Corotational Simulation of Deformable Solids

Michael Hauth Wolfgang Strasser
WSI/GRIS, University of Tübingen
Sand 14, D-72076 Tübingen, Germany
{hauth,strasser}@gris.uni-tuebingen.de

ABSTRACT

The classical formulation of large displacement visco-elasticity requires the geometrically nonlinear Green tensor. Keeping track of the rotational part of strain permits alternative formulations, that allow the tensor to stay linear, and at the same time maintaining rotational invariance. We replace a recently proposed heuristical warping technique by the application of the polar decomposition. The polar decomposition exactly extracts rotations, thus enhances stability and accuracy. We combine it with a hierarchical finite element basis, which allows us to compute accurate rotations from a coarse level nonlinear simulation and use them with corotated tensors for finer detail.
Keywords: Deformable Objects; Polar Decomposition; Strain Tensors.

1. Introduction

Basing visual simulations on physical laws established its place in computer graphics with the work of Terzopoulos et al.[TF88]. Similarly to the present work the authors used methods derived form numerical engineering [ZT00; HW96].

The applications for interactive deformable objects are manyfold. Virtual surgery, as a prominent example, poses strict requirements on fidelity. Thin deformable objects are employed for the simulation of cloth, virtually dressing up synthetical actors or e-commerce customers. Whereas these simulations should be able to reflect real materials, applications like virtual avatars are less focused on accuracy. So are, at least at the moment, computer games. But they will include more and more physical realism, as soon as the necessary computing power becomes widely available. Mass-spring-damper systems, the most favored technique for a long time, allow physical constants like spring-stiffness, but are unable to model homogenous materials[VG98]. It is also not possible to model incompressibility or transverse contraction in the model without additional penalty forces.

For all these reasons the techniques from numerical engineering, namely finite element (FEM) and finite difference (FDM) methods continue to be the most versatile and accurate methods. Their drawbacks are the computational expenses. But by now there is sufficient computational power available to allow real-time simulations using physical accurate modelling.

Simulations especially become costly, when nonlinear systems are involved. Unfortunately, as soon as finite deformations are considered, the measures for strain become nonlinear, known as geometrical nonlinearity. This is closely linked to the invariance of the strain measure under rigid body rotations. The classical linear Cauchy strain, widely used for metal or concrete, is not rotational invariant. The usual solution is to use Green's tensor, which is nonlinear in the displacements. The rotation-problem became obvious during the 60s in the analysis of orbiting structures[dV76], and has been solved by a rotating coordinate system, compensating rigid body movements. Coupling a single rigid body frame and a deformable simulation has also been exploited by Terzopoulos[TW88]. A generalization is given by the corotational formulation[Fel00], attaching local coordinate systems to points or elements of the mesh. This idea recently found its way into computer graphics as a warping heuristics[MMD+02]. We will instead employ the polar decomposition to enhance accuracy and stability with minor additional costs. In addition, we base rotation estimates on a hierarchical finite element basis. This allows us to use a stable nonlinear simulation on a coarse level and corotated strains at the finer levels, resulting in an overall very stable algorithm.

2. Previous and Related Work

Although Terzopoulos used the metric tensor, a large displacement measure, to describe deformation, small

Journal of WSCG, Vol.12, No.1-3., ISSN 1213-6972
WSCG 2004, February 2-6, 2003, Plzen, Czech Republic.

displacement measures were common thereafter. One of the first medical applications was published by Bro-Nielsen[BN98], who used linear finite-elements with Cauchy's small strain tensor. Real-time animation based on explicit finite elements using Green's tensor has been presented by Debunne[DDCB01]. Capell et al.[CGC+02] use a hierarchial basis and linearize around a floating reference frame. Hierarchical bases, introduced into numerics during the 80's[Ban96], have been traced back by Yserentant[Yse92] to the beginning of the 20th century. The work on adaptive bases has been extended to the concept of a quasi-hierarchical basis[GKS02]. The idea of a floating reference frame is also exploited in the work of Müller et al.[MMD+02]. Instead of linearizing Green's tensor, they introduce a technique named warping to extrapolate this frame locally and use Cauchy's tensor around this warped coordinate systems.

The computer graphic society has been relatively unaware of the polar decomposition. The only reference we were able to find was by Showmake and Duff[SD92], who used it for keyframe matrix animation. Their problem shows some similarities, as they also needed to separate strain and rigid body descriptions. A similar application than the present, also using the polar decomposition, is currently evaluated for cloth simulations[KE]. The methods are related, although we discuss the general 3D case.

