Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2014)
J. Bender, C. Duriez, F. Jaillet, and G. Zachmann (Editors)

Efficient Transfer of Contact-Point Local Deformations for Data-Driven Simulations

M. U. Seiler[1] and J. Spillmann[2] and M. Harders[3]

[1] Computer Vision Lab, ETH Zurich [2] VirtaMed AG [3] Interactive Graphics and Simulation Lab, University of Innsbruck

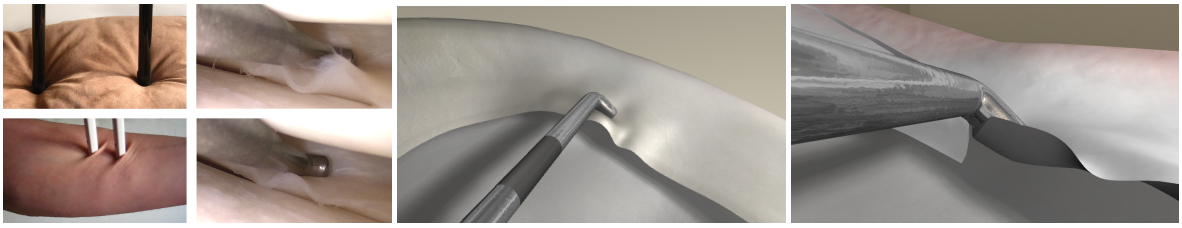


Figure 1: *Examples of characteristic deformation details around a local contact: indented pillow, sheared skin, probed meniscus (left). Coarse meniscus tissue simulation enriched with pre-generated samples (middle). The proposed method can also handle topological changes such as cuts (right).*

Abstract

We present a new approach for data-driven deformation enrichment, which requires a much smaller set of examples. The central idea is to reuse deformation samples. This is done by transferring pre-generated examples from one contact point to another, when the surrounding material as well as the induced deformation are similar. Our similarity notion is derived from two descriptors that use multivariate Hermite polynomials as a basis. The first descriptor comprehends information on the local material density near a contact point, which allows us to segment an object into regions with similar material neighborhoods. At each characteristic location, multiple samples are obtained for different interaction patterns. The obtained information is then encoded in the second descriptor – the deformation descriptor. At run-time, the two descriptors are evaluated at the current contact point. Based on the similarity to the example descriptors, suitable pre-generated data is selected, interpolated, and used to enrich an object surface. We demonstrate our method in several applications and provide quantitative evaluations.

1. Introduction

Physically-plausible simulation of deformations during user interaction is an important aspect of many virtual reality systems. Examples include the interactive manipulation of soft-tissue in surgical simulators or the interaction of a user-controlled game character with its virtual environment. The richness of the deformation directly impacts the fidelity of the simulation. However, it is often difficult to find a good trade-off between detailed deformations and high frame rates; even more so if an object is subject to detailed and non-linear deformations, such as wrinkling of skin or buckling of incompressible shells (see Fig. 1).

One possibility to address this problem is to resort to data-driven simulations. We follow a hybrid strategy similar to *e. g.*, [SSH12, SSH14], where a freely choosable coarse simulation is enriched with pre-generated example deformations. At run-time, the state of the coarse simulation serves as a *key*, which is correlated to the samples. Based on the resulting coefficients, example details are injected to enrich the surface of the object. However, such an approach requires that a similar example key the current one can be found. However, there are combinatorially many possible deformations of the coarse simulation grid. This state-space explosion renders the example generation process uneconomic,

DOI: 10.2312/vriphys.20141221

and memory consumption can quickly become unmanageable.

Therefore, our proposal is to exploit *similarities* in order to reduce the number of required examples. The fundamental observation is that deformations, such as wrinkles, often resemble each other in similar regions of the same object. Deformation details are typically influenced by the geometry and material properties in a neighborhood around a contact point as well as the induced external load. Commonly, one exploits similarities in data by approximating it with a low-rank basis, for example by using singular value decomposition (SVD), see *e. g.*, [JF03, FKY08]. However, it appears somewhat counterproductive to first acquire a large amount of examples only to discard most of it in a second reduction step. Especially in cases where the acquisition process is expensive, minimizing the amount of required examples becomes a key priority. Our assumption is that an *a priori* available signal, such as geometry, allows to decide where interesting behavior occurs, *e. g.*, thin meniscus regions exhibit more complex deformations than thicker ones.

To this end, we conceptually segment an object into regions that likely exhibit similar surface deformations when manipulated in the same fashion. This allows us to generate examples once per region at an arbitrary representative acquisition point. In order to classify regions and deformations we employ meshfree numeric quantities commonly called *descriptors* or *signatures*. In particular we use two descriptors, the *site descriptor* to segment the object into similar regions and the *deformation descriptor* to compute the similarity between deformations. Subsequently, examples can be transferred to other contact points if the descriptors match.

In order to transfer the actual deformations, we follow the approach in [SSH12]. Geometric surface modifications are represented via displacement textures, thus following a meshfree paradigm. A consequent use of meshfree representations allows us to transfer examples from one site to another and even support cutting by following the approach in [SSH14]. In fact, the destination could also be part of a different object consisting of similar material and geometrical structure. Summarizing, our contribution is a data-driven enrichment method that exploits similarities for example reuse:

- We propose a local site descriptor to classify an object into regions where we reuse the same examples. Subsequently, we generate examples for each region separately.
- We propose a meshfree deformation descriptor that represents the local material deformation around the contact point. This also supports topology modifications.
- At run-time we compute the site and deformation descriptors at a contact point. This feeds into a correlation technique to compute blending weights, which are used to compute blended example displacements to enrich an object's surface detail.

2. Related Work

Data-driven methods have for some time received increasing attention in computer graphics, for instance in human pose estimation [ACP02], crowd simulation [LCHL07] or wrinkle animation [RK13]. Our application domain is physically-based simulation of deformations [NMK*05]. Our work can be classified as data-driven surface enrichment post-processing. The basis of our method is a coarse real-time simulation from which we obtain a smooth surface, that is then enriched with pre-generated accurate deformations extracted from examples. It should be noted that our enrichment approach is independent of the underlying real-time deformation method. Any simulation that provides a set of discrete displacements can be used.

Data-Driven Animation: In principle, there are two kinds of data-driven simulations. In the first, the data influences the course of the simulation, see *e. g.*, [BBO*09]. In this context, the terms *data* and *example* are often used interchangeably. Example-based materials [MTGG11, KTUI12] also belong to this category of data-driven methods. In such methods, a mesh with a constant high-resolution is used. This is in contrast to the second kind of methods that produce a detailed output, given a low-quality input. With [LCF00] as an inspirational origin, many methods are one-way-coupled enrichment algorithms that do not form a closed loop with the simulation. One finds them in facial animation, cloth simulation, and recently elastic solids and fluids. In a way, all methods use some sort of *key* to control some sort of example blending. In Lewis' case the key is based on angles between animation bones and a differential displacement field is linearly blended. In contrast, [KV08] use directly the vertex positions of bones. Similarly, marker positions are employed in facial animation in [BLB*08, FKY08]. Alternatively, [ZBO12] examines the stretch-ratio of simulation element edges to construct a cloth enrichment method. This sub-category of data-driven garment simulation has attracted considerable attention, starting with the work of [CMT05] followed by [PZB*09, dASTH10, KGBS11].

