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

Jin Huang · Xiaohan Shi · Xinguo Liu* · Kun Zhou · Baining Guo · Hujun Bao*

Geometrically-Based Potential Energy for Simulating Deformable Objects

Abstract This paper presents a fast and stable technique for simulating deformable objects. Unlike in previous physically-based methods, our potential energy of deformation is purely geometrically-based. It is defined as the L^2 norm of the change of the differential coordinates. A key feature of this energy formulation is that the corresponding stiffness matrix is approximately constant, which enables fast and stable implicit integration and large deformations. Our algorithm can simulate various effects including solid, thin shell and plasticity. We also adopt two schemes to accelerate the simulation process: dimensionality reduction in frequency domain and adaptive rotation computation in spatial domain.

Keywords Laplacian · Simulation · Deformation

1 Introduction

Deformable objects simulation is a useful tool for many computer graphics applications, e.g., video games and virtual surgery. Most of previous simulation methods are based on physical laws. Although the physically-based models can faithfully capture all deformation effects, they are generally too complex to be used in interactive applications. There have been many efforts to simplify the physical models to get a tradeoff between the performance and accuracy. Examples include the mass-spring system, the linear elasticity model, and some dimensionality reduction methods.

Inspired by the recent mesh deformation methods that preserve differential surface properties [1,28,31], we present a novel geometrically-based formulation for the potential energy of deformation. The potential energy is defined as the summed squares of the differences between the current differential coordinates and the original ones under the local frame. Our simulation results are comparable to those from physically-based methods as shown in Fig. 1.

A key feature of our geometrically-based energy is that the corresponding stiffness matrix is approximately constant, which has many advantages. First of all, we can stably simulate large deformation efficiently by using implicit-Euler integration. Previous linear elasticity model [18,14] can also produce a constant stiffness matrix. Unfortunately, it is limited to small deformations. When the object undergoes a large deformation, some noticeable simulation error will occur. Another advantage is that the geometrical energy does not rely on a specific volumetric tessellation of the simulated object, and can be formulated on surface meshes as shown in Section 3.2. This is very useful for many objects with only surface representations.

To further improve the simulation performance, we adopt two acceleration methods. The first is a subspace integration method using a set of time invariant deformation basis. The second is an spatially adaptive scheme for tracking the local frames. By taking advantage of the spatial coherence among neighbor nodes, the adaptive method dramatically reduces the cost of tracking the local frames.

In the rest of the paper, we will first briefly review some related works in Section 2, then present our geometrical energy for simulation in Section 3. Section 4 and Section 5 will describe the two acceleration methods. Finally, we conclude this paper in Section 6.

This project is partially supported by Natural Science Foundation of China under Grant No. 60021201, The National Basic Research Program of China (973 Program) under Grant No. 2002CB312102, and Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China under Grant No. 705027.

*Corresponding Authors: Xinguo Liu, Hujun Bao

Jin Huang, Xiaohan Shi, Xinguo Liu, Hujun Bao
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China
Tel.: +86-571-8820 6881 Ext. 528
Fax: +86-571-8820 6880
E-mail: hj,shixiaohan,xgliu,bao@cad.zju.edu.cn

Kun Zhou, Baining Guo
Microsoft Research Asia
Tel.:+86-10-62617711
Fax:+86-10-88097306
E-mail: kunzhou,bainguo@microsoft.com

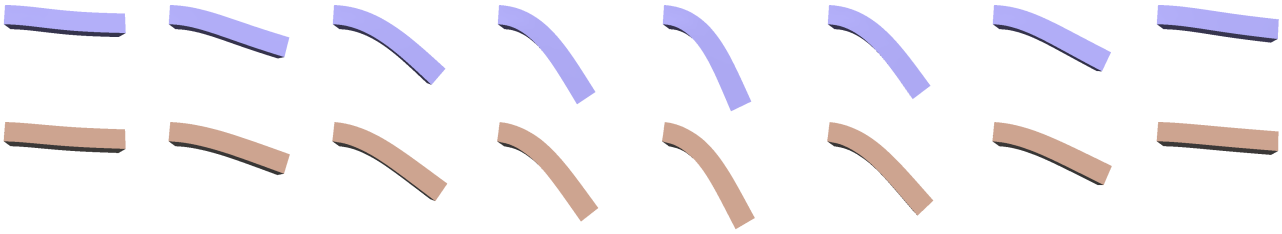


Fig. 1 Simulation comparison of a bending bar under gravity. The left end is fixed. The top row shows the results simulated by our method with geometrically based energy, while the bottom row shows the results simulated by the physically based method in [23] (using the spring element instead of the tetrahedron element).

2 Related Work

Many physically-based methods for simulating deformable objects in computer graphics have been proposed. A comprehensive survey can be found in [12,25]. Probably, the simplest deformable model is the mass-spring systems, which have been successfully applied to simulate many kinds of soft things, such as creatures and clothes [2,7]. Mass-spring system models an object as a set of mass points connected by some massless springs.

Despite the success of mass-spring systems in cloth simulation, there is a trend toward more versatile and complex deformable models based on continuum mechanics, in order to achieve more realistic simulation results [20,4,18,9,26].

