Simulating Almost Incompressible Deformable Objects

Raphael Diziol Daniel Bayer Jan Bender

Universität Karlsruhe (TH), Karlsruhe, Germany

Abstract

We present a new method for simulating almost incompressible deformable objects. A tetrahedral model is used to represent and restore the volume during the simulation. A new constraint, which computes impulses in the one-ring of each vertex of the tetrahedral model, is used in order to conserve the initial volume. With different parameters, the presented method can handle a large variety of different deformation behaviors, ranging from stiff to large deformations and even plastic deformations. The algorithm is easy to implement and reduces the volume error to less than 1% in most situations, even when large deformations are applied.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Physically based modeling, Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The dynamic simulation has become an important topic in computer graphics literature. In many applications, such as virtual environments, movie special effects and games, the demand for simulating multi-body systems, cloth and water is growing. Another research area needed for these applications is the simulation of deformable objects. The applications do not necessarily require physically correct deformable models, but a physically plausible dynamic behavior. The problem is that a volume loss can still be noticeable when large deformations are applied to the objects. Especially in the area of medical simulations, the deformable objects are nearly incompressible. Therefore, the model should conserve the volume as good as possible. However, the simulation of incompressible deformable objects is still a challenging problem until today.

In this paper we present a new method for simulating deformable objects, where the volume is almost conserved. In order to describe the volume we build a tetrahedral mesh which is deformed by external forces during the simulation. The edges of the mesh are assigned to distance constraints, which restore the initial form of the object. Additionally we introduce a new volume constraint, which restores the volume in a one-ring of a vertex by applying impulses to the surrounding particles. While the simulation and contact handling is done with the tetrahedral mesh, a high-resolution triangle mesh can be coupled with the tetrahedral mesh for the final visualization. The paper presents various scenes where the new method was applied to, and compares the volume error of the new approach to a similar existing one. The results show, that the new approach reduces the error to less than 1% in most situations, even when large deformations occur. We also incorporate plastic deformations in our test scenarios, which are easy to add to the presented method.

2. Related work

Since the presentation of a general physical model for simulating two- and three-dimensional deformable objects by Terzopoulos et al. [TPBF87], many approaches besides the classical FEM simulations [MDM*02, OBH02] have been intensively studied. In [MHTG05] an easy approach to simulate deformable objects is presented, where no connectivity information between the simulated particles is needed. In order to minimize the volume error during the simulation, a quadratic energy is minimized which matches the original configuration of the object with the deformed one. Irving et al. [ISF07] present an approach adopted from fluid dynamics simulation, which is able to nearly eliminate the volume error during the simulation. The key idea is to make the velocity field divergence free in a one-ring of tetrahedrons to preserve the volume. The interaction between fluids and deformable objects has also been studied [RMSG*08]. Becker et al. [BIT09] show how the smoothed particle hydrodynamics method can be used to simulate deformable solids, which

makes coupling between water and deformable objects quite easy. More control of the deformations can be achieved by using key frame interpolations as shown in [AOW*08]. In [DBB09] a new idea using the impulse-based dynamic simulation [Ben07] was presented to ensure the conservation of the volume, but the approach could not handle inverted tetrahedrons.

Our new approach is based on the work of Teschner et al. [THMG04]. They show how triangle and tetrahedral meshes with up to thousand primitives can be simulated at interactive speed.

3. Deformable objects

In order to simulate deformable solids a volumetric structure has to be created. Tetrahedral meshes are commonly used for this purpose. The vertices of the mesh represent particles with common attributes like mass and velocity, while the tetrahedrons represent the volume. There exist different approaches to built such meshes. In [LS07] a tetrahedral mesh is built from a signed distance field. The resulting tetrahedrons are small on the boundary to adapt the isosurface and can be bigger in the interior of the volume where accuracy is not that crucial. In contrast Spillmann et al. [SWT06] present an algorihm to build a tetrahedral mesh from an arbitrary triangle soup which works for damaged surfaces, too, where the enclosed volume is not defined. This method was also used in [DBB09] and works well even when the tetrahedral mesh is not perfectly aligned to the original mesh. We use this method to build our volumetric structure but others would work, too. Figure 1 shows the resulting tetrahedral mesh built from the corresponding triangle mesh.

