The Visual Computer manuscript No.
(will be inserted by the editor)

Ruediger Schmedding · Matthias Teschner

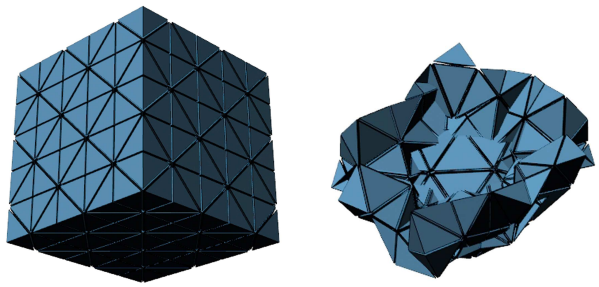
Inversion handling for stable deformable modeling

Abstract In 3D deformable modeling approaches based on FEM, inverted tetrahedral elements can cause undesired visual artifacts and the breakdown of the simulation. As inversion can never be avoided and sometimes even is the correct behavior of elements, there is a strong need for stable inversion handling. In this paper, we propose a novel method to resolve inverted elements which is motivated by previous work of Irving et al. [6]. In combination with an efficient handling of degenerated elements, our approach yields a stable simulation of arbitrary deformations. Although we focus on the corotational formulation of linear FEM, the method can be implemented within arbitrary constitutive models.

Keywords Deformable Modeling · Finite Element Method · Corotational FEM · Inverted Elements

1 Introduction

In Computer Graphics, the simulation of deformable objects based on physical laws started with the pioneering work of Terzopoulos [14]. Since then, several deformation models have been established, e. g. mass-spring-approaches, geometrically motivated models [10] or potential based methods [15]. Recently, the Finite Element Method (FEM) became famous to achieve physically realistic animations. In this context, 3D deformable modeling approaches are based on tetrahedral elements. Due to the computational cost of nonlinear FEM, linear FEM is commonly used in interactive simulations which have a great variety of applications in e. g. virtual surgery or games. However, the fact that linear FEM is not invariant under rigid body rotations has turned out to be a problem. It implies that the rotation of an object has to be extracted prior to the force computation. In many FEM applications, this is done by a polar decomposition of the deformation gradient, e. g. Hauth and Strasser [5] and Mueller and Gross [9]. However, this does not guarantee to return a pure rotation. Instead, it returns a rotation if and only if the tetrahedron is not inverted. Otherwise, the polar decomposition returns an orthogonal matrix that includes a reflection. Unfortunately, this results in force discontinuities when a tetrahedron gets inverted. Further, the forces of an inverted tetrahedron erroneously act to keep it inverted. Fig. 1 illustrates this problem.



(a) (b)

Fig. 1 (a) shows the tetrahedral mesh of a cube falling onto the ground. (b) shows an erroneous equilibrium state after the impact if inverted tetrahedrons are not adequately handled.

At this point, we want to emphasize that inversion is not a problem of linear FEM itself, but only of the corotational formulation. In Sec. 4.2, we show that problems with inverted elements are solely caused by the possibly improper rotation returned by the polar decomposition.

One way to address this problem would be to prevent the inversion of elements. However, this requires additional forces that are difficult to motivate. Further, it

R. Schmedding
University of Freiburg
Georges-Koehler-Allee 052
79110 Freiburg
Tel.: +49-761-2038286
Fax: +49-761-2038262
E-mail: schmedd@informatik.uni-freiburg.de

M. Teschner
University of Freiburg
Georges-Koehler-Alle 052
79110 Freiburg

is difficult to guarantee that inversion is really avoided. Besides, there are cases where inversion is the correct behavior of elements (see e. g. [6]). As the linear model works well for inverted elements, it seems to be more appropriate to allow the inversion and to adequately handle inverted elements.

In order to process inverted elements, a particular inversion direction has to be determined. This direction cannot be extracted by simply considering the current deformation state. If an inappropriate direction is chosen, the computation results in force discontinuities. Irving et al. [6] induced an inversion handling approach based on the heuristic assumption that an inverted tetrahedron is "as uninverted as possible". This implies that an inverted tetrahedron should get uninverted along the direction that causes the minimum movement of a vertex. While the underlying heuristic is very useful and appropriate, there exist cases where the inversion handling of [6] is not conform to this assumption.

1.1 Contribution

In this paper, we propose a new method to determine the inversion direction of inverted elements. The approach is similar to [6]. It is based on the same heuristic that elements are as uninverted as possible. However, we show that the proposed method exceeds the approach of Irving et al. in the fact that it always chooses the direction that is implied by the heuristic assumption. We further illustrate positive effects of our inversion handling approach in dynamic simulations. Compared to other strategies, inversions are efficiently resolved within a small number of simulation steps. Our approach requires that the deformation gradient can be transformed to a diagonal form. Thus, it can be implemented within any material constitutive model. The approach can easily be implemented in combination with a stable handling of degenerated elements. It improves the robustness and stability of FE based deformable modeling approaches.

2 Related work

The Finite Element Method has been used in various applications, e. g. muscle simulation [1], virtual surgery [13] and fracture simulation [11,12,4]. Debunne et al. [2] have introduced a multi-resolution model to use the Finite Element Method in interactive simulations. The first approach to adapt linear FEM to interactive simulations was the warped stiffness concept introduced by Mueller et al. [8]. In this approach, rotations are computed per vertex. However, ghost forces can occur. This issue has been addressed by Hauth and Strasser [5] and Mueller and Gross [9]. Hauth and Strasser [5] propose to extract the spatial rotation from the deformation gradient. And Mueller and Gross [9] show how this can easily be done if the barycentric coordinates are used as shape functions.

In [6], Irving et al. discuss that many deformation models are not valid for inverted elements and therefore, many FEM algorithms are not capable of inversion. However, as strategies like untangling meshes [3,16] are not guaranteed to work, they show the necessity to introduce a method that can handle inversion. They further propose to modify constitutive models near the origin, e. g. by a linearization, in order to avoid corruption in the case of inversion. This is justified as otherwise even a single inverted element could cause the breakdown of a simulation.

