Workshop On Virtual Reality Interaction and Physical Simulation (2005)
F. Ganovelli and C. Mendoza (Editors)

A Multigrid Framework for Real-Time Simulation of Deformable Volumes

Joachim Georgii and Rüdiger Westermann

ComputerGraphics & Visualization Group, Technische Universität München [†]

Abstract

In this paper, we present a multigrid framework for constructing implicit, yet interactive solvers for the governing equations of motion of deformable volumetric bodies. We have integrated linearized, corotational linearized and non-linear Green strain into this framework. Based on a 3D finite element hierarchy, this approach enables realistic simulation of objects exhibiting an elastic modulus with a dynamic range of several orders of magnitude. Using the linearized strain measure, we can simulate 50 thousand tetrahedral elements with 20 fps on a single processor CPU. By using corotational linearized and non-linear Green strain, we can still simulate five thousand and two thousand elements, respectively, at the same rates.

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

1. Introduction

Over the last decade, interactive, yet physics-based simulation of deformable volumetric bodies has received increasing attention in a number of applications. Especially in medical applications there is an ongoing demand for ever realistic simulations of such objects. Popular examples include plastic and reconstructive surgery, breast augmentation or virtual training simulators. In these scenarios, physical correctness is often sacrificed for efficiency, resulting in approximate behavioral simulations or in the restriction to non-realistic material properties and small deformations.

The reason why interactive and physics-based simulation of reasonably sized volumetric bodies is still difficult to achieve is twofold: First, the most efficient techniques known so far require ever smaller simulation time steps with increasing material stiffness. For the simulation of real-world volumetric bodies, which exhibit an elastic modulus with a dynamic range of several orders of magnitude, these techniques in general can not fulfill both the requirements on frame rate and numerical stability. Second, even for simple abstractions, calculations involved in stable techniques are usually too expensive as to allow for real-time simulations of such bodies.

In this work, we present a multigrid framework for constructing implicit and stable solvers for the governing equations of motion of deformable volumetric bodies. This framework provides an interactive means for simulating the dynamic behavior of an elastic solid under external forces, and it is open to a variety of different formulations of strain (see Figure 1). Independent of the formulation used, by tak-

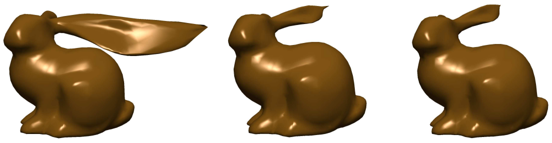

Figure 1: *Deformations of a tetrahedral model under the same external forces are shown. From left to right, linear Cauchy strain, corotational linear Cauchy strain, and non-linear Green strain is simulated. Because the Cauchy strain is not invariant under rotations, it introduces artificial forces. Very close results are obtained using the corotational and the non-linear formulation of strain.*

[†] georgii@in.tum.de, westermann@in.tum.de

ing advantage of a finite element hierarchy the proposed solvers are significantly faster than previous solution methods. If the linear strain measure is used, about 50 thousand tetrahedral elements can be simulated at 20 fps on a single processor CPU. At the same rate, up to five thousand elements can be deformed using the corotational multigrid simulation of the linear Cauchy strain. Even the simulation of non-linear Green strain can be performed interactively for a few thousand elements. The latter alternative is of particular interest in applications where internal stress becomes large and the corotational formulation of strain tends to produce artificial forces, e.g. if immediate or abrupt forces are applied. Due to the linear asymptotic complexity of the multigrid solvers proposed, with increasing problem size an ever increasing performance gain compared to previous approaches is achieved.

The implicit nature of the presented framework makes it amenable to the simulation of heterogeneous volumes exhibiting a wide range of stiffness characteristics. An example is shown in Figure 2.

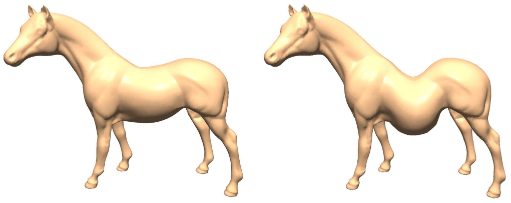

Figure 2: *Both images show a tetrahedral horse model under the influence of gravity. The model on the left is simulated using an elastic modulus of* $5\cdot10^{10}N/m^2$ *(aluminium) for all elements. On the right, the abdomen of the horse was soften using an elastic modulus of* $10^4N/m^2$ *(organic matter), while all other parts of the model remain stiff.*

This property makes the framework amenable to the deformation of real-world objects, in which an elastic modulus with a dynamic range of several orders of magnitude is not unusual to be found. These parameters considerably affect the objects dynamic behavior, which makes them important in a number of applications ranging from surgery simulation and image registration to solid mechanics. Besides physical realism, such parameters also provide a plausible means for controlling the behavior of arbitrary bodies as they allow for a flexible and realistic setting of deformation characteristics.

