Real-Time Example-Based Elastic Deformation

Yuki Koyama[1]   Kenshi Takayama[1,2]   Nobuyuki Umetani[1]   Takeo Igarashi[1,3]

[1]The University of Tokyo, Japan   [2]ETH Zürich, Switzerland   [3]JST ERATO, Japan
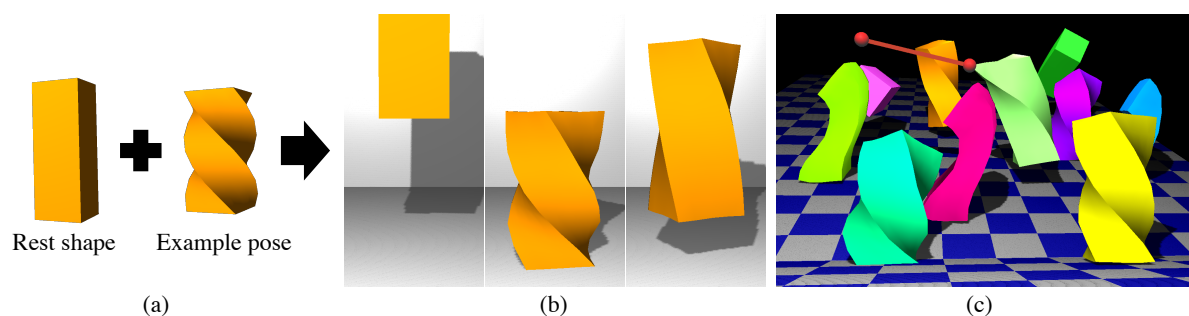

Figure 1: (a) *User-specified input data comprising the rest shape of the model and its preferred deformed pose(s).* (b) *A result animation of the example-based elastic material generated from the input data.* (c) *A slope scene containing two types of example-based elastic cuboids (generated from twist-shaped and S-shaped example poses) represented by 2250 particles in total; this scene is simulated at 49 frames per second on a desktop with Core i7 2.67GHz CPU and Radeon HD 6630M.*

Abstract

We present an example-based elastic deformation method that runs in real time. Example-based elastic deformation was originally presented by Martin et al. [MTGG11], where an artist can intuitively control elastic material behaviors by simply giving example poses. Their FEM-based approach is, however, computationally expensive requiring nonlinear optimization, which hinders its use in real-time applications such as games. Our contribution is to formulate an analogous concept using the shape matching framework, which is fast, robust, and easy to implement. The key observation is that each overlapping local region's right stretch tensor obtained by polar decomposition is a natural choice for a deformation descriptor. This descriptor allows us to represent the pose space as a linear blending of examples. At each time step, the current deformation descriptor is linearly projected onto the example manifold, and then used to modify the rest shape of each local region when computing goal positions. Our approach is two orders of magnitude faster than Martin et al.'s approach while producing comparable example-based elastic deformations.

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

1. Introduction

Achieving intuitive artist control in physically based animations has been a longstanding goal in computer graphics. Many approaches have been proposed for achieving various phenomena such as fluids [TKPR06], rigid bodies [TJ08], cloths [BMWG07], and deformable solids [KKA05].

Recently, Martin et al. [MTGG11] proposed an approach of example-based elastic materials, where an artist can con-

trol the elastic behavior of a deformable object by simply giving examples of poses, while other data-driven elastic simulations [SZT*08,JF03] mainly focus on physical plausibility rather than artistic control. Martin et al.'s system computes an additional force that pulls the current pose toward the space of preferable poses defined by examples (Fig. 1 (a), (b)). Compared to the previous techniques for artistic control of physically-based animations that are constrained by some fixed keyframe or trajectory, example-based elastic materi-

als can be used in a free environment without a predefined scenario or an animation trajectory. Martin et al.'s approach is more desirable for artist control than the approach of setting actual physical material parameters [BBO*09], because the artist only needs to specify examples of desired poses.

