

# Penggunaan Algoritma Z-Fail dan Z-Pass untuk Mendesain Shadow Volume

<sup>1</sup>Ricky Lincoln Z.S  
<sup>2</sup>William

<sup>1</sup>Universitas Gundarama(ricxzone@student.gunadarma.ac.id)

<sup>2</sup>Universitas Gundarama(william\_1106@student.gunadarma.ac.id)

## Abstract

Nowadays the use of shadowing has become an important factor in the world of visualization. In order to produce a high-grade images and also covering the 3D base, all such things above will not be realized without knowing the technique of lighting and shadowing. Because all of the problem above, on this paper we will try and discuss an algorithm method which is often used in the image shadowing technique. The two algorithm that will be discussed in this paper is Z-fail Algorithm and Z-pass Algorithm. We are expected with the existence of this paper will provide a visible images of the implementation of those algorithm within the applications of Shadow Volumes. In this paper, we show the effect of stencil buffer to each algorithm, so it can be seen the differences of each stencil operation. We also give the output appearance of shadow volumes project which included in OpenGL project to demonstrate the function of Z-fail and Z-pass algorithm.

Keywords : Shadow Volume, Z-Fail, Z-Pass, OpenGL, Stencil Buffer

## 1 Pendahuluan

Bayangan (Shadows) telah menjadi suatu hal yang lazim di dunia grafik komputer, dengan adanya teknik pembayangan (shadowing technique) yang baik akan dapat meningkatkan realisme gambar yang diciptakan oleh komputer. Karena pentingnya bayangan dalam pemrosesan citra, telah banyak teknik algoritma yang diusulkan untuk mendapatkan teknik pembayangan yang terbaik. Pada saat ini terdapat 2 teknik pembayangan yaitu shadow maps (peta bayangan) dan shadow volumes (volume bayangan), Kedua teknik diatas merupakan teknik yang sering dipakai pada saat ini dalam melakukan penterjemahan citra.

Terdapat banyak kelebihan dan kekurangan dalam kedua teknik diatas, yang pada akhirnya dapat digunakan dalam meningkatkan kualitas pemrosesan citra. Dalam hal kecepatan, teknik shadow maps memang lebih unggul jika dibandingkan teknik shadow volumes karena shadow maps tidak membutuhkan stencil buffer tambahan layaknya shadow volumes. Tetapi jika kita melihat dari sisi ketepatan (accuracy) tiap pixel, maka teknik terbaik yang harus digunakan adalah teknik shadow volumes karena memiliki jangkauan tepi siluet yang baik.

Meskipun shadow volumes terlihat memiliki banyak kelebihan, tetapi dalam implementasinya,

teknik shadow volumes masih memiliki banyak kekurangan. Hal itu dikarenakan sampai saat ini masih belum dapat ditemukannya algoritma yang cocok untuk teknik ini. Untuk menjawab hal tersebut, dalam makalah ini kami mencoba untuk membahas algoritma yang cocok untuk digunakan dalam teknik shadow volumes. Dengan harapan agar setelah makalah ini selesai akan dapat membantu dalam proyek-proyek penelitian maupun pihak lain yang bergerak di bidang grafik komputer.

Algoritma untuk menterjemahkan proses shadow volumes yang akan kami bahas dalam makalah ini adalah algoritma z-fail dan z-pass. Z-pass akan menjadi efisien pada saat sebagian besar volume bayangan tersembunyi, sedangkan z-fail adalah sebaliknya. Inti dari hasil pengamatan metode ini adalah kita mendapatkan bahwa kedua algoritma menghasilkan bayangan identik, jadi kita dapat memilih menggunakan z-fail atau z-pass berdasarkan hasil terjemahan shadow volumes.