Our research is carried out in the context of surgery simulation. Where we follow [HHS08, SH13, SSH12] and abstract the tool contact region into a single point. In order to support cutting, we reuse the approach presented in [SSH14] that uses a real-time influence map around cuts to handle discontinuities. As a significant improvement, our new method does not fade out examples in regions where the object is cut thanks to our localized mesh-free descriptor formulation.

Descriptors: Abstractly speaking, a descriptor is a set of numbers that is produced from a given original quantity. In general, the descriptor loses information, *i. e.*, the quantity cannot be reconstructed without error. However, this is ok as long as it can be used to measure mutual similarity. In particular, *shape descriptors* are used in shape retrieval to define similarity, see *e. g.*, [TV04, BKS*05].

3. Overview

Our hybrid data-driven method enriches a smooth approximate deformation of a virtual object – given by material domain $\Omega \subset \mathbb{R}^3$ – with additional details. The latter are not simulated, but obtained by mixing pre-generated example data. Hence, our method can be divided into two main phases: first, the preprocessing phase that generates the examples, and second, the interaction phase that enriches a current state using the examples.

The example generation is a major part of the preprocessing phase and can be done in a number of ways, such as computer-based simulation, real-world capturing, or manual modeling. The i-th generated example is used to obtain a differential displacement field $\mathbf{d}[i] : \partial\Omega \rightarrow \mathbb{R}^3$ that associates a displacement vector with each point on the object's surface $\partial\Omega$. For instance, $\mathbf{d}[i]$ could be obtained as the difference between a coarse and an accurate simulated surface, see [SSH12]. We store $\mathbf{d}[i]$ together with other quantities, such as the site descriptor that we will introduce later. Throughout the remainder of this paper, we follow the convention that we indicate quantities related to a pre-generated example with index i as subscript and, if the data is obtained from the database, with bracket notation $[i]$. In contrast, time-dependent quantities stemming from the interaction phase are denoted by a plain symbol without subscripts.

The real-time simulation is based on an approximate – and therefore fast – deformation model, computed on a coarse simulation mesh. A continuous time-dependent displacement function $\mathbf{u} : \Omega \rightarrow \mathbb{R}^3$ is determined that updates each rest-state material point $\mathbf{x} \in \Omega$ to its deformed position $\mathbf{x} + \mathbf{u}(\mathbf{x})$ at time t (see Fig. 2). The real-time simulation is used during the interaction phase. With it a coarse displacement field $\mathbf{u}[i]$ is computed that is used for correlation. In order to do this, we record the boundary conditions induced by the tool, *e. g.*, tool collision points, and apply them as hard constraints while the real-time simulation runs. In order to address the object's surface, we introduce the time-dependent surface deformation function $\mathbf{s}(\mathbf{x}) : \partial\Omega \rightarrow \mathbb{R}^3 := \mathbf{x} + \mathbf{u}(\mathbf{x})$ that is restricted to $\partial\Omega$. The function $\mathbf{s}$ will be enriched with the previously generated examples.

The detail enrichment in the interaction phase is achieved by first using the real-time simulation to obtain the current base-quantities, that is $\mathbf{s}$, the descriptors, and the surface contact point $\mathbf{p}$. Based on this we compute the time-dependent *enrichment function* $\mathbf{e} : \partial\Omega \rightarrow \mathbb{R}^3$ that associates with each surface point a displacement. To compute $\mathbf{e}$, we compare the local material neighborhood and deformation around $\mathbf{p}$ to all examples and mix the associated $\mathbf{d}[i]$ appropriately. Subsequently, $\mathbf{e}$ is employed to compute an enriched base surface

$$\tilde{\mathbf{s}}(\mathbf{x}) := \mathbf{s}(\mathbf{x}) + \mathbf{e}(\mathbf{x}) \tag{1}$$

that is more accurate and can exhibit considerably more surface detail, thus increasing visual fidelity. Our enrichment function follows a contact-point-centric displacement mapping technique denoted *stamping*.

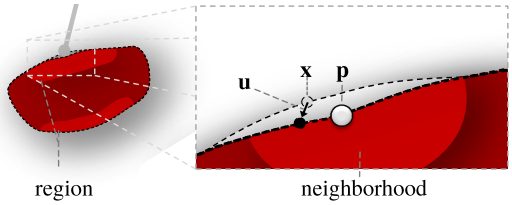


Figure 2: *Example of an object (in dark red) deformed by a probing tool (in light grey). The close-up depicts a single material point with its rest-state positions* **x** *and its displacement* **u**. *Examples can be reused at contact points that belong to the same* region. *In order to determine the region of a contact point* **p** *we analyze the geometry in its* neighborhood.

The stamp $\Psi[i] : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is generated by projecting the i-th example displacements $\mathbf{d}[i]$ to a surface-tangential plane that originates at $\mathbf{p}[i]$. Ψ is essentially a function that maps two-dimensional local surface parameters to displacements. Since we use a rather simple but fast orthogonal projection as a mapping function, the current method works best for locally planar surfaces. By employing displacement mapping the blended example displacements can be superimposed on a surface at any desired point given a local frame $\mathbf{B}$ (see Fig. 3).

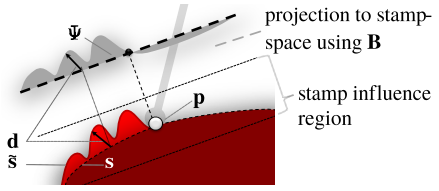


Figure 3: *Depicted is a slice through a deformable object (*e. g.*, an organ covered by skin providing tensile strength). The surface forms a characteristic wrinkle pattern when displaced laterally at a contact point* **p** *with a tool. The stamp* Ψ *contains blended example displacements. The base surface* **s** *is enriched with displacement* **d** *obtained from* **e**. *It involves a mapping of the surface points to the stamp-coordinates spanned by the surface tangential frame* **B**.

The local frame $\mathbf{B} \in \mathbb{R}^{3\times3}$ is an orthonormal basis and originates always at a contact point. $\mathbf{B}[i]$ stands for the stamping frame that is used for example i, originating at $\mathbf{p}[i]$. It is a crucial component of the algorithm since it defines the stamp-space transform that controls how stamps are aligned when added to a base surface $\mathbf{s}$. The third column of $\mathbf{B}[i]$ is chosen such that it aligns with the surface normal of $\partial\Omega$ at $\mathbf{p}[i]$. We describe how the remaining rotational degree of freedom can be removed in a canonical way in Section 4.4.

