A Finite Element Method for Interactive Physically Based Shape Modelling with Quadratic Tetrahedra

Johannes Mezger, Bernhard Thomaszewski, Simon Pabst
and Wolfgang Straßer

email: mezger@gris.uni-tuebingen.de
thomaszewski@gris.uni-tuebingen.de
pabst@gris.uni-tuebingen.de
strasser@gris.uni-tuebingen.de

WSI-2007-01
July 2007

Graphisch-Interaktive Systeme
Wilhelm-Schickard-Institut
Universität Tübingen
D-72076 Tübingen, Germany
WWW: http://www.gris.uni-tuebingen.de

ISSN 0946-3852

A Finite Element Method for Interactive Physically Based Shape Modelling with Quadratic Tetrahedra

J. Mezger, B. Thomaszewski, S. Pabst and W. Straßer
Graphical-Interactive Systems (GRIS)
Wilhelm Schickard Institute for Computer Science, Tübingen, Germany

{mezger,thomaszewski,pabst,strasser}@gris.uni-tuebingen.de

Figure 1. Plastic deformation of a toy. On the second model the plastic strain is visualised. The two models on the right show the approximation with the quadratic finite elements.

Abstract

We present an alternative approach to standard geometric shape editing using physically-based simulation. With our technique, the user can deform complex objects in real-time. The enabling technology of this approach is a fast and accurate finite element implementation of an elasto-plastic material model, specifically designed for interactive shape manipulation. Using quadratic shape functions, we avoid the inherent drawback of volume locking exhibited by methods based on linear finite elements. The physical simulation uses a tetrahedral mesh, which is constructed from a coarser approximation of the detailed surface. Having computed a deformed state of the tetrahedral mesh, the deformation is transferred back to the high detail surface. This can be accomplished in an accurate and efficient way using the quadratic shape functions. In order to guarantee stability and real-time frame rates during the simulation, we cast the elasto-plastic problem into a linear formulation. For this purpose, we present a corotational formulation for quadratic finite elements. We demonstrate the versatility of our approach in interactive manipulation sessions and show that our animation system can be coupled with further physics-based animations like, e.g. fluids and cloth, in a bi-directional way.

CR Categories: I.3.5 [Computer Graphics]: Physically based modeling

Keywords: Mesh deformation, quadratic finite elements, plasticity

1 Introduction

With the advent of 3D data acquisition devices such as structured light scanners, highly detailed geometric surface models are now easily available. The last ten years have seen many approaches for editing such surfaces, all of which somehow strive to achieve globally smooth deformation while preserving surface details and shape volume. It is however remarkable that all of the recent methods depart from a *purely geometric view* of this problem leading to approaches which are rather detached from the actual problem. Instead of taking such a circuitous route, we propose to consider this problem as one of *elasto-plastic modelling*, which means resorting to physically-based simulation - a simple and direct approach.

The reason why this way of editing highly detailed surfaces has not been considered before is probably due to a common misconception. For physical simulation, mass-spring systems remain the most widely used technique in the computer graphics community. Although they allow for efficient implementations, it is a well known fact that they are inherently unable to reproduce even simple isotropic materials correctly and fail to preserve volume. In contrast, (higher order) finite elements excel at these challenges. However, it is commonly believed that finite elements interchangeably stand for high computation times. In this work, we show that even a highly accurate non-linear approach can run at interactive rates, thus overcoming any justification for unwieldy mass-spring models.

Our approach originates from the theory of elasto-plasticity and is thus completely based on physical deformation, allowing for the most intuitive interaction. The plasticity model accounts for permanent deformations, which we think approximates the real problem of shape manipulation best. We allow the user to interactively assign different elastic and plastic properties as well as constraints to selected regions of the object. In this way, material resilience and reversibility of deformations can be adapted to the current editing objectives. We emphasise that no artificial enforcing of volume preservation is needed since this property directly follows from the physical approach. Additionally, there is no need for explicitly distributing deformations induced by handles. Deformation automatically propagates through the body according to the forces applied by the user through different interaction tools. Due to the physical nature of our approach, the integration of shape manipulation within complex animations comes at no extra cost. As we show in our examples, the soft body simulation and manipulation can directly be combined with fluid or cloth simulation.

The work presented in this paper is not the first to address interactive physically-based simulation. For the sake of computational efficiency, existing FE-based approaches rely on linear finite elements, which suffer from volume locking. This greatly reduces their applicability in the context of shape editing, where volume preservation is highly important. In contrast, our approach uses quadratic finite elements, which excel at preserving volume and offer good accuracy even with a smaller number of elements. To guarantee stability and real-time frame rates during simulation, a linear problem formulation is indispensable (see Sec. 5).

The rest of this paper is organised as follows. The next section reviews previous work from related fields. Sec. 3 provides the mathematical and physical background for this work, while the fourth section explains our novel finite element approach. The subsequent section presents our implicit integration of elasto-plasticity. Results are presented in Sec. 7 and the paper concludes with a discussion and an outlook on future work.

2 Related Work

Geometric Mesh Editing The problem of mesh editing can most simply be formulated as finding ways to create globally smooth and visually pleasing deformations while preserving surface details. Disturbing artefacts like surface distortion or significant change in volume have to be avoided. As a further requirement, a practically useful shape deformation algorithm has to be fast enough to deliver real-time frame rates and must offer intuitive interaction facilities [BK04].

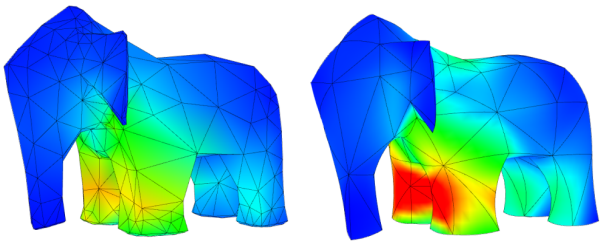
The first approaches relied on multi-resolution representations, decomposing a model into low frequency components and detail displacements [ZSS97, KCVS98, BK03]. A global deformation technique like free form deformation [SP86] is first applied to the coarse representation and the geometric detail is then transferred back to the deformed mesh. Because this manipulation technique is inherently local, artefacts are likely to occur in highly deformed regions.

A more recent approach to preserving surface details under global deformations is based on differential coordinates [Ale03]. In this context, detail preservation can be formulated as the minimization of an energy functional which is related to the change in differential coordinates after deformation [SLCO+04, LSCO+04, YZX+04]. The deformation (i.e. the editing objective) itself is incorporated as a set of positional constraints on the solution of the linear system arising from the minimization problem. Unfortunately, differential co-

ordinates are not rotation-invariant, which means that large rotational deformations lead to disturbing surface distortions. In the case of shape blending these rotations can be factored out locally [ACOL00]. However, for general shape editing the problem is significantly harder since the final state is not known in advance. In this context, pyramid coordinates [SK04] offer invariance under rigid body transformations but lead to a non-linear equation system with the associated computational and stability related problems. As an alternative, the rotation-invariant differential coordinates proposed by Lipman et al.[LSLCO05] only require the successive solution of two linear systems. However, surface and volume distortion still occurred for large deformations - a problem which was resolved in [LCOGL07] using a quasi-linear approach.