Makalah ini disusun sebagai berikut. Pertama-tama kami menjelaskan mengenai latar belakang penulisan makalah ini, dan pada bagian related works kami akan menjelaskan mengenai pekerjaan yang terkait dengan masalah dalam makalah ini. Lalu dalam bagian ketiga yaitu metodologi, kami akan membahas mengenai pendekatan yang digunakan dalam teknik shadow volumes dan juga mengenai algoritma z-fail dan z-pass. Dan bab ter-

akhir akan berisi kesimpulan daripada isi makalah ini

## 2 Kajian Pustaka

Dikarenakan shadow volumes merupakan teknik rendering image yang cukup sering digunakan, oleh karena itu telah banyak makalah-makalah lain yang membahas mengenai algoritma untuk kasus ini. Sehingga dalam bagian ini kami akan memberikan beberapa contoh makalah yang memiliki kesamaan permasalahan dalam beberapa aspect.

Joen Sindholt dalam makalahnya yang berjudul "A comparison of Shadow Algorithms"[4] mencoba untuk membuat aplikasi dengan menggabungkan 4 algoritma teknik rendering image, yaitu Shadows maps, Shadow volumes, Perspective shadow maps, dan algoritma penggabungan dengan menggunakan ide dari shadow Maps and Shadow Volumes. Aplikasi yang dibangun dengan menggunakan keempat teknik dirancang untuk dapat memilih algoritma yang terbaik berdasarkan kondisi pada saat program dijalankan. Makalah ini menjelaskan penggunaan teknik shadow volumes mulai dari cara pembuatan adjacent list dan menginisialisasikan depth value dari pada suatu gambar, kedua hal tersebut merupakan faktor penting dalam melakukan penggambaran teknik shadow volumes. Algoritma z-fail dan z-pass juga dipergunakan dalam hal untuk mendapatkan nilai stencil buffer, di kondisi receiver (penerima) berada di dalam bayangan objek, dengan cara meningkatkan atau menurunkan nilai stencil buffer pada saat pengecekan depth dilakukan.

Samuli Laine dalam makalahnya yang berjudul "Split-Plane Shadow Volumes"[5] membahas suatu metode yang dapat memilih antara algoritma z-fail atau z-pass bergantung kepada nilai-nilai pixel dasar. Pengimplementasian penggunaan z-fail dan z-pass dalam makalah ini terlihat dalam penyesuaian antara nilai zpixel dan zfrags. Dengan melakukan penyesuaian tersebut akan dapat menentukan apakah pixel tersebut merupakan front-shadow boundary atau back-shadow boundary. Split test dipergunakan untuk menguji nilai stencil update z-fail dan z-pass, dengan membandingkan nilai depth store yang terdapat didalam depth buffer dan zsplit.

Chenguang Dai, Yongsheng Zhang dan Jingyu Yang dalam makalahnya yang berjudul "Rendering 3D Vector Data Using The Theory Of Stencil Shadow Volumes"[3] mengemukakan suatu metode yang dapat digunakan dalam aplikasi penganalisis dan manajemen dari pemandangan virtual. Oleh karena itu dibutuhkan penggabungan visualisasi daerah dan geografis menjadi faktor yang penting. Metode ini menggunakan tiga

tahap dalam pengimplementasiannya, yang pertama adalah penggunaan data vektor polyhedra, lalu pembuatan aplikasi penghalang untuk stencil buffer dan terakhir pengimplementasian aplikasi sehingga dapat menterjemahkan pemandangan. Dalam pembuatan penghalang untuk stencil buffer, algoritma z-fail dan z-pass digunakan pada saat sudut pandang berada didalam shadow volumes dan terjadi interseksi dengan perpotongan bidang terdekat dari sudut pandang frustum digunakanlah algoritma z-fail dan algoritma z-pass digunakan pada kondisi sebaliknya.

Cass Everitt dan Mark J. Kilgard dalam makalahnya "Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering"[2] menunjukkan suatu metode untuk meningkatkan penggunaan stenciled shadow volumes pada perangkat-perangkat keras penterjemah citra. Pengadopsian skema stencil z-fail dan z-pass diperlukan untuk mendapatkan formulasi stencil shadow volumes yang konvensional. Didapatkan dengan cara meningkatkan atau menurunkan nilai shadow depths untuk tampilan belakang dan depan poligon pada saat pengecekan nilai depths dilakukan.