2. Related Work

To study the motion of a mechanical system caused by external forces, physics-based simulation is needed. For a set of connected rigid or flexible parts exhibiting material dependent properties the equations of motion can be formulated and solved to predict the dynamic behavior of such systems [Bra01]. In computer graphics and medical applications, a variety of interactive approaches for simulating such systems have been derived. From a large scale perspective, these techniques can be classified according to the underlying object discretization, the object's intrinsic deformation behavior, i.e. strain measure, and the method employed to integrate the equations of motion over time (see [GM97, NMK*05] for thorough overviews of the state of the art in this field).

In this work, we employ finite element methods [Bat02] to derive numerical solvers for the governing equations of motion of deformable volumetric bodies. Based on a discretization of the body into a set of elements, e.g. linear tetrahedral elements, boundary elements [JP99] or finite volumes [TBHF03], the solution of the equations to be solved on the domain is then characterized by parameters of these elements.

Implicit solution methods require the assembling of all element equations into a large system of algebraic equations, which can then be solved using matrix preinversion [BNC96] or the conjugate gradient method [MDM*02, EKS03, HS04, MG04]. An acceleration method was proposed in [CDA99], where a precomputed linear elastic model is interpolated at run-time. Besides the use of implicit methods in finite element simulations, they have also been employed in finite difference and mass-spring models [TPBF87, LTW95, BW98, DSB99] to enable stable simulations even for large time steps.

While the mentioned approaches consider the linear strain measure, i.e. the Cauchy strain, in [MG04, EKS03] a corotational formulation of the linear strain was used, which eliminates artifacts typically introduced by the Cauchy strain. In this method, the rotational part of the deformation is extracted for each finite element and the forces are computed with respect to the initial reference frame. In this way, stable and fast simulation results can be obtained. In contrast to an earlier approach, where the rotational part was extracted per vertex [MDM*02], the global stiffness matrix has to be reassembled at every time step.

Explicit finite element methods avoid the construction and solution of a large system of equations. Therefore, the non-linear Green strain can be integrated much more efficiently into these methods. Interactive simulation techniques using this measure have been presented in [ZC99, WDGT01, PDA01, DDBC01, ML03]. However, methods based on explicit time integration are limited due to the Courant condition, which significantly restricts the largest possible time step for very stiff materials.

To accelerate finite element methods, multiresolution techniques based on adaptive refinements have been proposed [CGC*02, DDBC01, GKS02]. An explicit multigrid scheme for the simulation of surface deformations was presented in [WT04]. To the best of our knowledge, an implicit yet interactive multigrid solver for volumetric bodies has not yet been developed.

3. Elasticity Theory

The motion of a deforming volumetric object can be simulated by a displacement field in an elastic solid. Given such a solid in the reference configuration $x \in \Omega$, the deformed solid is modeled using a displacement function $u(x), u : \mathbb{R}^3 \to \mathbb{R}^3$. This function describes the displacement vector at every point $\in \Omega$, yielding the deformed configuration $x + u(x)$.

3.1. Lagrangian Equation of Motion

Driven by external forces, the dynamic behavior of the deformed solid is governed by the Lagrangian equation of motion

$$M\ddot{u} + C\dot{u} + K(u) = f \tag{1}$$

where M, C, and K are respectively known as the system's mass, damping, and stiffness matrix. u consists of the linearized displacement vectors of all vertices and f is the linearized force vectors applied to these vertices.

By discretization of u,$\dot{u}$ and $\ddot{u}$ with respect to time, the differential equation can be transformed into a set of difference equations. To avoid artificial damping typical to implicit Euler integration, a second order accurate Newmark scheme is used for time integration:

$$\begin{aligned} \dot{u}^{t+dt} &= \dot{u}^t + \left(0.5\ddot{u}^t + 0.5\ddot{u}^{t+dt}\right) dt \\ u^{t+dt} &= u^t + \dot{u}^t\, dt + \left(0.25\ddot{u}^t + 0.25\ddot{u}^{t+dt}\right) dt^2 \end{aligned}$$

By discretizing u as well as the partial derivatives of u with respect to time, and by replacing $\dot{u}^{t+dt}$ and $\ddot{u}^{t+dt}$ in equation (1), the system of algebraic equations $\tilde{K}(u^{t+dt}) = \tilde{f}^{t+dt}$ is derived.

