REVIEW ON NARROW PHASE COLLISION DETECTION SYSTEM IN VIRTUAL ENVIRONMENT

Hamzah Asyrani Bin Sulaiman¹, Mohd Azlishah Othman¹, Mohamad Zoinol Abidin Abd. Aziz¹ and Abdullah Bade²

¹Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Melaka, Malaysia
²Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia
E-Mail: h.a.sulaiman@ieee.org

ABSTRACT
Collision detection is one of the most important tools that has been used widely in animation and simulation especially in computer games and medical simulation purpose. One of the critical issues in collision detection system is to actually prepared what kind of technique that is suitable for virtual environment to adapt collision detection system installed. Hence, this research paper described a detailed review on collision detection algorithm mainly in narrow phase collision detection phase.

Keywords: collision detection, virtual environment, computer games, simulation, animation.

INTRODUCTION
At initial level of collision detection, there are two subparts of dynamic collision detection techniques which are discrete collision detection (DCD) and continuous collision detection (CCD). Discrete collision detection only detects collisions at discrete moments in time (Dongliang and Yuen, 2000; Gottschalk, 2000; Bergen, 2004; Atencio, Esperanca et al. 2005; Kamat and Martinez, 2007; Tuft, 2007; Chang, Wang et al. 2008; Suaib, Bade et al. 2008; Chang, Wang et al. 2009; Kaiqiang and Jiewu, 2010; Meiping, Liyun et al. 2010; Ningjun, Kai et al. 2011; Razzaq, Shah et al. 2011; Chen, Ye et al. 2012; Jia, Chitta et al. 2012; Qu and Zhao, 2012; Wei and Jing, 2012; Xiyuan and Sourina, 2012; Taib, Othman et al. 2013; Xinyu and Kim, 2014). As a result, it may miss the collision between two successive configurations, also known by researchers as tunneling problems (Zhang and Du, 2011; Xinyu and Kim, 2012; Zhang, Kim et al. 2014).

Meanwhile continuous collision detection (CCD) techniques resolve the tunneling problem by first applying the interpolated motion to check for continuous movement for collisions, motion interpolation between successive configurations and used the interpolated motion to check for continuous movement collisions (Redon, Kheddar et al. 2002; Redon 2004; Redon, Kim et al. 2004; Zhang, Lee et al. 2006; Min, Sean et al. 2008; Tang, Curtis et al. 2008; Jing, Ying et al. 2009; Grady 2010; Sulaiman, Bade et al. 2010; Zhang and Du, 2011; Tang, Manocha et al. 2012). Continuous collision detection performs a higher-order check to find out if there is any intersection between two objects through the distance defined by configuration points (Redon, 2004; Zhang, Lee et al. 2006; He, Wang et al. 2009; Zhang, Kim et al. 2014). It is occurred when the object has thin surfaces or fast affecting objects that eventually missed the collision. It is important for hide-fidelity simulation even though it does not require for fast response simulation such as in computer games or animation. Thus, CCD hardly missed any potential collision between those configurations.

As shown in Figure-1, the pipeline shows two general phases of collision detection pipeline which are broad phase collision detection (starting from object handler until interesting pairs block) and narrow phase collision detection (starting from grid until colliding pairs block). Generally, the main challenges of narrow phase collision detection stage when precision becomes the important element to determine the colliding pairs.
Table-1. Techniques for distance computation.

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Author self claimed limitation/disadvantage</th>
</tr>
</thead>
</table>
| (Lin and Canny, 1991) | A Fast Algorithm for Incremental Distance Calculation | • Theoretical Implementation  
• Slow for complex object/large number of triangles |
| (Mirtich, 1998) | V-Clip: fast and robust polyhedra collision detection | • Advanced version of Lin-Canny algorithm  
• Depend heavily on complex calculation  
• Need to calculate for all vertices, edges and faces.  
• Sacrifices the face–face and edge–face cases in order to achieve better robustness |
| (Kawachi and Suzuki, 2000) | Distance computation between non-convex polyhedra at short range based on discrete Voronoi regions | • Pre-defined values of distances between near colliding objects  
• Large number of intersection tests and closest feature tests between triangles and voxels |
| (Tang, Lee et al. 2009) | Interactive Hausdorff distance computation for general polygonal models | • The technique itself is not an exact distance computation  
• User-specified accuracy with pre-defined configuration of objects. |
| (Jia, Sucan et al. 2013) | Real-time collision detection and distance computation on point cloud sensor data | • Use for broad phase type computations  
• Query more expensive due to usage of octree for the point cloud |

