Modeling of Personalized Anatomy using Plastic Strains

BOHAN WANG, University of Southern California, USA
GEORGE MATCUK, University of Southern California, USA
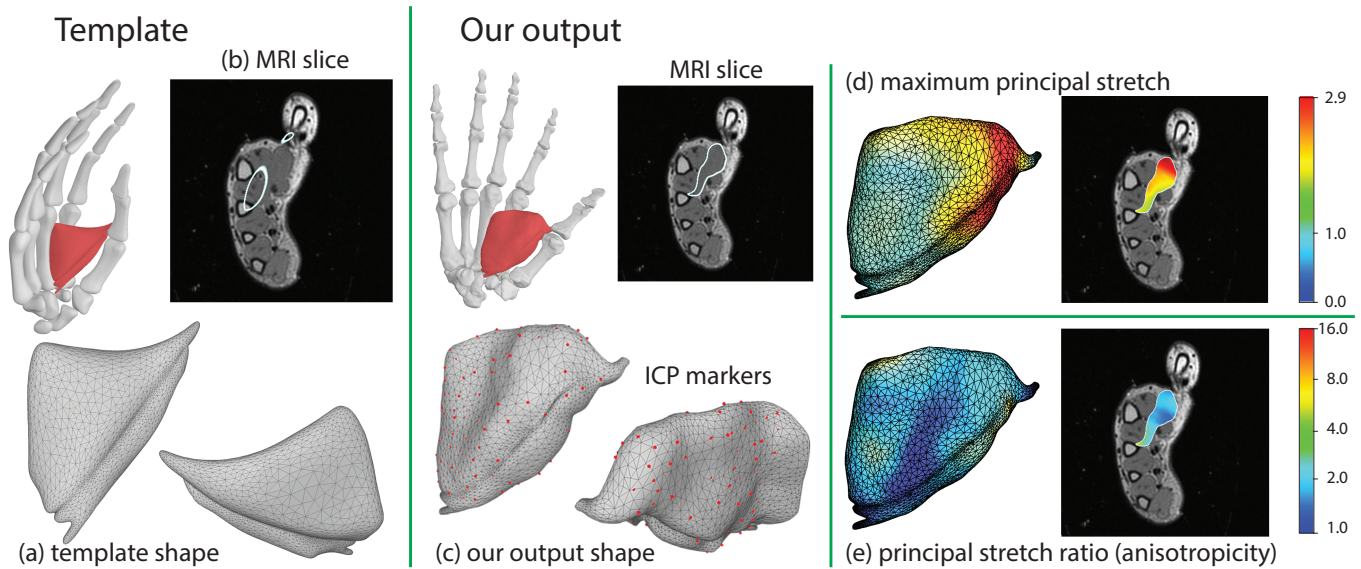JERNEJ BARBIČ, University of Southern California, USA

Fig. 1. **We optimized the shape of this hand muscle (Adductor Pollicis) to match the MRI scan.** (a) template shape [C. Erolin 2019]; (b) representative MRI slice [Wang et al. 2019]; we rigidly aligned the template mesh onto the markers in the MRI scan, producing the white contour which is obviously incorrect (deeply penetrates the bone; and extends out of the volume of the MRI-scanned hand); (c) our output shape, optimized to the MRI scan; the white contour now matches the scan; (d,e) large anisotropic spatially varying strains accommodated by our method (d: maximum principal stretch; e: ratio between maximum and minimum principal stretch).

We give a method for modeling solid objects undergoing large spatially varying and/or anisotropic strains, and use it to reconstruct human anatomy from medical images. Our novel shape deformation method uses plastic strains and the Finite Element Method to successfully model shapes undergoing large and/or anisotropic strains, specified by sparse point constraints on the boundary of the object. We extensively compare our method to standard second-order shape deformation methods, variational methods and surface-based methods and demonstrate that our method avoids the spikiness, wiggliness and other artefacts of previous methods. We demonstrate how to perform such shape deformation both for attached and un-attached ("free flying") objects, using a novel method to solve linear systems with singular matrices with a known nullspace. While our method is applicable to general large-strain shape deformation modeling, we use it to create personalized 3D triangle and volumetric meshes of human organs, based on MRI or CT scans. Given a medically accurate anatomy template of a generic individual, we optimize the geometry of the organ to match the MRI or CT scan of a specific individual. Our examples include human hand muscles, a liver, a hip bone, and a gluteus medius muscle ("hip abductor").

CCS Concepts: • **Computing methodologies** → **Shape modeling**; **Physical simulation**.

Additional Key Words and Phrases: shape deformation, large strain, FEM, plastic strain, optimization, MRI, CT, anatomy

ACM Reference Format:

Authors' addresses: Bohan Wang, University of Southern California, Los Angeles, CA, USA, bohanwan@usc.edu; George Matcuk, University of Southern California, Los Angeles, CA, USA, matcuk@usc.edu; Jernej Barbič, University of Southern California, Los Angeles, CA, USA, jnb@usc.edu.

1 INTRODUCTION

Modeling and simulating human anatomy is very important in many applications in computer graphics, animation, medicine, film and real-time systems such as games and virtual reality. In this paper, we demonstrate how to model anatomically realistic *personalized* three-dimensional shapes of human organs, based on medical images of a real person, such as Magnetic Resonance Imaging (MRI)

or Computed Tomography (CT). Such modeling is crucially important for personalized medicine. For example, after scanning the patient with an MRI or CT scanner, doctors can use the resulting 3D meshes to perform pre-operative surgery planning. Such models are also a starting point for anatomically based human simulation for applications in computer graphics, animation and virtual reality. Constructing volumetric meshes that match an organ in a medical image can also help with building volumetric correspondences between multiple MRI or CT scans of the same person [Rhee et al. 2011], e.g., for medical education purposes.

Although the types, number and function of organs in the human body are generally the same for any human, the shape of each individual organ varies greatly from person to person, due to the natural variation across the human population. The shape variation is substantial: any two individuals' organs $\Omega_1 \subset \mathbb{R}^3$ and $\Omega_2 \subset \mathbb{R}^3$ generally vary by a non-trivial shape deformation function $\Phi : \Omega_1 \rightarrow \Omega_2$ that often consists of large and spatially varying anisotropic strains (see Figure 1). By "large anisotropic strain", we mean that the singular values of the 3x3 gradient matrix of Φ are both different to each other and substantially different from 1.0, i.e., the material locally stretches (or compresses) by large amounts; and this amount is different in different directions and varies spatially across the model.

We tackle the problem of how to model such large shape variations, using volumetric 3D medical imaging (such as MRI or CT scan), and a new shape deformation method capable of modeling large spatially varying anisotropic strains. We note that the boundary between the different organs in medical images is often blurry. For example, in an MRI of a human hand, the muscles often "blend" into each other and into fat without clear boundaries; a CT scan has even less contrast. We therefore manually select as many reliable points ("markers") as possible on the boundary of the organ in the medical image; some with correspondence ("landmark constraints") to the same anatomical landmark in the template organ, and some without ("ICP constraints"). Given a template volumetric mesh of an organ of a generic individual, a medical image of the same organ of a new individual, and a set of landmark and ICP (Iterative Closest Point) constraints, our paper asks how to deform the template mesh to match the medical image.

Our first attempt to solve this problem was to use standard shape deformation methods commonly used in computer graphics, such as as-rigid-as-possible energy (ARAP) [Sorkine and Alexa 2007], bounded biharmonic weights (BBW) [Jacobson et al. 2011], biharmonic weights with linear precision (LBW) [Wang et al. 2015a], and a finite element method static solver (FEM) [Barbič et al. 2009]. As shown in Section 3.8, none of these methods was able to capture the large strains observed in medical images. Namely, these standard methods either cannot model point constraints when the shape undergoes large spatially varying strains, or introduce excessive curvature. For example, in the limit where the tetrahedral mesh is refined to finer and finer tets, the FEM static solver produces spikes (Figure 2, Appendix G). We explore limitations of other methods in Section 3.8.

We give a new shape deformation method that uses plastic strains and the Finite Element Method to successfully model shapes undergoing large and/or anisotropic strains, controlled by the sparse

(a) rest shape   (b) 195 tets   (c) 1,346 tets

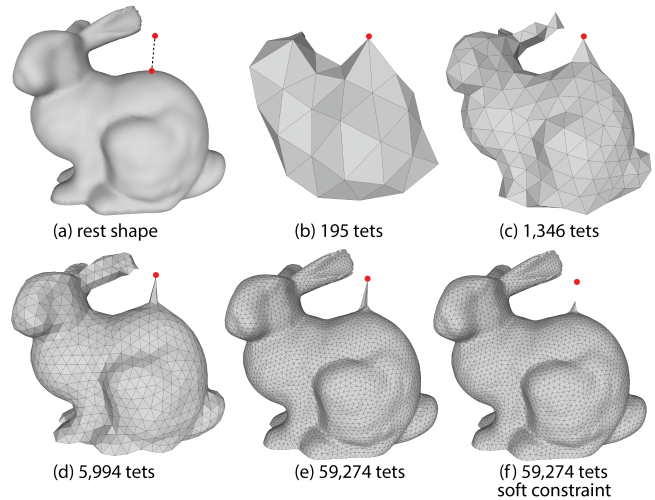(d) 5,994 tets   (e) 59,274 tets   (f) 59,274 tets soft constraint

Fig. 2. **Second-order methods produce spiky outputs as the tet mesh is refined.** Here, we show the output of an FEM static solver under a single hard point constraint (seen in (a)), i.e., we minimize the FEM elastic energy under the shown hard point constraint. Bunny is fixed at the bottom. Poissons's ratio is 0.49. As we increase the tet mesh resolution in (b)-(e), the spike becomes progressively narrower, which is undesirable. Changing the elastic stiffness (Young's modulus) of the bunny does not help (see Appendix G). Converting the constraint into a soft spring constraint also does not help (f); now, the constraint is not even satisfied.

landmark and ICP point constraints on the boundary of the object. In order to do so, we formulate a nonlinear optimization problem for the unknown *plastic deformation gradients* of the template shape Ω_1, such that under these gradients, the shape Ω_1 transforms into a shape that matches the medical image landmark and ICP constraints. The ICP constraints are handled by properly incorporating the ICP algorithm into our method. We note that in solid mechanics, plastic deformation gradients are a natural tool to model large volumetric shape variations. We are, however, unaware of any prior work that has used plastic deformation gradients and the Finite Element Method to model large-strain shape deformation.

In order to make our method work, we needed to overcome several numerical obstacles. The large-strain shape optimization problem is highly nonlinear and cannot be reliably solved with off-the-shelf optimizers such as the interior point method [Artelys 2019]. Furthermore, a naive solution requires solving large dense linear systems of equations. We demonstrate how to adapt the Gauss-Newton optimization method to robustly and efficiently solve our shape deformation problem, and how to numerically avoid dense linear systems. In order to optimize our shapes, we needed to derive analytical gradients of internal elastic forces and the tangent stiffness matrix with respect to the plastic strain, which will be useful for further work on using plasticity for optimization and design of 3D objects. In addition, we address objects that are attached to other objects, such as a hand muscle attached to one or more bones; as well as un-attached objects. An example of an un-attached object is a liver, where the attachments to the surrounding tissue certainly exist, but are not easy to define. It is practically easier to just model the liver as an un-attached object. In order to address un-attached

objects, we give a novel numerical method to solve linear systems with singular matrices with a known nullspace. Such linear systems are commonly encountered in applications in geometric shape modeling and nonlinear elastic simulation. Our examples include human hand muscles, a liver, a hip bone and a hip abductor muscle ("gluteus medius"), all of which undergo substantial and non-trivial shape change between the template and the medical image.

2 RELATED WORK

In this section, we introduce closely related work and discuss the relationship to our work.

Geometric shape modeling. Geometric shape modeling is an important topic in computer graphics research; e.g., see the Botsch and Sorkine [Botsch and Sorkine 2008] survey and the SIGGRAPH course notes by Alexa et al. [Alexa et al. 2006]. Popular methods include variational methods [Botsch and Kobbelt 2004], Laplacian surface editing [Sorkine et al. 2004], as-rigid-as-possible (ARAP) deformation [Igarashi et al. 2005; Sorkine and Alexa 2007], coupled prisms [Botsch et al. 2006] and partition-of-unity methods such as bounded biharmonic weights (BBW) [Jacobson et al. 2011] and biharmonic weights with linear precision [Wang et al. 2015a]; we provide a comparison in Section 3.8 and in several other Figures in the paper. Our method reconstructs the surface shape from a set of un-oriented point observations; this goal is similar to variational implicit surface methods [Huang et al. 2019; Turk and O'Brien 1999]; we give a comparison in Section 4. Point clouds can also be used to optimize rest shapes [Twigg and Kačić-Alesić 2011] and material properties of 3D solids [Wang et al. 2015b]. Such a method cannot be applied to our problem because it assumes a 4D dense point cloud input; whereas we assume 3D sparse point inputs as commonly encountered in medical imaging. Point constraint artefacts of second-order methods can be addressed using spatial averaging [Bergou et al. 2007; Kavan et al. 2011]; however this requires specifying the averaging functions (often by hand) and, by the nature of averaging, causes the constraints to be met only approximately. Our method can meet the constraints very closely (under 0.5 mm error in our examples), i.e., in the precision range of the medical scanners.

Plasticity. Elastoplastic simulations are widely used in computer animation. O'Brien et al. [2002] and Muller and Gross [2004] used an additive plasticity formulation, whereas Irving et al. [Irving et al. 2004] presented a multiplicative formulation and argued that it is better for handling large plasticity; we adopt multiplicative formulation in our work. The multiplicative model was used in many subsequent publications to simulate plasticity, e.g., [Bargteil et al. 2007; Chen et al. 2018; Stomakhin et al. 2013]. Because plasticity models shapes that undergo permanent and large deformation, it is in principle a natural choice also for geometric shape modeling. However, such an application is not straightforward: an incorrect choice of the optimization energy will produce degenerate outputs, elastoplastic simulations in equilibrium lead to linear systems with singular matrices, optimization requires second-order derivatives for fast convergence, and easily produces large linear systems with dense matrices. We present a solution to these obstacles. To the best

of our knowledge, we are the first paper to present such a comprehensive approach for using plasticity for geometric shape modeling with large and anisotropic strains.

Anatomically based simulation. Anatomically based simulation of the human body has been explored in multiple publications. For example, researchers simulated human facial muscles [Sifakis et al. 2005], the entire upper human body [Lee et al. 2009], volumetric muscles for motion control [Lee et al. 2018] and hand bones and soft tissue [Wang et al. 2019]. Anatomically based simulation is also popular in film industry [Tissue 2013]. Existing papers largely simulate generic humans because it is not easy to create accurate anatomy personalized to each specific person. Our method can provide such an input anatomy, based on a medical image of any specific new individual.

Medical image registration. Deformable models are widely used in medical image analysis [McInerney and Terzopoulos 2008]. Extracting quality anatomy geometry from medical images is difficult. For example, Sifakis and Fedkiw [2005] reported that it took them "six months" (including implementing the tools) to extract the facial muscles from the visible human dataset [U.S. National Library of Medicine 1994], and even with the tools implemented it would still take "two weeks". With our tools, we are able to extract all the 17 muscles of the human hand in 1 day (including computer and user-interaction time). Bones generally have good contrast against the surrounding tissue and can be segmented using active contour methods [Székely et al. 1996] or Laplacian-based segmentation [Grady 2006; Wang et al. 2019]. For bones, it is therefore generally possible to obtain a "dense" set of boundary points in the medical image. Gilles et al. [Gilles et al. 2010] used this to deform template skeleton models to match a subject-specific MRI scan and posture. They used the ARAP energy and deformed surface meshes. In contrast, we give a method that is suitable for soft tissues where the image contrast is often low (our hand muscles and liver examples) and that accommodates volumetric meshes and large volumetric scaling variations between the template and the subject. If one assumes that the template mesh comes with a registered MRI scan (or if one manually creates a template mesh that matches a MRI scan), musculoskeletal reshaping becomes more defined because one can now use the pair of MRI images, namely the template and target, to aid with reshaping the template mesh [Gilles and Magnenat-Thalmann 2010; Gilles et al. 2006; Schmid et al. 2009]. The examples in these papers demonstrate non-trivial musculoskeletal reshaping involving translation and spatially varying large rotations with a limited amount of volumetric stretching (Figure 14 in [Gilles and Magnenat-Thalmann 2010]). This is consistent with their choice of the similarity metric between the template and output shapes: their reshaping energy tries to keep the distance of the output mesh to the medial axis the same as the distance in the template [Gilles and Magnenat-Thalmann 2010], which biases the output against volume growth. For bones, a similar idea was also presented in [Schmid et al. 2011; Schmid and Magnenat-Thalmann 2008], where they did not use a medial-axis term to establish similarity to a source mesh, but instead relied on a PCA prior on the shapes of bones, based on a database of 29 hip and femur bone shapes. Our work does not require any pre-existing database of shapes. Because our method

uses plasticity, it can accommodate large and spatially varying volumetric stretching between the template and the subject. We do not need a medical image for the template mesh. We only assume that the template mesh is plausible. Of course, the template mesh itself might have been derived from or inspired by some MRI or CT scan, but there is no requirement that it matches any such scan.