Stable simulation algorithm usually needs an implicit integration scheme, which requires solving a linear system at each simulation time step. So it is computational expensive in general. In order to improve the efficiency and stability, some previous work used the linear elasticity model [6,5]. The linear elasticity model approximates the internal force as a linear function of the displacement, so the corresponding stiffness matrix is constant. Thus, the modal analysis method can be further applied for further acceleration [27, 14]. The linear elasticity model is simple and fast, but it also leads to distortions for large rotational deformation, which is a significant limitation for computer graphics applications. Embedding the object into a floating reference frame can greatly suppress the error [29]. But, if the simulated object has several parts undergoing very different rotational deformation, then distortions are inevitable. Capell et al. [6,5] address large deformations by subdividing the object into several zones and then blending the shared nodes. Huang et al [15] address large deformation via more advanced domain decomposition method.

Recently, several algorithms are proposed to handle large deformation in reduced dimension. The modal warping method [8] by Choi et al. combines the advantages of stiffness warping [22] and modal analysis. There are some ghosting forces with the node based warped stiffness, which make the objects randomly drift without external constraints. Although this problem can be correct by [23], modal warping cannot utilize this remediation. The method proposed in [3] is closely related to modal analysis, which can simulate StVK material efficiently in subspace only. Since the computational complexity is $O(r^4)$ (r is the mode number), it's costly to augment the deformation space by adding more basis.

To efficiently solve the simulation at each time step, some adaptive methods are proposed. Capell et al. proposed a multiresolution framework for dynamic simulation using volumetric subdivision [6]. By adaptively refining the basis functions, Grinspun et al. proposed another simple framework for adaptive simulation [13], called CHARMS. Wu et al. [30] and Debunne et al. [10] developed other kinds of adaptive methods using progressive meshes and LOD tetrahedral meshes. Debunne et al. also took adaptive time steps during simulation [10].

Our work is also related to the recent gradient domain mesh editing methods [31,19,32]. These mesh editing methods preserve surface details by minimizing via a quadratic energy of the differential coordinates. In this paper, we exploit and extend the differential coordinates to define a novel potential energy of the deformation for simulating deformable objects. Recently, Muller et al. [24] proposed a non-physical motivated method based on shape matching. They need to decompose the object into several overlapped clusters and then blend the clusters together for large deformation, while our method does not need any decomposition. Similar deformation energy is defined in [16]. But their iteration scheme cannot generate a rhythm realistic deformation sequence, because they only solve a energy optimization problem instead of a physical meaning partial differential equation - the motion equation. There is the same problem in [17].

3 Simulation with Geometrical Energy

In the following, a deformable object is represented as a triple $(\mathcal{V}, \mathcal{G}, \mathbf{x})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes, $\mathcal{G} = \{(i,j) | i,j \in \mathcal{V}, \text{ and } i \text{ and } j \text{ are connected } \}$ is a graph representing the node connectivity, and $\mathbf{x}$ is a vector of points in R^3 representing the node positions. For each node i, we denote its immediate neighborhood as $\mathcal{N}_i = \{ j \mid (i,j) \in \mathcal{G} \}$. And we assume that all mass is concentrate on the nodes, and let m_i be the mass of node i. For convenience, let $\mathbf{r}$ be the position vector of the rest state, and $\mathbf{r}_i$ and $\mathbf{x}_i$ be respectively the rest and deformed position of node i.

3.1 Simulation Framework

We first briefly review the Euler-Lagrange motion equation and implicit integration scheme. The elastic animation is governed by the following Euler-Lagrange equations:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \frac{\partial V(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{f}_{ext}, \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{D}$ is the damping matrix (Rayleigh damping model in used in this paper), V is the potential energy required to deform the object into the current configuration, and $\mathbf{f}_{ext}$ denotes the external forces acting on the object. And the differential of the potential energy $\frac{\partial}{\partial \mathbf{x}} V(\mathbf{x})$ is actually the internal elastic force, and the Jacobian $\mathbf{K}$ of the internal force is usually called the *stiffness matrix*. Therefore, $\mathbf{K} = \frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}} V(\mathbf{x})$.

To ensure stability for reasonably large time steps, the implicit Euler integration scheme is widely used to solve the above motion equation in the context of computer graphics. Let t be the current simulation time, and h be the time step. Then, we have

$$\left(\mathbf{M} + h\mathbf{D} + h^2\mathbf{K}\right) \Delta\dot{\mathbf{x}} = h\left(\mathbf{f}_{ext} - \mathbf{D}\dot{\mathbf{x}} - h\mathbf{K}\dot{\mathbf{x}} - \frac{\partial}{\partial \mathbf{x}} V(\mathbf{x})\right). \tag{2}$$

After solving the above linear system for $\Delta\dot{\mathbf{x}}$, the state at the next time step $t + h$ can be updated by:

$$\dot{\mathbf{x}}(t+h) \leftarrow \dot{\mathbf{x}}(t) + \Delta\dot{\mathbf{x}}$$
$$\mathbf{x}(t+h) \leftarrow \mathbf{x}(t) + h\dot{\mathbf{x}}(t+h).$$

3.2 Geometrical Energy

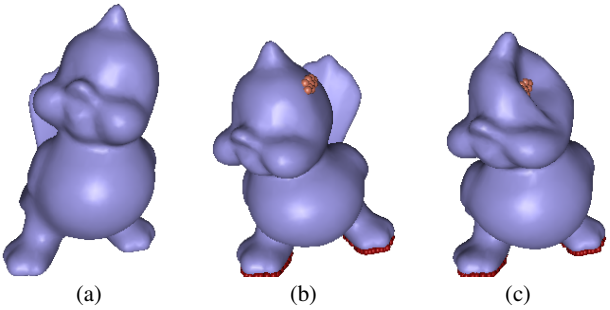
In traditional physically-based simulation methods, the potential energy $V(\mathbf{x})$ is integrated over the whole continuum materials based on the strain and stress tensors [5]. However, such physically-based potential energy gives a time variant stiffness matrix $\mathbf{K}$ when the object is deformed, which requires expensive computation for updating $\mathbf{K}$ and solving the linear equations in Eq. (2). In the following, we will define a novel potential energy of deformation that produces a constant stiffness matrix.

