EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

*Volume 34 (2015), Number 2*

# Real-Time Subspace Integration for Example-Based Elastic Material

Wenjing Zhang, Jianmin Zheng, Nadia Magnenat Thalmann

School of Computer Engineering & IMI, Nanyang Technological University, Singapore

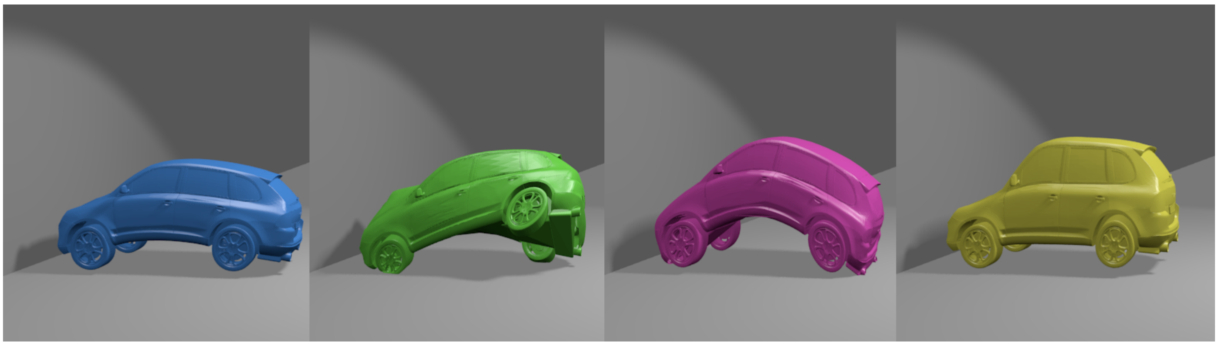

**Figure 1:** *Different example-based elastic material behaviours illustrated on deforming cars.*

**Abstract**

*Example-based material allows simulating complex material behaviors in an art-directed way. This paper presents a method for fast subspace integration for example-based elastic material, which is suitable for real-time simulation in computer graphics. At the core of the method is the formulation of a new potential using example-based Green strain tensors. By using this potential, the deformation can be attracted towards the example-based deformation feature space, the example weights can be explicitly obtained and the internal force can be decomposed into the conventional one and an additional one induced by the examples. The real-time subspace integration is then developed with subspace integration costs independent of geometric complexity, and both the reduced conventional internal force and additional one being cubic polynomials in reduced coordinates. Experiments demonstrate that our method can achieve real-time simulation while providing comparable quality with the prior art.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

## 1. Introduction

Simulating deformable objects is a common task in computer graphics and animation. In some applications, it is preferred for deformable objects to deform in an artistic way. Theoretically deformation can be controlled by the material properties, but setting art-directable materials is a challenging task.

The concept of example-based material was introduced to provide artistic control of physically-based animation using examples [MTGG11, STC*12]. The basic idea is that material behaviours can be implicitly specified by providing deformation examples and this setting is more intuitive than quantifying the material parameters directly. Martin et al. [MTGG11] proposed to construct an example space by interpolating example poses provided by users to identify how they wanted the object to deform. During the simulation, the current configuration of the object was projected onto the example space. The projection was used to generate additional forces which attracted the object towards the example space. Since reconstructing 3D shapes from the interpolation in the example space was time consuming, Schumacher et al. [STC*12] proposed incompatible interpolation to bypass the costly reconstruction. However, the new method is still too slow for real-time applications. It takes nearly twice the time required by the simulation without examples. This is because the method uses the projection as another rest shape and needs to evaluate the additional force besides the elastic force with respect to the real rest pose.

This paper is targeted at real-time simulation of example-based material under the finite element framework. To achieve this, we propose a new method consisting of two technical components:

- By interpolating the Green strain tensors, we construct the example-based potential energy directly in the deformation feature space without performing projection. As a result, the example weights can be explicitly obtained and the additional internal force induced by the example-based potential is cubic in the current configuration. The full-rank simulation only requires modest additional costs to the simulation without examples.
- A subspace integration is developed, which allows us to use model reduction to achieve real-time simulation. Note that model reduction is not readily available for the example-based material models of [MTGG11, STC*12]. This is because the example-based potentials there are based on a *changing* projection as a rest pose and model reduction involves a time-consuming precomputation based on the rest pose. A key feature of our example-based potential is that it is friendly for subspace integration. Particularly, the internal potential of our method can be decomposed into two parts: the conventional St.Venant-Kirchhoff (StVK) elastic potential and the example-based potential. The former can be evaluated in real-time using cubic integration [BJ05] or cubature integration [AKJ08]. The latter is a product of two quadratic polynomials in the reduced coordinates. Evaluation can be performed in $O(r^2)$ time complexity where $r$ is the dimension of the reduced model. The integration cost is then independent of the geometric complexity.

The experiments have demonstrated the effectiveness and the efficiency of the proposed method.

## 2. Related Work

### 2.1. Simulation of deformable models

Simulating the deformation of soft objects has been studied extensively in computer graphics [TPBF87, NMK*06]. An important component in physically-based simulation is material models that describe the relationship between geometry deformation and the resulting force [MG04]. Common material models include Saint Venant-Kirchhoff, Neo-Hookrean, and Mooney-Rivlin [BW97], each of which is quantified by a set of material parameters. The user can control the deformation behaviours by adjusting the parameters, which is however nonintuitive. To generate desired deformation behaviours, the animator may need to tune the unwieldy material parameters. Bickel et al. [BBO*09] presented an interesting approach to learning material properties from experiments. For some artistic design, nevertheless, there is even no real world counterpart.

Martin et al. [MTGG11] proposed example-based elastic materials in an finite element method (FEM) approach, which allows users to implicitly control the material behavior by specifying a set of example poses to indicate how an object is expected to deform. Nonlinear Green strain is used to create a deformation space and the deformable shapes are computed by minimizing an energy functional consisting of the elastic energy and an additional energy that reflects the examples effects. The underlying computation involves nonlinear optimization. Schumacher et al. [STC*12] eased the computation by performing element-wise interpolation. Our work stems from the same objective but aims to achieve significant speedup with subspace integration.