Our approach is motivated by [6]. Basically, an inversion direction is determined to handle inverted elements. In contrast to existing approaches, the proposed method estimates intuitive inversion directions in all cases.

3 Deformation model

In our work, we are interested in dynamic simulations of deformable objects. An object is characterized by a time-dependent mapping $\mathbf{\Phi} : [0, \infty) \times \Omega \rightarrow \mathbb{R}^3$ with $\Omega \subset \mathbb{R}^3$. $\mathbf{\Phi}$ maps each point in material coordinates to its position in world coordinates. For a fixed point in time t, the function can be written as $\mathbf{\Phi}_t : \Omega \rightarrow \mathbb{R}^3$. In the Finite Element Method, one partitions the object into a set of primitives and replaces $\mathbf{\Phi}_t$ by a piecewise polynomial interpolation $\mathbf{\Phi}_t^P$, where every polynomial is defined on a primitive. Usually, tetrahedrons are used as primitives. Let $\mathbf{\Phi}_{t,e}^P$ be the restriction of $\mathbf{\Phi}_t^P$ to some tetrahedron e. In the corotational model of Hauth and Strasser [5], the rotation of e is extracted by a polar decomposition of the deformation gradient $\mathbf{D}\mathbf{\Phi}_{t,e}^P$. If we apply linear shape functions, $\mathbf{\Phi}_{t,e}^P$ is an affine linear mapping and can be written as $\mathbf{\Phi}_{t,e}^P(x) = \mathbf{B}_e\mathbf{x} + \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^3$ is a translation. Clearly we have $\mathbf{D}\mathbf{\Phi}_{t,e}^P = \mathbf{B}_e$, and we will refer to $\mathbf{B}_e$ as the deformation gradient from now on. In the case that we take the barycentric coordinates as shape functions, Mueller and Gross [9] show a simple way to compute $\mathbf{B}_e$. As stated in the introduction, [9] and [5] compute the rotational part $\mathbf{R}_e$ by a polar decomposition $\mathbf{B}_e = \mathbf{R}_{B_e}\mathbf{S}_{B_e}$ of $\mathbf{B}_e$.

Looking at the determinant of $\mathbf{B}_e$, we can decide on the state of the tetrahedron. A tetrahedron is inverted if and only if $\det(\mathbf{B}_e) < 0$, it is degenerated if and only if the deformation gradient is singular, i. e. $\det(\mathbf{B}_e) = 0$, and in a "normal" deformation state otherwise.

3.1 Notation

For a brief review of the force computation and to show that problems with inversion come up with the corotational model, we introduce a few terms that are used in this paper. As forces are always computed at a fixed point in time, we dismiss the index t. For a tetrahedron, $\mathbf{x}_0, \ldots, \mathbf{x}_3$ denote the current positions of its ver-

tices, $\mathbf{x}_{cm}$ denotes its center-of-mass, and we set $\mathbf{r}_i := \mathbf{x}_i - \mathbf{x}_{cm}$, $i = 0, \dots, 3$. $\mathbf{x} \in \mathbb{R}^{12}$ without an index denotes a vector that contains the coordinates of all vertices, i. e. $\mathbf{x} = (\mathbf{x}_0^T, \dots, \mathbf{x}_3^T)^T$. An apostrophe $'$ indicates the initial value of a variable. $\mathbf{K}_e \in \mathbb{R}^{12\times12}$ denotes the stiffness matrix of element e and $\mathbf{f}_e \in \mathbb{R}^{12}$ collects the forces that act on the vertices like $\mathbf{x}$ collects the coordinates. With $\mathbf{R}_e \in \mathbb{R}^{12\times12}$ we denote a blockdiagonal matrix with $\mathbf{R}_{B_e} \in \mathbb{R}^{3\times3}$ being the four nonzero blocks along the diagonal and zero entries at all other positions.

3.2 Force computation

In linear FEM, the internal forces are computed by the equation $\mathbf{f}_{e,lin} = \mathbf{K}_e(\mathbf{x} - \mathbf{x}') =: \mathbf{K}_e\mathbf{q}_{lin}$. To eliminate the rotation from the force computation, the deformed tetrahedron is rotated by $\mathbf{R}_{B_e}^T$, so $\mathbf{x}$ is rotated by $\mathbf{R}_e$, and the forces are rotated back by $\mathbf{R}_{B_e}$ resp. $\mathbf{R}_e$ afterwards:

$$\mathbf{f}_e = \mathbf{R}_e\mathbf{K}_e(\mathbf{R}_e^T\mathbf{x} - \mathbf{x}') =: \mathbf{R}_e\mathbf{K}_e\mathbf{q}. \tag{1}$$

The corotational displacement $\mathbf{q} = (\mathbf{R}_e^T\mathbf{x} - \mathbf{x}')$ contains only the symmetric part of the deformation. For the dynamic simulation, the equation $\mathbf{M}\ddot{\mathbf{x}} + \mathbf{f}_{e,lin} = 0$ resp. $\mathbf{M}\ddot{\mathbf{x}} + \mathbf{f}_e = 0$ is solved, where $\mathbf{M}$ is the mass matrix. If we assume mass lumping and distribute the mass equally to all vertices, the equation simplifies to $m\ddot{\mathbf{x}} = -\mathbf{f}_e$. For the further discussion we do not need to consider friction and external forces and therefore omit the corresponding terms.

