

PEMODELAN OBYEK TIGA DIMENSI DARI GAMBAR SINTETIS DUA DIMENSI DENGAN PENDEKATAN *VOLUMETRIC*

Rudy Adipranata, Liliana

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

e-mail: rudya@petra.ac.id, lilian@petra.ac.id

ABSTRAK: Dalam penelitian ini dilakukan implementasi untuk pemodelan obyek tiga dimensi yang berasal dari gambar dua dimensi. Pemodelan ini dilakukan dengan menggunakan pendekatan *volumetric*. Dengan menggunakan pendekatan *volumetric*, ruang tiga dimensi dibagi menjadi bentuk diskrit yang disebut *voxel*. Kemudian pada *voxel-voxel* tersebut dilakukan metode pewarnaan *voxel* untuk mendapatkan hasil berupa obyek tiga dimensi yang bersifat *photorealistic*. Bagaimana memodelkan obyek tiga dimensi untuk menghasilkan hasil *photorealistic* merupakan masalah yang masih aktif di bidang komputer grafik. Banyak aplikasi lain yang dapat memanfaatkan hasil dari pemodelan tersebut seperti *virtual reality*, *augmented reality* dan lain-lain. Pewarnaan *voxel* merupakan pemodelan obyek tiga dimensi dengan melakukan rekonstruksi warna, bukan rekonstruksi bentuk. Metode ini bekerja dengan cara mendiskritkan obyek menjadi *voxel* dan kemudian melakukan pengecekan secara berurutan terhadap proyeksi *voxel* ke gambar input untuk menentukan apakah *voxel* tersebut merupakan bagian dari obyek atau bukan. Sehingga pada akhirnya akan dihasilkan suatu model tiga dimensi yang *photorealistic*.

Kata kunci: pewarnaan *voxel*, pemodelan tiga dimensi, obyek *photorealistic*.

ABSTRACT: In this paper, we implemented 3D object modeling from 2D input images. Modeling is performed by using volumetric reconstruction approaches by using volumetric reconstruction approaches, the 3D space is tessellated into discrete volumes called voxels. We use voxel coloring method to reconstruct 3D object from synthetic input images by using voxel coloring, we can get photorealistic result and also has advantage to solve occlusion problem that occur in many case of 3D reconstruction. Photorealistic 3D object reconstruction is a challenging problem in computer graphics and still an active area nowadays. Many applications that make use the result of reconstruction, include virtual reality, augmented reality, 3D games, and another 3D applications. Voxel coloring considered the reconstruction problem as a color reconstruction problem, instead of shape reconstruction problem. This method works by discretizing scene space into voxels, then traversed and colored those voxels in special order. The result is photorealistic 3D object.

Keywords: voxel coloring, 3D modeling, photorealistic object.

PENDAHULUAN

Dewasa ini, pemodelan obyek tiga dimensi secara otomatis dari satu atau lebih gambar dua dimensi merupakan area penelitian aktif di bidang komputer grafik. Hasil dari pemodelan tersebut penting bagi aplikasi-aplikasi seperti, *virtual reality*, visualisasi interaktif dari suatu obyek yang direkam oleh kamera video, modifikasi virtual dari gambar nyata untuk *augmented reality* dan lain-lain.

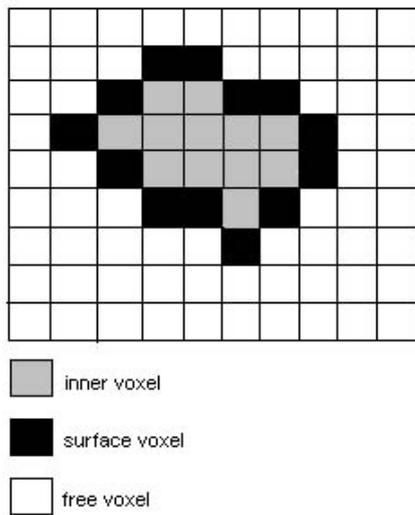
Sampai saat ini belum terdapat solusi yang dapat diterapkan untuk semua keadaan dikarenakan hampir semua obyek pada dunia nyata terdiri dari geometri yang kompleks dan menimbulkan kesulitan saat dilakukan pemodelan untuk membentuk obyek tersebut dalam tiga dimensi. Terdapat beberapa pendekatan yang biasa digunakan untuk memecahkan masalah tersebut. Pendekatan pertama adalah *image based rendering*. *Image based rendering* ini akan memberikan keluaran yang merupakan gambar

