2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

The Proposal of an Acceleration for Inside and Outside Judgment Using CUDA

Koki Sato†, Rei Nakagawa† and Satoshi Kodama†

†Department of Information Sciences
Faculty of Science and Technology
Tokyo University of Science
2641 Yamazaki, Noda-shi, Chiba, 278-8510, Japan
Email: 6316617@ed.tus.ac.jp, 6316627@ed.tus.ac.jp, kodama@is.noda.tus.ac.jp

Abstract—The inside and outside judgment for the coordinates of a given object is an essential concept in three-dimensional modeling. The winding number algorithm is a high-precision solution; however, it involves complicated calculations. We consider that parallel operations using Compute Unified Device Architecture(CUDA) can be used to resolve this problem. An experiment was conducted, which showed an increase in the overall speed.

1. Introduction

In general, the combination of multiple figures should be used for creating an object that comprises complicated figures. The well-known method for creating the object which is composed of the complicated figures exploits the combination of multiple figures which are used as a basic figure (e.g., box, cylinder, sphere, and cone).

This method is adopted by software packages in various cases of three-dimensional modeling due to the intuitive placement and shape transformation. The method can be easily implemented by OpenGL, Java3D, and DirectX.

The implementation of this method is associated with contact decision, which is used in the common system. Therefore, in three-dimensional space, there is no concept to determine whether a point is inside the object.

Herein, we use a solid angle as an algorithm to determine whether a three-dimensional point is inside the three-dimensional object. Although this method can appropriately perform inside and outside judgment, using multiple operations of a trigonometric function is time consuming. We conducted an experiment to demonstrate the inside and outside judgment for a three-dimensional object with a complicated shape.

2. Related works

2.1. Two-dimensional inside and outside judgment

There are various studies on the inside and outside judgment for the coordinates of a given object.[1]

In the case of two-dimensions, there are two well-known methods: the crossing number algorithm and the winding number algorithm.

2.1.1. Crossing number algorithm

The crossing number algorithm is a simple and fast technique for inside and outside judgment. The judgment element of this algorithm is defined as the number of times a half line with the edge of the target object. A half line is drawn from a given judgment point to any direction.

Fig.1 shows an example of the inside and outside judgment of the crossing number algorithm. In case (a), a half line and the target object edge cross in odd number time, thereby allowing us to judge whether the point is inside. In case (b), a half line and the target object edge cross in even number time, thereby allowing us to judge whether the point is outside.

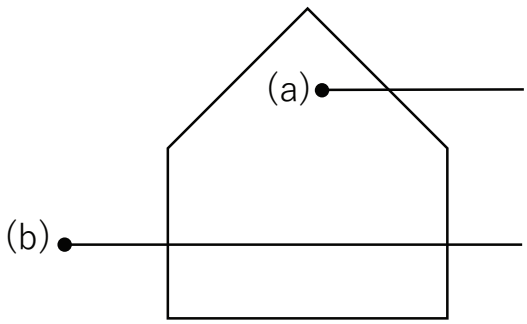

Fig. 1 An example where it can be determined appropriately

The crossing number algorithm is simple and fast, as mentioned earlier; however, the algorithm may judge incorrectly under specific conditions because the judgment result depends on the location of the given judgment point. Fig. 2 shows the point that may be incorrectly judged; therefore, it is not suitable for an exact judgment.[2]

2.1.2. Winding number algorithm

The winding number algorithm is a high-precision technique involving complicated calculations. [3]

The judgment element of the winding number algorithm is defined as the sum of plane angles. Each plane angle is

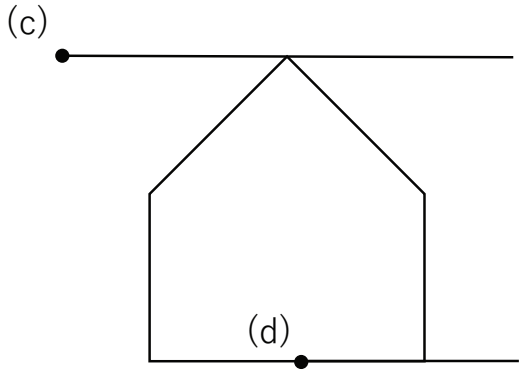

Fig. 2 An example where it cannot be determined appropriately

an interior angle at a triangle's given judgment point. The triangle is constructed by sequentially tracing all the edges of an object from any vertex around a given point in a two-dimensional surface.

Fig.3 shows a trace of the inside and outside judgment of the winding number algorithm. There is a triangle to be formed by the vertexes of the given points, V_i and V_{i+1}. V_i is the starting point of the trace, and V_{i+1} is the ending point of the trace.

