

Collision Adaptive Particle Systems

Olaf Eitzmuß

Bernhard Eberhardt

Michael Hauth

Wolfgang Straßer

Wilhelm-Schickard-Institut

Universität Tübingen

Auf der Morgenstelle 10/C9, 72076 Tübingen, Germany

phone [+49] (7071) 29-75465, fax [+49] (7071) 29-5466

E-mail: {etzmuss, beberh, mhauth, strasser}@gris.uni-tuebingen.de

Abstract

Particle systems have been widely employed to simulate deformable objects, most prominently virtual textiles. In this paper we introduce particle systems that adaptively generate new particles wherever this is required to model collisions correctly. When coarse meshes collide with an object they are automatically refined. Therefore this model allows physically accurate simulations that require a much smaller number of particles than regular particle systems. This allows for fast simulations with coarse meshes.

1. Introduction

Modelling animated cloth has been the focus of considerable research in recent years and the quest for fast simulations has led to the development of various new approaches.

In particular, mass-spring particle systems[2, 4] have been proven to be an appropriate model for the animation of textiles. A particle system models the deformable object as a two-dimensional grid the vertices of which represent the particles. The model implies that all mass is contained in the particles and interior and exterior forces act on the particles. Interior forces are most commonly tension, bend and shear forces and one can imagine that each particle applies forces through virtual springs, that connect a particle to its neighbours.

When a deformable object collides with another object in its environment and both objects are given as meshes (particle systems), two types of collisions occur (figure 1):

1. face/particle collision
2. edge/edge collision

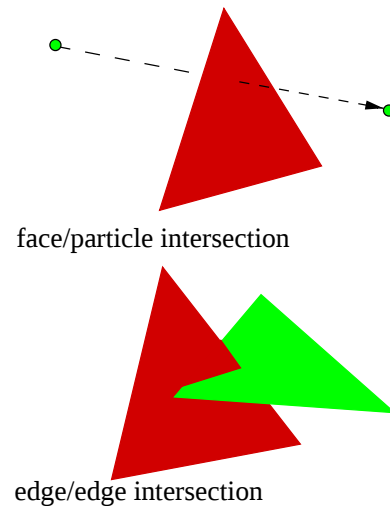


Figure 1. Types of collisions

In particle systems forces only act on particles. Therefore it is natural to handle such collisions that occur when a particle of the deformable object penetrates the environment (face/particle collision with an animated penetrating particle), for instance by applying forces to the penetrating particles or modifying their positions and velocities. It is not obvious, however, how edge/edge collisions or collisions where an environment particle penetrates an object face are to be handled.

Because of these collisions the number of particles (the discretization of the system) is required to be very large so that the deformable object can accurately fit the surface of the environment that it collides with. Adaptivity is a common concept to avoid a uniformly very fine discretization in numerical analysis and geometric modelling. Therefore we

aim to reduce the computational costs in physically based modelling with an adaptive approach as well.

We observe that in some regions of the particle system a coarse discretization is sufficient for an accurate animation. Wherever collisions occur, however, the deformable objects have to fit the object surface they collide with. These observations have led us to develop collision adaptive particle systems that generate new particles where collisions occur.

In this paper we suggest a new method that handles all types of collisions correctly and allows correct simulations requiring only a small number of particles. We describe collision of the textile with one or more rigid bodies in its environment.

The adaptively inserted particles give rise to a new class of particles which are only governed by the collision and not directly by the differential equation. Therefore, these particles will be called virtual particles.

Using this adaptivity we are able to simulate textiles with very coarse discretization and make a step forward towards physically based real time animation. The method is independent of the actual implementation of the particle system.

Notation Although we can model all two-dimensional deformable objects the deformable object will be called textile for simplicity. The object it collides with will be called environment, which we assume to be one or more rigid bodies. The adaptively inserted particles will be noted as virtual particles whereas the original particles are called base particles.

The remainder of this paper is organised as follows: in section 2 related work is briefly reviewed. In section 3 a short introduction to particle systems is given. The concept of collision adaptive particle systems is described in section 4. The computation of the positions of virtual particles in an adaptive particle system is laid out in section 5. Section 6 outlines the modification of forces by the use of virtual particles, followed by results and conclusions in sections 7 and 8.

