Voxel Coloring 을 이용한 3D 오브젝트 모델링

Rudy Adipranata, 양황규, 윤태수
동서대학교 소프트웨어 전문대학원
e-mail : rudyap@yahoo.com

Implementation of Photorealistic 3D Object Reconstruction Using Voxel Coloring

Rudy Adipranata, Hwang-Kyu Yang, Tae-Soo Yun
Graduate School of Software, Dong-Seo University

Abstract

In this paper, we implemented the voxel coloring method to reconstruct 3D object from synthetic input images. Then compare the result between using standard voxel coloring and using coarse-to-fine method. We compared using different voxel space size to see the difference of time processing and the result of 3D object.

Photorealistic 3D object reconstruction is a challenging problem in computer graphics. 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 in special order. Also, there is an extension of voxel coloring method for decreasing the amount of processing time called coarse-to-fine method. This method works using low resolution instead of high resolution as input and after processing finish, apply some kind of search strategy.

1. Introduction

Nowadays, automatic reconstruction of photorealistic three dimensional object from multiple images is a challenge in computer graphics. Almost all of real world scene consists of geometric complex that difficult to reconstruct. There are some approaches for addressing that problem. First approach is Image Based Rendering. Image Based Rendering generates synthetic image from photographs instead of geometric primitive. Another approach is Image Based Modeling, which produce three dimensional model from one or more images.

One approach in Image Based Rendering field for addressing the scene reconstruction problem has been presented by Seitz and Dyer called voxel coloring [1]. This method similar to Collins' Space-Sweep [4] that performs an analogous scene traversal for a plane in the scene volume, then accumulated vote for points on the plane that project to edge features in the image. But this method doesn't provide general solution for occlution problem.

Voxel coloring method defines the problem to be solved as color reconstruction problem instead of shape reconstruction problem. The basic idea of Voxel coloring is discretizing voxel space into set of voxels and then traverse and projects each voxel into each image in which it is visible. Later, check if the colors in image is consistent, mark voxel with that color, else mark voxel as empty.

2. Voxel Coloring Method

Voxel coloring traverses three dimensional space in "depth-order" to identify voxels that have a unique coloring, constant across all possible interpretations of the scene. A voxel is said to be consistent if it projects to the same color on all the images from where it is visible. And to avoid multiple passes through the voxel space, this method uses a constraint known as the ordinality visibility constraint. Because of the constraint, the voxel space is partitioned into layers. Then, each layer is traversed one by one from the location nearest to the camera position.

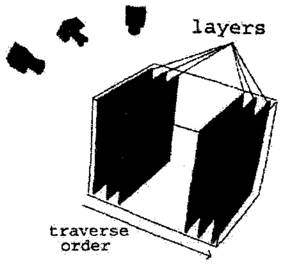
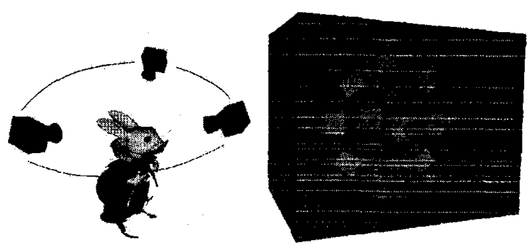


Figure 1. Voxel coloring traverse order

The key problem of voxel coloring is how to assign colors to voxels in 3D volume and still have consistency with a set of basis images. Formally, the voxel coloring problem is : given a set of basis image $I_0, \ldots, I_n$, of a static Lambertian scene and a voxel space V, determine a subset $S \subset V$ and coloring *color*(V, S), such that $S \in N$ where N is set of all consistent scenes.

2.1 Ordinal Visibility Constraint [1]

The ordinal visibility constraint enables the identification of the set of color invariants as a limit point of N. Let P and Q be scene points and I be an image from a camera centered at C. P occludes Q if P lies on the line segment $\overline{CQ}$. The input camera must be positioned so as to satisfy the constraint : There exists a norm $\| \cdot \|$ such that for all scene points P and Q, and input images I, P occludes Q in I only if $\| P \| < \| Q \|$.



(a) (b)

Figure 2. (a) Inward looking, camera position above the scene, (b) Outward looking, camera position inside the scene, locate around sphere.

