Voxelization with opencl for Virtual Sculpting

*[1]Kayhan Ayar, [1]Gülüzar Çit, [1]Cemil Öz and [1]Soydan Serttaş
[1]Faculty of Computers and Informatics, Department of Computer Engineering Sakarya University, TURKEY

Abstract

In recent years, voxel-based models gain importance in many areas such as medical imaging, volume modeling/rendering, haptic rendering and collision detection. In this study, open computing library(opencl) is used to accelerate voxelization process of three dimensional triangle-based models. Opencl library have the ability to use central and graphics processors in a computer for a common purpose. Möller's triangle-box overlap testing algorithm is used to voxelize triangle-based models. Voxelization process is carried out by using central processor kernel as well as central processor in parallel with opencl library. The results produced by this new method are better than the voxelization techniques using only the processor. Besides, performance results are compared with single-thread traditional and multi-threaded voxelization techniques and it is shown that as the resolution is increased, performance of voxelization is getting much better than these two techniques. Improvement on voxelization time on different resolutions is also shown in the paper.

Key words: voxel, voxelization, opencl, parallel computing

1. Introduction

Engineers and designers have been using digital computers to design 3D models for years. With the decrease in computer prices, designers can design their models by 3D modeling software. But available software does not have the perception of reality and interaction like virtual reality environments have. Therefore, new virtual environments have been developed which allows 3D modeling by using virtual reality tools. This field of virtual reality is referred as virtual sculpting.

Zhang defines virtual sculpting as the process of creating interactively three-dimensional (3D) models by carving a work piece on the user's computer screen like a real sculptor can do on a piece of clay, wax, or wood [1].

The purpose of a virtual sculpting system is to allow the designers to create and reshape three-dimensional free-form objects via an interactive environment equipped with virtual reality hardware and software.

Virtual sculpting applications, are available in many areas especially for training purposes such as virtual medicine [2], dental surgery [3], design and modeling [4, 5, 6] and rapid prototyping. In our virtual sculpting system, voxel-based binary volumetric data approach is used. Thus, relevant literature is discussed below.

*Corresponding author: Address: sakarya university building number:m5 / office number:416 SAKARYA 54187 Turkey. E-mail address: kayar@sakarya.edu.tr

The first application of volumetric sculpture in the literature was proposed by Galyean and Hughes in 1991 [7]. In their study, initial model defined in uniform discrete voxel grid is edited interactively to create 3D freeform shapes. Basic operations like addition or subtraction, and several tools definitions (heat gun, sand paper or color modifier) are proposed. Kaufman and Wang have created another sculpting system where the tools are based on carving and sawing [8].

In these studies mentioned above, uniform voxel representation which voxels represented by 3D discrete set, is used. Space is divided into equal-sized units voxels and these voxels labeled according to whether they were inside the object or not.

One of the most important parts of a virtual sculpting system is the voxelization of 3D models. Voxelization is concerned with converting geometric objects from their continuous geometric representation into a set of voxels that best approximates the continuous object. As this process mimics the scan-conversion process that pixelates (rasterizes) 2D geometric objects [9], it is also referred to as 3D scan-conversion [10].

Voxelization is used on every triangle in a model and this operation can be done simultaneously. We divide model's triangles into groups and each triangle group is voxelizied by a work-unit. Voxelization process can be time consuming on complex models. The aim of this study is to decrease the voxelization time by using parallelism property of open computing library.

2. Methodolgy

In this study binary voxelization method is used to voxelize the 3D objects. The basic idea of this method is to check every voxel against the object and determine whether they covered by it or not. If a voxel is covered then its value assign to "1", if not then its value assign "0". Figure 1 shows a triangle mesh and resulting binary voxel representations of a 3D object.

The following steps are performed to achieve binary voxel data.