2. Previous Work

Early work on animation of deformable objects was based on deformation energy functions using continuity formulation (e.g. Terzopoulos et al.[10]).

Other authors proposed particle systems as a discrete model for the simulation of the behaviour of clothes (cf. Breen et al.[2], Eberhardt et al.[4]). Particle systems will be described in more detail in the next section. Grimsdale and Ng[6] presented a survey of techniques for cloth modelling

in computer graphics.

Hutchinson et. al. [3] picked up the idea of adaptivity and developed a multiresolution particle system that was adaptive to high curvature, i.e. a higher resolution was used where the angles in the system deviated strongly from their rest angles.

The main focus of recent work was to achieve better performance of the simulations to reach reasonable computation times for animations. In particular the numerical performance was improved by Barraff and Witkin[1] who achieved considerably larger step sizes. They proposed a framework that was formulated in terms of constraints rather than forces. Forces, however, can be derived immediately from constraints. Their main contribution was the use of an implicit numerical ODE solver that carries out one solver step very fast and can also very efficiently deal with constraints as imposed by collisions.

Collision detection and response is a major problem in physically based modelling and was addressed by Volino et al.[11], who proposed a fast algorithm for collision detection but did not give special treatment to the different types of collisions. Provot[7] described the various types of collisions but did not address the problem of handling edge/edge collisions, either.

Often a minimal offset distance between the particles of the textile and the environment is used that prevents collisions that are not handled correctly by the system. When less particles are to be used the minimal distance has to be increased and the shape of the environment object covered by a coarse textile mesh is not correctly represented. Since our method works on top of a conventional collision detection and response method, it cannot be compared with one of these methods.

3. Animation with Coupled Particle Systems

Coupled particle systems discretize the object by a grid of particles that contain all the mass of the object but do not have any volume. In contrast to the particle systems introduced by Reeves[8] to model fuzzy objects there are permanent links that connect each particle to its neighbours. The particle system has a topology that defines the shape of the deformable object. Hence particle systems are represented by a mesh with this topology and the particles as its vertices.

There are interior energies due to tension, bending and shearing in the object. They are defined by the distances between linked particles (lengths of the edges of the mesh) and the angles between the links at one particle.

The forces due to the interior energies can be thought of as spring forces. Therefore energy dissipation is modelled with damping forces that counteract the velocity of the spring elongation.

In addition to the interior forces there are exterior forces that act on the particles due to gravitation, air resistance, collisions etc.

Since particle systems are already a discretization in space only a system of ordinary differential equations has to be solved over time, whereas continuity formulations require partial differential equations to be solved. The trajectory of each particle is computed by Newton's law, which is

$$F_i(x) = m_i \cdot \ddot{x}_i$$

for the i -th particle.

Various numerical methods are available for the solution of ordinary differential equations. Since the equations stem from the modelling of textiles, stiff implicit methods should be used (e.g. see [5]).

4. Adding Virtual Particles

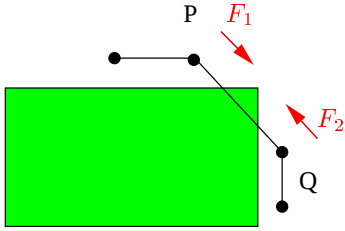


Figure 2. A collision of an edge in 2D-space

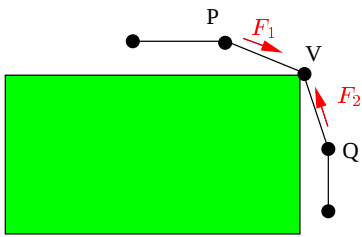


Figure 3. Adding a virtual particle to correct penetration

We start by considering the collision in 2D depicted in figure (2). The particles of the textile are guaranteed to

be outside of the environment by applying forces to the particles or by changing their position and velocity[4, 7, 1]. However, an edge collision causes a penetration. We can observe that this failure is due to an insufficient resolution of the deformable model. This can be fixed by adding a new particle to the system (figure 3). In general, the edge can intersect the environment arbitrarily often and several particles have to be inserted in the edge as it will be described in the next section.

We could now add new particles to the permanent mesh that describes the deformable object. But this would lead to an increase of the number of particles in each time step and the particles that were included at time t_n may not be needed at time t_{n+1} or they are needed at a different position according to the collision at this time.

