

Fast Octree Construction Endowed with an Error Bound Controlled Subdivision Scheme

HONG-LONG CHOU AND ZEN CHEN

Department of Computer Science and Information Engineering

National Chiao Tung University

Hsinchu, 300 Taiwan

E-mail: {hlchou; zchen}@csie.nctu.edu.tw

In the conventional octree construction method any grey octant is subdivided recursively until all its descendant octants are no longer grey. The subdivision process is executed no matter how small the white portion in the grey octant is. When there are many grey octants, each containing a fairly small white portion, then the huge increase in the resulting descendant nodes due to the subdivisions may cause the construction process to terminate due to an insufficient amount of memory. In such a case the subdivisions made are not worthwhile. In this paper, we shall make effective use of octant subdivision to improve the overall system performance. A new octree construction method is proposed with a novel subdivision strategy such that only those octants with a projection error exceeding a pre-specified error bound will be subdivided. Furthermore, we also present a fast way to compute the 2D projection of octant vertices and a new intersection test to reduce overall processing time. Computer simulations are conducted which show that the new method performs better than the conventional method in terms of memory space and computation time. Moreover, a theoretical analysis of the performance of the new method is included.

Keywords: octree construction, subdivision strategy, octant projection, cross ratio, distance map, construction quality, memory space, computation time

1. INTRODUCTION

An octree is a volumetric representation of 3D object's geometric model. It is well known in the field of computer vision that an octree can be constructed from silhouettes taken from different viewpoints of an object [1, 16]. The constructed object geometric model finds many applications including robotic system, computer aided design, virtual reality, and object tracking. The conventional construction methods can be categorized into two classes: 3D space approach [3, 5] and 2D space approach [6, 16]. Both approaches recursively subdivide a partially occupied octree octant into smaller octants until all the generated octants are entirely inside or outside the object. The overlap between an octant and the object is determined by an intersection test. The above two approaches differ in their intersection tests. In the 3D space approach the octant under examination is tested against the conic view volume formed by the individual silhouette and the center of projection for each viewpoint. In the 2D space approach the projected

Received March 19, 2004; revised April 20, 2004; accepted May 18, 2004.
Communicated by H. Y. Mark Liao.

image of the octant is tested against the silhouette for each viewpoint. Generally speaking, the 2D space approach is performed in a space with a lower dimension, and so it is more efficient than the 3D space approach [6]. Since the number of octants grows exponentially with subdivision level, an upper bound is usually imposed on the number of levels to avoid the insufficient memory problem. However, a larger value of subdivision levels leads to a better construction result. It is generally difficult to make a good balance between memory and construction quality.

A close look at the octant subdivisions reveals that the subdivided octants have varying projection errors; some may be very large and some may be rather small. We shall consider making effective use of the octant subdivisions to improve the overall system performance. In this paper, we propose a new subdivision strategy which is governed by the degree of overlapping between the generated octant and the object. This degree of overlapping will be measured by the maximum 2D projection error of the projected octant image relative to the object silhouette for all viewpoints. Those octants with a projection error exceeding the pre-specified upper bound will be subdivided recursively until all the octants satisfy the maximum constraint imposed on the projection error. Furthermore, we will also present a fast computation of the 2D projection and a new intersection test to reduce the computer processing time. A summary of contributions of the paper is given below:

- (1) The ultimate octree subdivision level of the new construction method is not fixed. Instead, it is controlled by a projection error bound.
- (2) The construction quality can be evaluated via an exclusive OR projection error.
- (3) Fast computation of 2D octant projections to the image planes is provided.
- (4) Fast 2D intersection test utilizes the distance maps which were generated for the multiple silhouettes in advance.
- (5) Analysis of the properties of the proposed method reveals its superiority over the conventional method in terms of memory and computation time.

The paper is organized as follows. Section 2 presents a fast 2D octant projection computation. Section 3 describes the use of distance maps generated from multiple silhouettes and the approximation of an octant projection image by a circle. Section 4 describes the new subdivision process and the evaluation of the quality of the final octree constructed for the object. Section 5 presents experimental results. Section 6 analyzes the new method. Section 7 gives our conclusions.

2. FAST OCTANT PROJECTION COMPUTATION USING CROSS RATIO AND VANISHING POINT

2.1 Computation of Perspective Projections of Octant Vertices

When a parent octant is subdivided into 8 child octants, the projection images of the child octants involves $8 \times 8 = 64$ vertices in the recursive construction process. However, since each child octant shares some of its 8 vertices with its siblings, there are only 27

(instead of 64) distinct vertices, as shown in Fig. 1. We shall compute the perspective projections of the 27 distinct vertices onto the image plane before dividing them into 8 octant projection images.

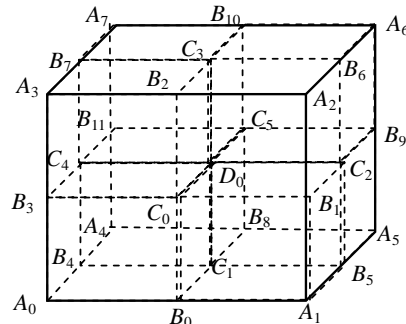


Fig. 1. The 27 distinct vertices involved in the subdivision of a parent octant including eight A-type vertices (A_0, A_1, \dots, A_7), twelve B-type vertices (B_0, B_1, \dots, B_{11}), six C-type vertices (C_0, C_1, \dots, C_5), and one D-type vertex D_0 .

