EG UK Theory and Practice of Computer Graphics (2009)
Wen Tang, John Collomosse (Editors)

Fast and Accurate Finite Element Method for Deformation Animations

W. Tang[1], T. R. Wan[2], C. Niquin[1], A. Schildknecht[1]

[1]School of Computing, University of Teesside, UK
[2]School of Informatics, University of Bradford, UK

Abstract

We present a matrix clustering method for speeding up finite element computations for non-rigid object animation. The method increases the efficiency of computing deformation dynamics through a compression scheme that decomposes the large force-displacement matrix into clusters of smaller matrices in order to facilitate the run-time computations of linear finite element based deformations. The deformation results are compared with the results produced by using modal analysis method and the standard linear finite element algorithm. We demonstrate that the proposed method is stable with comparative computational speed to modal analysis method. A hierarchical skeleton-based system is also implemented to add constraints to material nodes. Thus, real-time deformations can be directed by motion captured data sets or key-framed animations.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism -Animation

1. Introduction

In the fields of biomechanical engineering and interactive medical simulations, accurate finite element computations are able to produce reliable and realistic physical behaviours of volumetric deformations by using only a few number of physically-based control parameters. The challenges associated with finite element based modelling are numerical instability for large deformations and high computational costs [O'BH99] [DDCB01] [MG04].

Accurate finite element deformations demand the use of the full Green-Lagrange strain tensor both linear and nonlinear terms, which are often seen in non-interactive animations and off-line simulations. Alternatively, linear Green-Lagrange strain tensor produces fast and relatively stable deformations, albeit with noticeable volume growth artifacts for large deformations. Techniques such as stiffness warping [MG04] and modal warping [CK05] remedy the volume growth artifacts by evaluating the rotational motion component in the local deformation frame or by incorporating a rotational warping operation into computations.

In order to address nonlinear dynamics, Grinspun [GKS02] proposed a multi-resolution method to approximate the strain-stress relationship for nonlinear deformations by interpolating multi-level linear hierarchical bases. Barbič and James [BJ05] extrapolate the linear tensor field into a nonlinear quadratic polynomial formula with coefficients pre-computed off-line, resulting extremely fast nonlinear dynamic simulations. However, modal analysis method is commonly used for interactive linear finite element computations, which approximate original elastic deformation models with a set of vibration modals around the deformation origin.

The numerical integration process is one of major sources of high computations in dynamic simulations. It is also common that deformable objects consist of a large number of degrees of freedom, making the update of each simulation step costly. In this paper, we present a matrix clustering approach to solving dynamics finite element system efficiently by decomposing the large system matrix into low dimensional sub-matrices. The new clustering formulation enables run-time evaluations of force-displacement computations more efficiently than evaluating the full dimensional space.

Principal component analysis (PCA) was applied to the large dense inverse matrix of force-displacement matrix to reduce the dimensionality of dynamic evaluations [SHHS03] [HLBG05]. PCA based matrix clustering methods are essentially loss compressions of large matrices by approximating the dynamics using a selected set of feature vectors. We propose a lossless clustering method to compress the original large sparse system matrix into a

DOI: 10.2312/LocalChapterEvents/TPCG/TPCG09/125-131

subset of matrices without inverting the matrix and approximations to the matrix. Thus deformations are accurate and versatile.

A skeleton-based constraint system is developed to apply constraints to volumetric structure nodes. The hierarchical skeleton-based system integrates conventional motion captured or key-framed animation datasets with finite element deformations for interactive design and manipulations. Another important factor in linear finite element simulations is that, for long and complex deformable objects, volume growth artifacts often occur due to material rotations and stretches. The constrained system also applies stiffness warping operations to deformation nodes to control the problem of volume growth.

2. Related work

Finite element computations are typically solved in discrete fashions by subdividing the continuum structure into finite elements, commonly tetrahedron structures. The system of dynamic deformations is described by the Euler-Lagrange equations as derived in the following general form:

$$M\ddot{u} + D\dot{u} + Ku = f_{ext} + f_{\psi} \quad (1)$$

where u being the displacement vector in the material coordinates, *M, D, K*$\in R^{3n\times 3n}$ being mass, damping and stiffness matrices of the dynamic system, which are constants for linear finite element models, $\boldsymbol{f_{ext}}$ being external forces acting on the object, such as gravity and contact forces, $\boldsymbol{f_{\psi}}$ being dynamically defined constraint forces during simulation, for example skeleton constraints and user defined constraints in our system.

$$u(x) = \sum_{i=1}^{4} \phi_i(x) u_i \quad (2)$$

The total internal elastic force exerted at each vertex node is the result of the continuum displacement within each of the elements that are connected to the node. The total internal elastic force is calculated by the linear interpolation of the displacement at each nodal vertex with a set of piecewise basis functions $\phi_i(x) : i = 1,2...n,$ where n being the total number of connected vertices.