Meanwhile, point of contact technique for narrow phase collision detection was first introduced by (Gilbert, Johnson et al. 1988). The technique used simplex based implementation based on GJK but failed to satisfy the termination criteria (Gilbert and Foo 1990) that had been used to calculate the possible nearest points. Lin Canny technique was introduced then by implementing reducing resolution technique for detecting point of contact (Lin and Canny 1991). However, the technique missed certain cases where it may fall in infinite loop for complex polygons (Cameron, 1997; Chong-Jin and Gilbert, 2001; Sundaraj and Retnasamy, 2007; Shen and Li, 2010). Mirtich (1998) in (Chakraborty, Jufeng et al. 2008) proposed a V-Clip that provided solution for the Lin-Canny but not suitable when the coherence level is high. Later Ma et al (2012) proposed a distance computation technique for canal surfaces to detect point of contact but dedicated only for canal surfaces and involved complex calculations with only a few iterations and not suitable for massive models (Je, Tang et al. 2012; Yu and Yamane, 2013; Wang, Shi et al. 2014; Zhang, Kim et al., 2014). Trajectory based technique was implemented for detecting point of contact has been proposed in Schulman but according to the author itself, it involves expensive iteration and the trajectory could be missed sometimes (Schulman, Lee et al. 2013). Table-2 shows a summarized version of point of contact techniques.
Table-2. Techniques for point of contact.

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Author self claimed limitation/disadvantage</th>
</tr>
</thead>
</table>
| (Gilbert, Johnson et al. 1988) | A fast procedure for computing the distance between complex objects in three-dimensional space | • Simplex based technique that the most complex implementation required  
• Enhanced GJK exploit the simplex system but also failed to satisfy the termination criteria. |
| (Lin and Canny, 1991) | A Fast Algorithm for Incremental Distance Calculation | • Pre-defined polygons (18 pairs) using reducing resolution and need to undergo the technique more than once (up to infinity!) |
| (Mirtich, 1998) | V-Clip: fast and robust polyhedra collision detection | • Not suitable when the coherence level is high (not suitable for low count of polygon due to need to undergo various steps of technique) |
| (Chakraborty, Jufeng et al. 2008) | Proximity Queries Between Convex Objects: An Interior Point Approach for Implicit Surfaces | • Depend heavily on the complex algebraic equation for computing point of contact  
• Performance degradation when the object is nearly close for intersection |
| (Ma, Tu et al. 2012) | Distance computation for canal surfaces using cone-sphere bounding volumes | • Based on few restrictions and assumptions for canal surfaces only  
• Complex calculations involved |
| (Schulman, Lee et al. 2013) | Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization | • Trajectory point of contact determination for broad phase collision detection  
• Expensive iteration involves with complex techniques |

Considering the last component of narrow phase collision detection, Gilbert (Gilbert, Johnson et al. 1988) proposed a fast procedure to handle penetration of narrow phase collision detection but sometimes the technique reports false positive result for potential primitives (Cameron, 1997; Ponamgi, Manocha et al. 1997). Lin-Canny technique does not offer penetrating handling technique well but with a simple technique to cover penetration depth where it involves calculating the total distance travel from one point into another point that penetrate the other object (Bergen, 2001; Kim, Otaduy et al. 2003; Stephane and Lin, 2006). The technique itself could be a repeated cycle forever if penetration happens (Kim, Otaduy et al. 2003; Stephane and Lin, 2006; Zhang, Kim et al. 2007). Mirtich (Mirtich, 1998), developed a V-Clip complex technique that is capable to handle penetration but it is inefficient when the object completely penetrates into other object (the object completely inside the other object) (Escande, Miossec et al. 2007). Zhang et al., (Zhang, Kim et al. 2007) provided a generalized penetration depth computation which is able to compute object penetration depth faster than other previous method mentioned in their research paper but depends heavily on pre-processing configuration with a lot of predefined (Shengzheng and Jie, 2009; Choi, Ryu et al. 2010; Zhang and Du, 2011; Zhi, Shiqiu et al. 2011; Heng, KunChao et al. 2012). Later, Hausdorff distance in (Tang, Lee et al. 2009; Bartoň, Hanniel et al. 2010) was used to calculate the penetration depth for polygon but depends on closed models (where there is no hole for the object) and well-defined inside and outside vertices points (Je, Tang et al. 2012; Jia, Chitta et al. 2012; Zhang, Kim et al. 2014). Table-3 summarized all our findings for penetration depth issues.
Table 3. Techniques for penetration depth.