There are four types of sub-octant vertices in regard to their parent octant vertices:

- (1) A-type vertices $\{A_0, A_1, \dots, A_7\}$ belong to the parent octant. Their perspective projection data will be passed down to the subsequent octant subdivisions.
- (2) B-type vertices $\{B_0, B_1, \dots, B_{11}\}$ are located on the separate lines defined by pairs of A-type vertices. Their perspective projections onto the image plane can be calculated quickly from A-type vertices through the use of the cross ratio and vanishing points, as described later.
- (3) C-type vertices $\{C_0, C_1, \dots, C_5\}$ fall on the separate lines defined by pairs of B-type vertices. Their perspective projections onto the image plane can also be calculated quickly from B-type vertices through the use of the cross ratio and vanishing points.
- (4) D-type vertex D_0 falls on the line defined by two C-type vertices. Its perspective projection onto the image plane can be calculated from the C-type vertices through the use of the cross ratio and vanishing point.

In the following the concepts of cross ratio and vanishing point are used to derive the 2D projections of the octant vertices onto the image plane.

We only consider the 2D perspective projection computation for a B-type vertex, since the other types can be computed similarly. Let a B-type vertex be denoted by P_2 and its projected point by p_2 , and assume that it lies on the line between two A-type vertices P_1 and P_3 , whose projected points are p_1 and p_3 , respectively. Let the point at infinity along the direction from P_1 to P_3 be P_∞ , whose vanishing point is denoted by p_∞ . Fig. 2 shows the 2D projections of P_1, P_2, P_3 and P_∞ onto the image plane. The four collinear points, p_1, p_2, p_3 and p_∞ , together with their 3D corresponding vertices P_1, P_2, P_3 and P_∞ , define a common cross ratio [17], which is invariant under the perspective projection. Namely,

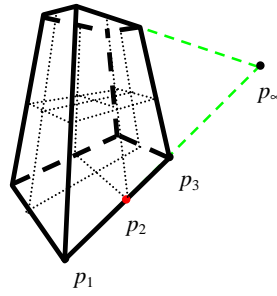


Fig. 2. The four collinear points, p_1 , p_2 , p_3 and p_∞ , together with their 3D corresponding vertices P_1 , P_2 , P_3 and P_∞ , define a common cross ratio value.

$$\overline{P_1 P_2} \cdot \overline{P_3 P_\infty} / (\overline{P_1 P_3} \cdot \overline{P_2 P_\infty}) = \overline{p_1 p_2} \cdot \overline{p_3 p_\infty} / (\overline{p_1 p_3} \cdot \overline{p_2 p_\infty}). \quad (1)$$

After some manipulation,

$$\overline{p_1 p_2} = (r \overline{p_1 p_3} \cdot \overline{p_1 p_\infty}) / (r \overline{p_1 p_3} + \overline{p_3 p_\infty}) \quad \text{with} \quad r = \overline{p_1 p_2} / \overline{p_1 p_3}. \quad (2)$$

Let parametric form of the projected line be $p_2 = p_1 + t \cdot (p_3 - p_1)$, $p_\infty = p_1 + k \cdot (p_3 - p_1)$. Then

$$k = \overline{p_1 p_\infty} / \overline{p_1 p_3} \quad \text{and} \quad t = r \cdot k / (r + k - 1). \quad (3)$$

Since the side length of the child octant equals one half of the side length of the parent octant, $r = 1/2$. It follows that $t = k/(2k - 1)$.

Thus the projected B-type vertex p_2 based on two previously computed A-type vertices is given by

$$p_2 = p_1 + k(p_3 - p_1)/(2k - 1). \quad (4)$$

2.2 Time Complexity Analysis

Before analyzing the time complexity of the computational method for the projections of octant vertices, recall that in the conventional octree construction method the projection of a 3D octant vertex onto the 2D image is done via a 3×4 projection matrix [17]. After some operations, the coordinates of the 2D projected point are

$$\begin{cases} u = \frac{h_{11}X + h_{12}Y + h_{13}Z + h_{14}}{h_{31}X + h_{32}Y + h_{33}Z + h_{34}} \\ v = \frac{h_{21}X + h_{22}Y + h_{23}Z + h_{24}}{h_{31}X + h_{32}Y + h_{33}Z + h_{34}} \end{cases} \quad (5)$$

Thus, a total of 9 multiplications, 2 divisions and 9 additions are needed in the conventional projection method to obtain one 2D projected point in the image plane based on Eq. (5). On the other hand, a total of 3 multiplications, 2 divisions and 6 additions are needed

in the new method based on the cross ratio and vanishing point. This includes 3 subtractions and 1 division to compute k by Eq. (3), 1 multiplication, 1 division and 1 subtraction to compute t by Eq. (3), and 2 multiplications and 2 additions to compute the 2D coordinates of the projected point by Eq. (4). Table 1 compares the new method with the conventional 3D projection method. Roughly speaking the new method requires less than half the computation time of the conventional method. In the experimental results, we will show that the new method does perform better than the conventional 3D-to-2D projection computation based on the projection matrix H .

Table 1. Comparison of projection computation of octant vertices between the new method and the conventional method.

