

REKONSTRUKSI OBYEK 3D DARI GAMBAR 2D DENGAN METODE *GENERALIZED VOXEL COLORING*

Rudy Adipranata¹, Resmana Lim², Anton Setiawan
Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya
Email : rudy@petra.ac.id¹, resmana@petra.ac.id²

Abstrak

Pada saat ini hampir semua *game*, animasi dan iklan sudah berbasis 3D, sehingga diperlukan suatu perangkat lunak yang mampu membuat obyek 3D dengan cepat dan mudah. Perangkat lunak yang ada sekarang adalah perangkat lunak untuk mendesain obyek 3D yang dilakukan oleh manusia, yang menghasilkan obyek tidak bersifat *photorealistic*.

Pada penelitian ini dibuat sebuah perangkat lunak untuk merekonstruksi obyek dalam tiga dimensi yang berasal dari kumpulan gambar dua dimensi secara *photorealistic* dengan menggunakan metode *Generalized Voxel Coloring*. Metode ini melakukan rekonstruksi obyek dengan bantuan *item-buffer* untuk mencari *voxel-voxel* yang berkoresponden berdasarkan warna. Dari kumpulan *voxel* yang saling berkoresponden tersebut kemudian dihitung standar deviasi warna kumpulan tersebut, apabila standar deviasi warna-warna tersebut kecil maka dapat diasumsikan *voxel* tersebut adalah obyek. Proses dilakukan secara berulang untuk semua *voxel* hingga didapatkan semua *voxel* yang membentuk obyek 3D.

Hasil dari proses rekonstruksi dapat membentuk obyek 3D yang secara visual dapat dilihat cukup baik, tetapi keakuratan obyek hasil terhadap obyek asli tergantung dari keakuratan proses kalibrasi kamera.

1. PENDAHULUAN

Dewasa ini animasi 3D seperti *game*, film dan lain-lain sudah mulai berkembang dan banyak diminati oleh banyak orang, baik anak-anak maupun orang dewasa. Apabila dulu semua animasi masih berbasis 2D, sekarang hampir semua *game*, film kartun, bahkan iklanpun mulai menggunakan animasi 3D.

Tetapi hingga sampai saat ini masih belum ada kamera ataupun *scanner* yang dapat digunakan untuk mengambil gambar langsung secara 3D, sehingga banyak *developer software* terkenal yang berusaha membuat suatu *software* yang dapat membuat obyek 3D dengan baik seperti 3D Max, Maya dan lain-lain. Namun *software* 3D yang ada tersebut memerlukan keahlian manusia untuk dapat membuat suatu obyek 3D yang menyerupai aslinya dan membutuhkan waktu pembuatan yang cukup lama.

Karena itu pada penelitian ini dikembangkan perangkat lunak yang dapat menghasilkan obyek 3D yang bersifat *photorealistic* yang berasal dari kumpulan gambar 2D. Adapun bidang penelitian untuk membentuk obyek 3D yang bersifat *photorealistic* ini masih terus berkembang dan belum ada sebuah metode yang benar-benar baik untuk segala hal. Pada penelitian ini, metode yang dikembangkan adalah metode *Generalized Voxel Coloring* (GVC) yang dikemukakan oleh Slabaugh [1], dimana metode ini merupakan pengembangan dari metode *Voxel Coloring* yang dikemukakan oleh Seitz dan

Dyer [2]. Metode GVC ini dapat menghasilkan obyek 3D yang bersifat *photorealistic* dan dapat mengatasi masalah *occlusion* yang terjadi pada proses rekonstruksi.

2. KALIBRASI KAMERA

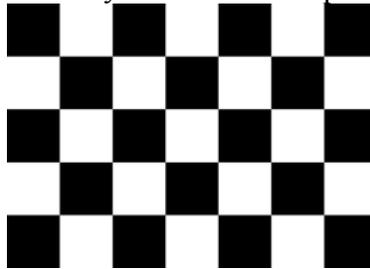
Sebelum dapat melakukan proses rekonstruksi 3D, kalibrasi kamera perlu dilakukan untuk mendapatkan beberapa parameter dari kamera digital yang digunakan. Parameter ini meliputi matriks intrinsik dan ekstrinsik yang nantinya digunakan untuk melakukan perhitungan, sehingga dapat ditentukan letak dari suatu benda dalam ruang 3 dimensi.

Parameter dari matriks intrinsik ini terdiri dari 4 unsur yaitu :

- nilai fokus kamera, yaitu jarak antara lensa kamera dengan bidang gambar.
- titik pusat proyeksi, yaitu lokasi titik tengah gambar dalam *pixel* koordinat.
- ukuran *pixel* efektif.
- koefisien distorsi, yaitu koefisien tingkat kelengkungan lensa meliputi radial dan tangensial distorsi.

