Discrete Shading Algorithms for 3D objects from Medial Sphere Representation

Jayanta Mukherjee
P.P.Das
Department of Computer Science & Engg.
Indian Institute of Technology
Kharagpur, INDIA 721302
jay,ppdas@cse.iitkgp.ernet.in

Aswatha Kumar M.
B.N.Chatterji
Department of E & ECE
Indian Institute of Technology
Kharagpur, INDIA 721302
bnc@ece.iitkgp.ernet.in

Abstract

In this paper we present a discrete shading technique using Medial Sphere Representation (MSR)[1] of 3D binary image data based on digital generalized octagonal distances. Our method is computationally attractive as it does not require the explicit computation of surface normals. We have compared our results with images rendered from voxel and octree reprentations while using analytical surface rendered images as bench marks. The quality of rendering by our method is certainly superior to those of obtained from voxel and octree representation.

1 Introduction

In various applications of medical imaging, image processing, solid modeling, 3D simulation and scientific visualization, three dimensional objects are represented in a 3D binary matrix, where occupancy of each unit object volume cell (or voxel) is described. The shape of the voxels is taken to be a cube and thus this model is popularly known as cuberille model[2]. Visualization of 3D objects represented in this form is one of the major challenges today. For achieving this, it is not only sufficient to project the object surface points as visible from the viewing direction on to a 2D screen, but shading is also required to provide the critical depth cues. In general, shading involves computation of the intensity (brightness) value that reaches the viewer's eye from each visible surface-point. The intensity value may be computed by different shading models taking into account the surface normal, the characteristics of the objects surfaces, distance from the viewer's position and the light sources[3].

In this paper, we discuss shading of 3D voxel objects which is referred to as discrete shading[4]. In conventional shading techniques, object surfaces are composed of polygonal meshes or mathematical surfaces which describe surface normals with high degree of accuracy at each surface point. In contrast, in the discrete shading technique, the object's actual surfaces and their normals are unknown and must be recovered from the voxel information. In the cuberille model, a voxel has six faces with normals in the directions of the primary axes and atmost three faces are visible from a viewing direction. Based on these facts, various discrete shading techniques have been developed. The major contentions of these work are to keep the computation time for recording surface normals as less as possible and to increase the accuracy of the computation of surface normals. There lies a trade-of between these two objectives as the increase in the degree of accuracy of extraction requires more computation. However the techniques are finally judged by the quality of produced images of known 3D geometric objects (voxel form) compared (visually) with the benchmark images (produced by conventional rendering techniques such as Gouraud's shading[5], Phong's shading [6] from the mathematical description of these surfaces). Some of the reported techniques of discrete shading in cuberille environment are: constant shading[3], distance-only(depth-only) shading[7][8], Image-based contextual shading[9], Normal-based contextual shading[2], congradient shading[10] etc. It is easy to observe that cuberille environment suits octree encoding of voxel data as the octree technique represents the object volumes by a set of disjoint cubes of different sizes. Hence similar methodologies have been adopted for discrete shading of octree encoded data[11].

The advantage of using octree encoded data is the reduction in the volume of the data to be processed (less number of cubes to be rendered). But the disadvantage lies in the extra effort required for computing the neighboring volume elements of a voxel that is required for more accurate computation of surface normals.

In this paper, we focuss our attention to discrete shading on a non-cuberille environment, called the medial sphere representation(MSR)[12][13] of a 3D object. The advantage of MSR not only lies in data reduction but also in representing surface normals in more number of directions exploiting the geometry of medial spheres.

2 Medial Sphere Representation using Octagonal Distance

In 2D, binary images could be represented by Medial Axis Transform (MAT), which is also called as Medial Circle Representation (MCR). MAT consists of a set of center and radii of the maximal block of the image Σ. Here every center is located half-way between the boundary on either side. Hence the collection of all these centers forms a sort of spine for the object and is called the medial axis. The computation of medial axis requires the distance transformation. Distance transform converts the binary image Σ consisting of object pixels(1) and background pixels(0), into an image where all pixels have a value corresponding to the distance of the nearest background pixel. For defining various digital distances in 2-D, different neighborhood sequences for path definitions have been used[14] [15]. The commonly used neighborhood sequences are 4-neighborhood (city block) and 8-neighborhood(chessboard). It is observed that neither cityblock nor chessboard distance is close to the Euclidean distance. 3D extension of MCR is known as Medial Sphere Representation(MSR), which describes a 3D object by the set of centers and radii of medial spheres. Similar to the 2D images, the distance maps are also used in 3D images. Commonly known distance functions in 3D use 6-neighbors(face neighbor), 18-neighbors (face and edge neighbor) and 26-neighbors(face, edge and vertex neighbor). It is interesting to note that the shape of the medial spheres varies if we choose different neighborhood sequences[13] for forming the distance map. In our shading algorithm we shall render each medial sphere individually and the combined effect of all the rendered spheres produces the desired image of the 3D object considering the fact that the graphics environment provides z-buffering hardware feature. The computation gets reduced as the shading of a sphere does not require the computation of normals by scanning the neighborhoods of any voxel. Rather, for any point, the vector from the center of the sphere to that point, gives the desired normal direction. Hence it is important to know the shape of the digital medial spheres for different octagonal distances in 3D, which could be shown as convex polytopes. These have been discussed elsewhere[12]. However a brief introduction of 3D octagonal distances are given below.

