Example-Based Elastic Materials

Sebastian Martin[1] Bernhard Thomaszewski[1,2] Eitan Grinspun[3] Markus Gross[1,2]

[1]ETH Zurich [2]Disney Research Zurich [3]Columbia University

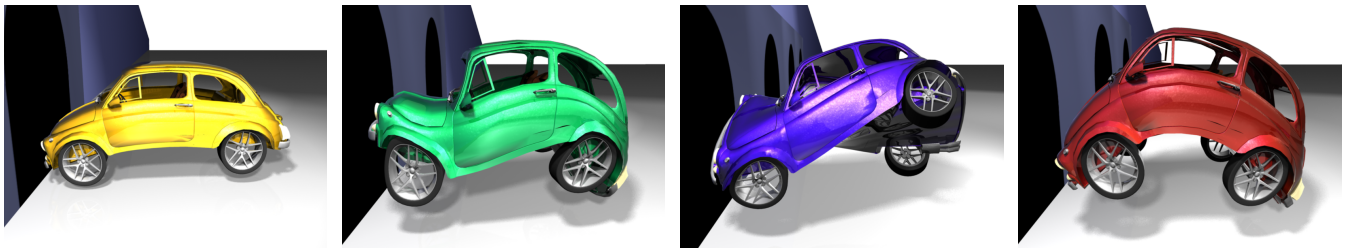

Figure 1: *Example-based materials allow the simulation of flexible structures with art-directable deformation behavior.*

Abstract

We propose an example-based approach for simulating complex elastic material behavior. Supplied with a few poses that characterize a given object, our system starts by constructing a space of prefered deformations by means of interpolation. During simulation, this example manifold then acts as an additional elastic attractor that guides the object towards its space of prefered shapes. Added on top of existing solid simulation codes, this example potential effectively allows us to implement inhomogeneous and anisotropic materials in a direct and intuitive way. Due to its example-based interface, our method promotes an art-directed approach to solid simulation, which we exemplify on a set of practical examples.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: physically-based simulation, elastic solids, control

Links: DL PDF WEB VIDEO

1 Introduction

Different materials deform in different ways. Therefore, physically-based animations offer control of material properties as a way of controlling the final deformation. But in creative applications such as computer animation, material properties are just middlemen in a process that really focuses on obtaining some desired deformation.

Indeed, we can flip the causality between materials and deformation: when we witness the deformation of an object, we implicitly draw conclusions about its underlying, constitutive material. By controlling the deformation of an animated object, we can *imply* complex material behaviors. Therefore, if we can expand the repertoire of possible deformations of an object, we can broaden the expressive palette available for physics-based computer animation.

The computational mechanics literature already describes many mathematical models for myriad materials, alas these models are intended for problems where material coefficients are easily quantified (e.g., from measurements). In artistic endeavors, we typically envision a desired *deformation* (the material properties are, to some extent, an afterthought — just a means to an end). Yet quantifying material coefficients that lead to a desired deformation behavior is difficult if not impossible. Indeed, just choosing a mathematical model can be daunting. Simpler models offer few coefficients but a small expressive range, while complex models have an unwieldy set of parameters.

Contributions Inspired by *example-based* graphical methods (for texture synthesis [Wei et al. 2009], rigging [Li et al. 2010], mesh posing [Sumner et al. 2005]), we present an intuitive and direct method for artistic design and simulation of complex material behavior. Our method accepts a set of poses that provide examples of characteristic *desirable* deformations, created either by hand (digitized from clay sculptures), with a modeling tool, or by taking 3D "snapshots" of previously run simulations. With these examples in hand, we provide a novel forcing term for dynamical integration that causes materials to obey the "physical laws" implied by the provided examples (see Fig. 1).

Our approach can be applied to "upgrade" any existing time integration code by incorporating three novel components:

- **Interpolation**: instead of restricting ourselves to individual poses, we construct a space of characteristic shapes by means of interpolation. We quantify the deformation of the example poses using a nonlinear strain measure. This *Strain Space* provides a rotation-invariant setting for shape interpolation — and the interpolated examples define a subspace of preferable deformations.
- **Projection**: having defined the space of preferable deformations, we can project configurations onto it by solving a minimization problem. Given an arbitrarily deformed pose, we can thus compute its closest point on the example subspace.
- **Simulation**: combining interpolation and projection, we can define an elastic potential that attracts an object to its space of preferable deformations. At each step of an animation, we first extract the point on the example space that is closest to the current configuration. Using this point as an intermediate rest configuration, we compute forces that pull the system toward the example space.

ACM Reference Format
Martin, S., Thomaszewski, B., Grinspun, E., Gross, M. 2011. Example-Based Elastic Materials. *ACM Trans. Graph. 30,* 4, Article 72 (July 2011), 8 pages. DOI = 10.1145/1964921.1964967 http://doi.acm.org/10.1145/1964921.1964967.

Applications The matrimony of example-based simulation and computational mechanics opens a multitude of exciting applications. Our method offers an intuitive way to design materials with desired deformation behavior. Instead of having to deal with unwieldy material descriptions, the artist can get straight to the point, specifying examples of the desired outcome. Although our formulations build on solid mechanics, we can also mimic the behavior of more complex models such as thin shells. We can embed a surface or curve into a volumetric mesh, which is then deformed such that the embedded geometry assumes the desired shapes. This enables us to control the deformation behavior of shells in very much the same way as for solids.

2 Related Work

Material models describe the connection between geometric deformations and resulting forces — they govern the way in which objects deform and are thus fundamental to every simulator. Ever since the groundbreaking works of Terzopoulos et al. [1987; 1988], extending the expressive range of materials towards elasticity [Irving et al. 2004], plasticity and viscoelasticity [Bargteil et al. 2007] or fracturing [O'Brien and Hodgins 1999] has been a primary focus. Though accurate, these conventional material models offer only limited and unwieldy control which is in opposition to the creative thinking of animators. Bickel et al. [2009] describe an interesting alternative for learning material properties from experiments, but most *artistic materials* do not have a real-world counterpart that could be subject to such measurements. Nevertheless, deducing a material description from given target deformations is a powerful concept. We follow this idea and employ examples of desired deformations to directly construct an elastic potential. Our method can be interpreted as a means of describing strongly anisotropic, heterogeneous and nonlinear materials. But while it is easy to design a set of example poses, defining a corresponding material law is a formidable task.