Recall that Laplacian differential coordinates have been successfully exploited in many mesh editing systems for preserving surface details [31,32]. Inspired by these works, we propose to define the following potential energy:

$$V(\mathbf{x}) = \frac{\lambda}{2} \sum_{i \in \mathcal{V}} \left\| \mathbf{L}_i \mathbf{x} - \mathbf{R}_i(\mathbf{x}) \mathbf{d}_i \right\|^2, \tag{3}$$

where $\mathbf{L}_i$ is the differential operator at node i, $\mathbf{d}_i$ is the differential coordinate of node i computed at the rest state, and $\mathbf{R}_i(\mathbf{x})$ is a 3×3 matrix representing the rotation of the local frame of node i from the rest state to the current configuration. The scalar λ serves as the Young's modulus.

Differential Operators Many differential operators can be applied in our geometrical energy. For general solid objects, we construct a volumetric graph and adopt the volumetric graph Laplacian operators [32]. For thin shell objects represented in triangular meshes, we adopt the cotangent form surface Laplacian operators [11]. Like most of these differential coordinates based mesh deformation algorithms, we use the initial weights calculated on the undeformed mesh throughout the whole simulation. This choice doesn't prohibit achieving large deformation as demonstrated in [32].



(a) (b) (c)

Fig. 2 Simulation results on a surface mesh. (a) the rest shape; (b) deformation effects achieved with cotangent form Laplacian based geometrical energy; (c) a thin shell deformation effect achieved with edge based geometrical energy.

We found in experiments that the potential energy based on the surface Laplacian tends to preserve the curvature of thin shells. This can be explained by the property that the Laplacian approximates the mean curvature normal, and the potential energy has a goal of maintaining the length of the Laplacian. Therefore, we can simulate solid shape behavior with the surface model only. An example is shown in Fig. 2(b). This is very useful, since the geometry is usually represented by a surface mesh and a specific internal volumetric model is usually not available.

In order to simulate thin shell properties on surface triangle mesh, we propose the edge based potential energy as follows:

$$V(\mathbf{x}) = \frac{\lambda}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}(i)} \left\| \mathbf{L}_{ij} \mathbf{x} - \mathbf{R}_i(\mathbf{x}) \mathbf{d}_{ij} \right\|^2, \tag{4}$$

where $\mathbf{L}_{ij}$ denotes the differential operator at edge (i, j), and $\mathbf{d}_{ij}$ is the corresponding differential coordinate of edge (i, j) computed at the rest state. In our current implementation, $\mathbf{L}_{ij}$ is simply defined as:

$$\mathbf{L}_{ij}\mathbf{x} = \omega_{ij}(\mathbf{x}_j - \mathbf{x}_x),$$

where ω_{ij} is the edge weight. ω_{ij} can be set as 1 for objects with evenly sampled nodes, or can take the same values as in the cotangent form Laplacian for general cases. Although carefully choosing the weight for more accuracy is possible, we found that such a simple uniform weighting can generate good results even for an irregular sampled object.

An example of thin shell deformation effect is shown in Fig. 2(c). We can also integrate the edge based potential energy into the Laplacian based potential energy by a linear combination, to control the deformation effects.

Rotation Estimation To evaluate the potential energy, we need to estimate the rotation for each node $i \in \mathcal{V}$. We take a mass weighted polar decomposition method similar to that in [24]. Let $\mathbf{r}_i$ be the rest position for each node $i \in \mathcal{V}$, and denote

$$\mathbf{r}_{ij} = m_j(\mathbf{r}_j - \mathbf{r}_i),$$
$$\mathbf{x}_{ij} = m_j(\mathbf{x}_j - \mathbf{x}_i).$$

Then the best linear transformation $\mathbf{A}_i$ at node i that minimizes $\sum_{j\in\mathcal{N}(i)} \|\mathbf{x}_{ij} - \mathbf{A}_i\mathbf{r}_{ij|}\|^2$ is $\mathbf{A}_i = \mathbf{A}_i^{\mathbf{xr}}\mathbf{A}_i^{\mathbf{rr}}$, where

$$\mathbf{A}_i^{\mathbf{xr}} = \Big(\sum_{j\in\mathcal{N}(i)} \mathbf{x}_{ij}\mathbf{r}_{ij}^t\Big), \text{ and } \mathbf{A}_i^{\mathbf{rr}} = \Big(\sum_{j\in\mathcal{N}(i)} \mathbf{r}_{ij}\mathbf{r}_{ij}^t\Big)^{-1}.$$

Since $\mathbf{A}_i^{\mathbf{rr}}$ is symmetric, it doesn't contain any rotation information. Therefore, the rotation $\mathbf{R}_i(\mathbf{x})$ can be found by applying a polar decomposition on $\mathbf{A}_i^{\mathbf{xr}}$. Because we use the implicit Euler method to solve the motion equation, the rotation is estimated on predicted next state $\mathbf{x} + h\dot{\mathbf{x}}$ instead of $\mathbf{x}$ in our implementation.