Another important deformation constraint is the preservation of volume [RSB96]. Combining differential coordinates with an approach for explicit volume preservation, Zhou et al. [ZHS+05] minimise both the change in surface details and shape volume. The latter was further improved in [HSL+06] by allowing for general non-linear constraints within a quasilinear approach. Lastly, the skeleton constraint [WG97, HSL+06] is also of interest when manipulating articulated shapes.

Real-time Physically-based Simulation Physically-based simulation of deformable objects in real-time has first been investigated in the context of general animation [JP99]. As another important application, virtual surgery simulation poses most stringent requirements on both speed and accuracy. The latter requirement has spurred the development of approaches based on continuum mechanics [PDA00] and finite elements [MDM+02, HGS03]. In order to reduce the computational complexity the problem is usually recast into a linear formulation using linear finite elements and a small strain measure coupled with methods for extracting rotations [MDM+02, HS04]. Unfortunately, linear finite elements are very susceptible to *locking* (see Fig. 2), which degrades accuracy substantially and lets soft objects appear overly rigid. This problem can be greatly alleviated using higher order basis functions as demonstrated in [MS06]. The latter approach, however, uses Newton iterations to handle geometric non-linearities and can therefore fail in finding a solution within a given period of time. Another problem that can be encountered during simulation is the inversion of tetrahedra due to excessive forces. A way to deal with this issue has been proposed in [ITF04].



(a) (b)

Figure 2. Visualisation of pressure in a soft toy exposed to gravity with 1200 linear (a) and 234 quadratic tetrahedra (b). Both simulations are real-time, but the linear model suffers from significant locking.

Plasticity The existing literature on mathematical and numerical plasticity is abundant and we refer the interested reader to [ZT00] and the references therein. In computer graphics, Terzopoulos et al. [TF88] were the first to incorporate plasticity and fracture effects into deformable object simulation. A simple linear elastic material with kinematic hardening was mapped to their computational framework combining viscoelastic spring-dashpot units and plastic slip units. O'Brien et al. [OBH02] resorted to the more accurate continuum-mechanics setting, using the von-Mises yield criterion with linear plasticity and a second elastic regime which limits the plastic strain. They used an explicit integration scheme, which greatly simplifies implementation but leads to high computation times and only conditional stability. In [MG04] Müller et al. extended their approach based on stiffness-warping to account for plasticity effects using a model similar to [OBH02]. However, since only finite tetrahedra with linear shape functions are used, curved surfaces are coarsely approximated, and nearly incompressible materials are likely to suffer from locking. For the plasticity model, we basically draw on the same idea and combine it with kinematic hardening and a prediction step for the elastic strain.

3 Background of Physical Soft Body Simulation

In this section we will briefly outline some mathematical and physical notions underlying our soft body simulator. We start with a short account of continuum mechanics.

3.1 Continuum Mechanics

In the most abstract view of continuum mechanics, there are three important concepts: the *strain* $\boldsymbol{\varepsilon}$, which is a dimensionless deformation measure, the *stress* $\boldsymbol{\sigma}$, which is a force per unit area, and a *material law* relating the two to each other as $\boldsymbol{\sigma} = \mathcal{C}(\boldsymbol{\varepsilon})$, where $\mathcal{C}$ is the elasticity tensor. For the simplest case of linear isotropic elasticity, this tensor has only two independent entries which are related to the well known Lamé constants λ and μ. Quantities in relation to the deformed state of the body (e.g. strain) are commonly expressed in terms of a fixed reference configuration $\Omega \subset \mathbb{R}$. The configuration mapping $\boldsymbol{\varphi} : \Omega \times [0, T]$ transforming material particles from their reference positions $\mathbf{x}^0$ to current positions $\mathbf{x}$ can be written as

$$\mathbf{x}(t) = \boldsymbol{\varphi}(\mathbf{x}^0, t) = \mathrm{id} + \mathbf{u}(\mathbf{x}^0, t) \ ,$$

where $\mathbf{u}$ is a displacement field from the initial configuration. For later use, we define the deformation gradient as

$$\nabla\boldsymbol{\varphi} = \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{x}^0} \ .$$

The strain energy of a deformed configuration is defined as

$$W = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\sigma}(\mathbf{u}) \ d\Omega \ ,$$

and the kinetic energy T is a function of the velocity of the object,

$$T = \int_\Omega \tfrac{1}{2} |\dot{\mathbf{u}}^2| \rho \ d\Omega \ ,$$

where ρ is the mass density. Including viscous stress contributions $\boldsymbol{\sigma}_\mathrm{v}$, the total energy $\Pi(\mathbf{u})$ follows as

$$\Pi(\mathbf{u}) = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\sigma}(\mathbf{u}) + \dot{\boldsymbol{\varepsilon}}(\dot{\mathbf{u}}) : \boldsymbol{\sigma}_\mathrm{v}(\dot{\mathbf{u}}) \ d\Omega + \tfrac{1}{2} |\ddot{\mathbf{u}}^2| \rho \ d\Omega \ .$$

Carrying out a variation of the above expression and taking into account external forces acting on the body, a partial differential equation (PDE) is obtained, which is the starting point for numerical discretisation.

3.2 Finite Element Discretisation

In order to approximate the solution of the aforementioned PDE, a subspace of finite dimension has to be defined. We employ isoparametric finite elements that discretise the problem in 3D space using the Ritz-Galerkin method, which is the favoured technique in structural mechanics.

The discretisation arises from a continuous partitioning of the domain into tetrahedra with locally defined basis functions $\mathbf{N}$ that interpolate the vertices $\mathbf{P}$ and their displacements $\mathbf{U}$, e.g.

$$\boldsymbol{\varphi}(\mathbf{x}) = \sum_{i=0}^{N-1} \mathbf{P}_i \mathbf{N}_i(\mathbf{x}) \tag{1}$$

with

$$\sum_{i=0}^{N-1} \mathbf{N}_i = 1, \quad \mathbf{N}_i(\mathbf{P}_j) = \delta_{ij}$$

and

$$\nabla\boldsymbol{\varphi}(\mathbf{x}) = \nabla\mathbf{u}(\mathbf{x}) + \mathbf{I} = \mathbf{U} \ \nabla\mathbf{N}(\mathbf{x}) + \mathbf{I}, \quad \mathbf{I} = \mathrm{diag}(1)_{3\times 3} \ .$$

This gives the ODE

$$\mathbf{F}(\mathbf{U}) + \mathbf{F}_\mathrm{v}(\dot{\mathbf{U}}) + \mathbf{M}\ddot{\mathbf{U}} = \mathbf{F}^\mathrm{ext} \tag{2}$$

with elastic forces $\mathbf{F}$ (Sec. 4.3), viscous forces $\mathbf{F}_\mathrm{v}(\dot{\mathbf{U}})$, dead external forces $\mathbf{F}^\mathrm{ext}$ and the mass matrix

$$\mathbf{M} = \int_V \rho \ \mathbf{N}\mathbf{N}^T \ dV \ . \tag{3}$$

The accuracy of this approximation strongly depends on the choice of the shape functions $\mathbf{N}$ and the size h of the elements. In engineering applications it is usually avoided to use linear basis functions as they achieve a convergence which is only linear in $1/h$. This weak convergence is caused by $\nabla\boldsymbol{\varphi}$ being constant and consequently also $\mathbf{F}$ being constant on the whole element.