Their FEM-based approach is, however, computationally demanding which limits its use in offline simulations only. This is unfortunate, as artistic expressions are of high importance also for many real-time applications such as interactive virtual environments. The fundamental cause of the high computational cost of Martin et al.'s approach is that they use an additional elastic potential to attract the current deformation toward the space of preferable deformations defined by example poses. This approach requires a projection of the current deformation to the space of preferable deformations, called an *example manifold*. Since this manifold is nonlinear with respect to their deformation descriptor, the projection is done by nonlinear optimization which is the bottleneck of their approach.

We achieve example-based elastic deformations in real time by basing our approach on shape matching [MHTG05]. With shape matching, we can attract the current deformation toward the space of preferable deformations without relying on an elastic potential; instead, we can simply modify the rest shape of each overlapping local region. Since the example manifold is linear in our approach, we only need to perform a linear projection at each time step, achieving real-time performance. Our approach is two orders of magnitude faster than Martin et al.'s approach while producing comparable example-based elastic deformations.

Similar to Martin et al., we can extend our approach to allow local examples by dividing the entire set of local regions into separate groups. In addition, by regarding each local region as a separate group (i.e., defining an independent example manifold for each local region), we can cause the effect of example-based elastic deformations to appear non-uniformly according to the stress applied to each local region. Because shape matching is a purely geometric technique, our approach does not provide as much physical accuracy as Martin et al.'s approach; however, we believe that physical accuracy would be of secondary importance in the context of art-directed animations.

1.1. Related Work

Shape matching [MHTG05] inspired a number of subsequent extensions. FastLSM [RJ07] achieves large deformations at fast rates by defining particles on a cubic lattice and performing shape matching on many overlapping local regions using a fast summation operator. Steinemann et al. [SOG08] extended this approach to achieve levels of details for deformations using an octree, while Rungjiratananon et al. [RKN10] modified FastLSM to simulate complex hairstyles. Müller and Chentanez [MC11] proposed
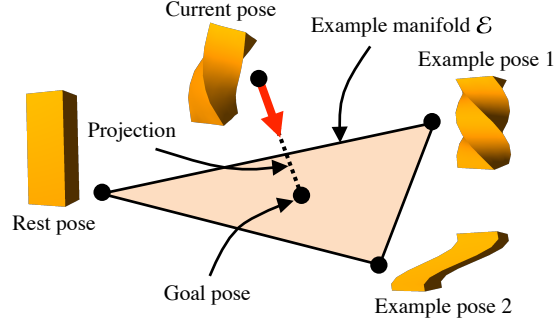

Figure 2: *Overview of run-time operations. Here, the elastic cuboid model has its rest pose, current pose, and two example poses. In each time step, the current pose is projected onto the example manifold $\mathcal{E}$; then, the projected point is assumed to be the goal pose, and the current pose is pulled toward it.*

oriented particles to achieve stability with much fewer particles by augmenting each particle with orientation information. Diziol et al. [DBB11] took a different approach, where surface meshes are used for simulation. To our knowledge, we are the first that extend shape matching to example-based elastic deformations. Note that our approach is orthogonal to these previous extensions and thus can benefit from them.

Ijiri et al. [ITYI09] used shape matching to procedurally generate active deformations of unarticulated objects such as jellyfish by expanding and contracting local regions. They actively change the rest shapes of local regions to generate a motion while we passively change rest shapes responding to deformations caused by external forces.

2. Overview of Run-time Operations

Before describing our method in detail, we first provide an overview of run-time operations. In the proposed method, the concept of example-based elastic materials is realized by processing the following two operations in each time step:

1. Linearly projecting the current deformed pose toward the example manifold $\mathcal{E}$ to obtain a goal pose. (§4.2)
2. Pulling the current deformed pose toward the goal pose. (§3)

The key difference between Martin et al.'s and our method is that we define the example manifold $\mathcal{E}$ linearly; that is, we define it as a convex hull of the linear blending of the rest and example poses (§4.1). Thus, the projection can be efficiently computed without any nonlinear optimizations. Fig. 2 presents a schematic representation of these operations.