Operation type	Our method		Direct 3D projection method	
	+/-	×/÷	+/-	×/÷
Operation number	6	5	9	11

3. 2D INTERSECTION TEST USING PRECOMPUTED DISTANCE MAPS OF SILHOUETTE IMAGES AND APPROXIMATION OF AN OCTANT IMAGE BY A CIRCLE

3.1 Distance Maps of Silhouette Images

After obtaining the 2D projections of the octant, which is generally a hexagon, we need to test this hexagon against the object silhouette images to determine whether it is completely inside or outside the object, or if it overlaps the object. The conventional intersection test uses the approximation of each silhouette by a set of convex sub-polygons and then examining the intersection between two convex polygons. However, rather than using the conventional method, we generate beforehand the signed distance map for each silhouette image using a chess board distance given by [18]:

$$I_0(u, v) = I(u, v) \text{ and} \\ I_k(u, v) = I_0(u, v) + \min_{\Delta(u, v; i, j)} \{I_{k-1}(i, j); ((i, j) : \Delta(u, v; i, j) \leq 1)\}, \quad (6)$$

where $\Delta(u, v; i, j)$ is the distance between (u, v) and (i, j) and $I(u, v)$ is the input image.

Here the map distance value is positive at a pixel located inside the silhouette, negative at a pixel outside the silhouette, and 0 at a pixel on the silhouette boundary. The absolute distance value at each pixel in the distance map represents the shortest distance from the pixel to the silhouette boundary.

3.2 Intersection Test using the Circular Approximation of the Octant Image

Next, we find the center of the 8 projected octant vertices and estimate the radius of the bounding circle of the projected octant image. The radius is roughly taken to be the

largest distance from the center to the eight projected octant vertices. The distance map value at the circle center c in the distance map $DistMap$ and the radius r of the bounding circle are then used to determine the intersection between the silhouettes and the projected octant image. Table 2 indicates the relationship between the projected octant image and a silhouette image. The processing must be carried out for all views. However, as is well known that if the octant image is outside any single silhouette, then the octant is definitely outside the object and classified as a white node; no view other is needed for further examination. Also, an octant is classified as a black node if it is marked black in all views, and an octant is classified as a grey node if it is grey in some view.

Table 2. Relationships between the projected octant image and a silhouette image.

$DistMap(c) \geq 0$	$r > DistMap(c)$	across the silhouette boundary (indicating a grey node)
	$r \leq DistMap(c)$	inside the silhouette (indicating a black node)
$DistMap(c) < 0$	$r > -DistMap(c)$	across the silhouette boundary (indicating a grey node)
	$r \leq -DistMap(c)$	outside the silhouette (indicating a white node)

4. NEW SUBDIVISION STRATEGY AND CONSTRUCTION QUALITY MEASURE

4.1 New Subdivision Strategy

Suppose we use N views to construct the octree for the object. Let A_i be the projection image of any grey octant A and S_i be the object silhouette for $i = 0, 1, 2, \dots, N$. Let $DistMap_i$ be the signed distance map of S_i defined previously. We find the center c_i and the radius r_i of the bounding circle of A_i , as shown in Fig. 3. Then we define the white portion (or white extent) of a grey octant to be $\{r_i - DistMap_i(c_i)\}$.

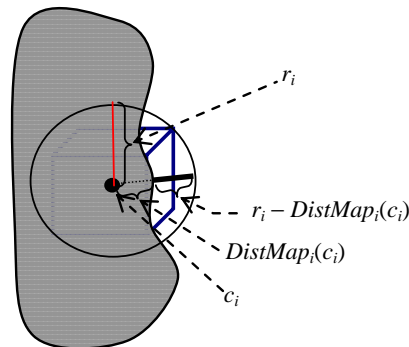


Fig. 3. Illustration of octant subdivision strategy based on the spatial relationship between the circle containing the projected octant and the object silhouette.

Now we introduce two types of grey nodes in the following:

Let “Error_bound” be a pre-specified projection error upper bound.

- (1) A grey node is said to be a “grey-grey” node if $\{r_i - DistMap_i(c_i)\} > \text{Error-bound}$ for some $i, 0 \leq i \leq N - 1$.
- (2) A grey node is defined as a “grey-black” node, if $\{r_i - DistMap_i(c_i)\} < \text{Error-bound}$ for all $i, 0 \leq i \leq N - 1$.

Based on the new types of grey nodes a new subdivision scheme for grey nodes is proposed below:

The new subdivision scheme for grey nodes:

- (1) If a node is classified as “grey-grey” it is subdivided recursively until its descendants contain no “grey-grey” nodes.
- (2) If a node is classified as “grey-black” it is retained without the need of subdivision.

4.2 Construction Quality Measure

To measure the quality of the constructed model, we define the Quality_Index which computes the exclusive OR between the binary silhouette image S_j^c of the constructed object C and the actual object silhouette S_j^o for all views ($j = 0, 1, \dots, N - 1$).