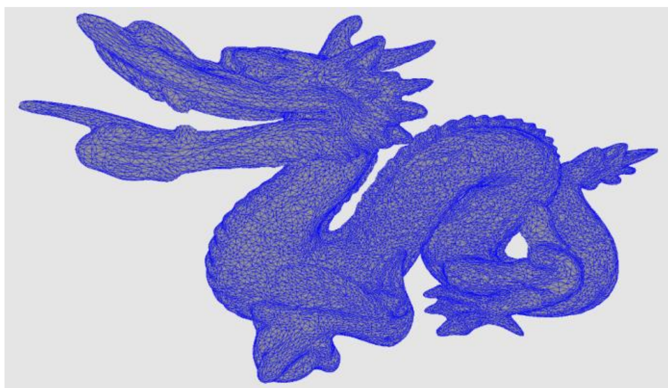
- Calculation of bounding box of the 3D object
- Creation of discrete voxel grid in this bounding box according to desired resolution and set each voxel value as "0"
- Surface voxelization: for every triangle of the 3D object

    - Calculate bounding box of the triangle

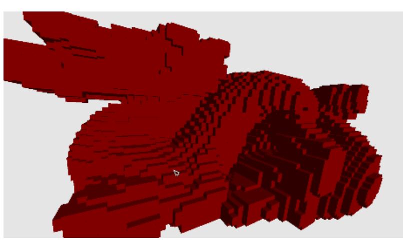
    -Apply Möller's 3D triangle-box overlap test on each voxel within this bounding box if its value is "0" [12]

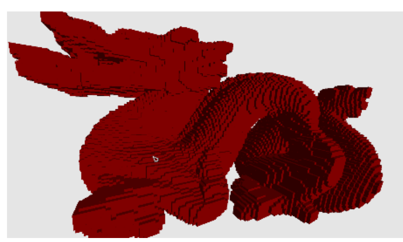
    -If voxel and triangle overlaps set this voxel value to "1"

- Interior voxelization: This step fills the interior volume of the 3D voxel model by using non-recursive 3D seed-filling algorithm [13]. Unlike the traditional seed-filling method this algorithm uses its own stack to fill the interior. Also it works only one slice (a plane of voxels) at a time. This way memory consumption is reduced.



(a)





(b) (c) (d)

Figure 1. Representation of 3D object (a) Triangle Mesh, (b) Voxelization for 64x64x64 resolution, (c) Voxelization for 128x128x128 resolution, (d) Voxelization for 512x512x512 resolution

In this study, surface and interior voxelization process is performed by multiple opencl work items simultaneously to decrease voxelization time. opencl has to ability to use both system CPU and GPU to run its code. This way, we try to increase computing capability for voxelization process.

First, triangles of the 3D object stored on a buffer and then according to total computing units, this buffer is split up into parts. Figure 2 shows how a 3D object's triangles is placed on a buffer.

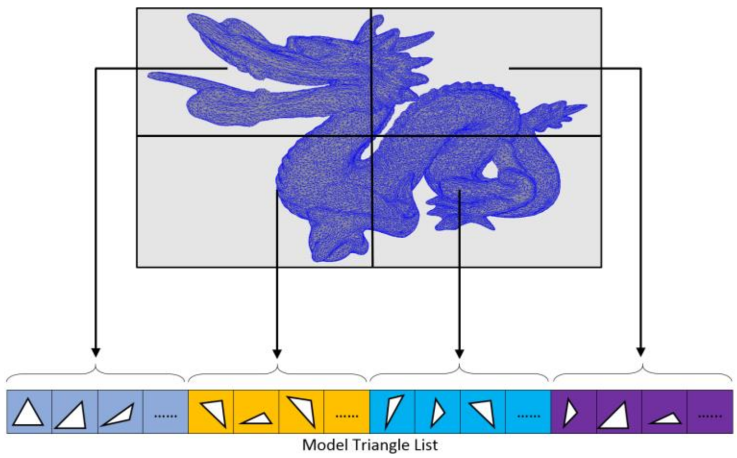


Figure 2. Triangles buffer

Each thread will work only one of these buffer parts. Figure 3 shows 8 work items voxelization process. "OpenCLManager" is responsible for initializing and controlling open computing library and its work items.