2.1 Nonlinear simulation of linear materials

Terzopoulos and Fleischer [TF88] introduced a framework of using energy functions defined by tensors to model visco-elasticity and plasticity of deformations. Such direct integration scheme was too expensive for interactive applications. In adaptive multi-resolution approaches, hierarchical linear deformation bases were used to refine the deformations through adaptive volumetric subdivision

Figure 1: *A deformable human model with 4822 elements and a hierarchal skeleton structure of 48 DOFs is computed in real-time with a walking motion animation created by key frames.*

schemes [CGCDP02], or progressive mesh and level of detail tetrahedrons [DDCB01] [GKS02]. At each simulation step, with multi-resolution methods, a large part of the model is simulated with a coarse level of mesh details. Computational cost would increase dramatically when refining the overall mesh with much detailed subdivisions and interpolating more linear base functions, thus revoking the savings on the multi-resolution adaptive processes.

James and Pai [JP99] proposed a pre-computational strategy that allowed run-time evaluations of forces within reduced-dimensional subspaces for nonlinear deformations derived from a small set of pre-selected impulses. Barbić and James further improved the scheme by pre-computing coefficients of cubic polynomials of a nonlinear force model in the reduced model coordinate space [BJ05].

2.2 Linear approximation methods

Modal analysis was also an attractive alternative approach to computing finite element deformations. The techniques are essentially reduction computations of complex systems [BW92] [FPT97]

Modal analysis produces stable and efficient linear finite element deformations by linearly combining a small set of independent harmonic oscillators around the deformation origin [PW89] [JP02]. Modal warping method [CK05] was used in linear modal analysis in order to counteract distortions presented in linear models.

Our matrix condensation method exploits the spatial coherence of the stiffness matrix with a lossless computation of deformation dynamics without the need to approximate the dynamic model. The method produces accurate finite element based deformations with interactive speed. Stiffness warp is used to reduce volume growth presented in linear models. In the same spirit of previous work [CGCDP02], a number of different constraints are also applied to the dynamic system, including hierarchical rotational joint constraints, point constraints and line constraints. Figure 1 demonstrates the use of our constrained system in conjunction with finite element based deformations with animated motion data. As can be seen, constrains can also be used to control the volume growth problem for the long deformable objects, such as limbs and legs of the characters.

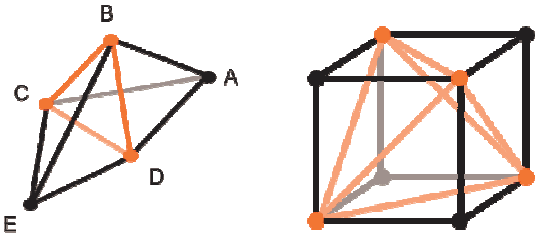

Figure 2: *A voxel structure is subdivided into tetrahedron volumetric structure. The two material elements e_1=ABCD and e_2= EBCD are sharing nodal vertices B, C and D, which are regarded as internal shared vertices. Nodal vertices A and E are boundary vertices.*

3. Matrix clustering

Equation (1) defines a coupled system of 3*n* ordinary differential equations for *n* number of nodal vectors contained in the vector **u**. There are a number of different ways to decouple the equation (1). Typically, equation (1) is solved with the following linear system of equation:

$$A\Delta\dot{u} = f \qquad (3)$$

Where *h* being the step length of each simulation step and *f* being the total force acting on the dynamic system:

$$A=(M+hD+h^2K)$$

$$f = h[-D\dot{u}-K(u+h\dot{u})+f_{ext}+f_{\psi}]$$

The displacement vector **u** at the time step $t+h$ is obtained by $u(t+h)=u(t)+h(\dot{u}+\Delta\dot{u})$. Equation (3) can be solved by iterative algorithms, such as the conjugate gradient method [HE01], since matrices **M, D,** and **K** are sparse.

