

Aplikasi Rekonstruksi 3D Menggunakan Metode *Voting-Based Voxel Carving*

Ricky Tanojo¹, Liliana², Djoni Haryadi Setiabudi³

Progeram Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Sihalankerto 121-131, Surabaya, Indonesia

Email: ricky.tanojo@gmail.com¹, lilian@petra.ac.id², djonihs@petra.ac.id³

ABSTRAK

Rekonstruksi 3D adalah teknik untuk membentuk objek 3D dari dunia nyata kedalam komputer. Dalam melakukan proses rekonstruksi, dapat digunakan berbagai macam metode, salah satunya adalah *voting – based voxel carving*, yaitu metode yang melakukan pembentukan suatu objek 3D dengan mengumpulkan gambar objek dari berbagai sudut, sehingga dapat diperoleh bentuk 3D.

Proses yang dilakukan adalah sebagai berikut: *user* meletakkan objek pada titik yang sudah ditentukan. Gambar dari objek akan diproses menjadi *silhouette* dari objek. Setelah diambil gambar dari satu sisi, hal yang sama akan dilakukan sampai semua sisi dari objek didapatkan. Aplikasi dibuat dengan C++ dengan Microsoft Visual Studio 2010 sebagai IDE-nya.

Hasil pengujian menunjukkan bahwa proses rekonstruksi 3D menggunakan *voting – based voxel carving* belum dapat menghasilkan objek 3D yang menyerupai objek nyata, dikarenakan terdapat beberapa efek dari faktor pencahayaan, jarak, dan jenis permukaan benda.

Kata Kunci

Voxel, voxel carving, shape from silhouette.

ABSTRACT

3D reconstruction is a technique for making 3D object from real world into a computer. There are many methods known for reconstructing 3D, one of the method is voting – based voxel carving, which are reconstructing 3D object, by taking images from several point of view, so 3D shape of the object can be retrieved.

The process is defined as follows: user put real object at certain point. Image of the object will be processed, until silhouette of the object can be retrieved. After taking image from one side, same method will be done until all side of the object has been processed. This application is built on C++ with Microsoft Visual Studio 2010 as the IDE.

Experimental results indicated that 3D reconstruction process using voting – based voxel carving have not achieved 3D object that similar to real object, because there are side-effects caused by lighting, distance, and type object surface.

Keywords

Voxel, voxel carving, shape from silhouette.

1. PENDAHULUAN

Pada saat ini, perkembangan teknologi sudah merambah kedalam banyak aspek kehidupan, salah satunya yaitu bidang *computer graphic*. Perkembangan tersebut sudah membantu dalam pembuatan objek *digital*, namun masih muncul masalah dalam pembuatannya, yaitu harus melakukan secara manual. Dengan melakukan rekonstruksi 3D dari objek nyata, maka dapat menghasilkan objek dari objek nyata dalam waktu lebih singkat dan efisien.

Metode yang akan digunakan adalah *voting-based voxel carving*, yang menggunakan dasar dari *shape from silhouette* sebagai acuan pembentukan objek. Proses rekonstruksi menggunakan gambar dari berbagai sudut sehingga dapat dihasilkan objek 3D yang merupakan perpotongan dari gambar yang digunakan.

2. DASAR TEORI

2.1 *Shape from Silhouette*

Metode *shape from silhouette* adalah metode dasar yang digunakan dalam pembuatan aplikasi rekonstruksi. Metode ini menggunakan gambar *silhouette* yang terdiri atas bentuk dari objek dan *background* dari objek. *Silhouette* adalah gambar dari objek, objek atau pemandangan yang direpresentasikan dalam sebuah bentuk *solid* yang memiliki satu warna. *Silhouette* dari objek akan digunakan sebagai acuan dari bentuk objek, sehingga dapat dilakukan proses rekonstruksi 3D. [1]

Metode ini sudah banyak digunakan dalam rekonstruksi 3D, dikarenakan rekonstruksi ini menggunakan gambar dari objek yang berada pada posisi statis atau tidak bergerak, sehingga gambar *silhouette* yang dihasilkan dapat lebih fokus terhadap objek. Selain itu, estimasi yang digunakan untuk rekonstruksi objek juga bergantung dari gambar *silhouette* yang digunakan, karena jumlah gambar yang dipakai sebagai *input* mempengaruhi tingkat *detail* dari objek. [1]