Directing animations is of paramount importance for setting bounds to otherwise uncontrolled physics in practical scenarios. Many methods have been proposed for this purpose, including explicit control forces [Thürey et al. 2006], space-time constraints [Witkin and Kass 1988; McNamara et al. 2004; Wojtan et al. 2006; Barbič et al. 2009] or tracking approaches [Kondo et al. 2005; Bergou et al. 2007; Barbič and Popović]. Similar goals are pursued by methods for editing animations [Popović et al. 2000; Kircher and Garland 2006] or sampling of probable animations [Twigg and James 2007]. Our approach resembles existing approaches in that it also induces additional forces into the simulation. A striking difference is, however, that these forces derive from a *conservative potential* as opposed to the *non-conservative control forces* of existing methods. As contrast to methods based on trajectory control, our approach does not require (or imply) a fixed plot of keyframes — it rather promotes a style of context-sensitive deformation control in which objects are guided towards prefered shapes but are otherwise unrestricted in their motion.

Shape Editing and Interpolation methods offer a rich repertoire of techniques for deforming meshes [Sheffer and Kraevoy 2004; Sorkine et al. 2004; Lipman et al. 2005; Botsch et al. 2006], interpolating between meshes [Alexa et al. 2000; Kilian et al. 2007; Winkler et al. 2010; Chao et al. 2010] and for transferring deformations from one mesh to another [Sumner and Popović 2004; Baran et al. 2009].

Most similar to our approach is the work of Sumner et al. [2005], whose *MeshIK* method combines shape interpolation and editing into an intuitive modeling paradigm. While we draw valuable inspiration from MeshIK, our method differs in two important aspects. First, whereas Sumner et al. interpolate between factored deformation gradients of triangles, we employ a nonlinear strain measure evaluated on tetrahedra. This is closer to the approach by Winkler et al. [2010], who interpolate between dihedral angles and edge lengths, corresponding to discrete strain measures on triangular surfaces [Grinspun et al. 2003]. Second and more important, whereas MeshIK is restricted to static geometric modeling, our approach is the first to leverage example-based methods for dynamic physical simulation.

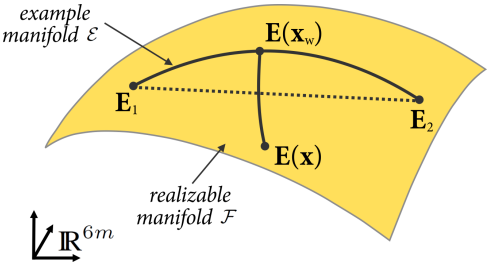

Figure 2: *Method Overview: The shape descriptor* $\mathbf{E}(\mathbf{x})$ *for the current configuration* $\mathbf{x}$ *is projected onto the example manifold spanned by* $\mathbf{E}_1$ *and* $\mathbf{E}_2$*. The projection* $\mathbf{E}(\mathbf{x_w})$ *is then used to construct an elastic potential attracting* $\mathbf{x}$ *toward the examples.*

3 Theory

Our method builds on the foundations of nonlinear continuum mechanics — but it will extend other types of simulators to handle example-based materials. The essential idea is to define an additional elastic potential that attracts a solid to its subspace of characteristic deformations, to which we refer as the *example manifold*.

We first introduce the example manifold (§3.1), then explain how to project arbitrary configurations onto it (§3.2) and finally derive the elastic example potential along with a dedicated integrator (§3.3).

Before we describe each of these components in detail, we will briefly sketch their interplay during example-based simulation. Although we eventually use an implicit solver, a clearer picture can be drawn when considering the case of explicit example-based dynamics, which is summarized in the algorithm below. See Fig. 2 for a visual illustration of the involved notions and concepts.

Algorithm 1 Explicitly integrated example-based simulation

Require: initial state $\mathbf{x}$, $\mathbf{v}$
Require: example poses $\mathbf{x}^0, \ldots, \mathbf{x}^k$
1: compute shape vectors $\mathbf{E}_i = \mathbf{E}(\mathbf{x}^i)$ // §3.1
2: **while** simulating **do**
3: $\mathbf{x_w}$ = project($\mathbf{x}$) // §3.2
4: $\mathbf{f}_{\text{ex}} = -\nabla_{\mathbf{x}} W(\mathbf{x_w}, \mathbf{x})$ // §3.3
5: compute elastic forces $\mathbf{f}_{\text{el}}$ and external forces $\mathbf{f}_{\text{ext}}$
6: step dynamics using $\mathbf{f}_{\text{tot}} = \mathbf{f}_{\text{ex}} + \mathbf{f}_{\text{el}} + \mathbf{f}_{\text{ext}}$
7: **end while**

We start by converting the k example poses into shape space descriptors $\mathbf{E}_i$ spanning the example manifold (line 1). In each simulation step, we first project the current configuration $\mathbf{x}$ onto the example manifold by minimizing (5) to obtain $\mathbf{x_w}$ (line 3). In order to compute forces $\mathbf{f}_{\text{ex}}$ that attract the current configuration to the example manifold, we temporarily use $\mathbf{x_w}$ as a rest state and construct an elastic potential $W(\mathbf{x_w}, \mathbf{x})$ (line 4). Adding forces from the conventional simulator (line 5), we step positions and velocities forward in time (line 6).

3.1 Example Manifold

A deformation is a change in shape. We will develop a definition of the *example manifold*, which describes the set of typical, desirable deformations of a solid. Before we can define this set, we need a way to think about sets of deformations.

Strain as a basis for the space of all deformations. When we think about deformations, we want to "factor out" global rotation and translations, as these do not affect *shape*. The same request also applies locally: if parts of a solid (the arms of a character) transform rigidly, they have (locally) not changed in shape. The same reasoning can be found in the construction of nonlinear deformation measures — and, indeed, the metric tensor offers exactly these desired properties [Terzopoulos et al. 1987]: it measures only local stretching and shearing and is therefore a natural basis for constructing a "space of all deformations."