Iterative methods could be inefficient for objects consisting of large number of finite elements, preventing interactive animations. Another way of solving the linear system of equation (3) is to invert matrix **A,** resulting the formulation $\Delta\dot{u} = A^{-1}f$, where the inverse matrix $\mathbf{A^{-1}}$ is a large dense matrix**.** In order to achieve fast matrix and vector multiplications, Bro-Nielsen and Cotin [BC96] applied a method to condense the inverse matrix $\mathbf{A^{-1}}$ by only taking into account of the forces acting on the boundary nodes and assuming that there were no forces acting on internal nodes, therefore only computing forces for boundary vertices. Lang and Seidel [LS03] used Green's functions for matrix of boundary volume partitions to approximate the force-displacement matrix **A**. There also have been considerable efforts in directly approximating an inverse matrix with the replacement of a sparse matrix in order to obtain a better pre-conditioner for fast convergence [BCT00] [BT01]. For deformation animations, Huang et al. [HLBGS05] applied PCA to the clustered $\mathbf{A^{-1}}$ to achieve a speed up matrix-vector multiplications with sub-space linear combinations of eigen vectors.

There are two shortcomings with above matrix condensation approaches. First, in order to achieve fast computations, inverting the system matrix **A** would results in a large dense matrix $\mathbf{A^{-1}}$, which has to be solved by approximations. Although results of modal analysis are excellent for some computer animation purposes, many applications demand higher accuracy of modelling as well as interactive frame rate, for example interactive biomechanical stress simulation and medical simulations with haptics. Second, in previous matrix compression methods, large sparse matrices **D** and **K** were kept in the right-hand side of equation 3, which need to be evaluated at each time step, yielding a slow simulation. Moreover a constant step length was required, ensuring the inverse $\mathbf{A^{-1}}$ being a constant matrix. Our matrix compression scheme removes this restriction and in addition it compresses all **M**, **D**, and **K** matrices.

We formulate the equation (2) into the Newton equation, such that the acceleration of a vertex point is computed as:

$$\ddot{u} = M^{-1}[-D\dot{u}-Ku+f_{ext}+f_{\psi}] \qquad (4)$$

Where damping matrix D is represented as a local Rayleigh damping model to represent damping forces as the form: $D=[\xi M+\zeta K]$, with ξ and ζ being two positive real values to control the low and high frequency components of the deformation. The internal elastic-damping force vector $f_{elastic} = -D\dot{u} - Ku$ and equation (4) becomes:

$$\ddot{u} = M^{-1}[f_{elastic}+f_{ext}+f_{\psi}] \qquad (5)$$

Damping matrix **D** and stiffness matrix **K** are sparse constant square matrices for linear finite element implementations, which can be pre-computed at the start of simulation. The internal elastic force $f_{elastic}$ at each node is the sum of forces excreted by tetrahedrons that are connecting to the node. We denote each of the 3×3 block elements in matrices **D** and **K** as $\mathbf{D_{ij}} \in R^{3\times3}$ and $\mathbf{K_{ij}} \in R^{3\times3}$.Thus rudimentary matrices $\mathbf{D_{ij}}$ and $\mathbf{K_{ij}}$ are responsible for the displacement acceleration at i^{th} vertex connected to j^{th} , An entry of the 3×3 blocks are $\mathbf{K_{vw}}$ and $\mathbf{D_{vw}}$, respectively, where $3i-2\leq v\leq 3i$ and $3j-2\leq w\leq 3j$.

We apply a matrix clustering procedure to reduce the number of multiplications in vector and matrix computation, making use of the built in sparsity and spatial coherence of the discrete finite element structures. This approach subdivides matrices **D** and **K** into sub-matrices according to the connectivity of each node in the geometric mesh. As shown in figure 2, the two tetrahedron elements: e_1=ABCD and e_2=EBCD have three vertices B, C and D shared by both elements e_1 and e_2, thus vertices B, C, and D are regarded as internal shared vertices, whereas vertices A and E are called boundary vertices since they are only connected to their own tetrahedron elements.

