COLLISION DETECTION FOR DEFORMABLE OBJECT USING BOUNDING SPHERE HIERARCHY

CHING SUE PING

THIS DISSERTATION IS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE WITH HONOURS

MATHEMATICS WITH COMPUTER GRAPHICS PROGRAMME
SCHOOL OF SCIENCE AND TECHNOLOGY
UNIVERSITI MALAYSIA SABAH

2013

DECLARATION

I affirm that this dissertation is of my own effort, except for the materials referred to as cited in the reference section.

_____

CHING SUE PING
(BS 10110066)
19 MAY 2013

CERTIFIED BY

Signature

SUPERVISOR

(DR. ABDULLAH BADE) _____

ACKNOWLEDGEMENT

First of all, I would like to take this opportunity to sincerely express my appreciation to my supervisor, Dr. Abudullah Bade for sacrificing his valuable time to give comments and suggestion throughout the preparation of this project. I have successfully gained a lot of knowledge during the guidance under him.

Secondly, I would like to thank my fellow friends whose spend their time and patience for the shared of knowledge, experiences and advices which lead a success in this project.

Last but not least, I would like to express my deepest gratitude to my family on their morale support when conducting this project. Besides, their financial support throughout this project and my three years degree course are much appreciated.

ABSTRACT

Collision detection has been widely used in the industry of Computer Graphic. The application of this technique on deformable objects has slowly getting attentions in the field of Computer Graphic. One of the deformable objects which have been widely used is cloth model. Simple mass spring model is used to model the cloth where the movement of the particles within the cloth was controlled by applying the Newton's second law. After the modeling stage, implementation of the collision detection algorithm on cloth has been done. The collision detection technique used is bounding sphere hierarchy. Then, quad tree is used to partition the bounding sphere with the collision search was based on the top-down approach. A prototype of the collision detection system is developed on cloth simulation and several experiments were conducted. Time taken for this system to be executed is around 235.258 milliseconds. Then the frame rate is at the average of 22 frames per second which is close to the real time system. Time taken for the collision detection system travels from root to nodes were 23 seconds. The OpenGL functions call test had also been conducted to show the number of functions called during the execution. This test showed that the highest number of functions call can reach up to 15,000 calls per frame which explain why the frame rate is lower than the frame rate for AABB hierarchy when making the comparisons. While the average CPUs utilization took around 34%. The number of vertices called per frame is around 42,000 calls. As a conclusion, the computational cost for bounding sphere hierarchy is much higher because the bounding sphere requires more vertices for generation process, however the execution time for bounding sphere hierarchy is faster than the AABB hierarchy.

CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATION

3D	3 Dimensional
AABB	Axis-Aligned Boundign Boxes
BSP	Binary Space Partitioning
BV	Bounding Volume
BVH	Bounding Volume Hierarchy
DOP	Discrete-oriented Polytopes
FEM	Finite Element Model
FFD	Free-form Deformation
GJK	Gilbert-Johnson-Keerthi
LDI	Layered Depth Image
OBB	Object-oriented Bounding Boxes
PIHM	Penalty Impulse Hybrid Method

CHAPTER 1

INTRODUCTION

1.1 Preamble

In this modern era, computers are commonly used for modeling and simulation as both the computational speed and realism of the models are constantly improving. The term "deformable objects" are objects in which their shape will change whenever there are external forces being applied on the object, but the shape of the object will return to its original state when there are no other forces applied. (Sulaiman & Bade, 2011). Deformable objects are mostly used in modeling fabric, facial expression or even water wave simulation. By using mathematical and computational techniques such as Bézier Curve, Splines, Free-form deformation and mass spring models, deformable objects can be formed efficiently (Gibson & Mirtich, 1997).

Free-form deformation (FFD) is a technique generally used in modeling deformable objects. FFD allows is greater control in terms of adjusting individual points compare to splines because it deform the space of the object to changes the shape of object which is why FFD is widely used in many graphical representations (Gibson & Mirtich, 1997). Modeling deformable objects using mass-spring models is much more complicated than FFD as these require a lot deeper understanding of mechanics when modeling it. However, this type of model can be used to develop more complex models such as those used for facial expression.