3. Modification in Shape Matching Dynamics

We closely follow the approach of convolving shape matching operators on overlapping local regions to drive deforma-

tions [RJ07, ITYI09]. The input to our system consists of a volumetric tetrahedral mesh (rest shape) and n example poses. A triangular surface mesh embedded into the tetrahedral mesh is used to render the model with detailed geometry. Each tetrahedral mesh vertex is treated as a particle, and a local region N_i is defined for each particle i by collecting the particle itself and its 1-ring neighbors [ITYI09]. Note that our approach does not assume any specific way of constructing particles and local regions, and works well with other extensions of shape matching such as FastLSM [RJ07] and oriented particles [MC11].

The rest position, current position, and mass of particle i are denoted as $\mathbf{x}_i^0$, $\mathbf{x}_i$, and m_i, respectively. The system first performs shape matching on each local region r independently. That is, a moment matrix is computed as

$$\mathbf{A}_r = \left(\sum_{i \in N_r} \tilde{m}_i \mathbf{p}_i \mathbf{q}_i^T\right)\left(\sum_{i \in N_r} \tilde{m}_i \mathbf{q}_i \mathbf{q}_i^T\right)^{-1} \in \mathbb{R}^{3\times3} \qquad (1)$$

where $\tilde{m}_i = \frac{m_i}{|N_i|}$ is the effective mass, $\mathbf{c}_r^0 = \frac{\sum_{i \in N_r} \tilde{m}_i \mathbf{x}_i^0}{\sum_{i \in N_r} \tilde{m}_i}$ and $\mathbf{c}_r = \frac{\sum_{i \in N_r} \tilde{m}_i \mathbf{x}_i}{\sum_{i \in N_r} \tilde{m}_i}$ the centers of mass, and $\mathbf{p}_i = \mathbf{x}_i - \mathbf{c}_r$ and $\mathbf{q}_i = \mathbf{x}_i^0 - \mathbf{c}_r^0$ the relative positions with respect to the centers of mass. The moment matrix is then decomposed into its rotation part $\mathbf{R}_r$ and stretch/shear part $\mathbf{S}_r$ using polar decomposition as $\mathbf{S}_r = \sqrt{\mathbf{A}_r^T \mathbf{A}_r}$ and $\mathbf{R}_r = \mathbf{A}_r \mathbf{S}_r^{-1}$. The goal position of particle i with respect to local region r is computed as

$$\mathbf{g}_{r,i} = \mathbf{R}_r\left(\mathbf{x}_i^0 - \mathbf{c}_r^0\right) + \mathbf{c}_r \qquad (2)$$

in conventional shape matching, while we modify it as

$$\mathbf{g}_{r,i} = \mathbf{R}_r \tilde{\mathbf{S}}_r\left(\mathbf{x}_i^0 - \mathbf{c}_r^0\right) + \mathbf{c}_r \qquad (3)$$

to achieve example-based elastic deformations. Here, $\tilde{\mathbf{S}}_r$ is obtained by performing projection onto the example manifold as explained in the next section. This effectively deforms the rest shape of the local region r when computing goal positions (Fig. 3). The goal position $\mathbf{g}_i$ of particle i is computed as an average of goal positions estimated by its overlapping local regions as $\mathbf{g}_i = \langle\mathbf{g}_{r,i}\rangle_{r \in N_i}$. Finally, the position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$ are updated as

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + \alpha \frac{\mathbf{g}_i(t) - \mathbf{x}_i(t)}{h} + h\frac{\mathbf{f}_{\text{ext}}(t)}{m_i} \qquad (4)$$

$$\mathbf{x}_i(t+h) = \mathbf{x}_i(t) + h\mathbf{v}_i(t+h) \qquad (5)$$

where h is the time step, $\mathbf{f}_{\text{ext}}$ is the external force, and $\alpha \in [0\ldots1]$ is a parameter of stiffness.