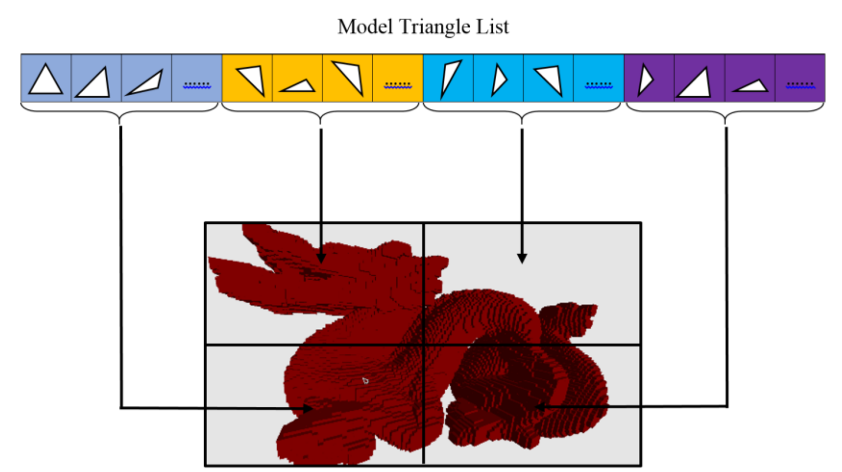


Figure 3. Four work-unit voxelization

3. Results

As shown in table 1 using open computing library multiple work items is actually decreased voxelization time. When work-item number is equal to total compute unit, time difference is maximum.

But if work item number is higher than compute unit time difference is becoming to decrease. While resolution is increasing, number of work items is becoming more important to decrease voxelization time. Main bottleneck in this method is the machine total compute unit.

All reported results were obtained on an Intel Core 2 Quad 2.50 Ghz machine.

Thread Count \ Resolution	64x64x64	128x128x128	256x256x256
1	0,4487	2,973	15,235
4	0,1887	0,962	6,072
16	0,072	0,317	0,253
24	0,453	0,172	0,124

Table 1. Voxelization time (in second) with different work-item count.

4. Discussion

This should explore the significance of the results of the work, not repeat them. The results should be drawn together, compared with prior work and/or theory and interpreted to present a clear step forward in scientific understanding. Combined Results and Discussion sections comprising a list of results and individual interpretations in isolation are particularly discouraged.

Conclusions

In this study, opencl library is used to accelerate voxelization process of three dimensional triangle-based models. Voxelization process is carried out by using central processor kernel as well as central processor in parallel with opencl library. The results produced by this new method are better than the voxelization techniques using only the processor.

Acknowledgements

This paper is prepared with the support of BAP, Sakarya University with the grant number 2010-50-02-010.

References

[1] Zhang W. Virtual Prototyping with Surface Reconstruction and Freeform Geometric Modeling Using Level-set Method; PhD thesis; 2008.
[2] Niu Q, Chi X, Leu MC, Ochoa J. Image processing, geometric modeling and data management for development of a virtual bone surgery system. Computer Aided Surgery 2008;13(1):30-40.
[3] Yau HT, Tsou LS, Tsai MJ. Octree-based Virtual Dental Training System with a Haptic Device. Computer-Aided Design & Applications 2006; 3:415-424.

[4] Perng KL, Wang WT, Flanagan M, and Ouhyoung M. A real-time 3d virtual sculpting tool based on modified marching cubes. In: Proceedings of International Conference on Artificial Reality and Teleexistence; 2001, p.64-72.
[5] Ho CC, Tu CH and Ouhyoung M. Detail Sculpting using Cubical Marching Squares. In:ICAT 05 - Proceedings of the 2005 international conference on Augmented tele-existence, p.10-15.
[6] Çit, G., Ayar, K., Serttaş, S., Öz, C., A Real-Time Virtual Sculpting Application with a Haptic Device. Turkic World Mathematical Society Journal of Applied and Engineering Mathematics 2013; 3:2:223–230.
[7] Galyean A, Hughes JF. Sculpting: An Interactive Volumetric Modeling Technique. Computer Graphics 25:4:268-274; 1991.
[8] Wang S and Kaufman AE. Volume sculpting. In:Symposium on Interactive Graphics; ACM SIGGRAPH; 1995.
[9] Foley JD, Van Dam A0, Feiner SK, Hughes JF. Computer Graphics: Principles and Practice; 2nd Edition; Addison-Wesley; pp.92-99; 1990).
[10] Kaufman A, Cohen D, Yagel R. Volume Graphics. IEEE Computer 26:7:51-64; 1993.
[11] Öz C, Çit G, Ayar K. Multithreaded voxelization method for virtual sculpting. In:European Conference of Technology and Society; 2013.
[12] Möller A,T and Haines E (2002), Real-Time Rendering. AK Peters Ltd; 2002.
[13] Jou S, Tsai M. A fast 3D seed-filling algorithm. Springer –Verlag; 2003.