5. EXPERIMENTAL RESULTS

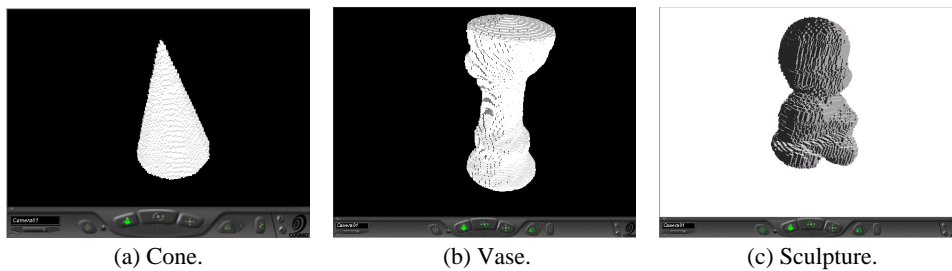
In this section, experiments are conducted to evaluate the performance of the method we proposed. Fig. 4 shows the setup of our hardware including a CCD, a turntable and a Pentium 4 3G Hz PC with 768M RAM. The camera is triggered by the PC to capture images of an object resting on the rotating turntable which is controlled by the PC. The whole reconstruction program is coded with Borland C++ Builder under the Windows environment. We tested our method on three objects with different geometric complexities: a cone, a vase and a sculpture, illustrated in Fig. 5. A number of views of each object are taken while the turntable is rotated by a fixed angle (36°) each time. Fig. 6 gives the new views rendered from the final constructed objects.



Fig. 4. Hardware setup of the system.



Fig. 5. Three objects used in the experiments.

Fig. 6. New views generated from the constructed octree models for (a), (b) and (c). The octree models are converted to VRML format and rendered using Cosmo[®] player.

We choose an octree construction method recently developed in [9] as the conventional method in order to make comparisons with our new construction method. We list the numbers of all types of octants generated at each subdivision level, together with the projection error upper bound value and the quality index of the final constructed models in Tables 3 to 6.

Table 3. Numbers of black, grey, and white nodes of the vase generated by conventional octree construction with a fixed subdivision level.

	Level = 4				Level = 5				Level = 6				Level = 7			
	XOR = 289493				XOR = 144377				XOR = 68356				XOR = 30069			
	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W
0	0	-	1	0	0	-	1	0	0	-	1	0	0	-	1	0
1	0	-	8	0	0	-	8	0	0	-	8	0	0	-	8	0
2	0	-	41	23	0	-	41	23	0	-	41	23	0	-	41	23
3	1	-	133	194	1	-	133	194	1	-	133	194	1	-	133	194
4	88	-	532	444	88	-	532	444	88	-	532	444	88	-	532	444
5	685	-	2074	1497	685	-	2074	1497	685	-	2074	1497	685	-	2074	1497
6									3811	-	7482	5299	3811	-	7482	5299
7													17526	-	21460	20870

Table 4. Numbers of black, grey-black, grey-grey, and white nodes of the vase generated by our octree construction method with a specified projection error bound.

	Protrusion = 30				Protrusion = 18				Protrusion = 10				Protrusion = 5			
	XOR = 178926				XOR = 118666				XOR = 68365				XOR = 28185			
	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
1	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0
2	0	0	41	23	0	0	41	23	0	0	41	23	0	0	41	23
3	1	17	116	194	1	8	125	194	1	1	132	194	1	0	133	194
4	12	284	188	444	40	151	365	444	81	76	455	444	88	32	500	444
5	0	316	0	1188	3	1137	283	1497	180	808	1155	1497	441	359	1703	1497
6					0	68	0	2196	1	3940	0	5299	1301	3100	3924	5299
7													496	8233	2489	20174
8													7	0	0	19905

Table 5. Node numbers of the sculpture generated by conventional octree.

	Level = 4				Level = 5				Level = 6				Level = 7			
	XOR = 366074				XOR = 186064				XOR = 89866				XOR = 41534			
	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W
0	0	-	1	0	0	-	1	0	0	-	1	0	0	-	1	0
1	0	-	8	0	0	-	8	0	0	-	8	0	0	-	8	0
2	0	-	51	13	0	-	51	13	0	-	51	13	0	-	51	13
3	7	-	232	169	7	-	232	169	7	-	232	169	7	-	232	169
4	206	-	957	693	206	-	957	693	206	-	957	693	206	-	957	693
5					1483	-	3738	2435	1483	-	3738	2435	1483	-	3738	2435
6									7307	-	13437	9160	7307	-	13437	9160
7													31273	-	40955	35265

Table 6. Node Numbers of the sculpture generated by our octree construction method.

	Protrusion = 30				Protrusion = 18				Protrusion = 10				Protrusion = 5			
	XOR = 226886				XOR = 156235				XOR = 89872				XOR = 37350			
	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W	B	GB	GG	W
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
1	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0
2	0	0	51	13	0	0	51	13	0	0	51	13	0	0	51	13
3	7	46	186	169	7	16	216	169	7	10	222	169	7	4	228	169
4	19	476	300	693	103	325	607	693	133	159	291	693	174	69	888	693
5	0	438	0	1962	6	1927	488	2435	415	1443	2035	2435	955	717	2997	2435
6					0	166	0	3738	13	7107	0	9160	2139	5531	6966	9160
7													822	14814	5743	34349
8													54	0	0	45890

In addition, the graphical display of the constructed object models and the XOR error images obtained by the conventional method and the new method are shown in Figs. 7 to 8.