Considering time integration methods, more advanced work has been presented in cloth simulation. Since Baraff et al.[BW98], implicit methods became standard, allowing larger time steps than standard explicit methods, but paying the price of solving large linear systems. We employ a variety of implicit methods based on the framework presented in [HE01].

3. Outline

The following section will describe the physical modelling process. Section 5 gives an overview over the finite element techniques we tailored to our needs. The next section completes the simulator by briefly describing the time integrator. A section about the application, presenting some results, follows. We conclude by a section about the lessons learned and further directions of research.

4. The Physical Model

In this section, a mathematical description of strain and related material behavior is given, leading to a partial differential equation, which describes a deformable object. We will prove the rotational invariance of Green's tensor, and show how a rotated coordinate system allows the same for the linear Cauchy tensor.

4.1. Continuum Mechanics of Solids

Mathematically a deformable solid is given by its configuration mapping

$$\Phi_t : \mathbb{R}^3 \supset \Omega \rightarrow \mathbb{R}^3 \tag{4.1}$$

which maps each point in the material coordinates Ω to its position in 3-space at time t. It is common to identify the material coordinates with the body in its rest state, usually at time $t = 0$, thus $\Phi_0 = id$. Using this convention the configuration Φ_t can be split

$$\Phi_t = id + u_t \tag{4.2}$$

into the identity and a *displacement* field u.

From continuum mechanics[AG00], we know that the elastic energy arising from a displacement field u is given by the outer tensor product

$$E_{el} = \int_\Omega \sum_{i,j} \epsilon_{ij}(u)\sigma_{ij}(u) =: \int_\Omega \epsilon(u) : \sigma(u) \tag{4.3}$$

with the symmetric strain tensor ϵ and the symmetric stress tensor σ. Writing the tensors in standard matrix notation, this product can be defined as $\epsilon : \sigma := \mathrm{tr}(\epsilon^T \sigma)$. In most applications, stress and strain are related by a material law, mapping the current strain to the current stress

$$\sigma = C(\epsilon). \tag{4.4}$$

Thus we have now

$$E_{el} = \int_\Omega \epsilon : C(\epsilon) \tag{4.5}$$

All that remains is to decide about how to compute ϵ and what kind of material law to apply.

4.2. Hooke's Law

A straight forward approach leads to a linear relation between the two tensors, Hooke's law

$$\sigma = C\epsilon. \tag{4.6}$$

For isotropic materials only two of the 36 entries of C are distinct, the Lamé constants λ and μ. In component notation we have

$$\sigma_{ij} = \delta_{ij}\lambda \sum_k \epsilon_{kk} + 2\mu\epsilon_{ij}, \tag{4.7}$$

with the Kronecker δ.

For a dynamical simulation including inertia, we need viscous damping forces for the system coming to rest. We use the standard approach, which is to apply the visco-elastic correspondence principle and to formulate damping forces in the same way as the elastic forces, replacing ϵ in (4.4) by the strain rate tensor $\dot{\epsilon}$[OH99]. Simpler, linear or lumped Rayleigh damping forces, proportional to the linear velocity of each point, would damp rigid body modes. A more complex material law is frequency dependant visco-elasticity, which

still can be held (quasi-)linear, although with more variables, for a wide class of materials[GHEB01]. The techniques proposed below can be generalized to this setting.

4.3. Elastic Strain Tensors

There are two commonly used strain tensors for Lagrangian formulations, which use the parametrization over the rest state. These are the nonlinear Green strain tensor

$$\epsilon_{ij}^G(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \sum_k \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right). \quad (4.8)$$

and its linearization, the Cauchy strain tensor

$$\epsilon_{ij}^C(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (4.9)$$

Green's tensor can also be written in terms of the gradient of the configuration ϕ

$$\epsilon^G(\phi) = \frac{1}{2}\left(\nabla\phi^T\nabla\phi - id\right). \quad (4.10)$$

From this its invariance under rotations R can be seen, because

$$\epsilon^G(R\phi) = \frac{1}{2}\left(\nabla\phi^T R^T R\nabla\phi - id\right) = \epsilon^G(\phi), \quad (4.11)$$

with the orthogonality of R. This does not hold for Cauchy's tensor. It produces ghost forces which lead to distortions, as will be demonstrated in the examples.