Especially in the case of almost incompressible materials the linear elements suffer from numerical locking effects, i.e. solving (2) results in significantly smaller displacements than expected (Fig. 2). Moreover, many small elements have to be placed at the object boundaries in order to approximate irregular shapes. The approach presented in the following exploits the benefits of quadratic basis functions regarding the demands of interactive shape deformation, namely good approximation of shape and fast convergence in the presence of plastic, nearly incompressible materials.

4 Real-time Soft Body Simulation

We propose two major improvements to current techniques for achieving both the objective of real-time simulation and the robust handling of substantial deformations, which are inherent to soft bodies:

- First, real-time is realised by a guaranteed upper bound on the computational costs per frame, while the number of elements is reduced to a minimum.

- Second, large deformations are managed using non-linear shape functions, and taking non-linear strain into account.

Implicit time integration with arbitrarily large step sizes (Sec. 5.2) and a linear system of equations to be solved at each time step serve for both objectives.

4.1 Quadratic Shape Functions

Representing the N shape functions in the general form

$$\mathbf{N}_i(\mathbf{x}) = \sum_{j=0}^{N-1} \alpha_{ij}\ \mathbf{x}_1^{e_{j1}}\mathbf{x}_2^{e_{j2}}\mathbf{x}_3^{e_{j3}}, \qquad i = 0..N-1, \quad 4 \le N \le 10\ ,$$

with quadratic exponents $e_{jk} = 0..2$, the conditions (1) define a linear system with N^2 equations that is solved for the shape coefficients α. This is performed once for the (unstressed) reference state of the object, storing the shape coefficients for later use.

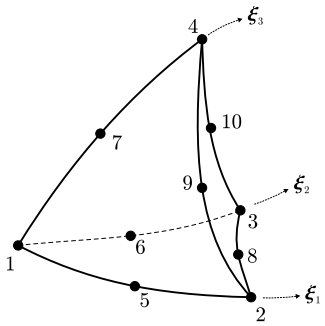


Figure 3. Curved quadratic 10-node tetrahedron with curvilinear coordinates.

The number of nodes and shape functions respectively can be chosen arbitrarily from four up to ten. For any $N > 4$ additional nodes are placed on the edges of the (linear) standard tetrahedron (Fig. 3). We construct the nodal positions $\mathbf{P}^0$ of the reference state as depicted later in Sec. 6.

4.2 Volume Integration

Choosing $N = 10$, the $\mathbf{N}_i$ are complete quadratic polynomials and the error of this Galerkin approximation is bounded to $O(h^3)$ [ZT00]. It is crucial to preserve this quadratic convergence by accurately integrating the matrices of (2) over the volume of the tetrahedron. This is achieved by a four point Gauss-Legendre cubature at the curvilinear coordinates

$$\boldsymbol{\zeta}_1 = \begin{bmatrix} r \\ r \\ r \end{bmatrix},\ \boldsymbol{\zeta}_2 = \begin{bmatrix} s \\ r \\ r \end{bmatrix},\ \boldsymbol{\zeta}_3 = \begin{bmatrix} r \\ s \\ r \end{bmatrix},\ \boldsymbol{\zeta}_4 = \begin{bmatrix} r \\ r \\ s \end{bmatrix}$$

with $r = \frac{1}{4} - \frac{1}{20}\sqrt{5}$ and $s = \frac{1}{4} + \frac{3}{20}\sqrt{5}$ [Str71] .

Thus e.g. the consistent mass matrix (3) of a tetrahedron with vertices $\mathbf{P}^0$ is precomputed using

$$\mathbf{M} = \int_0^1 \int_0^{(1-\boldsymbol{\xi}_1)} \int_0^{(1-\boldsymbol{\xi}_1-\boldsymbol{\xi}_2)} \det(\mathbf{P}^0\ \nabla\hat{\mathbf{N}})\ \rho\ \hat{\mathbf{N}}\hat{\mathbf{N}}^T d\boldsymbol{\xi}_3 d\boldsymbol{\xi}_2 d\boldsymbol{\xi}_1$$

$$\approx \frac{1}{24}\sum_{i=1}^{4} \det(\mathbf{P}^0\ \nabla\hat{\mathbf{N}})\ \rho\ \hat{\mathbf{N}}(\boldsymbol{\zeta}_i)\ \hat{\mathbf{N}}(\boldsymbol{\zeta}_i)^T\ . \qquad (4)$$

Here the shape functions $\hat{\mathbf{N}}$ interpolate the unit tetrahedron with volume 1/6.

4.3 Corotated Quadratic Tetrahedra

The elastic forces $\mathbf{F}(\mathbf{U})$ generally depend non-linearly on $\mathbf{U}$ due to the geometric non-linearity of the strain tensor even if $\mathcal{C}$ is linear. Simply using the linear Cauchy strain tensor

$$\boldsymbol{\varepsilon}^{\mathrm{C}} = \tfrac{1}{2}(\nabla\boldsymbol{\varphi} + \nabla\boldsymbol{\varphi}^T) - \mathbf{I} \qquad (5)$$

does not produce satisfying results as soon as significant deformations occur. A corotational formulation [Fel00, HS04] linearises (2) by first applying element-wise rotations $\mathbf{R}$ to the displacement vector and then solving the linear system

$$\mathbf{F}(\mathbf{R}\mathbf{U})\ \mathbf{R}^T + \mathbf{F}_{\mathrm{v}}(\dot{\mathbf{U}}) + \mathbf{M}\ddot{\mathbf{U}} = \mathbf{F}^{\mathrm{ext}}\ .$$