We can easily show that $V(\mathbf{x})$ defined in Eq. (3) is invariant under rigid transformation, so it satisfies the basic requirements for a valid potential energy function. First, since the differential operator is translation invariant, $V(\mathbf{x})$ is also translation invariant. Second, when the object undergoes an additional global rotation, say $\mathbf{T}$, and therefore $\mathbf{R}_i(\mathbf{T}\mathbf{x}) = \mathbf{T}\mathbf{R}_i(\mathbf{x})$ and $\mathbf{L}_i\mathbf{T}\mathbf{x} - \mathbf{R}_i(\mathbf{T}\mathbf{x})\mathbf{d}_i = \mathbf{T}(\mathbf{L}_i\mathbf{x} - \mathbf{R}_i\mathbf{d}_i)$, then $V(\mathbf{T}\mathbf{x}) = V(\mathbf{x})$. So, $V(\mathbf{x})$ is rotation invariant.

Stiffness Matrix Now we develop the stiffness matrix for the potential energy using Laplacian operators (The stiffness matrix of using the edge based differential operator can be similarly deduced). Let $\mathbf{L}$ be the matrix of the Laplacian operator, $\mathbf{R}(\mathbf{x})$ be a diagonal matrix consisting of all $\mathbf{R}_i(\mathbf{x})$, and $\mathbf{d}$ be a vector consisting of all $\mathbf{d}_i$. Then we have $V(\mathbf{x}) = \frac{1}{2}\lambda\|\mathbf{L}\mathbf{x} - \mathbf{R}(\mathbf{x})\mathbf{d}\|^2$, and

$$\begin{aligned}\frac{\partial}{\partial\mathbf{x}}V(\mathbf{x}) &= \lambda\Big(\mathbf{L} - \frac{\partial\mathbf{R}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{d}\Big)^t\Big(\mathbf{L}\mathbf{x} - \mathbf{R}(\mathbf{x})\mathbf{d}\Big)\\ &\approx \lambda\mathbf{L}^t\Big(\mathbf{L}\mathbf{x} - \mathbf{R}(\mathbf{x})\mathbf{d}\Big),\end{aligned}$$

$$\frac{\partial^2}{\partial\mathbf{x}^2}V(\mathbf{x}) \approx \lambda\mathbf{L}^t\Big(\mathbf{L} - \frac{\partial\mathbf{R}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{d}\Big) \approx \lambda\mathbf{L}^t\mathbf{L}. \quad (5)$$

Therefore, we have $\mathbf{K} \approx \lambda\mathbf{L}^t\mathbf{L}$. In the above, the derivatives $\frac{\partial}{\partial\mathbf{x}}\mathbf{R}(\mathbf{x})$ are dropped two times. Though this introduces some errors, our experiments show that such an approximation does not cause great problems in many cases, and it greatly simplifies the stiffness matrix down to a constant matrix. As shown in the following sections, the simulation stability and efficiency are greatly improved with such an approximation.

Such a energy seems like node based stiffness warping in [22]. Considering that the each row of the Laplacian operator L sums to zero, the internal elastic force sums to zero too for any deformed state, and the momentum is preserved. So, it's an advantage of our algorithm that we will not encounter the ghost force problem. And such a linearized stiffness matrix is constant during the simulation, which is more profitable for preprocessing to speedup the simulation than the stiffness warping technique.

Note that the stiffness matrix $\mathbf{K} = \lambda\mathbf{L}^t\mathbf{L}$ is $n\times n$, instead of $3n \times 3n$. And the x/y/z coordinate components in Eq. (2) are solved independently.

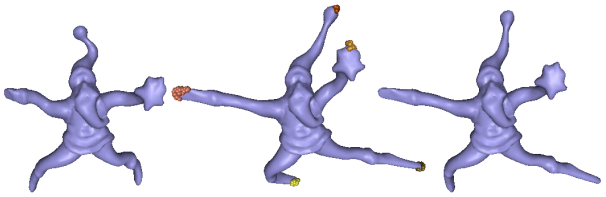


Fig. 3 Plasticity effect: the left image shows the undeformed Santa model. We deform the Santa by the colored nodes (the center image), then free the nodes. Because of the plasticity, the model cannot restore to original shape (the right most image).

3.3 Simulating Plasticity

Now we show how to simulate plasticity behaviors using the edge based potential energy. One of the characteristics of a plastic material is that when the deformation is too large, it can not restore to its rest shape when all external forces are released.

At each simulation step, the difference between the differentials $\mathbf{L}_{ij}\mathbf{x} - \mathbf{R}_i\mathbf{d}_{ij}$ indicates the degree of deformation. Therefore, we can define the deformation ratio as

$$\rho_{ij} = \frac{\|\mathbf{L}_{ij}\mathbf{x} - \mathbf{R}_i\mathbf{d}_{ij}\|}{\|\mathbf{d}_{ij}\|}.$$

When ρ_{ij} exceeds a given yield threshold c_{yield}, we basically need to change its rest state according to the deformation of the current configuration. But we don't need to directly do this in practice. Instead, we can achieve the same goal by updating the pre-computed differential coordinates $\mathbf{d}_{ij}$ as follows:

$$\mathbf{d}_{ij} \leftarrow \mathbf{d}_{ij} + c_{creep}(\rho_{ij} - c_{yield})\left(\mathbf{L}_{ij}\mathbf{x} - \mathbf{d}_{ij}\right), \quad (6)$$

where c_{creep} is the plasticity coefficient. In order to correctly estimate the rotation matrices, $\mathbf{r}_{ij}$ needs to be updated accordingly to keep the relationship: $\mathbf{r}_{ij} = \frac{m_j}{\omega_{ij}}\mathbf{d}_{ij}$.