A fast approach to simulate deformable solids was presented in [THMG04]. They define constraints of the form $C(\mathbf{p}_0,\ldots,\mathbf{p}_{n-1})$ to represent the deformations of the object. The scalar function C is zero, if the object is undeformed and greater than zero, when deformations occurred. These constraints depend on particle positions $\mathbf{p}_i$ and define the potential energy

$$E(\mathbf{p}_0,\ldots,\mathbf{p}_{n-1}) = \frac{1}{2}kC^2 \quad ,$$

where k denotes a stiffness coefficient. Forces $\mathbf{F}^i$ from these energy are derived at each particle i by taking the negative gradient of E:

$$\mathbf{F}^i(\mathbf{p}_0,\ldots,\mathbf{p}_{n-1}) = -\frac{\partial E}{\partial \mathbf{p}_i} = -kC\frac{\partial C}{\partial \mathbf{p}_i} \quad .$$

These forces always point in directions which minimize the constraint C. The overall force considered at particle i during a simulation step is the sum of all forces based on potential energies that consider this particle. For each edge of the model we get distance forces derived from

$$E_{\mathrm{D}}(\mathbf{p}_i,\mathbf{p}_j) = \frac{1}{2}k_{\mathrm{D}}(|\mathbf{p}_i-\mathbf{p}_j|-D_0)^2 \quad ,$$

Figure 1: *The Armadillo (top) converted into a tetrahedral mesh (bottom).*

where k_{D} is the distance stiffness and D_0 is the initial distance between particle positions $\mathbf{p}_i$ and $\mathbf{p}_j$. The corresponding derived forces restore the initial form of the object over the time. Damping [SGT09] for the distance forces can also be incorporated and greatly improve the stability of the numerical simulation.

For the tetrahedrons we have a volume preserving energy

$$E_{\mathrm{V}}(\mathbf{p}_i,\mathbf{p}_j,\mathbf{p}_k,\mathbf{p}_l) = \frac{1}{2}k_{\mathrm{V}}(\mathrm{vol}(\mathbf{p}_i,\mathbf{p}_j,\mathbf{p}_k,\mathbf{p}_l)-V_0)^2 \quad (1)$$

$$\mathrm{vol}(\mathbf{p}_i,\mathbf{p}_j,\mathbf{p}_k,\mathbf{p}_l) = \frac{1}{6}(\mathbf{p}_i-\mathbf{p}_l)((\mathbf{p}_j-\mathbf{p}_l)\times(\mathbf{p}_k-\mathbf{p}_l))$$

with V_0 representing the initial volume and k_{V} the volume stiffness of the constraint. The resulting force from this energy for the particle i is orthogonal to the supporting plane of $[\mathbf{p}_j,\mathbf{p}_k,\mathbf{p}_l]$ and its length is one third of the area defined by the triangle $\mathbf{p}_j$, $\mathbf{p}_k$ and $\mathbf{p}_l$. Because the constraint uses the

signed volume of a tetrahedron the forces always point in directions which restore inverted tetrahedrons. In contrast to the distance preserving forces, these forces try to restore the initial volume.

With the above defined constraints the dynamic behavior of the deformable object can be computed by integrating the particle positions according to Newton's equation of motion. The external forces $\mathbf{F}_{\text{ext}}$, such as gravity, define the particle positions $\mathbf{p}_i$ and velocities $\mathbf{v}_i$ at time t and thus, the deformation of the object. Now the resulting forces $\mathbf{F}(t) = \mathbf{F}_{\text{D}}(t) + \mathbf{F}_{\text{V}}(t)$ derived from the deformation energies can be computed. The new particle positions and velocities are then retrieved by integrating the external forces together with the deformation forces according to

$$\begin{aligned} \mathbf{v}_i(t+h) &= \mathbf{v}_i(t) + \int_t^{t+h} \frac{\mathbf{F}(\tau) + \mathbf{F}_{\text{ext}}(\tau)}{m_i} d\tau \\ \mathbf{p}_i(t+h) &= \mathbf{p}_i(t) + \int_t^{t+h} \mathbf{v}(\tau) d\tau \quad . \end{aligned} \tag{2}$$

While different integration schemes can be used for this purpose, Teschner et al. suggest to use the Verlet algorithm [Ver67] for a fast and robust integration.