Non-rigid ICP. Non-rigid ICP methods are widely used for deforming the template mesh to a given target surface mesh (or a dense point cloud) [Allen et al. 2003; Amberg et al. 2007; Li et al. 2008]. To apply ICP to medical imaging, one can first delineate the organs in medical images using segmentation techniques, and then deform a template mesh to the segmented surface. In prior work, this has been done by segmenting dense organ boundaries [Gietzen et al. 2019; Niculescu et al. 2009]. Organ boundaries in medical images are often not clear, and therefore segmenting dense boundaries is a laborious process, and even then, the boundaries are often unreliable. Our work only requires identifying a sparse set of boundary points in the medical image. Combining such sparse inputs with ICP methods has not been investigated in prior work. One could apply existing ICP methods to a sparsely sampled boundaries, but this produces suboptimal results under large strains and rotations (Figures 8, 24, 26). It is important to emphasize that the source of errors in previous methods is not the miss-correspondence of sparse markers: suboptimal results are produced even if given perfect correspondence (Figure 26).

Anatomy Transfer. Recently, great progress has been made on anatomically based simulations of humans. Anatomy transfer has been pioneered by Dicko and colleagues [Dicko et al. 2013]. Anatomical muscle growth and shrinkage have been demonstrated in the "computational bodybuilding" work [Saito et al. 2015]. Kadlecek et al. [2016] demonstrated how to transfer simulation-ready anatomy to a novel human, and Ichim et al. [Ichim et al. 2017] gave a state-of-the-art pipeline for anatomical simulation of human faces. Anatomy transfer and a modeling method such as ours are complementary because the former can interpolate known anatomies to new subjects, whereas the latter provides a means to create the anatomies in the first place. Namely, anatomy transfer requires a quality anatomy template to serve as its source, which brings up the question of how one obtains such a template. Human anatomy is both extremely complex for each specific subject, and exhibits large variability in geometry across the population. Accurate templates can therefore only be created by matching them to medical images. Even if one creates such a template, new templates will always be needed to model the anatomical variability across the entire population; and this requires an anatomy modeling method such as ours.

3 SHAPE DEFORMATION WITH LARGE SPATIALLY VARYING STRAINS

Given a template tet mesh of a soft tissue organ for a generic individual, as well as known optional attachments of the organ to other objects, our goal is to deform the tet mesh to match a medical image of the organ of a new individual. We use the term "medical image" everywhere in this paper because this is standard terminology; this does not refer to an actual 2D image, but to the 3D medical volume.
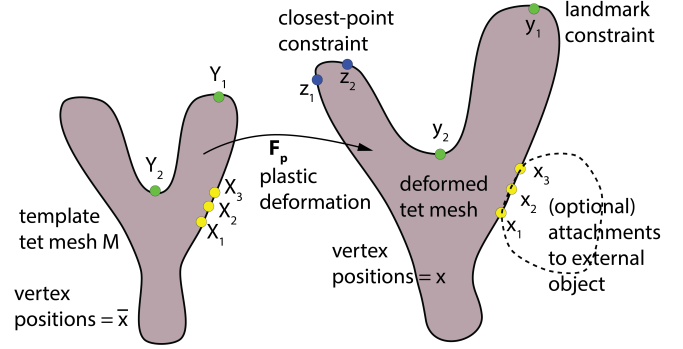

Fig. 3. **Our shape deformation setup.** Our optimization discovers plastic strains F_p and vertex positions x so that the model is in elastic equilibrium under the attachments, while meeting the medical image landmark and closest point constraints as closely as possible. The presence of attachments is optional; our work handles both attached and un-attached objects.

We now describe how we mathematically model the attachments and medical image constraints.

3.1 Attachments and medical image constraints

We start with a template organ tet mesh $\mathcal{M}$. We denote its vertex positions by $\bar{\mathbf{x}} \in \mathbb{R}^{3n}$, where n is the number of tet mesh vertices. In this paper, we use **bold** text to represent the quantities for the entire mesh and non-bold text to represent the quantities for a single vertex or element in the FEM mesh. We would like to discover vertex positions $\mathbf{x} \in \mathbb{R}^{3n}$ such that the organ shape obeys the attachments to other organs (if they exist), and of course the medical image. The attachments are modeled by known material positions $X_i \in \mathcal{M}$, $i = 1, \ldots, t$ that have to be positioned at known world-coordinate positions $x_i \in \mathbb{R}^3, i = 1, \ldots t$ (Figure 3). The medical image constraints come in two flavors. First, there are *landmark constraints* whereby a point on a template organ is manually corresponded to a point in the medical image, based on anatomy knowledge. Namely, landmark constraints are modeled as material positions $Y_i \in \mathcal{M}$, $i = 1, \ldots, q$, that are located at known world-coordinate positions $y_i \in \mathbb{R}^3$ in the medical image. Observe that landmark constraints are mathematically similar to attachments. However, they have a different physical origin: attachments are a physical constraint that is pulling the real-world organ to a known location on another (fixed, un-optimized) object; for example, a muscle is attached to a bone. With landmarks, there is no such physical force in the real-world; namely, landmarks (and also closest-point constraints) are just medical image observations.

The second type of medical image constraints are *closest-point constraints* ("ICP markers"). They are given by known world-coordinate positions $z_i \in \mathbb{R}^3, i = 1, \ldots r$ that have to lie on the surface of the deformed tet mesh. Locations z_i are easier to select in the medical image than the landmarks because there is no need to give any correspondence. As such, they require little or no medical knowledge, and can be easily selected in large numbers. We went through the medical image slices and selected clear representative points on the organ boundary. We then visually compared the template and the target shape inferred by the ICP marker cloud. This guided our positioning of the landmarks, which we place on anatomically "equal"

positions in the template and the medical image. We consulted a medical doctor to help us interpret medical images, such as identifying muscles in the scan, clarify ambiguous muscle boundaries, placing attachments, and disambiguating tendons.

3.2 Plastic deformation gradients

We model shape deformation using plastic deformation gradients, combined with a (small) amount of elastic deformation. In solid mechanics, plasticity is the tool to model large shape variations of objects, making it very suitable for shape deformation with large strains. Unlike using the elastic energy directly (without plasticity), plastic deformations have the advantage that they can arbitrarily and spatially non-uniformly and anisotropically scale the object. There is also no mathematical requirement that they need to respect volume preservation constraints. This makes plastic deformations a powerful tool to model shapes. Our key idea is to find a plastic deformation gradient F_p at each tet of $\mathcal{M}$, such that the FEM equilibrium shape under $\mathbf{F_p}$ and any attachments matches the medical image observations. Figure 3 illustrates our shape deformation setting. In order to do so, we need to discuss the elastic energy and forces in the presence of plastic deformations, which we do next.

Plastic strain is given by a 3×3 matrix F_p at each tetrahedron of $\mathcal{M}$. For each specific deformed shape $\mathbf{x} \in \mathbb{R}^{3n}$, one can define and compute the deformation gradient F between $\bar{\mathbf{x}}$ and $\mathbf{x}$ at each tet [Müller and Gross 2004]. The elastic deformation gradient F_e can then be defined as $F_e = F F_p^{-1}$ [Bargteil et al. 2007] (see Figure 5). Observe that for any shape $\mathbf{x}$, there exists a corresponding plastic deformation gradient $\mathbf{F_p}$ such that $\mathbf{x}$ is the elastic equilibrium under $\mathbf{F_p}$; namely $\mathbf{F_p} = F$. This means that the space of all plastic deformation gradients $\mathbf{F_p}$ is expressive enough to capture all shapes $\mathbf{x}$. The elastic energy of a single tet is defined as

$$\mathcal{E}(F_p, x) = V(F_p)\psi(x, F_p) = V(F_p)\psi\big(F(x)F_p^{-1}\big), \qquad (1)$$

where V is the rest volume of the tet under the plastic deformation F_p, and ψ is the elastic energy density function. We have $V = |F_p|\,V_0$, where V_0 is the tet's volume in $\mathcal{M}$, and $|F_p|$ is the determinant of the matrix F_p. Elastic forces equal $\mathbf{f_e}(\mathbf{F_p}, \mathbf{x}) = d\mathcal{E}(\mathbf{F_p}, \mathbf{x})/d\mathbf{x}$. When solving our optimization problem to compute $\mathbf{F_p}$ in Section 3.4, we will need the first and second derivatives of $\mathcal{E}(F_p, x)$ with respect to x and F_p. We provide their complete derivation in Appendix B.

Our method supports any isotropic hyperelastic energy density function ψ. In our examples, we use the isotropic stable neo-Hookean elastic energy [Smith et al. 2018], because we found it to be stable and sufficient for our examples. Note that we do model anisotropic plastic strains (and this is crucial for our method), so that our models can stretch by different amounts in different directions. Observe that plastic strains are only determined up to a rotation. Namely, let F_p be a plastic strain (we assume $\det(F_p) > 0$; i.e., no mesh inversions), and $F_p = QS$ be the polar decomposition where Q is a rotation and S a 3×3 *symmetric* matrix. Then, F_p and S are the "same" plastic strain: the resulting elastic deformation gradients differ only by a rotation, and hence, due to isotropy of ψ, produce the same elastic energy and elastic forces. Note that it is not required that rotations Q match in any way at adjacent tets. We do not need to even guarantee that F_p globally correspond to

any specific "rest shape", i.e., the F_p are independent of each other and may be inconsistent. This gives plastic deformation gradient modeling a lot of flexibility. Hence, it is sufficient to model plastic strains as symmetric 3×3 matrices. We can therefore model F_p as a symmetric matrix and parameterize it using a vector $s \in \mathbb{R}^6$,

$$F_p = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_4 & s_5 \\ s_3 & s_5 & s_6 \end{bmatrix}. \qquad (2)$$

We model plasticity globally using a vector $\mathbf{s} \in \mathbb{R}^{6m}$, where m is the number of tets in $\mathcal{M}$. We note that Ichim et al. [2017] used such a 6-dimensional parameterization to model facial muscle activations. In our work, we use it for general large-strain shape modeling. Our application and optimization energies are different, e.g, Ichim et al. [2017] causes muscle shapes to follow a prescribed muscle firing field, and biases principal stretches to be close to 1. Furthermore, we address the singularities arising with un-attached objects.

3.3 Shape deformation of attached objects

We now formulate our shape deformation problem. We first do so for attached objects. An object is "attached" if there are sufficient attachment forces to remove all six rigid degrees of freedom, which is generally satisfied if there are at least three attached non-colinear vertices. We find the organ's shape that matches the attachment and the medical image constraints by finding a plastic strain F_p at each tet, as well as static equilibrium tet mesh vertex positions $\mathbf{x}$ under the attachments and plastic strain $\mathbf{F_p}$, so that the medical image observations are met as closely as possible,

$$\underset{\mathbf{s}, \mathbf{x}}{\arg\min} \quad ||\mathbf{L}\,\mathbf{s}||^2 + \alpha\mathcal{E}_{\mathrm{MI}}(\mathbf{x}) + \beta\mathcal{E}_{\mathrm{a}}(\mathbf{x}), \qquad (3)$$

$$\text{subject to:} \quad \mathbf{f_e}\big(\mathbf{F_p}(\mathbf{s}), \mathbf{x}\big) + \mathbf{f_a}(\mathbf{x}) = 0, \qquad (4)$$

where $\alpha \geq 0$ and $\beta \geq 0$ are scalar trade-off weights, and $\mathbf{L}$ is the *plastic strain Laplacian*. We define $\mathbf{L}$ as essentially the tet-mesh Laplacian operator on the tets, 6-expanded to be able to operate on entries of s at each tet (precise definition is in Appendix A). The Laplacian term enforces the smoothness of $\mathbf{F_p}$, i.e., F_p in adjacent tets should be similar to each other. The second equation enforces the elastic equilibrium of the model under plastic strains $\mathbf{F_p}$ and under the attachment forces $\mathbf{f_a}$. Attachments pull a material point embedded into a tet to a fixed world-coordinate location using spring forces; precise definitions of the attachment energy $\mathcal{E}_{\mathrm{a}}$ and attachment forces $\mathbf{f_a}$ are in Appendix F. It is important to emphasize that our output shapes are always in static equilibrium under the plastic strains $\mathbf{F_p}$, and both this equilibrium shape $\mathbf{x}$ and $\mathbf{F_p}$ are optimized together; this is the key aspect of our work. The first equation contains the smoothness and the medical image (MI) observations; we discuss the rationale for using the attachment energy $\mathcal{E}_{\mathrm{a}}$ in Equation 3 in the next paragraph. The medical image energy measures how closely $\mathbf{x}$ matches the medical image constraints,

$$\mathcal{E}_{\mathrm{MI}}(\mathbf{x}) = \sum_{i=1}^{q} ||S\mathbf{x} - y_i||^2 + \sum_{i=1}^{r} ||z_i - \mathrm{closestPoint}(\mathbf{x}, z_i)||^2, \quad (5)$$

where S is the interpolation matrix that selects Y_i, namely $S\bar{x} = Y$. The function $\mathrm{closestPoint}(\mathbf{x}, z_i) \in \mathbb{R}^3$ computes the closest point to $z_i \in \mathbb{R}^3$ on the surface of the tet mesh with vertex positions $\mathbf{x}$.
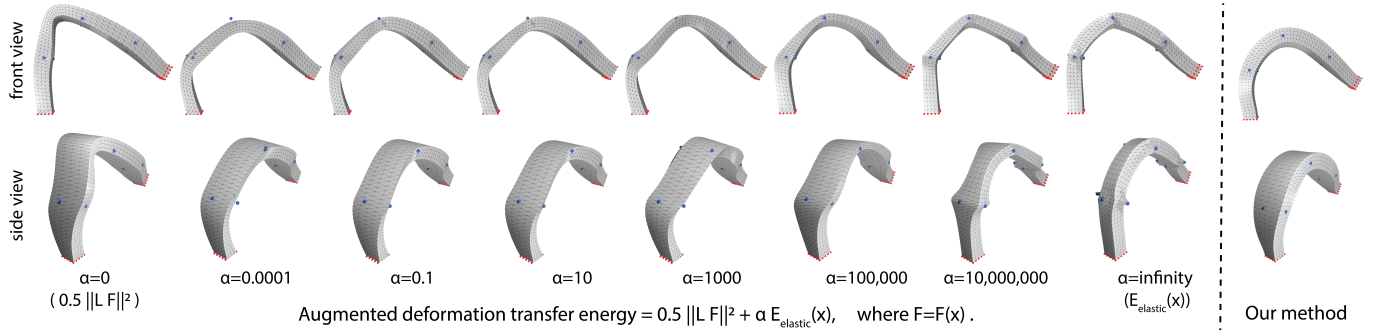
Fig. 4. **Comparison to augmented deformation transfer.** The beam's attachments (red) cause the beam to bend, whereas the ICP markers (blue) cause it to stretch 2x in one of the two transverse directions. Our method can easily recover such a shape deformation, whereas deformation transfer [Sumner and Popović 2004] cannot, even if augmented with an elastic energy.
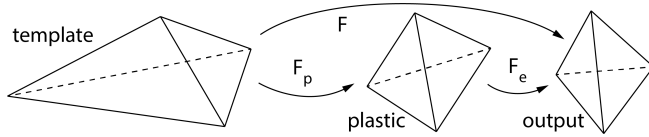

Fig. 5. **Plastic and elastic deformation gradient for a single tet.**

Our treatment of attachments in Equations 3 and 4 deserves a special notice. Equation 4 is consistent with our setup: we are trying to explain the medical images by saying that the organ has undergone a plastic deformation due to the variation between the template and captured individual. The shape observed in the medical image is due to this plastic deformation and the attachments. We formulate attachment forces in Equation 4 as a "soft" constraint, i.e., $\mathbf{f_a}(\mathbf{x})$ is modeled as (relatively stiff) springs pulling the attached organ points to their position on the external object. This soft constraint could in principle be replaced for a hard constraint where the attached positions are enforced exactly. We use soft constraints in our examples because they provide additional control to balance attachments against medical image landmarks and ICP markers. These inputs are always somewhat inconsistent because it is impossible to place them at perfectly correct anatomical locations, due to medical imaging errors. Hence, it is useful to have some leeway in adjusting the trade-off between satisfying each constraint type. With soft constraints, it is important to keep the spring coefficient in $\mathbf{f_a}(\mathbf{x})$ high so that constraints are met very closely (under 0.5 mm error in our examples). We were able to do this using some small amount of spring coefficient tuning.

