Contact Surface Computation for Coarsely Sampled Deformable Objects

Jonas Spillmann Matthias Teschner

Computer Graphics Laboratory
University of Freiburg
Germany

Abstract

We introduce a new collision response scheme that handles collisions between deformable objects with triangulated surfaces. Based on internal forces and penetration depth information, the approach computes the contact surface of interpenetrating objects. In particular, we address discontinuity problems that arise in discrete-time simulations with coarse surface representations.

The approach is independent of the actual deformation model and the applied collision detection algorithm, and in contrast to penalty-based collision response schemes, the method does not require any user-defined parameters. We compare the contact surface method to a penalty-based collision response scheme to illustrate the conceptual advantages of the proposed technique.

1 Introduction

Handling collisions of rigid or deformable objects is a challenging problem in physically-based simulations. Collision handling involves two steps: First, a collision between bodies has to be detected. Second, the collided bodies have to be separated in order to provide a physically correct state. This separation results from the exchange of momentum in the contact region between the bodies. Thus, the resulting momentum of the bodies depends on the contact state and contact region, which implies that the contact region has to be determined very carefully. This problem is particularly challenging if the surfaces of simulated objects are coarsely sampled. In this case, small relative movements of colliding objects may result in rapid changes of the contact surface and, thus, discontinuous contact forces.

A common way to handle collisions between deformable objects is to apply penalty forces to the collided mass points. The penalty force depends on the penetration depth of the point, and it accelerates the point out of the volume of the collided object. The penetration vectors can be approximated such that magnitude and direction change smoothly, even for deep penetrations and coarsely sampled objects [10]. However, this method introduces physically inplausible states. In particular, resting contact of stacked objects is a problem. Further, the applied penalty forces do not correspond to the stress resulting from the object deformation. Instead, the penalty forces are weighted with a user-defined parameter. The choice of this scalar is difficult, especially when simulating bodies with different elasticities. Moreover, penetrations tend to be deep at high pressures and for large relative velocities.

Contribution. We present a method that overcomes problems inherent to penalty-based approaches. Instead of applying penalty forces to colliding points, we compute an explicit representation of the contact surface between flexible objects with triangulated surfaces. Thus, penetrations between simulated objects are avoided and a physically correct contact state is provided in every simulation pass.

In contrast to existing contact surface approaches, we address discontinuity problems due to discrete time steps and coarse surface representations. Further, our method handles objects with different surface sampling densities (see Fig. 1).

We compare our method to a penalty-based approach described in [10] in order to show the conceptual advantages. Our method is not based on user-defined parameters and the computation of the contact surface exclusively depends on the elasticities of the colliding bodies. Moreover, the method is independent of the deformable model. Although we rely on the internal forces acting on the surface points, we do not make any assumption on how

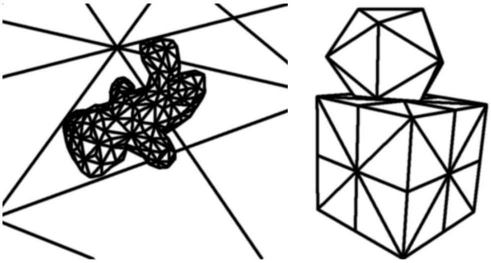

Figure 1: We address problems in contact surface computation related to coarse surface representations. Left: A collision between two objects with different sampling densities. Right: In this collision, the contact state consists of only two contact points. In both cases, continuous collision response is a challenging problem.

these forces are computed. The actual implementation uses a deformation model proposed in [19], but the method works with any finite-element or mass spring deformation model.

2 Related work

Collision response approaches have been investigated for rigid and deformable objects.

For **rigid bodies**, there exist three different collision response schemes. First, a constraint-based method can be used to prevent objects from penetration. In this case, the inequality-constrained problem can be formulated as a linear complementary problem (LCP) [1, 3, 4, 8]. These approaches work especially well for resting contacts of rigid objects. Second, the contact forces can be computed by applying the impulse laws [14]. The third kind computes penalty forces to penetrated points [15, 5]. Penalty-based collision response schemes are usually fast and easy to implement, however, penetrations can occur. Thus, they are less suitable for simulating stacked objects.

