Comparison and Multi-GPU based Implementation of a Collision Detection Method with Z-buffer in Cyber Space

Tomoaki Iida, Hidemi Yamachi, Yasushi Kambayashi and Yasuhiro Tsujimura
Department of Computer and Information Engineering,
Nippon Institute of Technology, 4-1 Gakuendai Miyashiro-cho, Minamisaitama-gun, Japan

Keywords: Z-buffer, Collision Detection, Object Detection, Cyber Radar.

Abstract: We propose a new technique to detect objects and collisions of geometric objects in cyber space. This technique uses depth values of the Z-buffer in rendering scene, and contributes to construct an intelligent interface for mobile software agent that uses augment reality. We use the orthographic projection for collision detection. Our method uses two depth value sets. One is obtained through rendering the cyber space from the sensor object toward a target point. This set does not have the depth values of the sensor object. Another one is obtained through rendering only the sensor object in the reverse direction. From these two depth value sets, we obtain the distance between the sensor object and others for each pixel. This technique requires only one or two rendering processes and it is independent from the complexity of object's shape, deformation or motion. In this paper we evaluate the efficiency in comparison with one of the most popular collision detection method. GPU based methods have advantage for utilizing multi-GPU system. In order to take the advantage, we have implemented our method for multi-GPU system and evaluate the performance of collision detection.

1 INTRODUCTION

In the last decade, robot systems have made rapid progress not only in their behaviors but also in the way they are controlled. We have demonstrated that a control system based on multiple software agents can control robots efficiently (Kambayashi et al., 2009). We have constructed various mobile multi-robot systems using mobile software agents.

In order to augment the usability and efficiency, Kurane et al. propose an intelligent interface that enables for users to assist the behaviors of the mobile software agents by visualizing the mobile software agents using augmented reality (AR) (Kurane et al., 2013).

When implementing AR for mobile software agents, it is important to detect their collisions in order to construct natural scene with co-existing multiple agents on a robot. We have proposed an algorithm for the collision detection of geometric objects named *Cyber Radar* (Yamachi et al., 2011). In this paper, we describe the details of our method and show the efficiency of it through the comparative experiment with one of the most popularly used collision detection method. We also implement into multi-thread with multi GPU and evaluate the performance by numerical experiments.

The rest of this paper is organized as follows. In the next section, we briefly review related works. In section 3, we describe the details of our object detection algorithm. We show the efficiency of our method through numerical experiments in section 4. We conclude our discussions in section 5.

2 RELATED WORKS

Since 1980s, many researches have been conducted to solve the collision detection problems. There are two basic ideas for collision detection algorithms. One technique uses bounding volume hierarchy (BVH) which consists of convex hulls. Convex hulls hierarchically cover an object from the whole to each part. AABB (Bergen, 1997), Bounding Sphere (Hubbard, 1996), OBB (Gottschalk et al., 1996) are well known and being used. Because those techniques detect collisions only one pair of objects at once, they have $O(n^2)$ computational complexity for n objects.

Iida T., Yamachi H., Kambayashi Y. and Tsujimura Y..
Comparison and Multi-GPU based Implementation of a Collision Detection Method with Z-buffer in Cyber Space.
DOI: 10.5220/0004331204350438
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 435-438
ISBN: 978-989-8565-38-9
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

3 MECHANISM OF CYBER RADAR

In order to detect other objects in the target area, we set the view point at the sensor object (Figure 1). By defining the orthogonal view volume which has the same depth as the length of the sensor object and the distance to move, as the target area, the scene is rendered without the sensor object (Figure 2). We call the obtained depth value set *Range Distance Scope* denoted as **R**. If there are any pixels in the *Range Distance Scope* that have the depth values less than the target area distance ***d***, there are some objects in the target area.