The plasticity effect can be controlled by the two parameters c_{yield} and c_{creep} [26]. Fig. 3 demonstrates an example of plasticity simulation.

4 Dimensionality Reduction

It is nevertheless slow to directly solve the simulation by Eq. (2) for a large number of nodes. At the same time, we may run into stability problems when the shape is extremely deformed. To address these critical issues, we can adopt some dimensionality reduction methods.

Let Φ be a matrix whose column vectors are the deformation bases. We first represent the deformation by $\mathbf{x} = \mathbf{r} + \Phi\mathbf{z}$,

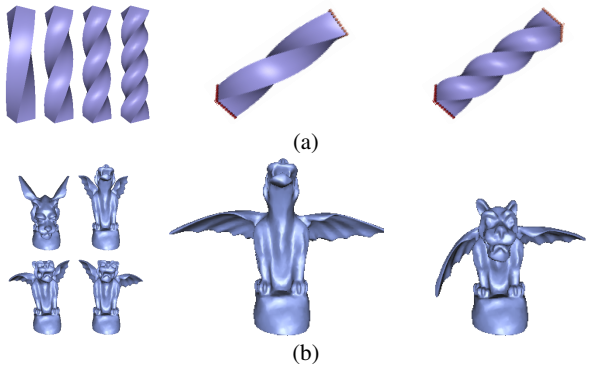


Fig. 4 Simulation results for the bar model (top row) and the Gargoyle model (bottom row) with sample based dimensionality reduction. Four user-provided samples are used, as shown in the left column. The middle and right columns show two deformation results.

where $\mathbf{z}$ is a vector of unknown coefficients. Then, we have the following simulation system at each step:

$$\Phi^t \left(\mathbf{M} + h\mathbf{D} + h^2\mathbf{K}\right) \Phi \Delta\dot{\mathbf{z}} = h\Phi^t \left(\mathbf{f}_{ext} - \mathbf{D}\dot{\mathbf{x}} - h\mathbf{K}\dot{\mathbf{x}} - \frac{\partial}{\partial \mathbf{x}} V(\mathbf{x})\right). \quad (7)$$

This is a much smaller linear system and can be solved more efficiently and robustly.

We have employed two methods to obtain the deformation bases. One is based on some input deformation samples [3,21]. Given m samples $\{\mathbf{s}_j\}_{j=1}^m$, we first apply mass-PCA [3] on the deformation vectors $\{\mathbf{s}_j - \mathbf{r}\}_{j=1}^m$, then choose the most significant principal components as the deformation bases. Fig. 4 shows two deformation results with the sample based dimensionality reduction method.

The other is based on modal analysis [27,14], which is useful when no example is available. Similar to the standard modal analysis method, we first solve the generalized eigen problem $\mathbf{M}\Psi Ł\lambda = \mathbf{K}\Psi$, where the columns of Ψ are the generalized eigen vectors, and $Ł\lambda$ is a diagonal matrix consisting of the eigen values. Then we analyze the eigen values, and select some significant eigen vectors (corresponding to small eigen values) as the deformation bases.

For the traditional modal analysis method of physically-based simulation, there are 6 eigen values that are zeros, and the corresponding eigen vectors represent the six rigid transformation modes. But in our method, the x/y/z components are processed separately, so there are only one zero eigen value for each component, which corresponds to the translation bases.

To add the rotational bases into the deformation space, we can combine it into the example based method as follows: first we generate rotation examples by rotating the object around the x/y/z axes, then regenerate the deformation bases by the above sample based method with the rotation samples and the bases generated in modal analysis. Fig. 5 shows deformation results with our modal analysis based dimensionality reduction method.

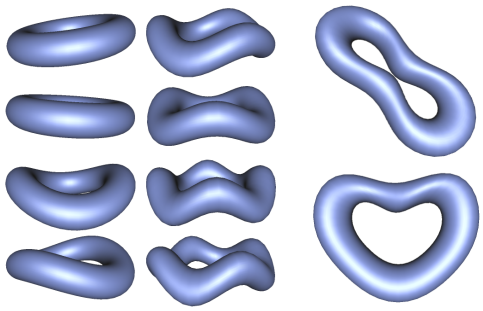


Fig. 5 Simulation result with modal analysis based dimensionality reduction. Eight non-trivial deformation bases are shown in the left two columns, and two deformation results are shown in the right column.

5 Spatially Adaptive Method

When the dimensionality reduction method is applied, the most expensive computations lie in the phase of estimating the local rotation for each node, which involves computing matrix $\mathbf{A}_i^{\mathbf{xr}}$ and performing polar decomposition on it.

Based on the observation that nearby nodes usually share very similar local rotations, we propose an adaptive scheme to reduce the computation cost in the rotation estimation phase. Our scheme is very different from the adaptive methods in previous work [6,10,13,30], which all need a subdivision structure or multi-resolution mesh representation. We propose an efficient greedy activation method without the requirement of the hierarchical structure. The basic idea behind it is to select some active nodes and estimate their local rotation, then propagate their rotation to elsewhere during the breadth-first graph travel. A few seed nodes can be randomly selected before simulation. In all of our results, the seed node is just the index zero node. For the sake of symmetry or more accuracy, seeds can be chosen symmetrically or at tips of the object.

Propagation Procedure We first estimate the rotation for the seed nodes and put them in a active node queue. Then we start with an active node, and traverse the graph $\mathcal{G}$ in a breadth-first manner. Let $\mathbf{R}_c$ be the rotation of the current active node. When visiting a node whose rotation is not determined, we first judge if $\mathbf{R}_c$ is applicable to it. If yes, then set $\mathbf{R}_c$ to it. Otherwise, apply the polar decomposition method to estimate its rotation. Then we put it into the active node queue. This process is repeated until all nodes have rotation information.