Following the similar ideas, Song et al. formulated example-based deformation based on a modified linear Cauchy strain, which can avoid complex non-linear optimization [SZW*14]. Zhu et al. performed example-based deformation in a shape space spanned by the Laplace-Beltrami eigenfunctions, which does not require examples to have the same topology [ZLW14]. Jones et al. proposed dynamic sprites for creating dynamic objects and characters from static drawings, in which artistic control is achieved by allowing the artist to specify a set of example poses and the navigation among the poses [JPM*13]. To realize real-time simulation, Koyama et al. formulated the concept of example-based materials using the shape matching framework [KTUI12]. With shape matching, the deformation descriptor is defined as a local region's right stretch tensor and the pose space is represented as a linear combination of examples. At each time step the projection is also linear. Different from Koyama et al.'s work, our work follows Martin et al.'s FEM-based approach. However, we specially design the example-based potential resulting in a quadratic minimization problem, which is similar to [KTUI12]. The idea of designing special energy potentials that can be solved efficiently can also be found in [BML*14] for fast implicit time integration of general physical systems.

### 2.2. Fast simulation

A lot of research has been conducted to improve the speed of simulating deformable models. Condensation is used to ignore the inner nodes [BNC96]. Embedding methods use a coarse mesh for simulation while embedding a fine mesh into the coarse mesh for rendering [SDF07]. Multi-resolution approaches adaptively refine the mesh based on the deformation state [GKS02]. Subspace techniques use a low-dimensional subspace to simplify the deformation.

In particular, linear modal analysis was introduced to computer graphics for simple and fast simulation of dynamic deformation [PW89]. Via eigendecomposition, a set of vibration modes corresponding to the low vibration frequencies is used to approximate the dynamic deformation [HSO03]. The number of the selected vibration modes is small compared to the size of the model and the resulting motion equations can be integrated analytically. However, linear modal analysis leads to distortions for large rotational deformation. Modal warping was proposed to overcome this limitation by warping the linear modes with a ramped rotation [CK05]. Efficient subspace integration that preserves nice properties of linear modal analysis was proposed by Barbič and James who exploited the fact that for StVK material, the elastic forces are cubic polynomials in the reduced coordinates [BJ05]. Our work proposes a new example-based potential that retains the benefits of StVK material.

### 2.3. Controlling animation

Animation satisfying some constraints is important in practice. Key frame animation requires the user to provide "keys": the deformation of the object at certain time points [WK88, WMT06]. Interpolation is used to generate in-betweens. Space-time constraints provide this control to physically based deformation. An optimal control force is computed to ensure the objects to satisfy the keys. Such an optimization process is time consuming. Babič et al. proposed to perform optimization with a subspace model [BdSP09]. An analytic solution was proposed to gain high efficiency [HSvTP12]. Coros et al. proposed to use internal deformations to drive motions [CMT*12]. Kondo et al. [KKA05] proposed to edit the motion by keyframing the rest shape. These methods specify the trajectories the object should move along and provide fine level control of animation. Our method, on the other hand, encourages the object to deform as desired in forward simulation.

### 2.4. Example-based mesh deformation

While typical mesh editing performs deformation using some geometric criteria [SCOL*04, SA07], Sumner et al. [SZGP05] proposed an example-based approach. An example space is constructed by nonlinear span of a set of example poses. When the user moves a subset of vertices, the algorithm searches the example space to find the one that best satisfies the user's constraints. High-level information can be incorporated with example poses. Fröhlich [FB11] combines physically based deformation with the example-based approach to elegantly solve the artefact when the deformation leaves away from the example space. While these methods are mostly designed for static geometry deformation, our work mainly focuses on dynamic simulation.

## 3. New Example-based Material

The input to our method is a deformable object represented by a tetrahedral mesh with $n$ vertices and $m$ tetrahedra. The examples consist of $k$ poses $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_k\}$ which have the same connectivity as the input tetrahedral mesh. Our goal is to achieve real-time simulation for example-based StVK material for the deformable object. The basic ideas in our approach include (1) exploring subspace techniques to accelerate the simulation and (2) designing an example-based potential that attracts the deformable object towards the desirable deformation characterized by the examples and meanwhile supports subspace integration conveniently. This section focuses on a new formulation of example-based material and the next section describes the real-time subspace integration.

Let $\mathbf{x} \in R^{3n}$ represent the vertex positions of the current tetrahedral mesh and $\mathbf{X} \in R^{3n}$ represent the vertex positions in the rest configuration. The deformation induced by configuration $\mathbf{x}$ can be measured per tetrahedron by Green strain tensor

$$G(\mathbf{X}, \mathbf{x}) = \frac{1}{2}\left(F^T F - I\right) \tag{1}$$

where $F(\mathbf{X}, \mathbf{x}) = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ is the deformation gradient and $I$ is the $3 \times 3$ identity matrix. $G(\mathbf{X}, \mathbf{x})$ is constant per tetrahedron, and is invariant under translation and rotation. Without causing ambiguity, denote by $E_i$ the vector representation of the Green strain tensor $G(\mathbf{X}, \mathbf{x})$ for tetrahedron $i$. Furthermore, we let $E(\mathbf{X}, \mathbf{x}) = [E_1, E_2, \cdots, E_m]^T \in R^{6m}$ denote a column vector of dimensional $6m$ for the global strain, which combines all the elemental strains and forms a descriptor of the deformation.

The motion of the deformable object can be described by the second order system of differential equations:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \frac{\partial W_{in}}{\partial \mathbf{x}} = \mathbf{f}_{ext} \tag{2}$$

where $\mathbf{M}$ is the mass matrix that depends only on the object's mesh and mass density distribution in the rest configuration, $\mathbf{C}$ is the damping matrix (Rayleigh damping is used in this paper), $W_{in}$ is the internal potential energy, and $\mathbf{f}_{ext}$ is the external force. While in the conventional physically-based deformation the internal potential energy depends on the material property of the object, and the current and rest configurations, the approach of example-based elastic material introduces an additional elastic potential that attracts the object to the example manifold. Similar to previous work [MTGG11, STC*12], our approach is to formulate the internal potential energy as a linear combination of the conventional potential energy and a new example-based potential that encourages the desirable deformations. Specifically, we let

$$W_{in} = (1-\alpha)W_{StVK} + \alpha W_E \tag{3}$$

where $W_{StVK}$ is the StVK potential that will be described in

Section 3.1, $W_E$ is the example-based potential that will be formulated in Section 3.2, and $\alpha \in [0,1]$ is the tradeoff factor.

### 3.1. StVK potential

StVK material is characterized by a linear stress-strain relationship. For tetrahedron $i$, its elastic energy is $E_i^T D_i E_i$ where

$$D_i = \frac{V_i}{2}\begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}$$

encodes the material property with Lamé parameters $\lambda$ and $\mu$, and $V_i$ is the volume of the tetrahedron.

