

Klasifikasi *Shadow Algorithm*

Alfa Faridh Suni¹ dan Khoirudin Fathoni²

Universitas Negeri Semarang
Kampus Unnes Sekaran Gunungpati Semarang, 50229 Indonesia
alfafs@mail.unnes.ac.id¹, khoirudinfathoni@mail.unnes.ac.id²

Abstrak— Dalam bidang Grafika Komputer *Shadow Algorithm* merupakan salah satu kajian yang menarik. *Shadow Algorithm*, suatu algoritma yang digunakan untuk menghasilkan sebuah bayangan objek pada suatu *scene*. Pengembangan berbagai *shadow algorithm* sudah banyak dilakukan. Klasifikasi algoritma menurut tipe interaktifitasnya dan hasil render dengan cepat (*immediate render*), algoritma terbagi menjadi dua yaitu *real-time* dan *non-real-time*. Sedangkan melihat dari bayangan yang dihasilkan terdapat dua jenis yaitu *hard shadow* dan *soft shadow*.

Kata kunci— *shadow algorithm, projection shadow, shadow volumes, shadow mapping*

I. PENDAHULUAN

Pada masa awal studi Grafika Komputer dimulai menggunakan representasi gambar dua dimensi. Kemudian dimulai era gambar tiga dimensi objek dirender tanpa ada bayangannya, sehingga nampak tidak menyatu dengan lingkungannya pada *scene*.

Bayangan menjadi bagian penting pada suatu *scene*. Dengan adanya bayangan dapat memberikan informasi penting akan posisi sebuah objek. Bayangan dihasilkan karena cahaya dari sumber cahaya (*light source*) yang terhalang oleh suatu objek (*occluder*), sehingga menghasilkan sebuah daerah dengan tingkat kegelapan relatif pada suatu daerah (*receiver*) pada *scene*.

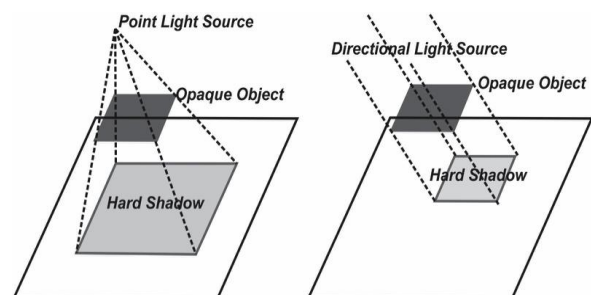
Dalam bidang Grafika Komputer *Shadow Algorithm* merupakan salah satu kajian yang menarik. *Shadow Algorithm*, suatu algoritma yang digunakan untuk menghasilkan sebuah bayangan objek pada suatu *scene*. Terdapat berbagai *Shadow Algorithm* dan pengklasifikasian algoritma tersebut telah melewati banyak perkembangan. Sebelumnya Crow [1] pertama kali mengusulkan *shadow polygon* yang kemudian lebih akrab disebut *Shadow Volume*. Serta memaparkan *Shadow Algorithm* dalam diklasifikasikan menurut tiga kategori umum. Hingga klasifikasi berdasarkan jenis bayangan yang dihasilkan seperti dipaparkan oleh Woo [2]. Artikel ini mencoba menjelaskan berbagai Klasifikasi *Shadow Algorithm* yang ada saat ini.

II. HARD SHADOW DAN SOFT SHADOW

Bayangan memiliki dua bagian area yaitu *umbra* dan *penumbra*. *Umbra* adalah bagian bayangan objek dengan tingkat kegelapan maksimum dikarenakan cahaya dari sumber cahaya terhalang sepenuhnya oleh objek (*occluder*). *Penumbra*, bagian bayangan yang masih terkena cahaya sebagian, yang membentuk batas area dari bayangan dengan perubahan tingkat kegelapan yang halus.