If some objects are detected in the target area, the collision detection is performed. The view point is set at the other side of the target area with the direction toward the sensor object. Then only the sensor object is rendered to obtain the depth value set. We named the depth value set *Mask Distance Scope* denoted as **M**. With these two depth value sets, we can obtain the distances between the sensor object and other objects. We named this distance value set *LookAt Distance Scope* denoted as **L**.

Before we describe the detail algorithm of *Cyber Radar*, we define the following notations.

p_w, p_h : Pixel width and height of depth values $i = 1,2,\dots, p_w, j = 1, 2, \dots, p_h$

R: *Range Distance Scope* , a set of R_{ij}

M: *Mask Distance Scope* , a set of M_{ij}

L : *LookAt Distance Scope* , a set of L_{ij}

d : Target area distance

The distance between prove point and a target object is $d - R_{ij}$ and the distance from the end of the target area to the sensor object is $d - M_{ij}$. Using these values, the pixel value of *LookAt Distance Scope* becomes as follows.

$$L_{ij} = d - (d - R_{ij}) - (d - M_{ij}) = R_{ij} + M_{ij} - d \qquad (1)$$

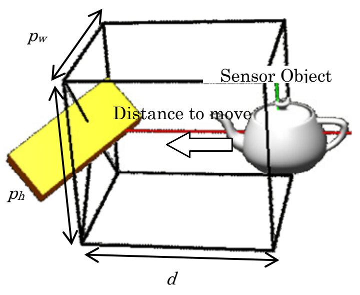


Figure 1: Definition of viewing volume.

If *min*(**L**), the minimum value of **L**, is less than the distance to move, the collision will occur at the pixel. The time of collision is calculated from *min*(**L**) and the velocity of objects.

The algorithm of *Cyber Radar* is described as follows.

```
1. Set up the rendering conditions (Figure 1)
   View point at the sensor object
   Rendering direction toward sensor object to move
   width and height of the target area
   d := Target area distance
   p_w, p_h := Projection width and height
2. Render scene without the sensor object (Figure 2)
3. Obtain R (Range Distance Scope)
4. if min(R) = d
     return false (no object)
   else
     return true (objects are detected)
5. Set the rendering conditions (Figure 3)
   View point at the end of the target area
   Rendering direction toward sensor object
6. Render scene only sensor object
7. Obtain M (Mask Distance Scope)
   Swap right and left pixel values
8. Calculate L (LookAt Distance Scope) (Figure 4)
```

$$\mathbf{L} := \mathbf{R} + \mathbf{M} - d \qquad (2)$$

```
if min(L) > Distance to move
    return false (no collision)
else
    return true (collision is detected)
```

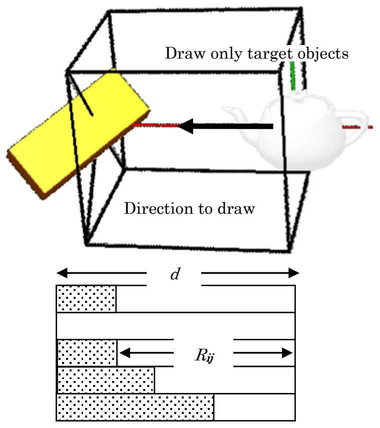


Figure 2: *Range Distance Scope*.

4 NUMERICAL EXPERIMENTS

We conducted two types of experiments. First, we compare the performance of *Cyber Radar* with Axis-aligned bounding box (AABB). We use OPCODE for AABB collision detection implementation. Figure 5 shows the experiment scene. We allocate rectangular shaped objects and change the number of objects from 10,000 to 30,000 by 2,000 to obtain the time for collision detection. The environment is Windows7, CPU Intel Core i7 3.4GHz , GPU NVIDIA GTX 560 , 4Gbyte memory.

Table 1 and Figure 6 show the results of the experiment. The time of AABB varies almost linearly while that of *Cyber Radar* is almost constant when the number of objects is more than 16,000. Since the performance of *Cyber Radar* largely depends on the GPU, *Cyber Radar* is effective than AABB for large number of objects to be tested collision detection.