Adaptive Criteria There are several possible ways to judge if the rotation of node i is applicable to node j before doing polar decomposition for node j. The naive approach is to directly compare $\mathbf{A}_i^{\mathbf{xr}}$ and $\mathbf{A}_j^{\mathbf{xr}}$ or compare $\mathbf{A}_i$ and $\mathbf{A}_j$. But both $\mathbf{A}_{i/j}^{\mathbf{xr}}$ and $\mathbf{A}_{i/j}$ contains scaling information, the naive criterion produce unsatisfactory results, as shown in Fig. 6(a) and (b), which cannot accurately adapt nodes to the deformation.

Recall that polar decomposition will factorize $\mathbf{A}_j^{\mathbf{xr}}$ into a rotation $\mathbf{R}_j$ and a symmetric transformation $\mathbf{S}_j$, i.e. $\mathbf{A}_j^{\mathbf{xr}} = \mathbf{R}_j\mathbf{S}_j$. Denote $\mathbf{S}_{ij} = \mathbf{R}_i^t\mathbf{A}_j^{\mathbf{xr}}$. If the rotation $\mathbf{R}_i$ is applicable

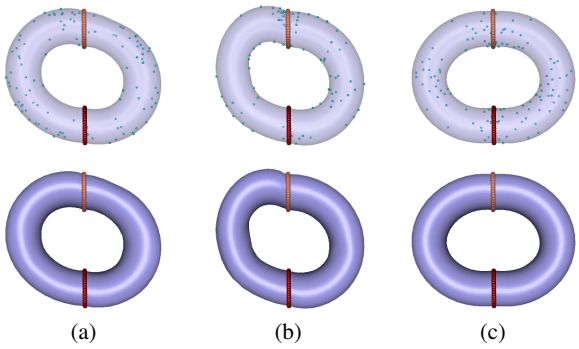


Fig. 6 Comparison of active nodes (shown as small dots in cyan) with three adaptive criterion: (a) naive criterion I:$\mathbf{A}_i^{\mathbf{xr}} - \mathbf{A}_j^{\mathbf{xr}}$, (b) naive criterion II: $\mathbf{A}_i - \mathbf{A}_j$, (c) rotation sensitive criterion. The bottom row shows the corresponding deformed shapes.

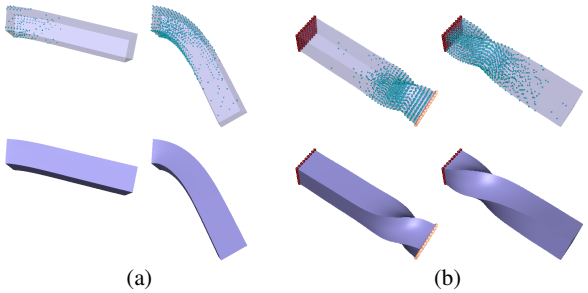


Fig. 7 Distribution of adaptively activated nodes. The corresponding deformations are shown in the bottom row.

to node j, i.e., $\mathbf{R}_i \approx \mathbf{R}_j$, then $\mathbf{S}_{ij} = \mathbf{R}_i^t \mathbf{R}_j \mathbf{S}_j \approx \mathbf{S}_j$ is very close to a symmetric matrix. Therefore, we propose the following criterion:

$$\left\| \mathbf{R}_i^t \mathbf{A}_j^{\mathbf{xr}} - \left(\mathbf{R}_i^t \mathbf{A}_j^{\mathbf{xr}} \right)^t \right\|^2. \tag{8}$$

We call it a rotation sensitive criterion, since it basically compares the rotation part. If the error is below the specified threshold (we choose the threshold from $1e^{-2}\|A_i^{\mathbf{rr}}\|^2$ to $1e^{-4}\|A_i^{\mathbf{rr}}\|^2$), then $\mathbf{R}_i$ is propagated from node i to node j.

Fig. 6(c) shows that more nodes in the large deformation area are activated with the rotation sensitive criterion. Compared with Fig. 6(a) and (b), the rotation sensitive criterion performs much better. More examples are showed in Fig. 7 and Fig. 8.

The cost of the traveling the graph depends on the deformed shape and the threshold. We setup some typically cases to measure the efficiency of our adaptive method. For the gargoyle model with 14190 nodes, the polar decomposition for all the nodes consumes about 58ms. When the approximation is always acceptable (for very large threshold or keeping the object undeformed), the traveling needs 7ms, and on the contrary, needs 16ms when no approximation is acceptable (set very small threshold for large deformation). There's no gain when number of polar decomposition cannot decrease below 75% or so. The spatial adaptive rotation tracking works better when applying dimensionality reduction, because the variation of local transformations is smoother than in full dimensionality simulation. But in full dimensionality simulation, it still can cut down many nodes. Less than 30% nodes are needed to be tracked for a moderate deformation without noticeable artifacts.