Unfortunately, a single rotation matrix $\mathbf{R}$ is not enough to rotate a quadratic tetrahedron into a configuration that leaves a rotation-free deformation gradient $\mathbf{R}\nabla\boldsymbol{\varphi}$. Hence, we apply a separate polar factorization of $\nabla\boldsymbol{\varphi}$ at each cubature point to obtain the corotated strain tensor

$$\varepsilon^{\mathrm{CR}} = \tfrac{1}{2}(\mathbf{R}\nabla\boldsymbol{\varphi} + \nabla\boldsymbol{\varphi}^T\mathbf{R}^T) - \mathbf{I} \qquad (6)$$

and the rotation-invariant stress tensor $\boldsymbol{\sigma}^{\mathrm{CR}} = \mathcal{C}(\boldsymbol{\varepsilon}^{\mathrm{CR}})\mathbf{R}^T$. The linear elastic forces at the cubature points simply become

$$\mathbf{F}(\mathbf{R}\mathbf{U})\ \mathbf{R}^T = \nabla\mathbf{N}\ \boldsymbol{\sigma}^{\mathrm{CR}}$$

and are calculated efficiently with precomputed $\nabla\mathbf{N}$. Furthermore, because of the constant gradient $\boldsymbol{\varepsilon}_{\mathrm{U}}^{\mathrm{CR}}$ the element stiffness matrix

$$\mathbf{F}_{\mathrm{U}} = \frac{\partial\mathbf{F}}{\partial\mathbf{U}} = \mathbf{R}\ \mathbf{F}_{\mathrm{U}}^{\mathrm{C}}\ \mathbf{R}^T\ ,$$

depends only on the current rotation $\mathbf{R}$ and a constant matrix $\mathbf{F}^{\mathrm{C}}_{,\mathrm{U}}$.

For the viscous forces no corotation is applied. Instead, simple linear damping is achieved with the time-derivative of the Cauchy strain tensor using

$$\mathbf{F}_{\mathrm{v}}(\dot{\mathbf{U}}) = \nabla\mathbf{N}\, \mathcal{D}(\dot{\boldsymbol{\varepsilon}}^{\mathrm{C}}) \ ,$$

which is sufficient for shape editing purposes.

4.4 Lazy Corotation

A significant speed-up in the solution of the ODE (2) is achieved by a "lazy" update of the rotation matrices. It is motivated by the observation that small changes of the stiffness matrix lead to imperceptible changes of the static equilibrium and that slightly deferred revaluations of the stiffness matrix are not noticeable at all in a dynamic simulation.

We roughly estimate the change of the corotation $\mathbf{R}$ of an element between the last evaluation at time t_1 and the current time t by means of the maximum absolute row sum norm

$$d(t) = \|\nabla\boldsymbol{\varphi}(t) - \nabla\boldsymbol{\varphi}(t_1)\|_{\infty}$$

with respect to the difference of the two deformation gradients. This expression can be evaluated efficiently, and the polar factorization is not necessary in order to determine whether the rotation changed significantly. Empirically, the stiffness matrix should be recomputed if $d(t)$ exceeds a tolerance value of 0.1 in at least one cubature point.

4.5 Guaranteed Framerate

Obviously, any jerking of the application is annoying in interactive physically-based modelling. Jerky frame rates are produced by explicit time integration with adaptive time stepping when the step-size has to be decreased due to increasing forces. In contrast, with implicit time integration the step-size for any damped problem can be kept constant while the solution stays stable.

In order to safely limit the computation time of an arbitrarily large time step, we solve the corotated problem with a direct linear solver. In the worst case, the simulation will require recomputing the corotations, refactoring the system matrix and solving the linear system. With our implementation, current CPU cores perform this task within 40 milliseconds for more than 1500 linear tetrahedra or more than 300 quadratic tetrahedra, allowing a frame rate of 25Hz. The lazy corotation further serves for reducing the CPU load, but of course does not increase the lower bound of the frame rate.

5 Implicit Time Integration of Elasto-Plastic Material

The constitutive law addressed in the previous sections leads to material behaviour independent of the deformation history (also called a hyperelastic material). Once the loading is removed the deformation will (possibly delayed by viscous effects) recover a state of zero deformation. This assumption of ideal elasticity is only a rough approximation and real world materials do not obey this model. In fact, every solid material will fail, i.e. undergo irreversible deformation or even fracture, if the applied loading exceeds a certain threshold. The effect of irreversible deformation actually is the most general definition of *plasticity*, which we will use as the basic mechanism for conveying permanent shape deformation in the following.

5.1 Plasticity and Hardening

In order to extend the elastic model to account for plasticity effects we first introduce the decomposition of the total strain $\boldsymbol{\varepsilon}^{\mathrm{tot}}$ as

$$\boldsymbol{\varepsilon}^{\mathrm{tot}} = \boldsymbol{\varepsilon}^{\mathrm{el}} + \boldsymbol{\varepsilon}^{\mathrm{pl}} \ .$$

The total strain can be interpreted as the true geometric strain, which is readily evaluated using the finite element approximation (see Eq. (6)). As a result, the elastic stress can always be expressed as

$$\boldsymbol{\sigma} = \mathcal{C} : \boldsymbol{\varepsilon}^{\mathrm{el}} = \mathcal{C} : (\boldsymbol{\varepsilon}^{\mathrm{tot}} - \boldsymbol{\varepsilon}^{\mathrm{pl}}) \ . \tag{7}$$

We will generally assume that the material behaves ideally elastic up to a certain point of stress where the plastic deformation regime begins. Using a yield function F, this criterion can be expressed as $F(\boldsymbol{\sigma}) = 0$ which, depending on the current state of stress, indicates whether plastic deformation occurs or not. Similar to [OBH02] we will restrict our considerations to an isotropic von-Mises yielding model, which is particularly simple. In this case F does not depend on the hydrostatic (i.e. volumetric) part of the stress tensor and, hence, plastic deformation does not affect the volume. As a consequence, e.g. twisting of a mesh will not result in unrealistic loss of volume - an important aspect which is hard to achieve with previous surface based shape editing methods.

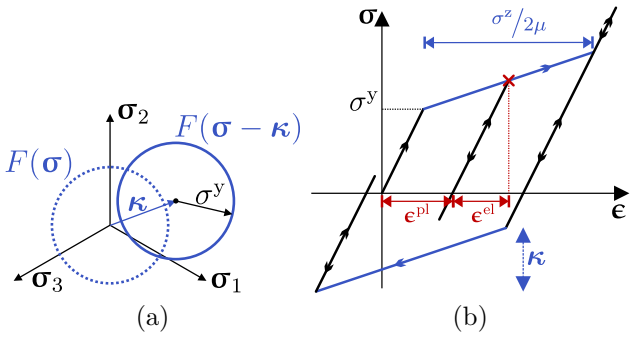


Figure 4. With kinematic hardening the yield surface F is allowed to translate by the backstress κ (a). In the exemplary uni-axial loading cycle (b) the sense of traversal is indicated by arrows. Unloading is always elastic and reveals the stored plastic strain when the stress vanishes.

The condition $F = 0$ can best be pictured as an implicit (yield) surface in stress space (circles in Fig. 4a), where the radius of the surface is a material property. Inside the yield surface the material behaves entirely elastic. Once the elastic stress reaches the yield surface, it cannot further increase and any additional deformation will result in plastic deformation. In the simplest model, the location of the surface (i.e. the centre and radius) stays fixed and as a consequence the plastic strain will increase while the elastic stress stays constant. Because this is impractical for our application, we include the effect of kinematic hardening. Here, the centre of the yield surface is allowed to move along the direction of the deviatoric strain.