We construct a $6m \times 6m$ matrix $D$ which concatenates all $D_i$ along the diagonal. Then the StVK potential of the whole object is written in matrix form as

$$W_{StVK} = E^T DE. \tag{4}$$

### 3.2. Example-based potential

While the $k$ input examples serve as a guidance for deformation, we need some ways to describe the desirable deformations that reflect the characteristics of the examples. Since the global Green strain tensor $E(\mathbf{X},\mathbf{x})$ is a good descriptor of deformation, we can use all the descriptors of the examples as a basis. For example $\mathbf{P}_j$, we compute its global Green strain $E(\mathbf{X},\mathbf{P}_j)$. All these descriptor vectors form a linear space

$$\Xi = \{E_\omega | E_\omega = \sum_{j=1}^{k} w_j E(\mathbf{X},\mathbf{P}_j),\ \omega = [w_1,\cdots,w_k]^T\} \tag{5}$$

where coefficients $w_j$ are called the *example weights*. We also call $\Xi$ the *example-based deformation feature space*.

The deformation feature space provides a computational tool to measure whether the current configuration moves towards the desirable deformations. In fact, we can compute the distance of the descriptor vector of the current configuration to the deformation feature space. Considering the formulation of the StVK potential, we define the distance metric as follows:

$$d(E,\Xi) = \sqrt{\min_{\omega=[w_1,\cdots,w_k]^T} (E-E_\omega)^T D(E-E_\omega)} \tag{6}$$

where $E = E(\mathbf{X},\mathbf{x})$ is the Green strain vector of the current configuration $\mathbf{x}$.

The objective function in Eq.6 is quadratic in the example weights. After a simple derivation (see Appendix), we can obtain an explicit solution for the optimal example weights:

$$\mathbf{w} = \arg\min_{\omega}(E-E_\omega)^T D(E-E_\omega) = (\mathcal{E}_p^T D\mathcal{E}_p)^{-1}\mathcal{E}_p^T DE \tag{7}$$

where $\mathcal{E}_p$ is a matrix of dimension $6m \times k$ whose $j$-th column is $E(\mathbf{X},\mathbf{P}_j)$. In previous example-based methods [MTGG11, STC*12], the optimal weights can be obtained only numerically. Note that the weights obtained in Eq.7 are not guaranteed to be positive. If a convex combination is required, we can follow Martin et al's approach [MTGG11] to add simple quadratic energies to the objective function to enforce the constraints weakly, which may produce harmless small extrapolations. Since the added energies are quadratic, an analytic solution is still available.

Given the current configuration $\mathbf{x}$, once its corresponding optimal example weights $\mathbf{w}$ are computed by Eq.7, we can define our example-based potential to be

$$W_E = (E-E_\mathbf{w})^T D(E-E_\mathbf{w}). \tag{8}$$

The underlying consideration behind this definition is that when the Green strain vector of $\mathbf{x}$ lies in the space of the example-based deformation feature space, $W_E = 0$; and otherwise, the force caused by $W_E$ will pull the Green strain tensor towards the deformation feature space.

By expanding Eq.8 and substituting it into Eq.3, we arrive at

$$W_{in} = W_{StVK} + \alpha W_a \tag{9}$$

where

$$W_a = -2E^T DE_\mathbf{w} + E_\mathbf{w}^T DE_\mathbf{w}. \tag{10}$$

Eq.9 implies that in our example-based model the internal potential is modified by adding a new term $\alpha W_a$ to augment the influence of the examples.

### 3.3. Discussions

Different from previous work [MTGG11, STC*12], our example-based potential does not perform non-linear projection. However, our method still has two important properties. First, our potential can effectively attract the object towards the example space by penalizing the deformation outside the example-based deformation space. Second, the potential is conservative. In fact, $W_E$ is translation and rotation invariant so that it conserves momentum and angular momentum. This property is important in animation because otherwise the object may float.

While Koyama et al.'s method [KTUI12] uses the local region's right stretch tensor as the deformation descriptor and defines an example manifold as a convex hull of example deforamtion descriptors, our method uses the Green strain tensor as the deformation descriptor, based on which we define an example-based deformation feature space. As a result, both methods lead to a quadratic minimization problem, whose solution can be explicitly obtained. Though both methods can achieve real-time simulation for large models, our method is not supposed to be faster than Koyama et al.'s method. This is because our method is based on the FEM framework and Koyama et al.'s method uses shape matching. On the other hand, due to its purely gemoetrical-based nature, Koyama et al.'s method is limited in physical accuracy. By contrast, our method fits the existing FEM framework well and naturally introduces example-based elastic materials into the FEM framework. Fig. 2 shows an example that compares our method with [KTUI12]. A box bar is fixed at the top and deforms under gravity. Fig. 2(a) and (b) show the rest pose and example pose. Fig. 2(e) shows the deformation without examples, and Fig. 2(c) and (d) show the deformations obtained by our method and Koyama et al.'s method, respectively. It can be seen that our method preserves better the physical accuracy at the top of the box bar as demonstrated in the deformation without any example.

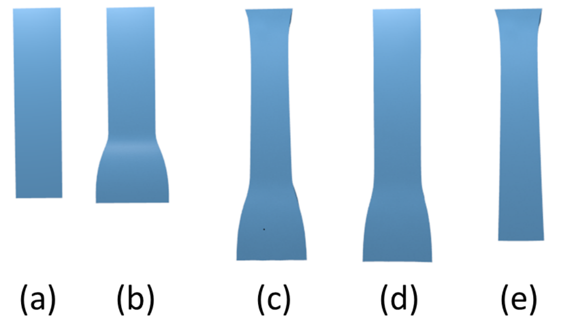


**Figure 2:** *Comparison of our method with [KTUI12]: (a) rest pose; (b) example pose; (c) deformation using our method; (d) deformation using [KTUI12]; and (2) deformation without examples.*

Similar to the incompatible interpolation [STC*12], our method also has "drift" and "resistance" problems as $E_\omega$ may not correspond to a real configuration $\mathbf{x}$. For a given $\omega$, the minimization of $W_E$ with respect to $\mathbf{x}$ may not be zero if no $\mathbf{x}$ exists for $E = E_\omega$. When the object deforms inside the example space, if the minimization of $W_E$ is not zero, $W_E$ will drift the object towards the example poses where $W_E$ is zero. Our method may also generate resistance to deformation inside the example space since moving from one example pose to another needs some amount of work to be done. To estimate the drift and resistance effects, we run the same experiment as in [STC*12]. We interpolate the rest and twisted poses of a cuboid using equidistance samples for $\omega$. For each sample we obtain $E_\omega$ and compute the corresponding deformed configuration $\mathbf{x} = \arg\min W_E(E, E_\omega)$. We record the values of $W_{StVK}$ and $W_E$, from which their derivatives with respect to $w$ are computed using finite differences. The derivatives represent the generalized forces due to weight variation and are the source of both drift and resistance effects. As shown in Fig. 3, the generalized force created by $W_{StVK}$ is significantly larger than that by $W_E$. Thus we have the similar conclusion as in [STC*12]: the drift and resistance effects are not significant in practice.