4. A more accurate volume constraint

The presented volume constraint in Section 3 uses the coefficient k_{V} to define the stiffness of the volume. Now the question arises, how to choose k_{V} in order to get an incompressible solid. Using a low value for k_{V} can result in a huge volume loss when large deformations occur. On the other hand, choosing a too big value for k_{V} can even blow up our simulation, because volumes get worse in each simulation step. Therefore, the optimal way would be to compute k_{V} explicitly for each constraint during a simulation step, so that the resulting forces restore the initial volume V_0 of a tetrahedron.

Instead of using forces to correct the volume, we can compute impulses, similar to [Ben07, MHHR07], which change the velocity of our particles. Assuming that the derived forces $\mathbf{F}_{\text{V}}^i$ from Equation 1 are constant over the simulation step with time step size h the resulting impulses are

$$\mathbf{I}^i(k_{\text{V}}) = \Delta\mathbf{v} = \frac{\mathbf{F}_{\text{V}}^i(k_{\text{V}})}{m} h = -\frac{\partial E_{\text{V}}(k_{\text{V}})}{\partial \mathbf{p}_i} \frac{h}{m} \quad ,$$

where the coefficient k_{V} defines the magnitude of the impulses.

For one tetrahedron with previewed particle positions $\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_j, \bar{\mathbf{p}}_k$ and $\bar{\mathbf{p}}_l$ at time $t+h$ we want to compute the impulses $\mathbf{I}^i, \mathbf{I}^j, \mathbf{I}^k$ and $\mathbf{I}^l$ which restore our initial volume V_0. The previewed particle positions are integrated using the already applied forces and impulses according to Equation 2. Using previewed particle positions is crucial in order to react to deformations from external forces immediately. As the impulses are constant over time, the corrected particle positions according to Equation 2 are

$$\mathbf{p}_i(t+h) = \bar{\mathbf{p}}_i + h\mathbf{I}^i(k_{\text{V}}) \quad .$$

Thus, after applying the impulses, the new volume

$$V(k_{\text{V}}) = \text{vol}(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l) = \sum_{i=0}^{3} \alpha_i k_{\text{V}}^i$$

only depends cubically on k_{V} with coefficients α_i. Therefore, the equation $V(k_{\text{V}}) - V_0 = 0$ has up to three possible solutions k_{V} which restore the initial volume of the tetrahedron. The optimal root k_{V} is the smallest positive value, because it changes the particle positions the fewest of all. Due to the fact, that the four impulses sum to zero for every k_{V}, the conservation of momentum is guaranteed.

Given a tetrahedral mesh where no tetrahedrons are inverted, the resulting mesh after applying the impulses should have no inverted tetrahedrons either. If there are only inverted tetrahedrons at the boundaries, the resulting impulses can easily repair the inverted tetrahedrons. But if there are too many of them in the interior, one should use more advanced algorithms like presented in [ST08] to get an initial uninverted tetrahedral mesh. While applying the impulses that correct the volume of one tetrahedron, these impulses can invert a neighbor tetrahedron. These inverted tetrahedrons could be corrected in the next step, but large changes of the particle positions can occur, resulting in unnatural movement. To prevent this situation, we have to see how the volume of the neighbors will change, before applying any impulses. Depending on the connectivity of neighbor n, its volume will change linear, quadratic or cubic, too. So the new neighbor volumes can be computed by $V^n(k_{\text{V}}) = \sum_{i=0}^{c} \beta_i k_{\text{V}}^i$, where $1 \leq c \leq 3$ denotes the connectivity. If $V^n(k_{\text{V}}) < \varepsilon$ for one neighbor, k_{V} has to be clamped, so that $V^n(k_{\text{V}}) \geq \varepsilon$ for all neighbors n. We tried several values for ε and got good results with $\varepsilon = \frac{V_0^n}{2}$. Using this constant we prevent too large correction impulses when large deformations occur.

The main problem with this correction technique is the fact, that under large deformations unnatural movements can still occur. Each particle has three degrees of freedom, but for n particles we have about $4n$ tetrahedrons, and thus $4n$ constraints. As the system has only $3n$ degrees of freedom, locking can occur during large deformations. This problem was also noticed in [ISF07] and we want to incorporate their idea to our simulation.