With the linear kinematic hardening, a resilient plastic strain can be cancelled by a corresponding deformation in the opposite direction, leading to the stress-strain relationship depicted in Fig. 4b. The slope in the plastic phase is subject to

$$\dot{\boldsymbol{\kappa}} = H\dot{\boldsymbol{\varepsilon}}^{\mathrm{pl}}$$

with the kinematic hardening factor H. However, it would still be possible to achieve arbitrarily large plastic deformation which turned out to be inconvenient for the user. For this reason we limit the range of plastic deformation by another, user-defined threshold, producing the rightmost branch of the curve. Beyond, purely elastic behaviour is regained. The curve corresponds to a rate independent material, i.e. effects due to viscosity are not considered. Taking into account viscous stress contributions (4.3) the sharp transitions are actually smoothed according to the strain rate $\dot{\boldsymbol{\varepsilon}}^{\mathrm{el}}$.

5.2 Time Integration

With an explicit integration scheme, the time stepping of Eq. (7) is straightforward, since it requires only quantities from the current state, which are trivially known. Using implicit integration is more involved because the unknown strains $\boldsymbol{\varepsilon}^{\mathrm{tot}}(t+\Delta t)$ and $\boldsymbol{\varepsilon}^{\mathrm{pl}}(t+\Delta t)$ are required.

We deal with this issue using a return map algorithm similar to [AdV03]. Assuming that the time step will be entirely elastic, we predict the total strain $\boldsymbol{\varepsilon}^{\mathrm{tot}}(t+\Delta t)$ in an explicit manner using the current strain rate as

$$\tilde{\boldsymbol{\varepsilon}}^{\mathrm{tot}} = \boldsymbol{\varepsilon}^{\mathrm{tot}}(t) + \Delta t\, \dot{\boldsymbol{\varepsilon}}^{\mathrm{tot}}(t)\ , \tag{8}$$

where the tilde denotes trial quantities. The predicted elastic strain is computed as

$$\tilde{\boldsymbol{\varepsilon}}^{\mathrm{el}} = \tilde{\boldsymbol{\varepsilon}}^{\mathrm{tot}} - \boldsymbol{\varepsilon}^{\mathrm{pl}}(t)\ ,$$

and the deviatoric strain follows as

$$\tilde{\boldsymbol{\varepsilon}}^{\mathrm{dev}} = \tilde{\boldsymbol{\varepsilon}}^{\mathrm{el}} - \tfrac{1}{3}\,\mathrm{tr}(\tilde{\boldsymbol{\varepsilon}}^{\mathrm{el}})\,\mathbf{I}\ .$$

If we further assume that the plastic strain $\boldsymbol{\varepsilon}^{\mathrm{pl}}$ remains constant we can evaluate the yield function,

$$F(\mathcal{C} : \tilde{\varepsilon}^{\mathrm{dev}}) = ||2\mu\, \tilde{\boldsymbol{\varepsilon}}^{\mathrm{dev}} - \boldsymbol{\kappa}(t)|| - \sigma^{\mathrm{y}} = 0\ ,$$

where σ^{y} is the yield stress (cf. Fig. 4). If the yield function signals that in the next step there will be no transition to the plastic range, the assumption holds and we can safely use the standard implicit formulation to integrate the elastic forces. Otherwise, the plastic strain for the end of the time step is computed as

$$\boldsymbol{\varepsilon}^{\mathrm{pl}}(t+\Delta t) = \boldsymbol{\varepsilon}^{\mathrm{pl}}(t) + \lambda \frac{2\mu\, \tilde{\boldsymbol{\varepsilon}}^{\mathrm{dev}} - \boldsymbol{\kappa}(t)}{||2\mu\, \tilde{\boldsymbol{\varepsilon}}^{\mathrm{dev}} - \boldsymbol{\kappa}(t)||}$$

The consistency parameter λ ensures that the yield condition is met after the time step and reads

$$\lambda = \frac{||2\mu\, \tilde{\boldsymbol{\varepsilon}}^{\mathrm{dev}} - \boldsymbol{\kappa}(t)|| - \sigma^{\mathrm{y}}}{2\mu\, \tilde{\boldsymbol{\varepsilon}}^{\mathrm{dev}} + H}\ .$$

To limit the norm of the plastic stress components by σ^{z}, the plastic strain is clamped to ${}^{\sigma^{\mathrm{z}}}/_{2\mu}$ (cf. O'Brien et al. [OBH02]). Finally, we have to compute the new backstress using

$$\boldsymbol{\kappa}(t+\Delta t) = H \cdot \left(\boldsymbol{\varepsilon}^{\mathrm{pl}}(t+\Delta t) - \boldsymbol{\varepsilon}^{\mathrm{pl}}(t)\right)\ .$$

Clearly, the method described above is only an approximation which would not be of sufficient accuracy for engineering applications. However, it is more than sufficient for computer graphics purposes and leads to considerable computational advantages. Similar to

Müller et al. [MG04] we obtain a linear system of equations which, for the sake of simplicity, we abbreviate as

$$\left(\mathbf{M} - \mathbf{R}\,\nabla\boldsymbol{\mathcal{F}}\,\mathbf{R}^T\right)\mathbf{Y} = \mathbf{F}^{\text{const}}\ ,$$

where the state vector $\mathbf{Y}$ is the concatenation of nodal positions and velocities. Because of the prediction step (8) the plastic strain does not depend on $\mathbf{Y}$ and can be considered as a dead load during the time step. Hence, we are able to keep the stiffness matrix $\nabla\boldsymbol{\mathcal{F}}$ constant over time. Note that in contrast to explicit time integration schemes the mass matrix $\mathbf{M}$ does not have to be inverted and we can use the non-diagonal consistent mass matrix (4) without computational drawbacks.

5.3 Annealing

In the course of repeated deformation and sculpting by the user, substantial plastic strains can accumulate. While the simulation always remains stable, very large deformations are likely to degrade computational efficiency and accuracy. This problem can be avoided by *annealing* the solid from time to time. When the user stops deforming the object for a moment, its rest state is recomputed from the current deformation and the geometric as well as the plastic strains are reset. This procedure can be carried out as a background operation without the user taking notice of it, or, as shown in the accompanying video, by manually clicking a button to "commit" the current plastic state and to continue with further manipulations.

6 Geometric Model Reduction and Detail Preservation

In order to provide the user with as much freedom as possible we do not make specific assumptions on the size and resolution of the input model. We do, however, assume a closed manifold surface mesh of the object to be deformed. Such meshes can be obtained, e.g. using a geometry acquisition device like a structured light scanner and a subsequent post-processing step (i.e. reconstruction and triangulation). The resolution of meshes obtained in this way is usually very high. Since globally smooth deformations can be captured on a much coarser level it is common practice to treat high frequency surface details apart from possibly large low frequency deformations.