Let us now formalize this construction in the discrete setting: assume that we are given a discrete representation of a solid in the form of a tetrahedral mesh with n nodes and m elements. Further, let $\mathbf{X} \in \mathbb{R}^{3n}$ and $\mathbf{x} \in \mathbb{R}^{3n}$ denote position vectors describing undeformed and deformed configurations, respectively. The deformation induced by a given configuration $\mathbf{x}$ can be quantified pointwise (equivalently) by the deformation gradient $\mathbf{F}(\mathbf{X}, \mathbf{x}) = \partial\mathbf{x}/\partial\mathbf{X}$, the rotation-invariant right Cauchy-Green ("metric") tensor $\mathbf{C}(\mathbf{X}, \mathbf{x}) = \mathbf{F}^T\mathbf{F}$, or the Green strain tensor $\mathbf{E}(\mathbf{X}, \mathbf{x}) = \frac{1}{2}(\mathbf{C}(\mathbf{X}, \mathbf{x}) - \mathbf{I})$ [Bonet and Wood 1997].

Restricting $\mathbf{x}(\mathbf{X})$ to be piecewise linear over elements, the Green strain is constant per tetrahedron. Excluding degenerate configurations (with inverted elements), the $6m$-vector of elemental strains $\mathbf{E} = [\mathbf{E}_1, .., \mathbf{E}_m]^T \in \mathbb{R}^{6m}$ fully encodes any specific deformation, i.e., it serves as a unique *descriptor* of the deformation, and in particular one that is invariant under elemental rotations.

While every deformation maps to a descriptor, the converse is false: Not every descriptor is *reconstructible* in the sense that it corresponds to a deformation. The space of reconstructible descriptors, the image of the map $\mathbf{x} \to \mathbf{E}(\mathbf{x})$, is the *realizable manifold* $\mathcal{F} \subset R^{6m}$ (see Fig. 2). This definition fulfills our first goal, for we can now refer to *sets of deformations*, in a rotation- and frame-invariant manner, by refering to subsets of $\mathcal{F}$.

Example manifold by example interpolation Suppose that we are given two specific example poses $\mathbf{x}_1$ and $\mathbf{x}_2$. We might interpolate between these examples by interpolating their descriptors $\mathbf{E}_1 = \mathbf{E}(\mathbf{x}_1)$ and $\mathbf{E}_2 = \mathbf{E}(\mathbf{x}_2)$ in R^{6m}:

$$\mathbf{E}(w) = (1 - w)\mathbf{E}_1 + w\mathbf{E}_2, \tag{1}$$

for some interpolation weight w. This approach linearly interpolates the stretch and shear of each element, and therefore results in smooth interpolation of all elements as illustrated in Fig. 3. Indeed, it can be shown that the length of any line segment inside an interpolated element is bounded by its corresponding length in the two examples.

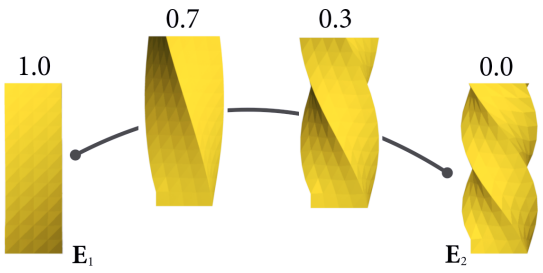

Figure 3: *Example interpolation: reconstructed geometry for different convex combinations of shape descriptors $\alpha\mathbf{E}_1 + (1 - \alpha)\mathbf{E}_2$ (as indicated above each pose).*

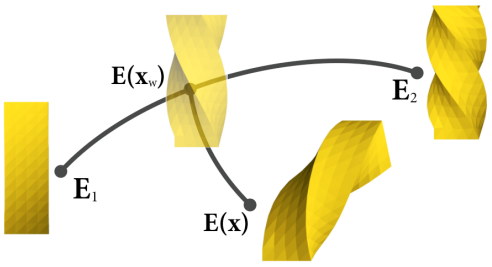

Figure 4: *Example projection: the current configuration (in strain space representation) $\mathbf{E}(\mathbf{x})$ is projected onto the example manifold, yielding the closest point $\mathbf{E}(\mathbf{x_w})$.*

Unfortunately, the interpolated descriptor is generally not realizable: Isolated elements can always satisfy the prescribed strains of the descriptor, however, assemblies of elements generally cannot. In a second step, we therefore find the closest realizable strain $\mathbf{E}(\mathbf{x}_w) \in \mathcal{F}$ and corresponding configuration $\mathbf{x}_w$ by solving the least squares minimization

$$\min_{\mathbf{x}_w} W_I(\mathbf{x}_w, w) = \min_{\mathbf{x}_w} \frac{1}{2} |\mathbf{E}(\mathbf{x}_w) - \mathbf{E}(w)|_F^2 , \tag{2}$$

where the vector norm $|\cdot|_F$ is defined to match the sum of the Frobenius norms of the elemental strain tensors. We refer to the objective function $W_I(\mathbf{x}_w, w)$ as the *interpolation energy*, whose minimization defines the projection $\Pi : R^{6m} \to \mathcal{F}$ as $w \mapsto \mathbf{x}_w$. The image of the interpolating line segment $\mathbf{E}(w)$, $w \in [0, 1]$ under the projection Π is an *example curve* $\mathbf{E}(\mathbf{x}_w)$ on the realizable manifold $\mathcal{F}$.

The procedure described above is readily generalized to an arbitrary number of poses n, where we introduce a weight w_i for each example pose $\mathbf{E}_i = \mathbf{E}(\mathbf{x}_i)$. The interpolated strain then simply becomes $\mathbf{E}(\mathbf{w}) = \sum_i^n w_i\mathbf{E}_i$ with $\mathbf{w} = (w_1, \ldots, w_n)^T$. By using these definitions in (2) we obtain the *example manifold* $\mathcal{E} \subset \mathcal{F}$ of realizable strains.

3.2 Example Projection

Our ultimate goal is to formulate a force that attracts the current configuration of the solid toward the example manifold, equivalently, attracting $\mathbf{x}$ toward its projection $\mathbf{x_w}$ on $\mathcal{E}$ (see Fig. 4).

Formulating the projection requires a suitable distance measure. Inspired by Chao et al. [2010] and Wirth et al. [2010], we approximate the geodesic distance on $\mathcal{F}$ between two shapes [Kilian et al. 2007] using the elastic potential $W(\mathbf{X}, \mathbf{x})$. Many reasonable choices for W are compatible with our approach; for concreteness we choose the potential arising from the energy density

$$W(\mathbf{X}, \mathbf{x}) = \mu|\mathbf{E}(\mathbf{X}, \mathbf{x})|_F^2 + \frac{\lambda}{2}\left(\frac{V(\mathbf{x})}{V(\mathbf{X})} - 1\right)^2 , \tag{3}$$

where λ and μ are material coefficients and $V(\cdot)$ measures the volume of a given configuration as the sum of elemental volumes. This simple extension of the St. Venant-Kirchhoff material [Bonet and Wood 1997] replaces the usual second term with one that allows the simulation to recover from inversions [Picinbono et al. 2003].