Hence we do not compute trajectories from the differential equation for these particles but we compute the positions of these virtual particles from the collision at each time step. They are only valid for this single time step. Thus the positions of these virtual particles are governed by the current position of the textile and the environment.

Having inserted the virtual particles at a certain time they can be used to display the current frame, such that the user cannot observe any penetration of the animated object.

Furthermore, we observe that the length of the modified edge in figure 3 has increased due to the inserted point and the model does not match the original physical description, i.e. there is a stretch of the textile without an increase of tension energy. When we use a coarse mesh even cases like in figure 4 may occur, where the textile drapes on the box in the environment penetrating the box without any interaction. The textile would eventually fall through the box because there are no forces to prevent this.

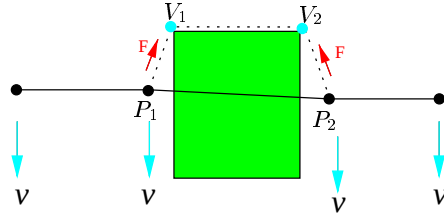


Figure 4. No interaction results from collision, unless virtual particles are used

Hence physical accuracy is improved by integrating the virtual particles into the model. Virtual particles apply forces that act on their neighbours. In the situation of figure 3 the forces acting on P are now computed using the

topology of the textile mesh in which V is neighbour of P instead of Q . In figure 4 the textile now is held by the virtual particles at the corners of the box.

In each time step a new topology of the mesh is generated and forces (or energies) are computed for this topology. Note that no forces that act on virtual particles must be computed because these particles do not move on a trajectory of a differential equation. But the virtual particles apply forces that pull the adjacent particles back together to counteract the stretch due to the collisions.

5. Computing Virtual Particles

This section describes how, starting from a triangular textile mesh, all virtual particles can be computed. If the textile is given as a rectangular mesh it can be converted to a triangle mesh easily. The method assumes that all base particles are outside the environment mesh, i. e. our method cooperates with a conventional collision detection and response method that prevents the base particles from penetrating the environment. Only after this conventional collision response virtual particles are computed for each face.

The objects in the environment are required to be given as a mesh (this is not a serious limitation as in practice all geometric objects are given as meshes or can be converted to one).

We aim to guarantee that all edges of the adaptively refined textile mesh be outside the environment mesh. This is achieved by a projection method in which all edges or segments of edges of the environment are projected onto the textile. Those particles and segments of edges of the environment that have penetrated the textile are inserted in the topology of the original textile mesh.

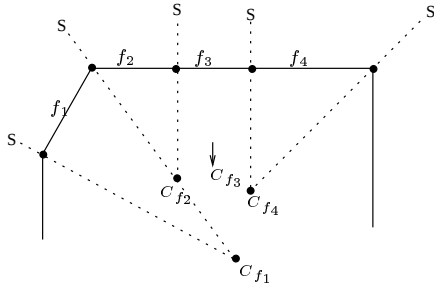


Figure 5. Planes S partition the volume above the faces and define centres of projection C_{f_i}

In the following the projection method will be described:

The projection P , that maps the edges of the environment mesh onto the textile mesh, is given procedurally by defining the restriction of P to one textile mesh face f noted as P_f . All face normals n_f are assumed to be normalised.

As a first step we define a centre of projection for each face f :

For each face f we define a bounding volume by three planes S_i , where S_i is a plane separating f from the adjacent textile face f_i . S_i is given by the normal $n_{av_i} = \frac{n_f \times n_{f_i}}{\|n_f \times n_{f_i}\|}$, which is defined to be perpendicular to both face normals and a common vertex P_i :

$$S_i : \langle n_{av_i}, (X - P_i) \rangle = 0$$

Let C_f be the intersection points of these three separating planes. The restriction of P to face f is a central projection given by its centre of projection C_f . The projections are defined such that the space above the faces is partitioned by the bounding volumes. Figure 5 illustrates this projection for the two-dimensional space.

If the normals n_{av_i} are linearly dependent then C_f is a point at infinity and the central projection degenerates to a parallel projection. In either case the projection P_f is defined by a point in projective space represented by its homogeneous coordinates.

We have defined a projection that maps an edge e onto the plane spanned by f . Since we want to restrict the central projection to a single mesh face f and also only parts of the textile mesh that actually penetrate the environment are to be considered, some clipping steps have to be taken.

