

**EFEK PENGUMPULAN CAHAYA BIAS PADA OBYEK TRANSPARAN 3D
MENGUNAKAN METODE BACKWARD RAY TRACING
DAN PHOTON MAPPING**

Liza Setyaning Pertiwi¹

ABSTRAK

Penelitian ini merancang suatu perangkat lunak yang mampu menghasilkan efek realistik pada obyek-obyek transparan dengan menggunakan metode backward ray tracing dan photon mapping. Obyek transparan yang dimodelkan pada penelitian ini berupa bola, kotak, tabung, dan kerucut. Hasil penelitian yang menggunakan software Delphi 7 dan OpenGL ini adalah pemunculan efek pembiasan sinar pada obyek transparan dari sumber cahaya yang mengumpul di suatu daerah sehingga pada daerah tersebut akan tampak lebih terang daripada daerah sekitarnya. Penggunaan gabungan metode backward ray tracing dan photon mapping memerlukan waktu render rata-rata untuk dua obyek transparan (bola) yaitu 39717.25 ms dan waktu render rata-rata untuk satu obyek transparan (tabung) yaitu 16041.75 ms.

Kata kunci : transparan, photon mapping, backward ray tracing.

¹ Liza Setyaningsih Pertiwi, Mahasiswa S2 Matematika, FMIPA, Universitas Sumatera Utara

1 Pendahuluan

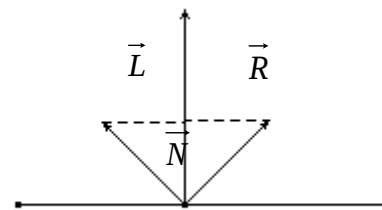
Dalam dunia komputer grafis saat ini, metode rendering untuk menghasilkan gambar-gambar realistik telah berkembang. Salah satu metode yang sering dipakai adalah ray tracing. Dengan metode ray tracing, maka hasil rendering gambar yang dihasilkan sesuai dengan aslinya karena memperhitungkan efek-efek pencahayaan seperti ambient, diffuse dan specular.

Metode ray tracing dibedakan menjadi dua, yaitu forward ray tracing dan backward ray tracing. Forward ray tracing adalah metode penelusuran cahaya yang dipancarkan dari sumber cahaya, sedangkan metode backward ray tracing menelusuri cahaya yang dipancarkan dari kamera atau mata. Metode forward ray tracing dapat menghasilkan gambar yang memiliki efek pengumpulan cahaya bias. Akan tetapi metode forward ray tracing tersebut memerlukan waktu render yang sangat lama. Hal ini disebabkan banyaknya sinar yang dipancarkan sumber cahaya dan tidak semua sinar tersebut mengenai obyek atau menuju ke arah kamera. Oleh karena itu, diperlukan kombinasi dengan metode lain. Pada fase pertama, sinar yang dipancarkan sumber cahaya dan mengenai obyek akan dipetakan ke dalam photon mapping. Fase kedua dilakukan dengan menggunakan algoritma backward ray tracing. Adapun software aplikasi untuk memvisualisasikan hasil penelitian yaitu OpenGL dan Delphi 7.

2 Pemantulan dan pembiasan cahaya

2.1 Pemantulan Cahaya

Sinar yang datang mengenai suatu permukaan obyek akan dipantulkan dengan sudut yang sama besar terhadap normal pada bidang yang dikenai. Gambaran dari sinar datang dan sinar pantul dapat dilihat pada Gambar 1. Secara sederhana dapat dikatakan sudut antara $\vec{L}$ dengan $\vec{N}$ sama besar dengan sudut antara $\vec{N}$ dengan $\vec{R}$.



Gambar 1. Pembentukan sinar pantul

Untuk mencari arah sinar pantul dapat dirumuskan (Whitted, 1998) seperti pada persamaan 1.