Contact problems for **deformable bodies** have first been addressed in engineering [11, 13]. These approaches focus on analytical and accurate solutions for simple geometries. However, deformable bodies are commonly discretized into mass points which motivates methods that consider discrete settings.

Two different schemes can be distinguished in this context: Penalty-based schemes and contact surface schemes. Penalty-based methods for deformable bodies are similar to penalty-based methods for rigid bodies: A force is exerted on a collided point, proportional to the penetration depth of the point. In [10], a way is proposed to approximate the penetration depth such that the resulting collision response scheme is particularly stable. In [16], Müller et al. propose a method that handles collisions between particles and deformable surfaces. This approach does not cause penetrations, but also exerts forces on points depending on the distance of the particle to the surface.

A combination of contact surface schemes and penalty-based schemes is presented in [12]. Here, a ’virtual’ contact surface is computed for colliding deformable objects. Then, penalty forces are applied to the points, proportional to the distance between the actual point and its computed position on the contact surface.

All these approaches require the choice of a user-defined parameter that scales the penalty force, while our approach exclusively depends on the elasticities of the objects.

An early work that treats the contact problem by computing the contact surface is presented in [9]. Here, a simple elasticity model based on a scalar potential field is used. However, the model is difficult to relate to the physics of elastic deformations. Baraff and Witkin [2] use a constraint-based method that prevents penetration. They solve an LCP that is expensive to compute for complex objects. Duriez et al. [6, 7] describe a method that is based on Signorini’s contact theory. Contact forces are computed for flexible bodies by formulating an LCP. However, they are not able to simulate locally different elasticities, whereas our approach enforces a force equilibrium in small areas around surface vertices and thus overcomes this limitation. In [17], an alternative LCP scheme for computing the contact surface is presented. Here, a combination of point-based rigid and deformable models is used. The approach works well for densely sampled surfaces and small penetrations. In contrast, our work addresses discontinuity problems when using coarsely sampled surfaces combined with deep penetrations. Further, in [17], volume preservation is explicitly enforced without considering the actual deformation model. In contrast, our approach processes the forces of the underlying deformation model. Thus, volume preservation is implicitly handled.

3 Contact surface

In general, there must be a force equilibrium on the contact surface of colliding objects. The internal force on an infinitesimal small surface patch at position $\mathbf{x}$ of body $\mathfrak{M}$ must be equal in magnitude to the internal force at $\mathbf{x}$ on body $\mathfrak{N}$. This condition also holds for discretized surfaces. However, in the discrete case, forces are only given at vertices and have to be interpolated over the surface. Thus, the *pressure* on surface patches is compared. With respect to accuracy, the considered surface patches should be as small as possible. However, our approach requires spatial degrees of freedom on both objects to ensure symmetry. Thus, the surface patches must be large enough to contain at least one vertex of the surfaces of both colliding objects.

4 Method

This section provides an overview of the proposed algorithm, followed by a detailed description of its five consecutive stages.

The method takes a set of pairs of colliding bodies and computes an explicit representation of the contact surfaces. After collision response, the collided vertices are displaced onto the contact surfaces and a collision-free setting is reached. The method passes through the following stages:

Stage 1 detects all *colliding vertices* based on a spatial hashing approach.

Stage 2 computes a *consistent penetration depth* and *direction* for each collided vertex.

Stage 3 takes the set of colliding vertices and computes the *deformation region* on the surface of each collided body. For all vertices in the deformation regions, a *displacement vector* is calculated.

Stage 4 performs a binary search to compute the contact surfaces by displacing the vertices in the deformation regions until a force equilibrium is reached.