5. Volume constraint in a one-ring

As discussed in the last section, locking can occur during large deformations. The solution described in [ISF07] suggests to restore the initial volume in the one-ring surrounded by each particle. The one-ring of particle i is defined by all particles, which are connected over an edge with particle

i. Using one constraint per particle results in n constraints, giving more freedom to the system. Let the initial volume V_0^p of the one-ring surrounded by the particle p be the sum of the m initial volumes from the surrounded tetrahedrons. After applying external forces to the particles a preview of the particle positions for each constraint is made. The preview of the particle positions is important in order to correct the wrong volume from the external forces for the next simulation step immediately. Then the impulses $\mathbf{I}^i(k_V)$ for each particle i in the one-ring are computed. These impulses are the sum of the individual impulses for each tetrahedron as described in Section 4. Applying these impulses to each individual tetrahedron j in the one-ring would change the tetrahedron's volume by $\sum_{i=0}^{3} \alpha_{i,j} k_V^i$. Thus the overall volume change is $\sum_{j=1}^{m} \sum_{i=0}^{3} \alpha_{i,j} k_V^i = \sum_{i=0}^{3} \beta_i k_V^i$. Therefore, the root of $\sum_{i=0}^{3} \beta_i k_V^i - V_0^p = 0$ restores the volume in the one-ring of the particle p. Because we are using the impulses as presented in Section 4, particles surrounded by tetrahedrons with an unchanged volume stay fixed.

A drawback of restoring the volume in the one-ring is the fact, that the volume of one tetrahedron might shrink over time, while its volume is stored in another tetrahedron. To overcome this problem we can also adjust the individual volumes with the impulses computed in Section 4. To avoid the locking effects, we only correct with the coefficient γk_V. The experimentally chosen $\gamma = \frac{1}{8}$ gives good results.

Only applying the impulses for the particles in a one-ring once in a simulation step will not necessarily correct the whole volume. Surprisingly the volume error stays below 1% in most situations, even when large deformations occur. See Section 6 for some examples and for volume errors during the simulation.

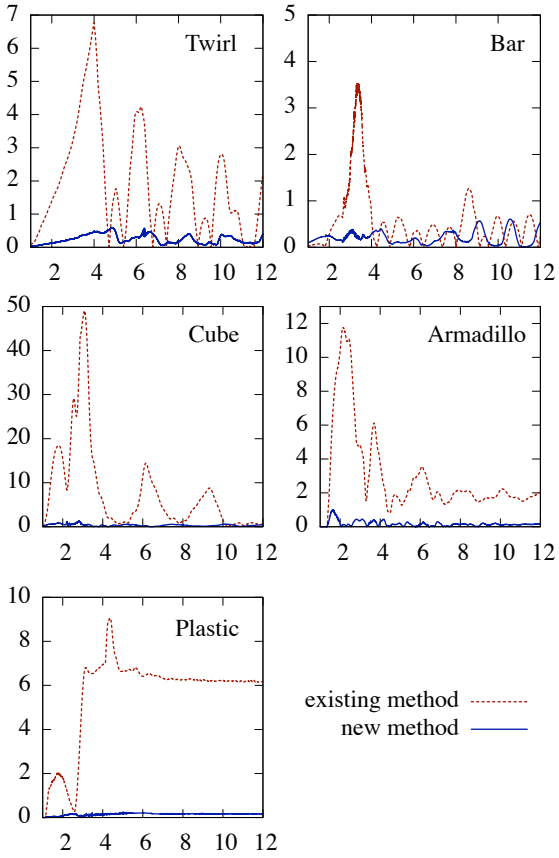

Figure 2: *The measured volume error in % (y-axis) over time in seconds (x-axis) for both methods in all examples.*

6. Results

In this section the existing approach from Section 3 is compared with our new method based on various examples. Figure 2 shows the volume error occurred during the simulation for each scenario. All tetrahedral models got assigned a high detailed triangle mesh for the visualization. Due to the fact, that the collision detection is currently based on the tetrahedrons, the triangle mesh does not collide with the ground in all scenarios. We used the parameter $k_V = 1.4$ for the existing method in all scenarios, while varying the distance stiffness k_D for the different examples to achieve different effects. The simulation was run with a fixed time step size of $h = 10$ms. While the existing method is faster to compute, it loses more volume under large deformations. On the other hand the new method is able to hold the volume error below 1% under most situations even when large deformations occur. Figures 3-7 show the different scenarios for the existing and new method taken at equal time stamps.