2.1 Octagonal Distances in 3D

In 3D digital space Z^3, three types of neighborhoods have been identified. They are face neighborhood (type 1), edge neighborhood (type 2) and vertex neighborhood (type 3). The neighborhood sequences of length upto 3 are {1}, {1,1,2}, {1,1,3}, {1,2}, {1,2,2}, {1,2,3}, {1,3}, {1,3,3}, {2}, {2,2,3}, {2,3}, {2,3,3}, {3}. It is found [16] that the distance function defined by neighborhood sequence B is a metric if B is sorted. Hence all the neighborhood sequences mentioned above are metric. The computations of the vertices of these digital spheres, which are convex polytopes in shape, are discussed in [12].

3 Shading of 3D Objects

For the purposes of comparative study on shading, we have considered three different representations, MSR, octree and voxel data for rendering.

<u>Using MSR:</u> In this technique, we have considered the input as an MSR of a 3D object. Each medial sphere is rendered independently aided by the z-buffering features of the graphics hardware environment. Hence hidden surface elimination is automatically performed by the graphics workstation. To render a medial sphere the vertex of the polytopes are computed and the normal at a vertex has been taken as the direction of the vector drawn from the center of the sphere to that vertex. This normal has been used to compute the intensity value at that point by using simple shading model [3]. The faces of the polytope are then shaded by Gouraud's interpolation technique [5] using the intensity values computed at the vertices forming a face.

<u>Using voxel and Octree representations</u> :
A voxel or a leaf node of octree is of cubic shape. Hence each face of this cube is rendered and the combined effect produces shaded image of the 3D object. To render a face of the cube, the intensity at each vertex of the cube is computed by applying the same shading model followed by Gouraud's interpolation technique. The normals at the vertices are drawn from the center of the cube.

<u>Surface Rendering:</u>
In this paper, we have considered only known 3D geometric objects such as sphere, cylinder and cone. We have applied the same shading model to render the boundary surfaces (from analytical model) of these

objects. These images are taken as benchmark images and are compared with the discretely shaded images both visually as well as quantitatively by computing a correlation measure between these images.

It is interesting to note here that MSR description of a 3D object may consist of several medial spheres of small radii (≤ 3) lying near the boundary of the object. Even if we ignore the shading of these spheres, the appearance of the shaded image does not degrade and in fact with our quantitative measure (described in the next section), the quality has been shown to improve in most of the cases. Hence in our study we have also considered shading of Truncated MSR (TMSR) of a 3D object. In TMSR spheres of smaller sizes are ignored.

3.1 Measure on Relative performance of discrete shading over different representation schemes

We have carried out experiments to study the relative performance of discrete shading techniques. For this, same shading model has been used for producing the images of the 3D objects from different representation on the same position of projection screen using identical coordinate convention. We have used different representation schemes such as voxel, octree and MSR. For MSR, we have considered different digital octagonal distances. The benchmark image is provided by the surface rendered image.

Correlation Measure:
Let the benchmark image be denoted as $I_b(i,j), i = 1,2,...,M$, and $j = 1,2,...,M$. Let the discretely rendered image be denoted as $I_d(i,j), i = 1,2,...,M$, and $j = 1,2,...,M$. Then we have used simple correlation measures as follows:

$$r = \frac{(\text{Covariance of } I_b \text{ and } I_d)}{(\text{s.d of } I_b * \text{ s.d of } I_d)} \quad (1)$$

We have considered background pixel as dark(0) and in the computation of r, if we get $I_b(i,j) = I_d(i,j) = 0$ at any point (i,j), we have ignored them, (i.e., for each occurance of such case, the value of M^2 is decreased by 1 and the total number of object pixels (N) also adjusted accordingly. This removes the biasing in the correlation measure to its higher value as both the images will have a large number of common background pixel. It may be noted that the value of r will lies between -1 and +1, with 1 indicating strong correlation. Naturally the value nearing zero implies poor correlation.