As per the attachment energy $\mathcal{E}_a$, we initially tried solving the optimization problem of Equations 3 and 4 without it. This seems natural, but actually did not work. Namely, without $\mathcal{E}_a$, there is nothing in Equations 3 and 4 that forces the plastic strains to reasonable values. The optimizer is free to set $\mathbf{F_p}$ to arbitrarily extreme values, and then find a static equilibrium $\mathbf{x}$ under the attachment forces. In our outputs, we would see smooth nearly tet-collapsing plastic strains that result in a static equilibrium $\mathbf{x}$ whereby the medical image constraints were nearly perfectly satisfied. Obviously, this is not a desired outcome. Our first idea was to add a term that penalizes the elastic energy $\mathscr{E}(\mathbf{F_p}, \mathbf{x})$ to Equation 3. Although this worked in simple cases, it makes the expression in Equation 3 generally nonlinear. Instead, we opted for a simpler and more easily

computable alternative, namely add the elastic spring energy of all attachments, $\mathcal{E}_a$. This keeps the expression in Equation 3 quadratic in $\mathbf{x}$ and $\mathbf{F_p}$, which we exploit in Section 3.4 for speed. Observe that $\mathcal{E}_a$ behaves similarly to the elastic energy: if the plastic strain causes a rest shape that is far from the attachment targets, then both $\mathcal{E}_a$ and the elastic energy will need to "work" to bring the shape x to its target attachments. Similarly, if the plastic strain already did most of the work and brought the organ close to its target, then neither $\mathcal{E}_a$ nor the elastic energy will need to activate much.

Because our units are meters and we aim to satisfy constraints closely, we typically use weights close to $\alpha = 10^9$ and $\beta = 10^8$ in our examples. The weights α and β permit adjusting the trade-off between three desiderata: make plastic strains smooth, meet medical image observations, and avoid using too much elastic energy (i.e., prefer to resolve shapes with plastic strains).

Finally, we note that our formulation is different to approaches that optimize the deformation gradient F directly (i.e., without an intermediary quantity such as the plastic deformation gradient). In Figures 4 and 7, we compare to two such approaches: deformation transfer [Sumner and Popović 2004] and variational shape modeling [Botsch and Kobbelt 2004]. We demonstrate that our method better captures shapes defined using our inputs (landmarks, ICP markers, large spatially varying strains). Among all compared approaches, the variational method in Figure 7 came closest to meeting our constraints, but there is still a visual difference to our method. We provide a further comparison to variational methods in Section 4.5.

3.4 Solving the optimization problem for attached objects

We adapt the Gauss-Newton method [Sifakis et al. 2005] to efficiently solve the optimization problem of Equations 3 and 4 (example output shown in Figure 6). Although the Gauss-Newton method is commonly used to solve nonlinear optimization problems, the dimension of our optimization space is several hundred thousands; contrast this to recent related work in computer graphics [Miguel et al. 2012; Sifakis et al. 2005] where the dimension is usually less than 50. In our case, a direct application of the Gauss-Newton method will result in large dense matrices that are costly to compute and store, causing the method to fail on complex examples. Below, we demonstrate
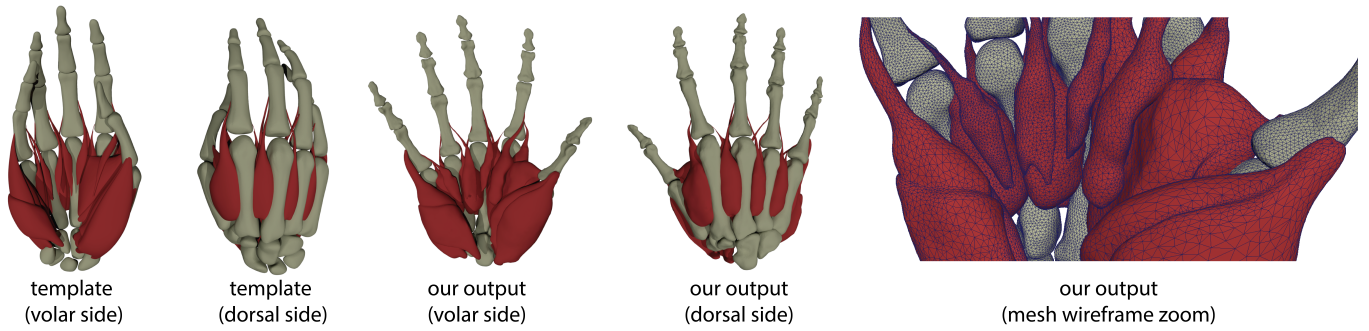
Fig. 6. **17 muscles of the human hand extracted from MRI.** Observe that the template hand is bigger than the scanned hand. Pose is also different. Our method solves this using bone attachments.
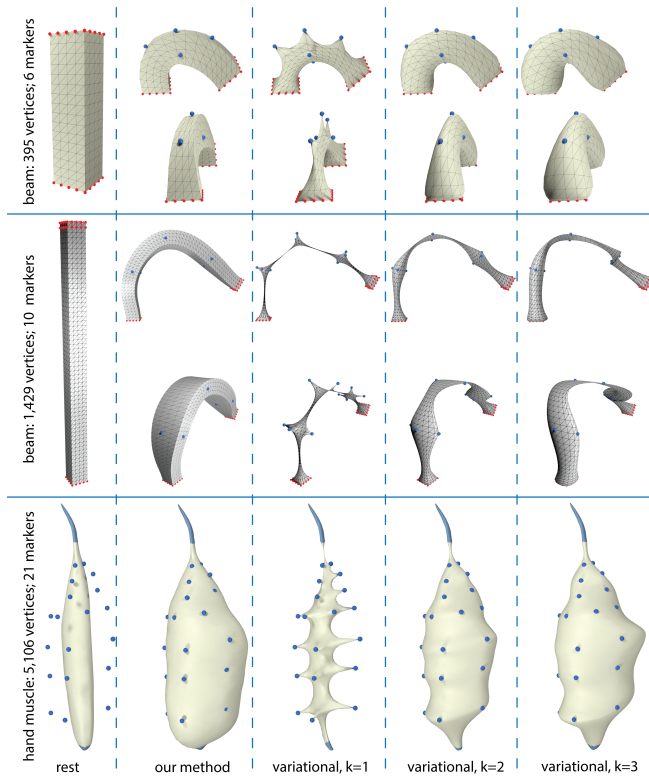

Fig. 7. **Comparison to variational shape modeling.** In a variational method [Botsch and Kobbelt 2004], the wiggles increase if one imposes a stricter constraint satisfaction. First row: under a small number of landmarks, variational methods with $k = 2, 3$ produce a smooth and reasonable result, albeit somewhat smoothening the rest shape. Middle row: under more landmarks, it becomes more difficult for variational methods to meet the landmarks while avoiding the wiggles. Bottom row: variational methods produce wavy results. Our method meets the landmarks and produces fewer wiggles. This is because the plastic deformation field can arbitrarily rotate and non-uniformly scale to adapt to the inputs; the elastic energy then finally irons out the kinks.

how to avoid these issues, producing a robust method capable of handling geometrically complex examples involving complex spatially varying plastic strains. Before settling on our specific Gauss-Newton

approach, we attempted to use the interior-point optimizer available in the state-of-the-art Knitro optimization library [Artelys 2019]. This did not work well because our problem is highly nonlinear. The interior-point method (IPM) worked well on simple examples, but was slow and not convergent on complex examples. IPM fails because it requires the constraint Hessian, which is not easily available (because it involves the third derivative of the elastic energy). When we approximated it, IPM generated intermediate states too far from the constraints, and failed. The strength of our Gauss-Newton approach is that we only need constraint gradients. Our method inherits the convergence properties of the Gauss-Newton method. While not guaranteed to be locally convergent, Gauss-Newton is widely used because its convergence can approach quadratic when close to the solution.

We note that our method is designed for sparse medical image landmarks and ICP markers. In Figure 8, we give a comparison to a related method from medical imaging which used an elastic energy, but with dense correspondences. Our method can produce a quality shape even under sparse inputs, and can consequently work even with coarser MRI scans (such as our hip bone example; Figure 13). The ability to work with sparse markers also translates to lower manual processing time to select the markers in the medical image.

Because the object is attached, Equation 4 implicitly defines $\mathbf{x}$ as a function of $\mathbf{s}$. The Gauss-Newton method uses the Jacobian $\mathbf{J} = d\mathbf{x}/d\mathbf{s}$, which models the change in the static equilibrium $\mathbf{x}$ as one changes the plastic deformation gradient. It eventually relies on the derivative of elastic forces with respect to the plastic deformation gradient, which we give in Appendix B. Although closestPoint($\mathbf{x}, z_i$) is a nonlinear function of $\mathbf{x}$, we can treat the barycentric coordinates of the closest point to z_i as fixed during one ICP iteration, and therefore closestPoint($\mathbf{x}, z_i$) becomes a linear function of $\mathbf{x}$. Therefore, $\mathcal{E}_{\mathrm{MI}}(\mathbf{x})$ can be seen as a quadratic function of $\mathbf{x}$, and so is $\mathcal{E}_{\mathrm{a}}(\mathbf{x})$. We can thus rewrite Equations 3 and 4 as

$$\arg\min_{\mathbf{x}, \mathbf{s}} \frac{1}{2} ||\mathbf{L}\mathbf{s}||^2 + \sum_{k=1}^{q+r+t} \frac{c_k}{2} ||\mathbf{A_k}\mathbf{x} + \mathbf{b_k}||^2, \qquad (6)$$

$$\text{s.t.} \quad \mathbf{f}_{\text{net}}(\mathbf{s}, \mathbf{x}) = \mathbf{0}, \qquad (7)$$

where $\mathbf{f}_{\text{net}}(\mathbf{s}, \mathbf{x}) = \mathbf{f}_e\big(\mathbf{F_p}(\mathbf{s}), \mathbf{x}\big) + \mathbf{f}_a(\mathbf{x})$ is the net force on the mesh, and constant matrices, vectors and scalars $\mathbf{A_k}, \mathbf{b_k}, \mathbf{c_k}$ are independent of $\mathbf{s}$ and $\mathbf{x}$ (we give them in Appendix F). The integer t denotes

3,547 markers (all vertices)	1,867 markers (~1/2 # vertices)	944 markers (~1/4 # vertices)	475 markers (~1/8 # vertices)	236 markers (~1/16 # vertices)	99 markers (~1/32 # vertices)	99 markers

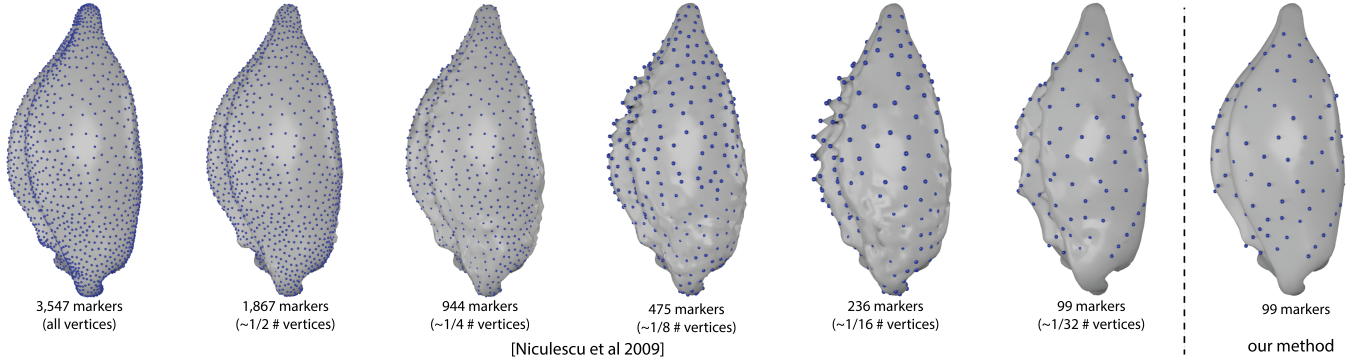[Niculescu et al 2009]   our method

Fig. 8. **Elastic energy methods only work with dense markers.** In this figure, we compare to a state-of-the art medical imaging technique [Niculescu et al. 2009], whereby the output shape is calculated by minimizing an elastic energy of a template shape, subject to dense medical image markers. With dense markers, elastic energy methods work well (left). As the constraints sparsify, elastic energy produces artefacts (middle). Our plasticity method (right) produces a good shape even with sparse markers.

the number of attachments. We now re-write Equations 6 and 7 so that the plastic strains are expressed as $\mathbf{s} + \Delta\mathbf{s}$, and the equilibrium $\mathbf{x}$ as $\mathbf{x} + \Delta\mathbf{x}$, where $\Delta\mathbf{x} = \mathbf{J}\Delta\mathbf{s}$. At iteration i of our Gauss-Newton method, given the previous iterates $\mathbf{x}^i$ and $\mathbf{s}^i$, we minimize a non-linearly constrained problem,

$$\underset{\mathbf{x}^{i+1},\Delta\mathbf{s}^i}{\arg\min} \frac{1}{2}||\mathbf{L}(\mathbf{s}^i + \Delta\mathbf{s}^i)||^2 + \sum_{k=1}^{q+r+t} \frac{\mathbf{c_k}}{2}||\mathbf{A_k}\left(\mathbf{x}^i + \mathbf{J}\Delta\mathbf{s}^i\right) + \mathbf{b_k}||^2, \quad (8)$$

$$\text{s.t.} \quad \mathbf{f}_{\text{net}}(\mathbf{s}^i + \Delta\mathbf{s}^i, \mathbf{x}^{i+1}) = \mathbf{0}. \quad (9)$$

After each iteration, we update $\mathbf{s}^{i+1} = \mathbf{s}^i + \Delta\mathbf{s}^i$. Observe that Equation 8 does not depend on $\mathbf{x}^{i+1}$, and that the constraint of Equation 9 is already differentially "baked" into Equation 8 via $\Delta\mathbf{x} = \mathbf{J}\Delta\mathbf{s}$. We therefore first minimize Equation 8 for $\Delta\mathbf{s}^i$, using unconstrained minimization; call the solution $\overline{\Delta\mathbf{s}^i}$. A naive minimization requires solving a large dense linear system of equations, which we avoid using the technique presented at the end of this section. We regularize $\overline{\Delta\mathbf{s}^i}$ so that the corresponding F_p is always positive-definite for each tet; we do this by performing eigen-decomposition of the symmetric matrix F_p at each tet, and clamping any negative eigenvalues to a small positive value (we use 0.01). Our method typically did not need to perform clamping in practice, and in fact such clamping is usually a sign that the method is numerically diverging, and should be restarted with better parameter values.

We then minimize the optimization problem of Equations 8 and 9 using a 1D line search, using the search direction $\overline{\Delta\mathbf{s}^i}$. Specifically, for $\eta \geq 0$, we first solve Equation 7 with $\mathbf{s}(\eta) := \mathbf{s}^i + \eta\overline{\Delta\mathbf{s}^i}$ for $\mathbf{x} = \mathbf{x}(\eta)$ using the Knitro library [Artelys 2019]. Direct solutions using a Newton-Raphson solver also worked, but we found Knitro to be faster. We then evaluate the objective of Equation 6 at $\mathbf{x} = \mathbf{x}(\eta)$ and $\mathbf{s} = \mathbf{s}(\eta)$. We perform the 1D line search for the optimal η using the gradient-free 1D Brent's method [Press et al. 2007].

Initial guess: We solve our optimization problem by first assuming a constant s at each tet, starting from the template mesh as the initial guess. This roughly positions, rotates and globally scales the template mesh to match the medical image. We use the output as the initial guess for our full optimization as described above.
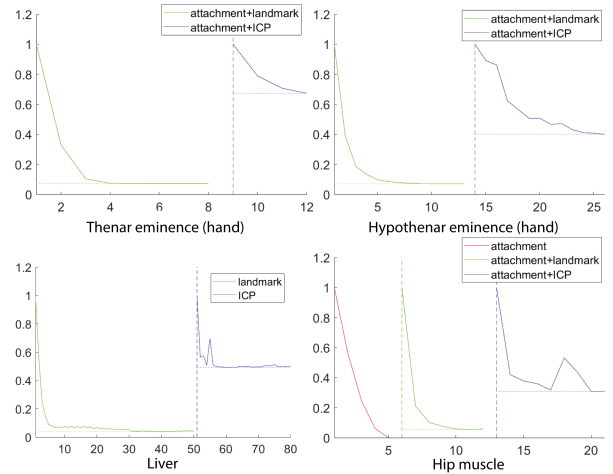

Fig. 9. **Convergence plots.** X-axis are iterations, and Y-axis is the optimization energy. The initial optimization energies are normalized to 1.0.