From Tables 3 to 6 and Figs. 7 to 8 the two methods can be compared in terms of memory requirements, computation time, and construction quality. Since the object model is represented by leaf nodes of the constructed octree including black nodes and grey-black nodes, the amount of memory required to store the constructed object model is mainly proportional to the total number of black and grey-black nodes. On the other hand, the computer processing time required is mainly proportional to the size of the entire set of leaf and internal nodes generated. Finally, the construction quality of the result is evaluated based on the XOR projection error given by the “*Quality Index*.”

















Conventional method	Level = 4	Level = 5	Level = 6	Level = 7
Image of the constructed model				
XOR error image of the image				
New method	Protrusion = 30	Protrusion = 18	Protrusion = 10	Protrusion = 5
Image of the constructed model				
XOR error image of the image				

Fig. 7. Results for the vase obtained by the conventional method and the new method.

















Conventional method	Level = 4	Level = 5	Level = 6	Level = 7
Image of the constructed model				
XOR error image of the image				
New method	Protrusion = 30	Protrusion = 18	Protrusion = 10	Protrusion = 5
Image of the constructed model				
XOR error image of the image				

Fig. 8. Results for the sculpture obtained by the conventional method and the new method.

The following specific observations can be made from above:

- (1) Based on the quality of the constructed object models as expressed by the XOR value we can rank the construction results obtained by both the conventional and the new methods.
- (2) During the construction process the patterns of the white nodes generated by the two methods are identical. To be more specific, the white nodes at a lower subdivision level must be completely generated before any white node can be generated at the next higher level. For instance, in Table 4 the full set of 7482 white nodes must be generated at level 6 before any white node at level 7 can be generated as the error control bound P decreases from 18 to 10 to 5.
- (3) For the same quality of the construction the total number of leaf nodes of the object model generated by the conventional method is greater than that generated by the new method. For instance, in Tables 3 and 4 the total number of leaf nodes of V_{10}^N is smaller than that of V_6^C . For the same construction quality, the total number of internal and leaf nodes is less in the new method.

Figs. 9 and 10 illustrate the observations (3). In these plots a linear interpolation of the discrete set of construction results is used.

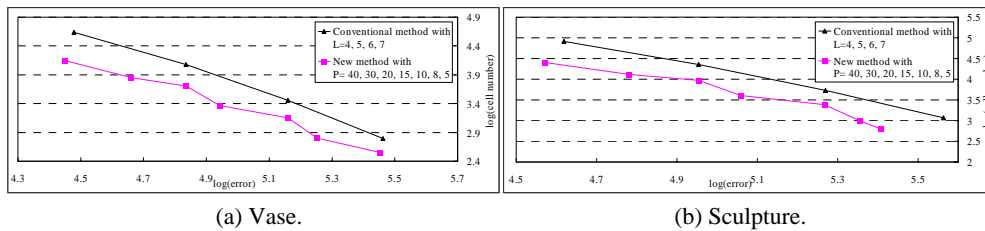


Fig. 9. The total number of leaf nodes vs. XOR error plot of the construction results obtained by the new method and the conventional method.

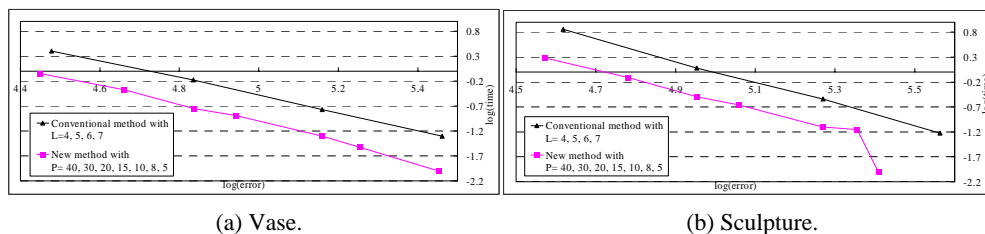


Fig. 10. Computation time vs. XOR error plot of the construction results obtained by the new method and the conventional method.

6. ANALYSIS OF THE NEW OCTREE CONSTRUCTION METHOD

The conventional method and the new method can be compared in terms of memory storage, computation time, and visual quality of the construction. First, the object model

is represented by the black nodes and grey-black nodes, so the memory space required to store the constructed object model is the total number of black and grey-black nodes. Next, the computer processing time required is mainly proportional to the entire set of nodes including the leaf nodes and internal nodes. Finally, the construction quality of the result is evaluated based on the XOR projection error given section 4.2.

Before the theoretical analysis on the performance of the new octree construction method we shall first define the notations and symbols used in the analysis:

- (1) $V_T, V_{Root}, V_L^C, V_P^N$: the volumes of the true object, the root node of the octree, the constructed object obtained by the conventional method with a maximum subdivision level L , and the constructed object obtained by the new method with a specified projection error upper bound P , respectively.
- (2) B_l^C, W_l^C, G_l^C : the sets of black, white, and grey nodes generated at level l in the conventional method.
- (3) $B_l^N, W_l^N, GB_l^N, GG_l^N$: the sets of black, white, grey-black, and grey-grey nodes generated at level l in the new method.
- (4) $V_{N_p^m}$ (e.g., $V_{B_L^C}, V_{B_P^N}$): the volume of the set of a particular type of nodes $N \in \{B, GG, GB, W\}$ generated by the one of the two methods “ m ” $\in \{C(\text{conventional method}), N(\text{new method})\}$ with a “parameter” $\in \{L(\text{level value}), P(\text{protrusion value})\}$.
- (5) I_V : the binary image obtained through the projection of a particular object or a particular node set onto the image plane in a view, where $V \in \{V_T, V_{Root}, V_L^C, V_P^N, V_{B_L^C}, V_{B_P^N}, \dots\}$.
- (6) $E_{V_L^C}, E_{V_P^N}$: the total XOR projection errors of $I_{V_L^C}$ and $I_{V_P^N}$, respectively, with the true object silhouette over all views.