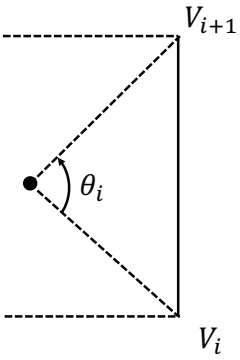

Fig. 3 Interior angle at a triangle's given judgment point

Moreover, the judgment n-square result is expressed as follows:

$$S = \sum_{j=1}^{n} \theta_j. \tag{1}$$

The above equation expresses that for a given judgment point, when the sum of the plane angle is 2π, we can judge the point to be inside; however, when the sum of the plane angle is not 2π, we can judge the point to be outside.

Therefore, in Fig.4, we can judge the point to be inside, whereas in Fig.5, we can judge the point to be outside.

Thus, compared to the crossing number algorithm, the winding number algorithm involves more complex calculations. However, the high precision of the winding number algorithm has attracted considerable attention as a potential application to more complicated models.

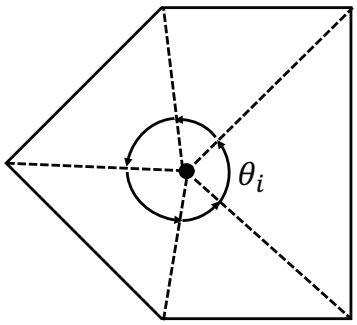

Fig. 4 An example of the judgment point being inside

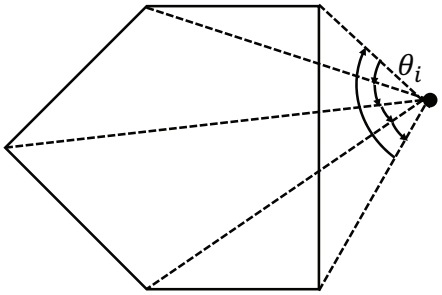

Fig. 5 An example of the judgment point being outside

2.2. Expansion of three-dimensional inside and outside judgment

Here, we describe expansion to three-dimension with regard to the winding number algorithm, which is the primary objective of this study.

2.2.1. Winding number algorithm

In the three-dimensional winding number algorithm, the judgment method cannot use the angle for a vertex in two-dimension. Therefore, the method calculates the solid angles among the vertexes of the target object. A solid angle is defined using the projection of each vertex to a sphere. [4] Fig.6 shows an example of projecting a sphere from the triangle polygon. The calculation method for this solid angle depends on the following equation:

$$S = (\theta_1 + \theta_2 + \theta_3 - \pi). \tag{2}$$

θ_i is the vertex angle i(where $i = 1, 2,$ and 3) in the spherical triangle.

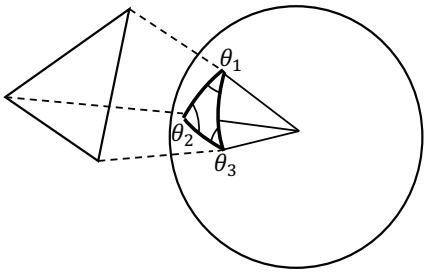

Fig. 6 When projecting a sphere from the polygon

The solid angle S_{all} of the vertex size for all planes is expressed as follows:

$$S_j = \left(\sum_{i=1}^{n} \theta_i - \pi \right) \qquad (j \geq 1). \tag{3}$$

$$S_{all} = \sum_{j=1}^{m} S_j. \tag{4}$$

Here, in equation (4), m is the number for the polygon comprosing the object.

The above equation expresses that for a given judgment point, when the sum of the solid angle is 4π, we can judge the point to be inside. However, when the sum of the solid angle is not 4π, we can judge the point to be outside. Fig.7 shows an example of the judgment point being inside.

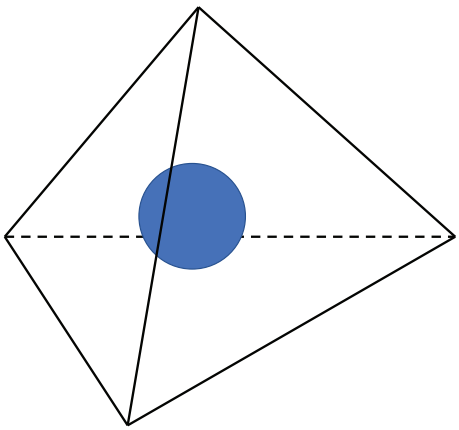

Fig. 7 An example of the judgment point being inside

The winding number algorithm is known to be computationally expensive when using a trigonometric function.

2.3. Existing technique by parallel operation

This study proposes a fast method to calculate the inside and outside judgment in three-dimension. The method performs parallel operations using Compute Unified Device Architecture(CUDA) developed by NVIDIA. CUDA is a general-purpose parallel calculation platform and programming model for performing calculations using a GPU(Graphics Processing Unit). [5]

It is easy to introduce a GPU into a system because GPUs are relatively cheap and spread widely used. Furthermore, CUDA shows remarkable parallel operation speeds for an optimized calculation algorithm. Therefore, the calculation method employing CUDA be used for fast inside and outside judgment without using large-scale facilities or machines.

