Chapter 3
Collision Detection: A Fundamental Technology for Virtual Prototyping

Gabriel Zachmann
Clausthal University, Germany

ABSTRACT

Collision detection is one of the enabling technologies in many areas, such as virtual assembly simulation, physically-based simulation, serious games, and virtual-reality based medical training. This chapter will provide a number of techniques and algorithms that provide efficient, real-time collision detection for virtual objects. They are applicable to various kinds of objects and are easy to implement.

INTRODUCTION

In the product development process of manufacturing industries, prototyping is an essential step. Prototypes represent important features of a product that are to be investigated, evaluated, and improved. They are used to analyse design alternatives, to plan assembly lines, to support management decisions, or to market studies by showing a new model to potential customers.

The vision of virtual prototyping is to utilize virtual reality techniques for design evaluations and presentations based on a digital model instead of physical prototypes.

One of the most challenging classes of virtual prototyping in manufacturing industries are functional simulations virtual assembly simulations, and ergonomic studies. These are VR applications that usually comprise a lot of interaction among objects and interaction between the user and virtual objects.

In order to make the virtual objects behave exactly like real objects, fast and exact collision detection of polygonal objects undergoing rigid motions or deformations is an enabling technology in many virtual prototyping applications (and many other applications of computer graphics). It is a fundamental problem of the dynamic simulation of rigid bodies, simulation of natural interaction with objects, and haptic rendering.
For example, in virtual assembly simulations, parts should be rigid and slide along each other.

It is very important for a VR system to be able to do all simulations and renderings at interactive frame rates. Otherwise, the feeling of immersion and even the usability of the VR system will be impaired. For instance, haptic rendering is very demanding because it requires the collision detection algorithms to handle at least 1000 collision queries per second for each pair of objects.

Virtually all approaches to this problem utilize some kind of acceleration data structure. One particular requirement in the context of product engineering and virtual prototyping is that this auxiliary data structure is to be built fairly fast and efficiently. This is important because manufacturing industries do not want to store any auxiliary data in their product data management systems.

In this chapter, we will first describe some of the approaches to collision detection that we have developed in recent years, targeted at differing scenarios and conditions. Then, we will highlight some scenarios in virtual prototyping where collision detection is an enabling technology.

**BACKGROUND**

**BVH Based Methods**

Bounding volume hierarchies have proven to be a very efficient data structure for rigid collision detection, and, to some extent, even for deformable objects.

One of the design choices with BV trees is the type of BV. In the past, a wealth of BV types has been explored, such as spheres (Hubbard, 1996; Palmer & Grimsdale, 1995), OBBs (Gottschalk, Lin & Manocha, 1996), DOPs (Klosowski, Held, Mitchell, Sowriraj & Zikan, 1998; Zachmann, 2009), Boxtrees (Bala, Walter & Greenberg, 2003; Zachmann, 2002), AABBs (van den Bergen, 1997; Larsen, Gottschalk, Lin & Monocha, 1999), and convex hulls (Ehmann & Lin, 2001).

**Space Subdivision Methods**

Another alternative are space-subdivision approaches, for instance by an octree (Kitamura, Smith, Takemura & Kishino, 1998) or a voxel grid (Mcneely, Puterbaugh & Troy, 1999). In general, non-hierarchical data structures seem to be more promising for collision detection of deformable objects (Agarwal, Basch, Guibas, Hershberger & Zhang, 2000; Fisher & Lin, 2001; Huh, Metaxas & Badler, 2001), although some geometric data structures suggest a natural BV hierarchy (Lau, Chan, Luk & Li, 2002). Deformable collision detection is not the focus of our work presented here.

**GPU Based Methods**

A clever way to utilize graphics hardware was presented by Knott & Pai (2003). Based on the observation that an intersection can occur if and only if an edge of one object intersects the other one, they render edges of one object and polygons of the other. This even works for deformable geometry. Unlike many previous approaches, objects do not need to be convex. However, they must still be closed. Furthermore, it seems to work robustly only for moderate polygon counts.

A hybrid approach was proposed by Govindaraju, Redon, Lin & Manocha (1996). Here, the graphics hardware is used only to detect potentially colliding objects, while triangle-triangle intersections are performed in the CPU. While this approach alleviates previous restrictions on object topology, its effectiveness seems to degrade dramatically when the density of the environment increases.

The approach presented by Argawal, Krishnan, Mustafa & Venkataprasad (2003) can compute the penetration depth using graphics hardware, but only for convex objects.

Earlier image-based methods include (Baciu & Wong, 2002; Baciu & Wong, 2003; Lombardo, Cani & Neyret, 1999; Myszkowski, Okunev & Kunii, 1995; Shinya & Forgue, 1991).