Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Graphics 30 (2006) 432–438
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Interruptible collision detection for deformable objects

C. Mendoza[a,*], C. O'Sullivan[b]

[a] Virtual Reality and Modeling Group, Deptal. II, Desp. 147, Rey Juan Carlos University, c/Tulipan s/n, 28933 Mostoles (Madrid), Spain
[b] Image Synthesis Group, Computer Science Department, Trinity College Dublin, Dublin 2, Ireland

Abstract

This paper presents an approach to performing time-critical collision detection for deformable objects. The deformable objects are represented by dense meshes and their deformations are steered by a coarser mesh (reduced model) based on explicit finite elements, to achieve an interruptible algorithm, we use a sphere tree constructed using an adaptive medial-axis approximation of the dense mesh. The bounding spheres are updated using the coarse mesh, thus balancing computational accuracy and speed.

Keywords: Collision detection; Interruptible algorithms; Deformable objects

1. Introduction

Physically based modeling of deformable objects and collision detection have been extensively researched. Different approaches have been proposed to reproduce physically correct behaviors of deformable objects in real-time. Mostly, they are based on simplifications of more complex representations or on the use of fast and stable integration techniques on their governing equations. More recently, different levels of resolution of the physical model, *reduced models*, have been used to speed up the simulation.

In collision detection, most effort has been focused on solving the collision detection problem for rigid bodies. Most of the proposed techniques tackle the problem in two phases. The first phase, a *broad phase*, culls out pairs of objects that cannot possibly be interacting. The second phase, *narrow phase*, carries out more detailed intersection calculations. To accelerate the collision queries, this narrow phase most often uses pre-computed data structures that are hierarchical representations of the objects. To ensure real-times rates throughout the simulation, it is desirable to use a time-critical system which can interrupt the collision detection process to fit a time budget. Unfortunately, these techniques cannot be applied directly in cases where the objects are deformable, since the data structures need to be updated after every deformation. The update process is normally slow and constitutes a major bottleneck for real-time computations. Real-time collision detection for deformable objects is therefore a growing research area.

1.1. Contributions and outline

We present an approach to performing time-critical collision detection for deformable objects. To our knowledge, all previous collision detection algorithms that used an interruptible, or *just in time*, system were focused on rigid bodies and none of them on deformable objects. Our approach is based on [1] since we also use a reduced model to update the hierarchy tree. However they do not consider a fully interruptible system and some artifacts arise due to the use of a simple bounding volume generator.

The main idea behind our approach is to trade accuracy for speed in order to guarantee that collision processing is always performed in less than or equal to a stipulated *critical time*. First we create a deformable model based on two different resolution representations: The first resolution is a dense triangular mesh that we use for graphical rendering and the second is a coarser mesh (*reduced model*) used for deformations. The coarser mesh uses inner tetrahedrons so explicit finite elements can be used. The

*Corresponding author. Tel.: +34 91 488 8300; fax: +34 91 488 7049.
E-mail addresses: cesar.mendoza@urjc.es (C. Mendoza), carol.osullivan@cs.tcd.ie (C. O'Sullivan).

0097-8493/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2006.02.018

dense mesh follows the deformations of the coarse mesh by using *rigid links*. Next, we produce multiple approximations of the object's real surface using a sphere hierarchy. Our collision detection algorithm checks for intersections between such hierarchies until it exceeds the given critical time. The sphere hierarchies are not updated using the vertices of the surface of the object, but rather the vertices of the reduced model, thus reducing the time to update the hierarchies.

The rest of the paper is organized as follows. Section 2 gives an overview of some related works in deformable objects and collision detection. Then, in Section 3, we describe how we simulate deformations by using two different resolution meshes. In Section 4 we present our approach to handling collision detection between deformable objects using an interruptible mechanism, followed by experiments and results in Section 5. Finally, some conclusions and future plans are discussed in Section 6.