## 3 Metodologi

Shadow volume merupakan salah satu teknik yang digunakan dalam grafik komputer 3D untuk menambahkan suatu bayangan pada pembuatan adegan. Shadow volume ini pertama kali diusulkan oleh Frank Crow pada tahun 1977. Sebuah shadow volume dapat digambarkan sebagai ruang daerah tertutup yang terlihat seperti kerucut atau piramida. Ujung shadow volume bayangan adalah sumber cahaya, dimana permukaan dari shadow volume ditentukan oleh garis dari objek dan setiap bagian dari adegan yang berada di dalam shadow volume adalah sebuah bayangan.

Keuntungan utama dari shadow volume adalah akurat pada pixel (meskipun banyak implementasi masalah bayangan diri sepanjang tepi siluet), sedangkan akurasi peta bayangan tergantung pada memori tekstur yang dialokasikan untuk sudut di mana bayangan dilemparkan (di beberapa sudut, akurasi peta bayangan agak sulit). Namun, teknik shadow volume ini membutuhkan pembentukan bayangan geometri. Tes bayangan sering diwujudkan dengan menggunakan teknologi buffer kedalaman dan buffer stencil, yang dapat dibagi menjadi algoritma z-pass dan z-fail

Effect dari project shadow volume sudah mencapai 3 fase :

- Dapat menggambarkan seluruh citra dengan ambient dan sedikit pembiasan cahaya. Fase ini juga menunjukkan nilai-nilai kedalaman warna untuk objek-objek yang terlihat.

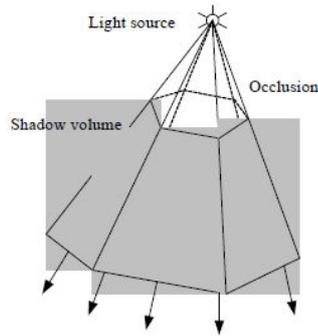


Figure 1: The illustration of shadow map

- Dapat menggambarkan volume bayangan ke dalam stencil buffer. Secara default volume bayangan menggunakan teknik "zFail", dengan perpotongan bidang yang tak terbatas.
- Dapat menggambarkan seluruh citra dengan pembauran dan penspekularan penuh, di daerah tanpa bayangan (dimana stencil == 0)

### 3.1 Teknik Shadow Volume

#### 3.1.1 Algoritma Z-Pass

Seperti yang terlihat pada figure 2, Algoritma Z-pass memiliki masalah dasar yang sering terjadi yaitu pada saat bidang terdekat dari kamera terletak di dalam shadow volume. Stencil buffer mengalami penurunan volume, dengan menetapkan nilai sebesar -15. Ketika sinar masuk ke dalam shadow volume, stencil meningkat sebesar satu dan pada saat sinar itu keluar, nilai stencil berkurang satu. Jika stencil memiliki nilai lebih besar dari nol, itu berarti bahwa pixel berada di dalam bayangan, sebaliknya jika stencil nilai sama dengan nol, pixel berada di luar bayangan.

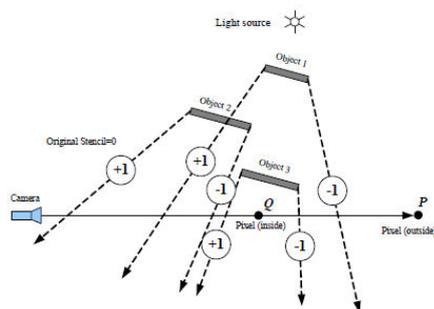


Figure 2: Algoritma z-pass

Di dalam algoritma Z-pass, jika shadow volume memotong bidang terdekat dari pandangan frustum, tutup permukaan tambahan harus ditambahkan ke shadow volume untuk mendapatkan hasil yang benar. Alasannya adalah bahwa setelah dipotong dari pandangan frustum, shadow vol-

ume akan cenderung terbuka. Jika tutup tambahan tidak ditambahkan, stencil nilai tidak akan meningkat dalam pembuatan shadow volume dan uji titik-titik bayangan di daerah yang terkena akan gagal.