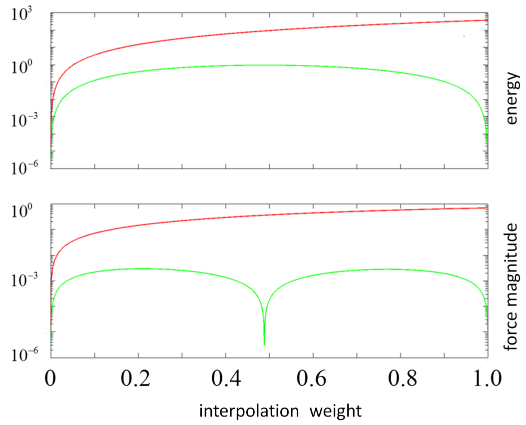


**Figure 3:** *Top: Conventional energy (red) vs example-based energy (green)). Bottom: Conventional force (red) vs example-based force (green)).*

Note that in previous work, the linear combination was given in the form of $W_{in} = W_{StVK} + \alpha W_E$. This will increase the stiffness of the model. Consider the situation where $E_\omega = 0$ and thus $W_E$ is equal to $W_{StVK}$. The stiffness is scaled by $1+\alpha$. By contrast, our formulation is free from this problem.

## 4. Subspace Integration

### 4.1. Model reduction

When the deformable object consists of a large number of vertices, it is very time-consuming to integrate the motion of Eq.2. Model reduction is a technique widely used to efficiently speed up the integration. In our case, we use a $r$-dimension ($r \ll 3n$) displacement vector to approximate the original model: $\mathbf{x} = \mathbf{X} + \mathbf{U}\mathbf{z}$, where $\mathbf{U} \in R^{3n\times r}$ is the time-independent matrix whose columns form a basis of a dimensionally reduced subspace and $\mathbf{z} \in R^r$ is the reduced (displacement) coordinates. The motion equation of Eq.2 is then transformed into:

$$\bar{\mathbf{M}}\ddot{\mathbf{z}} + \bar{\mathbf{C}}\dot{\mathbf{z}} + \frac{\partial W_{in}}{\partial \mathbf{z}} = \bar{\mathbf{f}}_{ext} \qquad (11)$$

where $\bar{\mathbf{M}} = \mathbf{U}^T\mathbf{M}\mathbf{U}$, $\bar{\mathbf{C}} = \mathbf{U}^T\mathbf{C}\mathbf{U}$ and $\bar{\mathbf{f}}_{ext} = \mathbf{U}^T\mathbf{f}_{ext}$.

In model reduction, how to construct a good reduced deformation basis is a crucial problem. There is an infinite number of possible choices for the basis matrix $\mathbf{U}$. A good deformation basis should be able to generate a low-dimensional subspace that well approximates the original deformation. However, as pointed out in [BJ05], the generation of a good deformation basis is a hard problem. Barbič and James proposed two approaches–modal derivatives and interactive sketching–for generating deformation bases [BJ05]. Modal derivatives augment the standard linear modal analysis basis by the derivatives of the linear modal basis vectors, and interactive sketching requires the user to interact with a linear vibration model, records the forces imposed by the user and generates deformation samples by an offline FEM solver.

In our example-based approach, we use examples to generate extra deformation samples. Specifically, since the deformation is expected to reflect the characteristics of the example poses, the deformation bases should take these example poses into account; otherwise no matter how we choose $\alpha$ in Eq.9, the examples may not be fully activated. To this end, for each example pose $\mathbf{P}_i$, we generate a set of shapes between $\mathbf{P}_i$ and the rest pose by

$$\mathbf{x} = \arg\min_{\mathbf{x}} (E(\mathbf{X},\mathbf{x}) - wE(\mathbf{X},\mathbf{P}_i))^T D(E(\mathbf{X},\mathbf{x}) - wE(\mathbf{X},\mathbf{P}_i))$$

with uniform sampling values in (0,1] for $w$. These shapes are then augmented by the deformation bases generated by the modal derivatives approach of [BJ05], which determine the subspace. Finally, we apply mass-PCA on all the resulting shapes to extract the mass-orthonormal deformation bases.

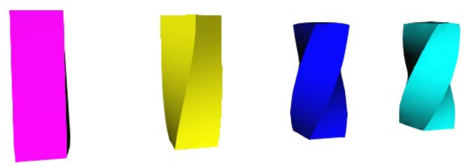

**Figure 4:** *From left to right: 10 modes, 20 modes, 40 modes (without considering examples); 20 modes (with examples).*

Fig. 4 shows the effectiveness of our deformation basis selection. The three results on the left correspond to $r = 10, 20, 40$, respectively, using the deformation bases generated by the modal derivatives method only. Augmented with examples, the modal derivatives method can produce better deformatin bases, which result in better deformation with fewer modes as shown on the right.

### 4.2. Time stepping

For dynamic simulation, we numerically integrate Eq.11, the reduced system of motion. The implicit integration of Eq.11 can be reformulated as an optimization problem that minimizes the following objective function

$$\begin{aligned} H(\mathbf{z}_{n+1},\mathbf{w}_{n+1}) &= \frac{1}{2h^2}(\mathbf{z}_{n+1}-\mathbf{y})^T\bar{\mathbf{M}}(\mathbf{z}_{n+1}-\mathbf{y}) \\ &+ \frac{1}{2h}(\mathbf{z}_{n+1}-\mathbf{z}_n)^T\bar{\mathbf{C}}(\mathbf{z}_{n+1}-\mathbf{z}_n) \\ &+ W_{StVK}(\mathbf{z}_{n+1}) + \alpha W_a(\mathbf{w}_{n+1},\mathbf{z}_{n+1}) \\ &- \mathbf{z}_{n+1}^T\bar{\mathbf{f}}_{ext} \end{aligned} \tag{12}$$

where $h$ is the step size, $\mathbf{z}_n$ and $\mathbf{z}_{n-1}$ are the reduced coordinates in previous steps, $\mathbf{z}_{n+1}$ and $\mathbf{w}_{n+1}$ are the reduced coordinates and weights in the current step that are to be solved for, and $\mathbf{y} = 2\mathbf{z}_n - \mathbf{z}_{n-1}$.