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

dengan,

- $\vec{R}$ = sinar pantul,
- $\vec{N}$ = normal,
- $\vec{L}$ = negasi dari sinar datang.

2.2 Pembiasan Cahaya

Sinar yang mengenai benda transparan akan dibiaskan. Hasil pembiasan tergantung pada kerapatan material obyek. Sinar bias dihitung dengan acuan vektor normal dari obyek seperti yang terlihat pada Gambar 2. Sinar b yang berasal dari medium satu, kecepatannya adalah c_1 dan membentuk sudut

datang θ_1 terhadap garis normal, mengenai dan melewati obyek dengan medium dua, kecepatan sinar pada medium dua adalah c_2 dan sudut pembiasan yang terbentuk antara sinar bias dengan normal adalah θ_2 , maka hukum Snell (Halliday, 1960) dinyatakan pada persamaan berikut:

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

c adalah satu banding n sehingga persamaan diatas dapat ditulis:

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

dengan, θ_1 dan θ_2 = sudut datang dan sudut pantul,
 c_1 dan c_2 = kecepatan sinar di medium asal dan tujuan,
 n_1 dan n_2 = indeks bias medium asal dan tujuan sinar.

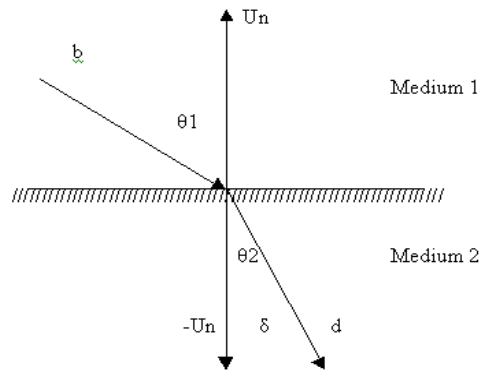
Untuk mencari arah sinar yang sudah dibiaskan (d), seperti yang terlihat pada Gambar 2 didapat dengan persamaan seperti pada persamaan berikut:

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

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

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

dengan, d = sinar bias,
 δ = jarak antara cahaya yang dibiaskan dengan garis normal,
 U_n = normal bidang,
 c_1 = kecepatan pada medium 1,
 c_2 = kecepatan pada medium 2, dan
 b = sinar datang yang mengenai suatu permukaan obyek.



Gambar 2 Pembiasan sinar dari medium 1

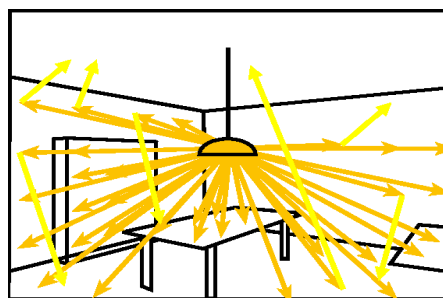
3 Photon Mapping dan Backward Ray Tracing

Photon mapping menyimpan photon-photon pada photon map dari penelusuran titik yang dikenai sinar, sedangkan backward ray tracing adalah suatu metode untuk menelusuri sinar yang dipancarkan dari kamera atau mata.

3.1 Photon Mapping

Prinsip dasar dari photon mapping adalah menyimpan sinar yang dipancarkan dari sumber cahaya ke dalam photon map (Tin-Tin dkk, 2005). Sinar yang disimpan

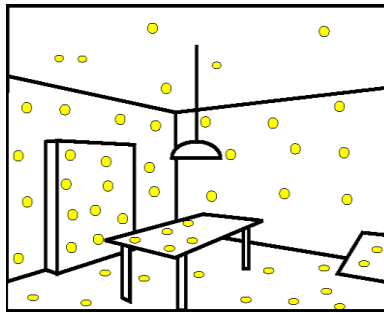
hanya sinar yang menabrak obyek bukan transparan dan merupakan sinar hasil dari pembiasan oleh obyek transparan. Penelusuran sinar pancaran dilakukan untuk setiap sumber cahaya. Tipe pemancaran sinar yang digunakan adalah memancarkan sinar ke segala arah dari suatu titik pusat, yaitu titik posisi sumber cahaya tersebut. Gambaran pemancaran sinar dapat dilihat pada Gambar 3.