Generasi dari nilai-nilai dalam stencil buffer adalah hasil dari operasi stencil berikut:

- pembuatan wajah depan shadow volume. Jika lulus uji kedalaman, maka akan mengalami kenaikan nilai stencil
- pembuatan wajah belakang shadow volume. Jika lulus uji kedalaman, maka akan mengalami penurunan nilai stencil

#### 3.1.2 Algoritma Z-Fail

Di dalam algoritma Z-fail, dilakukan perpanjangan sinar dari kamera sampai daerah yang tak terbatas. Ketika sinar keluar dari shadow volume, maka terjadi peningkatan nilai pada stencil bernilai satu dan apabila ketika sinar masuk ke dalam shadow volume, maka nilai stencil berkurang satu. Jika nilai stencil akhir lebih besar dari nol, maka pixel berada di dalam bayangan begitu pula sebaliknya jika nilai stencil akhir adalah sama dengan nol, maka pixel berada di luar bayangan. Seperti yang digambarkan dalam figure 3, jika algoritma z-pass diterapkan, maka nilai stencil Q adalah  $-1 + 1 = 0$  yang memiliki arti Q berada di luar bayangan, namun faktanya Q berada di dalam bayangan.

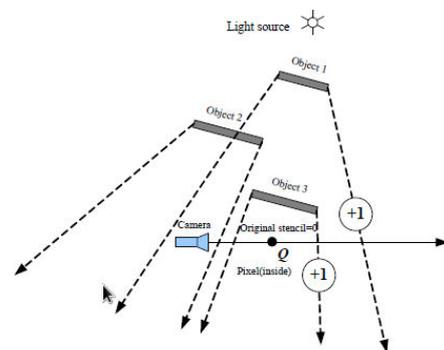


Figure 3: Algoritma z-fail

Algoritma ini dapat memecahkan masalah mata yang berada di bayangan, karena shadow volume antara mata dan objek tidak dihitung, tetapi kondisi pada bagian belakang/- akhir dari shadow volume harus ditutup atau bayangan akan hilang dan berakhir dimana nilai volume tak terbatas.

Catatan bahwa dalam algoritma Z-fail, shadow volume harus ditutup pada kedua ujungnya karena nilai stencil dihitung ketika pixel melakukan uji kedalaman. Jika tutup atas dan tutup bawah tidak ditambahkan, maka nilai stencil didalam shadow volume akan menjadi nol.

Berikut ini adalah algoritma Z-fail :

- Pembuatan wajah belakang shadow volume. Jika uji kedalaman gagal, maka nilai akan mengalami kenaikan stencil.
- Pembuatan wajah depan shadow volume. Jika uji kedalaman gagal, maka nilai akan mengalami penurunan stencil.

### 3.2 Membuat Shadow Volume dengan stencil 2 sisi

Proyek ini menggunakan EXT stencil two side extension, agar Shadow Volume dapat digambarkan hanya satu pada tiap objek. Hal ini memungkinkan perbedaan pengoperasian stencil di depan dan di belakang yang menghadap poligon dalam rendering yang sama. Penggunaan stencil dua sisi ini pun dapat meningkatkan kecepatan algoritma. Seperti yang digambarkan pada figure 2, terjadi beberapa peningkatan performa jika menggunakan teknik ini.

Berikut ini adalah pseudocode [1] untuk membuat shadow volume dengan stencil 2 sisi :

#### 1. menggunakan Algoritma Z-Fail

```

if(alwaysUseZPass || needZPass)
{
    glDisable(GL_CULL_FACE);

    //Increment(with wrapping) for front face depth pass
    glActiveStencilFaceEXT(GL_FRONT);
    glStencilFunc(GL_ALWAYS, 0, ~0);
    glStencilOp(GL_KEEP, GL_KEEP, GL_INCR_WRAP_EXT);

    //Decrement(with wrapping) for back face depth pass
    glActiveStencilFaceEXT(GL_BACK);
    glStencilFunc(GL_ALWAYS, 0, ~0);
    glStencilOp(GL_KEEP, GL_KEEP, GL_DECR_WRAP_EXT);

    //Enable 2 sided stencil
    glEnable(GL_STENCIL_TEST_TWO_SIDE_EXT);

    //Draw the shadow volume if not using vertex program
    if(!useVPVolumes)
        models[currentModel].DrawInfiniteShadowVolume(
            torus[i] objectSpaceLightPosition,false);
    //Draw the model if using vertex program
    if(useVPVolumes)
        glDrawElements(GL_TRIANGLES, models[currentModel]
            numIndices, GL_UNSIGNED_INT, models[currentModel].indices);
}