4. Descriptors

Before we introduce our descriptors, we highlight the observation that for many objects the occurrence of a typical deformation pattern depends mostly on two factors: first, the local object properties, such as geometric shape and material parameters, and second, the locally induced loads, *e. g.*, due to contact with a tool. Hence, we propose two types of numeric descriptors: the site descriptor and the deformation descriptor. In theory, all involved quantities could be merged into a single descriptor, but then one would lose the ability to transform and parameterize them independently, a capability that we will make use of later. The site descriptor contains information of the local material density $\rho : \mathbb{R}^3 \to \mathbb{R}_0^+$. Similarly, the deformation descriptor captures the information of the local material displacement, *i. e.*, the earlier introduced three dimensional vector $\mathbf{u}$. In order to compare descriptors we need a similarity measure that is fast to compute, thus allowing to quickly find similar descriptors. To this end, we base our approach on an a localized orthogonal basis, specifically on *Hermitian moments*, that we will discuss next.

4.1. Hermitian Moments

Mathematical moments have a lot in common with the well-known statistical quantities of *mean* and *variance*. The motivation to use moments lies in the fact that they abstract the underlying mesh structure. In particular, we use Hermitian moments that are based on the Hermite polynomials that find wide application, *e. g.*, in quantum physics or image analysis [SCER06]. They are interesting to us for two reasons: first, they operate in a smooth and localized window defined by a Gaussian weighting function (this property is not shared with other orthogonal bases, such as Legendre or Chebychev polynomials), and second, the inner product between two functions is particularly efficient to compute, since, due to orthogonality, most terms cancel out. Based on the inner-product, simple and efficient similarity measures between descriptors can be defined. For the same reasons, Hermite polynomials are used in other contexts, *e. g.*, [Gra49] and more recently [YLZD11]. Next, we define the relevant key properties.

Let $\mathbf{H}_n : \mathbb{R}^3 \to \mathbb{R}^{3^n}$ denote the n-th multivariate Hermite polynomial in 3D. The range $\mathbb{R}^{3^n}$ already hints at the fact that ordinary vectors and matrices will not be sufficient, but that *tensors* – a generalization of matrices that can be understood as multi-dimensional arrays, see *e. g.*, [BW89] – are required to represent moments of order three and more. One way to calculate $\mathbf{H}_n$ is to use a Rodrigues formula [AO05] that computes the polynomial components by taking partial derivatives of the weighting function $\omega : \mathbb{R} \to \mathbb{R}^+$ that is defined as

$$\omega(r) := \frac{1}{(2\pi)^{\frac{3}{2}}} \exp\left(-\frac{r^2}{2}\right), \tag{2}$$

where $r \in \mathbb{R}$ is the radius from the center of the evaluation point. Following Rodrigues, one computes $\mathbf{H}_n$ by taking all possible combinations of n partial derivatives of ω

$$\mathbf{H}_n(\mathbf{r}) := \frac{(-1)^n}{\sqrt{n!}\,\omega(||\mathbf{r}||_2)} \nabla^{\otimes n} \omega(||\mathbf{r}||_2) = [\underbrace{H_{\dots}}_{n}] \in \mathbb{R}^{n^3}, \tag{3}$$

where $\mathbf{r} = [r_i] \in \mathbb{R}^3 = \mathbf{x} - \mathbf{p}$, $\nabla = [\frac{\partial}{\partial x_i}]$, and the operator $(\cdot)^{\otimes n}$ denotes exponentiation using the tensor product or dyadic product, *e. g.*, $[\mathbf{r}^{\otimes 3}]_{ijk} = [r_i r_j r_k] \in \mathbb{R}^{3\otimes 3\otimes 3}$. To exemplify, we write down the second Hermitian moment – a symmetric second-order tensor – representable as a matrix

$$\mathbf{H}_2(\mathbf{r}) = \frac{1}{\sqrt{2}} \begin{bmatrix} r_1^2 - 1 & r_1 r_2 & r_1 r_3 \\ r_1 r_2 & r_2^2 - 1 & r_2 r_3 \\ r_1 r_3 & r_2 r_3 & r_3^2 - 1 \end{bmatrix} = \left[H_{ij}\right] \in \mathbb{R}^{3\times 3}.$$

A defining feature of the multivariate Hermite polynomials is that they form an orthonormal basis for function spaces. Thus, we can approximate a function $f : \mathbb{R}^3 \to \mathbb{R}$ by a finite weighted sum of Hermite polynomials. To this end, one first projects f onto each Hermite basis function and – analogously to projecting a Cartesian vector onto its basis vectors – obtains a set of coefficients which determine how to scale each particular basis function. As in the given vector analogy, the projection is based on inner products, which is in the case of Hermite polynomials defined as

$$\langle f, g \rangle_H := \int_{\mathbb{R}^3} \omega(||\mathbf{r}||_2)\, f(\mathbf{r})\, g(\mathbf{r}) dV, \tag{4}$$

where $g : \mathbb{R}^3 \to \mathbb{R}$ is a second function like f. Further, the differential volume $dV := dr_1 dr_2 dr_3$ is defined as the product of the differential components of $\mathbf{r}$. Naturally, the inner product leads to a norm $||f||_H := \sqrt{\langle f, f \rangle_H}$ that will be of use later. Given the inner product one can approximate f with $\tilde{f}$ as a finite series of $N+1$ multivariate Hermite polynomials

$$\begin{aligned} \tilde{f}(\mathbf{x}) &:= \underbrace{\langle H, f \rangle_H}_{\boldsymbol{\tau}_0} H(\mathbf{x}) + \sum_{i=1}^{3} \underbrace{\langle H_i, f \rangle_H}_{[\boldsymbol{\tau}_1]_i} H_i(\mathbf{x}) + \dots \\ &= \sum_{n=0}^{N} \boldsymbol{\tau}_n \bullet \mathbf{H}_n(\mathbf{x}) \end{aligned} \tag{5}$$

where the individual coefficients are grouped in tensors $\boldsymbol{\tau}_n \in \mathbb{R}^{3^n}$ that contain the projection of f onto $\mathbf{H}_n$. The formulation makes use of the *generalized dot product* ($\bullet$) that is essentially a component-wise multiplication with the tensor followed by summation (see Appendix A for details). More information on tensors can be found in [KB09]. If we let $N \to \infty$ in (5) then the approximation error becomes zero for all practically relevant functions. In our case, we cut off this approximation already at $N = 3$ which results in a good local approximation to f due to the strong falloff of the exponential in (2).

As already mentioned, thanks to orthogonality, most factors cancel out from $\langle \cdot, \cdot \rangle_H$ and we obtain a simple expres-

sion

$$\langle \tilde{f}, \tilde{g} \rangle_H = \langle \boldsymbol{\tau}, \boldsymbol{\upsilon} \rangle := \sum_{n=0}^{N} \boldsymbol{\tau}_n \bullet \boldsymbol{\upsilon}_n, \tag{6}$$

where $\boldsymbol{\upsilon}$ encodes g. Also note that we use the tensor name ($\boldsymbol{\tau}$ and $\boldsymbol{\upsilon}$) without indices as a symbol for the list of all Hermite moments, *e. g.*, $\boldsymbol{\tau} := (\boldsymbol{\tau}_0, \ldots, \boldsymbol{\tau}_N)$ for an approximation of order N. The inner-product $\langle \cdot, \cdot \rangle$ serves as a short hand for the sum of generic tensor dot products. Based on this, we can write the Hermitian norm simply by $||\tilde{f}||_H = ||\boldsymbol{\tau}|| := \langle \boldsymbol{\tau}, \boldsymbol{\tau} \rangle$ where we introduce $||\cdot||$ as an analogous short hand to $\langle \cdot, \cdot \rangle$. We will make more use of these concepts later on when we define the similarity measures for our descriptors.