The projection $\mathbf{x} \mapsto \mathbf{x_w}$ corresponds to the minimization

$$\min_{\mathbf{x_w}} W(\mathbf{x_w}, \mathbf{x}) \quad s.t. \quad \mathbf{x_w} \in \mathcal{E}$$

or equivalently, invoking the extremizing conditions of (2),

$$\min_{\mathbf{x_w}, \mathbf{w}} W(\mathbf{x_w}, \mathbf{x}) \quad s.t. \quad \nabla_{\mathbf{x_w}} W_I(\mathbf{x_w}, \mathbf{w}) = 0. \tag{4}$$

The resulting constrained optimization problem is nonlinear both in the objective function and the constraints. Applying the penalty method for constraint enforcement, we minimize

$$W_p(\mathbf{x_w}, \mathbf{w}, \mathbf{x}) = W(\mathbf{x_w}, \mathbf{x}) + \gamma \left|\nabla_{\mathbf{x_w}} W_I(\mathbf{x_w}, \mathbf{w})\right|^2 \tag{5}$$

with respect to $\mathbf{x_w}$ and $\mathbf{w}$, for a sufficiently large fixed penalty stiffness γ. The weights $\mathbf{w}$ are constrained such that $w_i \geq 0$ and $\sum_i w_i = 1$. These constraints restrict the example space to interpolations of the provided examples. Since small extrapolations are in general not harmful, we enforce these constraints also weakly by adding simple quadratic energies to (5).

Observe that the first term of (4) involves the optimization of the elastic energy with respect to the undeformed configuration. This formulation is reminiscent of problems found in, e.g., variational shape optimization and mesh adaptation [Thoutireddy and Ortiz 2004]. Although the derivations do not pose any particular problems, these so-called *configurational forces* are to our knowledge new to graphics, which is why we list the required gradients and Hessians in Appendix A.

3.3 Example-based Simulation

With the definition of the example manifold in place and the projection procedure defined, we are all set for proceeding to the actual example-based simulation. As mentioned before, our system integrates readily with existing solid simulators, but is particularly convenient to build on top of a finite element solver, allowing the reuse of code for deformation measure and elastic potentials.

Variational Statics Assume that the conventional potential is given by $W_c(\mathbf{x}) = W(\mathbf{X}, \mathbf{x})$. In order to solve for static equilibrium we require that the sum of all external forces $\mathbf{f}_{\text{ext}}$ equal the internal forces induced by the new, augmented material with potential $W_c(\mathbf{x}) + W_p(\mathbf{x_w}, \mathbf{x})$. This yields the following system of equations

$$\nabla_{\mathbf{x}} W_c(\mathbf{x}) + \nabla_{\mathbf{x}} W_p(\mathbf{x_w}, \mathbf{w}, \mathbf{x}) = \mathbf{f}_{\text{ext}}. \tag{6}$$

In this expression we implicitly assume that the projection $\mathbf{x_w}$ and weight vector $\mathbf{w}$ are always the corresponding minimizer of (4). In order to handle these two minimizations in the same framework, we can recall that equation (6) is actually the necessary condition for a minimizer of the joint total energy

$$W_{\text{tot}}(\mathbf{x_w}, \mathbf{w}, \mathbf{x}) = W_c(\mathbf{x}) + W_p(\mathbf{x_w}, \mathbf{w}, \mathbf{x}) + W_{\text{ext}}(\mathbf{x}), \tag{7}$$

where we assumed that the external forces are conservative. By minimizing W_{tot} simultaneously in $\mathbf{x_w}$, $\mathbf{w}$ and $\mathbf{x}$, we solve both problems at the same time: The static solution is found with the correct projection for the example potential.

The two elastic potentials in (7) can be chosen independently, but in our implementation we use (3) for both of them. In order to further reduce the number of variables, we set the material constants of the example potential to scalar multiples of those of the conventional potential, leaving a single parameter to set. Fig. 5 illustrates the impact of this parameter for a range of practical values.

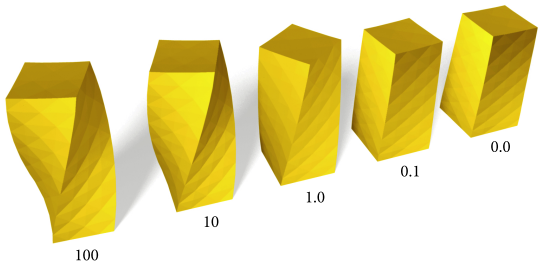

Figure 5: *Impact of scaling the material stiffness of the example potential relative to the conventional potential's stiffness. An elastic bar is subjected to gravity and simulated with different scaling coefficients (as indicated).*

Variational Implicit Euler For dynamic simulation, we opt for the implicit Euler method and pursue a similar strategy as for statics, i.e., we formulate the time stepping scheme as an optimization problem. We start from the canonical equations of motion

$$\mathbf{M}\ddot{\mathbf{x}} + \nabla_{\mathbf{x}} W_c(\mathbf{x}) + \nabla_{\mathbf{x}} W_p(\mathbf{x_w}, \mathbf{w}, \mathbf{x}) = \mathbf{f}_{\text{ext}} \tag{8}$$

where $\mathbf{M}$ is the mass matrix. In using our augmented elastic potential, we again assume that $\mathbf{x_w}$ and $\mathbf{w}$ are minimizers of (4). In order to arrive at a single optimization problem we first apply the implicit Euler integration scheme to obtain a nonlinear system of equations

$$\mathbf{M}\left(\frac{\mathbf{x}_n - \mathbf{x}_o}{h^2} - \frac{\mathbf{v}_o}{h}\right) + \nabla_{\mathbf{x}} W_c(\mathbf{x}_n) + \nabla_{\mathbf{x}} W_p(\mathbf{x_w}, \mathbf{w}, \mathbf{x}_n) = \mathbf{f}_{\text{ext}},$$

where h is the step size, $\mathbf{x}_n$ are new positions and $\mathbf{x}_o$ and $\mathbf{v}_o$ the old positions and velocities, respectively. We can solve this system in an elegant way by minimizing the objective function

$$H(\mathbf{x}_n, \mathbf{x_w}, \mathbf{w}) = \frac{h^2}{2}\left(\frac{\mathbf{x}_n - \mathbf{x}_o}{h^2} - \frac{\mathbf{v}_o}{h}\right)^T \mathbf{M}\left(\frac{\mathbf{x}_n - \mathbf{x}_o}{h^2} - \frac{\mathbf{v}_o}{h}\right) + W_c(\mathbf{x}_n) + W_p(\mathbf{x_w}, \mathbf{w}, \mathbf{x}_n)\,. \tag{9}$$

By optimizing (9) for $\mathbf{x}_n$, $\mathbf{x_w}$, and $\mathbf{w}$, we simultaneously solve the coupled problems of projection and time stepping.