4. Example-Based Elastic Deformation

Similar to Martin et al., we define a space of preferable deformations implied by example poses. We first define a descriptor of a deformation and an example manifold spanned by descriptors of the example deformations. We then explain how the current deformation descriptor is projected onto the example manifold. Finally, we show an extension
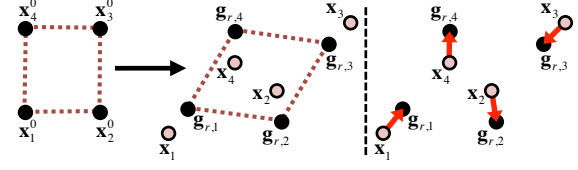

Figure 3: *In order to obtain the goal position $\mathbf{g}_{r,i}$ of particle i in local region r, the rest shape of the region is appropriately deformed by the strain $\tilde{\mathbf{S}}_r$ that is obtained by projection (§4.2) in our method, while the standard shape matching uses the undeformed rest shape as the goal shape.*
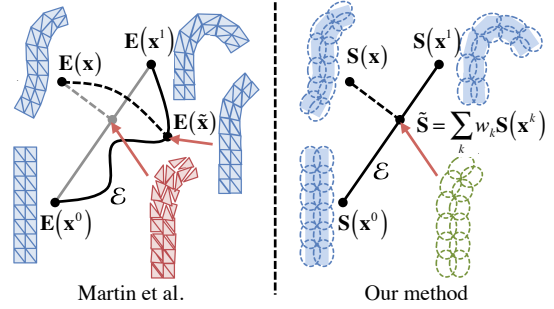

Figure 4: *Comparison between Martin et al.'s (left) and our (right) approach. In Martin et al.'s approach, example manifold $\mathcal{E}$ and the projection are nonlinearly defined. In contrast, in our approach, example manifold $\mathcal{E}$ can be defined as a simple linear interpolation, thus the projection can be handled easily.*

that achieves locally independent example-based elastic deformations by dividing the entire set of local regions into separate groups.

4.1. Deformation Descriptor and Example Manifold

We provide a brief review of Martin et al.'s approach for comparison with ours. Further details for Martin et al.'s approach can be found in [MTGG11]. In their approach, in order to compute an additional elastic potential, it is necessary to find the deformed pose from a space of preferable poses that is closest to the current pose. To measure the similarity between deformations, they define a unique descriptor of a deformation $\mathbf{x}$ by concatenating the Green strain tensor $\mathbf{E}_i \in \mathbb{R}^6$ of each tetrahedral element i as $\mathbf{E}(\mathbf{x})^T = \left(\mathbf{E}_1^T \cdots \mathbf{E}_m^T\right) \in \mathbb{R}^{6m}$ (3×3 symmetric matrices are represented as 6D vectors hereafter). An important fact is that the image of a map $\mathbf{x} \mapsto \mathbf{E}(\mathbf{x})$, called *realizable manifold* $\mathcal{F} \subset \mathbb{R}^{6m}$, is a nonlinear manifold which means that a linear combination of two descriptors $w_1\mathbf{E}(\mathbf{x}^1) + w_2\mathbf{E}(\mathbf{x}^2)$ is in general not *reconstructible*; i.e., it does not correspond to any deformed pose. Isolated tetrahedra can satisfy the prescribed strain but their assembly cannot (Fig. 4 left). A continuous space of preferable deformations, called an *example manifold* $\mathcal{E}$, is there-

fore defined by projecting the convex hull of example deformation descriptors $\left\{\sum_{k=0}^{n} w_k \mathbf{E}(\mathbf{x}^k)\right\}$ onto $\mathcal{F}$. At each time step, the current deformation is projected onto $\mathcal{E}$ using nonlinear optimization, which is the bottleneck of Martin et al.'s approach.