GPU based methods have advantage by utilizing multi-GPU system. In order to take the advantage of multiple GPU, we have implemented our method for multi-GPU system based multi-thread program and evaluate the performance of collision detection.

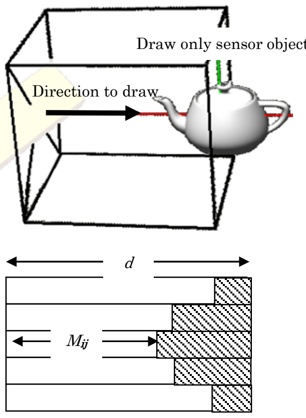


Figure 3: *Mask Distance Scope.*

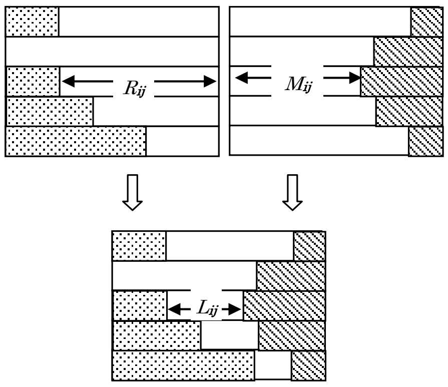


Figure 4: *Look At Distance Scope.*

We compare the time to complete the collision detection by single-thread and multi-thread procedures. In the case of multi-thread, the time means between the start time of the first thread and the finish time of the last thread. We have used two GPUs and generate multi-threads for collision detection. The experimental environment is WindowsXP, CPU Intel Core i7 3GHz, GPU NVIDIA Quadro2000, 3Gbyte memory.

We have employed five to thirty teapots with 6256 polygons moving through many fixed boxes. All of tea pots are sensor objects and detect collision at each time step. The results are shown in Figure 7. Two GPU cases take about half of one GPU cases for each sensor object number.

5 CONCLUSIONS

We propose a novel technique to detect objects and collisions of geometric objects in cyber space. This technique uses depth values of the Z-buffer in rendering scene. We use the orthographic projection for collision detection. By using this method, we should be able to produce natural AR scenes for the intelligent human computer interface we are building.

We have conducted experiments to evaluate the efficiency of *Cyber Radar* in comparison with AABB, one of the most popular collision detection methods. The results show that the *Cyber Radar* is more effective than AABB in case of more than 16,000 objects for collision detection.

We also implemented *Cyber Radar* as a multi-thread procedure with multiple GPUs and evaluate the difference of collision detection time with single GPU procedure. Two GPU procedure performs in half of time that of the single GPU procedure.

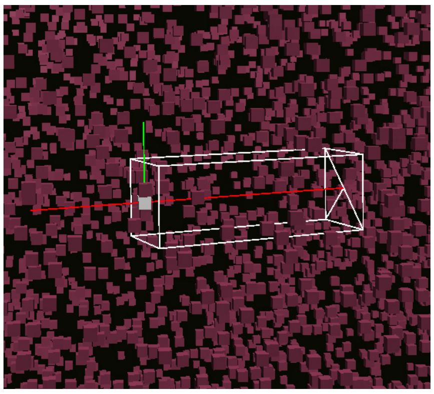


Figure 5: Scene of experiment for collision detection. White rectangle is target area of *Cyber Radar*.

Table 1: Percentage of detection time of *Cyber Radar* (CR) for AABB.