Different from the approach of [MTGG11, STC*12] where $\mathbf{z}_{n+1}$ and $\mathbf{w}_{n+1}$ are solved for simultaneously, in our formulation $\mathbf{w}_{n+1}$ can be analytically expressed in $\mathbf{z}_{n+1}$ by Eq.7. Hence we can consider $\mathbf{w}_{n+1}$ as a function of $\mathbf{z}_{n+1}$ and substituting it into Eq.12, which gives

$$\begin{aligned} H'(\mathbf{z}_{n+1}) &= \frac{1}{2h^2}(\mathbf{z}_{n+1}-\mathbf{y})^T\bar{\mathbf{M}}(\mathbf{z}_{n+1}-\mathbf{y}) \\ &+ \frac{1}{2h}(\mathbf{z}_{n+1}-\mathbf{z}_n)^T\bar{\mathbf{C}}(\mathbf{z}_{n+1}-\mathbf{z}_n) + W_{StVK}(\mathbf{z}_{n+1}) \\ &+ \alpha W_a(\mathbf{w}(\mathbf{z}_{n+1}),\mathbf{z}_{n+1}) - \mathbf{z}_{n+1}^T\bar{\mathbf{f}}_{ext}. \end{aligned} \tag{13}$$

To minimize $H'(\mathbf{z}_{n+1})$ for $\mathbf{z}_{n+1}$, Newton-Raphson iterations are performed per timestep, which involve solving a $r \times r$ linear system $\mathbf{A}\Delta\mathbf{z} = -\mathbf{b}$ with

$$\begin{aligned} \mathbf{A} &= \bar{\mathbf{M}} + h\bar{\mathbf{C}} + h^2\bar{\mathbf{K}}_{StVK} + \alpha h^2\bar{\mathbf{K}}_a, \\ \mathbf{b} &= \bar{\mathbf{M}}(\mathbf{z}_{n+1}-\mathbf{y}) + h\bar{\mathbf{C}}(\mathbf{z}_{n+1}-\mathbf{z}_n) + h^2(\bar{\mathbf{f}}_{StVK} + \alpha\bar{\mathbf{f}}_a - \bar{\mathbf{f}}_{ext}), \end{aligned}$$

$\bar{\mathbf{K}}_{StVK} = \frac{\partial^2 W_{StVK}}{\partial \mathbf{z}_{n+1}^2}$, $\bar{\mathbf{K}}_a = \frac{\partial^2 W_a}{\partial \mathbf{z}_{n+1}^2}$, $\bar{\mathbf{f}}_{StVK} = \frac{\partial W_{StVK}}{\partial \mathbf{z}_{n+1}}$, $\bar{\mathbf{f}}_a = \frac{\partial W_a}{\partial \mathbf{z}_{n+1}}$ and then updating $\mathbf{z}_{n+1} = \mathbf{z}_{n+1} + \Delta\mathbf{z}$.

Moreover, due to the formulation of our example-based potential, the evaluation of the quantities in the above processes such as the reduced internal forces can be quickly computed at runtime. Specifically, $\bar{\mathbf{f}}_{StVK}$ and $\bar{\mathbf{K}}_{StVK}$ can be computed by the runtime polynomail evaluation proposed for subspace integration of StVK deformable models in [BJ05], which has $O(r^4)$ complexity. Similarly, the computation of $\bar{\mathbf{f}}_a$ and $\bar{\mathbf{K}}_a$ can be accelerated by precomputed coefficients. Note that $\mathbf{w}$ is actually a quadratic polynomial in reduced coordinates $\mathbf{z}$. For simplicity, we introduce an auxiliary variable $\mathbf{w}' = \mathcal{E}_p^T DE$. It is a quadratic polynomial in $\mathbf{z}$ and thus can be written:

$$\mathbf{w}' = \frac{1}{2}\mathbf{z}^T\bar{\mathbf{B}}\mathbf{z} + \bar{\mathbf{q}}^T\mathbf{z} + \mathbf{c}$$

where $\bar{\mathbf{B}}, \bar{\mathbf{q}}$ and $\mathbf{c}$ are the coefficients of the polynomial, which can be precomputed. Then $\mathbf{w} = (\mathcal{E}_p^T D\mathcal{E}_p)^{-1}\mathbf{w}'$, $W_a = -\mathbf{w}^T\mathbf{w}'$, and $\bar{\mathbf{f}}_a$ and $\bar{\mathbf{K}}_a$ can be efficiently computed by:

$$\begin{aligned} &\tfrac{\partial \mathbf{w}'}{\partial \mathbf{z}} = \bar{\mathbf{B}}\mathbf{z} + \bar{\mathbf{q}}, \quad \bar{\mathbf{f}}_a = -2\mathbf{w}^T\tfrac{\partial \mathbf{w}'}{\partial \mathbf{z}}, \\ &\bar{\mathbf{K}}_a = -2\mathbf{w}^T\bar{\mathbf{B}} - 2\left(\tfrac{\partial \mathbf{w}'}{\partial \mathbf{z}}\right)^T(\mathcal{E}_p^T D\mathcal{E}_p)^{-1}\tfrac{\partial \mathbf{w}'}{\partial \mathbf{z}}. \end{aligned}$$

Compared to traditional StVK deformable models, our method just needs additional costs of evaluating $\mathbf{w}'$, $\mathbf{w}$, $\bar{\mathbf{f}}_a$ and $\bar{\mathbf{K}}_a$, which is only of $O(kr^2)$ complexity.

## 5. Experiments

This section presents some experimental results to demonstrate our method. Some results are best seen in the accompanying video. We implement our method using C++. All examples are run on a computer with Intel i7-2600K. Both reduced models and unreduced models are implemented. MKL is used to solve the linear system. The timestep size is set to 20ms. Vega Library [BSS12, SSB13] is used to compute the tangent stiffness matrix and the internal force resulting from $W_{StVK}$. The collision and global motion are treated by the method proposed in [HSO03].

**Timing**. The merit of our method is its computational efficiency. Timing for all the models in the paper is summarized in Table 1. $r = 50$ is chosen for all models. If only one iteration is used, the time integration corresponds to the classical semi-implicit methods [BW98]. More iterations should be done for higher accuracy. For unreduced models, the computation cost for multiple iterations is linearly scaled with the number of iterations. For reduced models, however, this is not the case. The full-rank external force $\mathbf{f}_{ext}$ is first projected to subspace $\bar{\mathbf{f}}_{ext}$ and after the Newton-Raphson iterations, the full-rank deformation is reconstructed with $\mathbf{x} = \mathbf{X} + \mathbf{U}\mathbf{z}$. For a single time step, these two computations should be performed only once. For a very large model, these two computations dominate the time cost. Further speedup can be achieved by using GPU for computing $\bar{\mathbf{f}}_{ext}$ and $\mathbf{x}$.