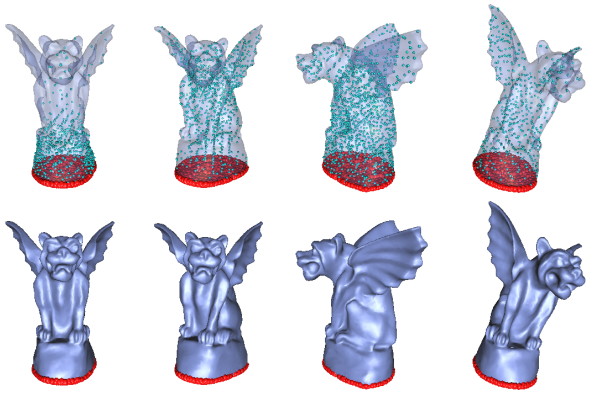


Fig. 8 Distribution of adaptively activated nodes. The corresponding deformations are shown in the bottom row.

	dimensionality	ratio of active nodes	time per step (ms)
Fig.1	full	-	33ms
Fig.2(b)	full	-	90ms
Fig.2(c)	full	-	46ms
Fig.3	full	-	60ms
Fig.4(a)	12 basis	-	18ms
Fig.4(b)	12 basis	-	95ms
Fig.5	8 basis	-	20ms
Fig.6(c)	full	4%	44ms
Fig.7(a)	10 basis	4%	9.2ms
Fig.7(b)	12 basis	10%	9.6ms
Fig.8	40 basis	10%	76ms
Fig.9(a)	full	-	19ms
Fig.9(b)	40 basis	10%	76ms

Table 1 Configuration and performence statistics for the results in this paper. The performance is measured on a 3.06GHz Intel Xeon machine with 2GB memory.

6 Conclusion

We have presented a geometrically-based potential energy function for simulating deformable objects. Our algorithm can achieve visually pleasing effects and runs interactively (see the accompanying video for animation demos). The statistics of the performance data and some experimental models are listed in Table 1 and Table 2. More simulating results are shown in Fig. 9. For the bar model, dimensionality reduction and the adaptive method can accelerate the simulation from 33ms per step to 9.6ms.

In our fomulation of the deformation energy, we do not take into account the volume, which will be a problem for volume preserving deformations. Our dimensionality reduction methods need to precompute some deformation examples or the sigular value decomposition of the stiffness matrix, which are not suitable for processing extremely large models.

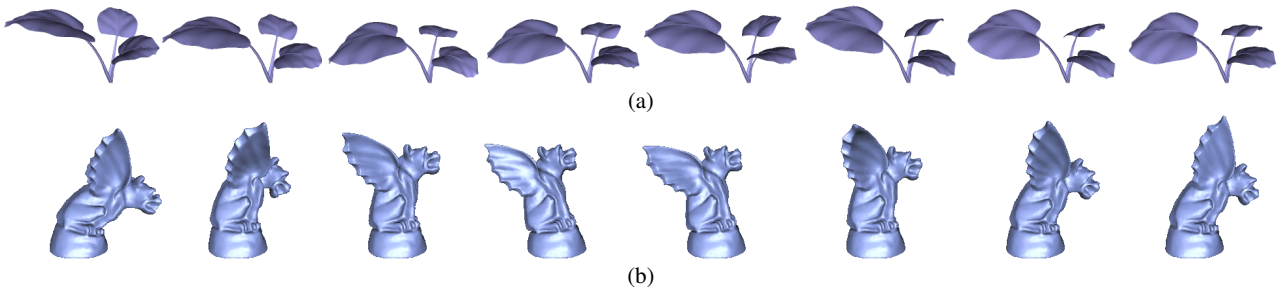


Fig. 9 The top row is the plant model vibrating under gravity solved in full dimension. The bottom row is the animation results of gargoyle model using 14 samples.

	bar	tweety	gargoyle	torus	plant	santa
node number	3321	5001	14190	3427	2606	4292
model type	VG	SM	VG	VG	NSM	VG

Table 2 The characteristics of the models used in our paper. VG: Volumetric Graph; SM: Surface Mesh; NSM: Non-manifold surface mesh.

We believe that this work will open an interesting new direction in simulating deformable objects. There are many avenues for future work. For example, we would like to preserve angular momentum in our simulation framework. And, it is worthwhile to investigate new geometrical energy functions for simulating specific material effects, e.g., materials with anisotropic structures, and to further investigate more acceleration methods to handle large scale simulation tasks.

Acknowledgements This project is partially supported by Natural Science Foundation of China under Grant No. 60021201, The National Basic Research Program of China (973 Program) under Grant No. 2002CB312102, and Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China under Grant No. 705027.