2. Related work

In general, there are two types of deformable objects in computer graphics: *geometrically based* and *physically based*. Geometrically based deformable objects change their shape by moving some control points or by calculating implicit functions (e.g. Free Form Deformation models [2]). Physically based models use physical laws and material properties to model the object. Among the best known are the mass-spring dampers [3], boundary element [4] and finite element [5,6] methods. The latter are the most accurate, but interactivity can be lost if the objects have a large number of primitives. To handle this, we can use multiresolution models (i.e., a set of reduced models) to achieve real-time simulations. Debunne et al. [7] use space and time adaptive sampling to achieve dynamic real-time deformations. Kondo and Kanai [8] simulated dense meshes using an underlying reduced physical model. A similar approach has been proposed in [9].

A significant amount of research has been focused in rigid body collision detection. Most of the resulting techniques need to be modified for deformable objects (please refer to [10,11] for recent surveys). The general process is divided in a broad phase to cull out non interacting pairs of bodies and in a narrow phase that traverses hierarchical representations of the objects to find intersecting regions. In general, these representations are made of bounding volumes such, as oriented bounding boxes (OBB) [12], *k*-dops [13], including their special case, the 6-dop axis-aligned bounding boxes (AABBs) [14], and sphere trees [15,16]. For rigid body time-critical collision detection, spheres have also been used to generate approximate responses to contacts [17]. Time-critical collision detection, also known as *graceful degradation*, was first proposed by Hubbard [18]. The objects are represented by sphere trees and collisions are tested in round-robin order, progressively increasing the level of accuracy until the interruptible mechanism stops the process after a given time. Later, this approach was extended to improve the mechanism for collision scheduling, contact modelling and collision response [17]. Klein and Zachmann [19] proposed an average case approach (ADB-trees) to abort the traversal of the hierarchy in a time-critical framework. They consider the probability that a pair of bounding volumes contains intersection volumes. To date, these approaches have only been used for collision detection between rigid bodies.

In collision detection for deformable objects, the hierarchies must be updated at each timestep that the object deforms. This update process can be very slow and thus the simulation may not meet real-time demands. Van den Bergen compared AABBs and OBBs [14] for deformable objects and determined that AABBs are the best option. He also showed that, although the hierarchies can also be rebuilt, updating is almost ten times faster than rebuilding. Larsson and Akenine-Möller [20] compared different methods for the hierarchy updating process based on bottom-up and top-down strategies. They found that these methods depend on the number of deep nodes processed. Based on this, they proposed an hybrid method that uses both strategies. Mezger et al. [21] speeded up the process by updating the hierarchy alter a few time steps and then only those branches whose primitives that have moved farther than a given distance. Recently, an approach to update the hierarchies by means of a reduced model has been proposed by James and Pai [1]. Alternatively, hardware accelerated collision detection methods have shown promising results [22,23]. However, accuracy is still limited due to the non-floating point precision and the size of the frame buffer memory. Other techniques based on spatial hashing have shown good results [24].

3. Deformable model approach

In this section, we present our dynamic model for the simulation of object deformations. It is based on two meshes: a dense mesh composed of a large number of triangles and used for graphical rendering. The vertices of this mesh are repositioned at each time-step to simulate complex deformations. These new positions of the vertices are steered by a coarser mesh, *reduced model*, see Fig. 1. This reduced model can be any physically based elastic model. In our case, we have implemented an explicit finite-element model as proposed by O'Brien and Hodgins [5] and used a fourth-order Runge-Kutta integration scheme.

3.1. Linking the coarse and dense meshes

The procedure to link the dense and the coarse meshes is done in an off-line process. Suppose that the dense mesh has $\mathbf{P} = (p_1, p_2, \ldots, p_N)^{\mathrm{T}}$ vertices on its surface. Next, consider only the vertices on the surface of the coarse tetrahedral mesh: $\mathbf{Q} = (q_1, q_2, \ldots, q_M)$. These are the reduced coordinates of the model. Note that $M < N$. Since