To conclude this short introduction, Fig. 4 depicts two Hermitian basis functions. The plots illustrate that low order Hermitian moments are suited to extract local information due to their mostly local support. In the next two sections we will use the multivariate Hermitian polynomials to define localized descriptors for geometry and deformation.

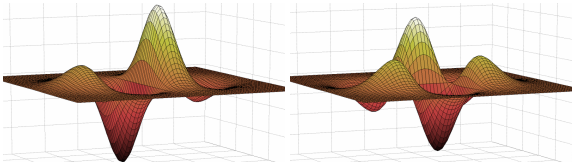


Figure 4: *Plot of two third-order multivariate Hermitian basis functions in 2D. Depicted on the left is* $H_{111} = (r_1^3 - 3r_1)\,\omega(||\mathbf{r}||)$ *and on the right* $H_{122} = (r_1 r_2^2 - r_1)\,\omega(||\mathbf{r}||)$. *Their smooth, localized shape due to* ω *is clearly visible.*

4.2. The Site Descriptor

In general, the site descriptor can contain all static information about the object in a neighborhood around its evaluation point $\mathbf{p}$. In this work, we only capture the density function $\rho : \mathbb{R}^3 \to \mathbb{R}_0^+$ in a neighborhood around $\mathbf{p}$. Note that the density function $\rho(\mathbf{x})$ is 0 iff $\mathbf{x} \notin \Omega$ and, hence, it is discontinuous. This makes it difficult to approximate it with the smooth Hermitian basis. However, our goal is not error-free reconstruction but computing similarities. By comparing the components of the moments evaluated at different surface points we can effectively quantify the similarity between regions despite a large reconstruction error, see the evaluation in section 6.1.

We define the n-th multivariate Hermite moment of ρ around $\mathbf{p}$ as $\boldsymbol{\rho}_n \in \mathbb{R}^{3^n}$ for $n \in \mathbb{N}^+$ as

$$\boldsymbol{\rho}_n := \int_{\mathbb{R}^3} \omega(||\mathbf{x} - \mathbf{p}||_2)\, \mathbf{H}_n(\mathbf{x} - \mathbf{p})\, \rho(\mathbf{x})\, dV \tag{7}$$

which corresponds to the inner product in (4) written in tensor form and evaluated at $\mathbf{p}$. Each moment can be understood as a projection of ρ onto a set of orthonormal basis functions by using the inner product $\langle \cdot, \cdot \rangle_H$ that is the fundamental building block of (7). Note that the integration domain is $\mathbb{R}^3$. However, since ρ is 0 outside of Ω one can restrict the latter to Ω.

In practice, we approximate Ω with a discrete point cloud given by M points $\mathbf{x}_m \in \Omega$. In fact, we simply use the points of a uniform tetrahedral mesh that represents Ω. We then solve the integral in (7) by employing a variant of Monte Carlo integration that results in an averaged sum over the discrete points $\mathbf{x}_m$:

$$\boldsymbol{\rho}_n \approx \eta_n \sum_{m=1}^{M} \rho(\mathbf{x}_m)\, \nabla^{\otimes n} \omega_m, \tag{8}$$

where $\omega_m := \omega(||\mathbf{x}_m - \mathbf{p}||)$ is the m-th weight and $\eta_n := (-1)^n \left(\sqrt{n!} \sum_{m=1}^{M} \omega_m\right)^{-1}$ the corresponding normalization factor of the n-th moment. Fig. 5 depicts an approximation to ρ using the first two (up to $\boldsymbol{\rho}_1$) and the first four (up to $\boldsymbol{\rho}_3$) Hermitian moments for three different density functions. It can be seen that the approximation error ε is reduced if higher-order moments are taken into account. Employing higher-order moments also facilitates the capturing of material asymmetries. In practice, we often chose to incorporate the first four moments since this provides a good trade-off between accuracy and speed.

There is one more crucial detail that has to be mentioned: In order to evaluate our similarity measure the tensor components need to be represented with respect to the same frame. For example, if one wants to compute the difference between two vectors $\mathbf{a}$ and $\mathbf{b}$, then the components have to be expressed with respect to the same basis before they can be subtracted from each other to compute a measure like $||\mathbf{a} - \mathbf{b}||$. Analogously, also the components of tensors need to be expressed with respect to the same basis to evaluate our similarity measure. To this end, all tensor components are rotated to the frame $\mathbf{B}$ at $\mathbf{p}$ after evaluation by using the rotation formula given in Appendix B. Hence, the descriptors can be evaluated without first transforming all points to $\mathbf{B}$ which is not even known at that time. Then, the obtained components can simply be rotated to any desired frame - including $\mathbf{B}$ once it is computed - thanks to the tensor property of the descriptors.

Analogous to (5), given the tensors $\boldsymbol{\rho}_k$ one can approximate ρ with $\tilde{\rho}$ in the neighborhood of $\mathbf{p}$ by superimposing a finite series of basis functions $\mathbf{H}$ scaled with the components found in $\boldsymbol{\rho}_k$. The approximations are then used to define the distance metric d_ρ that measures the mass difference between the two approximations $\tilde{\rho}$ and $\tilde{\rho}'$ with descriptors $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ that are evaluated at the two contact points $\mathbf{p}$ and $\mathbf{p}'$:

$$d_\rho\left(\boldsymbol{\rho}, \boldsymbol{\rho}'\right) := ||\tilde{\rho} - \tilde{\rho}'||_H = ||\boldsymbol{\rho} - \boldsymbol{\rho}'||. \tag{9}$$

Intuitively, this measures the sum of the squared difference between $\tilde{\rho}$ and $\tilde{\rho}'$ locally weighted with ω. By using the site descriptor and its similarity measure we can now find con-

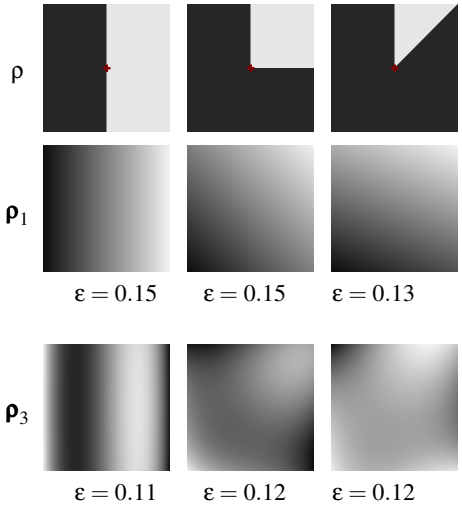


Figure 5: *Analysis of three different two-dimensional functions* ρ *with range* $[0,1]$ *in the first row: A flat region, a 90 degree corner, and a 45 degree corner. The following rows depict the approximation with Hermitian moments. The row* $\boldsymbol{\rho}_1$ *uses only the first two moments. The last row* $\boldsymbol{\rho}_3$ *depicts a superposition using the first four moments.* ε *denotes the average reconstruction error.*

tact points that we assume behave in the same way when deformed locally. Next, we need a way to measure similarity between the various types of locally inducible deformations.