Next, we begin to derive the useful lemmas for the theoretical analysis of the performance of the two construction methods. These lemmas justify the observations made in the last section.

Lemma 1 The new method always halts at a subdivision level with no remaining grey-grey node, while the conventional method generally terminates at a given maximum subdivision level with remaining grey nodes.

Proof: This is a consequence of the different subdivision strategies adopted by the two methods. There are only two cases in which the conventional method terminates without any remaining grey nodes. The first is when the octant resolution becomes so high that all the octant projections onto the image plane have a unit area. The second is when the object happens to be so neat that it can be fitted nicely by an octree with only a few resolution levels.

Lemma 2 If GG_L^N is not empty, then $W_l^N = W_l^C, l = 1, 2, \dots, L - 1$.

Proof: We prove the lemma by contradiction. Assume $W_l^N \neq W_l^C, l = 1, 2, \dots, L - 1$.

Let $w_{k-1}, 1 \leq k \leq L$, be a node in the set W_{k-1}^C , but not in the set W_{k-1}^N . Since w_{k-1} is not in the set W_{k-1}^N , its parent node gb_{k-2} should be in the set GB_{k-2}^N . Thus, we have $r_{k-2} -$

$|d_{k-2}| \leq P$, where r_{k-2} and d_{k-2} are the radius and the distance map value related to the bounding circle of the projected image of gb_{k-2} .

Since w_{k-1} is within gb_{k-2} , its projection should not exceed the white portion of the projection of its parent node. So $r_{k-1} + |d_{k-1}| \leq r_k - |d_{k-2}| \leq P$.

Also, since w_{k-1} is a white node, its entire projection should be outside the silhouette, so $|d_{k-1}| > r_{k-1}$. Thus,

$$P \geq r_{k-1} + |d_{k-1}| > r_{k-1} + r_{k-1} > 2r_{k-1} \quad (7)$$

given that GG_L^N is not an empty set, there exists at least one grey-grey node gg_L belonging to GG_L^N . There are two possible cases for the location of this gg_L .

Case 1. The centroid of the projection of gg_L is outside the silhouette image.

Then $r_L + |d_L| > P$, where r_L is the radius of the bounding circle of the projection of gg_L , $|d_L|$ is the shortest distance between the centroid of the projection of gg_L and the silhouette, and P is the pre-specified projection error upper bound used to control the node subdivision.

Since gg_L partially overlaps the silhouette, we have $|d_L| \leq r_L$. Thus, $r_L + r_L > r_L + |d_L| > P$. That is, $2r_L > P$.

Case 2. The centroid of the projection of gg_L is inside the silhouette image.

Then $r_L - |d_L| > P$. Thus, $2r_L > r_L + r_L - |d_L| > P$.

We conclude in both cases that $2r_L > P$.

From above equation and Eq. (8), we have $r_L > r_{k-1}$ for $1 \leq k \leq L$.

This contradicts the fact that $r_L < r_{k-1}$ was assumed, because radius r_L of a projected node at level L should be smaller than the radius r_{k-1} of the bounding circle of its ancestor at level $k - 1$. (Here the distances between the object and the camera in all the views are nearly constant.) Thus, $W_l^N = W_l^C$, $l = 1, 2, \dots, L - 1$.

Lemma 3

- (1) If $|GG_L^N| = 0$ and $W_L^N = W_L^C$, then $E_{V_P^N} = E_{V_L^C}$ and the total number of leaf nodes generated by the conventional method is greater than that of the new method.
- (2) The total number of leaf and internal nodes in the conventional method is greater.

Proof:

- (1) Since $W_L^N = W_L^C$ and GG_{L-1}^N is not empty, Lemma 2 implies that $W_l^N = W_l^C$, $l = 1, 2, \dots, L - 2$. We can then show that $W_{L-1}^N = W_{L-1}^C$ using the same technique used to prove Lemma 2.

Next, $V_L^C = V_{Root} - \sum_{l=1}^L V_{W_l^C} = V_{Root} - \sum_{l=1}^L V_{W_l^N} = V_P^N$, which implies that their projected images of the two object models constructed are equal, i.e., $I_{V_P^N} = I_{V_L^C}$. Also, their XOR images are equal, i.e., $I_{V_L^C} \otimes I_{V_T} = I_{V_P^N} \otimes I_{V_T}$, and so, $E_{V_P^N} = E_{V_L^C}$.

