2.0 LITERATURE REVIEW

2.1 Introduction

The literature review below starts with the introductory knowledge in collision detection, typically involving 3D objects. Following this, the applications of Bounding Volume (BV) and its relationships to collision detection are presented. A brief definition of Bounding Volume Hierarchies is also touched on. Oriented Bounding Box (OBB), chosen as the BV used in this project, is re-explained and compared to other types of BV in terms of performance and implementation feasibility.

2.2 Collision Detection

In real environment, collision detection may not seem to be important or emphasized. Human beings has multiple sensory feedback from various body parts, such as hands and legs, to tell us when not to knock into walls and how to hold a cup of coffee properly. For non-living things, such as chair and floor, physics factor like gravity governed the behavior of movement so that the chairs will not simply fall through the floors. However, in the virtual or simulated environment, such as computer games or physics simulation, collision detection plays an integral part of the environment itself to offer realism similar to the real environment’s scenarios.

Hence, the definition of collision detection is presented here as simply if two or more objects are intersecting (Ericson, 2005). Another definition is to determine whether any points of the two objects occupy the same location in space simultaneously over a fixed time interval (Hadap, Eberle, Volino, Lin, Redon, Ericson, 2004). In more details, collision detection deals with three types of queries: if, when, where. “If” includes Boolean operation that return the result based on simply whether the objects intersect or not. “When” additionally determine the time of the collision occurrence. “Where” determines the exact intersection point or location of a collision. These three queries become increasingly difficult to answer in the previous order (Hadap et al., 2004).

There are few aliases to the term “collision detection". They may include "collision determination", “intersection detection” and “interference detection”.

Collision detection plays a fundamental role in commonly used applications today, such as computer games, computer animation, robotic navigational system and virtual reality simulations. Accurate collision detection is required for two purposes: raise the degree of realism of virtual simulation system and strengthen the immersive feeling of virtual interaction environment. Using an example of a computer game, collision detection maintains the illusion
of a solid world. Players are kept from running through walls, falling through floors or walking normally on water surface.

However, in CAD applications or computer animations, collision detection has been regarded as the more expensive component of the system and viewed as an advanced feature. To provide collision detection while producing interactive rates for some applications consume a lot of processing resources, sometimes even more than other components of the system itself. Thus, several approaches has been proposed and implemented widely to increase the efficiency of collision detection algorithms. One of them is the use of bounding volume, which will be discussed in the following sections.

2.3 Bounding Volume

In a virtual environment, testing of all object pairs for collision is simply not feasible in terms of hardware requirements and sometimes can be a waste of resources and time. In most situations, each object pairs has a low probability of collision. Only object pairs in close proximity of each other may be possible to collide. In most situations, this does not happen as often.

The concept of bounding volume has been introduced to alleviate this problem. A Bounding Volume (BV) is defined as “a single simple volume encapsulating one or more objects of more complex nature” (Ericson, 2005). BV allows for less expensive and computational operations on simpler volumes (BV) instead of on the complex objects that they bound. BV also facilitates fast overlap rejection tests for objects that are far from each other. In other words, if two BVs do not intersect, it is known that the objects that they encapsulate do not collide. Thus, one only needs to test against the complex objects when the overlap tests for their encapsulating BV are positive.

Bounding Volume has different forms and shapes depending on the algorithm and requirements of the system constructed. Typically, BV should have the following characteristics (Ericson, 2005):

- Inexpensive intersection tests
- Tight fitting
- Inexpensive to compute
- Easy to rotate and transform
- Use little memory

The characteristics above should be self-descriptive. Based on these characteristics, there has been several type of BV - Figure in (Ericson, 2005):
A bounding **Sphere** is just a simple sphere enclosing the object. It is the most memory-efficient of all BV and thus one of the easiest BN to test intersections on. The drawback may be in that it does not offer a very tight-fitting approximation of volume of the original object.

**Axis-Aligned Bounding Box** (AABB) is a rectangle bounding the object. It uses little memory but it has a seemingly large disadvantage when the object is rotated where approximation of volume differs greatly.

**Oriented Bounding Box** (OBB) is similar to AABB but with an arbitrary orientation. OBB is created to solve the rotation scenario problem.

**Discrete Oriented Polytope** (k-DOP) is a general form of AABB. However, it consists of a variable number of suitably oriented planes at infinity that collide with the furthest protruding points of the object. It is more tight-fitting as compared to AABB or sphere but requires more memory.

**Convex Hull** is the most accurate and tight-fitting method to represent an object since it is the smallest convex volume containing the object.

### 2.4 Oriented Bounding Box

To recap, an OBB is just an AABB with a variable orientation. By virtue of this difference, OBB can enclose an object more tightly than AABB or sphere and offer a more accurate approximation to the volume of the object. This advantage is illustrated below (Hadap et al., 2004):
Notice that in the above figure, the human in stick form is slightly oriented. More empty corners are found in AABB as compared to OBB.

OBB has a few representations (Ericson, 2005):

- A collection of eight vertices
- A collection of six planes
- A collection of three slabs (a pair of parallel planes)
- A corner vertex plus an orientation matrix and three half-edge lengths
- A center point plus an orientation matrix and three half-edge lengths

The last representation is the most commonly used, since the inclusion of an orientation matrix results in less expensive intersection testing.

In situation of collision detection which uses OBB as the BV type, intersection testing is more complicated, although the end-result of the scenario shows a higher realism. This is due to OBB representing the object more precisely.

2.5 Conclusion

Collision detection performance can be improved through the use of Bounding Volume. OBB is more accurate but less memory-efficient compared to other type of BV. The choice of using OBB depends on the requirement of the virtual applications.