Each edge $e = \overline{XY}$ of the environment edge is projected onto f with vertices P_1, P_2, P_3 as follows:

1. Compute the heights of X and Y above f : $h_X = \langle n_f, X - P_1 \rangle, h_Y = \langle n_f, Y - P_1 \rangle$. Only points of an edge that have positive height are to be considered. Points with negative height do not penetrate. If a segment of an edge has negative height it is clipped off and a new end point is computed by linear interpolation. If $h_X < 0$ and $h_Y < 0$ the edge does not penetrate at all and is skipped.
2. Given C_f project e onto the plane spanned by f .
3. Intersect the projected edge with the edges of face f . Clip off segments that are outside the face f . The end

points of the segment contained in f are noted as $\overline{V}, \overline{W}$ (all projected points will be denoted using an overline).

Four cases can occur (figure 6). If no segment of the edge is contained in f , the edge is skipped (6(a)). In 6(b) the edge is clipped twice. In 6(c) only one new end point is computed, whereas in 6(d) e lies completely in f .

The mapping of two edges e_1 and e_2 is illustrated in figure 7.

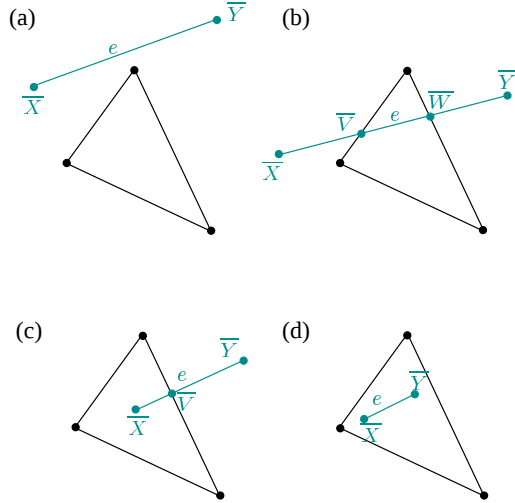


Figure 6. Four possible positions of the edge e with respect to the mesh face f

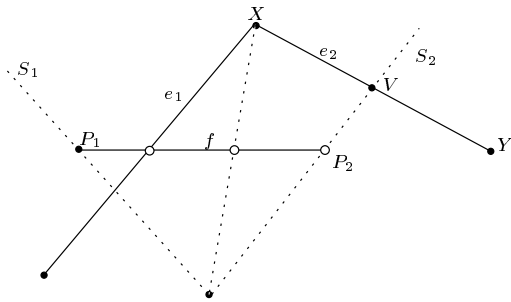


Figure 7. projection P in a two-dimensional cross section

4. If, say, $\overline{V}$ is not an end point of the projected edge e , $V = P^{-1}(\overline{V})$ is yet to be computed.

Let α, β be the barycentric coordinates of V with respect to X, Y . The $\text{ratio}(X, V, Y) = \frac{\alpha}{\beta}$ is invariant

under affine maps and V can be computed by linear interpolation. In this case of a central projection, however, V cannot be computed by linear interpolation because ratios are not invariant under projective maps, whereas the cross ratio is invariant:

$$\frac{\text{ratio}(X, Y, V)}{\text{ratio}(X, Y, A)} = \frac{\text{ratio}(\overline{X}, \overline{Y}, \overline{V})}{\text{ratio}(\overline{X}, \overline{Y}, \overline{A})}$$

This can be exploited to compute V if we have an auxiliary point A and its image $\overline{P}$. Hence we project the auxiliary point $A = \frac{1}{2}(X + Y)$ onto face f to get $\overline{A}$ and solve the above equation for V .

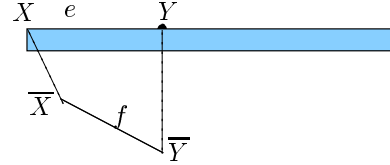


Figure 8. detection of non penetrating edges

5. The rays from the end points of an edge to their images under projection P must not intersect other objects. For instance, this case occurs if the textile has not penetrated, but is on the opposite side of the environment object. Then a ray from the opposite side of the environment object would penetrate an environment face facing the deformable object.