	10,000	12,000	14,000	16,000	18,000	20,000	22,000	24,000	26,000	28,000	30,000
CR(msec)	14.73	16.54	23.75	27.66	27.37	27.44	28.82	26.66	29.21	28.90	31.52
AABB(msec)	11.27	14.53	19.59	25.68	32.72	40.92	54.22	58.88	70.15	81.85	85.48
CR/AABB(%)	130.72	113.84	121.21	107.72	83.63	67.04	53.15	45.28	41.64	35.31	36.88

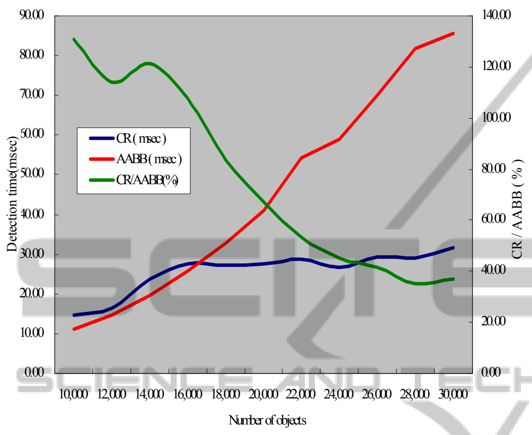
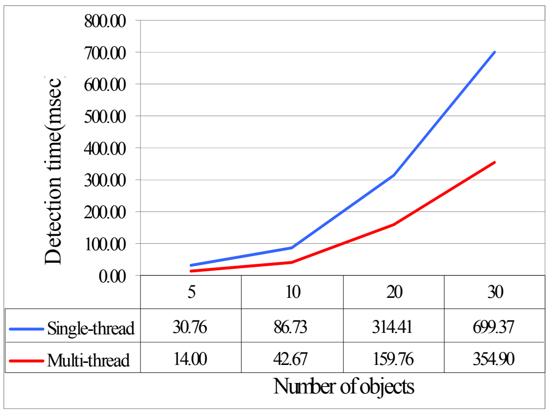


Figure 6: Detection time for different numbers of objects for *Cyber Radar* (CR) and AABB.



	5	10	20	30
Single-thread	30.76	86.73	314.41	699.37
Multi-thread	14.00	42.67	159.76	354.90

Figure 7: Performance of multi-GPU based procedure.

Producing natural AR scenes may include lots of complex objects that have to be tested for collision. The results of experiments show that *Cyber Radar* is effective to produce such scenes. However, detecting many collisions with *Cyber Radar* takes much computational cost. In order to mitigate this problem, we have implemented multiple GPU procedure. Through the experiments we can confirm that *Cyber Radar* is easy to implement.

Applying this method for constructing the intelligent human computer interface is our next aim.

ACKNOWLEDGEMENTS

This work is partially supported by Japan Society for Promotion of Science (JSPS), with the basic research program (C) (No. 23510178), Grant-in-Aid for Scientific Research.

REFERENCES

Bergen, G. van den. (1997). Efficient Collision Detection of Complex Deformable Models using AABB Trees. *Journal of Graphics Tools,* vol.2, no.4, 1-14.

Gottschalk, S. Lin, M. C. Manocha, D. (1996). OBBTree: A Hierarchical Structure for Rapid Interference Detection, *Proceedings of ACM SIGGRAPH '96*, 171-180.

Hubbard, P. M. (1996). Approximating Polyhedra with Spheres for Time-Critical Collision Detection, A*CM Transactions on Graphics,* vol. 15(3), pp.179-210.

Kambayashi, Y., Ugajin, M., Sato, O., Tsujimura, Y., Yamachi, H. and Takimoto, M. (2009). Integrating Ant Colony Clustering Method to Multi-Robots Using Mobile Agents, *Industrial Engineering and Management System*, vol.8, no.3, 181-193.

Kurane, K., Takimoto, M. and Kambayashi, Y. (2013). Design of an Intelligent Interface for the Software Mobile Agents Using Augmented Reality, *Proceedings of International Conference on Agents and Artificial Intelligence*, to appear.

Yamachi, H. Souma, Y. Kambayashi, Y. Tsujimura, Y. and Iida T. (2011). Evaluation of a Technique for Collision and Object Detection with the Z-buffer in Cyber Space. *Proceedings of International Conference on Cyberworlds 2011*, 85-92.