Numerical Optimization The discussed optimization problems for the static and dynamic case are solved robustly by a classic Newton-Raphson procedure. In order to improve convergence, we employ a line search scheme and additionally apply diagonal Hessian correction in case of indefinite matrices [Nocedal and Wright 2000]. The dimension of the resulting linear systems is roughly twice that of conventional simulations. We employ a sparse direct Cholesky solver for solving the resulting symmetric positive definite systems [Schenk and Gärtner 2002].

4 Example Design & Implementation

Our method relies on example poses to model the characteristic deformation behavior of elastic solids. This section describes how to design *good* examples, how to efficiently approximate geometrically and mechanically complex models, and how to extend the range of effects by controlling the influence region of examples.

4.1 Example Design

Our method accepts example poses in the form of deformed element meshes that have the same topology as the undeformed mesh.

There are no specific requirements on the way in which these examples are created and, for instance, any geometric modeling tool [Gain and Bechmann 2008] can be used for this purpose. An alternative way of designing natural examples is by using a static solver built on the same elastic potential W_c that is also used in simulation [Barbič et al. 2009; Mezger et al. 2008]. While the geometric modeling approach imposes virtually no restrictions on creativity, this physics-based metaphor has the advantage that deformations propagate naturally (unless enforced otherwise) and that the resulting meshes are unlikely to exhibit severely distorted or inverted elements.

Apart from the technique used for generating examples, another important question is what kind of examples should be used. In order to represent natural transitions to the rest pose, we use the undeformed configuration as part of the example set in all our experiments. Any additional example should, first and foremost, represent characteristic or extreme poses. However, examples should also be sufficiently different in order to span a diverse space of expressive deformations with as few poses as possible.

4.2 Embedding Triangle Meshes

Embedding denotes a class of techniques for deforming highly-detailed geometry in accordance to the deformations of a (potentially) much coarser approximation. On the practical side, embedding is a very efficient way for increasing the level of detail to impressive amounts, as recently demonstrated by the work of Wojtan et al. [2009].

Embedded meshes can augment coarse volumetric simulations with high-quality surface details. But since the deformations of the embedded surface follow the coarse physics of the embedding mesh, the physical detail can generally not be increased — or only to limited amounts. Our example-based approach does, however, not impose as strict restrictions as the usual fine-to-coarse coupling. Using a static solver, it is quite easy to deform an embedding mesh such that its enveloped surface assumes a desired shape. We can thus design examples that account for realistic deformations of subelement geometry (see Fig. 1), but we can also create examples that emulate the deformation behavior of more complex mechanics, such as the buckling patterns of thin-walled cylinders (see Fig. 11).

Apart from manually designing example poses, one could also use an approach similar to Barbič et al. [2009] in order to automatically compute the embedding mesh that best approximates a given input surface. Using this technique, one would first run an offline simulation of a high-resolution volumetric model and then automatically deform the embedding mesh to match the surfaces of some characteristic frames. This approach could also used to generate embedding meshes for thin shell or rod simulations.

4.3 Local and Global Examples

We have until now assumed that examples are defined on the entire domain of the solid. For such *global* examples, the deformation of one part of the object directly influences all other parts as shown in Fig. 9. However, it is also interesting to limit the influence of an example to individual parts of an object. Such *local* examples can be used to define deformation behavior locally and independent of other regions. Additionally, different local examples can be combined to yield even more complex global behavior as shown, e.g., in Fig. 8. Vice versa, by dividing a complex pose into local examples, we can already specify much of an object's characteristic behavior using only a single deformed configuration.

On a technical note, we constrain the example space to convex combinations of the individual poses in the case of global examples, which is necessary in order to obtain well-behaved strain interpolation. While it may appear tempting to allow extrapolation, doing so entails the risk of running into invalid strains: It is a simple matter to determine weights such that the extrapolation of two valid strain tensors yields a metric tensor with negative values on its diagonal — but this is not meaningful since $\mathbf{C}_{ii} = \sum_k \mathbf{F}_{ki}\mathbf{F}_{ki}$ is always positive.

For local examples, however, the convexity constraint can be relaxed: we can simply form groups of *interacting* examples such that any two poses from different groups have no deformed element in common. In this way, we can safely enforce the convexity constraints on each group in isolation. As a practical implication, doing so allows individual parts of an object to deform independently of other parts and enlarges the space of prefered shapes in an efficient manner.

5 Results

This section presents a set of examples that illustrate different aspects of our approach and demonstrate typical applications.

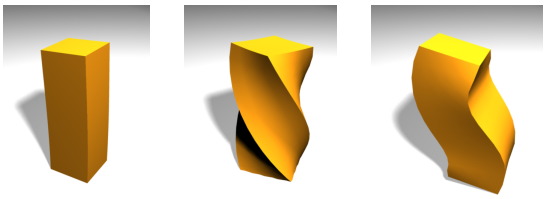

Figure 6: *An elastic cuboid deforms under gravity using no example, a twist example and an S-shaped example.*

Our method allows an animator to design and simulate complex elastic materials by merely providing a set of example poses that correspond to characteristic, desirable, or extreme deformations. Fig. 6 shows a simple example that illustrates this idea: by augmenting an elastic bar with an example potential constructed from twist or *S*-shaped poses, we can significantly change its deformation behavior and thus imprint different styles onto the animation. Though possible in theory, achieving the same results with a conventional simulator would require tedious tuning of an inhomogeneous, anisotropic, and probably nonlinear material. By contrast, our approach is intuitive and output-oriented, making it well-suited to design processes commonly used for creative applications.