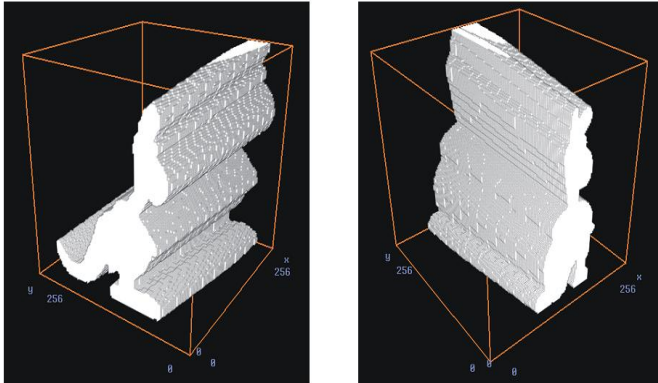
2.2 *Visual Hull*

Visual Hull merupakan istilah yang mewakili suatu bentuk dari objek dari satu sisi pengambilan gambar atau sudut pandang. Untuk memperoleh *visual hull* dari suatu objek, diperlukan berbagai *visual cone* dari objek tersebut. *Visual cone* yang dimaksudkan adalah bentuk dari objek dari suatu sisi atau sudut pengambilan gambar terhadap objek yang sama. Misalkan *visual cone* dari suatu apel yang diambil gambarnya dari depan, memiliki *visual cone* yang berbeda dengan sudut-sudut yang lainnya. Sehingga, dapat diketahui bentuk dari apel tersebut,

apabila semua *visual cone* dari apel tersebut dirangkai kedalam *visual hull* dan dibentuk ke dalam bentuk 3D.

Pembentukan *visual hull* dilakukan dengan menggabungkan semua perpotongan dari *visual cone* dan akan diproyeksikan kedalam *voxel*, yang dapat berupa urutan dari titik-titik yang terdeteksi sebagai proyeksi objek dari dunia nyata. Namun, metode ini masih memiliki kelemahan dalam rekonstruksi 3D, yaitu ketidakmampuan dalam melakukan rekonstruksi pada objek yang memiliki lubang. [1]

Contoh dari *visual hull* dari sebuah objek dapat dilihat pada Gambar 1.



Gambar 1. Gambar *Visual hull* [2]

2.3 Image Segmentation

Untuk mendapatkan gambar *silhouette* yang dibutuhkan untuk *shape from silhouette*, dibutuhkan cara untuk mendeteksi objek yang berada pada gambar, sehingga dapat dibentuk gambar *silhouette* dari objek tersebut. Ada berbagai macam cara *image segmentation*, seperti *threshold*, *edge detection*, *watershed*, dan sebagainya. Metode yang akan digunakan adalah bentuk dari *thresholding* dalam perbandingan warna, dengan menggunakan *hue*, *intensity*, dan *saturation*, agar dapat diperoleh bentuk dari objek tersebut dengan menghilangkan warna *background*.

$$I = \left(\frac{b + g + r}{3} \right) \quad (2.1)$$

$$S = 1 - \frac{\min(r, g, b)}{I}, I > 0.05 \quad (2.2)$$

$$H = \arccos\left(\frac{(r - g) + (r - b)}{\sqrt{(r - g)^2 + (r - b)(g - b)}}\right), b < g \quad (2.3)$$

Dari rumus di atas, dapat diketahui nilai dari *hue* (*H*), *intensity* (*I*), dan *saturation* (*S*), yang akan digunakan sebagai nilai untuk melakukan ekstraksi objek dari *background*. [2]

2.4 Morphological

Banyak metode yang digunakan sebagai metode untuk menghilangkan *isolated pixel*, seperti menggunakan *gaussian filter*, metode *salt-and-pepper*, dan lainnya. Metode yang digunakan untuk menghilangkan *noise* dari image yang

digunakan adalah metode *morphology*, yaitu metode yang menggabungkan 2 cara dasar dalam pengolahan citra digital, yaitu *erosion* dan *dilation*.

Erosion merupakan cara mengurangi luasan dari objek dengan menggunakan *mask* yang berupa *matrix* yang digerakkan ke seluruh area *image* dengan tujuan menghilangkan titik yang terisolasi, sedangkan *dilation* merupakan cara untuk memperluas area objek. Dengan menggunakan 2 cara tersebut, maka *image* yang didapatkan dapat menjadi lebih bersih dari *noise*. [3]



Gambar 2. Gambar *Morphology* [4]

Pada Gambar 2, bagian yang berwarna biru merupakan bentuk dari gambar asli. Hasil dari *dilation* merupakan bagian berwarna kuning, sedangkan bagian yang berwarna kuning merupakan hasil dari *erosion*.