3.2. Finite Element Method

If a finite element method is used to model the system, system matrices are built by assembling all element matrices. Since each element in the finite element discretization only has a very small number of neighbors, this system is very sparse. We are using tetrahedral elements with linear nodal basis functions [Bat02]. Displacements are expanded in a basis of shape functions Φ as

$$u(x) = \Phi(x)u^e, \tag{2}$$

where $u^e = (\underline{u}_1^T, \ldots, \underline{u}_4^T)^T$ contains the single node displacements. The matrices M and C are derived from simple mass lumping and Rayleigh damping.

The stiffness matrix K accounts for the strain energy associated with the displacement field, and it is thus dependent on the elastic energy stored in the solid and on the work done by body forces and tractions applied through the displacement field u. The Green strain tensor $\mathcal{E}$ describes the non-linear relation between deformation and displacement:

$$\mathcal{E}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\sum_{k=1}^{3} \frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k} \tag{3}$$

In an isotropic and fully elastic body, stress ($\mathcal{S}$) and strain tensors are coupled through Hooke's law (linear material law)

$$\mathcal{S} = \lambda \left(\sum_{i=1}^{3} \mathcal{E}_{ii}\right) \cdot I_{33} + 2\mu\mathcal{E}, \tag{4}$$

with the Lamé coefficients λ and μ.

Given the nodal basis functions as well as per-node strain and stress, the potential energy

$$V = \frac{1}{2}\int_{\Omega} \sum_{i,j} \mathcal{E}_{ij}\mathcal{S}_{ij}\, dx$$

of every element can be computed. For a solid element to be in equilibrium, the first variation of V with respect to the per-node displacements has to vanish. The resulting single element equations are finally assembled into a system of non-linear equations.

3.3. Corotational Linear Strain

A common simplification is to neglect the quadratic terms in the definition of the strain tensor (3), yielding the (linearized) Cauchy strain tensor. Then, the assembling process results in a system of linear equations, where the matrix does not change during the simulation. While this approximation is appropriate for small deformations, for large deformations it leads to non-realistic displacements. Furthermore, as the Cauchy tensor is not invariant under rotations, incorrect forces are very likely to occur in the linear setting (see Figure 1 and 6).

A rotational invariant formulation of the Cauchy strain tensor is obtained using the so-called corotational strain of linear elasticity [MG04]. In this formulation finite elements are first rotated into their initial configuration before the strain is computed. In this way, although strain is still approximated linearly, artificial forces as they are computed using the Cauchy strain are significantly reduced. Rotations are calculated per element using a polar decomposition of the deformation gradient $\nabla(x + u(x))$ as proposed in [EKS03, HS04]. Once the rotation O^e of all finite elements are calculated, the element stiffness matrix K^e is replaced by $O^e K^e (O^e)^T$, and the global stiffness matrix is reassembled. A solution to equation (1) is then found by solving a system of linear equations with the updated system matrix K.

In the following we present a multigrid framework that significantly speeds up the simulation of all three different strain measures. In particular we show, that the solution of the system of linear equations with fixed and changing system matrix as in the linear and corotational setting, as well as the solution of a system of non-linear equations as in the non-linear setting take advantage of such a framework.

4. Multigrid Method

Multigrid methods provide a general means for constructing scalable linear solvers. Such methods exploit the fact that a problem can be solved on different scales of resolution. At the core of such approaches two observations are of major importance. First, a basic property of many iterative solvers for linear or non-linear systems of equations is smoothing. Many relaxation methods like Gauss-Seidel reduce high frequencies in the error very quickly, while low frequencies are damped rather slowly. Second, the remaining low-frequency errors can be accurately and efficiently solved for on a coarser grid. Both observations can be combined into a multigrid strategy that enables improved convergence at the same time avoiding any loss in accuracy, because only the smoothed error is transferred to the coarser grid. Recursive application of this basic idea to each consecutive system on a hierarchy of grid levels leads to a multigrid V-cycle.

For the efficient simulation of an elastic deformable solid we have developed a geometric multigrid method. In particular, this method includes geometry-specific relaxation, restriction, and interpolation operators. These operators form the essential multigrid components, as they are used to transfer quantities along the object hierarchy.