sintetis dari *photograph* dimana gambar sintetis ini berupa obyek tiga dimensi tetapi tidak terdiri dari geometri primitif. Pendekatan yang lain adalah *image based modeling*, dimana dengan pendekatan ini keluaran yang didapat adalah model tiga dimensi yang terbentuk dari geometri primitif.

PENDEKATAN *VOLUMETRIC*

Pendekatan *volumetric* merepresentasikan ruang tiga dimensi menjadi bentuk diskrit disebut *voxel* yang biasanya berbentuk kubus kecil. *Voxel-voxel* tersebut tersusun pada ruang tiga dimensi membentuk ruang *voxel*. Dalam ruang *voxel*, dicari permukaan tiga dimensi yang jika di-*render* akan merepresentasikan model tiga dimensi. Representasi *volumetric* ini menyediakan perhitungan yang mudah dan topologi yang fleksibel untuk merekonstruksi permukaan pada ruang tiga dimensi. Jika diperlukan, model *voxel* dapat diubah menjadi representasi poligon sehingga

dapat di-render dengan lebih efisien menggunakan *hardware* grafik umum.



Gambar 1. Representasi *Voxel*

Pada penelitian ini, pendekatan *volumetric* diimplementasikan dengan menggunakan metode pewarnaan *voxel* yang dikemukakan oleh Seitz dan Dyer [1]. Karena merupakan pendekatan *volumetric*, maka langkah awal metode pewarnaan *voxel* ialah mendiskritkan obyek yang akan dimodelkan menjadi sekumpulan *voxel*, yang akhirnya membentuk suatu ruang yang disebut ruang *voxel*. Kemudian metode pewarnaan *voxel* akan melakukan pengecekan secara berurutan terhadap ruang *voxel* ini dimulai dari posisi yang terdekat dengan kamera hingga posisi yang terjauh. Pengecekan ini dilakukan untuk mengidentifikasi *voxel* mana yang mempunyai warna yang unik dan konsisten. Sebuah *voxel* dianggap konsisten jika hasil proyeksi dari *voxel* tersebut ke gambar input menghasilkan warna yang sama untuk semua gambar dimana *voxel* tersebut terlihat.

Masalah utama pada pewarnaan *voxel* adalah bagaimana menandai sebuah *voxel* dengan warna pada volume tiga dimensi dan tetap mempunyai konsistensi terhadap sekumpulan gambar input. Dari hasil pewarnaan *voxel* tersebut diharapkan dapat merekonstruksi obyek sesuai dengan gambar input. Secara formal, ruang tiga dimensi (*scene*) yang digunakan (S) direpresentasikan sebagai sekumpulan Lambertian *voxel* yang tidak tembus cahaya, dimana tiap *voxel* akan mencakup sebuah volume terbatas yang terpusat pada sebuah titik $V \in S$ dan mempunyai *isotropic radiance* $color(V, S)$. Asumsi yang digunakan adalah seluruh obyek tercakup dalam sebuah batasan yang diketahui yaitu *finite bounding volume*. Semua *voxel* yang terletak di dalam *bounding volume* disebut ruang *voxel* yang dilambangkan dengan

simbol V . Sebuah gambar I dispesifikasikan sebagai sekumpulan *pixel* dimana *pixel-pixel* tersebut terpusat pada titik $p \in I$ dan mempunyai *irradiance* $color(p, S)$.