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Author self claimed limitation/disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gilbert, Johnson et al. 1988)</td>
<td>A fast procedure for computing the distance between complex objects in three-dimensional space</td>
<td>• Failed sometimes when reporting penetration depth (false result)</td>
</tr>
<tr>
<td>(Lin and Canny, 1991)</td>
<td>A Fast Algorithm for Incremental Distance Calculation</td>
<td>• Cannot handle penetrating polyhedra (object must not penetrate) and the technique could cycles forever • Degeneracy problems with false result</td>
</tr>
<tr>
<td>(Mirtich, 1998)</td>
<td>V-Clip: fast and robust polyhedra collision detection</td>
<td>• Simplex based with multiple iterations of complex technique • Inefficient if the object completely penetrate into the other object.</td>
</tr>
<tr>
<td>(Zhang, Kim et al. 2007)</td>
<td>Generalized penetration depth computation</td>
<td>• Involved lot of pre-processing step and limited to convex object and depend heavily on the pre-bounding convex covering</td>
</tr>
<tr>
<td>(Tang, Lee et al. 2009)</td>
<td>Interactive Hausdorff distance computation for general polygonal models</td>
<td>• It is not an exact technique and the calculation for penetration depth only covered for closed models (no hole) and well defined inside and outside vertices.</td>
</tr>
<tr>
<td>(Weller and Zachmann, 2011)</td>
<td>Inner Sphere Trees and Their Application to Collision Detection</td>
<td>• Generalized technique that use spatial data structures technique for only limited polygon types.</td>
</tr>
</tbody>
</table>

Based on the discussion above, it is suggested that the previous techniques on narrow phase collision detection are experiencing at least one of those common issues such as:


Considering these issues, there is a need to generalize the technique for narrow phase collision detection scheme that covers all three components. Instead of manipulating limited type of polygon, the proposed technique should indulge a better solution for any type of object that consist of vertices, edges and surfaces. The generalized technique should be able to handle distance computation, point of contact, and penetration depth with a reasonable speed and nearly accurate computation.

**DISCRETE COLLISION DETECTION**

Discrete collision detection (DCD) works has been studied a decade ago and the pioneers of this work are (Goldsmith and Salmon, 1987; Gilbert, Johnson et al. 1988; James, 1988; Baraff, 1989; Omohundro, 1989; Gilbert and Foo, 1990; Lin and Canny, 1991) where one of the remarkable researches conducted by (Gilbert, Johnson et al. 1988). The author created an efficient technique for computing the Euclidean distance between triangles using mathematical programming method. It used the position and orientation of the object and detect any potential collision along a continuous path (refer to Figure-1). An improvement has been made later on by using a simple...
extension by replacing it with a general convex set that could reduce the error caused by an approximation of polytope itself. The result is an efficient technique that has been introduced (Gilbert and Foo 1990). Then, an efficient technique for DCD by using hierarchical representation of spheres (Quinlan 1994) is introduced that speed up the process of collision detection and one of the research that pioneering the broad phase collision detection scheme instead of single scheme of collision detection. By enclosing the object with spheres for multiple levels, it helps to reduce the complexity by huge factors and also reduced computational cost.

Hubbard introduced a sphere tree construction by using oct-tree based type based on previous research by Quinlan (Hubbard 1996). By using space-time bound technique, Hubbard technique allowed the collision detection technique to work fast by approximation and graceful deprivation on the objects. In his dissertation, he concluded that the implementation of the technique works faster than the previous one.


CONCLUSIONS

This paper describes our review extensively on narrow phase collision detection and how the generalize version of narrow phase collision detection is required for virtual environment and graphics simulation.

ACKNOWLEDGEMENTS

The authors would like to thank Universiti Teknikal Malaysia Melaka and Ministry of Higher Education Malaysia for the financial support given through the research grant FRGS(RA CE)/2013/FKEKK/4F00197 (A Novel Interference Detection Algorithm Using Planar Subdivision of Axis-Aligned Bounding-Box for Virtual Reality Application).

REFERENCES