Stage 5 updates the velocity and dissipative forces of each displaced vertex.

4.1 Collision detection

In our implementation, deformable bodies are represented by tetrahedral meshes, and a spatial hashing approach is used to detect vertices that penetrate tetrahedrons of other bodies. The method returns the set of colliding vertices and tetrahedrons. Details are found in [18, 20].

4.2 Consistent penetration depth

Based upon the set of colliding vertices, the second stage estimates the penetration depth and direction for collided vertices [10]. The method returns a consistent penetration depth for each collided vertex. The method considers the discrete nature of the simulated objects, i. e. for small displacements of the vertices, the penetration depth vectors change smoothly and the penetration directions are meaningful even for large penetrations.

Further, a *contact triangle* is calculated for each collided vertex. The contact triangle is the triangle on the surface of the penetrated object that intersects with the penetration direction vector. The contact triangle plays an important role when computing the contact surface, since each vertex must lie in the plane of its contact triangle after collision response.

4.3 Deformation region

A challenge of the collision response of triangulated meshes are the approximate surface representations: We consider collisions with only a few colliding vertices. In some cases, the collision can be asymmetric in a sense that the set of colliding vertices of two bodies contains only vertices of one body (see Fig. 2). In this case, it is not possible to reach a force equilibrium by only displacing the collided vertices.

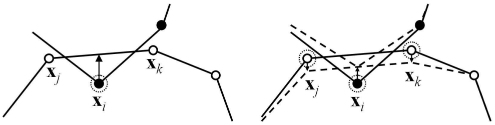

Figure 2: Deformation region. Left: If only the colliding vertex $\mathbf{x}_i$ is considered in the contact surface computation, then the lower body is not affected and a force equilibrium cannot be reached. Right: The deformation region, consisting of $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$, enables a symmetric reaction to the collision, a force equilibrium can be achieved.

This observation motivates that we do not only consider the colliding vertices, but define a *deformation region* (see Fig. 2). The deformation region is the union of the colliding vertices and the vertices of their contact triangles. Note, that the latter do not necessarily collide ($\mathbf{x}_j$ and $\mathbf{x}_k$ in Fig. 2). We denote the vertices contained in the deformation region as *displacement vertices*. For each displacement vertex a displacement vector is computed. When the displacement vertex is a colliding vertex, its displacement vector corresponds to the penetration depth. For a non-colliding vertex, the displacement vector represents a direction which is an additional degree of freedom in order to find the force equilibrium.

In order to guarantee continuous contact forces, the displacement vectors must change smoothly for small changes in the contact state. Therefore, we propose to calculate the displacement vector $\mathbf{s}$ by

$$\mathbf{s} = \frac{\sum_i w_i \mathbf{d}_i + \mathbf{d}}{\sum_i w_i + 1} \tag{1}$$

with $\mathbf{d}_i$ referring to the penetration depths of all vertices that have their contact triangle in the set of faces adjacent to the displacement vertex. If the displacement vertex is colliding, then its corresponding penetration depth $\mathbf{d}$ is added to the weightened sum of penetration depths. If the displacement vertex is not colliding, then $\mathbf{d}$ is $\mathbf{0}$.

In order to provide a continuous behavior of the displacement vector, the penetration depths $\mathbf{d}_i$ of the vertices are weighted with w_i where w_i is the barycentric weight of the vertex with respect to the displacement vertex (see Fig. 3).

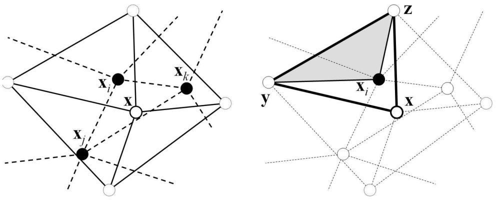

Figure 3: Left: The displacement vector for a vertex $\mathbf{x}$ is the weighted sum of the penetration depths of the vertices $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ having their contact triangles adjacent to $\mathbf{x}$. Right: The barycentric weight w_i of the vertex $\mathbf{x}_i$ with respect to $\mathbf{x}$ is $w_i = \frac{A(\mathbf{x}_i,\mathbf{y},\mathbf{z})}{A(\mathbf{x},\mathbf{y},\mathbf{z})}$. $A(\mathbf{x},\mathbf{y},\mathbf{z})$ is the area of the contact triangle of $\mathbf{x}_i$.