In our approach based on shape matching, the system pulls the current deformation toward the space of preferable deformations by modifying the rest shape of each local region (Fig. 3), as opposed to using an additional elastic potential in Martin et al.'s FEM-based approach. We observe that the moment matrix $\mathbf{A}_r$ in (1) can be regarded as an approximation of the deformation gradient of local region r as detailed in Appendix, and thus its strain component $\mathbf{S}_r$ as the right stretch tensor. Analogous to Martin et al., we define a deformation descriptor by concatenating $\mathbf{S}_r$ of each local region r as $\mathbf{S}(\mathbf{x})^T = \left(\mathbf{S}_1^T \cdots \mathbf{S}_m^T\right) \in \mathbb{R}^{6m}$. Unlike Martin et al., we can compute a valid deformation from an arbitrary deformation descriptor, because inconsistent goal positions estimated by overlapping local regions are blended by averaging. In other words, the issue of reconstructibility of deformation descriptors is nonexistent in our case, allowing us to define an example manifold as a convex hull of example deformation descriptors $\left\{\sum_{k=0}^{n} w_k \mathbf{S}(\mathbf{x}^k)\right\}$ (Fig. 4 right). We linearly project $\mathbf{S}$ onto the example manifold to yield $\tilde{\mathbf{S}}$ as explained below, and use it when computing goal positions in (3).

4.2. Projection to Example Manifold

Denoting the rest pose's descriptor (i.e., concatenation of identity matrices) and the k-th example pose's descriptor as $\mathbf{S}^0 = \mathbf{S}(\mathbf{x}^0)$ and $\mathbf{S}^k = \mathbf{S}(\mathbf{x}^k)$, respectively, our goal is to find a convex combination $\tilde{\mathbf{S}} = \sum_{k=0}^{n} w_k \mathbf{S}^k$ that is closest to the current pose descriptor $\mathbf{S} = \mathbf{S}(\mathbf{x})$. We first compute weights $w_1, \ldots, w_n$ by minimizing a quadratic energy

$$\left\| \sum_{k=1}^{n} w_k (\mathbf{S}_k - \mathbf{S}_0) - (\mathbf{S} - \mathbf{S}_0) \right\|^2 \qquad (6)$$

whose solution $\boldsymbol{w}^T = (w_1, \ldots, w_n)^T$ is obtained as

$$\boldsymbol{w} = \left(\mathbf{L}^T \mathbf{L}\right)^{-1} \mathbf{L}^T (\mathbf{S} - \mathbf{S}_0) \qquad (7)$$

where $\mathbf{L} = (\mathbf{S}_1 - \mathbf{S}_0 \cdots \mathbf{S}_n - \mathbf{S}_0) \in \mathbb{R}^{6m \times n}$ which is constant during simulation. We then obtain $w_0 = 1 - \sum_{k=1}^{n} w_k$.

This may yield negative weights that will result in extrapolation beyond example poses, which would be undesirable for artist control. We eliminate negative weights by iteratively performing the following simple procedure: the smallest negative weight is chosen and set to zero, and its absolute value is divided by n and subtracted from all the other weights.

Another concern is that the deformation may become stuck somewhere in the space of preferable poses instead of going back to the rest pose, because we change the rest shape
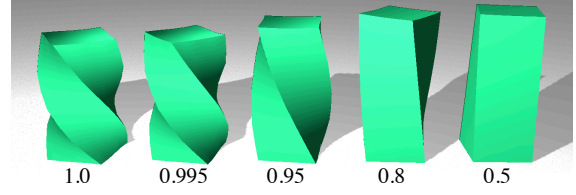

Figure 5: *Varying the magnitude of the example-based elastic deformation effect by changing β.*
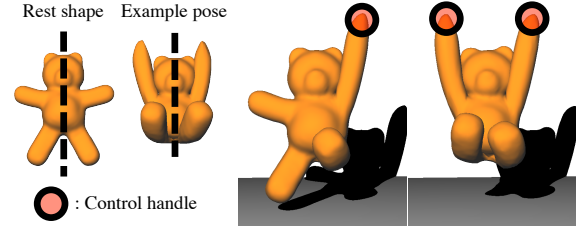

Figure 6: *Local regions of the teddy bear are divided into two separate groups at its middle. The teddy bear's left and right halves can be manipulated independently.*

of each local region when computing goal positions (Fig. 3). This is in contrast to Martin et al.'s approach where the conventional potential pulls the deformation back to the rest pose. To avoid this problem, we modify the weights such that the rest pose always gets a slightly higher weight than others. We introduce a parameter $\beta \in [0, 1)$ that modifies the weights as $w_0' = w_0 + (1-\beta) \sum_{k=1}^{n} w_k$ and $w_k' = \beta w_k$ ($k = 1, \ldots, n$). This modification has a large enough impact even when β is only slightly smaller than 1 (e.g., $\beta = 0.995$), and the effect of example-based elastic deformation decreases as β becomes smaller (Fig. 5).