In the situation of figure 8 edge e would be handled as an edge penetrating face f , although the textile face is on the other side of the solid environment object. This situation is detected by the rays intersecting the lower face of the solid.

The points $\overline{V}_i$ are either on the boundaries of f corresponding to an edge/edge collision or inside f corresponding to a particle/face collision (environment particle penetrating the textile).

Each face f of the textile becomes a sub mesh that contains the original three vertices (base particles) as well as all newly computed virtual particles and the edges connecting them. At this stage the geometry of this sub mesh is already known because we know the positions of P_1, P_2, P_3 and all virtual particles V_i . The topology is yet to be computed by a constrained Delaunay triangulation and for this we will use $\overline{V}_i$.

First the new boundary topology is obtained by inserting the new vertices that lie on the boundary of f in the face

boundary in the correct order. One component of their barycentrics with respect to P_1, P_2, P_3 is zero and therefore the barycentrics can be used to sort the vertices that lie on the boundary.

The line segments in the face as well as the face boundary are constraints to the topology of the sub mesh. We perform a constrained 2D Delaunay triangulation of P_1, P_2, P_3 and all $\bar{V}_i$ that lie in f in the plane of f (in our implementation we use a very stable triangulator by Shewchuk[9]). Then the topology computed in 2D is combined with the 3D geometry to give the sub mesh, i.e. we replace $\bar{V}_i$ with V_i .

Finally all sub meshes (faces) are merged to one single mesh that contains all base and virtual particles and all edges connecting them.

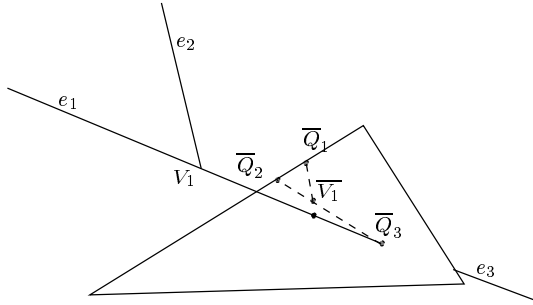


Figure 9. Edges projected onto a face

An example of the projection is given in figure 9. Those edges above the triangle penetrate the textile. Edge e_3 only partly penetrates the face and only the penetrating segment of e_3 is considered. Vertex V_1 is projected onto $\bar{V}_1$ and becomes an interior point. The edge/edge collisions give new vertices $\bar{Q}_1$ and $\bar{Q}_2$, while $\bar{Q}_3$ is the point of e_1 , where e_3 is clipped off.

A simple example of a collision is depicted in figure 10. The four particle system has edge/edge collisions with the upper right edge of the box. Three virtual particles have been inserted into the mesh of the textile due to these edge/edge collisions.

The use of acceleration structures is necessary because the projection method has a complexity of $O(\#edges_{env} \cdot \#faces_{textile})$, as all edges are projected onto all faces. In order to reduce the number of edge/face pairs for which a projection has to be carried out, the same algorithms as used for collision detection for deformable objects can be employed, which have been widely discussed in literature (e.g. Volino et al.[11]).

In our implementation we are using a grid based acceleration. The faces of the deformable object are sorted into a grid. Then for each edge those grid cells that the edge is contained in are selected. Thus the projection only has to be performed for those faces of the textile that are in these grid cells.

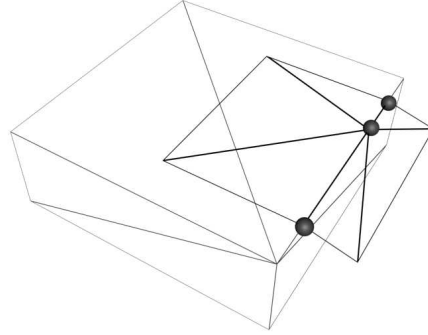


Figure 10. A four particle mesh falling onto a box

The complexity of the triangulation step is $O(\#faces_{textile})$ because the triangulation is local and its cost is limited by the maximum number of particles to be inserted into one face.

6. Physically Interacting Virtual Particles

As mentioned above virtual particles are not only used for visualisation but also apply forces acting on their neighbours.

If the triangle mesh computed in the previous section is used for the physical simulation the number of neighbours of a particle may change in each time step. The interior forces due to tension, shearing and bending depend on the particle/particle distances, in-plane shear angle and out-of-plane bend angle, respectively. The rest distances and rest angles, where the system is at an equilibrium state, must be adopted as well to fit the new topology and geometry.