2.2 Color Invariants [1]

Scene reconstruction is complicated because a set of images can be consistent with more than one rigid scene. A voxel may be a part of two consistent scene but the color are different, or a voxel contained in one consistent scene may not be contained in another scene. So to recover the intrinsic scene information, the only way is invariants, the properties that are satisfied in every scene.

A voxel V is a color invariant with respect to a set of images if, V is contained in a scene consistent with the images, and for every pair of scenes S and S', $V \in S \quad S'$ implies color(V,S) = color (V,S').

2.3 Voxel Coloring Algorithm

The pseudocode for voxel coloring method is as follows :

```
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 < treshold) {
            color this voxel
            marked pixel related with this voxel
        }
    }
}
```

All images must have no background before compute the projection from voxel. Color consistency is define as the standard deviation in the color of the set of pixels in all images, on which a voxel projects. A voxel will be colored only if the consistency is less than threshold value.

3. Coarse-To-Fine Method

Prock and Dyer present coarse-to-fine method [2], that is an extension of voxel coloring in order to make processing faster. Using coarse-to-fine method, the processing is used low resolution information as input to create a higher resolution result. This method will be faster because of using low resolution as beginning process, so the processing is focused only on important regions and concentrate only on locations that contain colored voxels.

There are some issues that must be considered using this method. The first issue is the result may be incorrect because of low resolution input. This because there are some surface can't be detected in low resolution. To addressing this issue, the voxels that remain uncolored at low resolution but actually contain small opaque sub-regions, should not be eliminated. So the voxels later can be considered when process in higher resolution.

Another issue is because the resolution is reduced, voxels with smaller occupied sub-volumes may be missed as false negatives. To addressing this issue, there is some kind of search strategy must be implemented that finds the missing voxels.

3.1 Searching for False Negatives

For searching for voxel that uncolored for mistakenly, we can use some kind of heuristic. The seach strategy has been choosen is a nearest neighbor search. All voxels within some 1-norm neighborhood of the original low resolution set is added to the set. These voxels are then subdivided into octants and then traversed in the standard layered order and colored according to the original algorithm.

4. Camera Calibration and Projection

Camera calibration is the process of determining the intrinsic and extrinsic parameters of camera. Intrinsic parameters include the internal camera geometric and optical parameters such as focal length, lens distortion coefficient, uncertainty scale factor, and the computer image coordinate for the origin in the image plane. The extrinsic parameters include the positional orientation of the camera with respect to the world coordinate system. These include the three Euler angles and three translation parameters.

Camera's projection matrix H is a 3x4 matrix that describes how a point P = (X,Y,Z) in 3D space projects to a point p = (x,y) in an image. Using homogeneous representation, the formula is [3] :

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

s is a scale factor. The projection matrix (H) can be composed using the camera calibration. The formula is

$$H = K\,[\,R \mid -RT\,] \qquad (2)$$

Where the matrix K is 3x3 matrix as follows :

$$K = \begin{bmatrix} \frac{f}{d_x} & 0 & c_x \\ 0 & \frac{f}{d_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

d_x = horizontal size of a pixel d_y = vertical size of a pixel

c_x = horizontal center of image c_y = vertical center of image

f = focal length

R matrix is 3x3 rotation matrix that describes the orientation of the camera. R has three degrees of freedom that describes as (ψ, θ, φ) that represent of x, y and z respectively.

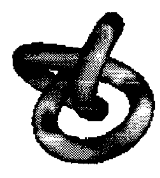
$$R = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \qquad (4)$$

and last matrix $T = [T_x, T_y, T_z]^T$ is the position of the camera's center of projection in world coordinates.

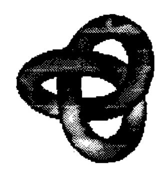
5. Experimental Result

In this paper, we try to implement the voxel coloring method using OpenGL as a tools. We create synthetic input image and make 3D reconstruction from that. We implement the standard voxel coloring method and compare it with coarse-to-fine method. The synthetic input image we create has a black background, because voxel coloring required that input images must have no background.

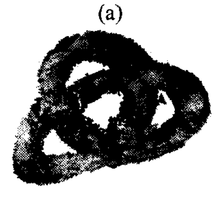
Following pictures are two of the eight synthetic input images that we use, and also we show the result using standard voxel coloring and coarse-to-fine method.