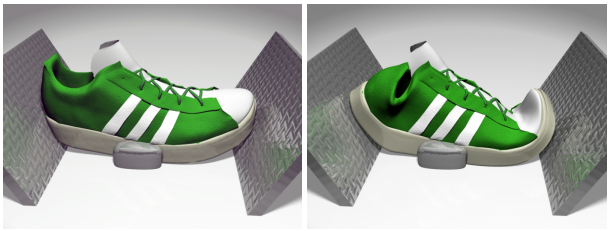

Figure 7: *Compressed sneaker simulated as a coarse solid. Without examples (left) and augmented with two local examples (right).*

Global examples are used to directly specify prefered deformations for an entire object, which can be understood as a *what-you-see-is-what-you-get* approach to material design. However, there are also many common objects for which the characteristic deformations are rather local than global. Moreover, different local deformations can typically occur simultaneously and independently of each other. This kind of behavior is illustrated in the animation shown in Fig.

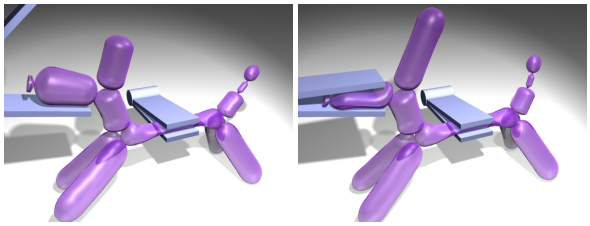

Figure 8: *Local examples defined over unconnected regions showcased on a balloon dog.*

7, for which two characteristic deformations of a shoe, namely the buckling of its toe and the bulging of its heel, are provided as local examples to the simulation.

This animation also showcases the application of embedding: the high-resolution geometry of the shoe deforms in accordance to the coarse embedding mesh — but it does so in a very plausible way. This, in turn, is due to the fact that the volumetric example meshes were generated such that the embedded mesh assumes the desired deformations, irrespective of the actual shape of the embedding mesh.

Local examples do not necessarily have to be defined over connected components, but can also couple remote regions while still affecting only a small part of the entire object. An example of this application can be seen in Fig. 8, which shows that, in an artistic setting, compressing the nose of a balloon dog can lead to an inflation of its ears.

Our method can be used to design deformation styles which are difficult to generate with conventional elastic materials, but are still within the range of what we might expect from some exotic material. We can, however, also go a step beyond and use examples to induce deformations that clearly exceed the realm of conventional elastic materials. An example that goes along these lines can be seen in Fig. 9, which shows a gummy bear that, despite its jelly-like appearance, seems to have a personality of its own, driving it to deform in very peculiar ways in response to user interaction.

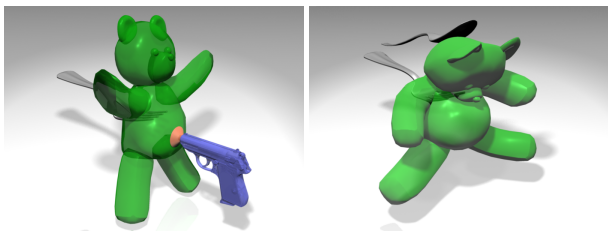

Figure 9: *A gummy bear is equipped with expressive examples to create an impression of personality.*

Similar in spirit is the example shown in Fig. 1, which depicts the unfortunate incident of a toy car hitting the wall beneath a fake tunnel. Using example-based simulation, we can make the car react to the impact in very diverse ways, following the exaggerated-physics style frequently found in cartoons.

Our method primarily aims at volumetric solids, but thanks to the embedding technique it can also be used to mimic the behavior of more complex mechanical models such as thin shells. The first two images in Fig. 11 show our approach applied to a cylindrical shell which, when compressed, exhibits the typical diamond-shaped buckling patterns. Note that the example was not computed with a thin shell simulation code but designed with a static version of our solid simulator. The reason why the buckling patterns can still appear in a plausible way is simply that the example potential

Figure 10: *A car with four example poses, each of which is activated during the animation in response to different impact events.*

renders these deformations energetically favorable. As shown in the remaining images of Fig. 11, we can again specify various deformation examples to obtain diverse material effects, ranging from physically plausible deformations to art-directed physical animation.

Performance We provide timings for all examples presented in this section in Tab. 1. All simulations were performed with a manifold constraint penalty of 10^4, a convex weight penalty of 100 and a timestep size of 0.02s.

Model	#DOFs	t_{asm}	t_{newton}	t_{tot}	α
Cuboid Twist	975	80	141 / 308	528 / 3064	40
Sneaker	942	107	159 / 173	680 / 1288	20
Teddy	828	59	61 / 65	1333 / 1410	40
Cylinder	227	43	57 / 328	502 / 1214	20
Car	1410	110	192 / 204	2990 / 3292	50
Balloon	1320	106	99 / 195	125 / 2196	1000

Table 1: *Timings (in ms) for single gradient/Hessian assembly (t_{asm}) and Newton step with line search (t_{newton}, min/max time), as well as min/max total time per timestep (t_{tot}), taken on a single core of a Intel Core i7 960, 3.2 GHz. α denotes the stiffness ratio between manifold and conventional potential.*

It should be noted that the benefits of example-based elastic materials come at the price of additional computation costs. As can be seen from Tab. 1, the largest fraction of the time is spent on the assembly of the linear system, including the computation of gradients and Hessians, and its solution. Currently, the performance of our method does not allow its use in interactive applications such as video games. In the context of artistic material design, however, the additional costs of our approach seem acceptable as they are very likely to pay off in terms of time saved on tuning material parameters of conventional material models.