We can find in Table 1 that our example-based method only adds modest costs compared to traditional deformable models without examples for unreduced models. This is because the computation of the additional internal force and stiffness matrix can be speeded up by precomputation described in Section 4.2. The subspace integration can achieve very large speedups. Table 1 shows that the speedup for the "dog" model with 1.99K vertices and 2 examples could be 38 times and quickly becomes larger as the mesh complexity increases (for example, the speedup for the "shoe" model with 42K vertices and 2 examples could be 766 times). Note that the method of [STC*12] cannot be faster than the full-rank simulation without examples.

**Table 1:** *Statistics. From left to right: models, number of vertices, number of tetrahedra, number of examples, average time for a single iteration with unreduced models, average time for a single iteration with reduced models, and average time for computing* $\bar{\mathbf{f}}_{ext} = \mathbf{U}^T \mathbf{f}_{ext}$ *and* $\mathbf{x} = \mathbf{X} + \mathbf{U}\mathbf{z}$.

| Model | $n$ | $m$ | $k$ | Times(ms) | Time(ms) | |
|---|---|---|---|---|---|---|
| shoe | 42315 | 212710 | 0 | 8025.09 | 1.439 | 9.05 |
| shoe | 42315 | 212710 | 2 | 8049.16 | 1.441 | 9.05 |
| dog | 1992 | 6882 | 2 | 54.92 | 1.441 | 0.394 |
| cup | 1875 | 7780 | 0 | 85.19 | 1.439 | 0.381 |
| cup | 1875 | 7780 | 1 | 86.87 | 1.44 | 0.381 |
| car | 70207 | 364258 | 0 | 15961.3 | 1.439 | 15.895 |
| car | 70207 | 364258 | 1 | 16012.1 | 1.44 | 15.895 |
| teddy | 11198 | 54560 | 0 | 1109.12 | 1.439 | 2.19 |
| teddy | 11198 | 54560 | 1 | 1115.2 | 1.44 | 2.19 |

We also evaluate the performance against the number of modes and the number of example poses (see Table 2). The computation of $\bar{\mathbf{f}}_{ext}$ and $\mathbf{x}$ is excluded for better evaluation. It can be seen that the number of example poses is not the key factor of our method: 32 examples only increase the time by 1%. However, the time cost increases drastically with the number of modes.

**Table 2:** *Running time (in milliseconds) for a single iteration with different numbers of examples and modes.*

| $r$ | 1ex. | 2ex. | 4ex. | 8ex. | 16ex. | 32ex. |
|---|---|---|---|---|---|---|
| 20 | 0.0317 | 0.0319 | 0.0326 | 0.0338 | 0.0356 | 0.0394 |
| 30 | 0.0829 | 0.0834 | 0.0841 | 0.0854 | 0.0873 | 0.909 |
| 50 | 1.44 | 1.4415 | 1.4431 | 1.4457 | 1.4501 | 1.4564 |
| 70 | 4.875 | 4.880 | 4.886 | 4.896 | 4.913 | 4.945 |

**Effectiveness**. Here we provide several examples to show that our method can effectively simulate example-based artistic deformations.

In particular, Figs. 1 and 5 showcase the influence of different examples. Our method can simulate different deformation behaviours by providing different example poses. Modeling the examples is more intuitive and simpler than tuning the actual material parameters.

Fig. 6 shows that our method can effectively generate meaningful deformation styles. The hands and legs of the teddy bear are synchronized by designing appropriate examples.

Figs. 7, 8 and 9 shows the effects of multiple examples.

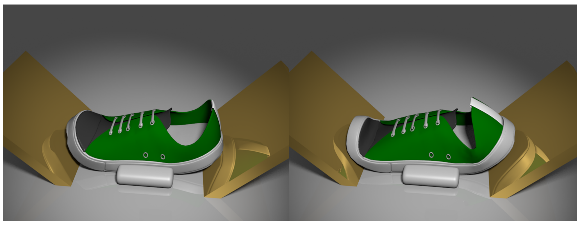

**Figure 8:** *Compressed shoes: without examples (left) and with two local examples (right).*

**Comparison.** We also compare our method with the example-based materail simulation of [STC*12]. Since in general [STC*12] cannot achieve real time as reported in its experiments, we here focus on the visual effects. Fig. 10 shows the deformations generated by our method and the method of [STC*12], which appear very similar. The comparison in animation can be found in the accompanying video. The qualitative comparison is visualized in Fig. 11, where the maximum difference is measured over the bounding box's diagonal of the models: $\sigma = \frac{\max(\mathbf{x}_i - \mathbf{x}_i^{ref})}{\rho}$ where $\rho$ is the diagonal of the bounding box of the models. For unconstrained motions, we first align $\mathbf{x}$ and $\mathbf{x}^{ref}$ to eliminate translation.