3. Implementation and experiment

3.1. Experiment

We performed a few experiments to confirm the effectiveness of parallel operation. Table. 1, Table. 2,and Table. 3 summarize the experimental environment.

Table. 1. Experimental environment

Components	Specifications
CPU	Intel Core i7-6700(3.40 GHz)
Memory	64 GB(DDR4-2400)
OS	Windows 10 Ent.
Compiler	Visual C++ 2010

Table. 2. Specification for Quadro M2000 [6]

Components	Specifications
Processor	NVDIA CUDA 768 core
Memory	4 GB(GDDR5)
Memory Band width	128 bit
System Interface	PCI Express 3.0*16
Max Power Consumption	75W
Graphics APIs	DirectX 12 OpenGL 4.5 Shader Model 5.0
Compute APIs	CUDA DirectCompute OpenCL

Table. 3. Environment for experiments(CUDA)

Components	Specifications
CUDA Driver Version / Runtime Version	8.0 / 8.0
CUDA Capability Major / Minor Version Number	5.2
Maximum number of threads per block	1024
Max dimension size of a thread block	(1024,1024,1024)
Max dimension size of a grid size	(2147483647, 65535, 65535)

We calculated the time required to complete the inside and outside judgment for the primitive figure wherein each component of the three-dimensional coordinates is from -50 to 50(integer type).

3.2. Experimental results

Table. 3 lists the processing time of the CPU and GPU for an object with a fixed number of vertexes.

The experimental results, indicated that the proposed method achieved faster processing time because the parallel operation programming model was effective.

Fig. 8 and Fig. 9 show the results of the inside and outside judgment.

Table. 3. Experimental results for each figure

Object	Vertex	CPU time[ms]	GPU time[ms]
Cone	12	7275	727
Sphere	92	60504	4470

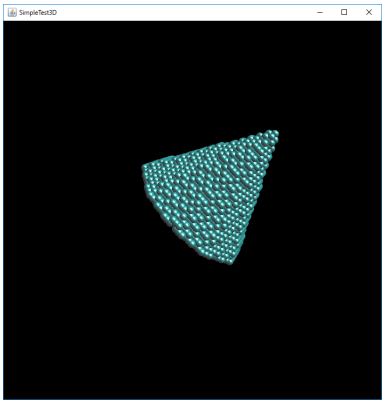

Fig. 8 Cone

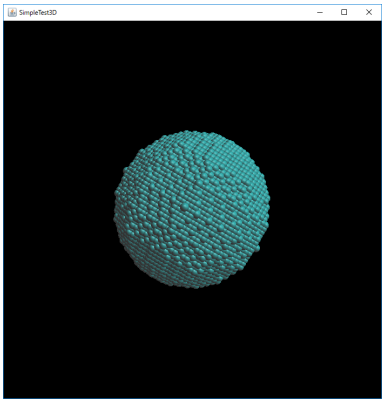

Fig. 9 Sphere

4. Conclusions

The inside and outside judgment for three-dimensional object is an important technique in various fields, such as 3D printing, AR, and VR. However, some methods face difficulties in dealing with great computational complexity while handling three-dimensional data. In particular, the winding number algorithm required a large amount of time when calculating a trigonometric function. This research aimed at handling the calculation with a GPU and proposed the use parallel operations by implementing CUDA. We considered that the parallel operations can be used as a solution to the stated problem. We conducted an experiment involving the inside and outside judgment of the winding number algorithm in three-dimension in order to perform the parallel operation of a trigonometric function for each judgment point. In conclusion, the experimental results showed an increase in the overall speed for the inside and outside judgment.

Acknowledgments

The authors would like to express their hearty thanks to NOLTA2017 organizers.

This work was supported by JSPS KAKENHI Grant Number JP16K21399.

References

[1] Shinobu Nagashima, “A Spatial Inclusion Test for Polyhedra Using a Spherical Projection,” *Transactions of Information Processing Society of Japan*, 27,7, 744–746, 1986.

[2] Yehuda E. Kalay, “Determining the Spatial Containment of a Point in General Polyhedra,” *J. Atoms. Sci.*, vol. 19, 303–334, 1982.

[3] David G. Alciatore and Rick Miranda, “A winding number and point-in-polygon algorithm,” *Technical report, Colorado State University*, 1995.

[4] Nakayama Atsushi, Kawakatsu Daisuke, Kobori Kenichi and Kutsuwa Toshirou, “A checking method for a point inside a polyhedron in grasping an object of VR,” *The 48th National Convention of Information Processing Society of Japan(IPSJ)*, 2, 297–298, 1984.

[5] NVIDIA Corporation,“CUDA Architecture”, *http://www.nvidia.com/object/cuda_home_new.html*

[6] NVIDIA Corporation,“Datasheet Quadro M2000”, *https://images.nvidia.com/content/pdf/quadro/data-sheets/38111-DS-NV-Quadro-M2000-Feb16-US-NV-Fnl-HR.pdf*