Factorizing sub-matrices of **D** and **K** yields four new sub-matrices, two for the shared nodes $\mathbf{D_s}$ and $\mathbf{K_s}$, two for the boundary nodes $\mathbf{D_b}$, and $\mathbf{K_b}$. Clustered matrices $\mathbf{D_s}$,

$\mathbf{K_s} \in R^{3m\times 3m}$ (m<<3n) are diagonal matrices for shared vertices with *m* number of shared vertices, containing pre-computed matrix clusters. Similarly, $\mathbf{D_b}$, $\mathbf{K_b} \in R^{3l\times 3l}$, (l<<3n) contains pre-computed matrix clusters for *l* number of boundary vertices.

The elastic force computed at vertex i becomes the sum of forces found on the shared nodes and the boundary nodes, i.e. $f_i = f_s + f_b$, where

$$f_s = D_s \dot{U}_s^T + K_s U_s^T,$$
$$f_b = D_b \dot{U}_b^T + K_b U_b^T \quad (6)$$

In equation (6), velocity vector clusters for *m* number of internal shared vertices are evaluated as $\dot{U}_s = (\dot{U}_1^T \cdots \dot{U}_m^T)$ and displacement vector clusters for shared vertices are calculated as $U_s = (U_1^T \cdots U_m^T)$. Similarly for *l* number of boundary vertices, the description for velocity vector clusters is $\dot{U}_b = (\dot{U}_1^T \cdots \dot{U}_l^T)$ and $U_b = (U_1^T \cdots U_l^T)$ for displacement vector clusters.

Clustered pre-computations lead to block matrices with reduced dimensionality of displacement bases that can be evaluated much more efficiently than the original matrices with the unreduced dimensionality of equation 5. The sparstiy of the original system matrix is utilized while condensing the matrices and all nodes of the objects are computed during run-time interactions. There is no approximations to the original system matrix, thus it is essentially a lossless compression of the system matrix. This simple clustering method also produces accurate deformations for linear finite element computations.

As shown in figure 3, test examples compare the results produced by our method to the results of generated by the standard linear finite element method , name as the ground truth method in figure 3 to be regarded as a benchmark algorithm for linear finite element computation for our evaluation purpose. The results of clustering method are also compared with that produced by the modal analysis method.

4. Constrained deformations

A skeleton-based constraint system is developed that is aimed at facilitating animations of deformable objects with the use of animated datasets. In addition, the constraint system is also effective for simulating long soft objects such as limbs and legs of the character shown in figure 1. Preserving the volume of long deformable objects is problematic with linear strain finite element models. Particularly, where there is no rotation existing but only stretching deformations presented in dynamic motions, the volume of deformable objects simulated by linear finite element models are enlarged, since linear strain tensor with constant **K** and **D** matrices are not account for large deformations on material rotations and stretches. The skeleton structure utilizes hierarchical representations of character animations to allow animations with motion capture data or key-frame animation datasets to be used for deformation simulations. Therefore, rigid body dynamics can be used for non-rigid body deformations by combining skeleton-based constraint manipulations with finite element structures to create natural looking global deformations.

4.1 Hierarchical skeleton-based deformation

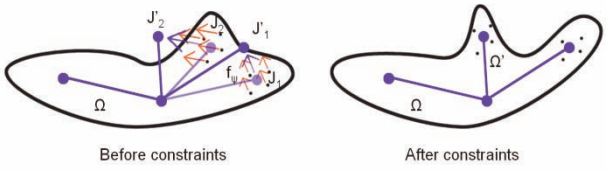

Figure 3. *Hierarchical skeleton-based deformation structure: image to the left shows joints 1 and 2 from positions J_1 and J_2 moved toJ_1' and J_2'. Constrained forces are applied along the arrowed directions. The deformation after constraints is shown on the image to the right.*

The hierarchy skeleton structure **Ψ** is attached to the deformation object **Ω** as shown in figure 1. The bone joints in **Ψ** are represented as a list of vectors that are constraints associated with a set of tetrahedron vertices Ω'. The dataset Ω', defining different levels of constrained vertex sets $\mathbf{\Omega_t'}$. The hierarchical constrained vertices controlled by **Ψ** and $\mathbf{\Omega_t'}$ are the union of the nested sequence $\Omega_1' \subset . \ \Omega_2' \subset . \ \Omega_3' \subset \ldots\ldots$