4.4 Contact surface computation

After having computed the deformation region and the displacement vectors of all affected vertices, the contact surface is computed. By displacing the vertices in the deformation region, a binary search for the position of the contact surface is realized. This process is governed by the constraint of force equilibrium at the contact surface.

4.4.1 Binary search

First, the vertices in the deformation region are displaced halfway in the direction of the displacement vector $\mathbf{s}$:

$$\mathbf{x}^1 \leftarrow \mathbf{x} + \frac{1}{2}\mathbf{s}$$

with $\mathbf{x}$ being the original displacement vertex and $\mathbf{s}$ the displacement vector of the vertex.

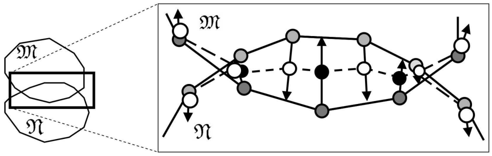

Figure 4: First iteration of the binary search: The vertices in the deformation regions are displaced halfway between the old position and the surface of the other body. The resulting contact surface is thus exactly in the middle of the intersection. This corresponds to the contact surface of two equally elastic bodies. Note, that the vertices on the very left and right are non-colliding displacement vertices, which are also displaced.

In each iteration, we divide the search interval by two:

$$\mathbf{x}^{i+1} \leftarrow \mathbf{x}^i \pm \frac{1}{2^{(i+2)}}\mathbf{s}$$

The direction of the movement is given by the sign of the pressure difference. The binary search converges very fast. Experiments indicate that four iterations provide sufficient accuracy.

4.4.2 Computing internal forces

As a result of the vertex displacement, the underlying deformed volume exerts a restoration force on the contact surface, i. e. the internal forces on the contact vertices have changed.

The internal forces are then updated by the deformation model. Although we use an approach presented in [19], we emphasize that the collision response method is independent from the actual deformation model.

4.4.3 Measuring and comparing forces

Based on the internal forces, the algorithm compares the pressure on the surface of $\mathfrak{M}$ and the pressure on the surface of $\mathfrak{N}$. In order to handle locally different elasticities, the method computes the pressure difference within a small comparison area A around each displacement vertex $\mathbf{x}$.

$$\mathbf{p}_{\text{diff}} = \mathbf{p}_{\mathfrak{M}}(\mathbf{x}) - \mathbf{p}_{\mathfrak{N}}(\mathbf{x}) = \frac{\mathbf{F}_{\mathfrak{M}}(\mathbf{x})}{A} - \frac{\mathbf{F}_{\mathfrak{N}}(\mathbf{x})}{A}$$

which is minimized by the binary search approach. Let $\mathbf{x}$ be in the deformation region of $\mathfrak{M}$. The challenge is to find *all* vertices within the comparison area A around $\mathbf{x}$ on the coarsely sampled surfaces of $\mathfrak{M}$ and $\mathfrak{N}$. We propose A to be the *union of surface triangles adjacent to* $\mathbf{x}$. Then the set of vertices inside A on the surface of $\mathfrak{M}$ consists of one element, namely $\mathbf{x}$. The set of vertices inside A on the surface of $\mathfrak{N}$ consists of all vertices having their contact triangle adjacent to $\mathbf{x}$ ($\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ in Fig. 3). The case of this set being empty is discussed later. The actual pressure is now computed by summing up the internal forces on these vertices, divided by the area A that is represented by these vertices.