6.1 Geometric Model Reduction

We adopt this strategy and combine geometric model reduction techniques with an elegant detail preservation algorithm which arises in a natural way from our finite element approach. Low resolution tetrahedral meshes like the ones in Fig. 5 are created following a three-step algorithm:

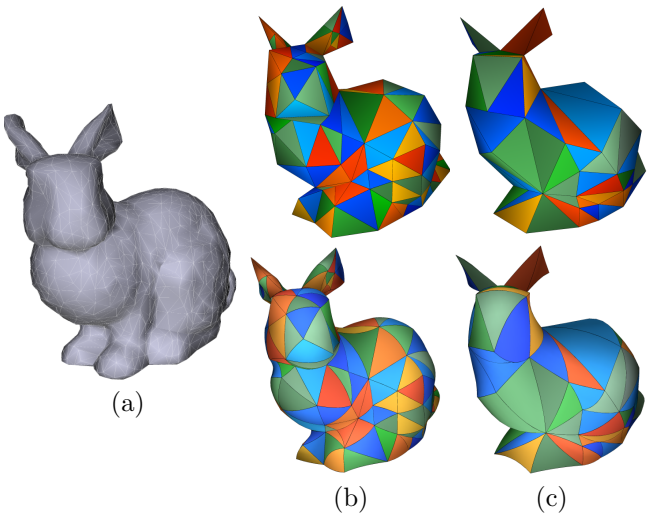


Figure 5. Automatic generation of quadratic tetrahedral meshes in two resolutions (b, c) from a detailed triangle mesh (a). The top row shows the result of the surface mesh simplification, the bottom row the boundary of the conformed quadratic FE meshes.

Coarse Surface Generation For the initial mesh simplification we employ standard triangle mesh reduction techniques available for polygonal modelling software, producing approximate Delaunay triangulations. The user defines by vertex painting which regions should keep higher resolutions, e.g. ears and tail of the armadillo.

Linear Tetrahedral Meshing The Delaunay property alleviates the subsequent generation of tetrahedral volume meshes, which is achieved using standard mesh generators. In our experience they often fail in generating coarse quadratic meshes for a given smooth surface. For this reason, a standard meshing first produces an FE model with linear 4-node tetrahedra.

Quadratic Surface Conforming Afterwards each tetrahedron of the coarse tetrahedral mesh is completed by the missing nodes on the six edges. Now the surface nodes are adjusted to lie on the initial high resolution surface (Fig. 6a). For this purpose, the closest surface face in the normal direction is found. If its

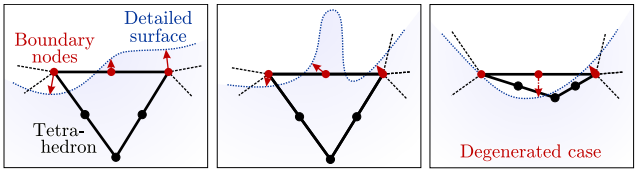


Figure 6. In the conforming step the surface nodes of the quadratic tetrahedral mesh are adjusted to lie on the detailed surface. (Simplified 2D drawing.)

distance exceeds a specific limit or if the surfaces turn out to be too spiky to determine a consistent normal direction, the surrounding area is searched for a closer face (Fig. 6b). Care is taken not to invert surface elements if the detailed surface lies inwards (Fig. 6c). In such rare cases the node must not be moved and it is advisable to increase the FE mesh resolution in the critical region.

While the whole reduction algorithm is rather simple, it turned out to be extremely effective and does not demand any FE modelling knowledge from the user. Curved element edges are preliminarily created at the FE mesh boundaries, but do of course also emerge in the inside during deformations. Hence, the full degrees of freedom of the quadratic tetrahedra are exploited by the simulation.

6.2 Detail Preservation

Since the plastic deformations are computed on the coarse FE mesh, the simulation does affect surface details. Instead, by interpolating the detailed surface with the isoparametric shape functions of the associated tetrahedron, the surface deforms realistically and the details inherit the affine invariance of the FE approach. That is, at any time t by

$$\mathbf{S}^t = \boldsymbol{\varphi}(\mathbf{S}^0, t) = \sum_{i=0}^{N-1} \mathbf{P}_i^t \, \mathbf{N}_i(\mathbf{S}^0) \ .$$

the interpolated detailed surface point $\mathbf{S}^t$ is calculated from the initial position $\mathbf{S}^0$ of the vertex. Cracks at the transitions from one quadratic tetrahedron to the next are automatically avoided because the FE surface nodes lie on the detailed surface, which is further interpolated smoothly by the curved edges. During annealing the rest positions $\mathbf{S}^0$ are reset to the current coordinates.

The colours $c(\mathbf{x}, t)$ for stress visualisations on the detailed surface first are linearly extrapolated from the cubature points to the nodes $\mathbf{P}_i^0$ of the reference state. Afterwards the colours $c(\mathbf{S}^t, t)$ for the detailed surface are interpolated likewise by the shape functions using

$$c(\mathbf{S}^t, t) = \sum_{i=0}^{N-1} c(\mathbf{P}_i^0, t) \, \mathbf{N}_i(\mathbf{S}^0) \ .$$

This is an intuitive and computationally efficient way of stress visualisation, which, to our knowledge, was not addressed in literature so far.

7 Results

The described techniques were applied successfully to perform several shape editing tasks on a Dual Xeon 5140, where only one core was used by our implementation. An implicit second order BDF (backward differentiation formula) solver ensured stable time integration. The linear system was solved with a direct sparse solver, which performed better than the conjugate gradient method due to the comparably small system matrices.

7.1 Examples

All examples were created in interactive sessions with a time step size of 40 milliseconds. The ability of the simulation to preserve volume and to correctly handle situations of extremely large deformations can be observed in Figs. 7 and 8. The surface meshes are coloured based on the norm of the plastic strain, red denoting that the plastic stress component is close to the limit σ^z. To the right hand side the surface of the tetrahedral meshes is visualised. In Fig. 1 it becomes evident that a complex surface (60.000 vertices) with rich features is interpolated smoothly by the quadratic basis functions, even when a low-resolution FE mesh (229 tetrahedra) is used. Fig. 11 proves that the volume is preserved even when large deformations are applied to a very detailed model.

A bi-directional integration with other physics based animation techniques, namely a grid-based fluid simulation and a cloth simulation based on finite elements (Fig. 9), is possible in a straightforward manner. This provides the animator with the opportunity to model complex environments which would be impossible to animate in the traditional way. Furthermore, the accompanying video shows interactive changes to elastic and plastic parameters to support a variety of shape manipulation tasks ranging from large-scale deformations to tweaks of fine features.

Surface mesh	Tetra-hedra	T_{mat}	T_{fac}	T_{solve}	T_{def}	T_{tot}
Arm 20k	146	10.1	3.9	2.5	1.9	18.4
Arm 60k	”	”	”	”	5.3	21.8
Arm 20k	229	16.6	6.1	3.5	1.7	27.9
Arm 60k	”	”	”	”	5.6	31.8
Dragon 20k	173	12.6	3.3	2.7	1.9	20.6
Dragon 50k	”	”	”	”	4.4	23.0
Dragon 100k	”	”	”	”	8.7	27.4

Table 1. Computation times in milliseconds for the armadillo and the dragon model in different resolutions up to 100.000 vertices. Between 30 and 50 frames per second are achieved.