Optimization stages and stopping criteria: We first do the optimization with attachments only. Upon convergence, we add the landmarks, ignoring any ICP markers. This is because initially, the mesh is far away from the target and the ICP closest locations are unreliable. After convergence, we disable the landmarks and enable the ICP markers and continue optimizing. After this optimization meets a stopping criterium, we are done. Our output is therefore computed with ICP markers only; landmarks only serve to guide the optimizer. This is because landmarks require a correct correspondence, and it is harder to mark this correspondence reliably in the scan than to simply select an ICP marker on the boundary of an organ. We recompute the closest locations to ICP markers after each Gauss-Newton iteration. We stop the optimization if either of the following three criteria is satisfied: (i) reached the user-specified maximal number of iterations (typically 20; but was as high as 80 in the liver example), (ii) maximum error at ICP markers is less than a user-specified value (1mm for hand muscles), (iii) the progress in each iteration is too small, determined by checking if η is under

a user-defined threshold (we use 0.01). Figure 9 shows the convergence of our optimization.

Avoiding dense large linear systems. Because the object is attached, $\frac{\partial \mathbf{f}_{net}}{\partial \mathbf{x}}$ is square and invertible. Therefore, one can obtain a formula for $\mathbf{J}$ by differentiating Equation 7 with respect to $\mathbf{s}$,

$$\mathbf{J} = -\left(\frac{\partial \mathbf{f}_{net}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{f}_{net}}{\partial \mathbf{s}}. \tag{10}$$

The matrix $\mathbf{J}$ is dense (dimensions $3n \times 6m$). Observe that because Equation 8 is quadratic in $\Delta \mathbf{s}^i$, minimizing it as done above to determine the search direction $\overline{\Delta \mathbf{s}^i}$ is equivalent to solving a linear system with the system matrix $\mathbf{H}$, where $\mathbf{H}$ is the second derivative (Hessian matrix; dimension $6m \times 6m$) of Equation 8 with respect to $\Delta \mathbf{s}^i$. Because $\mathbf{J}$ is dense, $\mathbf{H}$ is likewise a dense matrix,

$$\mathbf{H} = \mathbf{L}^2 + \mathbf{J}^T \Big(\sum_k c_k \mathbf{A_k}^T \mathbf{A_k}\Big)\mathbf{J} = \mathbf{L}^2 + \mathbf{Z}^T \mathbf{Z}, \quad \text{where} \tag{11}$$

$$\mathbf{Z} = \begin{bmatrix} \sqrt{c_1}\mathbf{A_1} \\ \sqrt{c_2}\mathbf{A_2} \\ \vdots \\ \sqrt{c_{q+r+t}}\mathbf{A_{q+r+t}} \end{bmatrix} \mathbf{J} = -\begin{bmatrix} \sqrt{c_1}\mathbf{A_1} \\ \sqrt{c_2}\mathbf{A_2} \\ \vdots \\ \sqrt{c_{q+r+t}}\mathbf{A_{q+r+t}} \end{bmatrix} \left(\frac{\partial \mathbf{f}_{net}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{f}_{net}}{\partial \mathbf{s}}. \tag{12}$$

Therefore, when the number of tet mesh elements m is large, it is not practically possible to compute $\mathbf{H}$, store it explicitly in memory or solve linear systems with it. To avoid this problem, we first tried solving the system of equations using the Conjugate Gradient (CG) method. This worked, but was very slow (Table 1). The matrix $\mathbf{Z} \in \mathbb{R}^{3(q+r+t) \times 6m}$ is dense. In our complex examples, the number of medical image constraints $q + r + t$ is small (typically 10 - 800) compared to the dimension of $\mathbf{s}$ ($6m$; typically ~200,000). Our idea is to efficiently compute the solution to a system $\mathbf{H}y = h$ for any right-hand side h using the Woodbury matrix identity [Woodbury 1950], where we view $\mathbf{L}^2$ as a "base" matrix and $\mathbf{Z}^T\mathbf{Z}$ a low-rank perturbation. Before we can apply Woodbury's identity, we need to ensure that the base matrix is invertible. As we prove in Appendix A, the plastic strain Laplacian $\mathbf{L}$ is singular with six orthonormal vectors ψ_i in its nullspace (assuming that $\mathcal{M}$ is connected). Each ψ_i is a vector of all ones in component i of s, $i = 1, \ldots, 6$ and all zeros elsewhere, divided by $\sqrt{m}$ for normalization. It follows from the Singular Lemma (i) (Section 3.5) that $\mathbf{L}^2$ is also singular with the same nullspace vectors. Therefore, we decompose

$$\mathbf{H} = \Big(\mathbf{L}^2 - \sum_{i=1}^{6} \psi_i \psi_i^T\Big) + \Big(\mathbf{Z}^T\mathbf{Z} + \sum_{i=1}^{6} \psi_i \psi_i^T\Big) = \mathbf{B} + \hat{\mathbf{Z}}^T\hat{\mathbf{Z}}, \tag{13}$$

where $\mathbf{B} = \mathbf{L}^2 - \sum_{i=1}^{6} \psi_i \psi_i^T$ and $\hat{\mathbf{Z}}$ is matrix $\mathbf{Z}$ with an additional added 6 added rows ψ_i^T. By the Singular Lemma (iii) (Section 3.5), $\mathbf{B}$ is now invertible, and we can use Woodbury's identity to solve

$$y = \mathbf{H}^{-1}h = \Big(\mathbf{B}^{-1} - \mathbf{B}^{-1}\hat{\mathbf{Z}}^T \left(I + \hat{\mathbf{Z}}\mathbf{B}^{-1}\hat{\mathbf{Z}}^T\right)^{-1} \hat{\mathbf{Z}}\mathbf{B}^{-1}\Big)h. \tag{14}$$

We rapidly compute $\mathbf{Z}$, without ever computing or forming $\mathbf{J}$, by solving sparse systems $\frac{\partial \mathbf{f}_{net}}{\partial \mathbf{x}} \tilde{z}_k = -\sqrt{c_k}\mathbf{A_k}^T$, and $z_k = \tilde{z}_k^T \frac{\partial \mathbf{f}_{net}}{\partial \mathbf{s}}$, for $k = 1, \ldots, (q+r+t)$, where $z_k \in \mathbb{R}^{3 \times 6m}$ is the k-th 3-row block of $\mathbf{Z}$. Observe that this sparse system matrix is symmetric and the same for all k. We factor it once using the Pardiso solver and then solve the

Table 1. **Solving a single linear system of equations with H,** using the conjugate gradients and our method. The naive direct solver failed in all cases. Note that $\mathbf{H}$ is a dense $6m \times 6m$ matrix. The column t_{prep} gives a common pre-processing time for both CG and our method.

Example	$6m$	$3(q+r+t)$	t_{prep}	CG	Ours
Hand muscle	237,954	1,143	17.5s	897.5s	9.5s
Hip bone	172,440	1,497	11.4s	408.9s	7.2s
Liver	259,326	1,272	24.4s	1486.7s	10.0s

multiple right-hand sides in parallel. Once $\tilde{z}_k^T$ has been computed, $z_k = \tilde{z}_k^T \frac{\partial \mathbf{f}_{net}}{\partial \mathbf{s}}$ is easy to compute because it is a multiplication of a thin dense matrix and a sparse matrix. The matrix $\mathbf{B}$ is constant, and we only need to factor it once for the entire optimization. Finally, the matrix $I + \hat{\mathbf{Z}}\mathbf{B}^{-1}\hat{\mathbf{Z}}^T \in \mathbb{R}^{3(q+r+t) \times 3(q+r+t)}$ is small, and so inverting it is fast. We analyze the performance of our algorithm in Table 1.

3.5 Singular lemma

In this paper, there are two occasions where we have to solve a singular sparse linear system with known nullspace vectors. Such systems occur often in modeling of un-attached objects, e.g., finding static equilibria, solving Laplace equations on the object's mesh, animating with rotation-strain coordinates [Huang et al. 2011], or computing modal derivatives [Barbič and James 2005]. Previous work solved such systems ad-hoc, and the underlying theory has not been stated or developed in any great detail; or it only addressed non-singular systems. For example, in order to constrain vertices in cloth simulation, Boxerman [2003] modified the linear system $Ax = b$ to $SAx = Sb$, for a properly chosen filtering matrix S. However, A is non-singular to begin with; there is no discussion of singularity in Boxerman's work.

We hereby state and prove a lemma that comprehensively surveys the common situations arising with singular systems in computer animation and simulation, and back the lemma with a mathematical proof (Appendix C). Recall that the *nullspace* of a matrix $A \in \mathbb{R}^{p \times p}$ is $\mathcal{N}(A) = \{x \in \mathbb{R}^p \, ; \, Ax = 0\}$, and the *range* of A is $\mathcal{R}(A) = \{Ax \, ; \, x \in \mathbb{R}^p\}$. Both are linear vector subspaces of $\mathbb{R}^p$.

Singular Lemma: Let the square symmetric matrix $A \in \mathbb{R}^{p \times p}$ be singular with a known nullspace spanned by k linearly independent vectors $\psi_1, \ldots, \psi_k$. Then the following statements hold:
(i) $\mathcal{N}(A)$ and $\mathcal{R}(A)$ are orthogonal. Every vector $x \in \mathbb{R}^p$ can be uniquely expressed as $x = n + r$, where $n \in \mathcal{N}(A)$ and $r \in \mathcal{R}(A)$. Vector r is orthogonal to n and to ψ_i for all $i = 1, \ldots, k$ (Figure 10).

Fig. 10. **Illustration of the Singular Lemma.**

(ii) Let $b \in \mathcal{R}(A)$. Then, the singular system $Ax = b$ has a unique solution x with the property that x is orthogonal to ψ_i for all $i = 1, \ldots, k$. This solution can be found by solving the *non-singular*
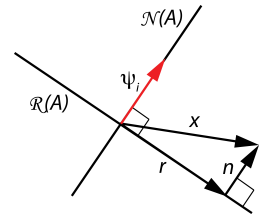
linear system

$$
\begin{bmatrix} A & \psi_1 & \dots & \psi_k \\ \psi_1^T & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \psi_k^T & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda_1 \\ \vdots \\ \lambda_k \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{15}
$$

All other solutions equal $x + \sum_{i=1}^k \mu_i \psi_i$ for some scalars $\mu_i \in \mathbb{R}$. (iii) For any scalars $\alpha_i \neq 0$, the matrix $B = A + \sum_{i=1}^k \alpha_i \psi_i \psi_i^T$ is invertible. If ψ_i are orthonormal vectors, then the solution to $By = h$ equals $y = x + \sum_{i=1}^k \frac{\lambda_i}{\alpha_i} \psi_i$, where x and λ_i are solutions to Equation 15 with $b = \text{proj}_{\mathcal{R}(A)} h = h - \sum_{i=1}^k (\psi_i^T h) \psi_i$. We give the proof of the singular lemma in Appendix C.

3.6 Un-attached objects

The difficulty with un-attached objects is that we now have $\mathbf{f}_{\text{net}} = \mathbf{f}_e$, and the equation $\mathbf{f}_e(\mathbf{F_p}(\mathbf{s}), \mathbf{x}) = 0$ no longer has a unique solution $\mathbf{x}$ for a fixed plastic state $\mathbf{s}$. This can be intuitively easily understood: one can arbitrarily translate and rotate any elastic equilibrium shape $\mathbf{x}$ under the given plastic state $\mathbf{s}$; doing so produces another elastic equilibrium shape. The space of solutions $\mathbf{x}$ is 6-dimensional. This means that we can no longer uniquely solve Equation 7 for $\mathbf{x}$ during our line search of Section 3.4. Furthermore, the square tangent stiffness matrix

$$
\mathbf{K}(\mathbf{F_p}(\mathbf{s}), \mathbf{x}) = \frac{\partial \mathbf{f}_e(\mathbf{F_p}(\mathbf{s}), \mathbf{x})}{\partial \mathbf{x}} \tag{16}
$$

is no longer full rank. In order to address this, we now state and prove the following Nullspace Lemma.

Nullspace Lemma: The nullspace of the tangent stiffness matrix of an elastoplastic deformable object in *static equilibrium* $\mathbf{x}$ under plasticity, is 6-dimensional. The six nullspace vectors are $\psi_i :=$ $[e_i, e_i, \dots, e_i]$, where $e_i \in \mathbb{R}^3$ is the i-th standard basis vector, and $\psi_{3+i} := [e_i \times x_1, e_i \times x_2, \dots, e_i \times x_n]$, for $i = 1, 2, 3$.

To the best of our knowledge, this fact of elasto-plasto-statics has not been stated or proven in prior work. It is very useful when modeling large-deformation elastoplasticity, as real objects are often un-attached, or attachments cannot be easily modeled. We give a proof in Appendix D. To accommodate un-attached objects, it is therefore necessary to stabilize the translation and rotation. For translations, this could be achieved easily by fixing the position of any chosen vertex. Matters are not so easy for rotations, however. Our idea is to constrain the centroid of all tet mesh vertices to a specific given position t, and to constrain the "average rotation" of the model to a specific given rotation R. We achieve this using the familiar "shape-matching" [Müller et al. 2005], by imposing that the rotation in the polar decomposition of the global covariance matrix

must be R. We therefore solve the following optimization problem,

$$
\underset{\mathbf{x}, \mathbf{s}, R, t}{\arg\min} \ ||\mathbf{L}\,\mathbf{s}||^2 + \alpha \mathcal{E}_{\text{MI}}(\mathbf{x}) \tag{17}
$$

$$
\text{s.t.} \quad \mathbf{f}_e(\mathbf{F_p}(\mathbf{s}), \mathbf{x}) = 0, \tag{18}
$$

$$
\left(\sum_{j \in D} w_j x_j \right) - \sum_{j \in D} w_j X_j = t, \tag{19}
$$

$$
\text{Polar}\left(\sum_{j \in D} w_j \left(x_j - t \right) \left(X_j - \left(\sum_{k \in D} w_k X_k \right) \right)^T \right) = R, \tag{20}
$$

where D is the set of points on the mesh surface where we have either a landmark or an ICP constraint, w_j is the weight of a point, X_j is the position of vertex j in $\mathcal{M}$ and $\text{Polar}(F)$ is the polar decomposition function that extracts the rotational part of a matrix F. We set all weights equal, i.e., $w_j = 1/|D|$. We choose the set D as opposed to all mesh vertices so that we can easily perform optimization with respect to R and t (next paragraph). We assume that our argument matrices F to Polar are not inversions, i.e., $\det(F) > 0$, which establishes that $\text{Polar}(F)$ is always a rotation and not a mirror. This requirement was easily satisfied in our examples, and is essentially determined by the medical imaging constraints; the case $\det(F) < 0$ would correspond to an inverted (or mirror) medical image, which we exclude.

We solve the optimization problem of Equations 17, 18, 19 and 20 using a block-coordinate descent, by iteratively optimizing $\mathbf{x}, \mathbf{s}$ while keeping R, t fixed and vice-versa (Figure 11, 12). Rigid transformations do not affect smoothness of $\mathbf{s}$ so we do not need to consider it when optimizing R, t. We need to perform two modifications to our Gauss-Newton iteration of Section 3.4. The first modification is that we need to simultaneously solve Equations 18, 19 and 20 when determining the static equilibrium in the current plastic state $\mathbf{s}$. As with attached objects, we do this using the Knitro optimizer. In order to do this, we need to compute the first and second derivatives of the stabilization constraints in Equations 19 and 20 (Section 3.7). The second modification is needed because the tangent stiffness matrix $\mathbf{K}(\mathbf{F_p}(\mathbf{s}), \mathbf{x})$, as explained above, is now singular with a known 6-dimensional nullspace. In order to compute the Jacobian matrix $\mathbf{J}$ using Equation 10, we use our Singular Lemma (ii) (Section 3.5). Note that the right-hand side is automatically in the range of $\mathbf{K}$ because Equation 10 was obtained by differentiating a valid equation, hence Equation 10 must also be consistent.