Sedangkan parameter ekstrinsik ini terdiri dari dua buah matriks, yaitu matriks translasi dan matriks rotasi. Parameter ekstrinsik ini menggambarkan orientasi posisi dari kamera terhadap sistem koordinat sebenarnya dalam ruang 3 dimensi atau *world coordinate*.

Untuk melakukan proses kalibrasi kamera digunakan papan catur (*chessboard pattern*) dengan ukuran sisi perkotaknya 1 atau 3 cm seperti pada Gambar 1.



Gambar 1. *Chessboard Pattern*

Langkah pertama yang harus diambil sebelum melakukan kalibrasi kamera adalah mengambil beberapa gambar dari *chessboard pattern*, kemudian melakukan pencarian titik-titik *chessboard pattern* pada gambar. Dengan mengetahui letak titik-titik pada *world coordinate* dan *pixel coordinate* maka dapat dicari matriks yang menghubungkan keduanya. Matriks inilah yang disebut sebagai matriks intrinsik dan ekstrinsik. Metode untuk mencari matriks ini menggunakan metode dari Zhang [5] yang telah diimplementasikan dalam fungsi-fungsi pada *OpenCV*.

2.1. Parameter Intrinsik Kamera

Setelah dilakukan proses kalibrasi maka didapatkan matriks intrinsik dan ekstrinsik dari kamera. Matriks intrinsik (W) terdiri atas fokus kamera, titik tengah proyeksi atau *principal point*, ukuran pixel efektif dan koefisien distorsi radial dan tangensial. Matriks intrinsik adalah matriks yang menyatakan geometri dari kamera itu sendiri.

$$W_{\text{int}} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$K = [k1 \quad k2 \quad p1 \quad p2] \quad (2)$$

f_x, f_y = fokus kamera

C_x, C_y = titik tengah

$k1, k2$ = distorsi radial

$p1, p2$ = distorsi tangensial

2.2. Parameter Ekstrinsik Kamera

Selain matriks intrinsik, kalibrasi kamera juga menghasilkan matriks ekstrinsik yang merupakan orientasi posisi dari kamera terhadap *world coordinate*. Matriks ekstrinsik ini terdiri atas matriks rotasi (R) dan matriks translasi (T).

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3)$$

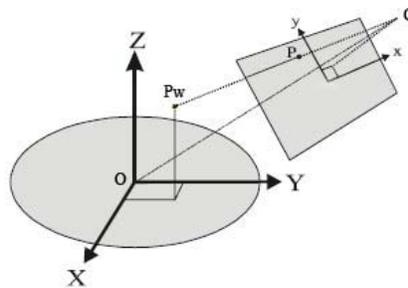
$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (4)$$

Matriks rotasi sebenarnya adalah gabungan dari rotasi terhadap sumbu X, Y dan Z yang masing-masing berotasi dengan sudut (ψ, θ, ϕ).

$$R = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix} \quad (5)$$

3. PROYEKSI GAMBAR

Perhitungan proyeksi suatu titik pada ruang 3D menjadi titik pada gambar 2D dilakukan dengan perkalian titik 3D terhadap parameter kamera yang terdapat pada matriks intrinsik dan ekstrinsik. Ruang 3D dimana benda berada disebut dengan *world coordinate*, sedangkan ruang 2D dimana gambar diambil dengan kamera disebut dengan *image coordinate*. Berikut ini digambarkan hubungan *world coordinate* dengan *image coordinate* serta formula untuk menghitung proyeksinya.



Gambar 2 Hubungan Antara *World Coordinate* dan *Image Coordinate*

Formula untuk menghitung proyeksi dari *world coordinate* (P_w) ke *image coordinate* (P) adalah sebagai berikut:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx & 0 & Cx \\ 0 & fy & Cy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

P = *Image coordinate* (u,v)

P_w = *World coordinate* (X,Y,Z)

R = Matrik rotasi (r_1 - r_9)

T = Matrik translasi (t_1 - t_3)

4. GENERALIZED VOXEL COLORING

Generalized Voxel Coloring (GVC) adalah suatu metode untuk merekonstruksi obyek 3D dari kumpulan gambar 2D yang dikemukakan oleh Slabaugh [1] yang merupakan pengembangan dari metode *Voxel Coloring* [2]. Secara umum metode GVC dibagi menjadi 2 macam yaitu GVC dengan *item buffer* (GVC-IB) dan GVC dengan *layered depth images* (GVC-LDI). Perbedaan dari kedua metode ini adalah penggunaan memori dan waktu proses. Pada metode GVC-IB memori yang digunakan lebih sedikit namun prosesnya akan lebih lama. Sedang pada GVC-LDI, memori yang digunakan lebih besar namun waktu proses akan lebih cepat dibandingkan metode GVC-IB. Pada penelitian ini metode yang digunakan adalah metode GVC-IB.