7.2 Benchmarks

Table 1 shows the computation times for two exemplary deformation tasks: repeatedly dragging one arm of the armadillo (Fig. 1) up and down, and stretching and bending the upper part of the dragon model (Fig. 10). For the armadillo model, surface meshes with 20.000 and 60.000 vertices, and volume meshes with 146 and 229 quadratic tetrahedra were used. The dragon model is discretised with 173 quadratic tetrahedra and consists of up to 100.000 vertices. Computation times per frame (not considering lazy corotation) are separated into corotation with stiffness matrix update (T_{mat}), matrix re-factorisation (T_{fac}), solution of the linear system (T_{solve}) and update time of the interpolated surface mesh (T_{def}). From the total time (T_{tot}) it is evident that time-step sizes from 32ms down to 19ms could be used and still real-time simulation would be achieved. Additionally, in more than half of the frames lazy corotation was active, leaving only T_{solve} and T_{def} for the respective frames, and therefore significantly reducing the average computation time per frame. Actually in our experiments the frame rate was limited mostly by the rendering system, which acted as a bottleneck when huge surface meshes were used.

7.3 Future Work

Although our approach does not directly support skeleton driven deformations, similar effects can be achieved by simply changing local material properties in an appropriate way: assigning a stiff material to the limbs and a comparably soft one to the joints yields the desired behaviour. However, in future work we hope to extend our system to account for totally rigid regions and with a more convenient user interface for specifying skeleton constraints like in the work of [ZHS+05].

References

[ACOL00] Alexa, M., Cohen-Or, D., and Levin, D. As-Rigid-As-Possible Shape Interpolation. In *Proceedings of ACM SIGGRAPH*, 157–164, 2000.

[AdV03] Auricchio, F., and da Veiga, L. B. On a new integration scheme for von-Mises plasticity with linear hardening. *International Journal for Numerical Methods in Engineering*, 56(10):1375–1396, 2003.

[Ale03] Alexa, M. Differential coordinates for mesh morphing and deformation. *The Visual Computer*, 19(2):105–114, 2003.

[BK03] Botsch, M., and Kobbelt, L. Multiresolution Surface Representation Based on Displacement Volumes. *Computer Graphics Forum*, 22(3):483–483, 2003.

[BK04] Botsch, M., and Kobbelt, L. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, 23(3):630–634, 2004.

[Fel00] Felippa, C. A. A Systematic Approach to the Element-Independent Corotational Dynamics of Finite Elements. Tech. Rep. CU-CAS-00-03, College of Engineering, University of Colorado, 2000.

[HGS03] Hauth, M., Gross, J., and Strasser, W. Interactive Physically Based Solid Dynamics. In *ACM Symposium on Computer Animation (SCA)*, 17–27, 2003.

[HS04] Hauth, M., and Strasser, W. Corotational Simulation of Deformable Solids. In *Proceedings WSCG*, 137–145, 2004.

[HSL+06] Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.-Y., Teng, S.-H., Bao, H., Guo, B., and Shum, H.-Y. Subspace gradient domain mesh deformation. *ACM Transactions on Graphics*, 25(3):1126–1134, 2006.

[ITF04] Irving, G., Teran, J., and Fedkiw, R. Invertible finite elements for robust simulation of large deformation. In *Proceedings of ACM Symposium on Computer Animation (SCA)*, 131–140, 2004.

[JP99] James, D. L., and Pai, D. K. ArtDefo: Accurate real time deformable objects. In *Proceedings ACM SIGGRAPH*, 65–72, 1999.

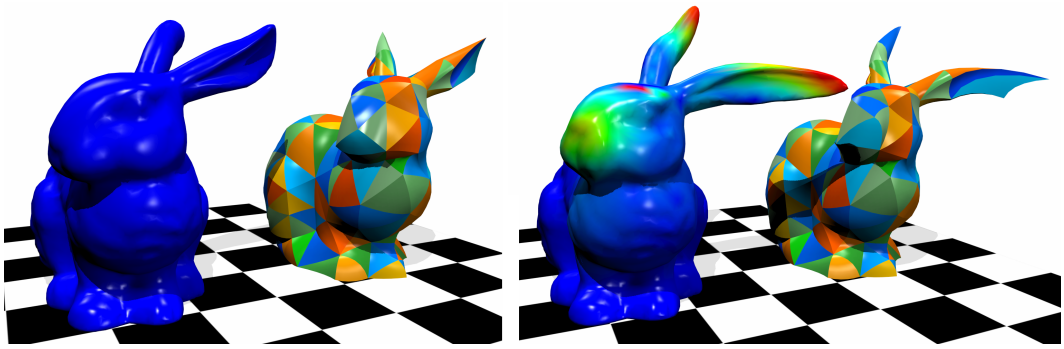


Figure 7. Bunny: 1379 surface vertices, 101 quadratic tetrahedra.

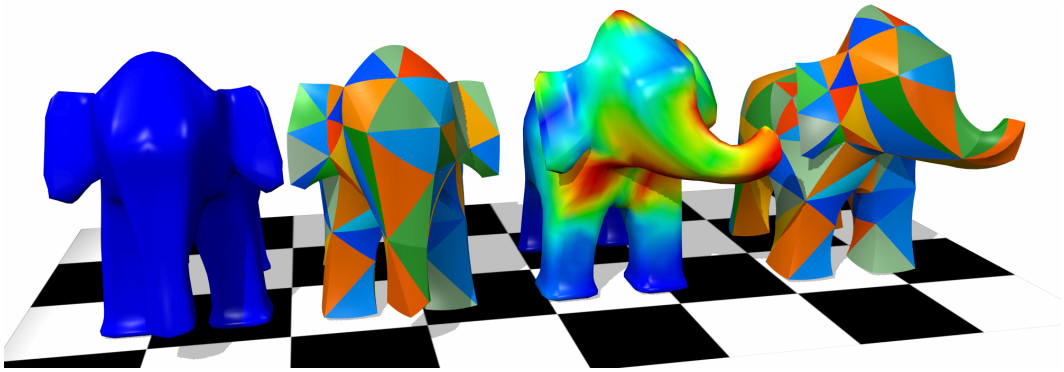


Figure 8. Elephant: 2578 surface vertices, 234 quadratic tetrahedra.

[KCVS98] Kobbelt, L., Campagna, S., Vorsatz, J., and Seidel, H.-P. Interactive Multi-Resolution Modeling on Arbitrary Meshes. *Computer Graphics (Proceedings of ACM SIGGRAPH)*, 32:105–114, 1998.

[LCOGL07] Lipman, Y., Cohen-Or, D., Gal, R., and Levin, D. Volume and shape preservation via moving frame manipulation. *ACM Transactions on Graphics*, 26(1):article no. 5, 2007.