A. Hard Shadow

Hard Shadow dihasilkan oleh *point light source*. Anatomi *hard shadow* biasanya hanya terdiri atas bagian *umbra* saja, lihat Gambar 1. *Hard shadow* memberikan kesan tidak realistis, karena pada praktiknya *point light source* ideal tidak ada di dunia nyata. Namun *point light source* sangat mudah dimodelkan pada studi *Computer Graphics*. Pembuatan *hard shadow* biasanya lebih mudah karena sesuai prinsipnya hanya menggunakan status biner saja, dengan mengecek sebuah titik tertentu berada pada area bayangan atau tidak. Cara untuk mengetahuinya cukup dengan memeriksa apakah *point light source* terhalang oleh objek atau tidak dari posisi titik acuan tersebut, jika iya maka set warna pada titik tersebut dengan warna gelap [3].

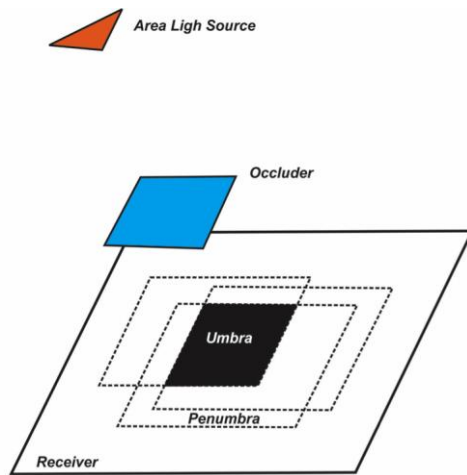


Gambar 1. *Hard shadow* dihasilkan oleh *point light source*

B. Soft Shadow

Secara umum *Soft Shadow* anatominya digambarkan memiliki area tengah yang gelap (*umbra*) yang dikelilingi batas area dengan perubahan kegelapan yang halus (*penumbra*) antara daerah bayangan menuju ke daerah tanpa bayangan [1].

Soft shadows dihasilkan oleh *area light source* jelas nampak lebih realistis dibandingkan *hard shadow*, apalagi dengan tingkat kehalusan bayangan yang bervariasi secara dramatis menurut jarak antara sumber cahaya (*light source*), penghalang (*occluder*) dan penerima (*receiver*) [3]. Terlihat pada Gambar 2.

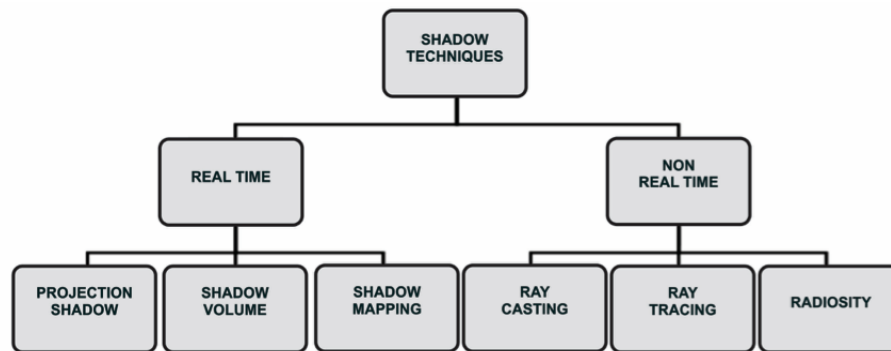


Gambar 2. *Soft Shadow* dihasilkan oleh *area light source*

III. SHADOW ALGORITHM

Jenis-jenis *shadows algorithm* juga telah dibahas secara komprehensif sebelumnya oleh Woo [2]. Woo mengklasifikasikan *Shadow Algorithm* kedalam dua kategori utama yaitu *Hard Shadow Generation*, algoritma-algoritma yang menghasilkan *hard shadows* dan *Soft Shadow Generation* algoritma-algoritma yang menghasilkan *soft shadow*.

Secara umum teknik-teknik dalam menghasilkan bayangan dibagi menjadi dua jenis yaitu: *Real-time* dan *Non-real-time* yang membedakan adalah interaktifitasnya. Teknik *real-time* biasanya interaktif serta susah untuk dipahami dan diimplementasikan. Meskipun teknik *non-real-time* tidak interaktif namun memiliki kualitas yang lebih baik dalam membuat bayangan [4]. Seperti terlihat pada Gambar 3.