4.3. Extension to Local Examples

We can achieve an effect of local examples similar to those demonstrated by Martin et al (Fig. 6). We divide the entire set of local regions into separate groups and define an example manifold for each group independently. Furthermore, by treating each local region as a separate group (i.e., each local region's example manifold lies in 6D space), we can achieve

Figure 7: *Each local region has its own example manifold independently. The difference from the case of using a globally defined example manifold is evident.*
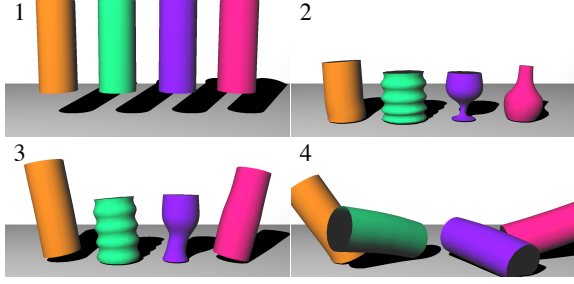
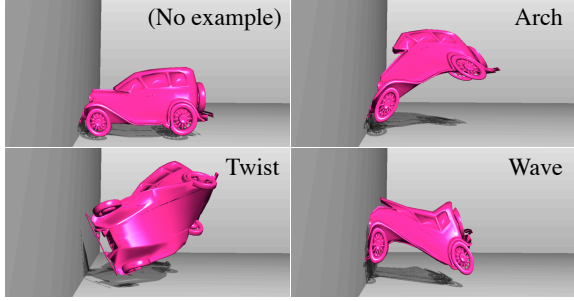**Figure 8:** *Four cylinders undergoing very different elastic deformations.*

Figure 9: *Cars colliding against wall, without example shape and with three kinds of example shapes.*

model	#particles	t_{sm}	t_{proj}	t_{tot}
cuboid (global)	225	0.21	0.091	0.33
cuboid (local)	225	0.21	0.17	0.40
cylinder	2025	2.0	0.86	3.1
teddy	1280	1.3	0.72	2.2
car	192	0.17	0.078	0.27

Table 1: *Timings per time step in milliseconds measured using a single core of Intel Core i7 M620 2.67 GHz CPU. t_{sm}, t_{proj}, and t_{tot} denote timings for shape matching, projection onto the example manifold, and total simulation, respectively.*
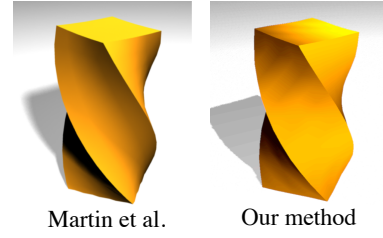

Figure 10: *Comparison between Martin et al.'s twisting elastic cuboid and ours under gravity. Note that the simulation and rendering conditions are not exactly identical. The left image is from [MTGG11], ©2011 Association for Computing Machinery, Inc. Reprinted by permission.*

an effect where each local region deforms non-uniformly according to the stress applied to it (Fig. 7). This is closer to the physical law than the case of using a globally defined example manifold, because in the real world an object's deformation occurs as a result of stress applied to each local region.

5. Results and Discussion

We can simulate many objects that exhibit example-based elastic deformations in real time (Fig. 1, 8, 9). Table 1 shows timings for the models shown in this paper. Note that the computational cost of projecting the current pose onto the example manifold is much smaller than those of the other processes. Our approach is two orders of magnitude faster than Martin et al.'s approach which typically took a few seconds per time step for models with about thousand DOFs. This real-time performance is already achieved with our unoptimized implementation; combining our approach with other techniques for accelerating shape matching such as FastLSM [RJ07] and oriented particles [MC11] would further improve the performance.