[LSCO+04] Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., and Seidel, H.-P. Differential Coordinates for Interactive Mesh Editing. In *Shape Modeling International 2004 (SMI)*, 181–190, 2004.

[LSLCO05] Lipman, Y., Sorkine, O., Levin, D., and Cohen-Or, D. Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics*, 24(3):479–487, 2005.

[MDM+02] Müller, M., Dorsey, J., McMillan, L., Jagnow, R., and Cutler, B. Stable Real-Time Deformations. In *ACM Symposium on Computer Animation (SCA)*, 49–54, 2002.

[MG04] Müller, M., and Gross, M. Interactive Virtual Materials. In *Proceedings of Graphics Interface*, 239–246, 2004.

[MS06] Mezger, J., and Strasser, W. Interactive Soft Object Simulation with Quadratic Finite Elements. In *Articulated Motion and Deformable Objects (AMDO)*, F. J. Perales and R. B. Fisher, Eds., vol. 4069 of *Lecture Notes in Computer Science*, 434–443. Springer, 2006.

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

[PDA00] Picinbono, G., Delingette, H., and Ayache, N. Real-Time Large Displacement Elasticity for Surgery Simulation: Non-linear Tensor-Mass Model. In *MICCAI*, 643–652, 2000.

[RSB96] Rappaport, A., Sheffer, A., and Bercovier, M. Volume-preserving free-form solids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):19–27, 1996.

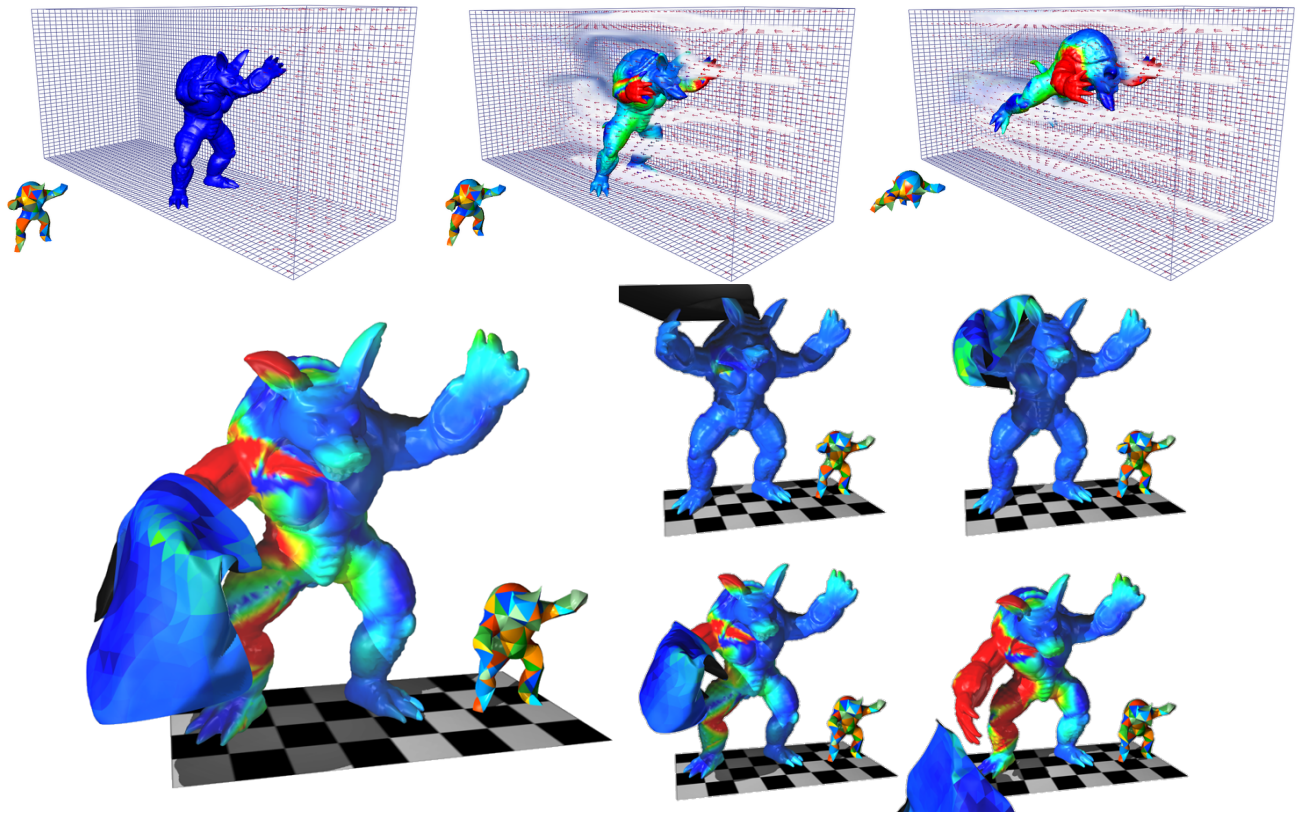


Figure 9. Armadillo: 16608 surface vertices, 146 quadratic tetrahedra. Interaction with wind (top), draped with an FE simulated piece of cloth (bottom).

[SK04] Sheffer, A., and Kraevoy, V. Pyramid Coordinates for Morphing and Deformation. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, 68–75, 2004.

[SLCO+04] Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rossl, C., and Seidel, H.-P. Laplacian surface editing. In *Proc. Eurographics/ACM SIGGRAPH symp. on Geometry processing*, 179–188, 2004.

[SP86] Sederberg, T. W., and Parry, S. R. Free-form deformation of solid geometric models. *Computer Graphics (Proceedings of ACM SIGGRAPH)*, 20(4):151–160, 1986.

[Str71] Stroud, A. H. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, Inc., 1971.

[TF88] Terzopoulos, D., and Fleischer, K. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. *Computer Graphics (Proceedings of ACM SIGGRAPH)*, 22(4):269–278, 1988.

[WG97] Wilhelms, J., and Gelder, A. V. Anatomically based modeling. In *Proceedings of ACM SIGGRAPH*, 173–180, 1997.

[YZX+04] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., and Shum, H.-Y. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics*, 23(3):644–651, 2004.

[ZHS+05] Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., and Shum, H.-Y. Large mesh deformation using the volumetric graph Laplacian. *ACM Transactions on Graphics*, 24(3):496–503, 2005.

[ZSS97] Zorin, D., Schröder, P., and Sweldens, W. Interactive multiresolution mesh editing. In *Proceedings of ACM SIGGRAPH*, 256–268, 1997.

[ZT00] Zienkiewicz, O. C., and Taylor, R. L. *The Finite Element Method*, vol. 1 and 2. Butterworth-Heinemann, fifth ed., 2000.

Figure 10. Dragon: 100.000 surface vertices, 173 quadratic tetrahedra. The surface colours visualise the mapping to the boundary tetrahedra.

Figure 11. High quality rendering of the dragon with 250.000 vertices and large deformation.