Untuk sebuah *pixel* $p \in I$ dan *scene* S , terdapat *voxel* $V \in S$ yang terlihat pada I dan terproyeksikan ke p oleh $V = S(p)$. *Scene* S dianggap lengkap terhadap sekumpulan gambar jika untuk setiap gambar I dan setiap *pixel* $p \in I$ terdapat sebuah *voxel* $V \in S$ sehingga $V = S(p)$. Sebuah *scene* yang lengkap disebut konsisten terhadap sekumpulan gambar jika untuk setiap gambar I dan setiap *pixel* $p \in I$, $color(p, I) = color(S(p), S)$.

Secara formal, masalah pada pewarnaan *voxel* dapat didefinisikan sebagai : untuk sekumpulan gambar dasar I_0, \dots, I_n , dari *scene* Lambertian statis dan sebuah ruang *voxel* V , tentukan sebuah *subset* $S \subset V$ dan pewarnaan $color(V, S)$, sehingga $S \in \mathcal{N}$ dimana \mathcal{N} adalah kumpulan dari semua gambar yang konsisten.

Invariant Warna

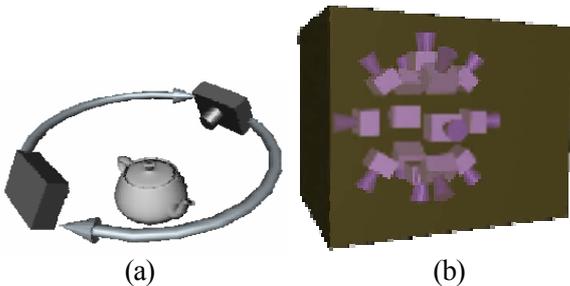
Satu-satunya cara untuk mendapatkan informasi dari sebuah *scene* untuk memecahkan persoalan pewarnaan *voxel* adalah dengan menggunakan *invariant*, yaitu suatu sifat yang terdapat pada semua *scene* yang konsisten. Sebagai contoh dapat diperhatikan sekumpulan *voxel* yang terdapat di setiap *scene* yang konsisten. Laurentini [3] telah mendefinisikan bagaimana cara untuk mendapatkan suatu *invariants* yang disebut *hard points*, dengan menggunakan perpotongan volume dari gambar *silhouette*. Dengan menggunakan *hard points*, informasi yang benar mengenai *scene* dapat dihasilkan, hanya saja, *hard points* ini jarang ditemukan, bahkan untuk beberapa gambar, tidak didapat *hard points*.

Untuk itu, masalah *invariant* yang digunakan di sini bukan *invariant* yang berupa bentuk seperti *hard points*, tetapi *invariant* yang berupa warna. Sebuah *voxel* disebut *invariant* warna terhadap sekumpulan gambar jika memenuhi dua kondisi yaitu : terdapat dalam sebuah *scene* yang konsisten terhadap gambar dan untuk setiap pasangan *scene* yang konsisten S dan S^1 , $V \in S \cap S^1$ termasuk $color(V, S) = color(V, S^1)$. Tidak seperti *invariant* bentuk, *invariant* warna tidak memerlukan sebuah titik harus terdapat di setiap *scene* yang konsisten. Sehingga sebagai hasilnya, *invariant* warna lebih umum daripada *hard points*. Gabungan dari *invariant* warna ini membentuk *scene* yang konsisten, sehingga permasalahan pewarnaan *voxel* dapat digambarkan sebagai permasalahan yang dapat diselesaikan dengan memecahkan masalah *scene* yang konsisten dan berhubungan dengan *invariant* warna. Tetapi untuk lebih mudahnya untuk menyelesaikan masalah tersebut dibutuhkan batasan-batasan lain.