For the pressure $\mathbf{p}_{\mathfrak{M}}(\mathbf{x})$ on the surface of body $\mathfrak{M}$ in $\mathbf{x}$ we have

$$\mathbf{p}_{\mathfrak{M}}(\mathbf{x}) = \frac{\mathbf{F}_{\mathfrak{M}}(\mathbf{x})}{A}$$

with $\mathbf{F}_{\mathfrak{M}}(\mathbf{x})$ the internal force on $\mathbf{x}$.

When computing the pressure $\mathbf{p}_{\mathfrak{N}}(\mathbf{x})$ acting on the surface of $\mathfrak{N}$ around $\mathbf{x}$, we must consider the discretized surface representation. In order to provide continuous pressure changes for small displacements of the vertices, the internal forces are weighted (see Fig. 5). Thus, the pressure $\mathbf{p}_{\mathfrak{N}}(\mathbf{x})$ acting on the surface of $\mathfrak{N}$ around $\mathbf{x}$ is

$$\mathbf{p}_{\mathfrak{N}}(\mathbf{x}) = \frac{\sum_i w_i \mathbf{F}_{\mathfrak{N}}(\mathbf{x}_i)}{A \sum_i w_i} \qquad (2)$$

with $\mathbf{F}_{\mathfrak{N}}(\mathbf{x}_i)$ the internal force on the vertex $\mathbf{x}_i$. w_i is the barycentric weight of the vertex $\mathbf{x}_i$ with respect to $\mathbf{x}$ (see Fig. 3).

Two notes: First, the pressure difference is actually calculated with scalar pressure values, namely the magnitude of the pressure vectors in direction of the displacement vector $\mathbf{s}$. Second, if the surface of body $\mathfrak{N}$ is much coarser sampled than the surface of $\mathfrak{M}$, then the set of vertices of $\mathfrak{N}$ within the comparison area can be empty. In this case, the area is extended and the vertices of the contact triangle of $\mathbf{x}$ also contribute to the pressure.

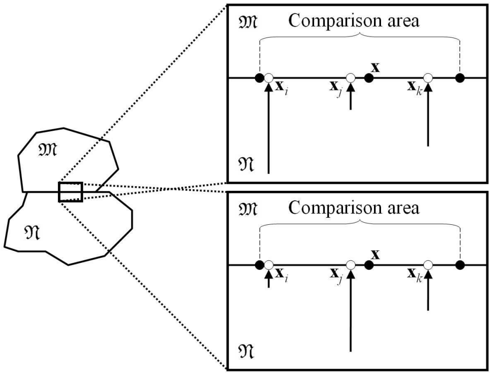

Figure 5: Weighting forces. Top: The forces are not weighted. If now vertex $\mathbf{x}_i$ leaves the comparison area, then the pressure around vertex $\mathbf{x}$ will change discontinuously. Bottom: The forces are weighted with the barycentric weight, thus the force on $\mathbf{x}_i$ has less influence than the force on $\mathbf{x}_j$ being closer to $\mathbf{x}$. Discontinuous pressure changes are avoided.

4.5 Updating velocity and dissipative forces

After having computed the contact surface, the velocity of the displacement vertices is updated. Further, the friction force that acts between vertices and contact triangles is calculated. We use the Coulomb friction law for modeling the friction.

Fig. 6 shows the final result of the collision response procedure. The initial configuration consists of two overlapping bodies. Then a binary search approach computes the contact surface, considering the force equilibrium.

4.6 Results

We have integrated our method in a simulation environment for deformable objects. We carried out various experiments to compare the proposed method to a penalty-based approach. We compare the quality of both methods with respect to visual plausibility and stability. All experiments have been per-

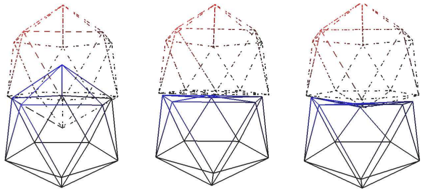

Figure 6: Illustration of the contact surface for two bodies with different elasticities. Left: The original configuration before contact surface computation. Middle: Contact surface for two bodies having the same elasticity. Right: Contact surface for two bodies with different elasticities. Note the different position of the contact surface.

formed on a PC Pentium 4, 3.20 GHz, NVIDIA Quadro FX 3400 GPU.