- (2) From the above $W_l^N = W_l^C$, $l = 1, 2, \dots, L$. On the other hand, $B_l^N \subseteq B_l^C$, $GB_l^N \subseteq G_l^C$ for $l = 1, 2, \dots, L$. To emulate the subdivision result of the conventional method, imagine that the grey-black nodes generated at any level lower than L are decom-

posed. Then the leaf nodes generated by the two methods satisfy the following inequality:

$$\sum_1^L |B_l^N| + \sum_1^{L-1} 8 \cdot |GB_l^N| + |GG_L^N| + |GB_L^N| + \sum_1^L |W_l^N| < \sum_1^L |B_l^C| + |G_L^C| + \sum_1^L |W_l^C|.$$

Implying

$$\begin{aligned} & \sum_1^L |B_l^N| + \sum_1^{L-1} |GB_l^N| + |GB_L^N| + \sum_1^L |W_l^N| \\ & < \sum_1^L |B_l^N| + \sum_1^{L-1} 8 \cdot |GB_l^N| + |GG_L^N| + |GB_L^N| + \sum_1^L |W_l^N| \quad (\text{Note that } |GG_L^N| = 0) \\ & < \sum_1^L |B_l^C| + |G_L^C| + \sum_1^L |W_l^C|. \end{aligned}$$

Therefore, the total numbers of leaf nodes generated by the two methods are not equal. In addition, because the grey-black nodes are not decomposed in the new method, the total number of leaf and internal nodes generated by the conventional method is greater than that generated by the new method.

Lemma 4

- (1) If $|GG_{L+1}^N| = 0$ and $|W_L^N| < |W_L^C| < [(|W_{L+1}^N|/8) + |W_L^N|]$, then $0 < E_{V_L^C} - E_{V_P^N}$.
- (2) Additionally if $\sum_{l=1}^{L-1} |GB_l^N| > |GG_L^N|$ holds, then the total number of leaf nodes generated by the conventional method is greater than that generated by the new method.
- (3) The total number of leaf and internal nodes generated by the conventional method is greater.

Proof:

- (1) Since $|GG_L^N|$ is not zero, $W_l^N = W_l^C$, $l = 1, 2, \dots, L-1$ due to Lemma 2. Since the volume of each node in W_{L+1}^N is one eighth of that of W_L^N , the condition $[(|W_{L+1}^N|/8) + |W_L^N|] > |W_L^C|$ indicates that the total volume of W_{L+1}^N and W_L^N is greater than the volume of W_l^C . Therefore, the object volume obtained by the new method is smaller than that obtained by the conventional method. Namely,

$$V_P^N = V_{Root} - \sum_{l=1}^{L+1} V_{W_l^N} < V_{Root} - \sum_{l=1}^L V_{W_l^C} = V_L^C.$$

Consequently, the area of $I_{V_P^N}$ is smaller than that of $I_{V_L^C}$. Also, the area of $I_{V_P^N} - I_{V_T}$ is smaller than that of $I_{V_L^C} - I_{V_T}$. It can be readily shown that the XOR projection errors of the two methods satisfy the inequality $I_{V_P^N} \otimes I_{V_T} < I_{V_L^C} \otimes I_{V_T}$. That is, $E_{V_P^N} < E_{V_L^C}$.

(2) As before, comparing the subdivision processes of the two methods reveals

$$\begin{aligned}
 & \sum_{l=1}^L |B_l^C| + |G_L^C| + \sum_{l=1}^L |W_l^C| > \sum_{l=1}^L |W_l^C| + 8 \sum_{l=1}^{L-1} |GB_l^N| + |GB_L^N| + \sum_{l=1}^L |W_l^N| + |GG_L^C| \\
 & = \sum_{l=1}^L |B_l^N| + \sum_{l=1}^{L-1} |GB_{l+1}^N| + 7 \sum_{l=1}^{L-1} |GB_{l+1}^N| + |GB_L^N| + \sum_{l=1}^L |W_l^N| \\
 & \quad + |B_{L+1}^N| + |GB_{L+1}^N| + |W_{L+1}^N| - 7 |GG_L^N| \\
 & > \sum_{l=1}^L |B_l^N| + \sum_{l=1}^{L-1} |GB_{l+1}^N| + |GB_L^N| + \sum_{l=1}^L |W_l^N| + |B_{L+1}^N| + |GB_{L+1}^N| + |W_{L+1}^N|, \\
 & \text{if } \sum_{l=1}^{L-1} |GB_{l+1}^N| > |GG_L^N|.
 \end{aligned}$$

Therefore, the total number of leaf nodes generated by the conventional method is greater than that generated by the new method.

A remark is in order here. When the subdivision level L is large enough, most nodes in the set of grey-grey nodes GG_L^N are occupied mainly by the white space due to the fact that the sets of black and grey-black nodes, B_L^N and GB_L^N have been already extracted from GG_{L-1}^N . Thus, the nodes in GG_L^N will likely be decomposed into white nodes belonging to W_{L+1}^N with $|W_{L+1}^N|/8 \approx |GG_L^N|$. Under this circumstance when the subdivision level is high enough, then the size of GG_L^N generally becomes small and the condition $\sum_{l=1}^{L-1} |GB_{l+1}^N| > |GG_L^N|$ is likely to hold.

(3) Next, because the grey-black nodes are not decomposed in the new method, the total number of internal grey nodes generated by the conventional method is greater than the number of the internal grey-grey nodes generated by the new method. Furthermore, the total number of leaf nodes generated by the conventional method is already shown to be greater than that generated by the new method. Consequently, the total number of leaf and internal grey nodes generated by the conventional method is greater than that of the new method.