Batasan Ordinal Visibility

Batasan *ordinal visibility* akan mengaktifkan identifikasi dari sekumpulan *invariant* warna sebagai titik batas dari \mathcal{S} , dimana \mathcal{S} adalah kumpulan semua *scene* yang konsisten. Diasumsikan P dan Q adalah titik *scene* dan I adalah gambar dari kamera yang mempunyai pusat di C. P akan menutupi Q jika P terletak pada segmen garis \overline{CQ} . Kamera input harus diletakkan pada posisi tertentu sehingga memenuhi batasan : terdapat sebuah $norm \parallel . \parallel$ sehingga untuk semua titik *scene* P dan Q, dan gambar input I , P menutupi Q pada I hanya jika $\|P\| < \|Q\|$.

Untuk memenuhi batasan ordinal visibility ini, maka penempatan kamera harus memenuhi konfigurasi seperti gambar di bawah ini dimana pada gambar 2.a adalah posisi kamera jika obyek berupa benda solid, dan gambar 2.b jika obyek berupa ruang.

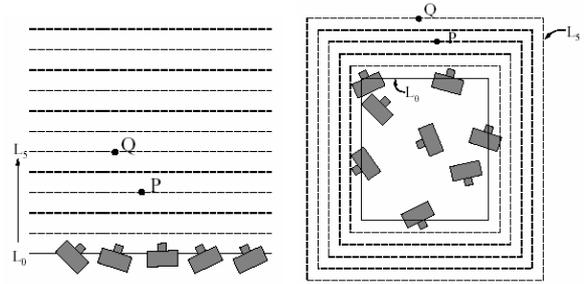


Gambar 2. Posisi Kamera

Algoritma Pewarnaan Voxel

Untuk merumuskan algoritma pewarnaan *voxel*, maka harus merumuskan cara untuk mendapatkan koleksi *invariant* warna dengan pengecekan tunggal pada *volume scene* yang telah didiskritkan dengan menggunakan batasan *ordinal visibility*. Batasan ini membatasi konfigurasi penglihatan dasar yang mungkin tetapi mempunyai keuntungan bahwa penglihatan tersebut dapat disederhanakan. Secara umum, dimungkinkan untuk membagi *scene* menjadi beberapa bagian *voxel layer* yang mempunyai hubungan penglihatan *monotonic*: untuk setiap gambar input, *voxel* hanya menutupi *voxel* lain yang terletak di *layer* yang mengikutinya. Dan sebagai konsekuensinya, hubungan penglihatan dapat diselesaikan dengan mengevaluasi *voxel* pada satu *layer* pada satu saat.

Untuk menggambarkan pembagian *layer* tersebut, dapat dilihat pada gambar di bawah ini, dimana *voxel* dapat dibagi menjadi beberapa *layer* yang makin lama makin jauh dari kamera.



Gambar 3. Pengecekan Per Layer

Untuk kasus tiga dimensi dapat dilakukan pembagian *layer* dengan cara yang sama. Pada tiga dimensi, sebuah *layer* akan menjadi permukaan yang tereksansi keluar dari kamera.

Secara lengkap, pseudocode untuk algoritma pewarnaan *voxel* dapat dilihat di bawah ini.

```

create empty voxel space and partition it into
layers
for each layer in the voxel space from nearest
camera location {
for each voxel in the layer {
project the voxel to each image
collect the set of pixels to which
this voxel projects and the pixels
haven't been marked yet
calculate the color consistency of set of
pixels

if (consistency < threshold) {
color this voxel
marked pixel related with this
voxel
}
}
}
    
```

Konsistensi Voxel

Sebagai kompensasi dari efek akibat kuantisasi pada gambar dan efek akibat *noise*, maka diasumsikan bahwa gambar telah didiskritkan pada grid dari *pixel* yang tidak saling tumpang tindih. Jika sebuah *voxel V* tidak tertutupi secara penuh pada gambar I_j , maka proyeksi *voxel* tersebut akan menindih sekumpulan *pixel* gambar yang tidak kosong π_j . Tanpa efek yang diakibatkan kuantisasi ataupun *noise*, sebuah *voxel* yang konsisten seharusnya akan terproyeksi ke sekumpulan *pixel* yang mempunyai warna yang sama. Dengan adanya efek tersebut, maka harus dilakukan evaluasi terhadap korelasi λ_V dari warna *pixel* untuk menghitung kemungkinan konsistensi *voxel*. Misalkan s adalah standar deviasi

