 ISBN：978-957-21-8031-0

Adaptable Voxelization

Hung-Kuang Chen
Electronic Engineering Department, National Chin-Yi University of Technology
hankchentw@gmail.com

Chih-Yuan Tu
Electronic Engineering Department, National Chin-Yi University of Technology
meditation0624@yahoo.com

Abstract

In this paper we proposed a novel approach to hardware accelerated voxelization. The new approach begins with subdividing the input 3D model into different groups by separating disjointed parts according to its topology. Then, for each group, solid voxelization is applied along three principle axis . The final result, or the complete voxel set, is derived by merging the voxels from all the groups. Unlike previous methods, our method is capable of processing both manifold and non-manifold meshes and those with intersecting parts. Moreover, the derived voxelization preserves model continuity and completeness.

1. Introduction

Unlike surface representations of the 3D models such as bi-cubic patch and polygonal mesh, volumetric representation, specifically, the voxel representation, contains more information than the surfaces by capturing subtle variations inside the model and is commonly used in a variety of fields such as medical imaging and scientific visualizations.

Owing to this reason, a vast amount of studies have been done on these fields. Among these efforts, the methods for reconstructing and visualizing volumetric representation of 3D objects by gathering images retrieved from CT and MRI have been extensively studied[14].

According to the derived results, methods of voxelization can be classified into two types, namely, the surface voxelization and the solid voxelization. So far, various types of voxelization algorithms have been proposed. Ealier methods mostly applies ray-casting[4]. Lately, efficient methods based on rendering sliced frame have been proposed[12]. With the help of GPU[5][6], the computational efficiency can be further improved and real-time voxelization is made possible.

Concurrent frame-based techniques accept only manifold meshes. For non-manifold meshes or those consisting of intersecting parts as shown in Figure 1, the computed results may contain incorrect voxels or disjoint voxel sets.

Figure 1: The Al Carpon mesh- consisting of twenty one intersecting subparts.

According to our study, there is no feasible solution to effectively deal with such problems so far. To address these issues, we propose a novel adaptive approach to surface mesh voxelization. The new

 ISBN：978-957-21-8031-0

method is able to voxelize both manifold and non-manifold meshes as well as those consisted of intersecting parts.

Moreover, with our new method, there is no need to repair cracked meshes in advance to ensure that the computed results contains no excess incorrect voxel outside the model range.

2. Related Works

Volume graphics has been extensively used and studied for over three decades. A vast amount of methods have been proposed. A overview of these works and a summary of related terms has been done by Arie Kaufman in [1].

According to his work, the input volume data for common volume rendering approaches may be either sampled or computed, where the volume data is collected by sampling from real world objects by means of technologies such as CT, MRI, remote sensing, or 3D range scanning, etc.

On the other hand, some volume data can be computed in the field of scientific simulation such as the meteorology for storm prediction, or the computational fluid dynamics for the simulation of the fluid dynamics.

In addition to these fields, volume data may also be presented in discrete form, called voxels, that usually is converted from continuous geometric models by a conversion process called voxelization. In later studies, the conversion process is also applied to the applications of collision detections[3][9] and skeleton extractions[2][13].

Among the previous works, a number of related works are briefly reviewed as follows. In [7], Aggeliki Karabassi et. al. proposed a way to do voxelization by making use of the depth values stored in the Z-buffer, which measures the maximum and minimum distance to each face of the bounding box by parallel projections. However, this method may result in incorrect voxels if the object has hidden cavities.

A further improvement on computational efficiency by using hardware acceleration was reported later by Fang and Chen [11] in which the surface models are converted to volumetric on the basis of slices. It begins with the volume space defined by the bounding box covering the object then slice the object along the directions of three coordinate axises, render the sliced sections to the frame buffer, and store the results as 3D textures. The voxelization can be either surface or solid. In the former case, only the boundary voxels converted from the surfaces of input model are recorded; hence is usually called the surface voxelization. For the later case, the interior parts of the object are also filled with voxels. The slice-based method benefits from hardware acceleration but suffers from incomplete boundary and missing thin region problems

Zhao Dong et. Al.[15] proposed an efficient GPU-accelerated voxelization algorithm for complex polygonal models. Their method begins with converting the model into three discrete voxel spaces according to its surface orientation. The resultant voxels are encoded as 2D textures in three separate sheet buffers synthesized into a worksheet recording the volumetric representation. The algorithm works entirely in GPU (graphics processing unit) and achieves real-time frame rate.

Another data-parallel approach for conservative and tile-based voxelizations are proposed by [10] in which both surface and solid voxelization methods are discussed. In addition, they proposed an octree-based sparse representation of the voxel sets for better utilization of memory resources, which enables higher voxelization resolution over 4096×4096×4096.

