Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Local Constraint Methods for Deformable Objects

Marc Gissler, Markus Becker, Matthias Teschner

Computer Graphics, University of Freiburg, Germany

Abstract

We present an efficient scheme to enforce non-conflicting geometric constraints for deformable objects. Iterative solvers and stabilization techniques are avoided. The method is not subject to numerical drift and all constraints are accurately met at each simulation step. The approach does not require any pre-processing and dynamically changing constraints can be handled efficiently. Experiments indicate that thousands of constraints can be processed at interactive rates. Although our approach is restricted to non-conflicting constraints, experiments illustrate the versatility of the method in the context of deformable objects. Details can be found in [GBT06].

1. Introduction

Many existing constraint approaches are global methods. These techniques solve for all constraints simultaneously and minimization techniques are employed to handle conflicting constraints. In the context of rigid bodies, global methods are appropriate since conflicting constraints commonly occur due to the small number of degrees of freedom. However, solving the resulting linear systems can be expensive in terms of memory and computing costs.

In contrast to rigid bodies, deformable objects are characterized by a larger number of degrees of freedom. If a deformable object is represented as a mass-point system, more than one non-conflicting constraint can be defined per object by distributing the constraints to different mass points. This easily allows to avoid conflicting constraints without sacrificing the versatility of the applied constraints. E. g., closed loops can be realized by attaching adjacent models to different mass points of an object. Thus, a large class of conflicting rigid-body constraints can be replaced by non-conflicting constraints for deformable mass-point systems.

2. Results

A variety of test scenarios has been implemented to illustrate the efficiency and the versatility of the proposed constraint technique. We have therefore combined the constraint approach with a simple and efficient deformable modeling approach for tetrahedral meshes [THMG04] and with an efficient and robust collision handling scheme [THM*03, HTK*04]. In all experiments, the simulated tetrahedral mesh is geometrically coupled with a triangulated surface mesh. All tests have been performed on an Intel Pentium 4 PC, 3.4 GHz.

Fig. 1 illustrates the test scenario that we have used for performance measurements. The number of cubes and constraints can be varied and Fig. 2 shows the measurements for up to 9990 cubes with 49950 tetrahedrons and 39924 constraints.

Fig. 3 depicts an application for point-to-surface constraints. Fig. 4 illustrates a simulation sequence with dynamically changing constraints. Constraints can be activated or deactivated during a simulation. The set of constraints can be changed interactively. Again, these simulations can be computed and visualized at interactive rates.

References

[GBT06] GISSLER M., BECKER M., TESCHNER M.: *Local Constraint Methods for Deformable Objects*. Tech. rep., University of Freiburg, to appear, 2006.

[HTK*04] HEIDELBERGER B., TESCHNER M., KEISER R., MUELLER M., GROSS M.: Consistent penetration depth estimation for deformable collision response. In *Proc. of Vision, Modeling, Visualization VMV* (2004), pp. 339–346.

[THM*03] TESCHNER M., HEIDELBERGER B., MUELLER M., POMERANETS D., GROSS M.: Optimized spatial hashing for collision detection of deformable objects. In *Proc. of Vision, Modeling, Visualization VMV* (2003), pp. 47–54.

[THMG04] TESCHNER M., HEIDELBERGER B., MÜLLER M., GROSS M. H.: A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International* (2004), pp. 312–319.

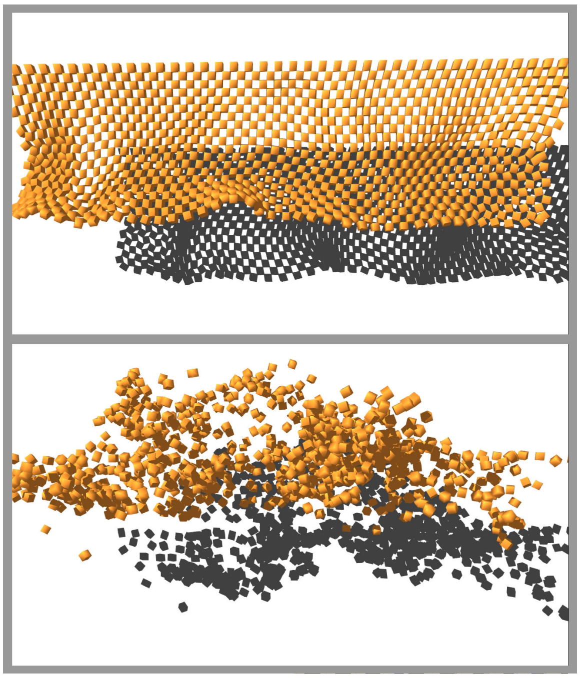

Figure 1: *Cubes with points-to-point and point-to-nail constraints. All constraints are active in the first image. In the second image, all constraints are deactivated. 3950 tetrahedrons, 160 point-to-nail and 2964 points-to-point constraints are used in this sequence. The forces are computed in* 3.42*ms, constraint forces take* 5.38*ms, and the numerical integration takes* 3.2*ms. So, one timestep can be simulated in* 12*ms. For the performance measurements shown in Fig. 2 the number of cubes and constraints is varied.*