For later use note, that we could equivalently define the corotational displacement using $\mathbf{S}_{B_e}$. For this purpose, we denote with $\mathbf{S}_e \in \mathbb{R}^{12\times12}$ a blockdiagonal matrix with $\mathbf{S}_{B_e}$ as the blocks along the diagonal and zero entries elsewhere. As $\mathbf{R}_{B_e}\mathbf{B}_e = \mathbf{S}_{B_e}$, we see that $\mathbf{R}_e^T\mathbf{x}$ and $\mathbf{S}_e\mathbf{x}'$ differ only by a translation. Therefore, defining the corotational displacement as $\mathbf{q}^S := (\mathbf{S}_e\mathbf{x}' - \mathbf{x})$ would result in the same forces like $\mathbf{q}$.

4 Inversion handling

In this section, our inversion handling approach is explained. In Sec. 4.1, we briefly describe the polar decomposition in order to illustrate the problem that has to be addressed. And in Sec. 4.2, we show that the problem can be solved by computing a proper rotation for inverted elements. In Sec. 4.3, the basic idea of the inversion handling method is outlined, while Sec. 4.4 briefly describes the idea of Irving et al. [6]. The properties of [6] are discussed in Sec. 4.5. In particular, a case of a non-intuitive inversion direction is shown. Our approach, described in Sec. 4.6, addresses this issue. In Sec. 4.7, the incorporation of degenerated elements is explained.

4.1 Polar decomposition

Let $\mathbf{B}_e$ be some nonsingular deformation gradient of a tetrahedron e and $\mathbf{B}_e = \mathbf{R}_{B_e}\mathbf{S}_{B_e}$ its polar decomposition. Then we know that $\mathbf{R}_{B_e}$ is an orthogonal matrix and $\mathbf{S}_{B_e}$ is symmetric positive definite. It follows that $\det(\mathbf{R}_{B_e}) = \operatorname{sign}(\det(\mathbf{B}_e))$. This implies that if e is inverted and hence, $\det(\mathbf{R}_e) = -1$, the internal forces are not only rotated, but also reflected. Hence, they act to keep the tetrahedron inverted (see Fig. 2). Furthermore, the reflection of forces implies force discontinuities during the inversion of a tetrahedron which cause visual artifacts.

For convenience, we briefly summarize one variant of the polar decomposition. First, we compute the square root of $\mathbf{B}_e^T\mathbf{B}_e$. This is done by computing the diagonalization $\mathbf{D}_e = \mathbf{Q}^T\mathbf{B}_e^T\mathbf{B}_e\mathbf{Q}$ of $\mathbf{B}_e$, where $\mathbf{Q}$ is an orthonormal matrix with the eigenvectors of $\mathbf{B}_e^T\mathbf{B}_e$ being the columns of $\mathbf{Q}$. The diagonalization exists because $\mathbf{B}_e^T\mathbf{B}_e$ is symmetric. The square root of $\mathbf{B}_e^T\mathbf{B}_e$ then is simply given by $\sqrt{\mathbf{B}_e^T\mathbf{B}_e} = \mathbf{Q}\sqrt{\mathbf{D}_e}\mathbf{Q}^T$. Then we set $\mathbf{S}_{B_e} := \sqrt{\mathbf{B}_e^T\mathbf{B}_e}$ and compute the rotation $\mathbf{R}_{B_e}$ as $\mathbf{R}_{B_e} = \mathbf{B}_e\mathbf{S}_{B_e}^{-1}$. The inverse $\mathbf{S}_{B_e}^{-1}$ is simply computed as $\mathbf{Q}\sqrt{\mathbf{D}_e}^{-1}\mathbf{Q}^T$ and does not produce significant computational overhead.

An overview of various techniques of polar decomposition can be found in [17].

4.2 Inversion problems due to rotation

In this section, we show that problems with inverted elements must be located in the improper rotation returned by the polar decomposition. Therefore, they can be resolved by computing a proper rotation.

We consider a tetrahedron e. As translation does not cause any forces, we can assume w. l. o. g. $\mathbf{x}_{cm} = \mathbf{x}'_{cm}$. For the analysis of the linear FEM, we assume that the deformation does not contain a rotational part. However, the tetrahedron might be inverted. As $\mathbf{K}_e$ is symmetric, it is diagonalizable. Furthermore, it is positive semidefinite. Let $\mathbf{v}_0, \dots, \mathbf{v}_{11}$ be the eigenvectors of $\mathbf{K}_e$ with eigenvalues $\lambda_0, \dots, \lambda_{11}$ and consider the displacement $\mathbf{q}_{lin}$ represented using the eigenvectors, i. e. $\mathbf{q}_{lin} = \sum_{i=0}^{11} q_{lin}^i\mathbf{v}_i$. We can simply write the internal forces as $\mathbf{f}_{e,lin} = \sum_{i=0}^{11} \lambda_i q_{lin}^i\mathbf{v}_i$. For the dynamic equation, it follows $m\ddot{\mathbf{x}} = \sum_{i=0}^{11}(-\lambda_i q_{lin}^i)\mathbf{v}_i$. The minus sign indicates that each component of the accelerating forces along an eigenvector acts opposite to the causal displacement. Hence, the displacement is reduced and the tetrahedron tends to its uninverted resting state. Note that we do not use any knowledge whether the tetrahedron is inverted or not. Thus, this argument holds true also for inverted tetrahedrons.

In the same manner, we can decompose the corotational displacement $\mathbf{q}$ into its components along the

eigenvectors and analogously conclude that each component causes a force that accelerates the vertices in the direction opposite to the causal displacement component. If the displacement is computed correctly, the forces will be, too. Therefore, the only possible reason for any errors that occur when inverted elements are involved must be located in the rotation $\mathbf{R}_e$. As seen in Sec. 4.1, element inversion yields an improper rotation in the polar decomposition. Due to this fact, the computed forces cause an acceleration of the vertices to the positions of the reflected resting state. This is described in the two-dimensional example in Fig. 2.