Currently, most voxelization techniques

 ISBN：978-957-21-8031-0

proposed so far only accepts closed or watertight objects as their input. For the non-manifold meshes or those with intersecting parts, the aforementioned methods cannot be applied to derive complete and cocrrect volume data.

To solve this problem, Fakir S. Noorudin et. al. [8] proposed an election-based voxelization technique for the non-manifold meshes. To deal with the cracks and the boundaries of the input mesh, they proposed a parity count method that begins with scan conversions by orthographic projections in 13 directions then decide the final volume data by a majority vote. Furthermore, they also propose a ray stabbing method for interpenetrating subparts. Unlike previous methods, in ray stabbing process, only the first depth and the last depth samples are kept for each ray.

Moreover, a voxel is considered as an interior voxel in the final volume data only when the scan conversions in all projections classify the voxel as interior. Since these two techniques are not compatible with each other, when an input mesh contains both cracks/boundaries and interpenetrating subparts, they needs an additional repair process to seal the cracks in the first place then applies the ray-stabbing method afterwards to get the final volume data. However, their approach may bother the user with the selection of proper methods.

3. The Adaptive Voxelization

Inspired by [8], our adaptive approach roposed in this paper is capable of voxelizing arbitrary surface meshes. Our method suggests a vertex grouping pre-process to divide the interpenetrating subparts into disjoint groups according to the topology; then, solid voxelizations are performed for each individual groups in three directions. Following to the solid voxelizations, we adopt the voting strategy for the decision of the final volume data during the merge process of the voxel sets from the voxelizations of all the vertex groups.

3.1 Vertex Grouping

The 3D models are often created for computer animation and scientific simulation. A common way to construct the model by the computer animists is to build upon a set of primitives. More often, owing to some special needs such as cloth simulation, animation, and texture mapping etc., the model may consists of interpenetrating subparts.

At times, the animists might have not assign group settings for each subpart. For these models, traditional solid voxelization methods mostly failed to find the interior voxels correctly and resulted in incomplete boundaries. Hence, holes might be generated in the interpenetrating regions, which results in discontinuous volumetric data and defected visual outlook.

Moreover, a further use of such volumetric data: for example, the skeletonization, is very likely to fail owing to the defects. To solve this problem, we suggest a vertex-grouping preprocess to prevent from the holes generated in the interpenetrating regions.

The vertex grouping process is used to find connected components from the input mesh on the basis of the given topological relationship. It begins with the assignment of a unique tag value to each vertex of the mesh representing its group ID followed by a merging process that iteratively replaces the tags of the vertices with the smallest tag value of their neighbouring vertices. An example of this process is shown in Fig. 2(a)-(e).

3.2 Slice-based Voxelization

Smilar to Fang and Chen [11], our method is also a slice-based, or the frame-based, technique. Initially, the model is aligned and normalized to a

 ISBN：978-957-21-8031-0

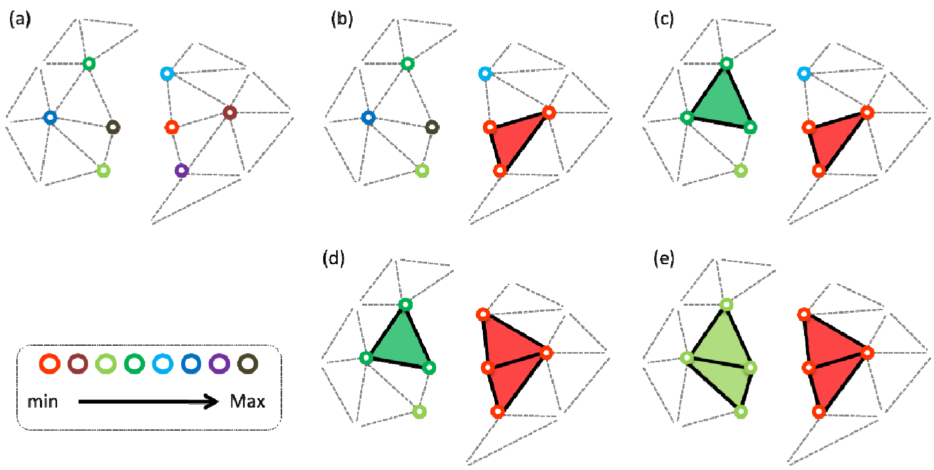

Figure 2: The vertex grouping process. : (a) assigning vertex tag ID (different color represents different tags); (b)-(d) merging connected regions by replacing the tag of a vertex with the smallest tag value in its neighbouring vertices; (e) the final result.