Another measure used to find the relation between the benchmark image and the shaded image from MSR is the Signal to Noise Ration (SNR). SNR_{rms} is defined as

$$SNR_{rms} = \frac{\sqrt{\sum_i \sum_j I_b(i,j)^2}}{\sqrt{\sum_i \sum j(I_b(i,j) - I_d(i,j))^2}} \quad (2)$$

4 Experimental Results

A set of shaded images obtained by rendering the sphere from its MSR of different reprsentative distances has been illustrated in Figure 1. In this case, the light direction is taken as (-1, -1, 0) (considering the vector from (1,1,0) to (0,0,0) in 3D coordinate space). In Figure 2 shaded cylinders using different representation schemes have been shown. The quality of shaded images using MSR is certainly better than those of voxel and octree reprsented ones.

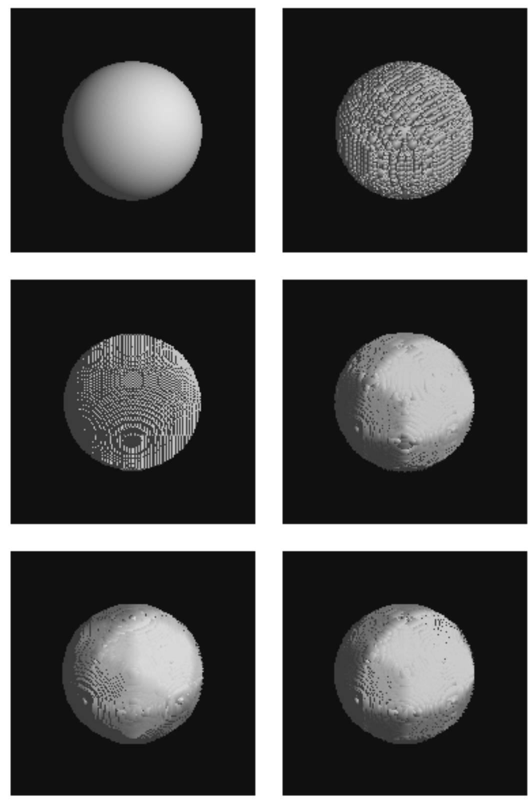

Figure 1: Shaded images of the sphere from MSR of different digital distance functions for the light direction (-1,-1,0)

Table 1 describes the correlation and SNR for different representations of different objects for light direction (-1,-1,0). To reduce the effect of light direction

Table 1: Correlation(r) and SNR between the benchmark images and the rendered images obtained from MSR,octree and voxel data for light direction (-1,-1,0).

B	Sphere		Cylinder		Cone	
	r	SNR	r	SNR	r	SNR
1	0.606	2.395	0.297	1.871	0.827	3.125
112	0.801	3.554	**0.595**	2.699	0.805	3.207
113	**0.877**	**5.206**	**0.655**	**3.424**	**0.868**	**5.04**
12	0.822	3.704	**0.596**	**2.731**	0.795	3.157
122	0.795	3.445	0.561	2.565	0.772	3.064
123	**0.866**	**4.533**	0.475	2.616	**0.827**	4.178
13	**0.861**	**4.695**	0.512	**2.785**	**0.856**	**5.173**
133	0.839	4.198	0.471	2.601	0.800	**4.414**
2	0.696	2.469	0.505	2.197	0.654	2.336
223	0.759	2.846	0.419	2.121	0.655	2.479
23	0.747	2.793	0.418	2.133	0.708	2.848
233	0.740	2.722	0.394	2.067	0.673	2.665
3	0.705	2.528	0.347	1.997	0.690	2.841
Octree	0.405	2.504	0.133	2.182	0.371	2.34
Voxel	0.313	2.165	0.108	1.572	0.361	1.718