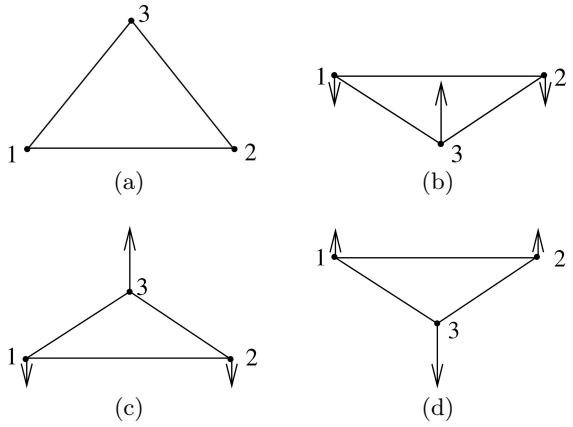


Fig. 2 This figure shows the error in the corotational force computation due to an improper rotation. In (a) we see the original triangle. (b) shows the current deformation state $\mathbf{x}$, where the triangle is inverted. The arrows indicate the correct forces that should be computed in this state. (c) visualizes an intermediate step in the corotational force computation (see eq. (1): $\mathbf{f}_e = \mathbf{R}_e\mathbf{K}_e(\mathbf{R}_e^T\mathbf{x} - \mathbf{x}')$), where the computation is decomposed in its single steps. It shows the transformed deformation state $\mathbf{R}_e^T\mathbf{x}$. The internal forces are computed w. r. t. this virtual state. As $\det(\mathbf{R}_{B_e}) = -1$, this virtual state is the reflection of the correct state seen in (b). Therefore, the triangle seems not to be inverted during force computation. The arrows depict the intermediate forces $\mathbf{f}_{e,i} = \mathbf{K}_e(\mathbf{R}_e^T\mathbf{x} - \mathbf{x}')$ that are caused by this virtual state. In (d), we see the current deformation state of the triangle and the corotational forces $\mathbf{f}_e = \mathbf{R}_e\mathbf{f}_{e,i}$, which are just the reflected intermediate forces shown in (c) and therefore act in the wrong direction.

4.3 Basic idea

Having recognized that problems with inverted elements are caused solely by the corotational formulation and come up due to polar decomposition, we see that eliminating the arising lacks should only be done by a modification of the polar decomposition. The idea of the corotational formulation was to separate the rotational part from the deformation, which is violated if $\mathbf{R}_{B_e}$ contains a reflection. So, we have to modify the polar decomposition in such a way that the reflection is contained in $\mathbf{S}_{B_e}$ and $\mathbf{R}_{B_e}$ is an orthonormal matrix.

Let $\hat{\mathbf{S}}_{B_e} = \mathbf{Q}^T\mathbf{S}_{B_e}\mathbf{Q}$ be the diagonalization of $\mathbf{S}_{B_e}$. To include a reflection in $\mathbf{S}_{B_e}$, we can invert either one or three of the diagonal entries of $\hat{\mathbf{S}}_{B_e}$ which results in some $\hat{\mathbf{S}}'_{B_e}$ with $\det(\hat{\mathbf{S}}'_{B_e}) < 0$ and redefine $\mathbf{S}_{B_e} := \mathbf{Q}\hat{\mathbf{S}}'_{B_e}\mathbf{Q}^T$, computing $\mathbf{R}_{B_e} = \mathbf{B}_e\mathbf{S}_{B_e}^{-1}$ afterwards. Optimally, we could identify the direction in which the tetrahedron got inverted and choose the corresponding entry of $\hat{\mathbf{S}}_{B_e}$ to be negative, because inverting another one would lead to force discontinuities. As this is not possible by just looking at the current state of the tetrahedron, we have to make some heuristic assumption about the current deformation.

4.4 Existing approach

Irving et al. [6] assume that a tetrahedron is as uninverted as possible, which implies two aspects: Only one component of $\hat{\mathbf{S}}_{B_e}$ should be chosen to be negative and this component should correspond to the direction that causes minimum movement to uninvert the tetrahedron in this direction. They use singular value decomposition (SVD) $\mathbf{B}_e = \mathbf{U}_{B_e}\hat{\mathbf{S}}_{B_e}\mathbf{V}_{B_e}^T$, where $\mathbf{U}_{B_e}$ and $\mathbf{V}_{B_e}$ are orthogonal matrices, to diagonalize the deformation gradient. This leads to the same diagonal matrix as the polar decomposition (see Sec. 4.7), where they choose the smallest diagonal entry of $\hat{\mathbf{S}}_{B_e}$ which should correspond to the desired direction.

4.5 Discussion of the existing approach

The choice of Irving et al. [6] seems to be founded in [7], where the minimization of the quadratic form

$$\sum_{i=0}^{3} ||\mathbf{r}_i - \mathbf{R}\mathbf{r}'_i||^2 \tag{2}$$

among all rotations $\mathbf{R}$ is considered. Remember that we set $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}_{cm}$ to be the coordinates relative to the center-of-mass. They show that the optimal rotation can be extracted from the matrix $\mathbf{A}_{rr'} := \sum \mathbf{r}_i\mathbf{r'}_i^T$ by polar decomposition $\mathbf{A}_{rr'} = \mathbf{R}_{rr'}\mathbf{S}_{rr'}$ in the case $\det(\mathbf{A}_{rr'}) > 0$. If $\det(\mathbf{A}_{rr'}) < 0$, they show that one has to compute the diagonalization $\hat{\mathbf{S}}_{rr'}$ of the symmetric part $\mathbf{S}_{rr'}$ of the polar decomposition and indeed invert the smallest diagonal entry to obtain the optimal rotation $\mathbf{R}_{rr'}$, like it is proposed by [6] for the deformation gradient.