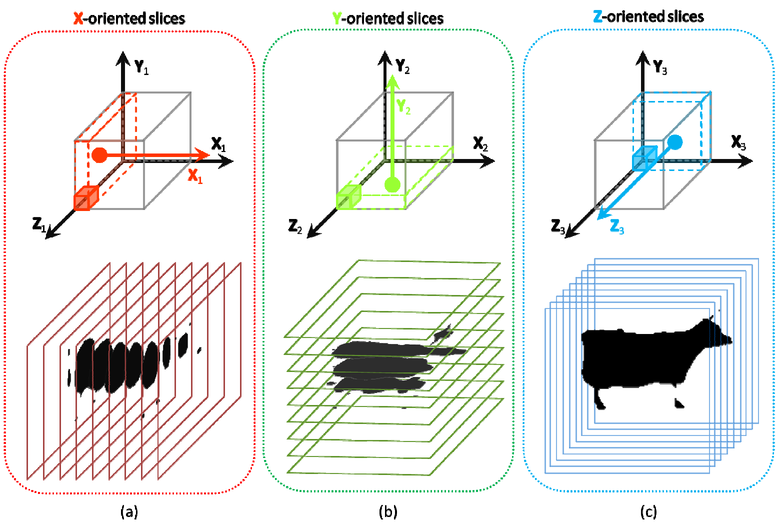

Figure 3: Adaptive solid voxelization in the direction of the (a) X-axis, (b) Y-axis, and (c) Z-axis.

box-like region that was later considered as the volume space of the model and its subparts.

By voxelizing all the subparts in the same volume space, we can easily merge of the volume data gathered from the voxelizations of the subparts without additional efforts for the alignments.

To prevent from the defects resulted by the incomplete boundaries and the thin regions, the voxelization is done individually to each subpart in the same volume space by a three-pass sliced-based solid voxelization process that converts the model along the directions of the three coordinate axises.

3.3 Voxels Synthesis

After the voxelizations of each group/subpart have completed, the voxel sets are then merged by a simple process as follows to get the final complete volumetric representation.

As illustrated by Figure 3(a)-(b), a voxel set is

ISBN：978-957-21-8031-0

Table 1 : The results of the meshes with interpenetrating parts.

Input Model			Separable Part		Solid Voxelization (Resolution : 128^3)			
					Tradition Method		Proposed Method	
Name	Vertex	Face	part	Time(sec)	Majority Voting	Time (sec)	Majority Voting	Time (sec)
Al Capone	3618	3440	21	0.148	138997	1.637	108873	14.333
teapot	3644	6320	4	0.049	200583	0.99	203370	2.505
dinopet	8145	15945	23	0.225	81624	1.883	83769	18.597
castle	6620	6493	21	0.5	131180	2.047	149012	19.163

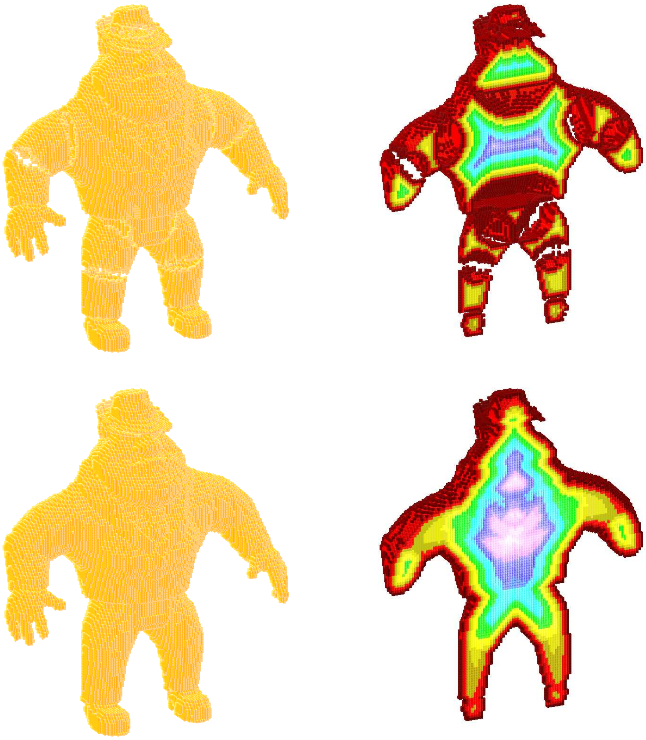

Figure 4 : Upper row: the voxelization results by Fang and Chen's method[10]; lower row: the voxelization results by our method.

derived from a single pass of the scan of a vertex-group/subpart of the model along the direction of an axis of the coordinate system of the volume space. The voxel sets derived from the same direction of scan are then merged by boolean OR. Afterwards, the final dataset is decided by majority vote on the three voxel sets.