In our first scenario, shown in Figure 3, we took a bar with 4320 tetrahedrons and rotated the front particles resulting in a twirl of the bar. The distance stiffness was set to $k_D = 0.4$, which caused the bar to restore its original form when releasing the front particles. When the rotation reached its peak, the existing approach lost nearly 7% of its volume, while the volume error for the new approach was constantly below 1%. Due to the better restored volume, the simulation of the two methods are different when releasing the front particles. The existing approach took an average of 1.7 ms to compute the forces in one simulation step while our method needed 40.7 ms.

In our second scenario, shown in Figure 4, we dropped a rigid sphere into the same bar. The distance stiffness was set to $k_D = 0.8$ to get a stiffer look. When the sphere collided with the bar, the existing approach lost only up to 3.5% of its volume, because the higher distance stiffness forbade larger deformations. Nevertheless our new method was below the volume error of the existing one. The existing approach took an average of 1.8 ms to compute the forces in one simulation step while our method needed 40.8 ms.

The third example, shown in Figure 5, demonstrates the

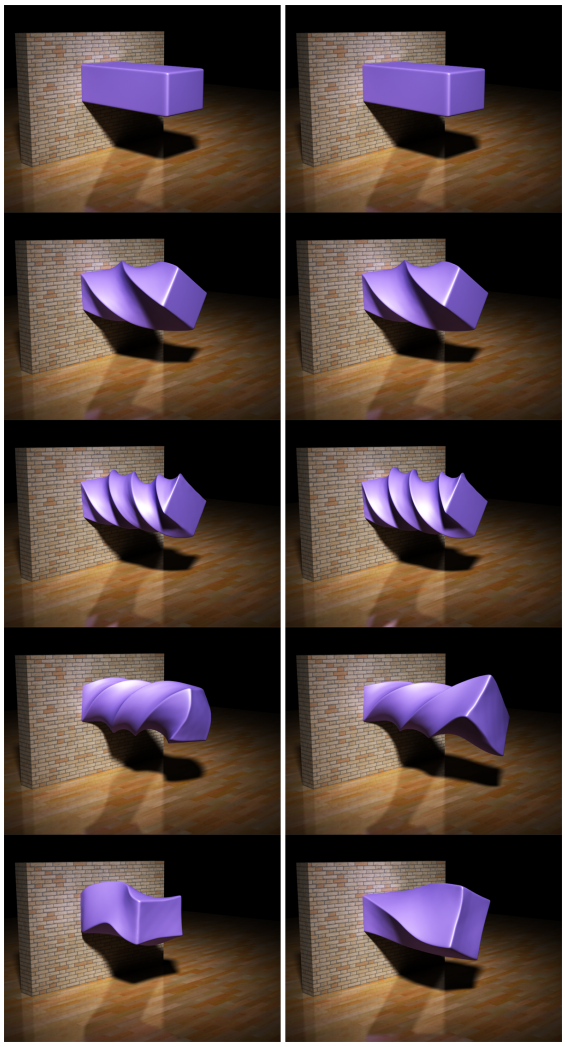

Figure 3: *While the visual difference is not that big between the existing method (left) and our new one (right), the different volume losses result in different movements.*