Metode GVC-IB dapat dijabarkan pada pseudocode berikut :

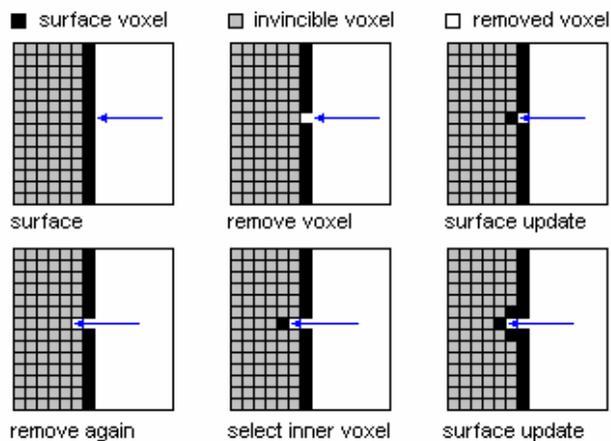
```

initialize SVL
for every voxel V
  carved(V) = false
loop {
  visibilityChanged = false
  compute item buffers by rendering voxel s on SVL
  for every voxel V ∈ SVL {
    compute vis(V)
    if (consist(vis(V)) = false) {
      visibilityChanged = true
      carved(V) = true
      remove V from SVL
      for all voxels N that are adjacent to V
        if (carved(N) = false and N ∉ SVL)
          add N to SVL
    }
    else
      color(V)
  }
  if (visibilityChanged = false) {
    save voxel space
    quit
  }
}

```

Dari *pseudocode* tersebut terlihat bahwa langkah pertama yang dilakukan adalah inialisasi *Surface Voxel List* (SVL). SVL adalah suatu *link list* yang berisi informasi *voxel* yang akan diproses. Masing masing *list* dalam SVL berisi koordinat *voxel* dan

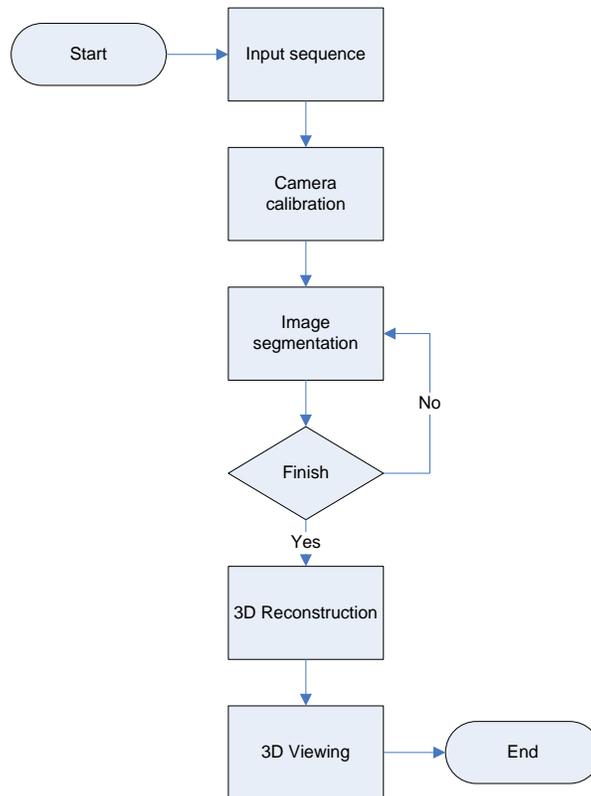
warna *voxel* dalam RGB. SVL diisi dengan *voxel-voxel* yang berada pada permukaan kotak yang akan diproses. Kemudian setiap *voxel* dalam SVL akan diproses, bila suatu *voxel* bukan merupakan obyek maka *voxel* tersebut akan di-*carved* atau dihapus dari SVL dan digantikan dengan *voxel* lain yang berada disekitar *voxel* tersebut. Dengan syarat *voxel* tersebut belum pernah ada pada SVL. Proses ini akan dilakukan secara terus menerus hingga tidak ada lagi *voxel* yang dapat di-*carved* atau dengan kata lain semua *voxel consistent*. Inti dari proses rekonstruksi ini dapat diasumsikan seperti halnya proses pemahatan patung, pertama-tama patung adalah sebuah bongkahan kayu padat yang berbentuk kubus atau silinder. Kemudian untuk menjadikan suatu bentuk patung maka sedikit demi sedikit dari kayu tersebut dipahat, proses pemahatan dilakukan terus menerus hingga tidak ada lagi bagian dari kayu tersebut yang bias dipahat, atau dengan kata lain sudah membentuk suatu patung. Contoh proses pemahatan (*carved*) dapat dilihat pada gambar 3.