However, $\mathbf{A}_{rr'}$ generally does not equal the deformation gradient $\mathbf{B}_e$. Although they differ only by a symmetric matrix (see [10]), e. g. $\mathbf{B}_e = \mathbf{A}_{rr'} \cdot \mathbf{A}_{sym}$, their polar decompositions return different rotations. This is founded by the fact that the product of two symmetric matrices is symmetric if and only if both matrices are

simultaneous diagonalizable. Therefore, $\mathbf{S}_{rr'} \cdot \mathbf{A}_{sym}$ in general is not symmetric and $\mathbf{B}_e = \mathbf{R}_{rr'} \cdot (\mathbf{S}_{rr'} \cdot \mathbf{A}_{sym})$ is not equal to the polar decomposition $\mathbf{R}_{B_e}\mathbf{S}_{B_e}$ of $\mathbf{B}_e$. It follows that the inversion of the smallest diagonal entry of $\hat{\mathbf{S}}_{B_e}$ does not necessarily minimize (2), which seemed to be a motivation, and that the minimization of (2) is not the goal of the corotational model, as $\mathbf{B}_e \neq \mathbf{A}_{rr'}$ in general.

Further, choosing the smallest diagonal entry does not necessarily fulfill the heuristic assumption as it does not always correspond to the direction that causes minimum movement to uninvert the tetrahedron. This is easily seen by the two-dimensional example in Fig. 3 which could analogously be performed in $\mathbb{R}^3$. The upper triangle shows its resting state that has small extension h in y-direction. The lower one shows its current deformation state. It is inverted in y-direction and compressed in x-direction, so the deformation gradient $\mathbf{B}_e$ is diagonal. As it has the same y-height h as in the resting state, the entry corresponding to this direction is -1, while the entry in x-direction is a little smaller than one, say 0.9. Then, the deformation gradient $\mathbf{B}_e$ would be $\mathbf{B}_e = \left(\begin{smallmatrix} 0.9 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ and the symmetric part $\mathbf{S}_{B_e} = \left(\begin{smallmatrix} 0.9 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Hence, as the entry corresponding to the x-direction is smaller, the triangle is chosen to reinvert along x-direction. This clearly contradicts the heuristic assumption. The problem is that the smallest diagonal entry does not reflect the distance of any vertex to its opposite face, because it obviously depends on the resting state. The flatter the triangle is, the more probable it is that this rule chooses a non-intuitive component.

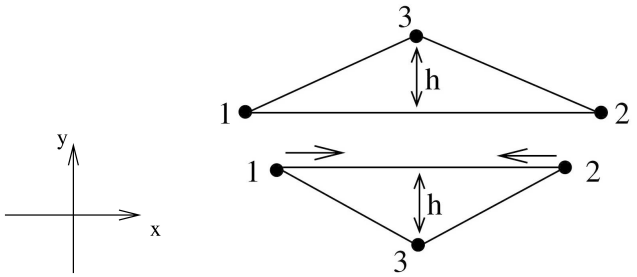


Fig. 3 This figure illustrates a non-intuitive choice of the inversion direction. The upper triangle shows the resting state, and the lower one shows the current deformation. The triangle is deformed such that it is inverted in y-direction and slightly compressed in x-direction. The entry of $\mathbf{B}_e$ corresponding to the y-direction is -1, because it is inverted and has the same height h as in the initial state. Hence, the entry of $\mathbf{S}_{B_e}$ corresponding to the y-direction is 1. The entry in x-direction is smaller than 1 and therefore, it is chosen as the inversion direction. Consequently, the triangle gets uninverted in x-direction as indicated by the arrows. However, according to the heuristic assumption, obviously the y-direction should be chosen.

4.6 Our approach

Our approach is based on the heuristic assumption of Irving et al. [6] which states that the tetrahedron is as uninverted as possible. This assumption demands that we locate the direction that causes minimum movement to uninvert the tetrahedron. As this direction does not necessarily correspond to the smallest entry in $\mathbf{S}_{B_e}$, there is some more work to do. First we recognize that the direction to reinvert the tetrahedron should be the direction in which one of the vertices has the shortest distance to its opposite face. Note that "distance" must not be interpreted as the standard orthogonal distance, but it rather means "distance along some given direction". The reason is that $\hat{\mathbf{S}}_{B_e}$ restricts the choice to three directions, which are given by the eigenvectors of $\mathbf{S}_{B_e}$. If we took the orthogonal direction, we would not know which diagonal entry to invert. Hence, we have only three predetermined directions that can be chosen as the inversion direction, and we have to compute the shortest distance along one of these directions.

To determine the direction causing minimum movement, we look for a pair $(\mathbf{c}, \mathbf{v})$ consisting of an eigenvector $\mathbf{v}$ of $\mathbf{S}_{B_e}$ with $||\mathbf{v}|| = 1$ and a vertex $\mathbf{c}$ such that the distance from $\mathbf{c}$ along $\mathbf{v}$ to its opposite face $\mathbf{F}_c$ is minimized among all possible pairs. This is done by computing a parameter $\lambda_{c,v}$ for each pair $(\mathbf{c}, \mathbf{v})$ with $||\mathbf{v}|| = 1$ such that $\mathbf{x}_c + \lambda_{c,v} \cdot \mathbf{v}$ lies on $\mathbf{F}_c$, where $\mathbf{x}_c$ denotes the position of $\mathbf{c}$. Clearly, $|\lambda_{c,v}|$ then denotes the distance of $\mathbf{c}$ along $\mathbf{v}$ to $\mathbf{F}_c$. Hence, the desired pair $(\mathbf{c}_0, \mathbf{v}_0)$ is the one that corresponds to the minimum $|\lambda_{c_0,v_0}|$.