4.4. Corotational Strain

Green's tensor leads to a stiff, nonlinear ODE, which is computationally expensive to solve. The corotational formulation aims at the elimination of the geometrical nonlinearity. The idea is to keep track of a rotated coordinate system of the body (fig. 1). It has first been employed with a single reference frame for the entire structure, when the analysis of orbiting space and aircraft structures made such techniques obviously necessary[dV76]. Unfortunately a single frame is not enough. Considering a long bar as in fig. 2, fixed at one end and bent at the other, shows that it is not sufficient to keep track of a single rigid body movement. The attached part is not rotated but the other end may undergo arbitrarily large rotations.

For the moment, we suppose that, at each point we know the local rigid body rotation $R(x)$. Then the corotational Cauchy stress

$$\epsilon^{CR}(\phi) := \epsilon^C(R^T\phi) \quad (4.12)$$

is a trivially rotationally invariant strain description and still linear. The arising forces are then transformed back by R. Note that in contrast to Müller et al.[MMD+02] we will discretize the rotation field $R(x)$ per element rather than per vertex. This results in a finer representation of the tensor field and is more consistent with the finite element modelling paradigm.

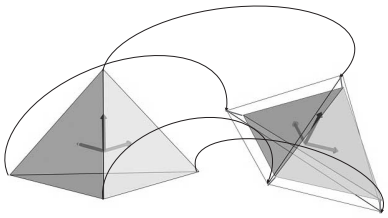

Figure 1: Separating rotation and deformation by a rotating reference frame.

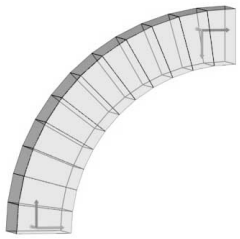

Figure 2: Non-moving bar with large rotations.

4.5. Extracting the Rotation Field

Another major difference to the work of Müller is the way the rotations are extracted. They 'found that the stability is not sensitive to the rotation field' and therefore use a heuristics called warping. It extracts rotations based on the deterministic selection of three edges per point of the mesh and tracks their rotations. This is sufficient for a rigid body movement superimposed on a deformed configuration, because the warping roughly follows it and therefore does not produce large ghost forces.

However, accuracy suffers from this approximation, as it is not directly clear, how the rigid body mode of an arbitrarily deformed element looks like. A way to extract a well defined rotation, is given by employing the polar decomposition of the deformation gradient $J := \nabla\phi$. It solves the following problem

$$\text{Find an orthogonal } R \text{ minimizing } \|J - R\|_F^2, \quad (4.13)$$

with the Frobenius norm $\|\cdot\|_F$. It gives rise to a decomposition $J = RS$, and therefore reduces (4.10) to

$$\epsilon^G(\phi) = S^T S, \quad (4.14)$$

where S does not contain any rotational component. The factor R is the rotation closest to the deforma-

tion gradient in the space of matrices equipped by the Frobenius norm[SD92].

In two dimensions it is straight forward to compute

$$R' = J + \text{sign}(\det(J)) \begin{bmatrix} J|_{22} & -J|_{21} \\ J|_{12} & J|_{11} \end{bmatrix}, \quad (4.15)$$

and a subsequent normalization of the columns gives R. In higher dimensions, Higham[HS90] proposed an efficient, quadratically convergent iteration scheme

$$R^{(0)} := J \quad (4.16)$$

$$R^{(n+1)} := \frac{1}{2}\left(R^{(n)} + {R^{(n)}}^{-T}\right). \quad (4.17)$$

Because the deformation gradient may be singular, e.g. in case of a pure rotation around one of the coordinate axes, we use a QR decomposition ahead of the core algorithm as also proposed by Higham. This results in a very robust and fast algorithm. We seldomly need more than three iterations of (4.17).

4.6. Principle of Virtual Work

By minimizing the energy given in (4.5), i.e. equating its first variation to 0, adding body forces $b(x)$ and surface traction $t(x)$, inertia and viscous damping forces, we arrive at the principle of virtual work

$$\int_\Omega \delta u \rho \ddot{u}\, dx + \int_\Omega \delta\epsilon : C(\epsilon)\, dx + \int_\Omega \delta\dot{\epsilon} : D(\dot{\epsilon})\, dx - \int_\Omega \delta u\, b\, dx - \int_{\delta\Omega} \delta u\, t\, ds = 0, \quad (4.18)$$

given as a weak partial differential equation.