	1 ex.	2 ex.	3 ex.	4 ex.	8 ex.	12 ex.
t_{asm}	120	124	128	132	148	165
t_{slv}	244	216	213	218	220	232

Table 2: *Performance scaling for multiple examples (top row) illustrated on the cuboid animation (Fig. 6). Average timings (in ms) for assembling (t_{asm}) and solving (t_{slv}) the linear system in a single Newton step.*

As another performance indicator, we also investigated the scaling of our method with respect to the number of example poses. Our method can faithfully handle the case of multiple examples as ex-

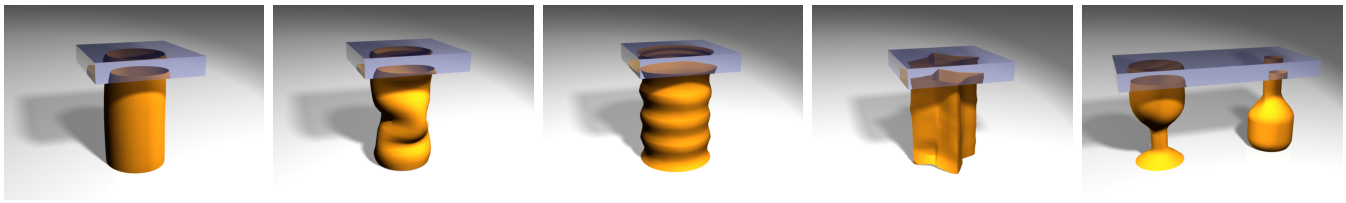

Figure 11: *A cylindrical surface mesh embedded in a volumetric simulation mesh. Using a buckled example pose allows us to emulate thin shell behavior. By varying the input examples we can effectively control the deformation behavior of the embedded geometry.*

emplified, e.g., in the animations of the car (Fig. 10), the balloon-dog, or the pendulum (see accompanying video). For a quantitative analysis, we measured computation times for an increasing number of example poses on the cuboid animation. The results shown in Tab. 2 indicate that the number of examples is not a limiting factor of our method: Using twelve poses instead of one, the time spent on solving the nonlinear system increases by only 15%.

6 Discussion

We have only scratched the surface of what is becoming possible with example-based simulation. We see many promising directions which we would like to explore in the future. While we framed our approach in the context of deformable solids, we believe that the underlying theory can be generalized to different elastic models, such as shells and rods. This could, for example, be interesting in the contexts of cloth and hair simulation. Furthermore, we have not used the information obtained through the projection on the example manifold — an exciting application could be to couple the example weights to secondary effects, e.g., for modulating texture or surface geometry. There is also much room for further exploring the definition of the shape space, which currently only considers position information via the Green strain. It might be interesting to also account for velocities (rate of strain) or even forces. Another idea would be to directly encode different aspects of deformation (such as incompressibility) into the definition of the shape space and, e.g., define an example manifold that *sees* only deviatoric (i.e., volume-preserving) deformations.

Our prototype implementation already indicates that our method has great potential in designing materials and art-directing simulations. However, we also see various possibilities for extensions and improvements. In particular the performance of our optimization scheme should be increased and we anticipate that Lagrangian methods will lead to better convergence than our current penalty approach. Furthermore, we currently rely on the user to create examples that are meaningful in that they do neither contradict each other nor strongly counteract the underlying elastic potential. It would be desirable to develop methods that assist the user in this process by providing appropriate feedback on the quality of examples. Another promising direction would be to automatically select a set of example poses from a given input animation.

Acknowledgments

We would like to thank Tobias Pfaff for his invaluable help in rendering results. We are also grateful to Gian-Marco Baschera, Bernd Bickel and Peter Kaufmann for many fruitful discussions.

Sebastian Martin is supported by the Swiss National Science Foundation (grant No. 200021-130596). Eitan Grinspun's laboratory is supported by the Sloan Foundation, NSF (CAREER Award CCF-0643268, IIS-0916129), Adobe, ATI, Autodesk, Intel, mental images, Microsoft Research, NVIDIA, Side Effects Software, the Walt Disney Company, and Weta Digital.

References

ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-Rigid-As-Possible Shape Interpolation. In *Proc. of ACM SIGGRAPH '00*, 157–164.

BARAN, I., VLASIC, D., GRINSPUN, E., AND POPOVIĆ, J. 2009. Semantic deformation transfer. In *Proc. of ACM SIGGRAPH '09*, 36:1–36:6.

BARBIČ, J., AND POPOVIĆ, J. Real-time control of physically based simulations using gentle forces. *Proc. of ACM SIGGRAPH Asia '08*, 163:1–163:10.

BARBIČ, J., DA SILVA, M., AND POPOVIĆ, J. 2009. Deformable object animation using reduced optimal control. In *Proc. of ACM SIGGRAPH '09*, 53:1–53:9.

BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. In *Proc. of ACM SIGGRAPH '07*, 16:1–16:8.

BERGOU, M., MATHUR, S., WARDETZKY, M., AND GRINSPUN, E. 2007. Tracks: toward directable thin shells. In *Proc. of ACM SIGGRAPH '07*, 50:1–50:10.

BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. In *Proc. of ACM SIGGRAPH '09*, 89:1–89:9.

BONET, J., AND WOOD, R. D. 1997. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge Univ. Press.

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. PriMo: coupled prisms for intuitive surface modeling. In *Proc. of Symp. on Geometry Processing '06*, 11–20.

CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. In *Proc. of ACM SIGGRAPH '10*, 38:1–38:6.

GAIN, J., AND BECHMANN, D. 2008. A survey of spatial deformation from a user-centered perspective. *ACM Trans. Graph. 27*, 107:1–107:21.

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proc. of Symp. on Computer Animation (SCA '03)*, 62–67.

IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. In *Proc. of Symp. on Computer Animation (SCA '04)*, 131–140.

KILIAN, M., MITRA, N. J., AND POTTMANN, H. 2007. Geometric modeling in shape space. In *Proc. of ACM SIGGRAPH '07*, 64:1–64:8.

KIRCHER, S., AND GARLAND, M. 2006. Editing arbitrarily deforming surface animations. In *Proc. of ACM SIGGRAPH '06*, 1098–1107.

KONDO, R., KANAI, T., AND ANJYO, K.-I. 2005. Directable animation of elastic objects. In *Proc. of Symp. on Computer Animation (SCA '05)*, 127–134.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. In *Proc. of ACM SIGGRAPH '10*, 32:1–32:6.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. In *Proc. of ACM SIGGRAPH '05*, 479–487.

MCNAMARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid control using the adjoint method. In *Proc. of ACM SIGGRAPH '04*, 449–456.

MEZGER, J., THOMASZEWSKI, B., PABST, S., AND STRASSER, W. 2008. Interactive physically-based shape editing. In *Proc. of ACM Symp. on Solid and Physical Modeling*, 79–89.

NOCEDAL, J., AND WRIGHT, S. J. 2000. *Numerical Optimization*. Springer.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proc. of ACM SIGGRAPH '99*, 137–146.