Gambar 3. Tipe-tipe *Shadow Algorithm*

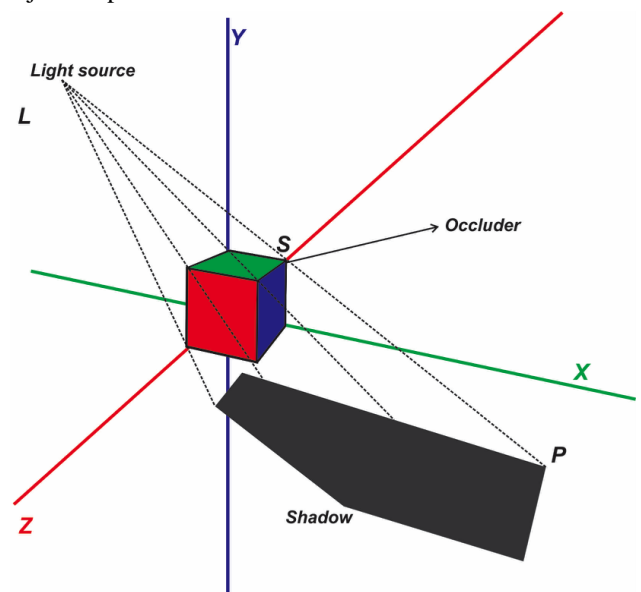
A. Teknik *Real-time*

1) *Projection Shadows*: teknik ini pertama kali diusulkan oleh Blinn pada 1988 seperti tertulis dalam artikel Nan [5]. Bayangan diproyeksikan pada permukaan bidang datar melalui *projection transformation*, mirip dengan proses memproyeksikan benda 2D ke layar/screen dilihat dari *view point*. Kemudian bayangan dirender dengan cara *projection transformation matrix* yang diperoleh dari posisi sumber cahaya (*light source*) dan bidang penerima (*receiver*) diaplikasikan terhadap object (*occluder*). Objek hasil proyeksi yaitu bayangan kemudian dirender dengan warna gelap dan tanpa iluminasi/pencahayaan. Ilustrasi ditunjukkan pada Gambar 4.

Metode *Projection Shadows* idenya sederhana dan komputasinya cepat namun memiliki kelemahan bayangan hanya dapat dilakukan pada bidang datar saja, semisal tembok atau tanah pada *scene* dan hanya salah satu bidang saja tidak bisa keduanya pada saat bersamaan. Karena keterbatasan tersebut bayangan yang dihasilkan biasa disebut *Fake Shadows*. Bayangan *Hard Shadows* akan dihasilkan oleh algoritma *Projection Shadows*.

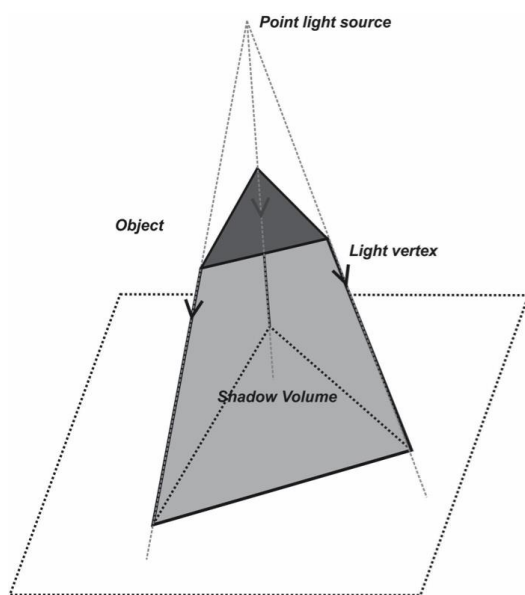
2) *Shadow Volumes*: pada 1977 Crow mengusulkan teknik *Shadow Volumes* [1]. Sebuah pendekatan untuk menghasilkan *umbra* lewat poligon bayangan sebuah objek pada *scene*, kemudian memasukannya ke dalam struktur data render

sebagai objek tak tampak (*invisible object*). Poligon bayangan/ruang bayangan (*shadow volumes*) adalah ruang yang dibentuk oleh poligon-poligon dengan batas-batas yang terbentuk oleh *occluder*, *receiver*, dan garis proyeksi yang ditarik dari *light source* terhadap *receiver*. Ilustrasi ditunjukkan pada Gambar 5.