Gambar 3. Penembakan sinar dari sumber cahaya

Sinar-sinar yang menabrak obyek disimpan posisi tabrak dan intensitas cahayanya. Jika menabrak obyek transparan maka sinar akan dibiaskan. Arah sinar bias dihitung dengan menggunakan teori

pembiasan cahaya, yaitu hukum Snell dan penelusuran sinar diteruskan sampai menabrak obyek yang tidak transparan. Hasil pemetaan dapat dilihat pada Gambar 4.

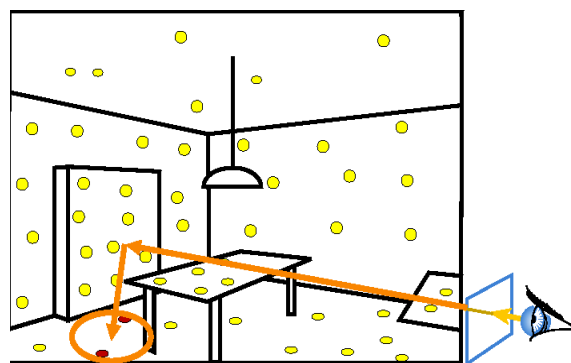


Gambar 4 Penyimpanan sinar

Efek pengumpulan cahaya bias dimodelkan dengan menggunakan algoritma photon mapping. Efek itu muncul dengan adanya sinar-sinar yang dibelok sehingga akhirnya menabrak di titik berdekatan, atau bahkan banyak sinar yang dibiaskan ke titik yang sama.

Setelah proses photon mapping selesai, maka dilanjutkan dengan fase kedua yaitu dengan menggunakan algoritma backward ray tracing. Gambaran perhitungan photon mapping dalam backward ray tracing dapat dilihat pada Gambar 5

3.2 Backward Ray Tracing



Gambar 5 Backward ray tracing dengan photon mapping

Struktur data yang digunakan untuk menyimpan sinar adalah struktur data kd-tree (k-dimensional tree) untuk mendapatkan waktu render. Data-data photon tersebut dibagi

berdasarkan posisinya, pembagian itu didasarkan pada salah satu sumbu (X,Y atau Z) (Tin-Tin dkk, 2005). Adapun model efek bias tampak pada persamaan berikut.

$$c = \frac{1}{r^2} \sum_{i=1}^M (\max(\bar{n} \cdot \bar{l}_i, 0) * k_d * color + \max(\bar{n} \cdot \bar{h}_i, 0)^a * k_s * color) * energy_i \quad (7)$$

dengan,

- c = efek pengumpulan sinar bias di titik P,
- r = radius dari bola S,
- M = jumlah photon yang ditemukan pada bola S,
- $\bar{n}$ = vektor normal di titik P,
- $\bar{l}_i$ = negasi arah photon i,
- $\bar{h}_i$ = vektor antara arah pembiasan dengan arah penampakan photon i,
- a = eksponen ketajaman specular,
- k_d = koefisien diffuse,
- k_s = koefisien specular,
- color = warna permukaan di titik P,
- energy_i = tingkatan energi photon i, dan
- max(a,b)= kembali pada nilai maksimum dari a dan b (Cheah, 1996).