3.7 Gradient and Hessian of Polar(F)

Previous work computed first and second-order *time* derivatives of the rotation matrix in polar decomposition [Barbič and Zhao 2011], or first derivative with respect to each individual entry of F [Chao et al. 2010; Twigg and Kačić-Alesić 2010]. In our work, we need the first *and* second derivatives of R with respect to each individual entry of F. We found an elegant approach to compute them using Sylvester's equation, as follows. Observe that $\text{Polar}(F) = FS^{-1}$, where S is the symmetric matrix in the polar decomposition. Because $\det(F) > 0$, S is positive-definite and uniquely defined as $S = \sqrt{F^T F}$. To compute the first-order derivatives, we start from $F = RS$, and
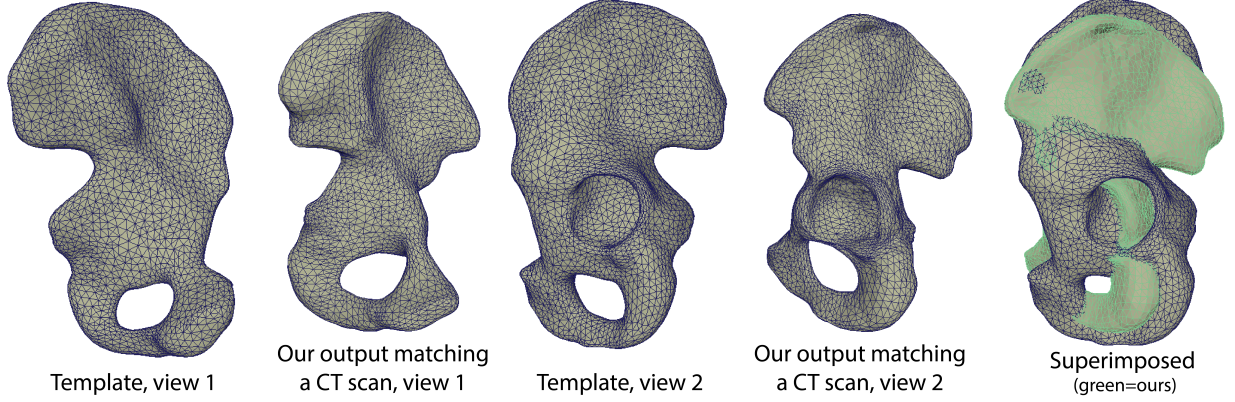
Fig. 11. **Un-attached optimization of a hip bone shape to a CT scan.** The scanned bone is smaller and has a substantially different shape to the template.
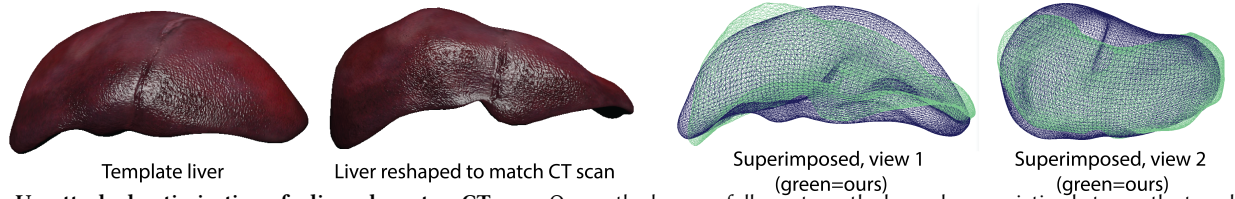

Fig. 12. **Un-attached optimization of a liver shape to a CT scan.** Our method successfully captures the large shape variation between the template and the scan. This figure also demonstrates that our method makes it possible to transfer the rendering textures and uv coordinates from the template onto the output.
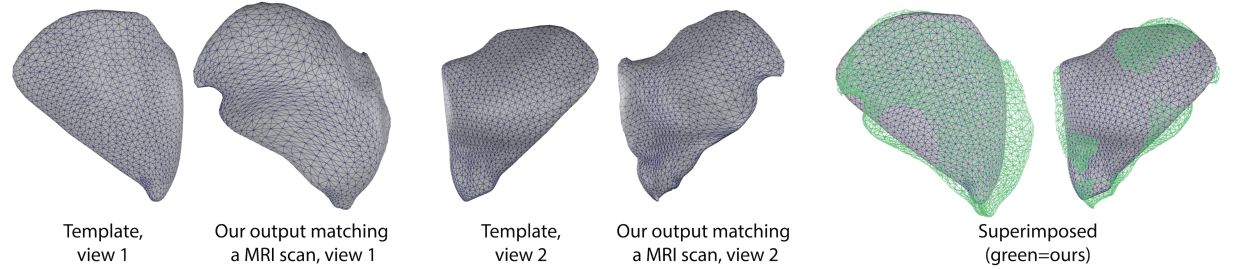

Fig. 13. **Attached optimization of a hip muscle (gluteus medius) to a MRI scan.** Our method successfully captures the large shape variation between the template and the scan.

differentiate,

$$\frac{\partial F}{\partial F_i} = \frac{\partial R}{\partial F_i} S + R \frac{\partial S}{\partial F_i}, \quad \text{hence} \quad \frac{\partial R}{\partial F_i} = \left(\frac{\partial F}{\partial F_i} - R \frac{\partial S}{\partial F_i} \right) S^{-1}, \quad (21)$$

Therefore, we need to compute $\partial S / \partial F_i$. We have

$$F^T F = S^2, \quad \text{and thus} \quad \frac{\partial F^T F}{\partial F_i} = \frac{\partial S}{\partial F_i} S + S \frac{\partial S}{\partial F_i}, \quad (22)$$

i.e., this is the classic Sylvester equation for the unknown matrix $\frac{\partial S}{\partial F_i}$ [Sylvester 1884]. The Sylvester equation $AX + XB = C$ can be solved as

$$\left(B^T \oplus A \right)^{-1} \text{vec}(X) = \text{vec}(C), \quad (23)$$

where $\oplus$ is the Kronecker sum of two matrices. In our case,

$$\text{vec}(\frac{\partial S}{\partial F_i}) = (S \oplus S)^{-1} \text{vec}(\frac{\partial F^T F}{\partial F_i}). \quad (24)$$

The computation of second-order derivatives follows the same recipe: differentiate the polar decomposition and solve a Sylvester equation. We give it in Appendix E.

We can now compute the gradient and Hessian of our stabilization constraints. The translational constraint is linear in **x** and can be expressed as $W_1 \mathbf{x} - d_1 = 0$, where W_1 is a $3 \times 3n$ sparse matrix. Although Polar is not linear, the argument of Polar is linear in **x**. The rotational constraint can be expressed as $\text{Polar}(W_2 \mathbf{x} - d_2) - \bar{R} = 0$, where W_2 is a $9 \times 3n$ sparse matrix. The Jacobian of the translational constraint is W_1, and the Hessian is zero. For the rotational constraint, the Jacobian is $\frac{\partial R}{\partial F} : W_2 \in \mathbb{R}^{9 \times 3n}$ and the Hessian is $(W_2^T : \frac{\partial^2 R}{\partial F^2} : W_2) \in \mathbb{R}^{9 \times 3n \times 3n}$, where : denotes tensor contraction.

3.8 Comparison to standard shape deformation methods

Our shape deformation setup is similar to standard shape deformation problems in computer graphics. In fact, we first attempted to solve the shape deformation problem with as-rigid-as-possible energy (ARAP) [Sorkine and Alexa 2007], bounded biharmonic weights (BBW) [Jacobson et al. 2011], biharmonic weights with linear precision (LBW) [Wang et al. 2015a], and a Finite Element Method static
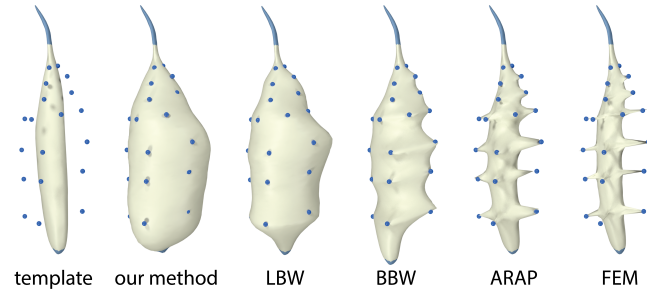
Fig. 14. **Standard volume-based shape deformation methods result in wiggly and spiky artefacts.** The shown hand Palmar interossei muscle has a tendon on one side and no tendon on the other; both ends are attached to a bone (light blue). Definitions of the acronyms are in Section 3.8. The MRI landmark constraints are shown in dark blue. The shape deformation between the template muscle and the shape in MRI consists of large and spatially varying stretches. Our method successfully models this deformation. We note that spikes cannot be avoided by using, say, a spherical region for the constraints as opposed to a point; the non-smoothness just moves to the spherical region boundary.
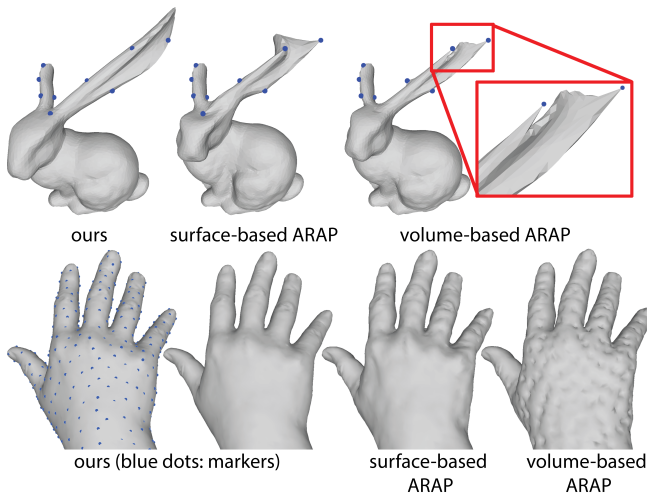

Fig. 15. **Comparison between a surface energy and volumetric energy.** In both examples (bunny and hand), we performed non-rigid iterative closest point alignment between a template triangle mesh, and a collection of target ICP markers (13 for bunny and 456 for the hand). For the bunny, we manually placed the markers to greatly enlarge bunny's left ear. For the hand, we placed the markers on a pre-existing target hand surface mesh that has different geometric proportions and mesh topology as the template. Template is a man and target is a woman. We then solved the ICP problem using the surface-based ARAP energy, volume-based ARAP energy and our volumetric plastic strains. Our method produces smooth artefact-free outputs.

solver (FEM) [Barbič et al. 2009]. Unfortunately, none of the methods worked well. Figures 14 and 15 demonstrate that these methods produce non-smooth shapes with spikes (ARAP, BBW, FEM), or wiggles (LBW).

Mathematically, the reason for the spikes in ARAP, BBW and FEM is that point constraints for second-order methods are inherently flawed. As one refines the solution by adding more tetrahedra, the solution approaches a spiky point function at each point constraint,
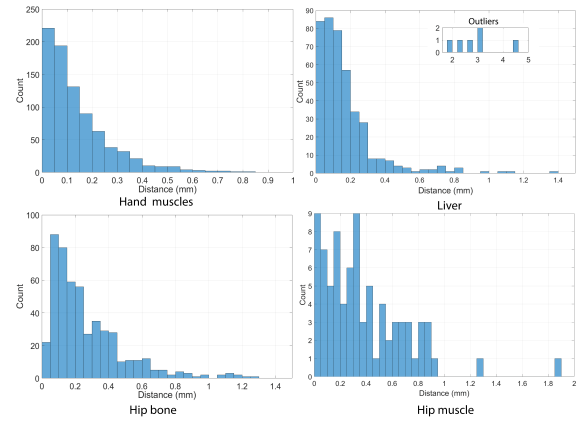

Fig. 16. **Output ICP error histograms.** Each medical image marker contributes 1 entry to the histogram. The hand muscles histogram is a combined histogram for all the 17 hand muscles.

which is obviously not desirable. This mathematical issue is exposed in our work because our shape deformation consists of large spatially varying stretches. Often, the template mesh needs to be stretched ~2x or more along some (or several) coordinate axes. The medical image constraints are distributed all around the muscle, pulling in different directions and essentially requesting the object to undergo a spatially non-uniform and anisotropic scale. This exacerbates the spikiness for second-order methods. We note that these problems cannot be avoided simply by using an elastic energy that permits volume growth. Namely, Drucker's stability condition [Drucker 1957] requires a monotonic elastic energy increase with increase in strain. An elastic energy therefore must penalize strain increases if it is to be stable; and this impedes large-strain modeling in methods that rely purely on an elastic energy. Our plasticity method does not penalize large strains and thus avoids this problem. Spikes can be avoided by using a higher-order variational method such as LBW. However, our experiments indicate that such methods suffer from wiggles when applied to medical imaging problems (see also Figures 7 and 23).

4 RESULTS

We extracted muscles of the human hand and the hip muscle from an MRI scan, and a hip bone and a liver from a CT scan. We analyze the performance of our method in Table 2. In Figure 16, we give histograms of the magnitude of the difference between the positions of the medical image markers and their output positions. It can be seen that our method produces shapes that generally match the medical image constraints to 0.5mm or better. In Figure 17, we demonstrate that the output quality of our tetrahedra is still good; if needed, this could be further improved by re-meshing [Bargteil et al. 2007]. Figures 18 and 19 superimpose our output meshes on the CT and MRI scans, respectively. In Figure 20, we compare to a recent implicit point set surface reconstruction method [Huang et al. 2019]. In Figure 21, we evaluate our method in the presence of known ground truth plastic deformation gradients. Figures 22, 24, 25, 26 provide comparisons to surface-based methods, in addition to Figures 7,15. In all comparisons, our method outperformed surface-based methods.

Table 2. **The statistics for our examples:** #vtx=number of vertices; #ele = number of tetrahedra in $\mathcal{M}$; #iter=number of ICP iterations; "time"=total computation time to compute the output shape; "attached" means whether the object is attached or not; e_{init} = error between our template mesh and the ICP constraints; e_{final} = error between our result and the ICP markers. The first and second reported error numbers are the average and maximum error, respectively. In the hand example, there are 17 groups of muscles; "min", "med" and "max" refers to the smallest, representative median and largest muscle group; "max-m" is the example with the largest number of ICP constraints. Observe that our markers are sparse; the ratio between the number of markers and the number of vertices is 0.3%-7.3% in our examples.

| Example	# vtx	# ele	# markers	# iter	time [min]	attached	$e_{init}[mm]$	$e_{final}[mm]$
Hand muscle (min)	4,912	20,630	15	8	14.2	yes	0.53 / 2.22	0.06 / 0.14
Hand muscle (med)	6,260	31,737	32	12	12.8	yes	0.62 / 1.56	0.11 / 0.55
Hand muscle (max)	8,951	42,969	96	11	14.2	yes	3.35 / 12.82	0.11 / 0.34
Hand muscle (max-m)	7,552	34,966	151	18	20.3	yes	3.28 / 9.11	0.16 / 0.47
Hip muscle (Fig 13)	6,793	34,118	82	21	28.3	yes	7.41 / 21.27	0.39 / 1.85
Hip bone	6,796	28,740	499	34	49.2	no	4.12 / 14.27	0.25 / 1.30
Liver	11,392	43,221	424	80	128.3	no	9.00 / 33.68	0.21 / 4.81
Hand surface (Fig 15)	11,829	49,751	456	31	43.8	no	4.87 / 16.78	0.07 / 0.86
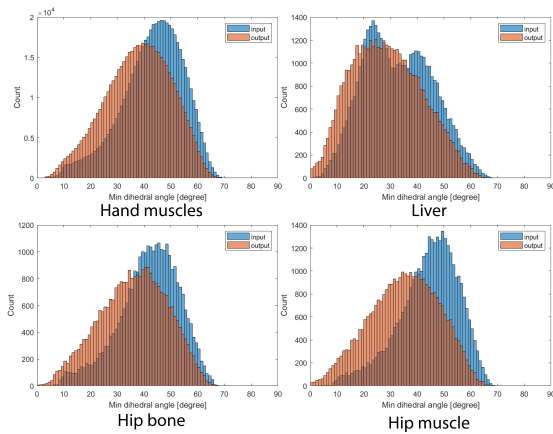
Fig. 17. **Minimum tetrahedral dihedral angles before and after our optimization.** It can be seen that the output angles are still relatively large. As expected, the output angles are somewhat worse than the initial ones, as the object has undergone a plastic deformation.

4.1 Hand muscles

In our muscle hand example, we extracted 17 hand muscles from an MRI scan (Figure 6). We obtained the scan and the already extracted bone meshes from [Wang et al. 2019]; scan resolution is 0.5mm x 0.5mm x 0.5mm . We considered two "templates", the first one from the Centre for Anatomy and Human Identification at the University of Dundee, Scotland [2019], and the second one from Zygote [Zygote 2016]. We used the first one (Figure 6, left) because we found it to be more medically accurate (muscles insert to correct bones). Muscle anatomy of a human hand is challenging (Figure 6). We model all muscle groups of the hand, namely the thenar eminence (thumb), hypothenar eminence (below little finger), interossei muscles (palmar and dorsal) (between metacarpal bones), adductor pollicis (soft tissue next to the thumb, actuating thumb motion), and lumbricals (on the side of the fingers at the fingers base). Our template models the correct number and general location of the muscles, but there are large muscle shape differences between the template subject and the scanned subject (Figure 1). We solve the optimization problem of Equations 3 and 4 separately for each muscle, starting from

the template mesh as the initial guess. In our results, this produces muscles that match the attachments and medical image constraints markers at 0.5 mm or better, which is at, or better than, the accuracy of the MRI scanner.