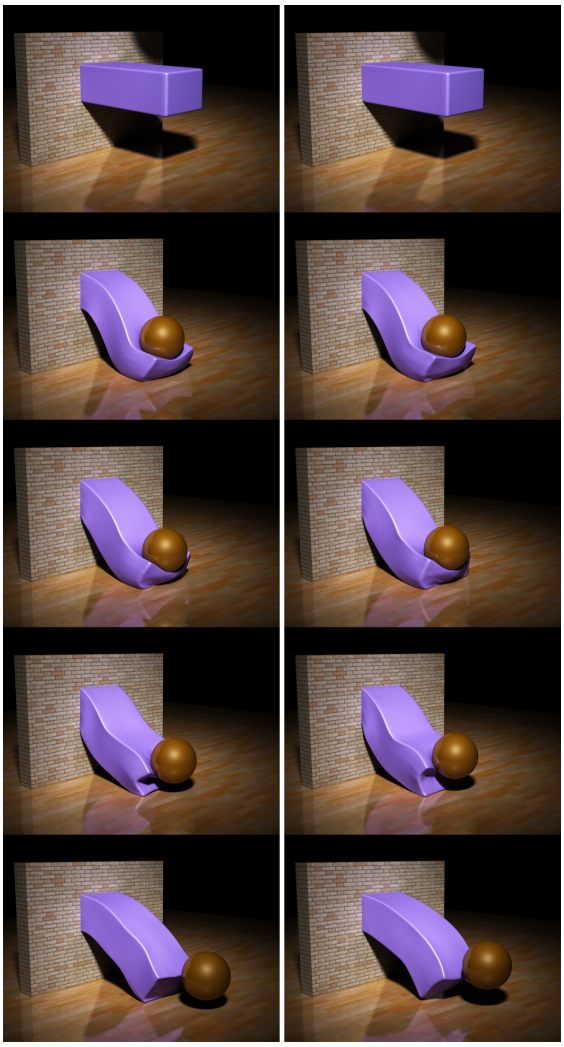

Figure 4: *Using a strong distance stiffness can prevent too much volume loss in the existing method (left). Still the new method (right) can better reduce the volume loss.*

effect of a highly deformable cube consisting of 8640 tetrahedrons. The distance stiffness was set to $k_D = 0.2$ to allow large deformations. When the heavy sphere collided with the cube, the existing approach lost nearly 50% of its volume, while our method was still able to reduce the error to below 1%. To compute the forces in one simulation step the existing approach took an average of 3.3 ms while ours needed 94.1 ms.

The fourth example, shown in Figure 6, presents a deformable Armadillo with 6510 tetrahedrons. Despite setting the distance stiffness to $k_D = 1.5$, the existing approach lost nearly 12% of its volume, when the feet contacted with the ground. Therefore, the Aramdillo could not jump up again, as it did with our new method. The existing approach took an average of 2.6 ms to compute the forces in one simulation step while our method needed 62.3 ms.

Our last example, which is shown in Figure 7, demonstrates the effect of plastic deformations, as presented in [THMG04,OBH02]. Because the plastic deformation is only applied to the distance constraints, we set the distance stiffness to $k_D = 8.0$. Due to the fact, that the springs do not return to its original form, the volume error of the existing approach stays at nearly 6%. In this scenario the existing approach took an average of 3.1 ms to compute the forces in one simulation step while our method needed 21.8 ms.

In our method the most computation time is spend for the

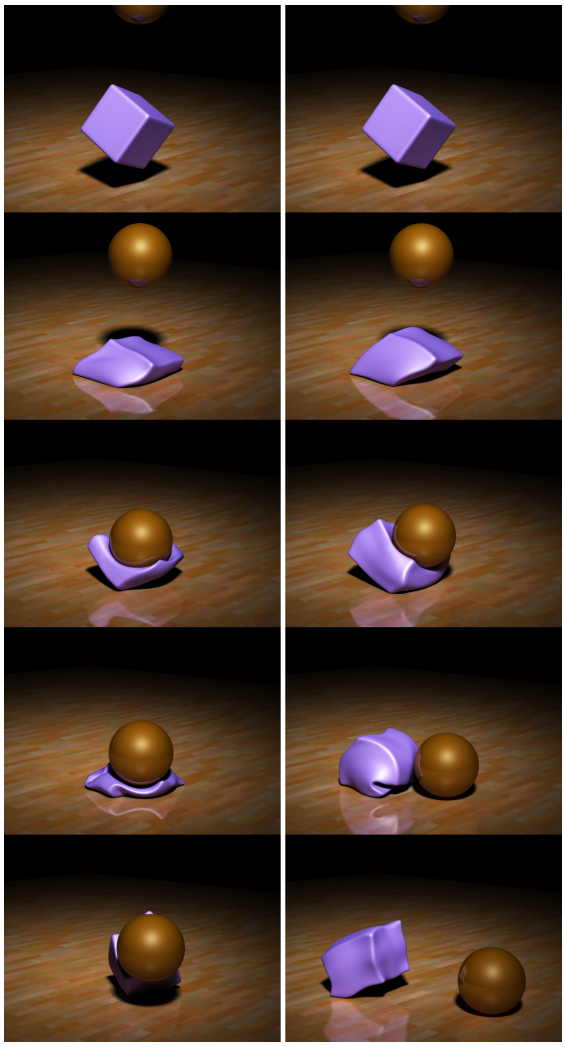

Figure 5: *Using a too low distance stiffness can result in a huge volume loss with the existing method (left), while the new method (right) prevents this situation.*