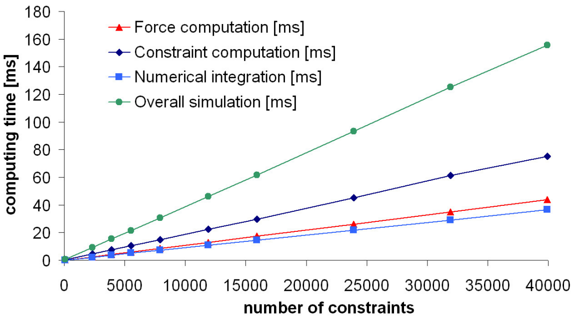

Figure 2: *This graph illustrates the performance measurements using the test scenario shown in Fig. 1. The number of objects and constraints is varied and measurements for up to 39924 constraints are shown which corresponds to 9990 cubes with 49950 tetrahedrons. Up to 25000 constraint forces are computed at interactive rates. Further, the entire dynamics of 2990 cubes with 14950 tetrahedrons and 11924 constraints is computed in* 46.15*ms.*

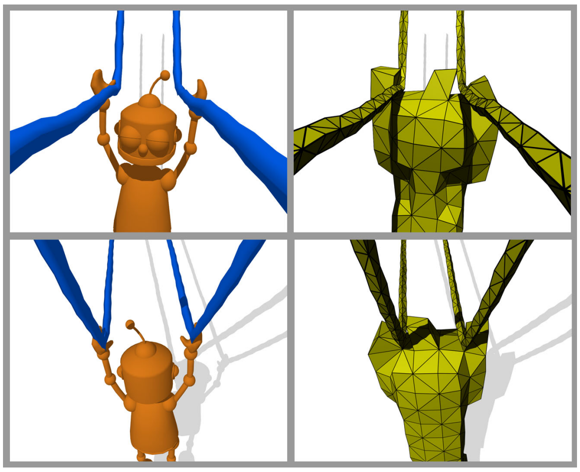

Figure 3: *Point-to-surface constraints are used to attached two points of the tetrahedral mesh of the robot to the rope models. This enables the robot to slide along the ropes. 1339 tetrahedrons, 14 points-to-point, and 2 point-to-surface constraints are used. Deformation forces are computed in* 0.81*ms, constraint forces take* 1.54*ms, and the numerical integration takes* 1.17*ms.*

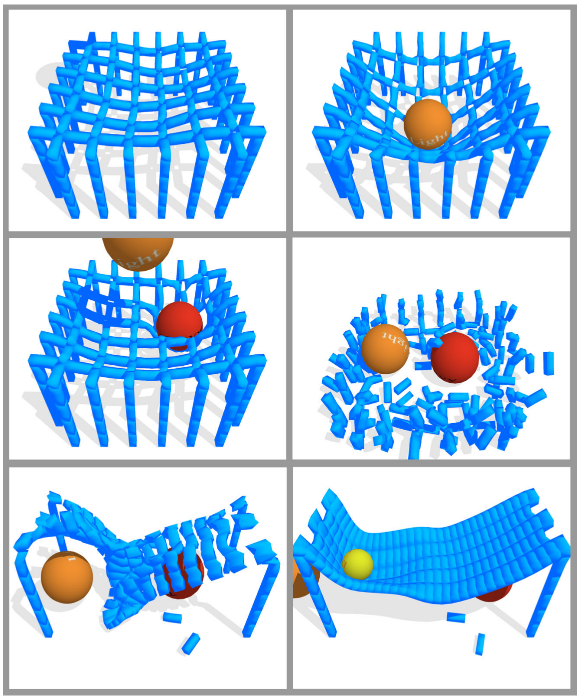

Figure 4: *Dynamically changing constraints can be handled. The constraints of the net are deactivated and new constraints for the bridge are activated. 2760 tetrahedrons and 1584 (net) or 1320 (bridge) constraints are used. The forces are computed in* 2.16*ms, constraint forces take* 1.6*ms (net) and* 2.38*ms (bridge), and the integration takes* 1.13*ms.*