```

#### 2. menggunakan Algoritma Z-Pass

```

if(alwaysUseZPass || needZPass)
{
    glDisable(GL_CULL_FACE);

    //Increment(with wrapping) for front face depth pass
    glActiveStencilFaceEXT(GL_FRONT);
    glStencilFunc(GL_ALWAYS, 0, ~0);
    glStencilOp(GL_KEEP, GL_KEEP, GL_INCR_WRAP_EXT);

    //Decrement(with wrapping) for back face depth pass
    glActiveStencilFaceEXT(GL_BACK);
    glStencilFunc(GL_ALWAYS, 0, ~0);
    glStencilOp(GL_KEEP, GL_KEEP, GL_DECR_WRAP_EXT);

    //Enable 2 sided stencil
    glEnable(GL_STENCIL_TEST_TWO_SIDE_EXT);

    //Draw the shadow volume if not using vertex program
    if(!useVPVolumes)
        models[currentModel].DrawInfiniteShadowVolume(
            torus[i] objectSpaceLightPosition,false);
    //Draw the model if using vertex program
    if(useVPVolumes)
        glDrawElements(GL_TRIANGLES, models[currentModel]
            numIndices, GL_UNSIGNED_INT, models[currentModel].indices);
}

```

Hasil output dari Shadow Volume dapat dilihat dari gambar berikut :

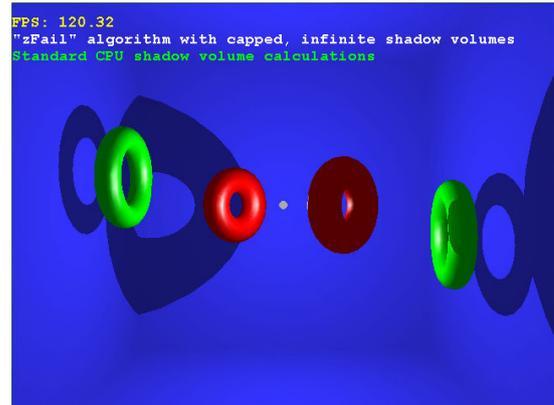


Figure 4: Hasil

## 4 Penutup

Berdasarkan teori-teori tentang teknik shadow volume diatas didapatkan suatu kesimpulan yaitu dalam metode shadow volume, algoritma Z-Pass memiliki kinerja yang lebih tinggi daripada algoritma Z-Fail. Tetapi algoritma Z-Pass akan gagal pada saat shadow volume memotong bidang terdekat yang tak terbatas. Masalah bidang terdekat adalah alasan untuk pengembangan teknik Z-Fail yang memproses fragmen shadow volume yang gagal (bukan lurus) pada saat uji kedalaman. Pendekatan ini dapat memindahkan masalah dari bidang terdekat sampai bidang terjauh yang dapat ditangani dengan menggerakkan bidang terjauh hingga tak terbatas.

## References

- [1] *OpenGL Shadow Volumes : Z-Pass and Z-Fail - Paul's Projects.* <http://www.paulsprojects.net/opengl/shadvol/shadvol.html>.
- [2] Cass Everitt and Mark J. Kilgard. *Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering.* page 8, Austin, Texas, 12 March 2002. NVIDIA Corporation,.
- [3] Jingyu Yang Chenguang Dai, Yongsheng Zhang. *RENDERING 3D VECTOR DATA USING THE THEORY OF STENCIL SHADOW VOLUMES.* volume Vol. XXXVII., page 6, Beijing, 2008. Zhengzhou Institute of Surveying and Mapping,, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

- [4] Joen Sindholt. *A comparison of Shadow Algorithms.* page 190, Lyngby, Denmark, 30 May

2005. Institute of Informatics and Mathematical Modelling, Technical University of Denmark.

- [5] Samuli Laine. Split-Plane Shadow Volumes. page 10, Helsinki, 2005. Helsinki University of Technology / TML, The Eurographics Association 2005.