Fig. 7 shows three cubes at different weights falling onto a flexible object. The setting consists of 323 vertices and 624 surface triangles, and the average time for collision response is 2 ms. Note the penetration of the heavy cube when the collision response is realized with a penalty-based approach. With our approach, the cubes do not intersect with the membrane.

An experiment that compares the behavior of the two response schemes at high pressure is shown in Fig. 8. A soft sphere is squeezed between two stiff cubes. The experiment consists of 95 vertices and 166 surface triangles and the average time for collision response is 0.9 ms. Note, that there are only 17 colliding vertices, which emphasizes the approximative nature of the setting. While the penalty-based approach results in significant overlaps of the object, our approach gains a collision-free situation by computing the contact surfaces between the sphere and the two cubes.

We also performed an experiment to illustrate how our approach scales with respect to the scene complexity. Fig. 9 shows a frame of an experiment where fruits are filled in a dish. The setting consists of 16030 vertices and 27116 surface triangles. The average time for collision response is 72 ms. The average time for a simulation step, consisting of deformation, collision detection and response, is 150 ms. A collision of two objects typically involves as few as 4 to 10 colliding points, and even in this inauspicious configuration, the method still provides a smooth behavior.

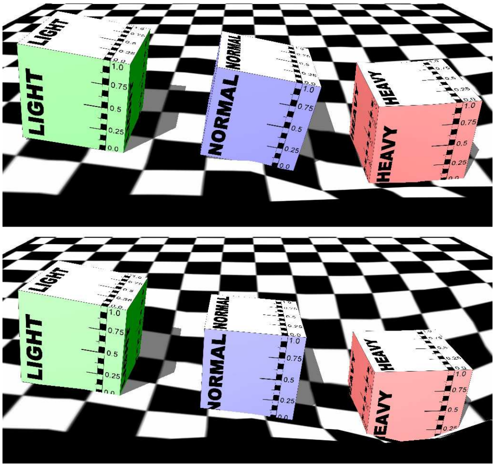

Figure 7: Three cubes at different weights falling onto a flexible object. Top: Collision response by computing the contact surfaces. Bottom: Collision response with a penalty-based approach. Note the significant penetration of the heavy cube.

A general drawback of penalty-based approaches is that stacked objects oscillate. A reason for this phenomenon is that the introduced penalty forces behave like springs. In order to illustrate this effect and compare it to the results of our approach, we simulated a stack of soft membranes (see Fig. 10). The scene consists of 2178 vertices and 4320 surface triangles. Further, we plotted the total kinetic energy of the setting. In Fig. 10, we see that the penalty-based approach indeed results in an oscillation of the stack, and thus, the time to reach a resting contact is significantly longer compared to our scheme.

5 Limitations

The presented method depends on the computed penetration depths. While it works very well for most scenarios, penetration depths cannot be computed with [10] if the colliding object is entirely enclosed by the penetrated object. Moreover, the penetration vectors computed by this method are approximations of the true penetration depths. Thus, we commonly do not arrive at an exact contact surface but get small overlaps. The size of these overlaps depends on the mesh surface curvature and the vertex sampling density.

Due to the coarse surface representation, it is not possible to reach a perfect local force equilibrium. While it is very well reached over a larger contact area, the force difference can be rather high for certain point-surface triangle pairs. Further, edge-edge collisions cannot be detected and thus are not handled.