5. Spatial Discretization

In this section we describe briefly, how we transform (4.18) into an ordinary differential equation, using finite elements. For a full description we refer to standard textbooks like Zienkiewicz and Taylor[ZT00].

5.1. Standard Finite Elements

In a standard finite element approach the functions u and δu are replaced by piecewise polynomial approximations over a decomposition T of Ω into disjoint elements of simple shape. Because of good shape-fitting properties, tetrahedra are usually preferred in computer graphics. Our discretization employs linear shape functions ϕ_i, resulting in a piecewise affine approximation. Therefore u will be approximated by

$$u(x,t) = \sum_{i=0}^{N} \mu_i(t)\phi_i(x), \qquad \mu_i(t) = u(x_i, t). \quad (5.1)$$

Inserting (5.1) into the principle of virtual work (4.18) and using $\delta\epsilon = \partial\epsilon/\partial\mu, \delta u = \partial u/\partial\mu$ leaves the ordinary differential equation

$$M\ddot{\mu}(t) + K\mu(t) + D\dot{\mu}(t) - \hat{b} - \hat{t} = 0, \quad (5.2)$$

with the mass matrix $M = [\int_\Omega \phi_i \rho \phi_j]$, where ρ is the mass density, the nonlinear operators

$$K\mu(t) = \int_\Omega (\frac{\partial}{\partial\mu}\epsilon) : C(\epsilon), \quad (5.3)$$

$$D\dot{\mu}(t) = \int_\Omega (\frac{\partial}{\partial\mu}\dot{\epsilon}) : D(\dot{\epsilon}), \quad (5.4)$$

and the load vectors

$$\hat{b} = [\int_\Omega \phi_i b] \quad \text{and} \quad \hat{t} = [\int_{\delta\Omega} \phi_i t]. \quad (5.5)$$

All integrals are evaluated locally and piecewise over the elements, then summed up. Because the basis $\{\phi_i\}$ is linear, ϵ is constant over each tetrahedron, greatly reducing the cost for the evaluation of K and D. Arranging the computations carefully gives a cost of around 300 flops for Green's tensor per element for the evaluation of each operator.

Using the corotational tensor ϵ^{CR} (5.6) transforms (5.3) to the now linear operator

$$K\mu(t) = \sum_{t_i \in T} R_i \int_{t_i} (\frac{\partial}{\partial\mu}\epsilon^C(R_i^T u) : C(\epsilon^C(R_i^T u)), \quad (5.6)$$

also evaluated over the triangulation and then summed up. Besides being consistent with finite elements, using a rotation per element keeps the Jacobian of the system symmetric, whereas using a rotation per node breaks symmetry.

To transform (5.2) into an explicit second order ODE in standard form, the mass matrix M needs to be inverted. Because this is very costly, even with precomputed decompositions, we apply row sum mass lumping. The price of mass lumping is a slight artificial increase of viscosity depending on the mesh[ZT00].

5.2. Hierarchical Approximations

For a multi-resolution representation we employ a hierarchical basis[Ban96], in order to be able to use different strain measures on different levels. For a given coarse tetrahedral discretization a series of nested meshes is constructed by octasection of each tetrahedron. For a hierarchical representation the function now is described by its values at the nodes of the coarsest level and the difference between the nodal value at the next finer level and the linear interpolation of its two parent nodes. A key to the efficiency of algorithms linked to this kind of hierarchical representations are

the fast basis transformations. Using the efficient O(N) wavelet transformation [Yse92], is it possible to carry out all computations using the wavelet approximation of u with minimal changes to the code. For details of our implementation we refer to Hauth et al.[HGS03].

6. Implicit Time Integration

A detailed analysis of (5.2) reveals, that all the eigenvalues of the linearisation for $t > 0$ are located in the negative complex halfplane, thus the problem can be considered stiff. A physical interpretation of this fact is, that the system always has a bounded solution, and, if only a limited amount of energy is transferred into it, it converges to a rest state.

Numerical stability analysis[HW96] characterizes stable integrations methods, that deliver a bounded solution, by their stability region, which covers parts of the complex plane. All the eigenvalues of the system, scaled by h must be located inside this region. Unfortunately all explicit methods possess a bounded and usually small stability region near zero, therefore h must be sufficiently small to fulfill the stability restrictions. Many of the implicit methods possess an unbounded stability region covering the whole complex halfplane with negative real parts. In this case, there is no artificial limit to the time step h, allowing arbitrary large steps $h > 0$[HW96; BW98]. The drawback of implicit methods is the solution of large and in the case of Green's tensor nonlinear equation systems. Usually Newton's method is used, involving the Jacobian of (5.2) with respect to the basis coefficients. The advantage for the corotional formulation now becomes obvious: the systems to be solved in each time step remain linear. This gives a speedup from about 2 to 10, depending on the original number of Newton iterations. Compared to classical linear finite elements the linear systems change over time, whenever the reference frames are updated.