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a fast and efficient method to construct an octree representation of an object from the multiple silhouettes. The computation of the projection of 3D octants onto the 2D image planes is reduced by using the invariant property of the cross ratio. A maximum projection error is specified to decide whether an octant needs subdivision. Experiments were conducted on three real objects to demonstrate the performance of the new method. The results demonstrate the improvement of the new method over the conventional method in terms of computer memory, computation time, and quality of the constructed result. Theoretical analysis of the new method was presented.

In the future, we shall convert the constructed object model into a polygonal representation and extract textures from the real object images so that we can map textures onto the object surface to obtain a photo-realistic effect. And, further investigations of other effective octant subdivision strategies are underway. Preliminary results show that using both grey-black and grey-white nodes yields additional reduction in the required memory as well as the computation time. Also, the replacement of the recursive construction process by a progressive octree construction process with a visual quality upgrading is also under study.

REFERENCES

1. R. Vaillant and O. D. Faugeras, "Using extremal boundaries for 3-D object modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, 1992, pp. 157-173.
2. W. B. Seales and O. D. Faugeras, "Building three-dimensional object models from image sequences," *Computer Vision and Image Understanding*, Vol. 61, 1995, pp. 308-324.
3. C. H. Chien and J. K. Aggarwal, "Volume/surface octrees for the representation of three-dimensional objects," *Computer Vision, Graphics and Image Processing*, Vol. 36, 1986, pp. 100-113.
4. H. Noborio, S. Fukuda, and S. Arimoto, "Construction of the octree approximation three-dimensional objects by using multiple views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, 1988, pp. 769-782.
5. Narendra Ahuja and Jack Veenstra, "Generating octrees from object silhouettes in orthographics Views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, 1989, pp. 137-149.
6. H. H. Chen and T. S. Huang, "A survey of construction and manipulation of octrees," *Computer Vision, Graphics and Image Processing*, Vol. 43, 1988, pp. 409-431.
7. M. Potmesil, "Generating octrees models of 3D objects from their silhouettes in a sequence of images," *Computer Vision, Graphics and Image Processing*, Vol. 40, 1987, pp. 1-29.
8. S. K. Srivastava and N. Ahuja, "Octree generation from object silhouettes in perspective views," *Computer Vision, Graphics and Image Processing*, Vol. 49, 1990, pp. 68-84.
9. R. Szeliski, "Rapid octree construction from image sequences," *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 58, 1993, pp. 23-32.
10. W. Niem, "Automatic reconstruction of 3D objects using a mobile camera," *Image and Vision Computing*, Vol. 17, 1999, pp. 125-134.
11. S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by octant coloring," *International Journal of Computer Vision*, Vol. 35, 1999, pp. 151-173.
12. A. Laurentini, "How many 2D silhouettes does it take to reconstruct a 3D object?" *Computer Vision and Image Understanding*, Vol. 67, 1997, pp. 81-87.
13. G. K. M. Cheung, T. Kanade, J. Y. Bouguet, and M. Holler, "A real time system for robust 3D octant reconstruction of human motions," in *Proceedings of the IEEE*

- Conference on Computer Vision and Pattern Recognition*, 2000, pp.714-720.
14. L. Bottino and A. Laurentini, "Interactive reconstruction of 3D objects from silhouette," in *Proceedings of 9th International Conference on Computer Graphics, Visualization and Computer Vision*, 2001, pp. 5-9.
 15. W. Matusik, C. Buehler, and L. McMillan, "Polyhedral visual hulls for real-time rendering," in *Proceedings of 12th Eurographics Workshop on Rendering*, 2001, pp. 115-125.
 16. G. Cheung, S. Baker, and T. Kanade, "Visual hull alignment and refinement across time: a 3D reconstruction algorithm combining shape-frame-silhouette with stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 375-382.
 17. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, New York, 1993.
 18. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, New York, 1988.



Hong-Long Chou (周宏隆) received his Ph.D. and M.S. degrees in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 1996, respectively, and his B.S. degree in Computer Science and Engineering from Yuan Ze University, Taoyuan, Taiwan, in 1994. He has been a supervisor at Altek Corporation since 2005. His research interests include computer vision, image processing, and virtual reality.



Zen Chen (陳稔) received a B.S. degree from National Taiwan University (1967), an M.S. degree from Duke University (1970), and a Ph.D. degree from Purdue University (1973), all in Electrical Engineering. He worked as a senior design engineer for Burroughs Corporation, Detroit, Michigan after his graduation from Purdue. In 1974 he joined National Chiao Tung University, Taiwan, where he is now a Professor of Computer Science. He spent the sabbatical leave at the Lawrence Berkeley Laboratory of University of California at Berkeley (1981-1982) and at the Center for Automation Research of University of Maryland at College Park, Maryland (1989-1990), respectively. He received an Elite Award of the Information Month in Taiwan (1985), the Outstanding Research Awards from the National Science Council of the Republic of China (1994-1995, 1995-1997), and an Outstanding Engineering Professor Award from the Chinese Institute of Engineers, Taiwan (1998). He was the founding president of the Chinese Society of Image Processing and Pattern Recognition, Taiwan (1990-1992). His current research interests include computer vision, pattern recognition, virtual reality, and parallel algorithms and architectures.