In this work, we define an appropriate finite element hierarchy, which allows for an efficient implementation of multigrid components. The result is a method that uniformly damps all error frequencies with a computational cost that depends only linearly on the problem size.

4.1. Unstructured Hierarchy

The geometric multigrid method requires a mesh hierarchy that represents the deformable object at different levels of resolution. On this hierarchy, appropriate transfer operators to map quantities between different levels have to be designed. Starting with a finite element mesh at the coarsest resolution level, a common way to construct the hierarchy in a top-down approach is to split each tetrahedron as shown in Figure 3. The octahedron is subsequently split into four tetrahedra, such that eight children are generated overall. This approach results in a nested hierarchy, which allows the transfer operators to be defined in a straight forward way, but it requires the initial mesh to be fine enough to achieve a proper representation of the object's boundary at ever finer resolution levels. Furthermore, subsequent subdivisions might lead to a fine mesh which contains ill-shaped tetrahedra that are not suited for finite element simulation.

To avoid these drawbacks, we propose linear transfer operators that do not require a nested hierarchy and can be integrated efficiently into the multigrid scheme. These operators establish relations in a multilevel hierarchy of unstructured and unrelated meshes by means of barycentric interpolation as illustrated in Figure 3.

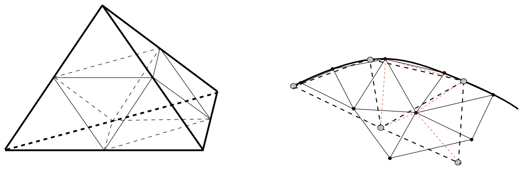

Figure 3: *Left: tetrahedral subdivision lends itself directly to a nested hierarchy. Right: geometric relations between elements in the non-nested hierarchy are illustrated (the 2D case is shown for simplicity). Dotted and solid lines indicate the coarse and the fine mesh respectively. Barycentric interpolation weights are highlighted by dotted red lines.*

Initially, we start with a coarse mesh H and a fine mesh h. For every tetrahedron in H, all vertices of h inside this tetrahedron are determined. The barycentric coordinates of these vertices with respect to the circumscribed element are calculated, and they are used as interpolation weights to map values from the coarse grid to vertices on the fine grid via the interpolation operator R_h^T. Each fine grid vertex stores the respective coarse grid vertices and corresponding weights. For vertices in h that lie outside the coarse mesh, barycentric coordinates to the closest tetrahedron in the coarse mesh are computed. The restriction operator that is required in the multigrid method to gather values from a finer resolution level is realized by inverse interpolation R_h.

We should also note that during the construction of the matrix hierarchy the Galerkin property [BHM00] is enforced. In particular, for all but the finest hierarchy level the system matrices are computed as

$$K^H = R_h K^h R_h^T.$$

The Galerkin property guarantees a consistent calculation on different levels of resolution, and it assures optimal convergence of the multigrid scheme.

It should be evident that the multigrid method computes the correct FEM solution on the entire mesh at the finest level. This is in contrast to other multiresolution techniques, e.g. [DDBC01], where the solution is computed adaptively for sub-meshes at different resolutions. This can lead to inconsistent deformations on different hierarchy levels, which is entirely avoided by the multigrid approach.

4.2. Linear Elasticity Multigrid

Given the linear transfer operator R_h, as well as an initial approximation $\hat{u}^h$ of the displacement values on the fine grid h of the deformable solid, a new approximation u^h can be computed as follows (e^h and e^H denote the error):

① compute residual	$r^h = f^h - K^h \hat{u}^h$
② restrict residual to coarse grid	$r^H = R_h r^h$
③ solution on coarse grid	$K^H e^H = r^H$
④ transfer correction	$e^h = R_h^T e^H$
⑤ correction	$u^h = \hat{u}^h + e^h$

The algorithm extends to a complete 2-grid approach by relaxation of $\hat{u}^h$ prior to stage ① to avoid the transfer of quantities from the h-grid that can not be reduced on the coarse H-grid, and by relaxation of the result u^h after stage ⑤ to avoid high frequencies introduced by numerical inaccuracies. In general, $1-2$ Gauss-Seidel steps are sufficient for both pre- and post-smoothing. By recursive application of the coarse grid correction to stage ③ and by using a preconditioned conjugate gradient method to compute the solution on the coarsest grid, a full multigrid V-cycle is derived.