Comparison. Fig. 10 shows a rough comparison between Martin et al.'s example-based elastic materials and ours. We believe that our method successfully reproduces practically the same behavior as Martin et al.'s approach, while achieving real-time performance. Note that our 3D model, example pose, and simulation and rendering conditions are not

exactly the same as Martin et al.'s. For a comparison of the animations, please see the supplementary material.

Limitation. Although physical accuracy would be of secondary importance in the context of art-directed animations, it is desirable that the other elastic behaviors orthogonal to example poses are as physically accurate as possible to ensure naturalness. Our approach based on shape matching is inherently limited in this respect because of its purely geometrically-based nature, as opposed to Martin et al.'s FEM-based approach.

Future work. It would be possible to apply our approach to elastic thin shells and rods by using other extensions of shape matching such as oriented particles [MC11] and chain shape matching [RKN10]. Another possibility of further investigation is different way of projecting the current pose onto the example manifold. It would be interesting to take into account other contextual information such as velocity, forces, and relative positions among characters in proximity.

References

[BBO*09] Bickel B., Bächer M., Otaduy M. A., Matusik W., Pfister H., Gross M.: Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph. 28* (July 2009), 89:1–89:9. URL: http://doi.acm.org/10.1145/1531326.1531395, doi:10.1145/1531326.1531395. 2

[BMWG07] Bergou M., Mathur S., Wardetzky M., Grinspun E.: Tracks: toward directable thin shells. *ACM Trans. Graph. 26* (July 2007). URL: http://doi.acm.org/10.1145/1276377.1276439, doi:10.1145/1276377.1276439. 1

[BW08] Bonet J., Wood R. D.: *Nonlinear Continuum Mechanics for Finite Element Analysis*, 2 ed. Cambridge University Press, 2008. 6

[DBB11] Diziol R., Bender J., Bayer D.: Robust real-time deformation of incompressible surface meshes. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 237–246. URL: http://doi.acm.org/10.1145/2019406.2019438, doi:10.1145/2019406.2019438. 2

[GBB09] Gerszewski D., Bhattacharya H., Bargteil A. W.: A point-based method for animating elastoplastic solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), SCA '09, ACM, pp. 133–138. URL: http://doi.acm.org/10.1145/1599470.1599488, doi:10.1145/1599470.1599488. 6

[ITYI09] Ijiri T., Takayama K., Yokota H., Igarashi T.: Procdef: Local-to-global deformation for skeleton-free character animation. *Computer Graphics Forum 28*, 7 (2009), 1821–1828. URL: http://dx.doi.org/10.1111/j.1467-8659.2009.01559.x, doi:10.1111/j.1467-8659.2009.01559.x. 2, 3

[JF03] James D. L., Fatahalian K.: Precomputing interactive dynamic deformable scenes. *ACM Trans. Graph. 22*, 3 (July 2003), 879–887. URL: http://doi.acm.org/10.1145/882262.882359, doi:10.1145/882262.882359. 1

[KKA05] Kondo R., Kanai T., Anjyo K.-I.: Directable animation of elastic objects. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), SCA '05, ACM, pp. 127–134. URL: http://doi.acm.org/10.1145/1073368.1073385, doi:10.1145/1073368.1073385. 1

[MC11] Müller M., Chentanez N.: Solid simulation with oriented particles. *ACM Trans. Graph. 30* (Aug. 2011), 92:1–92:10. URL: http://doi.acm.org/10.1145/2010324.1964987, doi:10.1145/2010324.1964987. 2, 3, 5

[MHTG05] Müller M., Heidelberger B., Teschner M., Gross M.: Meshless deformations based on shape matching. *ACM Trans. Graph. 24* (July 2005), 471–478. URL: http://doi.acm.org/10.1145/1073204.1073216, doi:10.1145/1073204.1073216. 2, 6