We built a layered solver core as described in Hauth et. al.[HE01], employing several integration formulas and a Newton method with backtracking. In the example section backward Euler and BDF(3) will be used. For the linear solver a cg method with a block diagonal preconditioner or a sparse LU-factorization with reordering is used[DD99]. The Gauss method usually is faster than cg for small systems. A small drawback is that factorizations are expensive, whereas solves are fast. This can result in jittering frame-rates for large systems.

7. Application Details and Results

We implemented a prototype application based on the framework laid out above. All tests were performed on a Desktop P4/Willamette-2000. There is still room

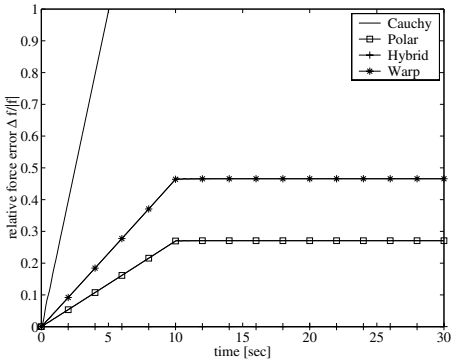

Figure 3: Example 1: Pressure Forces.

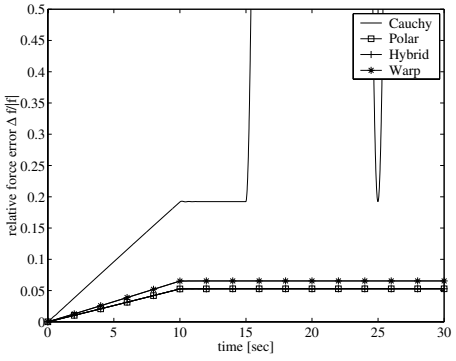

Figure 4: Example 1: Shear Forces.

for improvements. For example the corotational formulations use no special path. So for the implicit solver the Jacobian of the affine mapping is computed and for each force evaluation, the contributions from each tetrahedron are summed up again. Exploiting the already computed Jacobian would speed this up by roughly a factor of two, as the force computations are a major part of the workload. Memory consumption is moderate, the hierarchical bar (example 3) takes a total of 16 MB, the liver takes 18 MB, including texture.

7.1. Example 1: Forces

To compare the quality of the warping heuristics (applied per element) and the polar decomposition we performed the following experiment: Using Green's tensor for simulation we rotated a cube of 0.1m side length ($\mu = 10^3, \nu = 0.45$) with $\omega = 0.63s^{-1}$ performing a single rotation in 10s. Then the cube is sheared with a constant force on the top surface. After coming to rest, it is again rotated. The resulting local coordinate systems are shown in the first video. They pass the visual test of rotating globally with the rigid rotations and locally with the shear deformation.

For the second video, the internal stress forces from Green's (black), Cauchy's (red), corotation/polar (green) and warped tensors (violet) were computed. Ideally all should match the Green model. During

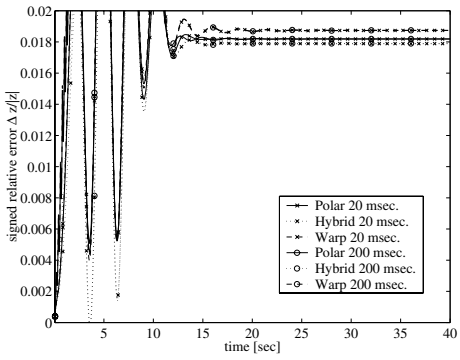

Figure 5: Example 2. Shearing.

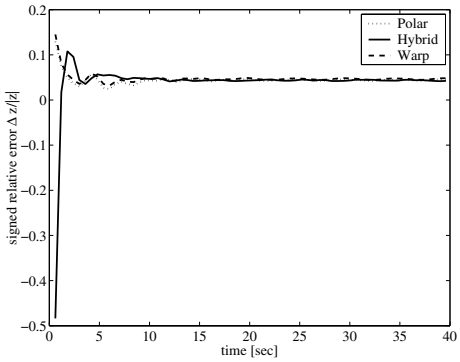

Figure 6: Example 2. Shearing and rotating.

the first rotation with no deformation you can see the phantom forces from Cauchy's tensor, which will be switch off for the rest of the video. All other forces are zero. During the shear movement the corotated and warped forces split from the Green forces. This is due to the fact, that Green's tensor is a second order model of strain, whereas the other ones are linear. The polar decomposition roughly halves the error compared to warping. Of course all are rotational invariant.