Gambar 3. Proses *Voxel Carving*

5. IMPLEMENTASI DAN PENGUJIAN

Pada penelitian ini, aplikasi dibuat dengan menggunakan bahasa pemrograman C++ dan bantuan *QT Library* pada sistem operasi *Linux SUSE 9.3*. Sedangkan spesifikasi komputer yang digunakan adalah *CPU 1,6 Ghz Intel Pentium 4*, memori *512 Mb*, *VGA Card Nvidia GeForce 256*. Berikut ini adalah *flowchart* sistem secara keseluruhan.



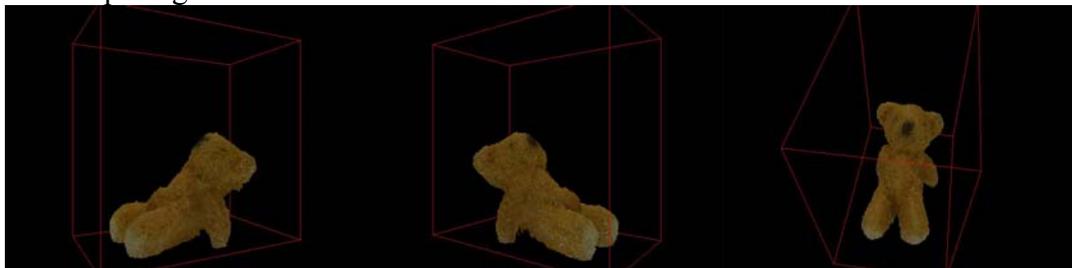
Gambar 4. *Flowchart* Sistem

Adapun input yang digunakan pada pengujian sistem adalah seperti terlihat pada gambar 5 berikut. Input tersebut diambil dengan menggunakan kamera digital yang diletakkan di sebelah atas obyek yang dimaksud. Pengambilan gambar dilakukan sebanyak 12 kali dengan sudut pandang yang berbeda.



Gambar 5. Tiga dari Total 12 Gambar Input

Dari gambar input tersebut diproses dan diperoleh hasil obyek tiga dimensi seperti terlihat pada gambar 6.



Gambar 6. Hasil Rekonstruksi Tiga Dimensi

6. KESIMPULAN

Dari hasil implementasi dan pengujian dapat ditarik kesimpulan yaitu :

- Jumlah gambar sangat mempengaruhi hasil rekonstruksi. Semakin banyak gambar referensi dalam berbagai sudut pandang maka hasil rekonstruksi akan semakin bagus. Namun semakin banyak gambar dapat membuat proses semakin lambat.
- Besar ukuran *voxel* sangat berpengaruh pada hasil rekonstruksi. Semakin kecil ukuran *voxel* maka akan semakin halus bentuk dari obyek 3D hasil rekonstruksi namun semakin lama juga waktu prosesnya.
- Hasil kalibrasi kamera sangat berpengaruh untuk menentukan letak atau posisi dari suatu titik dalam *world coordinate* ke *image coordinate* pada gambar, sehingga dibutuhkan proses kalibrasi kamera yang sangat teliti.
- Perbandingan hasil rekonstruksi dengan obyek aslinya dilihat dari ukurannya sudah hampir menyerupai bentuk aslinya, selisih ukuran antara obyek asli dan obyek rekonstruksi antara 0 mm sampai 10 mm.

DAFTAR PUSTAKA

- [1] Gregory G. Slabaugh. *Novel Volumetric Scene Reconstruction Methods for New View Synthesis*. USA: Georgia Institute of Technology, 2002
- [2] Steven M. Seitz dan Charles R. Dyer, *Photorealistic Scene Reconstruction by Voxel Coloring*, Proc. Computer Vision and Pattern Recognition Conf, pp.1067-1073, 1997
- [3] Rudy Adipranata, Hwang-Kyu Yang and Tae-Soo Yun. *Implementation of Photorealistic 3D Object Reconstruction Using Voxel Coloring*. 19th Conference of Korean Information Processing Society, pp. 527-530, 2003
- [4] Kutulakos, N. And Seitz, M. *A Theory of Shape by Space Carving*. Proceedings of the Seventh IEEE International Conference on Computer Vision, vol 1, pp. 307-314, 1999
- [5] Zhang, Zhengyou. *Flexible Camera Calibration by Viewing a Plane from Unknown Orientation*. In International Conf. on Computer Vision, 1999
- [6] Open CV Documentation by Intel Research. <<http://sourceforge.net/projects/opencv/>>, akses : 3 Maret 2005
- [7] QT Documentation by QT Developer. <<http://www.trolltech.com> >, akses : 10 Maret 2005
- [8] Roger Y. Tsai. *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Camera and Lenses*. IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987
- [9] Andrew C. Prock and Charles R. Dyer. *Towards Real-Time Voxel Coloring*. Proc. Image Understanding Workshop, pp. 315-321, 1998