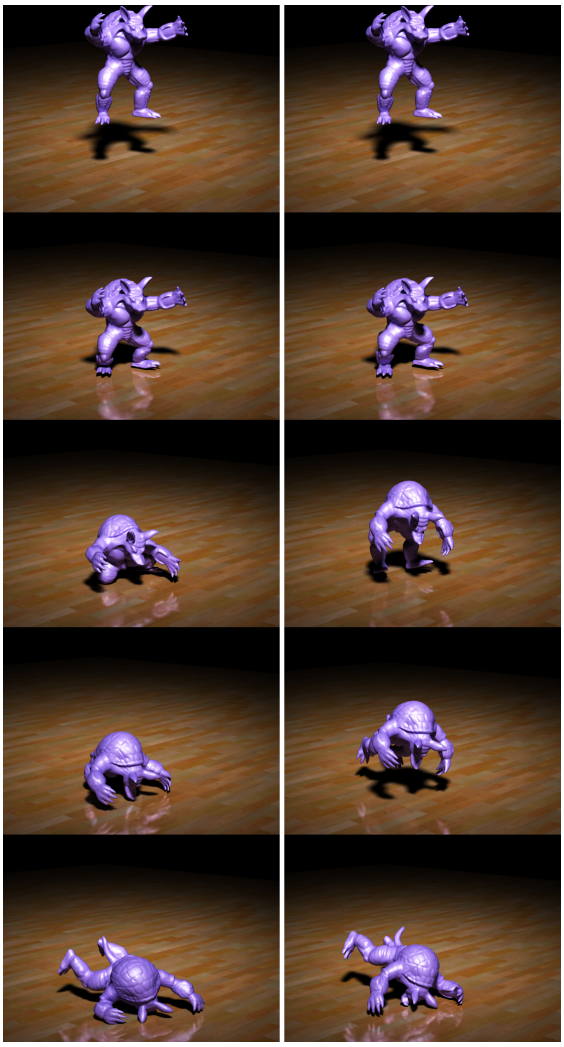

Figure 6: *With the existing method (left) the feet are getting compressed when hitting the ground, while in our method (right) the Armadillo jumps up again.*

k_V-correction of the neighbor tetrahedrons as described in Section 4. Due to the fact, that we scale the computed k_V with γ as described in Section 5, the problematic cases do not occur in all our test scenarios. When skipping the correction test, all examples run more than 50% faster. Without correcting k_V for the neighbors the cube example only needed an average of 31.2 ms to compute the forces in one simulation step.

7. Conclusion and future work

We have presented a new method for simulating deformable solids, which almost conserve the volume during the simulation. While we cannot guarantee an invariant volume, the volume error stays below 1% in all our scenarios with still a reasonable computation speed. Especially when high external forces are involved, or the distance stiffness is low and thus cannot restore the original form, the existing approach loses up to 50% of the volume, which is clearly visible during the simulation. Another advantage of the new approach is the fact, that the user does not need to find a good stiffness constant for the volume constraint, in order to get a nearly incompressible deformable object. Due to the fact that the simulation only needs a low resolution tetrahedron model, while a high detailed triangle model can be used for the visualization, we hope to be able to optimize the new approach to get real-time simulations. We are looking to parallelize the algorithm and make use of multi-resolution meshes to gain