Collision detection is one of the most frequently used techniques in computer graphics to detect the intersection between two or more objects. Collision occurs when there is at least one point of the two objects is overlapping (Jiménez et al., 2001). Collision detection is most commonly used in animation, modeling and simulation. As the time past, deformable collision detection had been commonly used in the field of computer graphics. Due to the recent efforts by researchers in solving collision detection problems, there has a great improvement in terms of speed and memory usage. Although there are many algorithms developed to detect the collision between objects, most of these algorithms which apply to deformable object are only valid for the rigid bodies (Teschner et al., 2005).

Nowadays there are many applications using deformable collision detection for simulating the movement for cloth, surgical procedures and animated objects (Larsson & Akenine-Möller, 2005). In cloth simulation, the collision detection is applied to the surface of the cloth and also to determine whether the object is in contact with iteself. Furthermore, if the simulation involves animated character, the collision between the character and cloth also need to be detected. This technique has also been applied to simulate the interaction between cloth and 3D objects to predict how the cloth will deform after a collision as shown in Figure 1.1.

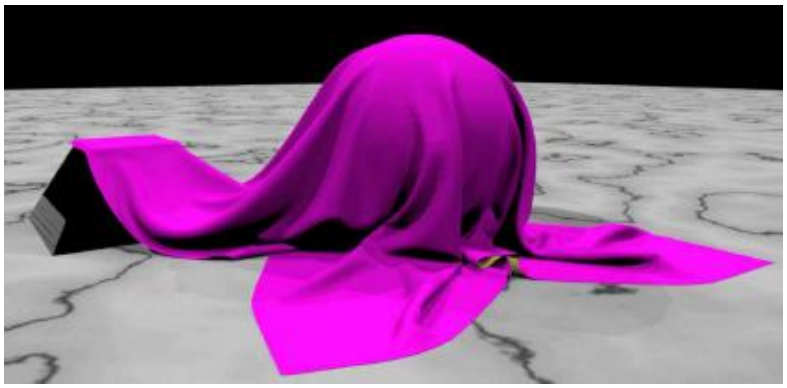

Figure 1.1 Cloth simulations on solid object (Source: Teschner et al., 2005)

As for surgical simulation, collision detections are needed on both the deformable organ and the surgical tools. Furthermore, the collision response has to be processed in order to get an accurate and efficient result. The complexity of this simulation increases as it requires a lot of responses from cutting, bleeding and fracture effect. There are many factors need to be considered when performing

surgery simulation such as material properties of the surgical tools, external forces , gravitational force and the environment issues (Teschner et al., 2005).

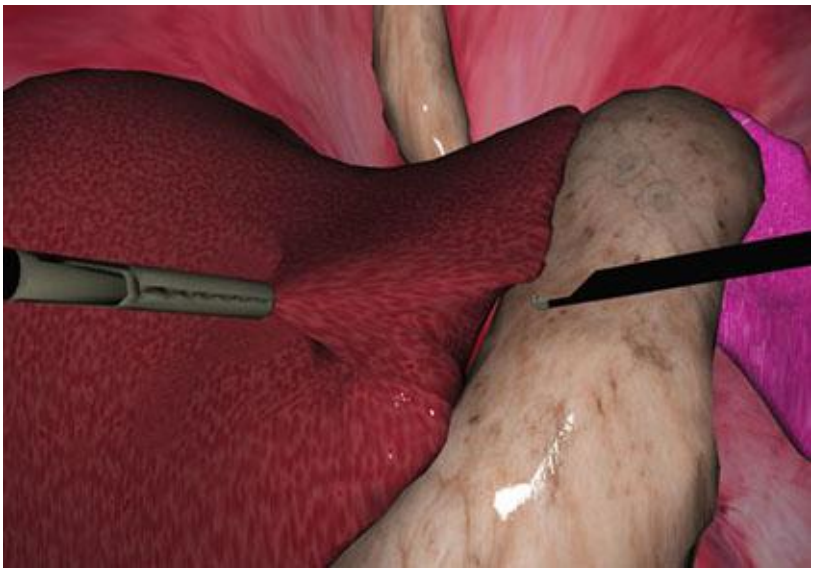

Figure 1.2 Surgical simulations (Source: http://www.inf.ufrgs.br/~amaciel/old_ projects/surgerySimulation.jpg)