4. Experimental Results

Our method emphasizes on its capability of dealing with non-manifold meshes and the meshes with interpenetrating parts. Hence, the test models we used in our experiments mostly are either non-manifold or comprising interpenetrations.

The experiments are performed on a PC equipped with Intel-Pentium Dual-Core CPU E5300 processor, 2GB RAM, and the ATI Radeon HD 4550 graphics processor. The program is compiled by Microsoft Visual Studio 2008 and executed in Microsoft Windows 7 operating system. The results compared with [11] are shown in Figure 4.

From Figure 1, we have observed that the trousers of the Al Capone mesh contains several

 ISBN：978-957-21-8031-0

vertex-groups and cracks. This issue is not properly handled by Fang and Chen's method[11] because they did not take care of the interpenetrations and simply derived the volume data from a single pass of sliced-based scan along Z-axis. According to the results shown in Figure 4, the rendered image shows that holes are found at the interpenetrating areas; furthermore, the sliced-view of its distance map indicates that the derived dataset is discontinuous.

This problem is caused by the duplicated boundaries formed in the interpenetrating regions, which results in incomplete boundaries and holes. Such error prone dataset may not be usable for follow-up applications such as the skeleton extraction. With our adaptive method, the result presented in Figure 4 has no such problems.

To provide more empirical evidences, a number of extra tests are given and the results are shown as follows in Figure 5-6 and Table 1. Figure 5 gives the results on meshes with no interpenetrations.

To observe the continuity of resulted volume data, a sliced-view of the distance map is presented to show the variation of the distances to the surface of the mesh. In the Figures 5-6, only a distance map along Z-axis is shown where the colors of the image represent the distance to the surface. Since the voxelization is performed along the three axises, we did not find any excessive voxel outside the mesh surface in the final dataset. For the nonmanifold meshes, the Hand and Bunny, the final volume data are closed, continuous, and have no unwanted holes. For meshes comprising many interpenetrating subparts, the Dinopet and Teapot, the final volume data are also closed, continuous, and have no unwanted holes.

The numeric data from the experiments on the meshes with interpenetrations are listed in Table 1. From Table 1, we fund that the total execution time is affected by the number of subparts, which means more subparts increases the total number of passes of scan; thus, increase the total execution time.

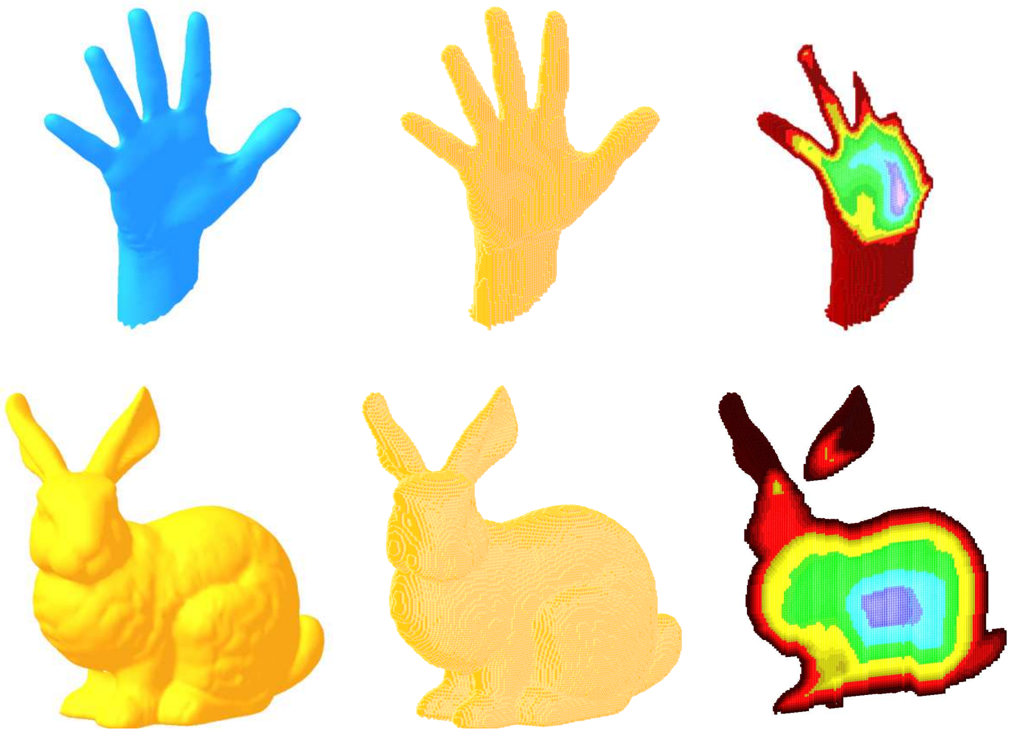

Figure 5 : The results of the hand mesh(upper) and the bunny mesh(lower): the input mesh(left), the results of adaptive voxelization(middle), the sliced-view of the distance map(right).