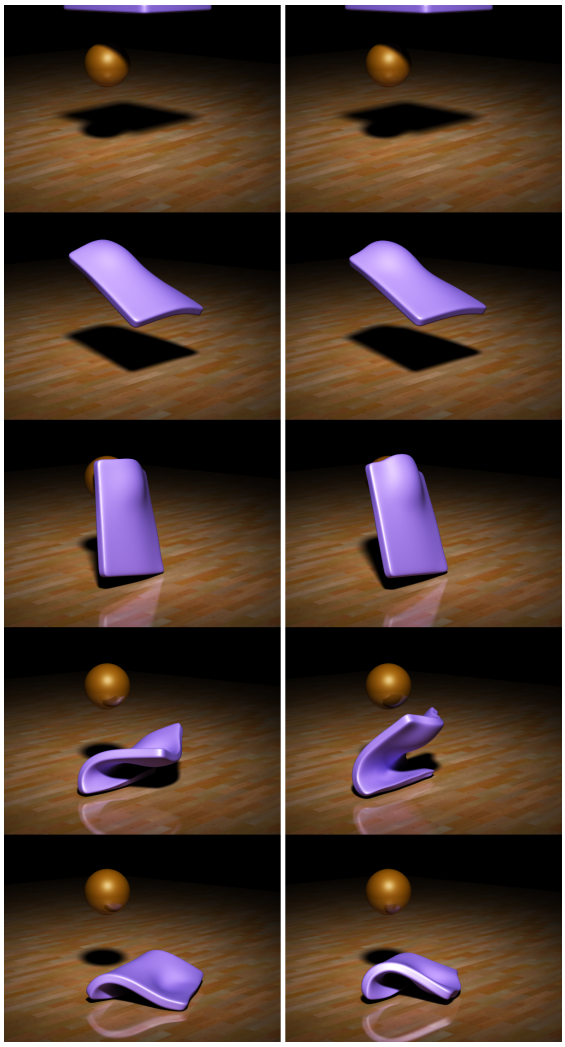

Figure 7: *Using plastic deformations the existing method (left) cannot completely restore the original volume, while the new one (right) makes it possible.*

performance. In the future we also want to incorporate a low resolution triangle model for a more accurate collision detection.

8. Acknowledgements

We would like to thank Dieter Finkenzeller for generating the wall and floor textures in our scenes and the Stanford Scanning Repository for the Armadillo.

References

[AOW*08] ADAMS B., OVSJANIKOV M., WAND M., SEIDEL H.-P., GUIBAS L. J.: Meshless modeling of deformable shapes and their motion. In *ACM SIGGRAPH/Eurographics SCA* (2008). 2

[Ben07] BENDER J.: Impulse-based dynamic simulation in linear time. *Computer Animation and Virtual Worlds* (2007), 225–233. 2, 3

[BIT09] BECKER M., IHMSEN M., TESCHNER M.: Corotated sph for deformable solids. In *Workshop on Natural Phenomena Proceedings of Eurographics* (2009), Galin E., Schneider J., (Eds.), pp. 27–34. 1

[DBB09] DIZIOL R., BENDER J., BAYER D.: Volume conserving simulation of deformable bodies. In *Short Paper Proceedings of Eurographics 2009* (2009), Alliez P., Magnor M., (Eds.), pp. 37–40. 2

[ISF07] IRVING G., SCHROEDER C., FEDKIW R.: Volume conserving finite element simulations of deformable models. In *Proceedings of SIGGRAPH* (2007), pp. 13:1–13:6. 1, 3

[LS07] LABELLE F., SHEWCHUK J. R.: Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics 26*, 3 (2007), 57. 2

[MDM*02] MÜLLER M., DORSEY J., MCMILLAN L., JAGNOW R., CUTLER B.: Stable real-time deformations. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), ACM, pp. 49–54. 1

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *J. Visual Communication and Image Representation 18*, 2 (2007), 109–118. 3

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Transactions on Graphics* (2005). 1

[OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics 21*, 3 (2002), 291–294. 1, 4

[RMSG*08] ROBINSON-MOSHER A., SHINAR T., GRETARSSON J., SU J., FEDKIW R.: Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics* (2008), 1–9. 1

[SGT09] SCHMEDDING R., GISSLER M., TESCHNER M.: Optimized damping for dynamic simulations. In *Proceedings of Spring Conference on Computer Graphics* (2009). 2

[ST08] SCHMEDDING R., TESCHNER M.: Inversion handling for stable deformable modeling. *The Visual Computer 24*, 7 (2008), 625–633. 3

[SWT06] SPILLMANN J., WAGNER M., TESCHNER M.: Robust tetrahedral meshing of triangle soups. In *Proceedings of Vision, Modeling, Visualization VMV'06* (2006), pp. 9–16. 2

[THMG04] TESCHNER M., HEIDELBERGER B., MÜLLER M., GROSS M.: A versatile and robust model for geometrically complex deformable solids. In *Proceedings of Computer Graphics International* (2004), pp. 312–319. 2, 4

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 205–214. 1

[Ver67] VERLET L.: Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev. 159*, 1 (Jul 1967), 98. 2