4.3. Corotational Elasticity Multigrid

Because the integration of corotational strain leads to a system of linear equations, the multigrid method is essentially the same as for the Cauchy strain tensor. However, as the system matrix K on the finest level changes in every time step, the coarse grid matrices have to be rebuilt, too. Fortunately, as K is very sparse, a row-based index data structure can be utilized to significantly speed up this process. In addition, such a data structure reduces the number of numerical and memory access operations to ensure the Galerkin property on every hierarchy level.

4.4. Non-linear Elasticity Multigrid

The simulation of deformations based on the Green strain tensor using an implicit time integration scheme requires a non-linear equation system to be solved. To calculate a solution to this system we employ the Newton method, which is based on the first order Taylor approximation of the system of equations:

$$K(u+e) \approx K(u) + K'(u)e$$

Here, K' is the Jacobian matrix of K. Given an initial solution u, this solution can be corrected by solving the equation for e, which only requires a system of linear equations based on the Jacobian matrix to be solved:

$$K'(u)e = f - K(u),$$

For this system, the multigrid approach as described in chapter 4.2 is used.

To construct the non-linear system of equations $K(u)$, we utilize symbolic algebra operations in the preprocessing step. The set of all non-linear element stiffness equations is assembled symbolically into a system of non-linear equations. From equation (3) one can see, that $\mathcal{S}_{ij}\mathcal{E}_{ij}$ can be expressed in terms of nodal basis functions $\Phi(x)$. By first applying the material law to express $\mathcal{S}$ in terms of $\mathcal{E}$, $\mathcal{E}$ can then be expressed by the partial derivatives of u, where u is interpolated as shown in equation (2). Symbolic calculation of $\mathcal{S}_{ij}\mathcal{E}_{ij}$ results in a polynomial in the unknown variables u_i. These polynomials, which share a large number of monomials, can then be assembled into a global system of symbolic equations. The number of monomials to be evaluated in this system is significantly smaller (about a factor of 3) than the number of monomials contained in the set of element equations. Consequently, multiple evaluations of monomials can be avoided at run-time.

In the same way, the Jacobian matrix can be expressed symbolically, and then calculated using the current parameter values. The multigrid solver then updates the hierarchy taking into account the current evaluation of the Jacobian matrix. Although building the matrix hierarchy in every step is expensive both in terms of numerical and memory access operations, the multigrid solution is still up to 10 times as fast as a preconditioned conjugate gradient method.

To further improve the evaluation of polynomials at runtime, all required monomials $\prod_{i,j} u_i^j$ are computed from the current displacement vector u. As these monomials occur in several equations of the system as well as in the Jacobian matrix, an additional speed-up of about a factor of 2 is achieved.

5. Results

In the following, we give several examples that demonstrate the efficiency of the proposed multigrid method. The models used in these examples are shown in the Figures below. All experiments were run on a Pentium4 3.0 GHz processor equipped with 1GB RAM. As shown in Table 1, the multigrid method scales linearly with the number of elements, and it achieves excellent performance rates even for large models. Tetrahedral meshes with about 50k elements can be simulated interactively using the linear strain measure. Particularly in the last example, where a larger hierarchy allows the multigrid approach to deploy it's full potential, for the number of elements used the method is considerably faster than implicit approaches utilizing the conjugate gradient method. The star $*$ in all tables denotes the use of a non-nested grid hierarchy.

				tps	tps
Model	# level	# tet	# vert	mgrid	CG
Liver	2*	1467	464	720	30
Bridge	3	3072	825	460	2.4
Liver	3*	8078	1915	140	4.3
Breast	3*	10437	2542	120	1.5
Horse	4*	49735	12233	18	-

Table 1: *Timing results in time steps per second (tps) for different models using the linearized Cauchy strain measure. The multigrid solver is compared to a preconditioned conjugate gradient solver (CG).*

Compared to previous approaches, the implicit multigrid solvers enable much larger integration time steps of up to one second. Even more importantly, the time step does not depend on material stiffness. This property enables stable simulations of heterogeneous bodies with an elastic modulus varying from 10^3 N/m^2 to 10^{11} N/m^2. Figure 5 shows the influence of wind forces and gravity to bars of different density and stiffness. As the force is constant everywhere, softer bars are deformed much more significantly than stiffer ones.