PICINBONO, G., DELINGETTE, H., AND AYACHE, N. 2003. Non-linear anisotropic elasticity for real-time surgery simulation. *Graph. Models 65*, 305–321.

POPOVIĆ, J., SEITZ, S. M., ERDMANN, M., POPOVIĆ, Z., AND WITKIN, A. 2000. Interactive manipulation of rigid body simulations. In *Proc. of ACM SIGGRAPH '00*, 209–217.

SCHENK, O., AND GÄRTNER, K. 2002. Solving unsymmetric sparse systems of linear equations with pardiso. In *Proc. ICCS '02*, 355–363.

SHEFFER, A., AND KRAEVOY, V. 2004. Pyramid Coordinates for Morphing and Deformation. In *Proc. 3D Data Processing, Visualization, and Transmission*, 68–75.

SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., ROSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Symp. Geometry processing (SGP '04)*, 179–188.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. In *Proc. of ACM SIGGRAPH '04*, 399–405.

SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. In *Proc. of ACM SIGGRAPH '05*, 488–495.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *Proc. of ACM SIGGRAPH '88*, 269–278.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *Proc. of ACM SIGGRAPH '87*, 205–214.

THOUTIREDDY, P., AND ORTIZ, M. 2004. A variational r-adaption and shape-optimization method for finite-deformation elasticity. *Int. J. Numer. Meth. Engng. 61*, 1–21.

THÜREY, N., KEISER, R., PAULY, M., AND RÜDE, U. 2006. Detail-preserving fluid control. In *Proc. of Symp. on Computer Animation (SCA '06)*, 7–12.

TWIGG, C. D., AND JAMES, D. L. 2007. Many-worlds browsing for control of multibody dynamics. In *Proc. of ACM SIGGRAPH '07*, 14:1–14:8.

WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. 2009. State of the art in example-based texture synthesis. In *Proc. of Eurographics '09, State of the Art Report*.

WINKLER, T., DRIESEBERG, J., ALEXA, M., AND HORMANN, K. 2010. Multi-scale geometry interpolation. In *Proc. of Eurographics '10*, 309–318.

WIRTH, B., BAR, L., RUMPF, M., AND SAPIRO, G. 2010. A continuum mechanical approach to geodesics in shape space. *Int. J. of Computer Vision*, 1–26.

WITKIN, A., AND KASS, M. 1988. Spacetime constraints. In *Proc. of ACM SIGGRAPH '88*, 159–168.

WOJTAN, C., MUCHA, P. J., AND TURK, G. 2006. Keyframe control of complex particle systems using the adjoint method. In *Proc. of Symp. on Computer Animation (SCA '06)*, 15–23.

WOJTAN, C., THÜREY, N., GROSS, M., AND TURK, G. 2009. Deforming meshes that split and merge. In *Proc. of ACM SIGGRAPH '09*, 76:1–76:10.

A Gradients and Hessians

In order to perform Newton optimization of the energies (7) and (9) we need first and second derivatives of (2) and (3). While derivatives of the second term (volume-change) in (3) are quite simple to derive [Picinbono et al. 2003], the derivatives of the first term (Green strain) are more involved and given here. We resort to index notation in order to write the element-wise derivatives in compact form. Let $\mathbf{X}_{mn}$ and $\mathbf{x}_{mn}$ denote the n-th components of position vectors for node m in the undeformed and deformed configuration, respectively. Introducing the matrix $\mathbf{S} = [[-1,1,0,0]^T,[-1,0,1,0]^T,[-1,0,0,1]^T]$ and defining $\mathbf{d}_{ij} = \mathbf{x}_{ki}\mathbf{S}_{kj}$ and $\mathbf{D}_{ij} = \mathbf{X}_{ki}\mathbf{S}_{kj}$, the deformation gradient becomes $\mathbf{F}_{ij} = \mathbf{d}_{ik}\mathbf{D}^{-1}_{kj}$. The deformed first derivative of $\mathrm{tr}(\mathbf{E}^T\mathbf{E}) = \mathbf{E}_{ij}\mathbf{E}_{ij}$ can then be stated compactly as

$$\frac{\partial(\mathbf{E}_{ij}\mathbf{E}_{ij})}{\partial \mathbf{x}_{mn}} = 2(\mathbf{S}\mathbf{D}^{-1}\mathbf{E}\mathbf{F}^T)_{mn},$$

and the second derivative is obtained as

$$\begin{aligned}\frac{\partial^2(\mathbf{E}_{ij}\mathbf{E}_{ij})}{\partial \mathbf{x}_{mn}\partial \mathbf{x}_{st}} &= (\mathbf{F}\mathbf{F}^T)_{nt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{D}^{-T}\mathbf{S}^T)_{ms}\\ &+ (\mathbf{S}\mathbf{D}^{-1}\mathbf{F}^T)_{mt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{F}^T)_{sn}\\ &+ 2\delta_{nt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{E}\mathbf{D}^{-T}\mathbf{S}^T)_{ms}.\end{aligned}$$

The derivatives with respect to the undeformed configuration assume a similar form: the First derivatives are

$$\frac{\partial(\mathbf{E}_{ij}\mathbf{E}_{ij})}{\partial \mathbf{X}_{mn}} = -2(\mathbf{S}\mathbf{D}^{-1}\mathbf{E}\mathbf{C})_{mn},$$

while second derivatives follow as

$$\begin{aligned}\frac{\partial^2(\mathbf{E}_{ij}\mathbf{E}_{ij})}{\partial \mathbf{X}_{mn}\partial \mathbf{X}_{st}} &= (\mathbf{C}\mathbf{C})_{nt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{D}^{-T}\mathbf{S}^T)_{ms}\\ &+ (\mathbf{S}\mathbf{D}^{-1}\mathbf{C})_{mt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{C})_{sn}\\ &+ 2\mathbf{C}_{nt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{E}\mathbf{D}^{-T}\mathbf{S}^T)_{ms}\\ &+ 2(\mathbf{S}\mathbf{D}^{-1})_{sn}(\mathbf{S}\mathbf{D}^{-1}\mathbf{E}\mathbf{C})_{mt}\\ &+ 2(\mathbf{S}\mathbf{D}^{-1})_{mt}(\mathbf{S}\mathbf{D}^{-1}\mathbf{E}\mathbf{C})_{sn}.\end{aligned}$$