We found it sufficient to employ the virtual particles on the face boundaries only for the physical simulation and we ignore such virtual particles that lie inside a mesh face. Thus we do not have to deal with a changing topology and locally we only have to adopt to change positions and velocities.

We will explain this looking at the following example: Assume that particle P_0 has four neighbours P_1, P_2, P_3, P_4 and two virtual particles V_1 and V_2 where computed on edge $\bar{P}_0, \bar{P}_1$ as shown in figure 11. The forces acting on

P_0 are then computed by using V_1 instead of P_1 as a direct neighbour of P_0 .

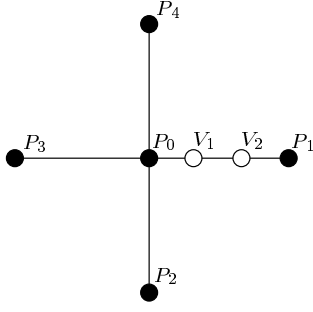


Figure 11. Local topology at P_0

Hence in the implementation we only have to replace the position and velocity of P_1 with V_1 . The velocity of the virtual particles is needed to compute damping forces and is set equal to the velocity of the point in the environment it corresponds to, because it is assumed that this point in the textile will stay attached to that point of the environment. For a static environment all the velocities of virtual points are set to zero.

Likewise virtual particles can replace the base particles P_2 , P_3 and P_4 , too.

New rest lengths for the springs controlling the tension must be computed. When an edge is split into several segments, the sum of the rest lengths of these segments must be equal to the rest length of the original edge to ensure area preservation. Hence the rest length r_j of the j -th segment is computed by

$$\frac{\Delta l_j}{\sum \Delta l_i} = \frac{r_j}{r}.$$

where r is the rest length assigned to that edge and l_i is the current length of the i -th segment.

This way every time the evaluation function is called each particle is checked for virtual neighbours. If virtual neighbours are found for a base particle they replace the original neighbours and forces guide the particle into the correct direction and restore the correct edge lengths. Hence the insertion of virtual particles is a trivial replacement of geometry coordinates.

One might also want to use virtual particles to compute bend and shear forces. Unfortunately this would decouple the system, because these forces cannot be transmitted from one base particle to the next if there are virtual particles in between.

Note that the insertion of virtual particles does not change the material properties. We can imagine that we first split a spring between two particles into two. Using new rest lengths as described above, this does not change the material properties. Then the virtual particle is moved to accommodate the collision such that the modelling becomes more accurate, while the simulation maintains the correct material parameters.

Multi-step scheme ODE solvers are not suitable because the system is changed in each step and previous values do not match the current system. On the other hand, in each particle system, where sudden collisions occur, the introduced stiffness makes the use of implicit single-step methods recommendable and the adaptive system is not so much different from the non-adaptive case.

7. Results

We implemented the presented techniques based on the particle system described by Eberhardt et al.[4]. In particular a rectangular mesh was used for the simulation. For visualisation the rectangle mesh is converted to a triangle mesh. The ODE is solved by the implicit Euler method.

Figure 13 shows a textile modelled by only two triangle patches falling over a box. The penetration in picture 13(a) is caused by three edge/edge collisions. Three virtual particles on the boundary are inserted as in figure 10 so that this new mesh does not intersect the box (figure 13(b)).

In colour plate (a) the mesh of a tablecloth that falls over a round table can be seen. Although it only contains 100 particles the circular shape of the table can be modelled accurately.

The third example shows a textile falling over a ball. This example demonstrates, what can be achieved by adaptive simulations. The mesh of the ball is quite complex and consists of 1227 edges. When no adaptively inserted particles are used a coarse mesh over a ball looks as shown in colour plate (e). The particles on top of the ball are still visible, but the faces have sunk into the ball. Colour plates (c) and (d) show two refinements at different time steps of the simulation. The very fine mesh of the ball is merged into the coarse mesh of the textile. Finally colour plate(f) shows the refined mesh of the textile covering the top of the sphere completely.

