EUROGRAPHICS 2006 / E. Gröller and L. Szirmay-Kalos
(Guest Editors)

Volume 25 (2006), Number 3

GPU-based Collision Detection for Deformable Parameterized Surfaces

Alexander Greß, Michael Guthe, and Reinhard Klein

Institute of Computer Science II, Universität Bonn, Germany

Abstract

Based on the potential of current programmable GPUs, recently several approaches were developed that use the GPU to calculate deformations of surfaces like the folding of cloth or to convert higher level geometry to renderable primitives like NURBS or subdivision surfaces. These algorithms are realized as a per-frame operation and take advantage of the parallel processing power of the GPU. Unfortunately, an efficient accurate collision detection, that is necessary for the simulation itself or for the interaction with and editing of the objects, can currently not be integrated seamlessly into these GPU-based approaches without switching back to the CPU.
In this paper we describe a novel GPU-based collision detection method for deformable parameterized surfaces that can easily be combined with the aforementioned approaches. Representing the individual parameterized surfaces by stenciled geometry images allows to generate GPU-optimized bounding volume hierarchies in real-time that serve as a basis for an optimized GPU-based hierarchical collision detection algorithm. As a test case we applied our algorithm to the collision detection of deformable trimmed NURBS models, which is an important problem in industry. For the trimming and tessellation of the NURBS on the GPU we used a recent approach [GBK05] and combined it with our collision detection algorithm. This way we are able to render and check collisions for deformable models consisting of several thousands of trimmed NURBS patches in real-time.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; Splines; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1. Introduction

Efficient collision and self-collision detection of deformable 3D objects is required in numerous applications: in virtual prototyping it is used to determine clearances, in physically based simulations to ensure that the bodies do not penetrate each other and in computer games that colliding objects bounce and slide on contact instead of passing through each other. In these contexts a collision is defined as a configuration of two objects whose surfaces, that are allowed to deform and move over time, intersect. Since the deformation and movement is mostly simulated by discrete updates of the objects, standard collision detection algorithms are only interested in intersections at these discrete time stamps.

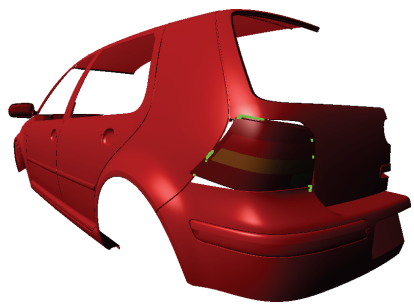

Figure 1: *Virtual assembly simulation (collisions highlighted in green).*

For the geometry of the objects different representations are used in these applications. While in most virtual reality applications meshes are still the predominant geometry representation for rendering and interaction, the use of higher order representations like NURBS or subdivision surfaces as rendering and interaction primitives has come into focus. The reason for this trend is that due to the rapidly growing parallel processing power that enables GPUs to per-