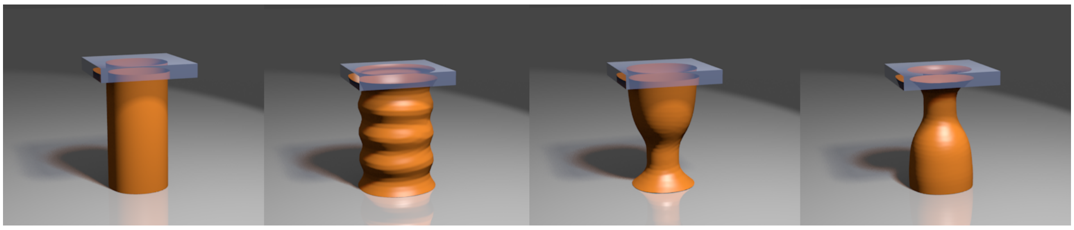

**Figure 5:** *Deformation of a cylinder guided by different examples.*

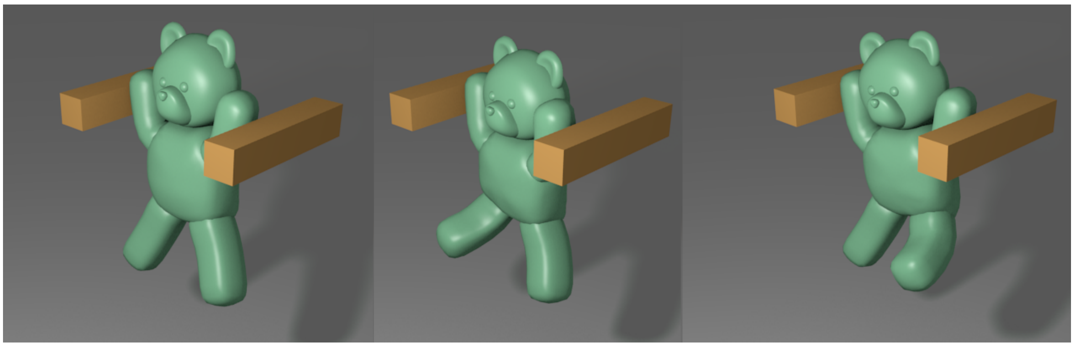

**Figure 6:** *A teddy bear falls through obstacles. The leg and arm are synchronized by the example-based control.*

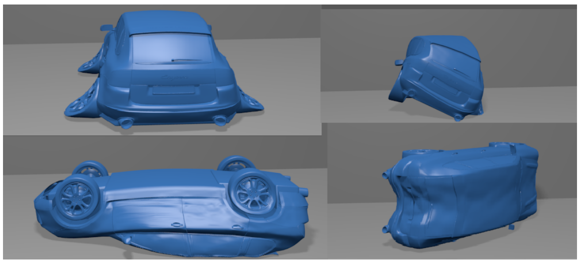

**Figure 9:** *Animation with four examples, each of which is activated in response to different impact events.*

## 6. Conclusion

We have described a method for fast simulation of example-based St. Venat-Kirchhoff deformable models. By using a new example-based potential, our method allows effective simulation of artistic deformation by providing example poses. Real-time subspace integration is readily applicable since the proposed example-based potential is just a fourth order polynomial in the reduced coordinates. Experiments show that our method can achieve real-time simulation for very large deformation.

Our method can be extended in many aspects. Currently, our method can only be applied to solid objects. Extending our method to shell models sees a promising application in both animation and geometry deformation. One limitation for subspace decomposition is that the number of modes should be limited for efficiency. The richness of deformation is thus limited. One approach may use substructuring [BZ11] to construct the reduced model for subdomains of an object or adaptive deformation basis with local deformation [HZ13] to extend our method.

**Acknowledgement**

Wenjing Zhang receives the PhD scholarship awarded by the Institute for Media Innovation (IMI), Nanyang Technological University.

**References**

[AKJ08] An S., Kim T., James D.: Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. Graph. 27*, 5 (2008), 165. 2

[BBO*09] Bickel B., Bächer M., Otaduy M., Matusik W., Pfister H., Gross M.: Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph. 28*, 3 (2009), 89. 2

[BdSP09] Barbič J., da Silva M., Popović J.: Deformable object animation using reduced optimal control. *ACM Trans. Graph. 28*, 3 (2009), 53. 3

[BJ05] Barbič J., James D.: Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Trans. Graph. 24*, 3 (2005), 982–990. 2, 3, 5, 6

[BML*14] Bouaziz S., Martin S., Liu T., Kavan L., Pauly

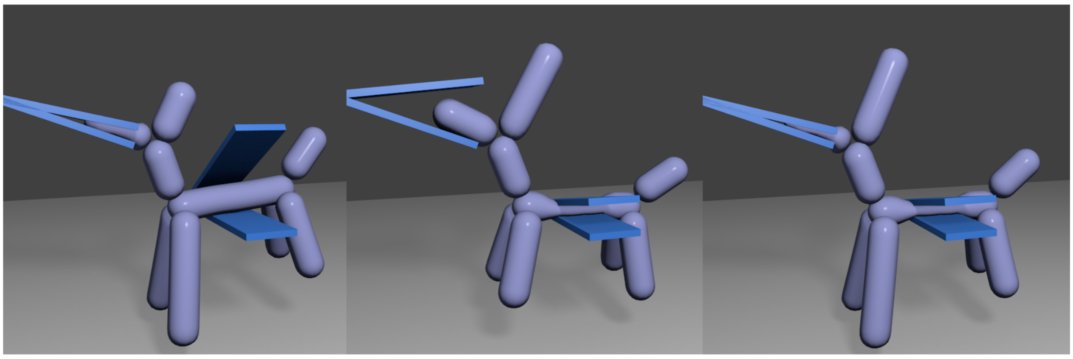

**Figure 7:** *Examples defined over unconnected regions. Left and middle: two examples are used, each of which is activated by a push. Right: two examples are activated simultaneously.*

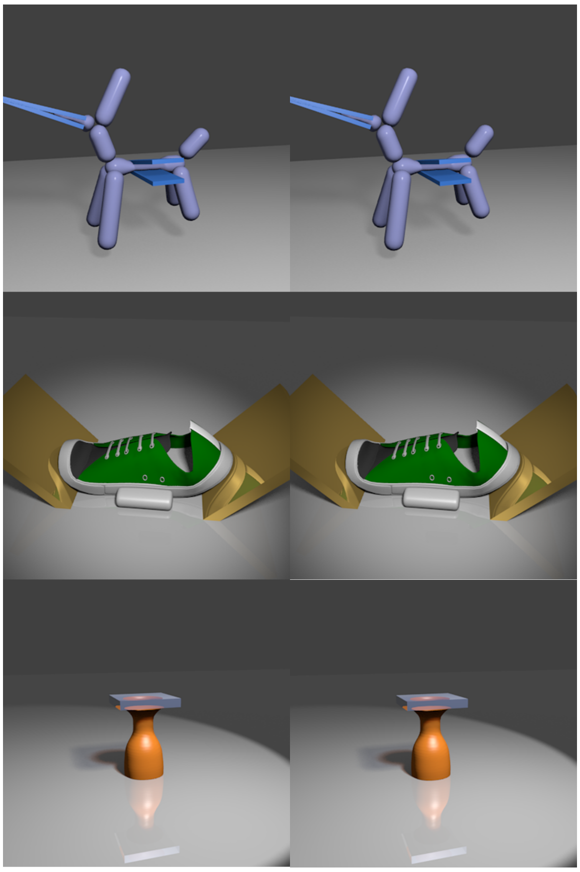

**Figure 10:** *Comparison of our method (left) and [STC*12] (right).*

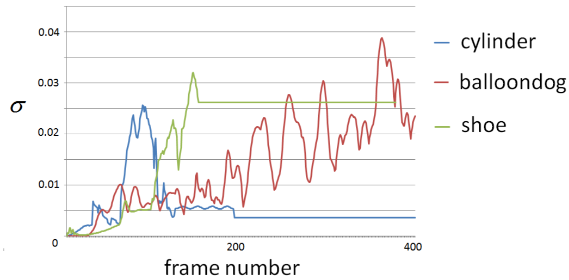


**Figure 11:** *The maximum difference σ of dynamic deformations between our method and [STC*12] on three models.*


M.: Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph. 33*, 4 (July 2014), 154:1–154:11. 2