The commonly used algorithms for collision detection are Sweep and Prune algorithms, Lin-Canny Bounding Volume, Gilbert-Johnson-Keerthi (GJK) Distance and Bounding Volume Hierarchy (BVHs) algorithms. Among these algorithms, BVHs has been proven to be one of the most efficient data structures for collision detection. There are many types of bounding volume been studied in the past, for examples, spheres, axis-aligned bounding boxes (AABBs), object-oriented bounding boxes (OBBs) and discrete-oriented polytopes (DOPs) (Teschner et al., 2005). Figure 1.3 illustrated the example of bounding volume.

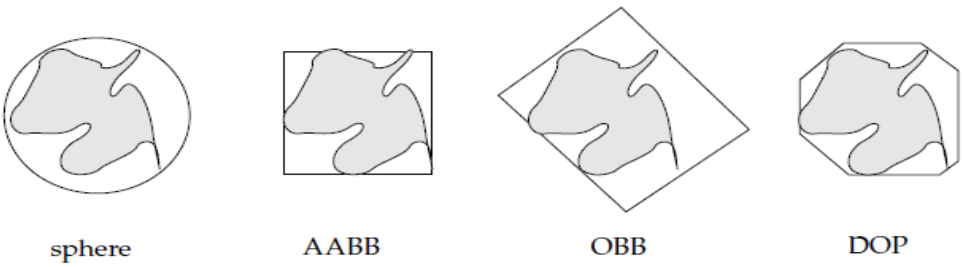

Figure 1.3 Types of bounding volume introduced (Source: Teschner et al.,2005)

In this project, collision detection for deformable objects will be simulated using the method of Bounding Volume Hierarchy. Since simulating deformable

objects requires a frequent update or detection of the object during deformation, a quick and accurate system is needed. During the simulation, the extent of the tree will be updated while the structure of the tree will remain the same so there will be no wastage of memory usage. Therefore, DOPs and OBBs for collision detection in deformable object had proven that they can provide faster update comparing to other BVHs (Bergen, 2004). However, the expensive computational cost make these two bounding volume unfavorable in developing the collision detection system.

There are 3 different strategies to build BVHs which are top-down, bottom-up and insertion. Among these strategies, top-down strategy is commonly used for collision detection of deformable object. The top-down strategy is the repetition of splitting the sets of the object parts until the smallest detectable points is reached (Teshner et al., 2005).

Other than that, the addition of collision responses of the object will make the whole simulation for deformable object to be in realism. Collision responses are the reactions of the object after the collision has been detected. The reactions are normally the motion of the object and change in shape. Sometimes, the response can also be in the way that the object deform then return to original shape. Furthermore, the final resting place is also one of the responses that the object could react.

1.2 Problem Background

Bounding Volume Hierarchy was initially used for collision detection in rigid body but it can also be used in detecting intersections in deformable objects. In the case for deformable objects, the hierarchy needs to be updated rapidly which cost burdens to the memory usage, hence making it to be less efficient. Applying a simple, efficient and fast algorithm will reduce the memory usage. A fast and accurate system was said to be impossible as the accuracy of detecting collision will decrease as the time required for the detection decreases. BHVs is said to be the most efficient algorithm in determining the collision detection (Teschner et al., 2005).