In figs. 3 and 4 we plotted the relative error $\|f - f^G\|/\|f^G\|$ of the different force contributions, leaving away the first 10s. The picture confirms the above observations, and shows that the error is especially large in the pressure component. Therefore we introduced a hybrid tensor shown in blue, combining Green's tensor for pressure and the corotated tensor for shear forces. This is very cheap in terms of force computations, but of course makes the system nonlinear again. For this tensor, the Jacobian is updated correctly including the nonlinear terms, whenever the rotation frame is updated.

7.2. Example 2: Displacements

This example employs a bar similar to fig. 2, of size 0.1x0.1x0.6m^3, with $\lambda = 10^5, \mu = 10^4$, $d\lambda = 10, d\mu = 1$ and a time step of 20ms with a Gauss

	CPU time [s]	Newton	evals	LU dec. (Jacobi)
time step 20ms, update rate 20ms				
Green	16.5	6490	21276	105
Warped	27.4	-	2000	2000
Polar	31.4	-	2000	2000
Hybrid	33.8	3092	3092	2000
time step 20ms, update rate 200ms				
Green	16.5	6490	21276	105
Warped	5.5	-	2000	201
Polar	5.9	-	2000	201
Hybrid	22.5	9981	10538	422
time step 60ms, update rate 300ms, rotating				
Green	13.6	4489	16688	202
Warped	2.7	-	667	135
Polar	3.0	-	667	135
Hybrid	22.4	7142	21576	411

Table 1: Runtime Details for Example 2.

solver. The bar is discretised in 337 tetrahedra. A shear force rising linearly from 0 to $200N/m^2$, is applied to the top surface, and this time the simulation is performed which each tensor separately. As an error measure we chose the relative error in the z-component of the upper right front corner (compared to Green). The results (fig. 5) are similar to those from above, identifying the hybrid formulation as best, warping as worst. This does not change, when the rotated frame is updated less than once per time step, eg. only every 200ms. This causes Gauss- or preconditioner updates to be less frequent and makes the cost for computing the reference frame negligible (table 1).

To bring the method to its limit, we again performed this test, but imposing a rotation, choosing a time step of 60ms and updating the frame only every 300ms (about every 10 degree of rotation, fig. 6). This brings the methods very close together, showing again a favor for the tensors based on the polar decomposition. Now after reaching the rest position the bar still vibrates, resulting in an oscillating error. The hybrid tensor suffers from the bad frame, combined with an inaccurate Jacobian, due to the nonlinear pressure forces. Delaying updates more than 0.5s (or about 20 deg.) makes the linearized methods unstable, whereas using Green's tensor the Newton iterations increase but the system remains stable. Without the rotation, the corotational formulations stay stable for larger update rates and time steps, and with Green's tensor one can even use steps of 1s and above.

From this examples we conclude, that the corotational formulations perform well, especially when the rigid body movements are not too fast compared to the time step. For 337 tetrahedra this gives up to 13x realtime, or a factor of 3-4 faster than the nonlinear simulation. The polar decomposition helps to cut down

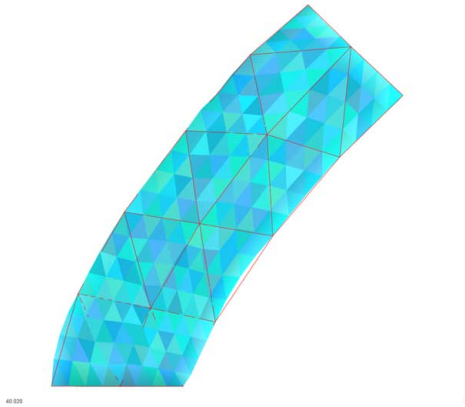

Figure 7: Example 3. 70/4480 Tetrahedra.

the induced error, at additional computational costs of less than 10%. The warping heuristics does a reasonable job, but is not able to really extract the essence of the rotation, especially obvious when comparing the forces.