4.3. The Deformation Descriptor

Our goal is to devise a deformation descriptor (paired with a similarity measure) analogous to the site descriptor from the previous section. We are again opting for a localized approach for much of the same reasons that we already pointed out for the site descriptor. An additional rationale to use a localized deformation measure is that it naturally extends to multiple interactive tools. Hence, we take the same approach, except, that this time we do not analyze a simple scalar function, like ρ, but the three components of the vector-valued displacement field $\mathbf{u}$.

In order to obtain good results, our deformation descriptor should be invariant under a super-imposed rigid body transformation $\mathbf{Rx}+\mathbf{t}$ where $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is the (infinitesimal) rotation tensor at $\mathbf{x}$, *i. e.*, $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, and $\mathbf{t} \in \mathbb{R}^3$ is a translation. Intuitively, this means that the deformation descriptor evaluated at a contact point $\mathbf{p}$ should not depend on how the object is positioned and oriented in space, but only on the locally induced deformation. One solution is to rely on the infinitesimal rigid body transformation at $\mathbf{p}$ that plays an important role in continuum mechanics [PB83]. It is commonly computed by a polar decomposition of the deformation gradient $\mathbf{F} = \nabla\mathbf{u} + \mathbf{I} = \mathbf{RS}$ that yields the aforementioned infinitesimal rotation tensor $\mathbf{R}$, as well as $\mathbf{S} \in \mathbb{R}^{3\times 3}$ that contains the pure deformation. To this end, we use a moving least squares approach to compute $\mathbf{F}$ based on the $\mathbf{u}(\mathbf{x}_m)$ and the ω_m as weights, see *e. g.*, [MHTG05]. As a result, we obtain an averaged translation $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{F}$ at $\mathbf{p}$ with which one can compute

$$\bar{\mathbf{u}}(\mathbf{x}) := \mathbf{R}^T(\mathbf{x}+\mathbf{u}-\mathbf{t}) - \mathbf{x}, \tag{10}$$

which is the displacement field without the super-imposed rigid body transformation at $\mathbf{p}$, *e. g.*, $\bar{\mathbf{u}}(\mathbf{p}) = \mathbf{0}$.

To compute the deformation descriptor, we perform in principle the same procedure as before, but now for each of the three components of $\bar{\mathbf{u}}$ individually. That is, we compute the Hermitian displacement moments $\boldsymbol{\mu}_n \in \mathbb{R}^{3\times 3^n}$ analogous to (8) via

$$\boldsymbol{\mu}_n \approx \eta_n \sum_{m=0}^{M} \bar{\mathbf{u}}(\mathbf{x}_m) \otimes \nabla^{\otimes n}\omega_m. \tag{11}$$

The zeroth moment $\boldsymbol{\mu}_0$ is $\mathbf{0}$ since it corresponds to the average displacement in a region around $\mathbf{p}$ which has been removed from $\bar{\mathbf{u}}$. Consequently, it can be dropped. In practice, using second-order moments yields sufficient results, since the local coarse deformations are sufficiently smooth. Finally, similar to $\boldsymbol{\rho}$, we express all tensor components in the local frame $\mathbf{B}$ at $\mathbf{p}$ by using the rotation formula given in Appendix B. The approximation $\tilde{\mathbf{u}}$ of $\bar{\mathbf{u}}$ at $\mathbf{p}$ can now be written as

$$\bar{\mathbf{u}}(\mathbf{x}) \approx \tilde{\mathbf{u}}(\mathbf{x}) := \sum_{n=1}^{N} \boldsymbol{\mu}_n \bullet \mathbf{H}_n(\mathbf{x}-\mathbf{p}) \tag{12}$$

which approximates $\bar{\mathbf{u}}$ in the neighborhood of $\mathbf{p}$.

In data-driven methods, the similarity between two keys, *e. g.*, between two deformation descriptors, is almost always based on inner products. For example, the Euclidean metric $||\mathbf{a}-\mathbf{b}||_2$ for vectors $\mathbf{a},\mathbf{b}$ can be written as $\langle\mathbf{a},\mathbf{a}\rangle - 2\langle\mathbf{a},\mathbf{b}\rangle + \langle\mathbf{b},\mathbf{b}\rangle$ by using inner products only. Consequently, it is sufficient to define an inner product $\langle\cdot,\cdot\rangle_{\mathbf{u}}$ for a particular key that can then be used to define most similarity measures, *e. g.*,

$$\langle\tilde{\mathbf{u}},\tilde{\mathbf{u}}'\rangle_{\mathbf{u}} := \int_{\mathbb{R}^3} \omega(\mathbf{x}-\mathbf{p})\, \tilde{\mathbf{u}}(\mathbf{x})^T\, \tilde{\mathbf{u}}'(\mathbf{x})\, dV = \langle\boldsymbol{\mu},\boldsymbol{\mu}'\rangle, \tag{13}$$

which states that the inner product between two Hermitian approximations can be expressed as the sum of generic dot products between coefficient tensors (also see equation (6)). The reason for this simple expression is again the orthogonality of the Hermitian polynomials.

Now that we have a fast inner product available that works directly with the descriptor components we continue to derive a similarity measure. Similar to [FKY08], we use the normalized cross-correlation which can also be expressed by only using inner products and norms. This shows the versatility of our novel contact point-centric formulation since it can easily be plugged into pre-existing methods. Our defini-

tion of $d_{\mathbf{u}}$ is

$$d_u(\boldsymbol{\mu},\boldsymbol{\mu}') := 1 - \frac{\langle \boldsymbol{\mu},\boldsymbol{\mu}' \rangle}{||\boldsymbol{\mu}||\,||\boldsymbol{\mu}'||}. \tag{14}$$

4.4. The Stamping Frame B

Until now, we left the stamping frame $\mathbf{B}$ undefined. It is, however, a crucial quantity for the stamping approach [SSH12] since it determines how stamps are aligned when applied at different contact points. It consists of three basis vectors $[\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z] \in \mathbb{R}^{3\times 3}$. We start by setting $\mathbf{b}_z$ to the surface normal at $\mathbf{p}$. This leaves one degree of freedom, namely an angle defining the orientation of the stamp. We found that there is a simple solution that provides satisfactory results at high performance. The idea is to remove the remaining degree of freedom by requiring that $\mathbf{b}_x$ point along the first moment $\boldsymbol{\rho}_1$. The last column vector is then given by $\mathbf{b}_y = \mathbf{b}_z \times \mathbf{b}_x$. This simple approach permits to define a canonical frame in most cases. In theory, $\boldsymbol{\rho}_1$ does not allow to chose a preferred direction for certain configurations, *e. g.*, an infinite plane; in such cases, however, it does not matter how the stamp is applied, since the neighborhood is rotationally symmetric with respect to the surface normal. In practice, there is always a slightly preferred direction due to numerical imprecision.