Since the BHVs is used in the collision detection system, a balanced and stable tree is needed as the update of the tree has to be done to check the latest

active nodes of BHVs. When there are many objects intersecting, an accurate collision detection will requires a lot of time just to determine the intersecting area and points because the area of environment is too large for the check of intersections (Sulaiman & Bade, 2011). Therefore, a suitable BVHs needed is depending on the constraint which the creator needed.

In order to make the simulation of the deformable objects to be more realistic, collision detection and self collision techniques need to be considered. The self collision detection is the intersection of the points within the object itself (Lau & Chan, 2002). Self collision needs to be detected efficiently on both the front and backs faces even though the backs of the object cannot be seen. The self collision for rigid body model can be ignored but it is not advised to ignore the self collision for deformable object as this will decrease the realism of the whole system (Teshner et al. 2005).

The volumetric collision detections will increase the realism of the system when simulating the collision detection of deformable object. To perform the volumetric collision detection by using BVHs, additional steps and memory usage are needed. Hence, applying other methods such as Layered Depth Image (LDI) is required for the simulation. However, this will increase huge load for the computer and affect the computational efficiency. Therefore, the volumetric collision detection is neglected in this simulation (Heidelberger et al., 2003).

The responses given after the collision detections play an important role especially on the deformable objects. Interactivity becomes the major key because after the collision happens, deformable objects need to undergo the deform state and returns to its original states. This process is important in making a more realistic system. Therefore, a quick and accurate algorithm is needed but there are no algorithm found to be perfects for all responses since different system requires a different set of reactions. (Bade et al., 2005).

There is an concerning issue of collision responses apart from the motion and changes of shape of the object, which is the final resting point of the objects after the collision. Developing a more nature and real collision system requires

combination of many responses on a single collision which may lead to inefficient and inaccurate system. Penalty Impulse Hybrid Method (PIHM) is the solution for this issue since PIHM is a fast and stable algorithm but it also burdens the whole system (Bade et al., 2005).

The gigantic graphics hardware will increase the computational efficiency and the times can be reduced when rendering the simulation of collision detection for deformable object. Over the past year, many improvements have been made. This includes collision detection algorithms for deformable object which required high performance can be done faster by using this graphics hardware. However, this tremendous graphic hardware is costly and it required a huge space to settle it down (Larsson & Akenine-Möller, 2005).

Figure 1.4 IBM's Blue Gene/L supercomputer (Source: http://news.cnet.com /Photo-2-IBMs-Blue-GeneL/2009-1010_3-5439494.html)

When modeling the deformable objects, there are many aspects needed to be considered such as the properties of the material, external forces applied on the deformable object and the environmental constraint (Haslach & Armstrong, 2004). Spline is one of the easiest ways to create deformable object. However, when the control points increases, the modification for the surface of the object will require heavy work as a simple change will affect many control points.

Modeling deformable object using mass-spring models can be constructed easily and fast but there are some limitations such as the physics applying on the model is hard to derive from the material properties since the model do not has the knowledge on controlling the nature of the objects. For example bending thin layer surface which can withstand the deformation. This problem can be solved by adding extra spring on the model but the cost of the computing this model will increase. The stability of the mass spring model is highly affected by the stiffness especially with large spring constant where the large spring constant is use to model rigid object. Time step is adjusted to be inversely proportional to the stiffness but this may lead to a slow simulation. Therefore, gravity force often been lowered to prevent large spring constant but this will lead to unnatural state for the deformable objects.

1.3 Problem Statement

Modeling deformable object require additional steps in comparing with the rigid objects as the nature of the deformable objects is hard to predict. Simple modeling techniques such as Free-form deformation can be used to construct a simple model but not complex model as changing one surface of the object may require adjustment to many controlling point. Developing a deformable collision detection system using BVHs requires a fast and efficient algorithm in order to make the simulation more accurate. Even though BVHs is a commonly used algorithm for detecting collision of the surfaces of the object, the volumetric collision detection of the objects using BVHs requires additional steps which will concern higher memory usage and hence affect the computational efficiency (Lau & Chan, 2002). Besides, the self collision on the object itself is needed in order to get a more realistic system. The response after the collision is also an important issue in order to make the whole system look more realistic.

