TJHSST Computer Systems Lab Senior Research Project Paper Three Dimensional Collision Detection for N Solids Using OpenGL 2007-2008

Richard Hooper

May 23, 2008

Abstract

Collision detection is a very useful concept, it is used in various applications from surgery to manufacturing to video game design. My project aims to create an efficient algorithm for detecting collisions so that it can be used in a gaming environment. The objects in collision will be simple solids, and multiple will be put in a space to monitor their interactions. The first step is simple 2D collisions followed by more complex 3D collisions.

Keywords: 3D, graphics, collision, collision detection

1 Introduction

The purpose of this project is to create an efficient algorithm for 3D collision detection. This project has value because there are many different applications for collision detection, and in game development, as with all other fields, efficiency is of extreme importance.

Collision detection is the concept of first detecting possible collisions, then contact, and then determining how to react to the collision. I intend to create an efficient algorithm that would detect collisions, so that the interactions of multiple solids could be modeled at once. The first step is to create a simple 2D algorithm that would model collisions as a prototype, followed by a simple 3D algorithm. This would then be optimized or redesigned, and then the number of solids in the given space would be increased, and the time taken and accuracy would be tested. The goal is to have the number of solids in space to be in the thousands, but the first benchmark would be in the hundreds.

2 Background

An important setting in which this would be used is in game development. In video games, it may be necessary for many objects to interact in space, and in video games, there can be no slowages as they are supposed to be a real-time simulation, and pauses for calculations cannot be accepted. Other applications include surgery, as simulations are used in the preparation, machining and animation.

Some alorithms used include raytracing, which creates vectors, or "rays" and uses them to detect possible collisions. Others include using bounding solids or using a simple point in polygon test, which is similar to raytracing. After conducting research, the system which would most apply to my particular project is a variation on bounding spheres where the number of points checked is minimized by having the the bounding solid constantly change its shape. This idea is called Non-static bounding solids.

3 Development

This project is an effort to create a fast and efficient collision detection algorithm. Success is considered a

working algorithm that can successfully detect collisions for one hundred solids (although one thousand would be preferable). Anything less would be considered a failure.

The language used is C using OpenGL, because C is a powerful language, and OpenGL is an easily accessible graphics library.

The workplan for the project is as follows: write a 2D algorithm, then write a 3D algorithm, then optimize the 3D algorithm or rewrite it to meet my time constraints.

It has also become clear that the display previously used to show the simulation was inadiquate, and a new one was created. This new interface features a box which clearly marks the boundaries of the 3D area in which the solids interact. You can also rotate the box so that you can get a better view of certain areas of interest, and if a certain event needs to be reexamined, the program can also pause, rewind, and fast forward the simulation.

So far the 2D and 3D algorithms have been completed, as has the interface, and the next stage is to optimize the 3D algorithm. Unfortunately the current algorithm is not very robust and only works with certain solids. The new algorithm has been started, and is in its early developmental stages..

4 Expected Results

I would expect the results to be a 100% success in respect to all collisions and contact made, and the speed in which the program executes should not experience any noticeable slowages. This means that in the end, there would be a large number of solids, hopefully at least one thousand would be able to interact seamlessly.

5 Conclusions

To come after project has been finished

6 References

1.The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis NC machining - T.D. Tang, Erik L.J. Bohez and Pisut Koomsap

2.Fast continuous collision detection among deformable models using graphics processors - Naga K. Govindaraju, Ilknur Kabul, Ming C. Lin and Dinesh Manocha

3.Deformable free-space tilings for kinetic collision detection - Pankaj K Agarwal, Julien Basch, Leonidas J Guibas, John Hershberger, Li Zhang

4.Fast Collision Detection based on nose augmentation virtual surgery - Kai Xie, Jie Yang, T.M. Zhu

5.Continuous Collision Detection for Articulated Models using Taylor Models and Temporal Culling - Xinyu Zhang, Stephane Redon, Minkyoung Lee, Young J. Kim

7 Appendixes