For the computation of $\lambda_{c,v}$, we have to note that the eigenvectors of $\mathbf{S}_{B_e}$ are related to the unrotated coordinate frame, while the current deformation state contains a rotation. Remembering Sec. 3.2, we see that we could compute the forces equivalently with respect to the state $\mathbf{S}_e\mathbf{x}'$. This state is related to the unrotated coordinate frame and obviously fits to the eigenvectors of $\mathbf{S}_{B_e}$. However, as shown in Sec. 4.2 it does not contain the inversion and therefore, it causes erroneous forces. Inverting one of the diagonal entries of $\hat{\mathbf{S}}_{B_e}$ now is equivalent to inverting the reference state $\mathbf{S}_e\mathbf{x}'$ in order to get the correct forces. $\mathbf{S}_e\mathbf{x}'$ and the current deformation state $\mathbf{x}$ differ by translation, rotation and inversion. Since none of these changes any distance, looking for the direction that causes minimum movement to uninvert the deformed, rotated tetrahedron is equivalent to looking for the direction that causes minimum movement to invert the unrotated reference state $\mathbf{S}_e\mathbf{x}'$. Therefore, we can compute the parameters $\lambda_{c,v}$ with respect to $\mathbf{S}_e\mathbf{x}'$.

Now let $\mathbf{c}'$ be any vertex of the face $\mathbf{F}_c$ and $\mathbf{x}^S_{c'}$ its position in the reference state $\mathbf{S}_e\mathbf{x}'$. Let $\mathbf{x}^S_c$ be the position of $\mathbf{c}$ in this state. As $\lambda_{c,v}\mathbf{v}$ and $(\mathbf{x}^S_{c'} - \mathbf{x}^S_c)$ are vectors that point from $\mathbf{c}$ to some point on the plane that contains $\mathbf{F}_c$, they have the same component along the face normal of $\mathbf{F}_c$. With $\mathbf{n}_c$ being the unit face normal of $\mathbf{F}_c$,

this fact could be expressed by the following equation.

$$\lambda_{c,v}\mathbf{v}\cdot\mathbf{n}_c = (\mathbf{x}^S_{c'} - \mathbf{x}^S_c)\cdot\mathbf{n}_c \tag{3}$$

Now it follows that $\lambda_{c,v}$ can be computed using

$$\lambda_{c,v} = \frac{(\mathbf{x}^S_{c'} - \mathbf{x}^S_c)\cdot\mathbf{n}_c}{\mathbf{v}\cdot\mathbf{n}_c}. \tag{4}$$

Choosing $\mathbf{v_0}$ according to the minimum $|\lambda_{c_0,v_0}|$ provides the direction with minimum movement to invert the reference state $\mathbf{S}_e\mathbf{x}'$. As argued above, $\mathbf{v}_0$ also provides the direction causing minimum movement to uninvert the tetrahedron, and we change the sign of the diagonal entry that corresponds to $\mathbf{v}_0$. Since the length of $\mathbf{n}_c$ cancels out in (4), we can drop the condition that it has to be a unit vector.

Further, we store $\mathbf{c}_0$ as the vertex that is seen as the one that caused the inversion by crossing $\mathbf{F}_{c_0}$. Until the tetrahedron returns to an uninverted state, we compute the optimal direction $\mathbf{v}_0$ with respect to $\mathbf{c}_0$, so that the tetrahedron gets uninverted by $\mathbf{c}_0$ crossing $\mathbf{F}_{c_0}$ again. Note that we cannot store any direction, because the eigenvectors of $\mathbf{S}_{B_e}$ and therewith the possible inversion directions change in each time step. Hence, we have to refer to a vertex to achieve a consistent choice of inversion directions in subsequent iterations.

We introduced this approach for the corotational formulation of linear FEM. However, it can be implemented for arbitrary deformation models. The only condition is that the deformation gradient can be transformed into diagonal form. This is always possible using the SVD $\mathbf{B}_e = \mathbf{U}_{B_e}\hat{\mathbf{S}}_{B_e}\mathbf{V}^T_{B_e}$, where $\mathbf{U}_{B_e}$ can be interpreted as a rigid body rotation, and $\mathbf{V}_{B_e}$ as a material rotation (see [6]). The columns of $\mathbf{V}_{B_e}$ can be interpreted as the possible inversion directions that are needed in our approach. Irving et al. [6] show how this can be generalized to anisotropic materials.

4.7 Degenerated tetrahedrons

As seen in Sec. 3, a tetrahedron is inverted if and only if $\det(\mathbf{B}_e) < 0$, whereas it is degenerated if and only if $\det(\mathbf{B}_e) = 0$. Therefore, inverted elements and degenerated elements are handled completely independent from each other. Together with an efficient handling of degenerated elements, our approach yields a stable simulation of arbitrary deformations.

Like inversion, problems with degenerated elements occur only in the corotational formulation. The problem is again located in the polar decomposition. We adopt a simple solution shown in [6] that is based on SVD and leads to a solution for the polar decomposition, which is described in this section. The polar decomposition and the SVD of a deformation gradient $\mathbf{B}_e$ are connected by $\mathbf{B}_e = \mathbf{R}_{B_e}\mathbf{S}_{B_e} = \mathbf{R}_{B_e}\mathbf{Q}\hat{\mathbf{S}}_{B_e}\mathbf{Q}^T =: \mathbf{U}\hat{\mathbf{S}}_{B_e}\mathbf{Q}^T$, which is the SVD of $\mathbf{B}_e$. If $\mathbf{B}_e$ is singular, then at least one of the diagonal entries of $\hat{\mathbf{S}}_{B_e}$ is zero. If it is exactly one entry, the two columns of $\mathbf{U}$ corresponding to the nonzero entries are uniquely determined. The third column then is computed as the cross product of the other two and therefore is unique, too. Hence $\mathbf{R}_{B_e} = \mathbf{U}\mathbf{Q}^T$ results in an unique solution for the polar decomposition. If more than one diagonal entry equals zero, the solution is no longer unique and we choose a system of orthonormal columns for $\mathbf{U}$. In every case, $\mathbf{R}_{B_e}$ is guaranteed to be a proper rotation.