4.1.1 Marking the muscles in MRI scans. During pre-processing, we manually mark as many reliable points as possible on the boundary of each muscle ($\sim 10 - 20$ landmarks and $\sim 50 - 100$ ICP markers per muscle) in the MRI scans. This process took approximately 5 minutes per muscle.

4.1.2 Attachments to bones. The template muscles are modeled as triangle meshes. We build a tetrahedral mesh for each muscle. Our tet meshes conform to the muscle's surface triangle mesh; this requirement could be relaxed. For each muscle in the template, we attach its tet mesh to the bones using soft constraints. We do this by marking where on one or multiple bones this muscle inserts; to do so, we consulted a medical doctor with specific expertise in anatomy. For each bone triangle mesh vertex that participates in the insertion, we determine the material coordinates (i.e., tet barycentric coordinates) in the closest muscle tet. We then form a soft constraint whereby this muscle material point is linked to the bone vertex position using a spring.

4.1.3 Direct attempt using segmentation: We note that we have also attempted to model the muscle shapes directly using segmentation, simply from an MRI scan. Recent work has demonstrated that this can be done for hand bones [Wang et al. 2019], and we attempted a similar segmentation approach for muscles. However, given that the muscles touch each other in many places (unlike bones), the contrast in the MRI scan was simply not sufficient to discern the individual muscles. Our conclusion is that a segmentation approach is not feasible for hand muscles, and one must use a pre-existing anatomically accurate template as in our work.

4.1.4 Removing inter-penetrations of muscles. Many hand muscles are in close proximity to one another and several are in continuous contact. One strategy to resolve contact would be to incorporate contact into our optimization (Equations 3 and 4). This approach is not very practical, as unilateral constraints are very difficult to optimize. Furthermore, such an approach couples all (or most) muscles, and
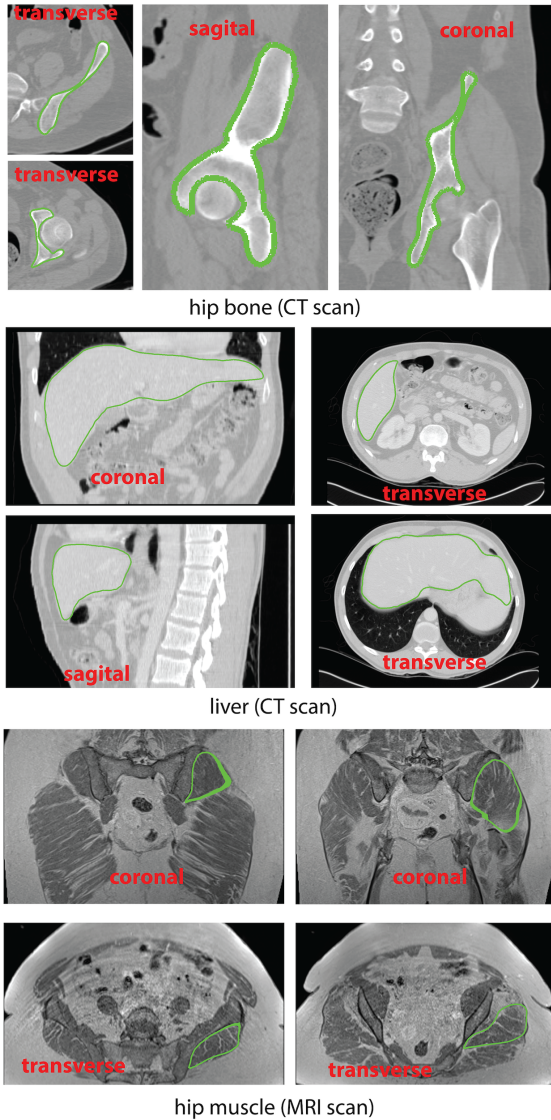
Fig. 18. **Our extracted organs match the medical image.** The intersection of the output mesh with the medical image slices is shown in green.
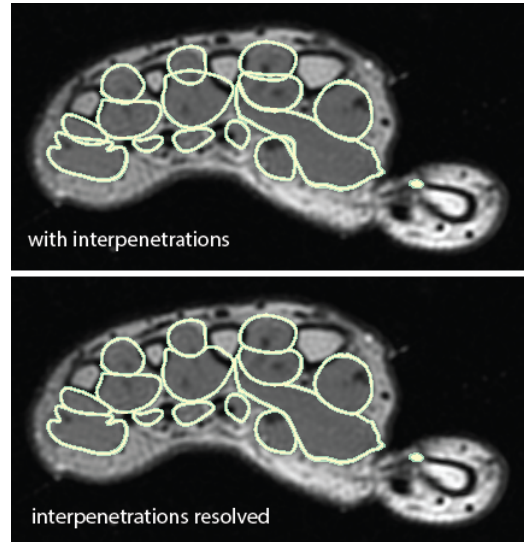

Fig. 19. **Interpenetration removal.** The yellow lines are muscle cross-sections in this representative MRI slice of the hand. It can be seen that our interpenetration removal method successfully removes penetrations, without modifying the interpenetration-free sections of the muscles' boundary.

requires one to solve an optimization problem with a much larger number of degrees of freedom. It is extremely slow at our resolution; it overpowers our machine. Instead, we optimize each muscle separately. Of course, the different muscles inter-penetrate each other, which we resolve as follows. For each muscle, we already positioned the MRI constraints so that they are at the muscle boundary. Therefore, observe that if our marker positioning and the solution to the optimization problem were perfect, inter-penetration would be minimal or non-existent. The marker placement is relatively straightforward at the boundary between a muscle and another tissue (bone, fat, etc.) due to good contrast. However, placing markers at the boundary between two continuously colliding muscles is less precise, due to a lower MRI contrast between adjacent muscles. This is the main cause of the inter-penetrations. We remove inter-penetrations with FEM simulation because it produces smooth organic shape changes; alternatively, one could use geometric approaches [Schmid et al. 2009]. Specifically in our work, for a pair of inter-penetrating muscles, we run collision detection to determine the set of triangles of each muscle that are inside the volume of the other muscle. On each muscle, we then determine the set of tetrahedra that are adjacent to the collision area. We then slightly enlarge this set, by including any tet within a 5-ring of tets. We then run a FEM contact simulation just on these two tetrahedral sets on the two muscles. FEM simulation pushes the muscle surface boundaries apart, without displacing the rest of the muscle (Figure 19). We handle contact islands of multiple muscles by running the above procedure on two muscles, then for a third muscle against the first two muscles, then the fourth against the first three, and so on. Although in principle this procedure is order-dependent, the input penetrations were shallow in our examples, making the procedure relatively unaffected by the processing order.

4.2 Hip bone

In our second anatomical example (Figure 11), we apply our method to a CT scan of the human hip bone (pelvis). We obtained the template from the human anatomy model of Ziva Dynamics [Ziva Dynamics 2019], and the CT scan from the "KidneyFullBody" medical image repository [Stephcavs 2019]. The template and the scanned hip bone differ substantially in shape, and this is successfully captured by our method.

4.3 Liver

In our third anatomical example, we apply our method to a CT scan of the human liver (Figure 12). We purchased a textured liver triangle mesh on TurboSquid [Turbosquid 2019]. We subdivided it and created a tet mesh using TetGen [Hang Si 2011]. This serves as our "template". We used a liver CT scan from the "CHAOS" medical

image repository [Kavur et al. 2019]. We then executed our method to reshape the template tet mesh to match the CT scan. Much like with the hip bone, our method successfully models large differences between the template and the scanned liver. Finally, we embedded the TurboSquid triangle mesh into the template tet mesh, and transformed it with the shape deformation of the tet mesh. This produced a textured liver mesh (Figure 12) that matches the CT scan.

4.4 Hip muscle

In our fourth anatomical example, we apply our method to a MRI scan of a female human hip muscle (gluteus medius) (Figure 13). We obtained the data from The Cancer Imaging Archive (TCIA) [Clark et al. 2013]. The image resolution is $384 \times 384 \times 240$ with voxel spacing of 1mm, which is 2x coarser to the hand MRI dataset. We use the template mesh from the human anatomy model of Ziva Dynamics [Ziva Dynamics 2019]. We subdivided it and created a tet mesh for it using TetGen [Hang Si 2011]. Because the muscle is attached to the hip bone and the leg bone, we needed to first extract the bones from the MRI scan; we followed the method described in [Wang et al. 2019]. Note that the subject in the Ziva Dynamics template is male. The template and the scanned hip muscle differ substantially in shape, and this is successfully captured by our method.
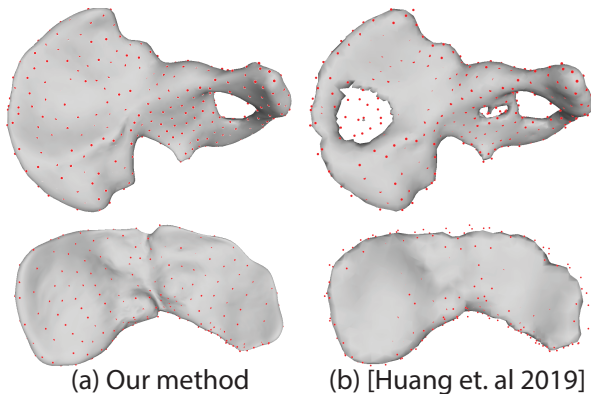

(a) Our method   (b) [Huang et. al 2019]

Fig. 20. **Comparison to [Huang et al. 2019].** Top row: hip bone. Bottom row: liver. Red points are the markers from the CT scan. We used the publicly available implementation of [Huang et al. 2019] to compute the normals, followed by screened Poisson surface reconstruction using the points and computed normals [Kazhdan and Hoppe 2013]. We used this combination because it produced better results than running [Huang et al. 2019] directly. Our method produces shapes that match the ground truth data more closely.

4.5 Comparison to variational methods

We compare our method to variational shape modeling methods on an illustrative 1D example. Note that Figure 7 gave a comparison on 3D muscle geometry. Consider an elastic 1D line segment whose neutral shape is the interval $[0, 1]$, and study its longitudinal 1D deformation under the following setup. Let us prescribe hard attachments whereby we attach endpoint 0 to position 0, and endpoint 1 to position 2. Furthermore, assume landmarks whereby point $1/4$ is at $1/4$, and point $1/2$ is at $3/2$. Effectively in this setup, we are specifying that the subintervals $[0, 1/4]$ and $[1/2, 1]$ do not stretch,
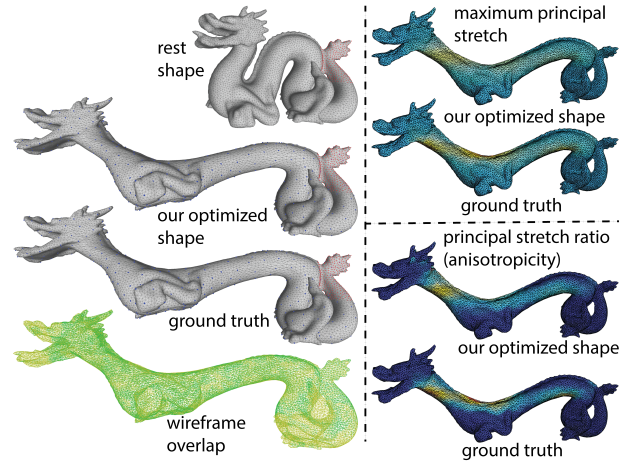

Fig. 21. **Quantitative evaluation on ground truth.** We first performed a dynamic FEM simulation with plasticity [Irving et al. 2004], whereby the back of the dragon is fixed (red), and the head was pulled to the left with uniform force. This produced our ground truth plastic deformation gradients. We then selected 488 sparse triangle mesh vertices as landmarks and ICP markers, and ran our method to compute $\mathbf{F_p}$ and shape $\mathbf{x}$. It can be seen that F_p and x match the ground truth closely.
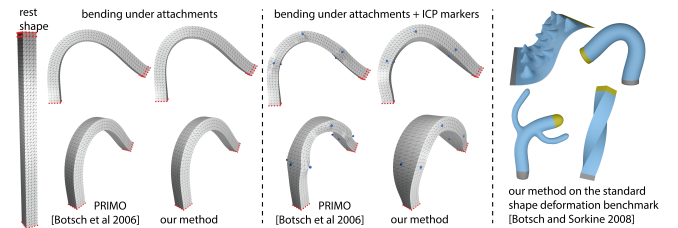

Fig. 22. **Comparison to surface-based methods.** We compare to the "PRIMO" method [Botsch et al. 2006] because this method is a good representative choice. It also has fewest artefacts in Figure 10 of the shape deformation survey [Botsch and Sorkine 2008]. Our methods produces a clearly superior result in the challenging scenario where the ICP markers were placed to anisotropically stretch the beam in one transverse direction. Our method also passes the standard shape deformation benchmark [Botsch and Sorkine 2008].

whereas the subinterval $[1/4, 1/2]$ stretches from its original length of $1/4$ to $5/4$, (5x stretch). A variational formulation of order r is,

$$\min_{x(t) \in C^r} \int_0^1 \left(\frac{d^r x}{dt}\right)^2 dt + \alpha\left(\left(x(\tfrac{1}{4}) - \tfrac{1}{4}\right)^2 + \left(x(\tfrac{1}{2}) - \tfrac{3}{2}\right)^2\right), \quad (25)$$

where C^r denotes all functions $[0, 1] \to \mathbb{R}$ whose derivatives exist and are continuous up to order r. We solved these problems analytically in Mathematica for $r = 1, 2, 3$, each time for 3 representative values α, and compared them (Figure 23) to our method in 1D,

$$\min_{f_p, \, x(t) \in C^r} \int_0^1 \ddot{f}_p^{\,2} dt + \alpha\left(\left(x(\tfrac{1}{4}) - \tfrac{1}{4}\right)^2 + \left(x(\tfrac{1}{2}) - \tfrac{3}{2}\right)^2\right)$$

$$+ \beta \int_0^1 \left(\frac{\dot{x}}{f_p} - 1\right)^2 dt \qquad (26)$$

$$\text{s. t. } x(t) = \arg\min_{x(t) \in C^r} \int_0^1 \left(\frac{\dot{x}}{f_p} - 1\right)^2 dt. \qquad (27)$$

Observe that, in the same vein as in 3D, we can decompose $\dot{x} = f_e f_p$, whereby scalar functions f_e and f_p and the β term are the equivalents of the elastic and plastic deformation gradients, and the elastic energy, respectively. It can be seen that our method produces a better fit to the data and a significantly less wiggly solution, compared to variational methods (Figure 23).
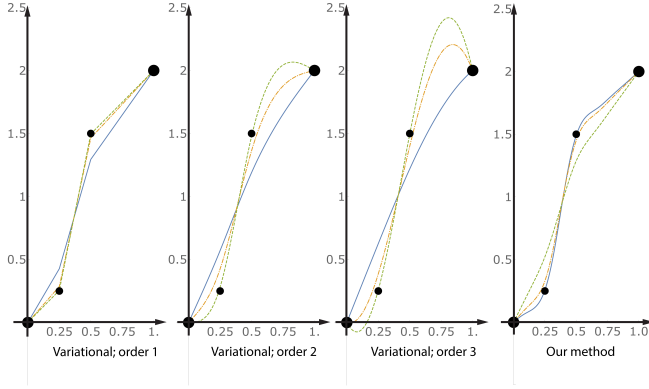

Fig. 23. **Comparison to variational methods.** A 1D string of length 1 is attached on both ends. The left attachment is fixed. The right attachment is moved to coordinate 2. The spring thus stretches longitudinally while trying to obey the two landmarks at $t = 1/4$ and $t = 1/2$. Y-axis gives the deformed 1D position of the corresponding material point on the X-axis. Big and small dots denote the attachments and landmarks, respectively. For each method, we plot the result under a few representative parameters. With variational methods, one has to either give up meeting the landmarks, or the curve becomes wiggly. Similar behavior can also be observed in 3D variational results (Figure 7). Our method produces a deformation profile whereby the slope (total 1D deformation gradient) on all three subintervals $[0, 1/4]$, $[1/4, 1/2]$, $[1/2, 1]$ approximately matches the slope implied by the attachments and landmarks (1, 5, 1, respectively). Note that the shown output curves are only C^{r-1} and not in C^r at the two juncture points $1/4$, $1/2$; however, their r-th derivative is integrable and they are the optimal Cauchy limit of a score-decreasing sequence of curves in C^r.
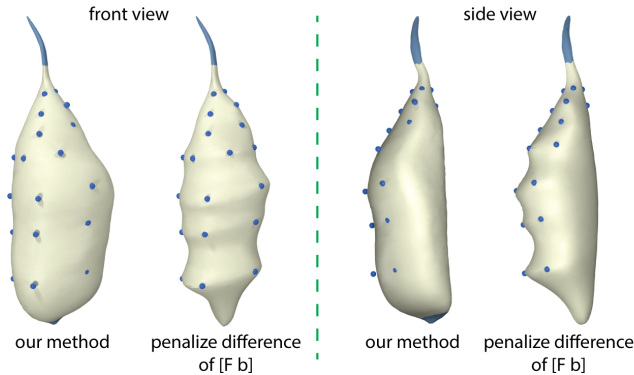
4.6 Comparison to ICP methods