Finally figure 14 shows a comparison between a textile with 2500 particles falling over a table without adaptivity and a mesh with 400 base particles and adaptively inserted

Example	#Particles	projection	topology	combined
round table	100	38.5ms	38ms	76.5ms
square table	400	24ms	147.5ms	171.5ms
ball	100	150.5ms	83.5ms	234ms

Figure 12. execution times for the computation of virtual particles averaged over 1s of simulation time

virtual particles. The 400 particle mesh is obviously much faster to compute and a good approximation of the high resolution mesh.

The additional computational costs of the adaptivity are due to the computation of virtual particles as described in section 5. In the simulation process we only have to fill in the new coordinates and new rest lengths. In table 12 the execution times for the three examples in the colour plates and 14 are given. The values are averaged over one second of simulation time on an R10000/180MHz processor. The third column gives the times for the execution of the projection step, the fourth column gives the times for the computation of the topology and the last column shows the combined values.

Note that for the physical simulation as described in section 6 we need to compute the new topology, since only virtual particles on the boundary are used.

The overall performance depends on the underlying physical system and the numerical solvers used.

8. Conclusions and Future Work

We have presented a method that handles all types of collisions that occur in animation with particle systems while ensuring very fast animation computation by using only a few particles. Our approach is not restricted to one cloth modelling approach but can be incorporated into many systems that were presented in recent years.

The main focus of future work will be the improvement of the performance of the projection algorithm. More specialised acceleration methods and algorithmic improvements will further reduce the costs of the projection.

The animation of wrinkles with coarse meshes hasn't been addressed. Coarse discretization particle systems suffer from the disability of modelling wrinkles and folds accurately. An approach tackling this problem was presented by Hutchinson et. al. [3] using curvature dependent particle systems. In the future we will combine collision dependency with curvature dependency to obtain fully adaptive systems.

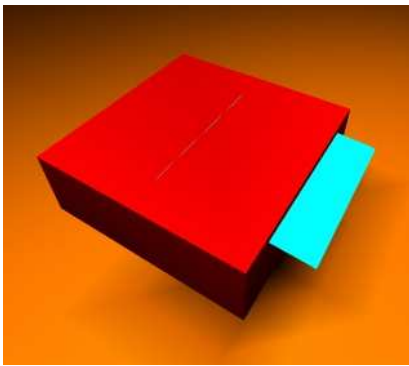
Furthermore the numerical solver has to accommodate the introduced adaptivity and we plan to customise our integration scheme to improve numerical performance.

9. Acknowledgements

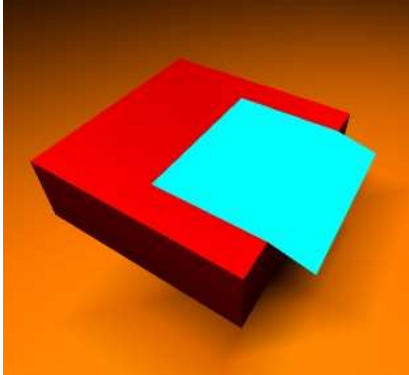
This work was supported by the DFG MoViTex grant. We would like to thank Reinhard Klein and Andreas Schilling for the valuable discussions. Ralf Sonntag supported the rendering of the images with RADIOLAB.

References

- [1] D. Baraff and A. Witkin. Large steps in cloth simulation. In M. Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 43–54. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [2] D. E. Breen, D. H. House, and M. J. Wozny. Predicting the drape of woven cloth using interacting particles. In A. Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 365–372. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [3] T. H. Dave Hutchinson, Martin Preston. Adaptive refinement for mass/spring simulations. In *7th Eurographics Workshop on Animation and Simulation*, pages 31–45. Eurographics, Springer, Sept. 1996.
- [4] B. Eberhardt, A. Weber, and W. Strasser. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, 16(5):52–60, Sept. 1996.
- [5] E. Hairer and G. Wanner. Solving Ordinary Differential Equations II. Springer-Verlag, Berlin, 1996.
- [6] H. N. Ng and R. L. Grimsdale. Computer graphics techniques for modelling cloth. *IEEE Computer Graphics and Applications*, 16(5):52–60, Sept. 1996.
- [7] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Graphics Interface '97*, pages 177–189. Canadian Information Processing Society, Canadian Human-Computer Communications Society, May 1997.
- [8] W. T. Reeves. Particle systems — A technique for modeling a class of fuzzy objects. *Computer Graphics*, 17(3):359–376, July 1983.
- [9] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In M. C. Lin



(a)



(b)

Figure 13. 4-particle mesh draping on a box, (a) without virtual particles, (b) with virtual particles

and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

- [10] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4:306–331, 1988.
- [11] P. Volino and N. Magnenat-Thalmann. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. In *EuroGraphics*, volume 13, pages 155–166, 1994.