As it is shown in Figure 6, the artifacts introduced by the linear strain measure can almost entirely be avoided by using the corotational formulation of linear strain. However, the performance gain of the multigrid approach is not as high as in the linear setting. This is due to the extra cost involved in updating the system matrices. As updating these matrices for a non-nested hierarchy (marked by a $*$ in the Tables) is much more costly than for a nested hierarchy, in case of a non-nested hierarchy only for larger meshes a significant speed up compared to [MG04] is achieved (see Table 2 and Figure 4). However, for stiffer materials the performance gain gets ever better as illustrated on the right of Figure 4. If a nested hierarchy is employed, however, matrix updates only require linear interpolation along element edges. As a matter of fact, a performance gain of up to 10 compared to the conjugate gradient method is achieved. Table 4 shows the time it takes to reassemble the system matrix in the corotational setting and to built up the matrix hierarchy used by the multigrid approach.

Model	# level	# tet	# vert	tps mgrid	tps CG
Bridge	2	384	153	150	110
Liver	2*	1467	464	21	18
Bridge	3	3072	825	19	1.9
Liver	3*	8078	1915	3.6	1.2
Breast	3*	10437	2542	2.8	0.7
Breast	3	10432	2541	6	0.7

Table 2: *Timings statistics for different models using the corotational simulation of linear strain. The multigrid solver is compared to a preconditioned conjugate gradient solver (CG)*

Compared to the linear setting, in the non-linear strain setting real-time can only be achieved if the number of elements is significantly reduced. However, compared to the corotational setting the performance is only about a factor of 2 lower (see Table 3). Explicit timings of the reassembling step in the non-linear setting are given in Table 4.

The efficiency and effectiveness of the non-linear solver is also demonstrated in Figure 7, where large and global deformations of about 3K tetrahedral elements including collision detection and response to static objects is performed in real-time. Figure 8 shows the application of the proposed simulation engine for breast augmentation. Deformations due to gravity as well as additional implants that are inserted into the breast can be simulated very realistically in real-time. In Figure 9 it is shown, that even the deformation of very large tetrahedral meshes is feasible on consumer class hardware by means of the proposed multigrid approach.

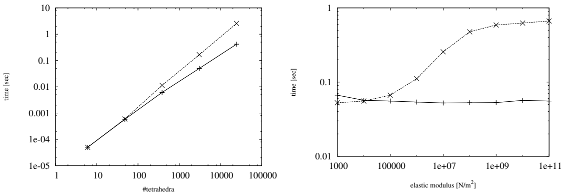

Figure 4: *Left: on a double logarithmic scale, timings are shown for the corotational simulation using the multigrid method including matrix reassembling and matrix hierarchy update (solid lines) and the CG method (dashed lines). Timings were measured using an increasingly refined tetrahedral cube model (elastic modulus = $2 \cdot 10^6 N/m^2$, integration time step = $0.02 sec$).*
Right: performance measures for a bridge model ($3k$ tetrahedra). For ever stiffer materials, the CG method requires more steps to be performed to achieve the same relative error of 10^{-4} as the multigrid method. The elastic modulus affects the performance of the CG method significantly, while it does not affect the performance of the multigrid method.

Model	# level	# tet	# vert	tps mgrid	tps CG
Bridge	2	384	153	95	60
Liver	2*	1467	464	13	8
Bridge	3	3072	825	12	1
Liver	3*	8078	1915	1.8	1.2
Breast	3	10432	2541	4.8	0.7

Table 3: *Timings statistics for different models using the non-linear Green strain measure. The multigrid solver is compared to a preconditioned conjugate gradient solver (CG)*

Model	# level	corot. reassemb.	non-linear reassemb.	rebuild mgrid
Bridge	2	4	9	1
Liver	2*	22	35	30
Bridge	3	42	72	17
Liver	3*	95	209	198
Breast	3	115	255	65

Table 4: *Timings in milliseconds for the most time-consuming parts of the multigrid solvers using the corotational and the non-linear formulation of strain.*

6. Conclusion

In this work, we have presented an implicit multigrid framework for interactive and physics-based simulation of deformable volumetric bodies, which is open to a variety of different strain measures. The proposed solvers effectively benefit from coarse grid correction in that they produce numerically stable results yet minimizing the number of iterations to be performed until convergence. The proposed multigrid solvers allow for the simulation of heterogeneous materials, *i.e.*,, materials exhibiting varying stiffness and density, without sacrificing speed or quality. Therefore, they have the potential to be integrated into real-time scenarios such as surgical simulators or virtual environments.