5. Run-Time Method

In the interaction phase, the previously discussed concepts are combined to build a complete method. We first evaluate both the current site and deformation descriptor at $\mathbf{p}$. Then, we compute the dissimilarities d_ρ and $d_{\mathbf{u}}$ between the current base state $(\boldsymbol{\rho},\boldsymbol{\mu})$ and all example base states $(\boldsymbol{\rho}[i],\boldsymbol{\mu}[i])$ in the current stamping frame $\mathbf{B}$. Similar to [MWLT13], we combine the dissimilarities d_ρ and $d_{\mathbf{u}}$ into example blending weights $\mathbf{w} = [w_i] \in \mathbb{R}^N$ by using Gaussian kernels

$$w_i := \exp\left(-\frac{d_\rho(\boldsymbol{\rho},\boldsymbol{\rho}[i])^2}{2\sigma_\rho^2} - \frac{d_{\mathbf{u}}(\boldsymbol{\mu},\boldsymbol{\mu}[i])^2}{2\sigma_{\mathbf{u}}^2} \right), \tag{15}$$

where $\boldsymbol{\rho}$ and $\boldsymbol{\mu}$ are the current descriptors based on ($\mathbf{p}$, $\mathbf{u}$) and $\boldsymbol{\rho}[i]$ and $\boldsymbol{\mu}[i]$ are analogously the descriptors for the i-th example. We determine the kernel-width parameters σ_ρ and $\sigma_{\mathbf{u}}$ by minimizing the cross-validation error against the example data (see [SSH14] for a detailed evaluation). This optimization is part of the pre-processing phase and is commonly finished in a few seconds with a simple brute-force algorithm. However, if more parameters, *e. g.*, Young's modulus, would be added to the site descriptor then the number of required examples would need to increase exponentially with the number of parameters. Otherwise, meaningful results cannot be achieved due to the curse of dimensionality. Thus we keep the number of parameters as minimal as possible.

The weights $\mathbf{w}$ could now directly be used to blend the examples. However, better results can be achieved if an additional interpolation step, such as radial basis function (RBF) interpolation is used (see *e. g.*, [FKY08]). In our case, we follow the RBF method presented in [SSH14] to obtain the interpolating weights $\tilde{\mathbf{w}} = [\tilde{w}_i] \in \mathbb{R}^N$. The detail enrichment is based on blending per-example displacement maps $\Psi[i](\phi,\varphi) : \mathbb{R}^2 \to \mathbb{R}^3$ that associate stamp coordinates (ϕ,φ) to surface displacements. The blending of the per-example stamps $\Psi[i]$ is done for each contact point with the corresponding weights $\tilde{\mathbf{w}}$ via

$$\Psi(\phi,\varphi) := \sum_{i=1}^{N} \Psi[i](\phi_i',\varphi_i')\ \tilde{w}_i, \tag{16}$$

where ϕ_i' and φ_i' are the transformed stamping coordinates using the transformation $\mathbf{T}_i := \mathbf{BB}[i]^T$ that ensures a correct alignment of the stamps. The surface is then enriched with Ψ via displacement mapping (see *e. g.*, [GH99]).

6. Results

We describe a number of experiments carried out to illustrate the performance of our proposed method.

6.1. Site Descriptor Evaluation

This experiment demonstrates the discriminative capability of the site descriptor. The descriptor is evaluated for all surface vertices in several tetrahedral meshes, with an order up to $N = 3$. Subsequently, we cluster all descriptors by using d_ρ as distance measure for the standard *k-means* algorithm [Ste56]. This effectively leads to a segmentation of the object into regions, each consisting of points that have a similar local material neighborhood. In Fig. 6 we show the clustering results for some shapes. Different regions are rendered with different colors. It can be seen, that the descriptor is able to discriminate thinner from thicker parts, as well as corners and edges, and large flat or round regions. It should be mentioned that these regions are only used for visualization purposes and not explicitly in our method. Nevertheless, the visualization is provided during example acquisition, serving as a guide to select appropriate interaction points. Finally, note that the transition between regions does not have to be considered in our algorithm.

6.2. Deformation Descriptor Evaluation

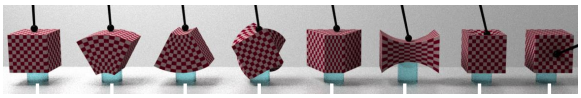
In this experiment we illustrate that our deformation descriptor is capable of discriminating various local deformation modes. To this end, we evaluate the descriptors up to order $N = 2$ at the same contact point on the top of a cube with edge length 2 that is deformed with different deformation modes. The resulting pair-wise dissimilarities are listed in Tab. 1. The resulting cross-dissimilarity matrix is symmetric and has a diagonal with all zeros. Deformation 8 is simply a rotated version of deformation 7. Since our measure is invariant under rigid body transformation, they are considered

(a) (b) (c)

Figure 6: *Evaluation of the site descriptor's discriminative capability. In (a): the segmentation of the pillow shows the large interior part for which one can use the same example. In (b): a top view of a lateral meniscus with 13 (conceptual) regions. In (c): the descriptor is able to differentiate between flat regions at various distances from the edges and corners.*

equal. The results indicate that using $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ yields a measure that differentiates sufficiently well between deformation modes.

Table 1: *Cross-dissimilarities $d_{\boldsymbol{\mu}}$ (with $\sigma_{\boldsymbol{\mu}} = 0.8$) of deformation descriptors for a set of synthetic example deformations.*



$d_{\boldsymbol{\mu}}$	1	2	3	4	5	6	7	8
1 rest state	0.0	4.1	4.1	4.2	4.1	5.7	4.1	4.1
2 bent up	4.1	0.0	8.0	4.3	4.2	6.9	3.0	3.0
3 bent down	4.1	8.0	0.0	4.0	4.2	4.0	4.3	4.3
4 twisted	4.2	4.3	4.0	0.0	4.2	7.4	3.7	3.7
5 tapered	4.1	4.2	4.2	4.2	0.0	5.7	4.2	4.2
6 stretched	5.7	6.9	4.0	7.4	5.7	0.0	5.0	5.0
7 dented	4.1	3.0	4.3	3.7	4.2	5.0	0.0	0.0
8 rotated 7	4.1	3.0	4.3	3.7	4.2	5.0	0.0	0.0

6.3. Performance Evaluation

The performance of our example correlation is such that a synthetic data-base with 1000 examples is handled in about 0.2 milliseconds (single 3GHz CPU, SIMD optimized). In practice, we use typically between 5 and 10 examples at runtime that are handled in the order of 10 microseconds. The weights are then sent to the GPU where we measured similar results to the ones reported in [SSH14].

6.4. Sampling Density Evaluation

A common question in data-driven approaches is that of how many examples should be used? In this experiment, we provide one way to answer this question. We use the pillow example and take an increasing amount of examples at uniformly sampled contact points. To measure the effectiveness of adding more examples, we perform a leave-one-out cross-validation and plot the measured RMS error, see [SSH14]. This quantity is a good measure for the *generalization error*, *i. e.*, how well the system generalizes to a new, untrained example. Fig. 7 reveals an elbow-like curve of diminishing returns. Such a slowly decreasing generalization error is an indicator that enough examples are present.