[MTGG11] Martin S., Thomaszewski B., Grinspun E., Gross M.: Example-based elastic materials. *ACM Trans. Graph. 30* (Aug. 2011), 72:1–72:8. URL: http://doi.acm.org/10.1145/2010324.1964967, doi:10.1145/2010324.1964967. 1, 3, 5

[RJ07] Rivers A. R., James D. L.: Fastlsm: fast lattice shape matching for robust real-time deformation. *ACM Trans. Graph. 26* (July 2007). URL: http://doi.acm.org/10.1145/1276377.1276480, doi:10.1145/1276377.1276480. 2, 3, 5

[RKN10] Rungjiratananon W., Kanamori Y., Nishita T.: Chain shape matching for simulating complex hairstyles. *Computer Graphics Forum 29*, 8 (2010), 2438–2446. URL: http://dx.doi.org/10.1111/j.1467-8659.2010.01755.x, doi:10.1111/j.1467-8659.2010.01755.x. 2, 5

[SOG08] Steinemann D., Otaduy M. A., Gross M.: Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2008), SCA '08, Eurographics Association, pp. 87–94. URL: http://dl.acm.org/citation.cfm?id=1632592.1632606, doi:10.2312/SCA/SCA08/087-094. 2

[SZT*08] Shi X., Zhou K., Tong Y., Desbrun M., Bao H., Guo B.: Example-based dynamic skinning in real time. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 29:1–29:8. URL: http://doi.acm.org/10.1145/1360612.1360628, doi:10.1145/1360612.1360628. 1

[TJ08] Twigg C. D., James D. L.: Backward steps in rigid body simulation. *ACM Trans. Graph. 27* (August 2008), 25:1–25:10. URL: http://doi.acm.org/10.1145/1360612.1360624, doi:10.1145/1360612.1360624. 1

[TKPR06] Thürey N., Keiser R., Pauly M., Rüde U.: Detail-preserving fluid control. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), SCA '06, Eurographics Association, pp. 7–12. URL: http://dl.acm.org/citation.cfm?id=1218064.1218066, doi:10.2312/SCA/SCA06/007-013. 1

Appendix: Moment Matrix as Approximate Deformation Gradient Tensor

Here we show that the moment matrix $\mathbf{A}_r$ in (1) approximates the deformation gradient tensor with respect to the local region r. As shown by Müller et al. [MHTG05], the moment matrix is the minimizer of a quadratic energy

$$\sum_i \tilde{m}_i \|\mathbf{A}\mathbf{q}_i - \mathbf{p}_i\|^2 \qquad (8)$$

where $\mathbf{q}_i = \mathbf{x}_i^0 - \mathbf{c}^0$ and $\mathbf{p}_i = \mathbf{x}_i - \mathbf{c}$ are relative particle positions with respect to the center of mass for the rest and current configurations, respectively (subscript r is omitted hereafter). In continuum mechanics [BW08], the deformation gradient is defined as follows: considering a material point inside an object as a pivot whose rest and current positions are denoted as $\mathbf{X}$ and $\mathbf{x}$, respectively, another material point infinitesimally close to the pivot whose rest and current positions are denoted as $\mathbf{X}'$ and $\mathbf{x}'$, respectively, is related to the pivot as

$$\mathbf{x}' - \mathbf{x} = \mathbf{F}(\mathbf{X}' - \mathbf{X}) \qquad (9)$$

where $\mathbf{F}$ is called the deformation gradient tensor. By setting the pivot to the center of mass and assuming that $\mathbf{F}$ approximates the deformations of particles within the region, $\mathbf{F}$ is the minimizer of a quadratic energy

$$\sum_i w_i \|\mathbf{F}\mathbf{q}_i - \mathbf{p}_i\|^2 \qquad (10)$$

where w_i are weights. By setting weights w_i to $\tilde{m}_i$, (10) is exactly the same with (8). Note that Gerszewski et al. [GBB09] used a similar formulation for the purpose of point-based simulation of elastoplastic solids.