[BNC96] BRO-NIELSEN M., COTIN S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum 15*, 3 (1996), 57–66. 2

[BSS12] BARBIČ J., SIN F. S., SCHROEDER D.: Vega FEM library, 2012. http://www.jernejbarbic.com/vega. 7

[BW97] BONET J., WOOD R.: *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997. 2

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (1998), pp. 43–54. 7

[BZ11] BARBIČ J., ZHAO Y.: Real-time large-deformation substructuring. *ACM Trans. Graph. 30*, 4 (2011), 91. 8

[CK05] CHOI M., KO H.: Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Trans Vis Comput Graph 11*, 1 (2005), 91–101. 3

[CMT*12] COROS S., MARTIN S., THOMASZEWSKI B., SCHUMACHER C., SUMNER R., GROSS M.: Deformable objects alive! *ACM Trans. Graph. 31*, 4 (2012), 69. 3

[FB11] FRÖHLICH S., BOTSCH M.: Example-driven deformations based on discrete shells. *Computer Graphics Forum* (2011). 3

[GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: CHARMS: a simple framework for adaptive simulation. *ACM Trans. Graph. 21*, 3 (2002), 281–290. 3

[HSO03] HAUSER K., SHEN C., O'BRIEN J.: Interactive deformation using modal analysis with constraints. In *Graphics Interface* (2003), pp. 16–17. 3, 7

[HSvTP12] HILDEBRANDT K., SCHULZ C., VON TYCOWICZ C., POLTHIER K.: Interactive spacetime control of deformable objects. *ACM Trans. Graph. 31*, 4 (2012), 71. 3

[HZ13] HARMON D., ZORIN D.: Subspace integration with local deformations. *ACM Trans. Graph. 32*, 4 (2013), 107:1–107:10. 8

[JPM*13] JONES B., POPOVIC J., MCCANN J., LI W., BARGTEIL A.: Dynamic sprites. In *Proceedings of Motion on Games* (2013), pp. 17:39–17:46. 2

[KKA05] KONDO R., KANAI T., ANJYO K.: Directable animation of elastic objects. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 127–134. 3

[KTUI12] KOYAMA Y., TAKAYAMA K., UMETANI N., IGARASHI T.: Real-time example-based elastic deformation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), pp. 19–24. 2, 4, 5

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Graphics Interface* (2004), pp. 239–246. 2

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM Trans. Graph. 30*, 4 (2011), 72. 1, 2, 3, 4, 6

[NMK*06] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. *Computer Graphics Forum 25*, 4 (2006), 809–836. 2

[PW89] PENTLAND A., WILLIAMS J.: *Good Vibrations: Modal Dynamics for Graphics and Animation*, vol. 23. ACM, 1989. 3

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proceedings of the fifth Eurographics Symposium on Geometry Processing* (2007), pp. 109–116. 3

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.: Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry processing* (2004), pp. 175–184. 3

[SDF07] SIFAKIS E., DER K., FEDKIW R.: Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), Eurographics, pp. 73–80. 2

[SSB13] SIN F. S., SCHROEDER D., BARBIČ J.: Vega: Nonlinear fem deformable object simulator. *Computer Graphics Forum 32*, 1 (2013), 36–48. 7

[STC*12] SCHUMACHER C., THOMASZEWSKI B., COROS S., MARTIN S., SUMNER R., GROSS M.: Efficient simulation of example-based materials. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012). 1, 2, 3, 4, 5, 6, 7, 9

[SZGP05] SUMNER R., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. In *ACM Trans. Graph.* (2005), vol. 24, ACM, pp. 488–495. 3

[SZW*14] SONG C., ZHANG H., WANG X., HAN J., WANG H.: Fast corotational simulation for example-driven deformation. *Comput. Graph. 40* (2014), 49–57. 2

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 205–214. 2

[WK88] WITKIN A., KASS M.: Spacetime constraints. *SIGGRAPH Comput. Graph. 22*, 4 (1988), 159–168. 3

[WMT06] WOJTAN C., MUCHA P., TURK G.: Keyframe control of complex particle systems using the adjoint method. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 15–23. 3

[ZLW14] ZHU F., LI S., WANG G.: Example-based materials in Laplace-Beltrami shape space. *Computer Graphics Forum*, in press (2014). 2

## Appendix: Derivation of the optimal weights

Let $g = (E - \sum_{j=1}^{k} w_j E(\mathbf{X}, \mathbf{P}_j))^T D(E - \sum_{j=1}^{k} w_j E(\mathbf{X}, \mathbf{P}_j))$. The optimal weights in (6) should satisfy the following equations for all $h = 1, 2, \cdots, k$:

$$\frac{1}{2}\frac{\partial g}{\partial w_h} = -E(\mathbf{X}, \mathbf{P}_h)^T D(E - \sum_{j=1}^{k} w_j E(\mathbf{X}, \mathbf{P}_j)) = 0$$

which give a system of linear equations

$$\sum_{j=1}^{k} \left( E(\mathbf{X}, \mathbf{P}_h)^T D E(\mathbf{X}, \mathbf{P}_j) \right) w_j = E(\mathbf{X}, \mathbf{P}_h)^T D E.$$

The linear system can be written in matrix form:

$$\begin{bmatrix} E(\mathbf{X}, \mathbf{P}_1)^T \\ \vdots \\ E(\mathbf{X}, \mathbf{P}_k)^T \end{bmatrix} D \begin{bmatrix} E(\mathbf{X}, \mathbf{P}_1), & \cdots, & E(\mathbf{X}, \mathbf{P}_k) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_k \end{bmatrix} = \begin{bmatrix} E(\mathbf{X}, \mathbf{P}_1)^T \\ \vdots \\ E(\mathbf{X}, \mathbf{P}_k)^T \end{bmatrix} DE.$$

Let $\mathcal{E}_p = [E(\mathbf{X}, \mathbf{P}_1), E(\mathbf{X}, \mathbf{P}_2), \cdots, E(\mathbf{X}, \mathbf{P}_k)]$ be a matrix of dimension $6m \times k$ whose $(i, j)$ entry is the $i$-th element of vector $E(\mathbf{X}, \mathbf{P}_j)$. Then the above linear system can be written as

$$\mathcal{E}_p^T D \mathcal{E}_p \begin{bmatrix} w_1 \\ \vdots \\ w_k \end{bmatrix} = \mathcal{E}_p^T DE.$$

Hence the optimal weights are

$$\begin{bmatrix} w_1 \\ \vdots \\ w_k \end{bmatrix} = \left( \mathcal{E}_p^T D \mathcal{E}_p \right)^{-1} \mathcal{E}_p^T DE.$$