7.3. Example 3: Hierarchical Simulation

The next example addresses the use of corotational tensors in a hierarchical setting. For the coarsest level, Green's tensor is used, allowing a very stable simulation. From the configuration of the coarse level, the rotation state is computed by the polar decomposition of the deformation gradient, then propagated to the finer levels. If necessary, this could be iterated, recomputing rotations at several levels. Here we just do it for the top level. Thus at the coarse level, where linear algebra is cheap, we perform a full nonlinear simulation, at the finer levels the system is linearized and now solved by cg, adding detail for reduced computational costs. Nevertheless the computational costs of the finer levels dominate.

The bar (0.1x0.1x0.3m^3,$\mu = 10^3, \nu = 0.3$) shown in fig. 7 is discretized by 70 tetrahedra, two octasection steps give 4480, i.e. $8^2 \times 70$, fine elements. As before we add a rising top shear force of $80N/m^2$. We rotate the bar as shown in the video during the 10s simulation, with a time step of 30ms and an update rate of 150ms for the coordinate systems. The specular highlights in the video and the silhouette in fig. 7 show the smooth deformation gained by the fine discretization. The coarse surface is given by the red wireframe. The computation time of 11.2s is almost real-time. It takes 6693 preconditioned cg steps, i.e. about 20 iterations per time steps to smooth out the error at the finer levels. For a comparison a complete nonlinear simulation was computed in 71.7s, the relative deformation error at the corner was 16%. This is larger than in the previous examples, because we are using the same frame for 64 tetrahedra.

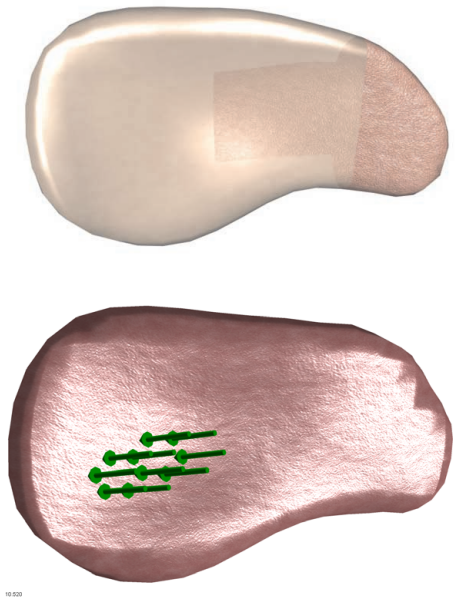

Figure 8: Example 4. Liver w. 327/2616 Tetrahedra.

7.4. Example 4: Deforming a Liver

The last experiment shows an application in virtual medicine. It employs a liver model (about 0.2x0.3x0.2cm^3, $\mu = 10^3, \nu = 0.495$, roughly as measured in [GHEB01]), shown in fig. 7 at the top. The opaque parts possess homogenous Dirichlet boundary condition, i.e. are fixed. The initial tetrahedralization has 327 elements, we add a single level with 2616 tetrahedra. The liver is pushed in the region marked by the green arrows. Again thanks to the fine level the surface stays very smooth, compared to the coarse initial disrcetization, which can be seen at the fixed parts and the silhouette.

For a simulation of 40s the application spent 25.8s, the time step was 30ms using BDF(3). The video discloses one small drawback of using a large number of coarse tetrahedra. About every second, the update of the rotation frames, which recomputes the linearized corotated Jacobian and the update of the coarse nonlinear Jacobian, including its LU-decomposition, coincide with each other, giving an unpleasant speed shift. This could be cured by just delaying one of the tasks, or by using a coarser top level mesh, such that the decomposition is faster.

8. Conclusion and Future Work

In this paper we presented a system for the visual simulation of deformable models. A corotational formulation keeps the systems to be solved linear, especially

important for fine discretizations, as the cost for solving linear systems grows polynomially with the dimension. The most significant contribution is the use of the polar decomposition for extracting the rotation state and its application in a hierarchical finite element setting. We compared our approach to the recent work of Müller et. al. [MMD+02]. In addition we examined the error that is arising from using the corotational approximation instead of the nonlinear tensor. This point has not been addressed in the previous work.

The examples show, that the corotational approximation induces an error of about 5-20%, but can give a speedup of about a factor of seven or more, when applied in the hierarchical case. The use of the polar decomposition instead of the warping heuristics helps to decrease the introduced error. Using a hybrid tensor can further reduce the error, but computation times rise due to the reintroduced nonlinearity. The stability is enhanced when comparing to the admissible time steps published in [MMD+02], which were all around 10ms or below. We use step sizes of 20-60ms in the above examples and are stable for 100ms and beyond. Especially the hierarchical approach benefits from its unconditionally stable top level.