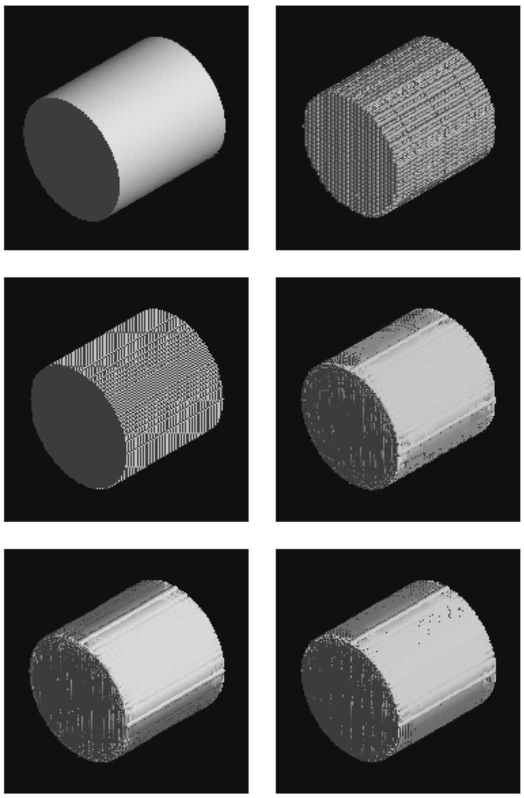

Figure 2: Comparison of shaded images for Cylinder computed from various representations and MSRs of different octagonal distances.

in the correlation measure, we have obtained correlations for different light directions (such as (-1,0,0),(0,-1,0),(0,0,-1),(0,-1,-1),(-1,0,-1),(-1,-1,0) and (-1,-1,-1)) and the averages of them are shown in Table 2. From Tables 1 and 2 we observe that quality of rendered images improves if we use MSR rather than voxel or octree representation of 3D objects. Even in MSR some octagonal distances such as {113},{123},{13} give better results compared to the rest(see also Figure 1).

Besides quality, another advantage of this scheme is that it does not require explicit normal computation. Even it does not require extra storage for coding the normals at the boundary points. The size of the data is also less. The major bottleneck of the system is the use of z-buffering which slows the speed. MSR is not a presorted data structure. In voxel representation and octree representation many rendering techniques have effectively used these features for speeding up the computation and avoiding z-buffering computation. But one can prepartition medial spheres into several groups so that parallel rendering tasks could be employed independently on those groups which will speed up the operation.

It is observed from the set of medial spheres that some of the spheres have small radii values. These small spheres are situated at the boundary points. Elimination of such small spheres (the representation is called as truncated MSR or TMSR) results in reduction in storage requirement as well as in smoothing the object. Due to the smoothing, the correlation is improved and is shown in Table 3. It could be noted

Table 2: Average Correlation(r) and SNR between the benchmark images and the rendered images obtained from MSR, octree and voxel data averaged for different light directions
((-1,0,0), (0,-1,0), (0,0,-1), (0,-1,-1), (-1,0,-1), (-1,-1,0) and (-1,-1,-1))

	Sphere		Cylinder		Cone	
B	r	SNR	r	SNR	r	SNR
1	0.635	2.800	0.490	2.643	0.611	3.487
112	0.844	4.485	0.720	**3.646**	0.689	4.415
113	**0.889**	**5.565**	**0.771**	**3.871**	**0.724**	4.983
12	0.865	4.895	**0.744**	**3.764**	0.713	5.017
122	0.858	4.771	**0.745**	3.513	0.716	**5.085**
123	**0.897**	**5.758**	0.728	3.413	**0.747**	**5.932**
13	**0.878**	**5.120**	0.728	3.385	**0.737**	**5.386**
133	0.852	4.469	0.710	3.175	0.713	4.725
2	0.765	3.550	0.705	2.898	0.635	3.962
223	0.761	3.663	0.663	2.696	0.663	4.425
23	0.747	3.467	0.662	2.662	0.657	4.167
233	0.730	3.252	0.639	2.557	0.637	3.752
3	0.632	2.847	0.601	2.479	0.575	3.338
Octree	0.235	2.417	0.325	2.207	0.202	2.662
Voxel	0.289	1.944	0.238	1.819	0.249	2.172

Table 3: Correlation(r) and SNR between the benchmark images and the rendered images obtained from truncated MSR for different light directions and compared with ordinary MSR

	Cylinder				Cone			
	MSR		TMSR		MSR		TMSR	
B	r	SNR	r	SNR	r	SNR	r	SNR
1	0.49	2.64	0.49	2.55	0.61	3.49	0.63	3.49
112	0.72	3.65	0.72	3.61	0.69	4.41	0.69	4.30
113	0.77	3.87	0.77	3.88	0.72	4.98	0.73	4.82
12	0.74	3.76	0.75	3.71	0.71	5.02	0.72	4.80
122	0.74	3.51	0.74	3.46	0.72	5.08	0.71	4.75
123	0.73	3.41	0.73	3.38	0.75	5.93	0.75	5.48
13	0.73	3.38	0.74	3.42	0.74	5.39	0.74	5.13
133	0.71	3.17	0.71	3.18	0.71	4.72	0.71	4.46
2	0.70	2.90	0.72	2.91	0.63	3.96	0.64	3.57
223	0.66	2.70	0.68	2.69	0.66	4.42	0.67	3.95
23	0.66	2.66	0.68	2.66	0.66	4.17	0.65	3.70
233	0.64	2.56	0.66	2.57	0.64	3.75	0.63	3.39
3	0.60	2.48	0.63	2.65	0.57	3.34	0.58	3.00