2.5 Voting-based Voxel Carving

Metode yang sering dipakai untuk merepresentasi kan *visual hull* adalah *voxel carving*, yang merupakan kumpulan titik yang merepresentasikan bentuk 3D objek. Setiap titik pada *voxel cube* disebut *voxel*, yang merupakan singkatan dari *volumetric pixel* atau *pixel* yang memiliki ukuran panjang, lebar, dan tinggi. *Voxel* dapat dijelaskan dengan bentuk sebuah kubus kecil yang digunakan untuk menyusun objek 3D. [5]

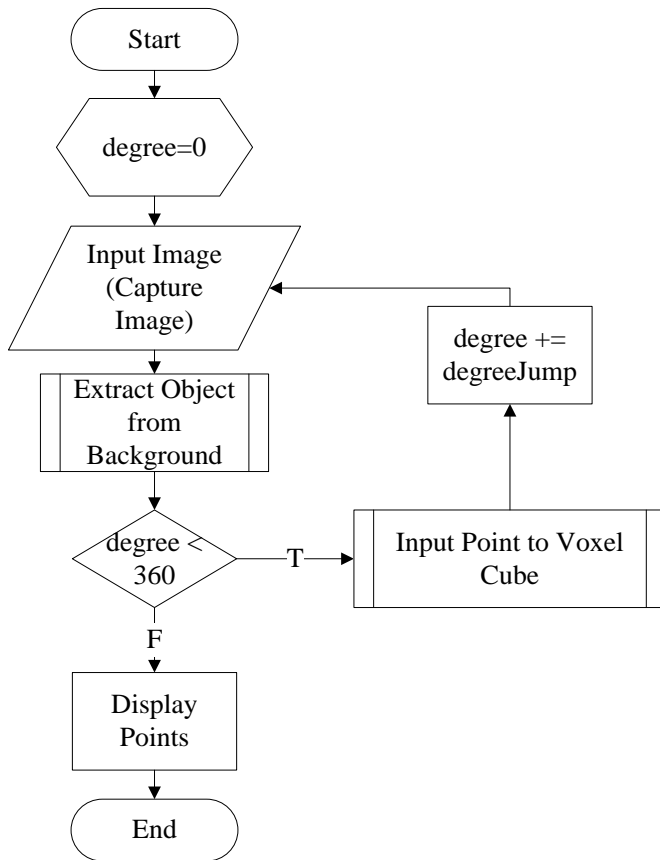
Voxel carving banyak digunakan dalam rekonstruksi 3D, karena merupakan salah satu cara yang menggunakan konsep *visual hull* ke dalam cara rekonstruksi 3D, dimana bentuk dari objek ditentukan oleh tampilan dari satu sisi objek tersebut dan bentuk 3D akan terbentuk setelah diperoleh bentuk atau *silhouette* dari sisi yang lain.

Cara kerja dari *voxel carving* adalah menerima hasil *input* gambar yang sudah berupa *silhouette*, lalu membedakan titik yang merupakan bagian dari objek dan bukan dari objek. Untuk setiap titik yang merupakan bagian dari *visual hull*, akan diletakkan sebuah *voxel* yang akan memiliki lokasi yang sama dalam *voxel cube*, sehingga objek yang diinginkan dapat terbentuk.

3. DESAIN SISTEM

3.1 Garis Besar Sistem

Secara garis besar sistem yang akan dibuat akan melalui proses seperti pada Gambar 3.