From the traditional kinematics theory, the spatial velocity vector of a rigid body $\Phi = (\omega^T \upsilon^T)^T$ is a 6×1 vector, where $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ are linear velocity and angular velocity, respectively. The velocity $\dot{c}_i$ at a vertex *i* is given by:

$$\dot{c}_i = [\omega]c_i + \upsilon = (-[c_i]I)\Phi \quad (7)$$

where [ω] is the standard skew-symmetric matrix of the cross product $\boldsymbol{\omega}\times$, and $I \in R^{3\times 3}$ is the 3×3 identity matrix. Given a constant time step size *h*, the acceleration of a constrained vertex i at time step *t* is given by:

$$\ddot{c}_i \approx \frac{1}{h}(\dot{c}_{i,t+1}, -\dot{c}_{i,t}) = \frac{1}{h}(-[c_i]I)(\Phi_t - \Phi_{t-1}) \quad (8)$$

Each constraint joint of **Ψ** applies constrain force $\mathbf{f_\psi}$ to the corresponding material vertices $\mathbf{\Omega_t'}$, based on **F=ma** formula.

$$f_{i,t}^{\psi} = \frac{1}{h} m_i^{\Omega_t'}(-[c^{\Omega_t'}{}_i]I)(\Phi^{\Omega t'}{}_{t+1} - \Phi^{\Omega t'}{}_{t-1}) \quad (9)$$

In the recent work [YBS07], a skeletal mesh based on a two-side Voronoi-based approximation was used for free-form deformations within the discrete differential coordinates. Skeleton-based skin deformation was presented in [JP02] using modal analysis approach with

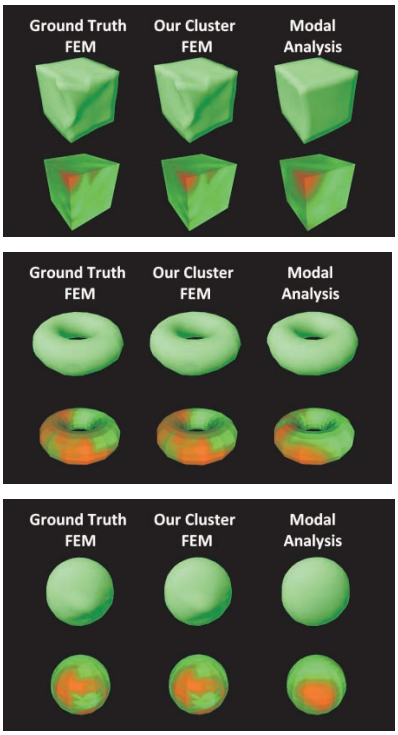

Figure 4: *Comparisons of displacement deformations and stress distributions at the same loading conditions for ground truth finite element (GFEM) method, our clustering method and modal analysis with 30 eigen vectors.*

hardware computations. In [CGCDP02], a skeleton-based framework for finite element based deformation was presented, which allowed animating arbitrary shapes with skeletal controls to generate realistic dynamic deformations. Our skeleton-based constraint system also solves constrained deformation, allowing both point-point constraints as well as line constraints along the joints. The constraint system works within an iterative solver based on fast rigid body constraint solver.

4.2 Rotational constraints with stiffness warping

Because of the linear elastic model used in our computation, the rotational transformation, the rotational constraints in equation (7) are a global rotation of the rigid body transformation of the finite element structure. Műller [5] introduced the stiffness warping approach by separating global rotation from the transformation and warping the rotational component in the element's local frame. This approach results in efficient and plausible results. Therefore, the elastic force computed at material vertex i is given by (see [5] for details):

$$f_i = \omega \sum_{j=1}^{n} k_{ij}(\omega^{-1} x_j - x_{0j}) \qquad (10)$$

where ω is calculated in the local frame of each element, x_j and x_{0j} are deformed and unreformed positions of vertex j . The rotational component ω is computed based on the rotation tensor field for the corresponding vertices of an element with respect to the unreformed and deformed states.

5. Results

All testing examples are obtained on a PC (an Intel Pentium 1.60 GHz processor with 512M RAM).

Table: Comparisons of computational costs of three methods.

object	Sphere	Cube	Torus
Number of vertices	376	453	609
Number of tetrahedron	1091	1198	1823
Standard FEM	1.60 ms/step	1.89 ms/step	2.15 ms/step
Our Cluster FEM	0.73 ms/step	0.92 ms/step	0.97 ms /step
Modal Analysis	0.83 ms/step	1.09 ms/step	1.12 ms/step

Table 1 shows the comparison of time spent on computing one step simulation using the clustering method with the standard finite element method (ground truth) and with the modal analysis method using 30 eigen vectors. On average our clustering method saves up to 50% of runtime computation compared with the standard finite element method and is comparable to modal analysis, even slightly faster, but more accurate in terms of deformation details compared with the results produced by bench mark standard finite element method.