dan m adalah *cardinality* dari $\bigcup_{j=0}^n \pi_j$. Satu kemungkinan pilihan untuk fungsi korelasi adalah dengan menspesifikasikan *threshold* dari kesalahan ruang warna

$$\lambda_v = s \tag{1}$$

Alternatif lain untuk penghitungan konsistensi *voxel* adalah dengan menggunakan statistik. Diasumsikan bahwa kesalahan pada sensor terdistribusi normal dengan standar deviasi σ_0 . Konsistensi *voxel* dapat diestimasi dengan menggunakan test ratio kemungkinan, yang terdistribusi sebagai χ^2 dengan derajat kebebasan $n-1$:

$$\lambda_v = \frac{(m-1)s^2}{\sigma_0^2} \tag{2}$$

KALIBRASI DAN PROYEKSI KAMERA

Kalibrasi kamera adalah proses untuk mendapatkan parameter intrinsik dan ekstrinsik kamera. Parameter intrinsik meliputi geometri internal kamera dan parameter optik seperti *focal length*, koefisien distorsi lensa, faktor-faktor skala yang tidak dapat diestimasi dan koordinat origin gambar pada komputer. Parameter ekstrinsik meliputi orientasi posisi kamera terhadap sistem koordinat dunia. Hal ini meliputi tiga sudut Euler dan tiga parameter translasi.

Matriks proyeksi kamera H adalah matriks 3×4 yang mendeskripsikan bagaimana sebuah titik $P=(X,Y,Z)$ pada ruang tiga dimensi terproyeksikan ke sebuah titik $p = (x,y)$ pada sebuah gambar. Dengan menggunakan representasi homogen, formula tersebut adalah sebagai berikut :

$$p = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = HP = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

s adalah faktor skala.

Matriks proyeksi (H) dapat disusun dengan menggunakan kalibrasi kamera. Formulanya adalah sebagai berikut :

$$H = K [R | -RT] \tag{4}$$

K adalah matriks 3×3 sebagai berikut

$$K = \begin{bmatrix} \frac{f}{d_x} & 0 & c_x \\ 0 & \frac{f}{d_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

- d_x = ukuran horisontal *pixel*
- d_y = ukuran vertikal *pixel*
- c_x = posisi pusat horisontal gambar
- c_y = posisi pusat vertikal gambar
- f = *focal length*

Matriks R adalah matriks rotasi 3×3 yang mendeskripsikan orientasi dari kamera. R mempunyai tiga derajat kebebasan yang dideskripsikan sebagai (ψ, θ, ϕ) dimana masing-masing merepresentasikan x, y and z .