References

1. Alexa, M.: Differential coordinates for local mesh morphing and deformation. The Visual Computer **19**(2), 105–114 (2003)
2. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 43–54. ACM Press (1998)
3. Barbič, J., James, D.L.: Real-time subspace integration for st. venant-kirchhoff deformable models. ACM Trans. Graph. **24**(3), 982–990 (2005)
4. Bro-Nielsen, M., Cotin, S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. Computer Graphics Forum **15**(3), 57–66 (1996)
5. Capell, S., Green, S., Curless, B., Duchamp, T., Popovic, Z.: Interactive skeleton-driven dynamic deformations. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 586–593. ACM Press (2002)
6. Capell, S., Green, S., Curless, B., Duchamp, T., Popovic, Z.: A multiresolution framework for dynamic deformations. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 41–47. ACM Press (2002)
7. Choi, K.J., Ko, H.S.: Stable but responsive cloth. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 604–611. ACM Press (2002)
8. Choi, M.G., Ko, H.S.: Modal warping: Real-time simulation of large rotational deformation and manipulation. IEEE Trans. Vis. Comput. Graph. **11**(1), 91–101 (2005)
9. Cotin, S., Delingette, H., Ayache, N.: Real-time elastic deformations of soft tissues for surgery simulation. IEEE Transactions On Visualization and Computer Graphics **5**(1), 62–73 (1999)
10. Debunne, G., Desbrun, M., Cani, M.P., Barr, A.H.: Dynamic real-time deformations using space & time adaptive sampling. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 31–36. ACM Press (2001)
11. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 317–324. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1999)
12. Gibson, S.F.F., Mirtich, B.: A survey of deformable modeling in computer graphics. Tech. Rep. TR-97-19, Mitsubish Electric Research Lab., Cambridge (1997)
13. Grinspun, E., Krysl, P., Schröder, P.: Charms: a simple framework for adaptive simulation. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 281–290. ACM Press (2002)
14. Hauser, K., Shen, C., O'Brien, J.F.: Interactive deformations using modal analysis with constraints. In: Proceedings of Graphics Interface 2003, pp. 247—256 (2003)
15. Huang, J., Liu, X., Bao, H., Guo, B., Shum, H.Y.: Clustering method for fast deformation with constraints. In: Proceedings of the 2005 ACM symposium on Solid and physical modeling, pp. 221–226. ACM Press, New York, NY, USA (2005)
16. Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.Y., Teng, S., Bao, H., Guo, B., Shum, H.Y.: Subspace gradient domain mesh deformation. ACM Trans. Graph. **25**(3) (2006)
17. Huang, J., Zhang, H., Shi, X., Liu, X., Bao, H.: Interactive mesh deformation with pseudo material effects. Computer Animation and Virtual Worlds **17**(3-4), 383–392 (2006)
18. 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, pp. 65–72. ACM Press/Addison-Wesley Publishing Co. (1999)
19. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: Proceedings of Shape Modeling International, pp. 181–190 (2004)
20. Metaxas, D., Terzopoulos, D.: Dynamic deformation of solid primitives with constraints. Computer Graphics (Proc. SIGGRAPH'92) **26**(2), 309–312 (1992)
21. Miao, L., Huang, J., Liu, X., Bao, H., Guo, B.: Computing variation modes for point set surfaces. In: Eurographics Symposium on Point Based Graphics, pp. 63–69 (2005)
22. Müller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B.: Stable real-time deformations. In: Proceedings of the 2002 ACM

SIGGRAPH/Eurographics symposium on Computer animation, pp. 49–54. ACM Press (2002)
23. Müller, M., Gross, M.H.: Interactive virtual materials. In: Graphics Interface, pp. 239–246 (2004)
24. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. ACM Trans. Graph. **24**(3), 471–478 (2005)
25. Nealen, A., Müller, M., Keiser, R., Boserman, E., Carlson, M.: Physically based deformable models in computer graphics. In: Eurographics 2005 state of the art report (STAR) (2005)
26. O'Brien, J.F., Bargteil, A.W., Hodgins, J.K.: Graphical modeling and animation of ductile fracture. In: SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 291–294. ACM Press, New York, NY, USA (2002)
27. Pentland, A., Williams, J.: Good vibrations: model dynamics for graphics and animation. In: Proceedings of the 16th annual conference on Computer graphics and interactive techniques, pp. 215–222. ACM Press (1989)
28. Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proceedings of the Eurographics symposium on Geometry processing, pp. 179–188. Eurographics Association (2004)
29. Terzopoulos, D., Witkin, A.: Physically based models with rigid and deformable components. IEEE Comput. Graph. Appl. **8**(6), 41–51 (1988)
30. Wu, X., Downes, M.S., Goktekin, T., Tendick, F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. Comput Graphics Forum **20**(3), 349–358 (2001)
31. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.Y.: Mesh editing with poisson-based gradient field manipulation. ACM Trans. Graph. **23**(3), 644–651 (2004)
32. Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., Shum, H.Y.: Large mesh deformation using the volumetric graph laplacian. ACM Trans. Graph. **24**(3), 496–503 (2005)

Jin Huang is a Ph.D candidate in the state key laboratory of CAD&CG at Zhejiang University, P.R.China. He received B.S. degree in computer science department from Zhejiang University. His research interests include geometric deformation and physical-based simulation.

Xiaohan Shi is a Ph.D candidate in the state key laboratory of CAD&CG at Zhejiang University, P.R.China. He received B.S. degree in Chu Kechen Honors College from Zhejiang University. His research interests include geometric deformation and user interface.

Xinguo Liu received a B.Sc. in 1995 and a Ph.D. in 2001 from the Department of Applied Mathematics in Zhejiang University. He is currently in the State Key Lab of CAD&CG, and is a Professor of the Computer Science School in Zhejiang University. Before joined CAD&CG in 2006 April, he was a researcher of the Internet Graphics Group in Microsoft Research Asia. His main research interests are in geometry processing, appearance modeling, real-time rendering, and deformable objects.

Baining Guo is the research manager of the internet graphics group at Microsoft Research Asia. Before joining Microsoft, Baining was a senior staff researcher in Microcomputer Research Labs at Intel Corporation in Santa Clara, California, where he worked on graphics architecture. Baining received his Ph.D. and M.S. from Cornell University and his B.S. from Beijing University. Baining is an associate editor of IEEE Transactions on Visualization and Computer Graphics. He holds over30 granted and pending US patents.

Hujun Bao received his Bachelor and Ph.D in applied mathematics from Zhejiang University in 1987 and 1993. His research interests include modeling and rendering techniques for large scale of virtual environments and their applications. He is currently the director of State Key Laboratory of CAD&CG of Zhejiang University. He is also the principal investigator of the virtual reality project sponsored by Ministry of Science and Technology of China.