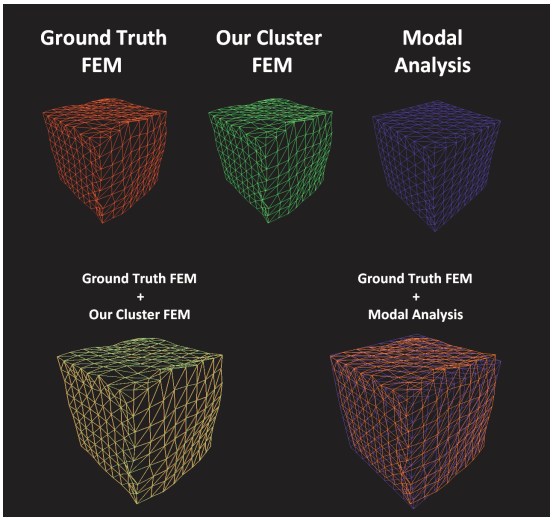

Figure 5. *Comparisons of difference in Displacement deformations under the loading conditions for standard finite element method, our clustering and modal analysis methods (with 30 eigen vectors).*

Figure 4 shows our matrix clustering method produces similar deformations as the standard finite element method with more details compared with modal analysis, whilst deformations produced by modal analysis are more centered around the origin of the initial elastic deformations. Thus fine details in deformation, are generally missing in the approximation-based approaches. Similarly in stress analysis (red colour indicates higher stress level than green colour), our method produces more distributed stress distributions as with standard finite

element method, while stresses are more centralized with modal analysis method.

Figure 5 shows comparisons of difference in displacement deformations under the loading conditions for standard (ground truth) finite element method, our clustering method and modal analysis with 30 eigen vectors. As can be seen in figure 4, the displacement produced by our clustering finite element method match that of standard finite element method quite closely (vertices in yellow colour have the same displacement values in the two methods), whilst deformations produced by modal analysis have been quite different to that of standard finite element method (vertices are displaced in red and blue colours, respectively), this is due to the fact that deformations of modal analysis are more centered around the origin of the initial elastic deformations.

Our system also handles large degrees of freedom (DOFs). The deformable jelly man model in figure 1 contains 4822 elements. Its skeleton bone structure consists of 48 DOFs. It takes the system 2 seconds to compute 1000 simulation steps and an interactive frame rate of 229Hz is obtained.

6. Conclusion

In this paper, we presented a new approach to fast compute finite element models for real-time deformation modeling and animations by reducing the dimensionality of the matrix through clustering procedures without approximation. Our study compares deformations and stress displacements computed by our non-approximation based approach with that of modal analysis, which shows more versatile and accurate deformations can be produced with comparable computational cost. For large and complex objects, our methods may be more numerically sensitive than modal analysis. However, all simulations on our testing examples are stable. We envisage that the proposed method would be particular useful for applications that prefer more accurate stress computations than the visual effects. Stiffness warp aids the efficiency of computation to accommodate the rotational components of the deformation.

Our system supports constrained deformable objects attached to rigid body dynamics with conventional skeleton based animation specifications. Hence it is capable of incorporating complex motions into deformation simulations, allowing flexibility in design and manipulations of deformable objects.

For future work, we would like to replace the linear strain model and to test the pre-computed coefficients of cubic polynomials of a nonlinear force model in the reduced model coordinate space [9] accurate interactive deformations, for example. Another interesting area to be further exploited is to test the model reduction method with statistical model with our non-approximation based approach.

References:

[BC96] Bro-Nielsen, M and Cotin, S. 1996. Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. Computer Graphics Forum. 15, 3, pp. 57-66, 1996.

[BCT00] Benzi, M, Cullum, J. M., and Tuma, M. 2000. Robust Approximate Inverse Perconditioning for Conjugate Gradient Method. SIAM Journal on Scientific Computing 22(4), pp 1318-1332

[BJ05] Barbič, J. and James, D. 2005. Real-time Subspace Integration for St. Venant-Kirchhoff Deformable Models. ACM Trans, on Computer Graphics (In Proceedings of ACM SIGGRAPH 2005), pp.982-990, 2005.