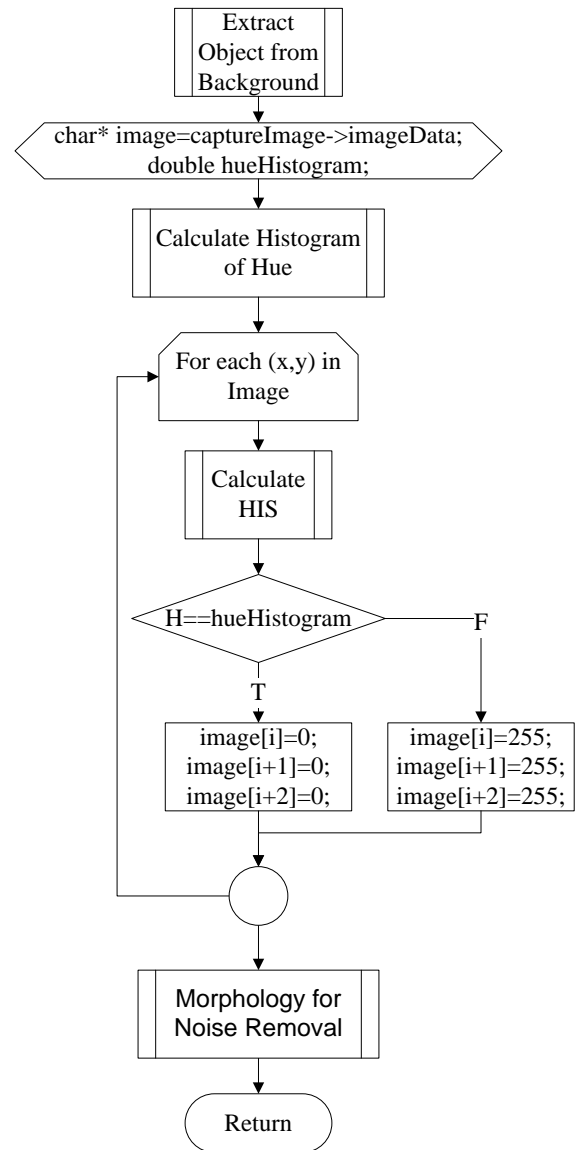


Gambar 3. Flowchart Kerja Garis Besar Sistem

Dapat dilihat pada Gambar 3, langkah pertama yang dilakukan adalah melakukan input *image* yang nantinya akan diambil gambar *silhouette* dari gambar tersebut. Apabila semua sisi sudah diambil gambar, maka dapat dilakukan pengecekan terhadap gambar *silhouette* sehingga dapat dimasukkan kedalam *voxel cube*.

3.2 Flowchart Proses Extract Object from Background

Langkah ekstraksi yang dilakukan oleh Sistem dapat dilihat pada Flowchart pada Gambar 4.

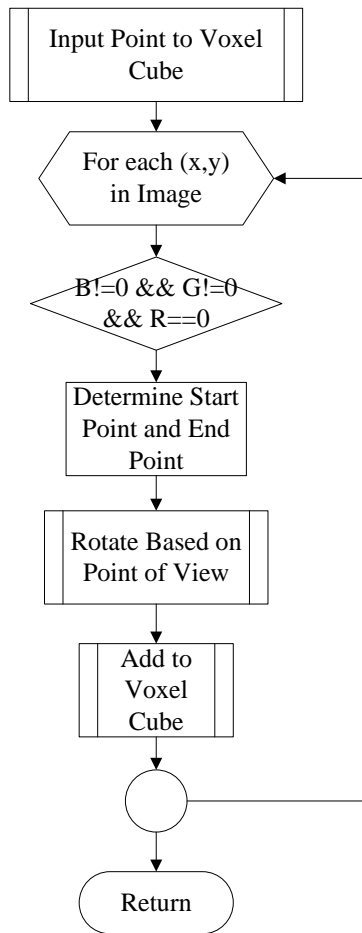


Gambar 4. Flowchart Ekstraksi Objek

Proses ini merupakan proses perhitungan semua nilai warna pada gambar, agar dapat diperoleh objek, dengan cara membedakan warna *background* dan warna objek.

3.3 Flowchart Proses Input Point to Voxel Cube

Proses ini melakukan *input* titik yang sudah terdeteksi sebagai objek, dengan membentuk proyeksi 3D dari *silhouette* objek.



Gambar 5. Flowchart Input Point to Voxel Cube

4. IMPLEMENTASI SISTEM

4.1 Implementasi Aplikasi

Pembuatan aplikasi ini menggunakan bahasa pemrograman C++ dengan menggunakan Microsoft Visual Studio 2010 sebagai *Integrated Development Environment (IDE)*. Alasan penggunaan bahasa C++ adalah kemampuan dalam menggunakan *library* yang ada untuk mendukung pembuatan aplikasi, yaitu *library OpenCV* dan *OpenGL*.

4.2 Fungsi yang digunakan

Dalam pembuatan aplikasi ini, terdapat beberapa *class* dan fungsi yang mendukung kerja dari aplikasi untuk mendapatkan objek 3D yang akan direkonstruksi. *Class - class* yang digunakan merupakan *class cPoint*, *cMatrix*, dan *cVoxel*. *Class cPoint* dan *cVoxel* digunakan untuk menyimpan titik yang diperoleh dari setiap pengambilan gambar, sedangkan *cMatrix* merupakan *class* yang dibuat untuk menghasilkan *matrix* rotasi yang akan merotasi setiap titik yang ada pada *class cVoxel*. Berikut fungsi yang digunakan:

- *initiateBackground*. Dalam fungsi ini, gambar yang diambil akan mencari nilai *hue* yang akan digunakan sebagai