As the described multigrid framework benefits extremely from a nested hierarchy, in the future we intend to integrate more advanced meshing strategies into this framework. In particular, our goal is to generate nested tetrahedral meshes that are amenable to multiresolution finite element simulations at the same time adapting to the object boundaries [MBTF03].

To integrate the proposed simulation framework into practical applications, collision detection as well as topology changes, i.e cutting, has to be supported. Challenging approaches to detect self-collisions between parts of a deforming object have been presented recently [TKZ*04]. It will be of particular interest in the future to evaluate the integration of these methods into the multigrid framework. Topology changes, on the other hand, require the multigrid matrices to be recomputed. Although this is possible in general, it remains to be shown that these updates can still be done in real-time.

References

[Bat02] BATHE K.-J.: *Finite Element Procedures*. Prentice Hall, 2002.

[BHM00] BRIGGS W. L., HENSON V. E., MCCORMICK S. F.: *A Multigrid Tutorial, Second Edition*. siam, 2000.

[BNC96] BRO-NIELSEN M., COTIN S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Proceedings of Eurographics* (1996).

[Bra01] BRAESS D.: *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge Univ. Press, 2001.

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of SIGGRAPH '98* (1998).

[CDA99] COTIN S., DELINGETTE H., AYACHE N.: Real-time elastic deformations of soft tissues for surgery simulation. In *IEEE Transactions on Visualization and Computer Graphics* (1999), pp. 62–73.

[CGC*02] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIC Z.: A multiresolution framework for dynamic deformations. In *ACM SIGGRAPH Symposium on C. Animation* (2002).

[DDBC01] DEBUNNE G., DESBRUN M., BARR A., CANI M.-P.: Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of SIGGRAPH '01* (2001).

[DSB99] DESBRUN M., SCHRÖDER P., BARR A.: Interactive animation of structured deformable objects. In *Graphics Interface* (1999), pp. 1–8.

[EKS03] ETZMUSS O., KECKEISEN M., STRASSER W.: A fast finite element solution for cloth modelling. In *Pacific Conference on Computer Graphics and Applications '03* (2003).

[GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: Charms: a simple framework for adaptive simulation. In *Proceeding of SIGGRAPH '02* (2002).

[GM97] GIBSON S. F., MIRTICH B.: A survey of deformable models in computer graphics. In *Technical Report TR-97-19, Mitsubishi* (1997).

[HS04] HAUTH M., STRASSER W.: Corotational simulation of deformable solids. In *Proc. of WSCG '04* (2004).

[JP99] JAMES D. L., PAI D. K.: Artdefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 65–72.

[LTW95] LEE Y., TERZOPOULOS D., WATERS K.: Realistic modeling for facial animation. In *Proceedings of SIGGRAPH '95* (1995).

[MBTF03] MOLINO N., BRIDSON R., TERAN J., FEDKIW R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *IMR* (2003).

[MDM*02] MÜLLER M., DORSEY J., MCMILLAN L., JAGNOW R., CUTLER B.: Stable real-time deformations. In *ACM SIGGRAPH/EG symp. on Computer animation '02* (2002).

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proceedings of the 2004 conference on Graphics interface* (2004).

[ML03] MENDOZA C., LAUGIER C.: Simulating soft tissue cutting using finite element models. In *Proceedings of IEEE International Conference on Robotics and Automation* (2003).

[NMK*05] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. In *Eurographics '05* (2005).

[PDA01] PICINBONO G., DELINGETTE H., AYACHE N.: Nonlinear and anisotropic elastic soft tissue models for medical simulation. In *ICRA 2001* (2001).

[TBHF03] TERAN J., BLEMKER S., HING V. N. T., FEDKIW R.: Finite volume methods for the simulation of skeletal muscle. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2003).

[TKZ*04] TESCHNER M., KIMMERLE S., ZACHMANN G., HEIDELBERGER B., RAGHUPATHI L., FUHRMANN A., CANI M.-P., FAURE F., MAGNETAT-THALMANN N., STRASSER W.: Collision detection for deformable objects. In *Proc. Eurographics, State-of-the-Art Report* (2004).

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. In *Proceedings of SIGGRAPH '87* (1987).

[WDGT01] WU X., DOWNES M. S., GOKTEKIN T., TENDICK F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Proceedings of Eurographics '01* (2001), pp. 349–358.

[WT04] WU X., TENDICK F.: Multigrid integration for interactive deformable body simulation. In *Medical Simulation '04* (2004).

[ZC99] ZHUANG Y., CANNY J.: Real-time simulation of physically realistic global deformation. In *IEEE Vis '99* (1999).