that with TMSR for sphere, the correlation is almost similar to that of MSR case (for many of the distance functions), since all the medial spheres in MSR are of radii greater than the threshold value. Hence there is no improvement in the correlation measure in such cases. However for the cylinder, the correlation is improved from 0.70 to 0.72 by using TMSR for distance function $\{2\}$ and from 0.60 to 0.63 for the distance function $\{3\}$. Similarly for cone, the correlation is improved from 0.61 to 0.63 for the distance function $\{1\}$ (Please refer Table 2 and 3). It may however, be noted that truncation has hardly any effect for distances that produce good shading effect.

5 Conclusion

In this paper we have presented a shading technique using MSR, which does not require explicit computation of surface normals. We have also studied on the suitability of a set of generalized octagonal distance for this purpose and found that $\{113\}$,$\{13\}$,$\{123\}$ distances perform well in this regard. On comparison with the shaded images obtained by octree and voxel representation under similar conditions it has been observed that MSR performs better in most of the cases (irrespective of the use of any 3D octagonal distance). Another interesting observation is that use of Truncated Medial Sphere Representation (TMSR) improves the quality of shading in some cases as it smoothens the appearance of the shaded surface. It is interesting to note also that TMSR occupies less storage space as it prunes many spheres of smaller radii lying on the boundary.

Acknowledgements

The support of Indian National Science Academy (INSA) is gratefully acknowledged.

References

[1] M. Aswatha Kumar, Jayanta Mukherjee, B. N. Chatterji, and P. P. Das. Medial circle and sphere representation using octagonal distances. In *Vision Geometry III, Proc. SPIE*, volume 2356, pages 284–294, 2 - 3 November 1994. (Robert A Melter and Angela Y Wu):editors.

[2] L. S. Chen, G. T. Herman, A. Reynolds, and J. K. Udupa. Surface shading in the cuberille environment. *IEEE Computer Graphics and Applications*, 5(12):33–43, December 1985.

[3] J. D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Pub. Co, 1984.

[4] D. Cohen, A. Kaufman, R. Bakalash, and S. Bergman. Real time discrete shading. *Visual Computer*, 6:16–27, 1990.

[5] H. Gouraud. Continuous shading of curved surfaces. *IEEE Trans. Computers*, C-20(6):623–629, June 1971.

[6] B. T. Phong. Illumination for computer generated pictures. *Communications of ACM*, 18(6):311–317, 1975.

[7] H. K. Tuy and L. T. Tuy. Direct 2-d display of 3-d objects. *IEEE Computer Graphics and Applications*, 4(10):29–33., Oct.1984.

[8] G. Frieder, D. Gordon, and A. Reynolds. Back-to-front display of voxel-based objects. *IEEE Computer Graphics and Applications*, 5(1):52–56, January 1985.

[9] G. T. Herman and J. K. Udupa. Display of three dimensional discrete surfaces. In *Proc. SPIE*, volume 283, pages 90–97, 1981.

[10] D. Gordon and R. A. Reynolds. Image space shading of 3-dimensional objects. *Computer Vision Graphics and Image Processing*, 29:361–376, 1985.

[11] L. J. Doctor and J. G. Torborg. Display techniques for octree,encoded objects. *IEEE Computer Graphics and Applications*, 1(7):29–38, July 1981.

[12] M. Aswatha Kumar, Jayanta Mukherjee, B. N. Chatterji, and P. P. Das. Representation of 2D and 3D binary images using medial circles and spheres. *Intl. Jrnl. Pattern Recognition and Artificial Intelligence*, pages 365–387, October 1996.

[13] M. Aswatha Kumar, Jayanta Mukherjee, and B. N. Chatterji. Medial sphere representation for 3D rendering. In *Intl. Conf. CAD/CAM,Robotics and Autonomous Factories*, volume II, pages 25–33, December 1993.

[14] A. Rosenfeld and J. L. Pfaltz. Distance functions in digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[15] A. Rosenfeld and A. V. Kak. *Digital Picture Processing, Vol.I.* Academic Press, 1982.

[16] P. P. Das and B. N. Chatterji. Octagonal distances for digital pictures. *Information Sciences*, 50:123–150, 1990.