[BT01] Bridson, R. and Tang, W. P. 2001. Mutilresolution Approximate Inverse Perconditioner. SIAM Journal on Scientific Computing 23(2), pp 463-479

[BW92] Baraff, D., and Witkin, A. 1992. Dynamic Simulation of Non-penetrating Flexible Bodies. Computer Graphics (In Proceedings of ACM SIGGRAPH 1992) 26, 2, pp. 303-308, 1992.

[BW98] Baraff, D. and Witkin, A. 1998. Large Steps in Cloth Simulation. ACM Trans. Graphics (In Proceedings of ACM SIGGRAPHICS 1998) pp. 43-54, 1998

[CGCDP02] Capell, S., Green, S., Curless, B., Duchamp, T., and Popovic, Z. 2002. A Multiresolution Framework for Dynamic Deformations. In Proceedings of the 2002 ACM SIGGRAPHICS/ EUROGRAPHICS Symposium on Computer Animation, ACM Press. pp. 41-47, 2002.

[CK05] Choi, M. G., and Ko. H-S. 2005. Modal Warping: Real-time Simulation of Large Rotational Deformation and Manipulation. IEEE Tran. On Visualization and Computer Graphics, 11, 1, pp. 91-101, January/February 2005.

[DDCB01] Debunne, G., Desbrun, M., Cani, M.-P., And Barr, A. H., 2001. Dynamic Real-time Deformation using Space and Time Adaptive Sampling. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press. pp.31-36, 2001.

[FPT97] Faloutsos, P., van de Panne, M., and Terzopoulos, D. 1997. Dynamic Free-form Deformations for Animation Synthesis. IEEE Trans. On Visualization and Computer Graphics. 3,3 pp. 201-214, 1997.

[GKS02] Grinspun, E., Krysl, P., and Schröder, P. 2002. CHARMS: A Simple Framework for Adaptive Simulation. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press, pp. 281-290, 2002.

[HE01] Hauth, M. and Etzmuss, O. 2001. A High Performance Solver for the Animation of Deformable Objects Using Advanced Numerical Methods. In *Proceedings of Eurographics* 2001 pp. 137-151

[HLBGS05] Huang, J. Liu, X., Bao, H., Guo, B., and Shum, H.-Y. 2005. Clustering Method for Fast Deformation with Constraints. In Proceedings of the

2005 ACM Symposium on Solid and Physical Modeling, pp. 221-226, 2005.

[JP99] James, D. L., and Pai, D. K., 1999. ARTDEFO: Accurate Real Time Deformable Objects. ACM Trans. Graphics (In Proceedings of ACM SIGGRAPH 1999) pp. 65-72, 1999.

[JP02] James, D. L., and Pai, D. K. 2002. DyRT: Dynamic Response Textures for Real-time Deformation Simulation with Graphics Hardware. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press. pp. 582-585, 2002.

[LS03] Lang, J. D. K. P. and Seidel, H-P. 2003. Real-time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation. Proceedings of Graphics Interface.

[MG04] Műller, M. and Gross, M. 2004. Interactive Virtual Materials. In Proceedings of the 2004 Conference on Graphics Interface, Canadian Human-Computer Communications Society, pp. 239-246, 2004.

[O'BH99] O'Brien, J. F., and Hodgins, J. K. 1999. Graphical Modeling and Animation of Brittle Fracture. Computer Graphics (in Proceedings of ACM SIGGRAPH 1999) pp. 137-146, 1999.

[PW89] Pentland, A. and Williams, J. 1989. Good Vibrations: Model Dynamics for Graphics and Animation. Computer Graphics (in Proceedings of ACM SIGGRAPH 1989) pp. 207-214, 1989.

[SHHS03] Sloan, P.-P., Hall, J. Hart, J. and Sayder, J. 2003. Clustered Principal Components for Precomputed Radiance Transfer. ACM Trans. on Graphics. 22, 3, pp.382-391.

[TF88] Terzopoulos, D and Fleischer, K. 1988. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. Computer Graphics (Proceedings of ACM SIGGRAPH'88) pp. 269-278, 1988.

[YBS07] Yoshizawa, S., Belyaev, A., and Seidel, H.-P., 2007. Skeleton-based Variational Mesh Deformations. Computer Graphics Forum (In Proceedings of EUROGRAPHICS 2007). 26,3,pp. 255-264