Sedangkan algoritma yang digunakan sebagai berikut:

1. untuk setiap sumber cahaya
2. cari arah pancaran sinar
3. cari titik tabrak sinar dengan semua obyek
4. jika menabrak obyek maka
5. jika mengenai benda reflektif maka
6. cari sinar pantulannya
7. petakan dengan parameter sinar pantulnya
8. jika mengenai benda transparan maka
9. cari sinar biasnya
10. petakan dengan parameter sinar biasnya

3.3 Implementasi Algoritma

Uji coba terhadap perangkat lunak yang telah dibuat dengan menggunakan Delphi 7 dan OpenGL ini bertujuan untuk mengetahui apakah output nya sudah sesuai. Untuk uji coba, data – data diatur sebagai berikut:

- Posisi Camera (0,300,900), view (0,300,0)
- Posisi Camera (0,300,900), view (0,300,0)

- Warna background diatur dengan kombinasi pink 0.3, hijau 0.3 dan biru 0.3,kuning 1.0
- Lampu dengan koefisien ambient (0.5, 0.5, 0.5), koefisien diffuse dan specular (0.8 0.8 0.8). Lampu diletakkan di posisi (0, 19, 0). Energinya 20.
- Bidang datar dengan normal (0, 1, 0), jarak –5 dari bidang y = 0. Untuk koefisien

ambient 0.7, diffuse 0.2, specular 0.8, reflektifitas 0 dan transparansinya 0,

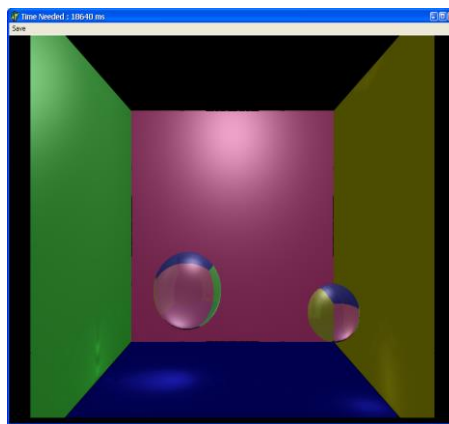
- Bola dengan titik pusat (2, 1, -1), jari - jari 4. Koefisien ambient 0.3, diffuse 0.9, specular 0.2, reflektifitas 0.2, transparansinya 0.7 dan index biasnya 1.25. Bola tidak bertekstur dan berwarna (0, 1, 0).
- Bola dengan titik pusat (-12, 2, 4), jari - jari 6.5. Untuk koefisien ambient 0.5,

diffuse 0.8, specular 0.2, reflektifitas 0.2 dan transparansinya 0 dan index biasnya 0.

Bola tidak bertekstur dan diset dengan warna (1, 0, 0).

- Skala map (ukuran luas pada dunia nyata yang diwakili oleh setiap satuan luas pada illumination map) pada bidang datar adalah 2.

Dengan data – data tersebut di atas maka didapatkan hasil seperti pada gambar 6.



Gambar 6.Efek sinar bias dengan 2 bola

4 Kesimpulan

1. Hasil pemodelan efek pengumpulan cahaya bias tergantung pada intensitas sumber cahaya, posisinya dan indeks bias obyek.

2. Pengaruh penggunaan struktur data kd-tree pada software yang akan dibangun adalah mempercepat waktu render karena menyimpan photon pada leaf lebih dari satu

Daftar Pustaka

- Arvo, J. 1986. Backward Ray Tracing. Apollo Computer, Inc. Chelmsfort, MA, <http://graphics.stanford.edu/courses/cs348b-97/basics/intersection/slides/>, (diakses 27 Februari 2008).
- 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>, (diakses 27 Februari 2008).
- Halliday, Resnick; Krane. 1960. Physics 4th Edition Volume 2. John Willey & sons.
- Purcell, Varberg. 1987. Calculus with Analytic Geometry 5th Edition. Prentice-Hall.
- Whitted. 1998. Basics of Ray Tracing. Lecture notes for Spring Quarter.
- Woo, M., Neider, J. dan Davis, T. 1997. OpenGL Programming Guide Second Edition. Silicon Graphics.
- Wright, Right S. 1996. OpenGL-Super Bible. San Francisco: Waite Group Press.
- 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>, diakses 27 Mei 2008).