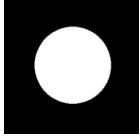
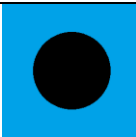
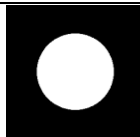
acuan, dengan cara mencari *hue* yang paling banyak muncul dari semua titik pada gambar, sehingga dapat

- diketahui apakah titik tersebut merupakan *background* atau objek.
- *subtractObject*. Dalam fungsi ini, akan dilakukan ekstraksi objek pada gambar dengan cara membandingkan nilai *hue*, yang sudah diperoleh dari fungsi *initiateBackground*, dengan nilai *hue* pada gambar, sehingga warna *background* dapat diubah menjadi hitam, dan objek diubah menjadi putih.
- *addToVoxel*. Fungsi ini adalah fungsi yang menginputkan semua titik yang ada pada gambar kedalam *cVoxel* dengan cara membedakan nilai *RGB* pada gambar yang sudah diproses didalam *subtractObject*, dengan cara menambahkan status dari titik yang berada pada *cVoxel*.
- *thresholdVoxelCube*. Didalam fungsi ini, semua titik yang ada dalam *cVoxel* akan dilakukan *thresholding* dengan cara menentukan nilai status agar dapat diketahui apakah titik tersebut merupakan bentuk asli dari objek, atau titik tersebut merupakan proyeksi 3D dari suatu sudut pengambilan gambar.
- *initiateOpenGL*. Fungsi ini merupakan proses inisialisasi dari *library OpenGL* yang digunakan untuk menampilkan hasil rekonstruksi dari objek.
- *renderPoint*. Fungsi ini merupakan proses dari menampilkan semua titik yang terdapat pada *voxel cube* yang merupakan bentuk dari objek.

5. PENGUJIAN SISTEM

5.1 Pengujian Terhadap Pengambilan Gambar

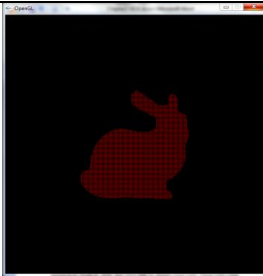
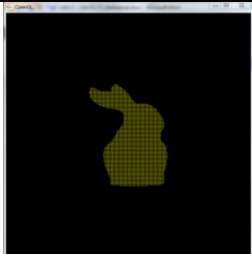
Pengujian terhadap pengambilan gambar dari objek meliputi pengujian terhadap *image segmentation* dan *opening*, pada pengujian ini, dilakukan ekstraksi gambar dari objek dengan menggunakan berbagai warna sebagai latar untuk menguji *image segmentation*. Gambar 4, Hasil ekstraksi dari objek menggunakan *image segmentation*.

| Warna Background | Hasil | |
|------------------|---|---|
| Hijau |  |  |
| Biru |  |  |

Gambar 2. Pengujian warna objek terhadap ekstraksi

5.2 Pengujian Terhadap Rekonstruksi Objek

Pengujian terhadap rekonstruksi meliputi jumlah gambar yang digunakan dan nilai *threshold* yang digunakan dalam seleksi titik.

| Objek dan Sudut Pandang Gambar | Hasil Rekonstruksi dari objek |
|--------------------------------|---|
| 0' Bunny |  |
| 90' Bunny |  |

6. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Proses *voting-based voxel carving* yang dilakukan belum dapat menghasilkan bentuk objek yang diinginkan, ini dikarenakan masih terdapat beberapa *noise* yang ada, dan masih terdapat kekurangan dari penampilan objek 3D yang akan dihasilkan.
- Gambar yang digunakan sebagai input dari *voxel cube* sudah bebas dari *noise* dan dapat digunakan sebagai input dari *voxel cube*.
- Proses *voting-based voxel carving* memakan waktu yang cukup lama dalam menampilkan bentuk 3D dan dalam proses *input* kedalam *voxel cube*.

7. REFERENSI

- [1] Cheung, K. M., Baker, S., & Kanade, T. (2005). Shape-From-Silhouette Across Time Part I: Theory and Algorithms. *International Journal of Computer Vision*, 62(3), 221-247.
- [2] Kuzu, Y., & Rodehorst, V. (2001). Volumetric modeling using shape from silhouette. *Proc. the 4th Turkish-German Joint Geodetic Days*, 469-476..
- [3] R. C. Gonzalez and R. E. Woods. (2002). *Digital image processing*, 2nd ed. Chapter 9.
- [4] Wikipedia. (2008). *Mathematical Morphology*. Retrieved July 03, 2013, from http://en.wikipedia.org/wiki/Mathematical_morphology
- [5] Kuzu, Y. (2002). Photorealistic object reconstruction using color image matching. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(5), 169-174.