1.4 Aim

The aim of this project is to develop a collision detection system for deformable object using bounding sphere hierarchy.

1.5 Objective

The objectives of this project are to:

1. Develop a simple 3 dimensional deformable object using simple mass spring model.
2. Develop a simulation system for collision detection between deformable objects using Bounding Sphere Hierarchy.

1.6 Scope

The scopes to be considered in this project are:

1. The input devices used to control the system are mouse and keyboard only
2. The deformable object used is cloth.
3. Number of particles of the cloth will be in the range of less than 30.
4. Simple responses after the collision will be made on the cloth.
5. No texture mapping will be applied on the model.
6. Only simple lighting techniques will be use in the system.
7. The collision detection simulation only occurs on the surface of the object.
8. The volumetric collision detection will not be computed due to the complexity.

1.7 Justification

The simulation of collision detection on deformable objects plays an important role in the application for computer graphics, surgery simulation and cloth simulation. This will enhance the field of computer graphics as many of the games and movies were produced using this technique in computer graphics. By using collision detection, the responses of deformable object will definitely increase the realism of the objects and its surrounding. The deformable object used in this project is cloth model and the modeling of cloth will be using the simple mass spring model.

Technique used to detect the collision for deformable object is bounding sphere hierarchy due to the simple construction and lesser storage is needed. The search tree to check the intersecting point will be the quad tree in order to speed up the process since the cloth is formed using particles connected through spring.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, collision detection and deformable objects are the main topic to be discussed. The discussion of collision detection will be the history, bounding volume hierarchy and bounding volume. The types of bounding volume are bounding sphere, axis-aligned bounding box (AABBs), discrete oriented polytopes (k-DOPs) and oriented bounding boxes (OBBs). Later in this chapter, modeling of deformable objects will also be discussed. Models of deformable objects which will be discuss in this chapter are splines and patches, free-form deformation (FFD), mass spring model and finite element models (FEMs)

2.2 Collision Detection

In real world, no two objects can share the same point in the space. In other words, it is impossible for two solid objects to penetrate which allows the motion of pressing and stacking to happen in reality. Besides, the impenetrability also applies on human's daily motion such as walking and standing. In the computer graphic world, many works are created based on the real time system and this impenetrability is one of the behaviors from real world that the simulation needs to be considered. Hence, collision is used to prevent the penetration between two objects in the

simulated environment that behave like the real world. It is the arrangement of two objects occupying the same points in the same space (Bergen, 2004).

The detection of collision between 3-dimentional object is the main concern in the field of computer graphic. Generally, the change in motion of the objects when the time passes is an interesting issue to in collision detection. These changes of the motion might only happen for one object and on the direction or shape of the object. Motion is a continuous flow of how the object arranges itself and it can be done by using the computer animation system by updating the arrangement of the object from time to time. Although collision might not happen within the period but the probability of this situation to happen is almost equal to zero as it is hard for two objects not to meet up within a certain space when the objects are in move (Bergen, 2004).

Collision detection can be conducted using proper mathematical logic and algorithms for different type of objects and their behaviors. This can be conducted easily but when it comes to a complex environment especially when there is a lot of moving objects and intersecting points, collision detection had become a more challenging task. Therefore, 2 fundamentals have been come out to solve this challenging task in order to meet up the real time necessity. These 2 fundamentals are Spatial coherence and Temporal coherence (Teschner et al., 2005).

Spatial coherence is when collisions are being ignored due to the small involvement areas of the objects in a wide space where the collision could barely involve. Spatial coherence is used to make sure that the number of colliding objects is much lesser than the actual number of the objects in order to decrease the work loads. While on the other hand, the temporal coherence is used to avoid the extra unused computation by recycling the data obtained from previous arrangement because sometimes the changes made only involved a small area when updates were made. Normally the motions are not affected (Bergen, 2004).