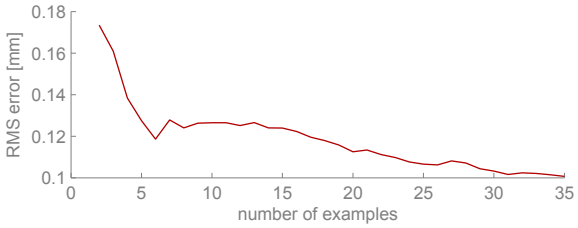
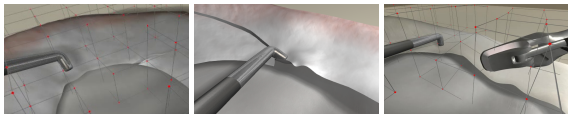


Figure 7: *RMS error for an increasing number of examples.*

6.5. Meniscus

The driving application of our developments is surgical simulation. The key target is the real-time reproduction of detailed deformation behavior of knee menisci during arthroscopy. For this application, our proposed approach allows us to reduce the number of examples to only seven, which are then transferred to appropriate similar sites. Note that in the case of the meniscus, deformation details such as folds mainly appear on the thin sections near the inner rim. Therefore, we acquire examples only in this region.

Our method also permits the transfer of details between similar objects. In our simulation, we for instance reuse the samples from the medial meniscus mesh on the lateral meniscus of the same patient (see Figs. 8(a) and (b)). Technically, the acquired details could also be reused for different patients. Finally, the proposed strategy can also serve as a first basis for handling multiple tools, as illustrated in Fig. 8(c). However, in the current formulation results are only plausible when the tools are not in close proximity; otherwise stamp interferences might occur. This issue will be addressed in future work.



(a) (b) (c)

Figure 8: *Lateral meniscus with transferred examples (a) originally generated for a medial meniscus that is now ruptured next to an example (b). To some extent, even multiple tools can be supported (c).*

6.6. Captured Pillow

In this experiment, we employ real-world captured deformations and examine the transfer to different geometries. The example data is obtained and modelled using the 3D scanning software *ReconstructMe*[†] and a suitable sensor. Intermediate states of the acquisition and modelling process are illustrated in the top row of Fig. 9. After capturing a partial surface we mark the deformed contact point $\mathbf{p}[i]'$ on the mesh and register the partial scan with the base mesh of the pillow. After generating a displacement map using a standard 3D modeling tool, we compute the descriptors. The site descriptor is simply given by $f_\rho(\mathbf{p}[i])$, while $f_\mathbf{u}$ has to be determined by running a real-time simulation of the pillow object. In the simulation we displace $\mathbf{p}[i]$ to $\mathbf{p}'[i]$. This results in a corresponding base deformation $\mathbf{u}[i]$ that we use to evaluate $\boldsymbol{\mu}[i]$. The final data-driven enrichment simulation of the pillow runs at 14ms per frame. An example deformation with surface details is depicted in the Fig. 9(c). Finally, we also examined the transfer of deformations to different geometries. In the bottom row of Fig. 9 an enriched smaller pillow is shown. Lastly, to demonstrate the versatility of the approach, we show the transfer of captured details to a cubic (pillow) geometry.

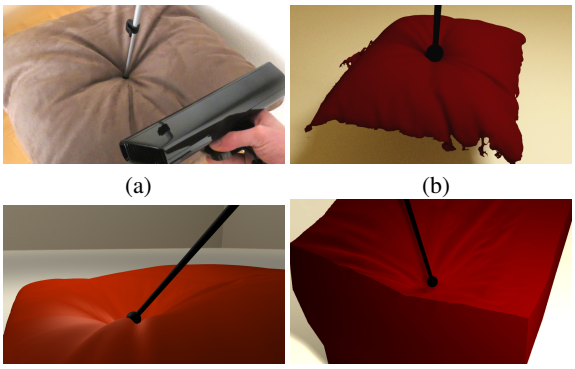


Figure 9: *We scan the surface of a real pillow (a). Note that a partial scan with reasonable quality in the stamping region, is sufficient for our purposes (b). Results at run-time are shown in (c). As a limit case, we apply the pillow examples to a cube. In (d) the enriched result.*

7. Discussion

Our approach is quasi-static and contact point-centric, which is in general a good abstraction for our main application of tool-based surgery simulation. Nevertheless, the technique is less suited for situations where larger areas are in contact or dynamic scenarios. However, our method could be extended to arbitrarily shaped contact regions by adding an additional descriptor based on the Hermitian moments of the contact region's shape.

Theoretically our method can handle multiple boundary conditions since our descriptors are local. However, the support, *e. g.*, for multiple tools, is still rather rudimentary since examples are only acquired with one tool and all other boundary conditions are implicitly present, *e. g.*, all our objects are standing on a surface in their undeformed state. The issue might be resolved by combining our approach with other pre-computation methods, such as [KKN*13].

Finally, the proposed method can also handle topological changes. Due to the mesh topology independence of all involved quantities an object can undergo dynamic modifications, such as cutting or tearing, without impairing our method. In contrast to [SSH14], our method works with local similarities, and as long as locally similar regions exist, our method will simply continue to enrich even cut objects.

8. Conclusion

We have presented an extension to data-driven approaches that enrich coarse simulations with pre-generated example deformations. In contrast to previous work, we allow for the reuse of example deformations at contact points whose local material neighborhood is similar to the one of the original recording location. This is achieved by consequently employing meshfree quantities: First, the Hermitian moment basis represents local quantities, such as material density and displacement, and are used to implement the correlation phase, and second, displacement stamps for surface detail enrichment. Hence, we can effectively transfer examples even between different objects of the same material. As future work, we plan to improve the support for multiple tools.

Acknowledgements The authors would like to thank the anonymous reviewers for their much appreciated comments on earlier versions of this paper and the team at VirtaMed for their continued support. This work has been supported by the Swiss CTI project RapidPRO *12688.1*.

References

[ACP02] ALLEN B., CURLESS B., POPOVIĆ Z.: Articulated body deformation from range scan data. In *ACM Trans. Graphics (TOG)* (2002), vol. 21, ACM, pp. 612–619. 2

[AO05] ALTMANN S. L., ORTIZ E. L.: *Mathematics and Social Utopias in France: Olinde Rodrigues and His Times*, vol. 28. Amer. Mathematical Society, 2005. 4

[BBO*09] BICKEL B., BAECHER M., OTADUY M., MATUSIK W., PFISTER H., GROSS M.: Capture and modeling of nonlinear heterogeneous soft tissue. *Proc. of ACM SIGG. 28*, 3 (2009), 89. 2

[BKS*05] BUSTOS B., KEIM D., SAUPE D., SCHRECK T., VRANIĆ D.: Feature-based similarity search in 3d object databases. *ACM Computing Surveys 37*, 4 (2005), 345–387. 2

[†] http://www.reconstructme.net

[BLB*08] BICKEL B., LANG M., BOTSCH M., OTADUY M. A., GROSS M.: Pose-space animation and transfer of facial details. In *SCA* (2008), pp. 57–66. 2

[BW89] BROIDA J. G., WILLIAMSON S. G.: *A comprehensive introduction to linear algebra*. Addison-Wesley, 1989. 4

[CMT05] CORDIER F., MAGNENAT-THALMANN N.: A data-driven approach for real-time clothes simulation. In *Comp. Graph. Forum* (2005), vol. 24, pp. 173–183. 2

[dASTH10] DE AGUIAR E., SIGAL L., TREUILLE A., HODGINS J. K.: Stable spaces for real-time clothing. *ACM TOG 29*, 3 (2010). 2

[FKY08] FENG W.-W., KIM B.-U., YU Y.: Real-time data driven deformation using kernel canonical correlation analysis. *ACM TOG 27* (August 2008), 91:1–91:9. 2, 6, 7

[FYK10] FENG W.-W., YU Y., KIM B.-U.: A deformation transformer for real-time cloth animation. In *ACM SIGG.* (2010), pp. 108:1–108:9.