Future work will look at adaptivity and better subdivision schemes, not suffering from the fast factor-of-eight growth. Also the use of adaptive mesh coarsening could alleviate the jittering effects seen in the last example.

9. Acknowledgements

The authors thank R. Kuchar and T. Schairer for modelling and multimedia assistance, M. Wand for always making a good point and the anonymous reviewers for their valuable comments. This work has been partially funded by the DFG ElastoMedTrain grant.

10. References

[AG00] T. M. Atanackovic and A. Guran. *Theory of Elasticity for Scientists and Engineers*. Birkhäuser, Boston, 2000.

[Ban96] Randolph E. Bank. Hierarchical bases and the finite element method. *Acta Numerica*, pages 1–43, 1996.

[BN98] M. Bro-Nielsen. Finite Element Modeling in Medical VR. *Journal of the IEEE*, 86(3):490–503, 1998.

[BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, pages 43–54, July 1998.

[CGC+02] Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popovic. Interactive skeleton-driven dynamic deformations. In *SIGGRAPH 2002*, pages 586–593. ACM SIGGRAPH, 2002.

[DD99] Timothy A. Davis and Jain S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Trans. Math. Softw.*, 25(1):1–20, 1999.

[DDCB01] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic Real-Time Deformations using Space and Time Adaptive Sampling. In *SIGGRAPH 2001*, pages 31–36, 2001.

[dV76] B.M. Fraeijis de Veubeke. The dynamics of flexible bodies. *Int. J. Engrg. Sci.*, pages 895–913, 1976.

[Fel00] C.A. Felippa. Nonlinear finite element methods. Lecture Notes of The University of Colorado, http://caswww.colorado.edu/courses.d/NFEM.d/Home.html, 2000.

[GHEB01] J. Gross, M. Hauth, O. Etzmuß, and G. F. Buess. Modelling Viscoelasticity in Soft Tissues. In *Int. Workshop on Deformable Modelling and Soft Tissue Simulation*. Elsevier, November 2001.

[GKS02] E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A Simple Framework for Adaptive Simulation. In *SIGGRAPH 2002 Conference Proceedings*, pages 281–290. ACM Press/ACM SIGGRAPH, 2002.

[HE01] M. Hauth and O. Etzmuß. A High Performance Solver for the Animation of Deformable Objects using Advanced Numerical Methods. In *Proc. Eurographics*, pages 137–151, Manchester, UK, 2001.

[HGS03] M. Hauth, J. Gross, and W. Straßer. Interactive physically based solid dynamics. In *Proc. SIGGRAPH Symposium on Computer Animation 2003*. ACM Press, 2003.

[HS90] Nicholas J. Higham and Robert S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM J. Sci. Stat. Comput.*, 11(4):648–655, 1990.

[HW96] E. Hairer and G. Wanner. Solving Ordinary Differential Equations II. Springer-Verlag, Berlin, 1996.

[KE] Michael Keckeisen and Olaf Etzmuss. Cloth Simulations for Virtual TryOn. Private communications.

[MMD+02] M. Müller, L. McMillan, J. Dorsey, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the ACM SIGGRAPH Symposium on Computer Animation*, pages 49–54. ACM Press, 2002.

[OH99] James F. O'Brien and Jessica K. Hodgins. Graphical Modeling and Animation of Brittle and Fracture. In *SIGGRAPH 1999*, pages 137–146, 1999.

[SD92] Ken Shoemake and Tom Duff. Matrix animation and polar decomposition. In *Proceedings of Graphics Interface '92*, pages 258–264, May 1992.

[TF88] D. Terzopoulos and K. Fleischer. Deformable Models. *The Visual Computer*, 4:306–331, 1988.

[TW88] Demetri Terzopoulos and Andrew Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, 8(6):41–51, November 1988.

[VG98] Allen Van Gelder. Approximate Simulation of Elastic Membranes by Triangle Meshes. *Journal of graphics tools*, 3:21–42, 1998.

[Yse92] H. Yserentant. Hierarchical bases. In *ICIAM91*, pages 281–290. SIAM, 1992.

[ZT00] O.C. Zienkiewicz and R.L. Taylor. *The finite element method. Vol. 1-3.* Oxford: Butterworth-Heinemann., 2000.