Fig. 24. **Comparison to methods that penalize the difference of neighboring affine transformations** $[F\ b]$. These methods produce a result similar to the variational method and suffer from wiggle artifacts.
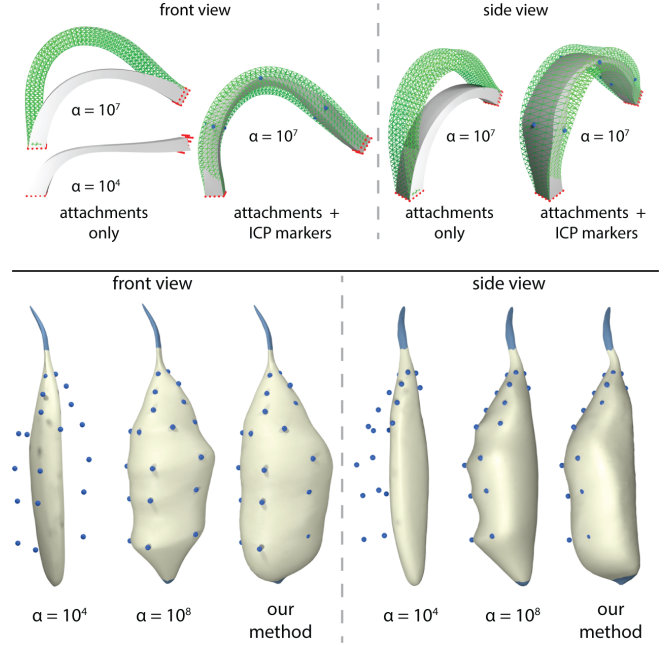
Fig. 25. **Comparison to** E_{bend}**.** We deformed the surface using the ShapeOp library. Here, α is the strength of the marker constraints (attachments and ICP markers) relative to the bending energy. In the beam example, our method is shown in green wireframe, and E_{bend} is shown as solid. Low values of α (such as, for example, $\alpha = 10^4$) cannot satisfy the constrains; and we therefore use $\alpha = 10^7$ and $\alpha = 10^8$ to meet the constraints in the beam and muscle examples, respectively. It can be seen that the E_{bend} method suffers from volume loss in the beam example. In addition, it fails to grow the beam (the middle is collapsed). Wiggle artefacts can be observed in the E_{bend} method on the muscle.

Penalizing the differences of affine transformations between neighboring vertices is one of the commonly used smoothness terms in "classical" ICP algorithms [Allen et al. 2003; Amberg et al. 2007; Li et al. 2008]. Namely, these methods optimize for a 3×4 affine transformation $A = [F\ b]$ ($x \rightarrow Fx + b$) at each vertex, and penalize an energy that is the sum of the differences in the $[F\ b]$ matrices between adjacent vertices, plus the ICP energy. This is similar to variational methods [Botsch and Kobbelt 2004; Sumner and Popović 2004], which penalize the difference in F (as opposed to $[F\ b]$). We provide a comparison on our hand muscle example; results are similar to variational methods (Figure 24). Note that "classical" ICP algorithms assume dense correspondences; they produce artefacts under sparse correspondences (Figure 24).

4.7 Comparison to ShapeOP

As shown by previous work, variational methods suffer from artefacts under large rotations [Botsch and Sorkine 2008]. One can enhance variational methods by adding local rotations to each vertex, producing the energy

$$E_{\text{bend}} = \frac{1}{\sum_i A_i} \sum_i A_i \left\| \Delta \mathbf{x_i} - \mathbf{R_i} \Delta \bar{\mathbf{x_i}} \right\|^2, \tag{28}$$
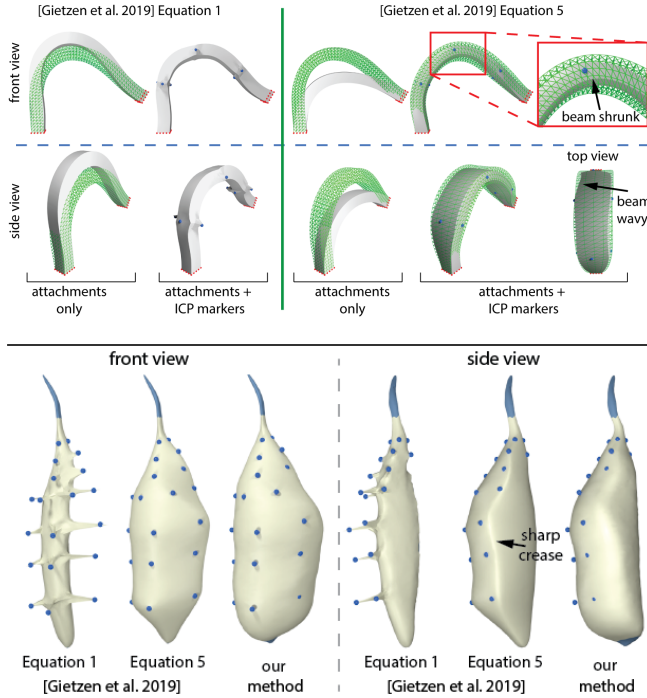
Fig. 26. **Comparison to [Gietzen et al. 2019].** We compare our method to two different deformation energies used in [Gietzen et al. 2019]. Equation 1 penalizes the volume change relative to the previous ICP iteration. Equation 5 penalizes E_{bend} relative to the previous ICP iteration ("incremental plasticity"). On the beam example, our method is shown in green wireframe, whereas the compared method is shown solid. It can be seen that the "incremental plasticity" method suffers from similar artefacts as other prior methods (but to a lesser degree). However, "incremental plasticity" introduces its own problems, namely volume loss and waviness (observable during bending the beam) and sharp creases (occurring while growing the muscle). To remove any effects of incorrect correspondence from the experiment, both examples use landmark constraints (shown as blue spheres); results are even more favorable to our method if using ICP constraints.

used by [Achenbach et al. 2015; Bouaziz et al. 2014; Gietzen et al. 2019]. Here, A_i is the local Voronoi area of vertex i. We used the publicly available ShapeOp library [Bouaziz et al. 2012, 2014] to implement E_{bend}. We then use E_{bend} for shape deformation by adding the ICP energy to it, and compare it to our method (Figure 25). The E_{bend} energy produces visible artefacts: when the beam is bent, there is a visible volume loss. When the beam is bent and forced to grow by the ICP markers, E_{bend} cannot grow the volume properly (the middle of the beam collapses). Similarly, wiggling artefacts can be observed in our hand muscle example. This is not surprising as rotations are relatively small in this example, and therefore E_{bend} becomes the variational biharmonic energy.

4.8 Comparison to incremental plasticity

One idea explored in prior work is to incrementally convert "elastic" deformations into "plasticity" after each ICP iteration [Gietzen et al. 2019; Sahillioğlu and Kavan 2015]. However, doing so introduces an

unwanted hysteresis-like effect. To demonstrate this, we compare our method to [Gietzen et al. 2019] who used two different energies combined with the "incremental plasticity" strategy. For the skull, they used an energy that penalizes volume change (Equation 1 in [Gietzen et al. 2019]). In the beam example, we can see that as one incrementally converts "elastic" deformation into "plastic" deformation, the bending beam undergoes a large spurious volume change. When imposing constraints that should cause the beam to inflate in its central region, the incremental plasticity method produces spikes (Figure 26, top left). This is because the constraints require volume to grow, but the energy penalizes volume growth. For the skin, Gietzen et al. used an energy similar to Equation 28 (Equation 5 in [Gietzen et al. 2019]), combined with "incremental plasticity". On the bending beam, the method suffers from a loss of volume and still causes the middle of the beam to collapse (Figure 26, top right). In the muscle example (Figure 26, bottom), the wiggle artefacts can be reduced by applying "incremental plasticity". However, the method introduces a vertical sharp edge at the middle of the muscle (clearly seen in the side view); our method has no such artefacts. The key difference to our method is that in the incremental plasticity method, the elasticity is explicitly converted to plasticity without any controlling criteria. Our method, however, finds **optimal** plastic strains, via optimization. For example, this makes it possible to employ minimal plastic strains when elastic energy can already do a good job.
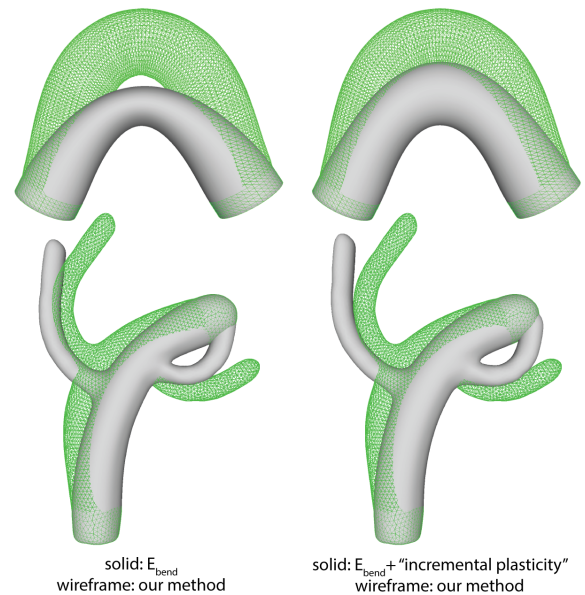
Fig. 27. **Comparison to E_{bend} on the shape benchmark of [Botsch and Sorkine 2008].**

We also compare to ShapeOp and "incremental plasticity" on the benchmark examples of [Botsch and Sorkine 2008] (Figure 27). The ShapeOp cylinder suffers from the same volume loss problem as the beam. In addition, E_{bend} causes self-collisions on the cactus example. "Incremental plasticity" improves the results somewhat, but it still cannot pass the benchmark.

5 CONCLUSION

We gave a shape deformation method that can model objects undergoing large spatially varying strains. Our method works by computing a plastic deformation gradient at each tet, such that the mesh deformed by these plastic deformation gradients matches the provided sparse landmarks and closest-point constraints. We support both constrained and unconstrained objects; the latter are supported by giving a numerical method to solve large sparse linear systems of equations with a known nullspace. We applied our method to the extraction of shapes of organs from medical images. Our method has been designed to extract as much information as possible from MRI images, despite the soft boundaries between the different organs. We extracted hand muscles, a liver, a hip bone and a hip muscle.

Our method does not require a dense target mesh; only a sparse set of observations is needed. If a dense target mesh is available, the problem becomes somewhat easier, as one can then use standard ICP algorithms. However, medical images contain many ambiguities and regions where there is not a sufficient contrast to clearly disambiguate two adjacent medical organs; making it impractical to extract dense target meshes. We apply our method to solid objects, but our plastic strain shape deformation method could also be used for shells (cloth). Doing so would require formulating the elastic energy of a plastically deformed FEM cloth, and computing the energy gradients and Hessians with respect to the plastic parameters. The size of the small square dense matrix that we need to invert in our incremental solve is three times the number of markers. While we easily employed up to a thousand of markers in our work, our method will slow down under a large number of markers. We do not re-mesh our tet meshes during the optimization. If the plastic strain causes some tetrahedra to collapse or nearly collapse, this will introduce numerical instabilities. Although not a problem in our examples (see Figure 17), such situations could be handled by remeshing the tet mesh during the optimization [Bargteil et al. 2007]. Our method requires a plausible template with a non-degenerate tet mesh. Re-meshing is important future work as it could extend the reach of our method, enabling one to start the optimization simply from a sphere tet mesh.

ACKNOWLEDGMENTS

CONFLICT OF INTEREST STATEMENT

The hip bone and hip muscle template meshes were obtained at Ziva Dynamics. Jernej Barbič is a shareholder, CTO and board member of Ziva Dynamics. Ziva Dynamics was not involved in this research. Nothing in this paper is to be understood as endorsement of Ziva Dynamics or its products.

REFERENCES

Jascha Achenbach, Eduard Zell, and Mario Botsch. 2015. Accurate Face Reconstruction through Anisotropic Fitting and Eye Correction. In *Vision, Modeling & Visualization*, David Bommes, Tobias Ritschel, and Thomas Schultz (Eds.).

M. Alexa, A. Angelidis, M.-P. Cani, S. Frisken, K. Singh, S. Schkolne, and D. Zorin. 2006. Interactive Shape Modeling. In *ACM SIGGRAPH 2006 Courses*. 93.

Brett Allen, Brian Curless, and Zoran Popović. 2003. The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans. *ACM Trans. Graph.* 22, 3 (2003), 587–594.

Brian Amberg, Sami Romdhani, and Thomas Vetter. 2007. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Artelys. 2019. Knitro. https://www.artelys.com/solvers/knitro/.

J. Barbič, M. da Silva, and J. Popović. 2009. Deformable Object Animation Using Reduced Optimal Control. *ACM Trans. on Graphics* 28, 3 (2009).

J. Barbič and D. L. James. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. on Graphics* 24, 3 (2005), 982–990.

J. Barbič and Y. Zhao. 2011. Real-time Large-deformation Substructuring. *ACM Trans. on Graphics (SIGGRAPH 2011)* 30, 4 (2011), 91:1–91:7.

A. W. Bargteil, C. Wojtan, J. K. Hodgins, and G. Turk. 2007. A finite element method for animating large viscoplastic flow. In *ACM Transactions on Graphics (SIGGRAPH 2007)*, Vol. 26. 16.

Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. 2007. TRACKS: Toward directable thin shells. *ACM Trans. on Graphics (SIGGRAPH 2007)* 26, 3 (2007), 50:1–50:10.

Mario Botsch and Leif Kobbelt. 2004. An Intuitive Framework for Real-Time Freeform Modeling. *ACM Trans. on Graphics (SIGGRAPH 2004)* 23, 3, 630–634.

Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. 2006. PriMo: Coupled Prisms for Intuitive Surface Modeling. In *Eurographics Symp. on Geometry Processing*. 11–20.

M. Botsch and O. Sorkine. 2008. On linear variational surface deformation methods. *IEEE Trans. on Vis. and Computer Graphics* 14, 1 (2008), 213–230.

Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. 2012. Shape-Up: Shaping Discrete Geometry with Projections. *Comput. Graph. Forum* 31, 5 (2012), 1657–1667.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.* 33, 4, Article 154 (July 2014), 154:1–154:11 pages.

Eddy Boxerman. 2003. *Speeding up cloth simulation.* Ph.D. Dissertation. University of British Columbia.

C. Erolin. 2019. Hand Anatomy. University of Dundee, Centre for Anatomy and Human Identification. https://sketchfab.com/anatomy_dundee/collections/hand-anatomy.

I. Chao, U. Pinkall, P. Sanan, and P. Schröder. 2010. A Simple Geometric Model for Elastic Deformations. *ACM Transactions on Graphics* 29, 3 (2010), 38:1–38:6.

W. Chen, F. Zhu, J. Zhao, S. Li, and G. Wang. 2018. Peridynamics-Based Fracture Animation for Elastoplastic Solids. In *Computer Graphics Forum*, Vol. 37. 112–124.

K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior. 2013. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *J Digit Imaging* 26, 6 (Dec 2013), 1045–1057.

Ali Hamadi Dicko, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, Francois Faure, Olivier Palombi, and Marie-Paule Cani. 2013. Anatomy Transfer. *ACM Trans. on Graphics (SIGGRAPH 2013)* 32, 6 (2013), 188:1–188:8.

Daniel Charles Drucker. 1957. *A definition of stable inelastic material.* Technical Report. DTIC Document.

Thomas Gietzen, Robert Brylka, Jascha Achenbach, Katja zum Hebel, Elmar Schömer, Mario Botsch, Ulrich Schwanecke, and Ralf Schulze. 2019. A method for automatic forensic facial reconstruction based on dense statistics of soft tissue tissue thickness. *PLOS ONE* 14 (2019), 1.

B. Gilles and N. Magnenat-Thalmann. 2010. Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations. *Med. Image Anal.* 14, 3 (2010), 291–302.

Benjamin Gilles, Laurent Moccozet, and Nadia Magnenat-Thalmann. 2006. Anatomical Modelling of the Musculoskeletal System from MRI. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*. 289–296.