[GH99] GUMHOLD S., HUTTNER T.: Multiresolution rendering with displacement mapping. In *SIGG./Eurograph Conference On Graph. Hardw.* (1999), vol. 8, pp. 55–66. 7

[Gra49] GRAD H.: Note on n-dimensional hermite polynomials. *Comm. on Pure and Applied Math. 2*, 4 (1949), 325–330. 4

[HHS08] HOEVER R., HARDERS M., SZEKELY G.: Data-driven haptic rendering of visco-elastic effects. In *Haptic interfaces for virt. env. and teleop. systems, 2008* (2008), pp. 201 –208. 2

[JF03] JAMES D. L., FATAHALIAN K.: *Precomputing interactive dynamic deformable scenes*, vol. 22. ACM, 2003. 2

[KB09] KOLDA T. G., BADER B. W.: Tensor decompositions and applications. *SIAM review 51*, 3 (2009), 455–500. 4

[KGBS11] KAVAN L., GERSZEWSKI D., BARGTEIL A. W., SLOAN P.: Physics-inspired upsampling for cloth simulation in games. In *ACM SIGG.* (2011), pp. 93:1–93:10. 2

[KKN*13] KIM D., KOH W., NARAIN R., FATAHALIAN K., TREUILLE A., O'BRIEN J. F.: Near-exhaustive precomputation of secondary cloth effects. *ACM TOG 32*, 4 (July 2013). Proc. of ACM SIGG. 2013. 9

[KTUI12] KOYAMA Y., TAKAYAMA K., UMETANI N., IGARASHI T.: Real-time example-based elastic deformation. In *Proc. SCA* (2012), pp. 19–24. 2

[KV08] KIM T.-Y., VENDROVSKY E.: Drivenshape: a data-driven approach for shape deformation. In *Proc. of the 2008 SCA* (2008), pp. 49–55. 2

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. ACM SIGG.* (2000), pp. 165–172. 2

[LCHL07] LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: a data-driven approach to crowd simulation. In *Proc. SCA* (2007), pp. 109–118. 2

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM TOG (SIGG.) 24*, 3 (2005), 471–478. 6

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM TOG (SIGG.) 30*, 4 (2011), 72:1–72:8. 2

[MWLT13] MA C., WEI L.-Y., LEFEBVRE S., TONG X.: Dynamic element textures. In *SIGG. 2013* (2013), p. to appear. 7

[NMK*05] NEALEN A., MÜLLER M., KEISER R., BOXERMANN E., CARLSON M.: Physically Based Deformable Models in Comp. Graph. In *EG-STAR* (2005), pp. 71–94. 2

[PB83] PIETRASZKIEWICZ W., BADUR J.: Finite rotations in the description of continuum deformation. *International Journal of Engineering Science 21*, 9 (1983), 1097–1115. 6

[PZB*09] POPA T., ZHOU Q., BRADLEY D., KRAEVOY V., FU H., SHEFFER A., HEIDRICH W.: Wrinkling captured garments using space-time data-driven deformation. *Comp. Graph. Forum (Proc. Eurograph.) 28*, 2 (2009), 427–435. 2

[RK13] RÉMILLARD O., KRY P. G.: Embedded thin shells for wrinkle simulation. vol. 32. 2

[SCER06] SILVÁN-CÁRDENAS J. L., ESCALANTE-RAMÍREZ B.: The multiscale hermite transform for local orientation analysis. *Image Processing, IEEE Tr. 15*, 5 (2006), 1236–1253. 4

[SH13] SIANOV A., HARDERS M.: Data-driven haptics: Addressing inhomogeneities and computational formulation. In *World Haptics Conference* (2013), pp. 301–306. 2

[SSH12] SEILER M., SPILLMANN J., HARDERS M.: Enriching Coarse Interactive Elastic Objects with High-Resolution Data-Driven Deformations. In *SCA* (2012), pp. 9–17. 1, 2, 3, 7

[SSH14] SEILER M., SPILLMANN J., HARDERS M.: Data-driven simulation of detailed surface deformations for surgery training simulators. *IEEE TVCG* (2014), 1–1. 1, 2, 7, 8, 9

[Ste56] STEINHAUS H.: Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci 1* (1956), 801–804. 7

[TV04] TANGELDER J., VELTKAMP R.: A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004. Proc.* (2004), IEEE, pp. 145–156. 2

[YLZD11] YANG B., LI G., ZHANG H., DAI M.: Rotation and translation invariants of gaussian–hermite moments. *Pattern recognition letters 32*, 9 (2011), 1283–1298. 4

[ZBO12] ZURDO J., BRITO J., OTADUY M.: Animating wrinkles by example on non-skinned cloth. *Visualization and Comp. Graph., IEEE Trans. PP*, 99 (2012), 1. 2

Appendix A

The generalized dot product is a coefficient-wise multiplication followed by summation of as many dimensions as available. It allows for simple linear mappings from a tensor $\mathbf{T} = [T_{i_1 \dots i_N}]$ of order N using a tensor $\mathbf{S} = [S_{i_1 \dots i_M}]$ of order M. The resulting tensor $\mathbf{R} = [R_{i_1 \dots i_{|M-N|}}] = \mathbf{T} \bullet \mathbf{S}$ is of order $|M-N|$, *e. g.*, in the case of $M = N$ a scalar which is the expected result of a dot product. In the case of $M \geq N$ it is

$$R_{i_1 \dots i_{M-N}} = \sum_{j_1,\dots,j_N} T_{i_1 \dots i_{M-N} j_1 \dots j_N} S_{j_1 \dots j_N}. \qquad (17)$$

Appendix B

Basis transformation of tensors is required to transform descriptors into the stamping frame $\mathbf{B}$ with the transformation $\mathbf{T} = [T_{ij}] \in \mathbb{R}^{3\times 3}$. For an n-th-order tensor $\mathbf{G} = [G_{j_1 \dots j_n}] \in \mathbb{R}^{3^n}$ the transformation to $\mathbf{G}' = [G'_{i_1 \dots i_n}] \in \mathbb{R}^{3^n}$ is given by

$$G'_{i_1 \dots i_n} = \sum_{j_1,\dots,j_n} G_{j_1 \dots j_n} T_{i_1 j_1} \dots T_{i_n j_n}. \qquad (18)$$