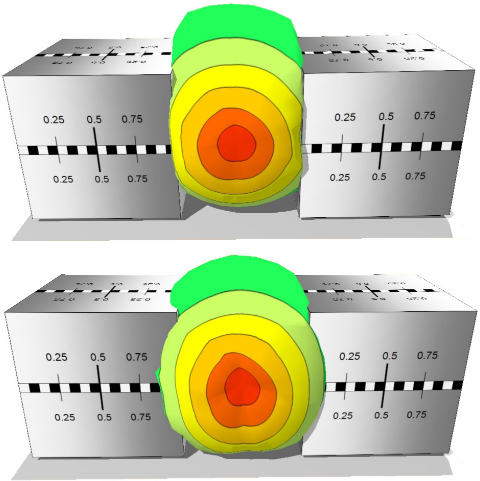

Figure 8: A soft sphere is squeezed between two stiff cubes. The purpose of this experiment is to illustrate the behavior of the collision response scheme at high pressures. Top: Collision response by computing the contact surfaces. Bottom: Collision response using a penalty-based approach. Note the significant overlaps of the sphere and the cubes in the resting state.

6 Conclusion and future work

We have presented a collision response scheme that computes the contact surface for colliding objects. We have focused on problems related to simulations with discrete time steps and discretized surface representations. Moreover, we have showed that the computed contact surface is valid even for coarsely sampled surfaces with locally different vertex densities.

By comparing the method to a penalty-based approach, we have illustrated the conceptual advantages of our scheme: The contact surfaces exclusively depend on the internal forces, no user-defined parameters are needed. Further, the method is independent of the deformable model and object representation, as long as the object surfaces are triangulated. Finally, the method even processes complex scenarios in real-time which makes it suitable for surgery simulations and applications in entertainment technologies.

Currently, we are working on a generalization of the approach in order to handle collisions of non-watertight surfaces such as cloth or shells.

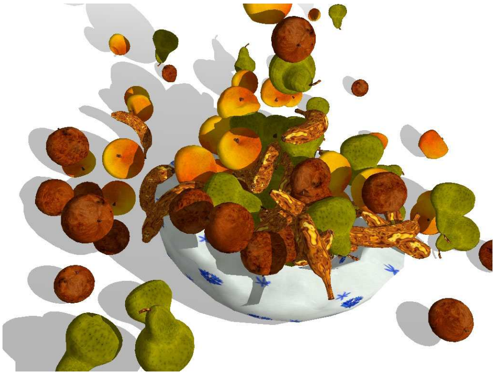

Figure 9: A dish is filled with soft fruits. Note, that the fruits are visualized with a high-resolution textured mesh while the simulated surface mesh is coarsely sampled. This leads to a strongly discontinuous behavior of contact surfaces that is difficult to handle. However, our method still provides plausible results.

References

[1] D. Baraff, “Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies”, *Proc. Siggraph*, pp. 223-232, 1989.

[2] D. Baraff, A. Witkin, “Dynamic Simulation of Non-penetrating Flexible Bodies”, *Computer Graphics*, vol. 26, no. 2, pp. 303-308, 1992.

[3] D. Baraff, “Issues in Computing Contact Forces for Non-penetrating Rigid Bodies”, *Algorithmica*, vol. 10, pp. 292-352, 1993.

[4] D. Baraff, “Fast Contact Force Computation for Non-penetrating Rigid Bodies”, *Proc. Siggraph*, pp. 23-34, 1994.

[5] M. Desbrun, P. Schröder, A. Barr, “Interactive Animation of Structured Deformable Objects”, *Proc. Graphics Interface*, pp. 1-8, 1999.

[6] C. Duriez, C. Andriot, A. Kheddar, “Signorini’s Contact Model for Deformable Objects in Haptic Simulations”, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[7] C. Duriez, F. Dubois, A. Kheddar, C. Andriot “Realistic Haptic Rendering of Interacting Deformable

Objects in Virtual Environments", *IEEE Transactions on Visualization and Computer Graphics*, to appear.

