

Collision Detection in VR

Yi-Ning Chang & Ming-Jing Lin

NTUCSIE

Parallel and Distributed Processing Laboratory

February 22, 2017

Outline

- 1 Introduction
- 2 Work Items
- 3 Reference

1 Introduction

2 Work Items

3 Reference

Motivation

- Collision detection is one of the critical technique in improving the user experience in VR.
 - Collision between the users and the objects.
- An user should receive feedbacks while the avatar touches the objects or other users.
 - Be blocked, vibration or the vision feedback.

Current Status

- Online 3D games
 - Some MMORPGs don't handle the collisions between users.
 - Others wraps the characters in cubic or sphere bounds, which makes collision detection much easier.
- VR Application
 - An user interacts with objects using the controllers.
 - Real-time human-avatar interaction increases the complexity of multiplayer VR application.

Project Goal

- Evaluate the amount of resources required to perform collision detection between user avatars in a social VR application on cloud server.

1 Introduction

2 **Work Items**

3 Reference

Server

- Set up the server.
- Receive sensor data from clients.
- Perform collision detection.
- Monitor the usage of each resource.

Client

- Send data to server: position, orientation and body information.
 - Two or three real users demonstrate the collision effect by keyboard and screen.
 - Lots of artificial users to test the scalability.
 - Every user has artificial parts with random path, like arms or legs.
- Receive information from server.
 - The information of other users and objects.
 - Collision detection.
- Display

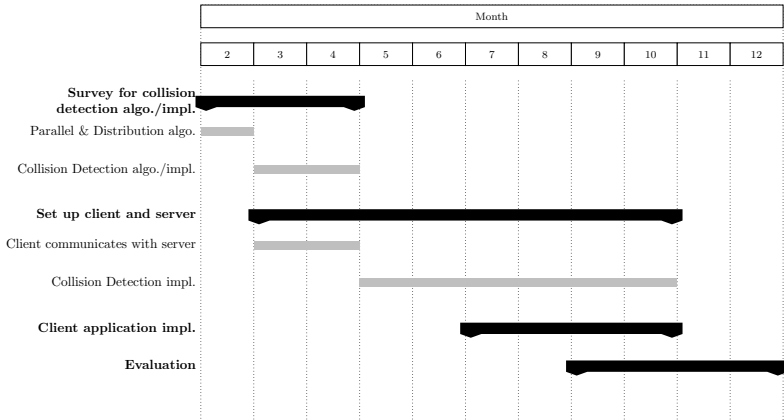
Evaluation

- System scalability
 - Increase the complexity of bounds on avatars and the number of users in limited resources.
 - Resource management.
- System stability
 - Correctness
 - Time consumed.

Expected Deliverables

- A better collision detection algorithm in parallel and distributed system.
- A resource management system
- A simple application with GUI by Unity 3D.
- Experimental data.
 - eg. resource usage

Gantt Chart



1 Introduction

2 Work Items

3 Reference

Paper

- Performance comparison between state-of-the-art point-cloud based collision detection approached on the CPU and GPU
 - from *IFAC-PapersOnline*
- Collision detection between point clouds using an efficient k-d tree implementation
 - from *Advanced Engineering Informatics*
- Real-time KD-Tree construction on graphics hardware
 - from *ACM Advanced Materials Research*
- Self-Customized BSP trees for collision detection
 - from *Computational Geometry*
- Unified GPU voxel collision detection for mobile manipulation planning
 - from *Intelligent Robots and Systems*

Code

- Algorithms in Game Engine Development
- HMD Initialization and Sensor Enumeration Documentation
 - from Oculus.com
- Collision Detection from *jeffThompson on GitHub*
- *JS Game Development - 3D AABB collisions on GitHub*