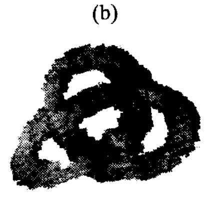
(a)



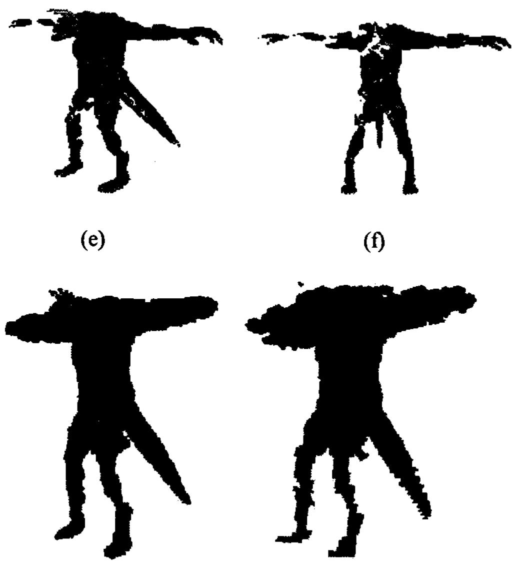
(b)



(c)



(d)



(e) (f)

(g) (h)

Figure 3. (a), (b) , (e), (f) Two of eight input images,
(c), (g) Result using voxel coloring method,
(d), (h) Result using coarse-to-fine method

In the following table we present the result of comparison amount of time required and number of reconstruction voxels between standard voxel coloring method and coarse-to-fine method.

Table 1. Comparison between standard voxel coloring and coarse-to-fine method.

Voxel Space Size	Standard Voxel Coloring		Coarse-to-Fine		Difference (%)
	Time *(seconds)*	Number *of Voxels*	Time *(seconds)*	Number *of Voxels*	
$(50)^3$ voxels	2.18700	18,479	2.36000	56,382	-7.9103
$(100)^3$ voxels	8.23500	42192	3.03100	54,017	63.1937
$(150)^3$ voxels	17.95300	42,192	5.65600	56,382	68.4955
$(200)^3$ voxels	35.87500	42,192	24.01600	56,959	33.0564

Voxel space size must be greater than size of the object to be reconstruct. In our experiment using figure 3(a,b), $(50)^3$ voxel space size can't be used because too small, so there are many voxels can't be reconstruct.

By observe the result table, optimum voxel space size in this experiment is between $(100)^3$ and $(150)^3$ voxels using coarse to fine method.

6. Conclusions

Using voxel coloring method, we can success reconstruct photorealistic 3D object, but the system must has excellent camera calibration. To speed up the time require for reconstruction can use coarse-to-fine method, because this method apply to the low resolution scenes that required little computation for processing, and then can create high resolution by implement some kind of search strategy.

References

[1] Steven M. Seitz and Charles R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *Proc. Computer Vision and Pattern Recognition Conf.* pages 1067-1073, 1997

[2] Andrew C. Prock and Charles R. Dyer. Towards Real-Time Voxel Coloring. In *Proc. Image Understanding Workshop,* pages 315-321, 1998

[3] *Gregory G. Slabaugh. Novel Volumetric Scene* Reconstruction Methods for New View Synthesis. In *A Thesis Presented to The Academic Faculty, School of Electrical and Computer Engineering, Georgia Institute of Technology,* 1992.

[4] Collins, Robert. T. A Space-Sweep Approach to True Multi-Image Matching. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 358-363, 1996.

[5] Akihiro Katayana, Koichiro Tanaka, Takahiro Oshino, and Hideyuki Tamura. A Viewpoint Dependent Stereoscopic Display Using Interpolation of Multi-Viewpoint Images. In *Proc. SPIE Vol. 2409A,* pages 21-30, 1995.

[6] Steven M. Seitz and Charler R. Dyer. Complete Structure From Four Point Correspondence. In *Proc. Fifth Int. Conf. on Computer Vision,* pages 330-337, 1995.

[7] Steven M. Seitz. Image-Based Transformation of Viewpoint and Scene Appearance. In *A Dissertation Submitted in Partial Fulfillment of The Requirements For The Degree of Ph.D at the University Of Wisconsin-Madison,* 1997.