Gambar 4. *Projection shadow*

Setelah didapat *shadow volumes*, dapat ditentukan permukaan *shadow volume* yang dekat / menghadap *view point* yang disebut *frontfacing polygon*. Permukaan *shadow volume* yang jauh yang disebut *backfacing polygon*. *Frontfacing polygon* menentukan batas dimulainya *shadow volume* dan *backfacing polygon* menentukan batas berakhirnya *shadow volume*. Penentuan bayangan dikomputasi menggunakan *shadow count* yang diinisialisasi dengan nilai 0. Apabila melewati *frontfacing polygon* maka *shadow count* ditambah nilai sebesar 1. Ketika melewati *backfacing polygon*, *shadow count* dikurangi nilai sebesar 1. Setelah *shadow count* bernilai 0 artinya sudah melewati area yang terkena cahaya atau diluar bayangan.



Gambar 5. *Shadow volumes*

Sama seperti *projection shadow* teknik yang digunakan pada *shadow volumes* akan menghasilkan bayangan jenis *hard shadow*.

3) *Shadow Mapping* : Pertama kali dikemukakan oleh Lance Williams pada 1978 [6], sebuah teknik yang disebut dengan *Shadow Mapping*. *Shadow mapping* membutuhkan *depth map (Z-buffer)* yang dirender dengan acuan *light source* sebagai *point of view*-nya. Meskipun algoritma *shadow mapping* sudah banyak berkembang namun secara prinsipnya algoritma *shadow mapping* melalui 2 tahap. Adapun langkah algoritmanya adalah sebagai berikut:

- Konstruksikan *view of scene* dengan *light source* sebagai *point of view*-nya. *Z-buffer* dikomputasi dan disimpan nilainya. *Z-buffer* adalah penanda apabila suatu titik/point berada pada area yang tidak terlihat (*hidden surface*).
- Konstruksikan *view of scene* dengan pandangan pengamat (*observer view*) sebagai *point of view*-nya. Kemudian transformasikan koordinat X,Y,Z pengamat (*observer view*) ke dalam koordinat X, Y, Z *light source*. Ketika poin sudah dihasilkan menurut

pandangan pengamat (*observer view*) akan ditransformasikan menurut hasil komputasi berdasarkan ruang koordinat *light source* dan akan dicek apakah poin/koordinat tersebut terlihat menurut *light source*. Apabila tidak terlihat artinya poin/koordinat tersebut berada dalam bayangan.

Shadow mapping banyak mengalami pengembangan selain *Standar Shadow Maps (SSM)* oleh Williams [6], *Light Space Perspective Shadow Map (LiSPSM)* oleh Wimmer [7] dan *Cascade Shadow Maps CSM* [8]. Adapun algoritma *shadow map* menghasilkan bayangan tipe *soft shadows*. Menjadikannya algoritma *soft shadow* yang banyak diminati oleh para peneliti. Studi tentang *soft shadow* sendiri biasanya dilakukan menilik karakteristik dari *penumbra*-nya, seperti yang sudah dilakukan oleh Hecher [9] salah satunya.

B. Teknik *Non-real-time*

1) *Ray-tracing*: Selain digunakan untuk memodelkan refleksi (pantulan) dan refraksi (pembiasan) cahaya, *ray-tracing* juga digunakan untuk menghasilkan bayangan. Prinsip yang digunakan pada *ray-tracing* yaitu sebuah *shadow-ray* ditembakkan dari titik perpotongan (titik potong dengan permukaan) menuju *light source*. Jika *ray/sinar* berpotongan dengan objek apapun antara titik awalnya (titik potong dengan permukaan) sampai ke *light source* maka dianggap titik tersebut berada dalam bayangan. Jika tidak berpotongan maka tidak berada dalam bayangan [10].

2) *Ray-casting*: *Ray-casting* merupakan versi cepat dari algoritma *ray-tracing* namun tidak mampu melakukan fungsi yang dimiliki *ray-tracing*. Prinsipnya algoritma dibagi menjadi dua bagian, bagian pertama menemukan perpotongan antara sinar/*ray* pada tiap piksel pada scene. Bagian kedua menentukan warna yang digunakan pada *environment map* tersebut [4].