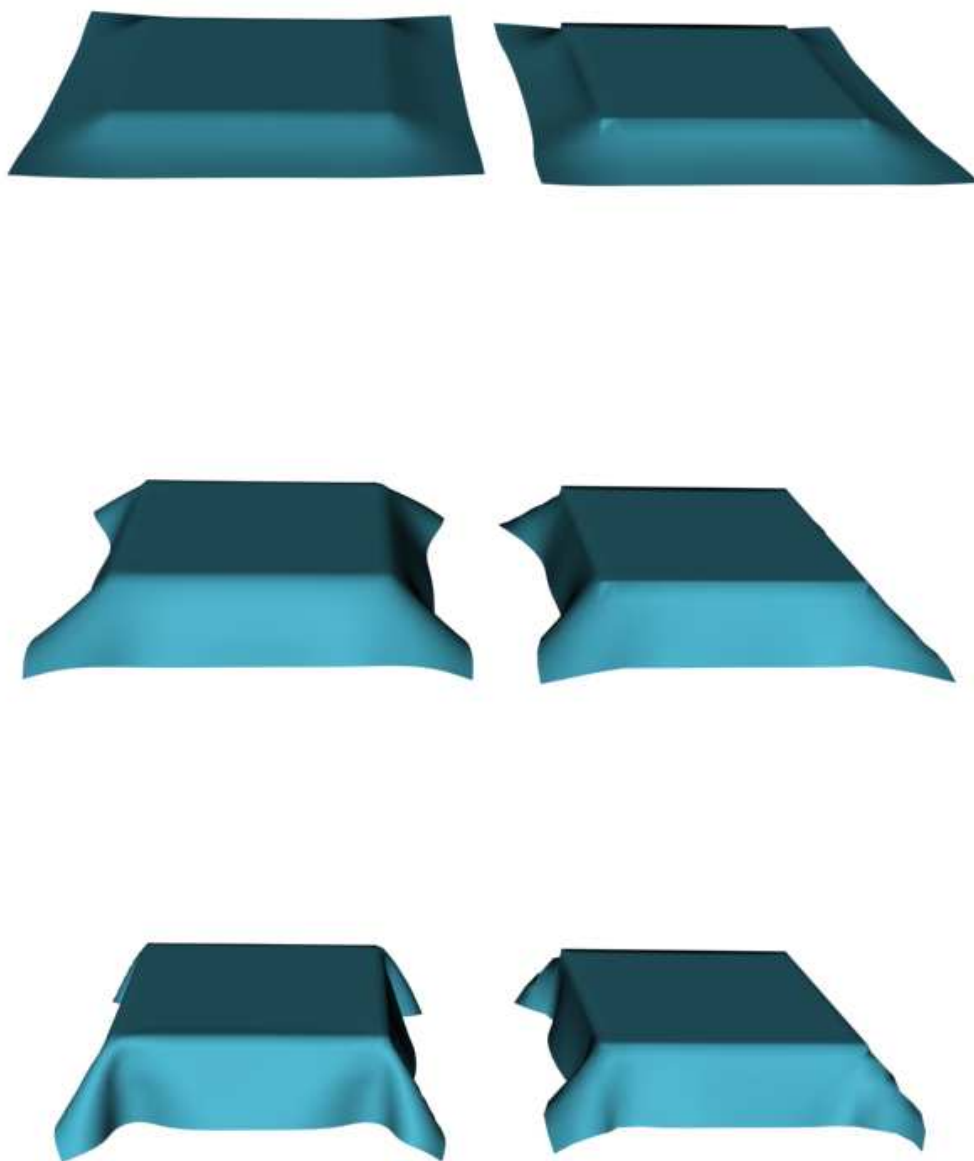


Figure 14. Right: 2500 particles without adaptivity, Left: fast simulation with only 400 particles and adaptivity