97/basics/intersection/slides/>.
- [2] Cheah, Shu Chiun. 1996. An Implementation Of A Recursive Ray Tracer That Renders Caustic Lighting Effects. Dept. of Computer Science, University of Maryland, <http://www.cs.umd.edu/~mount/Indep/Scheah/causticpaper.html>.
- [3] Halliday, Resnick; Krane. 1960. Physics 4<sup>th</sup> Edition Volume 2. John Willey & sons.
- [4] Purcell, Varberg. 1987. Calculus with Analytic Geometry 5<sup>th</sup> Edition. Prentice-Hall.
- [5] Whitted. 1998. Basics of Ray Tracing. Lecture notes for Spring Quarter.
- [6] Woo, M., Neider, J. dan Davis, T. 1997. OpenGL Programming Guide Second Edition. Silicon Graphics.
- [7] Wright, Right S. 1996. OpenGL-Super Bible. San Francisco: Waite Group Press.
- [8] Yu, Tin-Tin., Lowther, John. dan Shene, Ching-Kuang. 2005. Photon Mapping Made Easy. Houghton: Department of Computer Science Michigan Technological University, (<http://www.cs.mtu.edu/~shene/PUBLICATIONS/2005/photon.pdf>).