3) *Radiosity*: Sebuah teknik yang lain dalam menghasilkan bayangan, terutamanya *soft shadow*. *Radiosity* menghitung releksinya cahaya *diffuse* antar permukaan dengan menentukan keseimbangan energi pada ruang tertutup. Algoritma ini termasuk kompleks dan membutuhkan waktu eksekusi yang lama. Serta hanya dapat digunakan pada permukaan poligon saja [4].

IV. PENUTUP

Pengembangan berbagai *shadow algorithm* sudah banyak dilakukan. Klasifikasi algoritma menurut tipe interaktifitasnya dan hasil render dengan cepat (*immediate render*), algoritma terbagi menjadi dua yaitu *real-time* dan *non-real-time*. Secara umum ada tiga algoritma *real-time*: *projection shadow*, *shadow volumes* dan *shadow mapping*. Dan tiga algoritma untuk *non real-time* yaitu: *ray-tracing*, *ray-casting* dan *radiosity*. Jika klasifikasi ditengok berdasarkan hasil jenis bayangannya maka dapat dikategorikan yaitu yang menghasilkan bayangan jenis: *hard shadow* dan *soft shadow*.

Masing-masing algoritma punya kelebihan dan kelemahan tersendiri. *Projection shadow* dan *shadow volumes* lebih umum digunakan untuk menghasilkan *hard shadow* meskipun telah ada upaya untuk memodifikasi algoritma tersebut guna menghasilkan *soft shadow*, seperti yang ditunjukkan oleh Liu menggunakan *projection shadow* dan Hasenfratz menggunakan *shadow volumes*. Untuk bayangan *soft shadow* yang biasa digunakan adalah algoritma *shadow mapping*.

Sama halnya dengan kedua algoritma sebelumnya, *shadow mapping* juga telah banyak mengalami kemajuan, dan modifikasi selain LiSPSM oleh Wimmer, adapula *Cascade Shadow Maps CSM*.

Untuk klasifikasi *shadow algorithm non real-time*, semuanya dapat menghasilkan bayangan jenis *soft shadow*. Namun memiliki *trade off* implementasi algoritma yang susah dan rumit serta membutuhkan *resource* yang besar. Ketiganya berdasarkan iluminasi untuk proses pembuatan bayangannya serta berbasis *image/citra*.

REFERENSI

- [1] F. C. Crow, "SHADOW ALGORITHMS FOR COMPUTER GRAPHICS," pp. 187–199, 1981.
- [2] A. Woo, P. Poulin, and A. Fournier, "A Survey of Shadow Algorithms," *IEEE Comput. Graph. Appl.*, vol. 10, no. 6, pp. 13–32, 1990.
- [3] J. M. Hasenfratz, M. Lapierre, N. Holzshuch, and F. Sillion, "A Survey of Soft Shadow Algorithm," *Comput. Graph. Forum*, vol. 22, no. 4, 2003.
- [4] H. Kolivand and M. S. Sunar, "An Overview on Base Real-Time Shadow Techniques in Virtual Environments," *Telkommika*, vol. 10, no. 1, pp. 171–178, 2012.
- [5] N. Liu and M. Pang, "Rendering soft shadows with OpenGL based on planar projection method," *Proc. - 2010 2nd IEEE Int. Conf. Netw. Infrastruct. Digit. Content, IC-NIDC 2010*, pp. 235–239, 2010.
- [6] L. Williams, "Casting Curved Shadows on Curved Surfaces.pdf," *ACM SIGGRAPH Comput. Graph.*, vol. 12, no. 3, pp. 270–274, 1978.
- [7] M. Wimmer, D. Scherzer, and W. Purgathofer, "Light Space Perspective Shadow Maps," *Render. Tech. 2004 (Proceedings Eurographics Symp. Render.)*, no. 2004, pp. 143–151, 2004.
- [8] R. Dimitrov, "Cascaded shadow maps," *Dev. Doc. NVIDIA Corp.*, no. August, 2007.
- [9] M. Hecher, M. Bernhard, O. Mattausch, D. Scherzer, and M. Wimmer, "A Comparative Perceptual Study of Soft-Shadow Algorithms," vol. 0, no. 0, 2014.
- [10] A. Woo, "Efficient Shadow Computations in Ray Tracing," *IEEE Comput. Graph. Appl.*, vol. 13, no. 5, pp. 78–83, 1993.