5 Results

We performed several experiments to show how our new approach works. Remember the heuristic assumption that a tetrahedron always is as uninverted as possible. In Sec. 5.1, we show that our method chooses the direction corresponding to this assumption and that it guarantees fast and reliable recovery from inversion. In Sec. 5.2, we show that it can easily be implemented together with a stable handling of degenerated elements.

5.1 Recovery from inversion

In the first experiment, we take a single tetrahedron and invert it manually. The resting state and the inverted state are shown in Fig. 4.

After the manual inversion, we simulate the behavior of the tetrahedron starting with the inverted state. The inversion direction that is chosen by the existing approach is indicated by the red line in Fig. 5(a). Compared to Fig. 4(c), we see that this choice is non-intuitive. Fig. 5(b) illustrates the resting state that is reached using the existing approach.

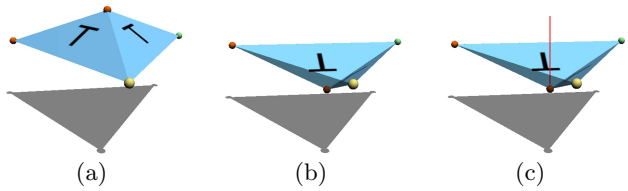
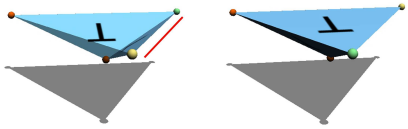


Fig. 4 This figure illustrates the setting of the first experiment. (a) shows the resting state of the tetrahedron. (b) shows an inverted state where the top vertex was moved below its opposite face. (c) depicts the expected inversion direction, which is preferred by the heuristic assumption.

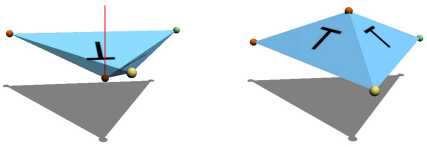
In Fig. 6, we show that the tetrahedron is uninverted correctly by our method. As the picture illustrates, our approach locates the correct direction and uninverts the tetrahedron as expected.

In the experiment shown in Fig. 7, a cube falls down onto the ground (Fig. 7(a)) which leads to the inversion of many tetrahedrons. It demonstrates that inver-



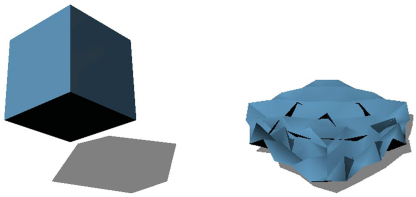
(a) (b)

Fig. 5 (a) illustrates the reinversion direction that is chosen by the existing approach. (b) shows the resting state after the existing approach is applied.



(a) (b)

Fig. 6 (a) repeats the expected reinversion direction to illustrate that our approach restores the intuitive resting state which is depicted in (b).

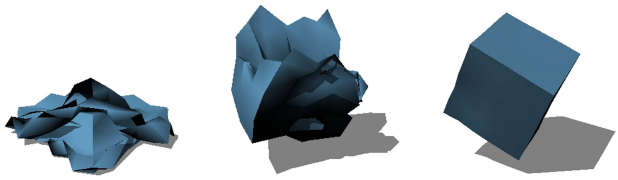


(a) (b)

Fig. 7 (a) shows a cube that is falling down onto the ground. (b) If inversion is not handled correctly, the cube stays in an erroneous equilibrium state.

sion handling is required (Fig. 7(b)) and that the chosen inversion direction significantly influences the behavior of an object (Fig. 8). Fig. 7(b) depicts what happens if we use the corotational FEM approach without any inversion handling. Many tetrahedrons stay inverted due to the improper rotations, and as a consequence the cube loses volume. Also, its energy gets lost in this example, and it does not bounce up any more.

Fig. 8 illustrates a naive inversion handling, where just any diagonal entry is chosen as the inversion direction. The cube recovers to its original shape, but it takes a long time and it suffers from self intersections during the recovery. Due to this fact, it loses energy and does not bounce as high as it should. In Fig. 9, we illustrate the behavior of the falling cube using our approach. First of all, it is not deformed as much as in the other two cases, because the inversion handling locates the correct inversion direction and therefore, the internal forces react faster. The original shape is restored after a few simulation steps, which shows the efficiency of the approach. Further, we can see that the inversion handling does not result in an artificial rotation of the object.



(a) (b) (c)

Fig. 8 Naive inversion handling. (a) shows the maximum compression of the cube. (b) illustrates that there are self-intersections during inversion recovery. (c) shows that the cube restores its original shape.

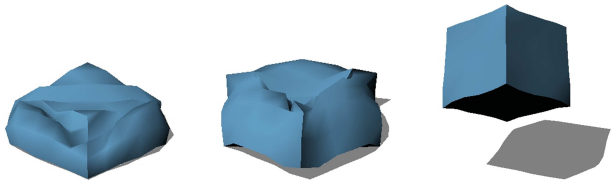


Fig. 9 Our approach. Compared to Fig. 8, the original shape is restored in less simulation steps.

5.2 Handling degenerated elements

This section shows the combination of inversion handling with degenerated elements. We shrink a cow in a virtual sphere until the radius of the sphere is zero and all tetrahedrons are degenerated to a single point. After that, the spherical boundary is removed and the internal forces restore the shape of the cow. During the deformation and the recovery process, many inverted tetrahedrons have to be processed. As stated in Sec. 4.7, the rotation in the degenerated case is not unique when a tetrahedron is degenerated in more than one direction. Therefore, when all tetrahedrons are compressed to a single point, one has to choose any kind of rotation. Hence, it is possible that although the cow does restore its correct shape, it does find the correct rotation.

6 Conclusion

In this paper, we have shown that there is a strong need for efficient and stable inversion handling in FEM algorithms that are commonly based on a corotational formulation. We have reviewed an existing approach, where we have recognized a non-intuitive behavior in some configurations. Therefore, we have introduced a new method for inversion handling that always chooses the most appropriate direction to uninvert an inverted tetrahedron. In contrast to existing approaches, our method can store the inverted component to guarantee a consistent processing of inverted elements in subsequent simulation steps. Thus, our approach improves the stability and robustness of FE based deformable modeling approaches.