[8] F. Faure, "An Energy-Based Method for Contact Force Computation", *Proc. Eurographics*, pp. 357-366, 1996.

[9] M.-P. Gascuel, "An Implicit Formulation for Precise Contact Modeling between Flexible Solids", *Proc. Siggraph*, pp. 313-320, 1993.

[10] B. Heidelberger, M. Teschner, R. Keiser, M. Müller, M. Gross, "Consistent Penetration Depth Estimation for Deformable Collision Response", *Proc. Vision, Modeling, Visualization*, pp. 339-346, 2004.

[11] K.L. Johnson, *Contact Mechanics*, Cambridge University Press, ISBN 0521255767, 1985.

[12] R. Keiser, M. Müller, B. Heidelberger, M. Teschner, M. Gross, "Contact Handling for Deformable Point-Based Objects", *Proc. Vision, Modeling, Visualization*, pp. 315-322, 2004.

[13] N. Kikuchi, J.T. Oden, *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, ISBN 0898714680, 1988.

[14] B. Mirtich, J. Canny, "Impulse-based Dynamic Simulation", *Algorithmic Foundations of Robotics*, pp. 407-418, 1995.

[15] M. Moore, J. Wilhelms, "Collision Detection and Response for Computer Animation", *Proc. Siggraph*, pp. 289-298, 1988.

[16] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, M. Gross, "Interaction of Fluids with Deformable Solids", *Journal of Computer Animation and Virtual Worlds (CAVW)*, vol. 15, no. 3-4, pp. 159-171, 2004.

[17] M. Pauli, D.K. Pai, L.J. Guibas, "Quasi-Rigid Objects in Contact", *Eurographics/ACM Siggraph Symposium on Computer Animation*, pp. 109-119, 2004.

[18] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, M. Gross, "Optimized Spatial Hashing for Collision Detection of Deformable Objects", *Proc. Vision, Modeling, Visualization*, pp. 47-54, 2003.

[19] M. Teschner, B. Heidelberger, M. Müller, M. Gross, "A Versatile and Robust Model for Geometrically Complex Deformable Solids", *Proc. Computer Graphics International*, pp. 312-319, 2004.

[20] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raphupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, P. Volino, "Collision Detection for Deformable Objects", *Computer Graphics Forum*, vol. 24, no. 1, pp. 61-81, 2005.

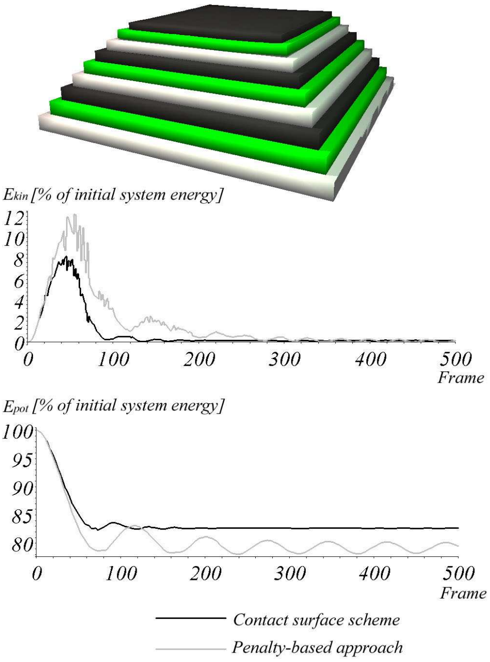

Figure 10: Simulation of a stack of nine soft membranes. Top: Experimental setup. Middle: Plot of the total kinetic energy of the scene over time, in percent of the initial system energy. Note, that less time is required to reach a resting state when the contact surface response scheme is applied. Bottom: Plot of the total potential energy in percent of the initial system energy. Note the strong oscillation of the stack when collision response is done using the penalty-based approach. Further, the stack is more compressed, because the method allows penetrations. When using the contact surface method, the oscillation is significantly reduced and the stack is stiffer due to the non-penetration constraint.