$$R = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix}$$

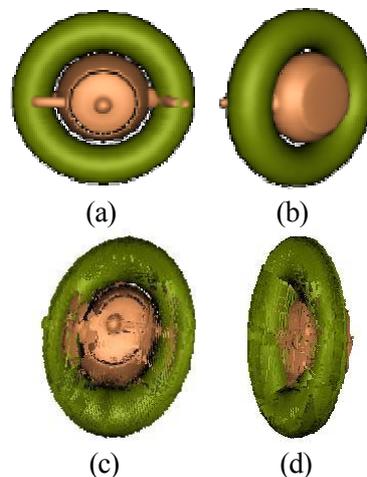
Matriks yang terakhir adalah matriks

$$T = [T_x, T_y, T_z]^T \tag{7}$$

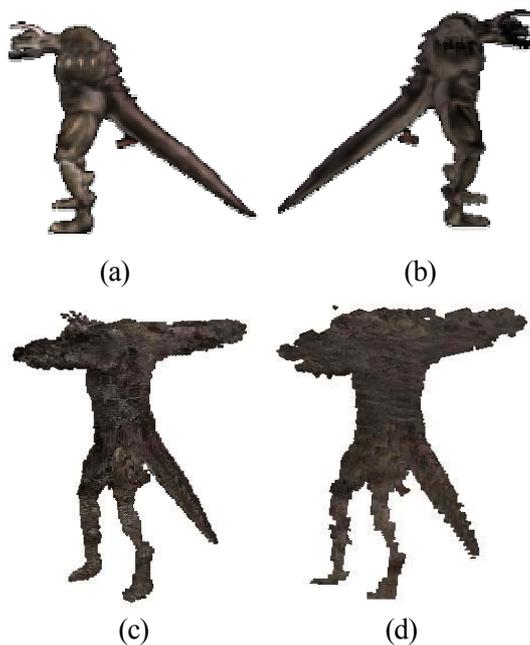
Matriks ini merupakan matriks posisi pusat proyeksi kamera pada koordinat dunia.

HASIL EKSPERIMEN

Pada penelitian ini, pengimplementasian metode pewarnaan *voxel* adalah dengan menggunakan OpenGL dan Microsoft Visual C++. Sebagai input adalah gambar sintesis yang dibuat dengan menggunakan 3D Studio Max. Pada eksperimen ini dibuat dua obyek yaitu pot dan monster yang kemudian dilakukan pemodelan tiga dimensi. Secara lengkap untuk masing-masing obyek terdapat delapan gambar input yang diambil dari dengan menggunakan kamera dengan delapan posisi. Pada gambar dibawah ini ditampilkan gambar input beserta hasil pemodelan dari masing-masing obyek yang dilihat dari dua sudut pandang.



Gambar 4. (a), (b). Gambar Input Pot. (c), (d) Hasil Pemodelan



**Gambar 5. (a), (b) Gambar Input Monster
(c), (d) Hasil Pemodelan**

Berikut ini juga disajikan tabel yang menyatakan waktu yang dibutuhkan untuk melakukan pemodelan dan jumlah *voxel* yang terbentuk dari kedua obyek input.

Tabel 1. Waktu dan Jumlah *Voxel* Yang Terbentuk

Ukuran Ruang <i>voxel</i>	Waktu (detik)	Jumlah <i>Voxel</i>
$(50)^3$ <i>voxel</i>	2.18700	18,479
$(100)^3$ <i>voxel</i>	8.23500	42192
$(150)^3$ <i>voxel</i>	17.95300	42,192
$(200)^3$ <i>voxel</i>	35.87500	42,192

KESIMPULAN

Dari hasil eksperimen yang dilakukan, dapat diambil kesimpulan:

- Dengan menggunakan pewarnaan *voxel* dapat dihasilkan pemodelan obyek tiga dimensi yang mempunyai sifat *photorealistic* dan sekaligus dapat mengatasi masalah penutupan obyek (*occlusion*).
- Pendefinisian ruang *voxel* (*voxel space*) tergantung dari ukuran obyek yang hendak dimodelkan. Jika ruang *voxel* terlalu besar maka akan terjadi pemborosan memori dan pemrosesan karena jumlah *voxel* yang dihasilkan tetap sama. Sedangkan jika ruang *voxel* terlalu kecil maka obyek tidak akan termodelkan seluruhnya.

DAFTAR PUSTAKA

1. Aldo Laurentini, "How far 3D shapes can be understood from 2D silhouettes", IEEE Trans. on Pattern Analysis and Machine Intelligence, 1995. pp.188-195.
2. Collins, Robert. T, "A Space-Sweep Approach to True Multi-Image Matching", Proc. Computer Vision and Pattern Recognition Conf., 1996. pp. 358-363.
3. Rudy Adipranata, Yang Hwang Kyu dan Yun Tae Soo, "Implementation of Photorealistic 3D Object Reconstruction Using Voxel Coloring", 19th Conference of Korean Information Processing Society, 2003. pp. 527-530.
4. Steven M. Seitz dan Charles R. Dyer, "Photo-realistic Scene Reconstruction by Voxel Coloring", Proc. Computer Vision and Pattern Recognition Conf, 1997, pp.1067-1073.