2.2.1 History

3D collision detection was first applied on robotics and automation. The simulation of the products was tested using computer to identify the disturbance and the checking of the disturbance was usually done continuously in four-dimensional space-time. This simulation can only applicable for certain objects and motions as the exact space-time of checking the disturbance is very challenging for interactive 3D applications even on a computer hardware. In the earlier years of computer graphics world, collision detections were invented to render a more life-liked 3D image (Teschner et al., 2005).

Moreover, geometric algorithms from a mathematical point of view has transformed into a new research area named "Computational Geometry". Computational geometry has created a lot of algorithms and data structure for problems such as convex hull computation, intersection for detection and computation, distance computation and linear programming. These new areas open up many innovative techniques to solve the collision detection problem.

When the first game device and home computer first showed up in 1980's, video games have thus gained their popularity and hence became the first 3D application. Elite, a space combat game by Ian Bell and David Braben in 1984 was the first interactive 3D video game on a home computer. The collision detection between the spaceship and the space station was determined based on simple primitives such as spheres and boxes even though the spaceship and shape station both have much more complex shape. This practice happens because the computer in the early days do not has sufficient processing power to carry out the real time collision detection. The interface of the Elite game is shown in Figure 2.1.

REFERENCES

Abdullah Bade, Saandilian Devadas, Daut Daman & Norhaida Mohd Suaib. 2005. Collision Response between Deformable Objects in Computer Games Environment. *University of Technology Malaysia*.

An Nguyen. 2006. Implicit Bounding Volumes and Bounding Volume Hierarchies.

Bergen, G.V. 1998. Efficient Collision Detection of Complex Deformable Models using AABB trees.

Bergen, G.V. 2004. Collision Detection in Interactive 3D Environment. Elsevier, Inc.

Brown, J., Sorkin, S., Bruyns, C., & Latombe J. C. 2001. Real-Time Simulation of Deformable Objects: Tools and Application.

Gibson, S.F., & Mirtich, B. 1997. A Survey of Deformable Modeling in Computer Graphic. *MERL* .

Gilberg R.C. & Forouzan B.A. 2005. *Data Structures A Pseudocode Approach with C.* Thomson Course Technology.

Gottschalk, S. 2000. Collision Queries using Oriented Bounding Boxes.

Hamzah Asyrani Sulaiman & Abdullah Bade. 2011. The Construction of Balanced Bounding-Volume Hierarchies using Spatial Object Median Splitting Method for Collision Detection. *International Journal on New Computer Architecture and Their Application.*

Haslach, Jr.H.W. & Armstrong, R.W. 2004. Deformable Bodies and Their Material Behavior. John Wiley & Sons, Inc.

Heidelberger, B., Teschner, M., & Gross, M. 2003. Volumetric Collision Detection for Deformable Objects. *Computer Graphics Laboratory ETH Zurich.*

Jiménez, P., Thomas, F. & Torras, C. 2001. 3D Collision Detection: A Survey. *Computers & Graphics 25*: 269-285.

Larsson, T., & Akenine Moller, T. 2005. A Dynamic Bounding Volume Hierarchy for Generalized Collision Detection.

Lau, R.W.H. & Chan, O. 2002. A Collision Detection Framework for Deformable Objects. *ACM VRST' 02*.

Lucchesi, B.J. 2002. A Parallel Liner Octree Collision Detection Algorithm.

Provot, X. 1996. Deformation Constraint in a Mass-Spring Model to Describe Rigid Cloth Behavior.

Rajiv, P. 2011. Cloth Simulation using mass-spring technique. *NCCA CGIT.*

Robotics, A. 2010. RoboWare Option.

Teschner, M.E., Kimmerle, S., Heidelberger, B., Zachmann, G. Raghupathi, L., Fuhrmann, A., Cani, M.-P., Faure, F., Magnenat-Thalmann, N., Strasser, W. & Volino, P. 2005. Collision Detection for Deformable Objects.