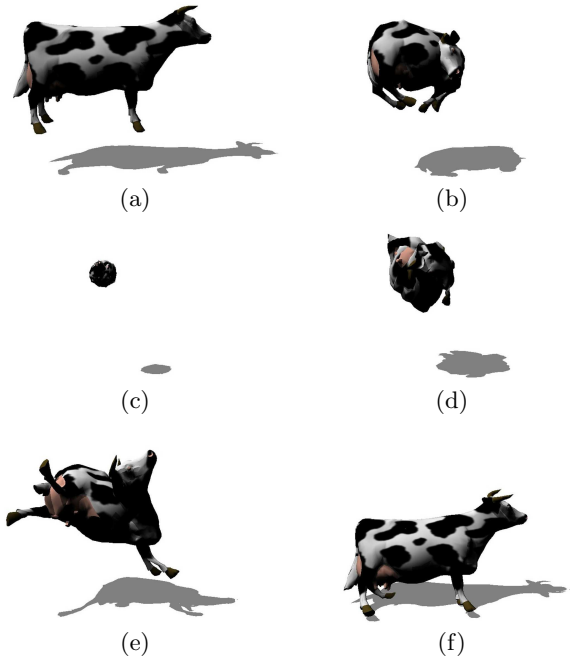


Fig. 10 A cow is shrunk by a virtual sphere ((a)-(c)). After the sphere is removed, the cow restores its original shape ((d)-(f)).

Acknowledgements This project is supported by the German Research Foundation (DFG) under contract number SFB TR8 and by the AO Foundation. We would like to thank the reviewers for their helpful suggestions.

References

1. Chen, D., Zeltzer, D.: Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. Proceedings of ACM SIGGRAPH pp. 89–98 (1992)
2. Debunne, G., Desbrun, M., Cani, M.P., Barr, A.H.: Dynamic Real-Time Deformations using Space & Time Adaptive Sampling. Proc. annual conference on Computer Graphics and iterative techniques, ACM SIGGRAPH pp. 31–36 (2001)
3. Escobar, J.M., Rodríguez, E., Montenegro, R., Montenero, G., González-Yuste, J.M.: Simultaneous untangling and smoothing of tetrahedral meshes. Comput. Meth. in Appl. Mech. and Eng. **192**, 2775–2787 (2003)
4. Espinosa, H., Zavattieri, P., Emore, G.: Adaptive FEM computation of geometric and material nonlinearities with application to brittle fracture. Mechanics of Materials **29**, 275–305 (1998)
5. Hauth, M., Strasser, W.: Corotational Simulation of Deformable Solids. Journal of WSCG **12**, 137–145 (2004)
6. Irving, G., Teran, J., Fedkiw, R.: Invertible Finite Elements For Robust Simulation of Large Deformation. Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation pp. 131–140 (2004)
7. Kanatani, K.: Analysis of 3-D Rotation Fitting. IEEE Transactions on pattern analysis and machine intelligence **16**, 543–549 (1994)
8. Müller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B.: Stable real time deformations. Proceedings of ACM SIGGRAPH Symposium on Computer Animation pp. 49–54 (2002)
9. Müller, M., Gross, M.: Interactive Virtual Materials. Proceedings of Graphics Interface pp. 239–246 (2004)
10. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless Deformations Based on Shape Matching. Proceedings of ACM SIGGRAPH International Conference on Computer Graphics and Interactive Techniques pp. 471–478 (2005)
11. O'Brian, J., Bargteil, A., Hodgins, J.: Graphical modeling of ductile fracture. Proceedings of ACM SIGGRAPH Trans. Graph. 21 pp. 291–294 (2002)
12. O'Brian, J., Hodgins, J.: Graphical modeling and animation of brittle fracture. Proceedings of ACM SIGGRAPH **18**, 137–146 (1999)
13. Picinbono, G., Delingette, H., Ayache, N.: Non-linear and anisotropic elastic soft tissue models for medical simulation. IEEE Int. Conf. Robotics and Automation (2001)
14. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. Proceedings of ACM SIGGRAPH pp. 205–214 (1987)
15. Teschner, M., Heidelberger, B., Müller, M., Gross, M.: A Versatile and Robust Model for Geometrically Complex Deformable Solids. Proc. Computer Graphics International pp. 312–319 (2004)
16. Vachal, P., Garimella, R.V., Shashkov, M.J.: Untangling of 2D meshes in ALE simulations. Journal of Computational Physics **196**, 627–644 (2004)
17. Zieliński, P., Ziętak, K.: The polar decomposition - properties, applications and algorithms. Matematyka Stoswana **38**, 23–40 (1995)

R. Schmedding Ruediger Schmedding is a research assistant and Ph.D. candidate in the Computer Graphics group at the University of Freiburg. He received his diploma in Mathematics at the University of Bonn in 2007. From 2004 to 2007, he was member of the VLSI-Design group at the Research Institute for Discrete Mathematics at the University of Bonn and developed chip design tools for industrial application. His main research interest is the physically-based simulation of deformable objects. In particular, he is focusing on FE models and contact handling.

M. Teschner Matthias Teschner is professor of Computer Science and head of the Computer Graphics group at the University of Freiburg. He received the PhD degree in Electrical Engineering from the University of Erlangen-Nuremberg in 2000. From 2001 to 2004, he was research associate at Stanford University and at the ETH Zurich. His research interests comprise physical simulation, computer animation, computational geometry, collision handling, and real-time rendering with applications in entertainment technology and medical simulation. He has served on program committees of major graphics conferences including Eurographics, IEEE Vis, and ACM Siggraph/Eurographics SCA.