 ISBN：978-957-21-8031-0

Figure 6 : The results of the Dinopet(upper) and Teapot mesh(lower): the input mesh(left), the results of adaptive voxelization(middle), the sliced-view of the distance map(right).

5. Conclusion

In this paper we proposed a novel surface and solid voxelization method that is adaptive to manifold, non-manifold meshes, and the meshes comprising interpenetrating subparts. The new method successfully solved the problem introduced by the cracks of the surface and interpenetrating subparts without the use of model repair and user intervention.

According to the experimental results, the method is capable of deriving continuous volume data with complete boundaries with no erroneous and redundant voxels. The derive dataset is very useful for follow-up applications such as collision detection and skeleton extraction.

Furthermore, it is possible to use an alternative algorithm to replace the algorithm of voxelization. Provided that the efficiency is prominent, accelerations of the voxelization by parallel computing on GPU can be put into practice.

Although our method is capable of processing meshes with cracks and interpenetrations, for meshes with very thin shells or tinny features, the sampling resolution may greatly influence the correctness of voxelization.

References

[1] A. Kaufman, D. Cohen, and R. Yagel, "Volume graphics," *IEEE Computer Society*, Vol. 26, No. 7, Pages 51-64, 1993.

[2] B. Miklos, J. Giesen, and M. Pauly, "Discrete scale axis representations for 3D geometry," *ACM Trans. on Graphics*, Vol. 29, No. 4, 2010.

[3] B. Heidelberger, M. Teschner, and M. Gross, "Real-time volumetric intersections of deforming objects," *Proceedings of Vision, Modeling, Visualization 2003*, pp. 461–468, 2003.

 ISBN：978-957-21-8031-0

[4] B. Lee, J. Yun, J. Seo, B. Shim, Y.-G. Shin, and B. Kim, "Fast High-Quality Volume Ray-Casting with Virtual Samplings," *IEEE Trans. Visualization and Computer Graphics*, Vol. 16, No. 6, 2010.

[5] E. Eisemann and X. D´ecoret, "Fast Scene Voxelization and Applications," *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2006*, Pages 71–78, 2006.

[6] E. Eisemann and X. D´ecoret, "Single-pass GPU solid voxelization for real-time applications," *Proceedings of Graphics Interface 2008*, Pages 73–80, 2008.

[7] E.-A. Karabassi, G. Papaioannou, and T. Theoharis, "A Fast Depth Buffer Based Voxelization Algorithm," *ACM Journal of Graphics Tools*, Vol. 4, No. 4, Pages 5–10, 1999.

[8] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 9, No. 2, 2003.

[9] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. G. Kry, "Volume Contact Constraints at Arbitrary Resolution," *ACM Trans. on Graphics*, Vol. 29, No. 4, 2010.

[10] M. Schwarz and H.-P. Seidel, "Fast parallel surface and solid voxelization on GPUs," *ACM Trans. on Graphics*, Vol. 29, No. 6, 2010.

[11] S.-F. Fang and H.-S. Chen, "Hardware Accelerated Voxelization," *Computers & Graphics*, Vol. 24, No. 3, Pages 433–442, 2000.

[12] S.-F. Fang and H.-S. Chen, "Fast CSG Voxelization by Frame Buffer Pixel Mapping," *Proceedings of IEEE Symposium on Volume Visualization 2000*, 2000.

[13] Y.-S. Wang and T.-Y. Lee, "Curve-Skeleton Extraction Using Iterative Least Squares Optimization," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 14, No. 4, 2008.

[14] Z. Hossain and T. Moller, "Edge Aware Anisotropic Diffusion for 3D Scalar Data," *IEEE Trans. on Visualization and Computer Graphics*, vol. 16, no. 6, 2010.

[15] Z. Dong, W. Chen, H.-J. Bao, H.-X. Zhang, and Q.-S. Peng, "Real-time Voxelization for Complex Polygonal Models," *Proceedings of 12th Pacific Conference* on *Computer Graphics and Applications*, 2004.