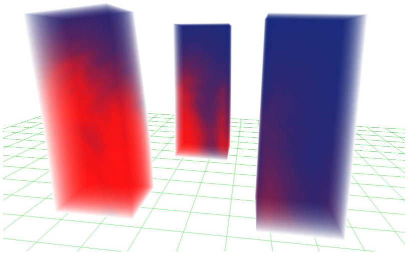

Figure 5: *A visualization of the internal states (*i.e.,*von Mises stress) of three towers exhibiting different stiffness is shown. A standard approach for direct volume rendering is used to generate the images. A constant wind force is applied to all models. Stress values are color coded ranging from red (high) to blue (low).*

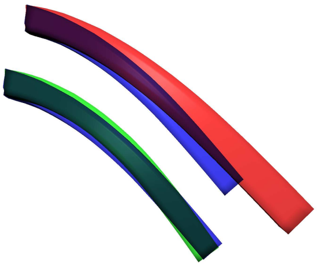

Figure 6: *Comparison of the linear, corotational and nonlinear strain measure. The deformation of the latter one are shown as reference in blue color. While the linear Cauchy strain (red color) fails to approximate the deformation properly, only very small differences can be observed in case of the corotatational strain (green color).*

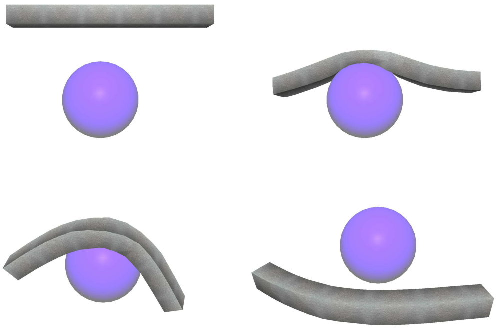

Figure 7: *A simple example demonstrates the potential of the proposed non-linear simulation engine. The bridge is discretized into 3K tetrahedral elements. Simulation and collision detection with static obstacles is run at 12 fps.*

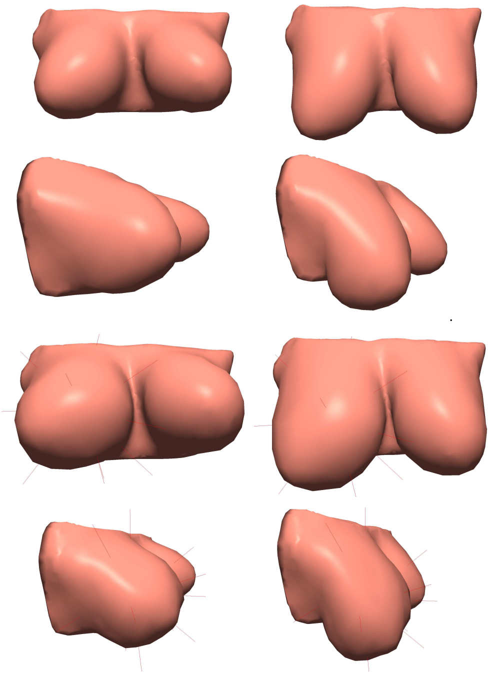

Figure 8: *Breast augmentation as one potential application of the proposed multigrid simulation framework is demonstrated. The simulation of gravity, different material properties as well as additional forces induced by implants can be simulated interactively at high physical accuracy.*

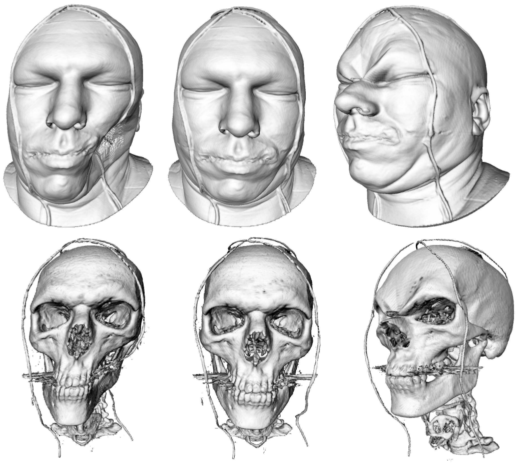

Figure 9: *Iso-surface visualization from a deformed CT data set. A tetrahedral simulation mesh was adaptively refined along the surface and bone structures in the 3D medical data set. This mesh consists of 1.1M tetrahedra, and it was deformed using the linear strain measure at 0.5 fps.*