Benjamin Gilles, Lionel Reveret, and Dinesh Pai. 2010. Creating and animating subject-specific anatomical models. *Computer Graphics Forum* 29, 8 (2010), 2340–2351.

Leo Grady. 2006. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28, 11 (2006), 1768–1783.

Hang Si. 2011. TetGen: A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator.

Jin Huang, Yiying Tong, Kun Zhou, Hujun Bao, and Mathieu Desbrun. 2011. Interactive Shape Interpolation through Controllable Dynamic Deformation. *IEEE Trans. on Visualization and Computer Graphics* 17, 7 (2011), 983–992.

Z. Huang, N. A. Carr, and T. Ju. 2019. Variational implicit point set surfaces. *ACM Trans. on Graphics (SIGGRAPH 2019)* 38, 4 (2019).

A.E. Ichim, P. Kadlecek, L. Kavan, and M. Pauly. 2017. Phace: Physics-based Face Modeling and Animation. *ACM Trans. on Graphics (SIGGRAPH 2017)* 36, 4 (2017).

T. Igarashi, T. Moscovich, and J. F. Hughes. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. on Graphics (SIGGRAPH 2005)* 24, 3 (2005), 1134–1141.

G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. In *Symp. on Computer Animation (SCA)*. 131–140.

A. Jacobson, I. Baran, J. Popović, and O. Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. on Graphics (TOG)* 30, 4 (2011), 78.

Petr Kadlecek, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Krivanek, and Ladislav Kavan. 2016. Reconstructing Personalized Anatomical Models for Physics-based Body Animation. *ACM Trans. Graph.* 35, 6 (2016).

Ladislav Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. 2011. Physics-Inspired Upsampling for Cloth Simulation in Games. *ACM Trans. Graph.* 30, 4, Article 93 (2011), 10 pages.

Ali Emre Kavur, M. Alper Selver, Oğuz Dicle, Mustafa Barış, and N. Sinem Gezer. 2019. *CHAOS - Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data.* https://doi.org/10.5281/zenodo.3362844

Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM Trans. on Graphics (TOG)* 32, 3, Article 29 (2013).

Seunghwan Lee, Ri Yu, Jungnam Park, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. 2018. Dexterous manipulation and control with volumetric muscles. *ACM Transactions on Graphics (SIGGRAPH 2018)* 37, 4 (2018), 57:1–57:13.

S. H. Lee, E. Sifakis, and D. Terzopoulos. 2009. Comprehensive Biomechanical Modeling and Simulation of the Upper Body. *ACM Trans. on Graphics* 28, 4 (2009), 99:1–99:17.

Hao Li, Robert W. Sumner, and Mark Pauly. 2008. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Computer Graphics Forum (Proc. SGP'08)* 27, 5 (July 2008).

T. McInerney and D. Terzopoulos. 2008. Deformable Models. In *Handbook of Medical Image Processing and Analysis (2nd Edition)*, I. Bankman (Ed.). Chapter 8, 145–166.

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Computer Graphics Forum (Proc. of Eurographics)* 31, 2 (may 2012). http://www.gmrv.es/Publications/2012/MBTBMOM12

M. Müller and M. Gross. 2004. Interactive Virtual Materials. In *Proc. of Graphics Interface 2004*. 239–246.

M. Müller, B. Heidelberger, M. Teschner, and M. Gross. 2005. Meshless Deformations Based on Shape Matching. In *Proc. of ACM SIGGRAPH 2005*. 471–478.

G. Niculescu, J.L. Nosher, M.D. Schneider, and D.J. Foran. 2009. A deformable model for tracking tumors across consecutive imaging studies. *Int. J. of Computer-Assisted Radiology and Surgery* 4, 4 (2009), 337–347.

James F. O'Brien, Adam W. Bargteil, and Jessica K. Hodgins. 2002. Graphical Modeling and Animation of Ductile Fracture. In *Proceedings of ACM SIGGRAPH 2002*. 291–294.

W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. 2007. *Numerical recipes: The art of scientific computing* (third ed.). Cambridge University Press, Cambridge, UK.

T. Rhee, U. Neumann, J. Lewis, and K. S. Nayak. 2011. Scan-Based Volume Animation Driven by Locally Adaptive Articulated Registrations. *IEEE Trans. on Visualization and Computer Graphics* 17, 3 (2011), 368–379.

Yusuf Sahillioğlu and Ladislav Kavan. 2015. Skuller: A volumetric shape registration algorithm for modeling skull deformities. *Medical image analysis* 23, 1 (2015), 15–27.

Shunsuke Saito, Zi-Ye Zhou, and Ladislav Kavan. 2015. Computational Bodybuilding: Anatomically-based Modeling of Human Bodies. *ACM Trans. on Graphics (SIGGRAPH 2015)* 34, 4 (2015).

J. Schmid, E. Gobbetti J. A. I. Guitián, and N. Magnenat-Thalmann. 2011. A GPU framework for parallel segmentation of volumetric images using discrete deformable models. *The Visual Computer* 27 (2011), 85–95.

Jérôme Schmid and Nadia Magnenat-Thalmann. 2008. MRI Bone Segmentation Using Deformable Models and Shape Priors. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*. 119–126.

J. Schmid, A. Sandholm, F. Chung, D. Thalmann, H. Delingette, and N. Magnenat-Thalmann. 2009. Musculoskeletal Simulation Model Generation from MRI Data Sets and Motion Capture Data. *Recent Advances in the 3D Physiological Human* (2009), 3–19.

Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. on Graphics (SIGGRAPH 2005)* 24, 3 (Aug. 2005), 417–425.

Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2 (2018), 12:1–12:15.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symp. on Geometry Processing*, Vol. 4. 109–116.

O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Symp. on Geometry processing*. 175–184.

Stephcavs. 2019. KidneyFullBody 1.0.0: Full CT scan of body. https://www.embodi3d.com/files/file/26389-kidneyfullbody/.

A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. on Graphics (SIGGRAPH 2013)* 32, 4 (2013), 102:1–102:10.

Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. *ACM Trans. on Graphics (SIGGRAPH 2004)* 23, 3 (2004), 399–405.

J. Sylvester. 1884. Sur l'equations en matrices p x = x q. *C. R. Acad. Sci. Paris.* 99, 2 (1884), 67–71, 115–116.

G. Székely, A. Kelemen, C. Brechbühler, and G. Gerig. 1996. Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier contour and surface models. *Medical Image Analysis* 1, 1 (1996), 19–34.

Tissue. 2013. Weta Digital: Tissue Muscle and Fat Simulation System.

Turbosquid. 2019. www.turbosquid.com.

G. Turk and J. O'Brien. 1999. Shape transformation using variational implicit functions. In *Proc. of ACM SIGGRAPH 1999*. 335–342.

C. Twigg and Z. Kačić-Alesić. 2010. Point Cloud Glue: constraining simulations using the procrustes transform. In *Symp. on Computer Animation (SCA)*. 45–54.

Christopher D. Twigg and Zoran Kačić-Alesić. 2011. Optimization for Sag-Free Simulations. In *Proc. of the 2011 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 225–236.

U.S. National Library of Medicine. 1994. The visible human project. http://www.nlm.nih.gov/research/visible/.

Bohan Wang, George Matcuk, and Jernej Barbič. 2019. Hand Modeling and Simulation Using Stabilized Magnetic Resonance Imaging. *ACM Trans. on Graphics (SIGGRAPH 2019)* 38, 4 (2019).

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015b. Deformation capture and modeling of soft objects. *ACM Transactions on Graphics (TOG) (SIGGRAPH 2015)* 34, 4 (2015), 94.

Yu Wang, Alec Jacobson, Jernej Barbič, and Ladislav Kavan. 2015a. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG) (SIGGRAPH 2015)* 34, 4 (2015), 57.

Max A. Woodbury. 1950. Inverting modified matrices. *Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ* (1950), 4pp.

Hao Zhang. 2004. Discrete combinatorial Laplacian operators for digital geometry processing. In *Proceedings of SIAM Conference on Geometric Design and Computing*. Nashboro Press, 575–592.

Ziva Dynamics. 2019. Male Virtual Human "Max". http://zivadynamics.com/ziva-characters.

Zygote. 2016. Zygote body. http://www.zygotebody.com.

A PLASTIC STRAIN LAPLACIAN AND ITS NULLSPACE

Let $\mathbf{L}^{sc} \in \mathbb{R}^{m \times m}$ denote the discrete mesh Laplacian for *scalar* fields on mesh tetrahedra [Zhang 2004],

$$\mathbf{L}^{sc}_{i,j} = \begin{cases} \#\text{adjacent tets} & \text{if } i = j, \\ -1 & \text{if } i \neq j, \text{ and } i, j \text{ are adjacent}, \\ 0 & \text{otherwise.} \end{cases} \quad (29)$$

Given a plastic strain state $\mathbf{s} = [\mathbf{s_1}, \mathbf{s_2}, \mathbf{s_3}, \mathbf{s_4}, \mathbf{s_5}, \mathbf{s_6}] \in \mathbb{R}^{6m}$, where $\mathbf{s_i} \in \mathbb{R}^m$, define the *plastic strain Laplacian*

$$\mathbf{Ls} = [\mathbf{L}^{sc}\mathbf{s_1}, \sqrt{2}\mathbf{L}^{sc}\mathbf{s_2}, \sqrt{2}\mathbf{L}^{sc}\mathbf{s_3}, \mathbf{L}^{sc}\mathbf{s_4}, \sqrt{2}\mathbf{L}^{sc}\mathbf{s_5}, \mathbf{L}^{sc}\mathbf{s_6}], \quad (30)$$

where the $\sqrt{2}$ were added to account for the fact that s_2, s_3, s_5 control two entries in the symmetric matrix $F_p \in \mathbb{R}^{3 \times 3}$.

Lemma: Assume that the tet mesh $\mathcal{M}$ has a single connected component. Then, the nullspace of $\mathbf{L}$ is 6-dimensional and consists of vectors $\psi_i = [\mathbf{s_1}, \ldots, \mathbf{s_6}]$ where $\mathbf{s_j} \in \mathbb{R}^m$ is all ones when $j = i$, and all zeros otherwise.

Proof: First, observe that $\mathbf{L}^{sc}$ is symmetric positive semi-definite with a single nullspace vector, namely the vector of all 1s. This follows from the identity

$$\mathbf{x}^T \mathbf{L}^{sc} \mathbf{x} = \sum_{i \text{ and } j \text{ adjacent}} (x_i - x_j)^2, \quad (31)$$

i.e., $\mathbf{x}^T \mathbf{L}^{sc} \mathbf{x} = 0$ is only possible if all x_i are the same.

We have $0 = \mathbf{s}^T \mathbf{Ls} = \sum_{i=1}^6 \xi_i \mathbf{s_i}^T \mathbf{L}^{sc} \mathbf{s_i}$, where $\xi_i = \sqrt{2}$ for $i = 2, 3, 5$; and 1 otherwise. Because $\mathbf{L}^{sc}$ is symmetric positive semi-definite, each $\mathbf{s_i}$ must either be $\mathbf{0}$ or a non-zero nullspace vector of $\mathbf{L}^{sc}$, i.e., a vector of all 1s. A linearly independent orthonormal nullspace basis emerges when we have a vector for all 1s for exactly one i. There are 6 such choices, giving the vectors ψ_i; we normalize them by dividing with $\sqrt{m}$. ∎

B FIRST AND SECOND DERIVATIVES OF ELASTIC ENERGY WITH RESPECT TO PLASTIC STRAIN

For convenience, we denote $F_{p,i}$ as i−th entry of the vector $\text{vec}(F_p) \in \mathcal{R}^9$. The first-order derivatives are

$$\frac{\partial \mathscr{E}}{\partial x_i} = V \frac{\partial \psi}{\partial F_e} : \frac{\partial F_e}{\partial x_i} = VP : \frac{\partial F_e}{\partial x_i}, \tag{32}$$

$$\frac{\partial \mathscr{E}}{\partial F_{p,i}} = V \frac{\partial \psi}{\partial F_{p_i}} + \frac{\partial V}{\partial F_{p_i}} \psi = VP : \frac{\partial F_e}{\partial F_{p_i}} + \frac{\partial V}{\partial F_{p_i}} \psi, \quad \text{where} \tag{33}$$

$$\frac{\partial F_e}{\partial x_i} = \frac{\partial F}{\partial x_i} F_p^{-1}, \quad \frac{\partial F_e}{\partial F_{p,i}} = F \frac{\partial F_p^{-1}}{\partial F_{p,i}}, \quad \text{and} \tag{34}$$

$$\frac{\partial V}{\partial F_{p,i}} = \frac{\partial |F_p|}{\partial F_{p_i}} V_0. \tag{35}$$

Here, P is the first Piola-Kirchhoff stress tensor and $\partial F/\partial x$ is a constant matrix commonly used in the equations for FEM simulation. For the second-order derivatives, we first compute $\partial^2 \mathscr{E}/\partial x^2$. This is the tangent stiffness matrix in the FEM simulation under a fixed F_p. It is computed as

$$\frac{\partial^2 \mathscr{E}}{\partial x_i \partial x_j} = V \frac{\partial F_e}{\partial x_j}^T : \frac{\partial P}{\partial F_e} : \frac{\partial F_e}{\partial x_i}. \tag{36}$$

Here, $\partial P/\partial F_e$ is a standard term in FEM nonlinear elastic simulation; it only depends on the strain-stress law (the material model). Next, we compute $\partial^2 \mathscr{E}/(\partial x \partial F_p)$,

$$\frac{\partial^2 \mathscr{E}}{\partial x_i \partial F_{p,j}} = \frac{\partial V}{\partial F_{p,j}} \left(P : \frac{\partial F_e}{\partial x_i} \right) + \tag{37}$$

$$V \frac{\partial F_e}{\partial F_{p,j}}^T : \frac{\partial P}{\partial F_e} : \frac{\partial F_e}{\partial x_i} + VP \frac{\partial^2 F_e}{\partial x_i \partial F_{p,j}}, \quad \text{where} \tag{38}$$

$$\frac{\partial^2 F_e}{\partial x_i \partial F_{p,j}} = \frac{\partial F}{\partial x_i} \frac{\partial F_p^{-1}}{\partial F_{p,j}}. \tag{39}$$

Finally, we have

$$\frac{\partial^2 \mathscr{E}}{\partial F_{p,i} \partial F_{p,j}} = \frac{\partial^2 V}{\partial F_{p,i} \partial F_{p,j}} \psi + \tag{40}$$

$$V \left(P : \frac{\partial^2 F_e}{\partial F_{p,i} \partial F_{p,j}} + \frac{\partial F_e}{\partial F_{p,j}}^T : \frac{\partial P}{\partial F_e} : \frac{\partial F_e}{\partial F_{p,i}} \right) + \tag{41}$$

$$\frac{\partial V}{\partial F_{p,j}} \frac{\partial \psi}{\partial F_{p,i}} + \frac{\partial V}{\partial F_{p,i}} \frac{\partial \psi}{\partial F_{p,j}}, \quad \text{where} \tag{42}$$

$$\frac{\partial^2 V}{\partial F_{p,i} \partial F_{p,j}} = \frac{\partial^2 |F_p|}{\partial F_{p,i} \partial F_{p,j}} V_0, \tag{43}$$

$$\frac{\partial^2 F_e}{\partial F_{p,i} \partial F_{p,j}} = F \frac{\partial^2 F_p^{-1}}{\partial F_{p,i} \partial F_{p,j}}. \tag{44}$$

The quantities ψ, P and $\partial P/\partial F_e$ are determined by the chosen elastic material model. After computing the above derivatives, there is still a missing link between F_p and s. Because we want to directly optimize s, we also need the derivatives of $\mathscr{E}(F_p(s), x)$ with respect to s. From Equation 2 we can see that F_p is linearly dependent on s. Therefore, so we can define a matrix Y such that $\text{vec}(F_p) = Ys$. Then all the derivatives can be easily transferred to derivation by s by multiplying with Y.

C PROOF OF SINGULAR LEMMA

Statement (i) follows from well-known linear algebra facts $\mathcal{R}(A) = \mathcal{N}(A^T)^\perp$ and $\dim(\mathcal{N}(A)) + \dim(\mathcal{R}(A)) = p$, and the symmetry of A. As per (ii), A maps $\mathcal{R}(A)$ into itself, and no vector from $\mathcal{R}(A)$ maps to zero, hence the restriction of A to $\mathcal{R}(A)$ is invertible, establishing a unique solution to $Ax = b$ with the property that $x \perp \psi_i$ for all $i = 1, \ldots, k$. This unique solution is the minimizer of

$$\min_x \frac{1}{2} x^T A x - b^T x \tag{45}$$

$$\text{s.t.} \quad \psi_i^T x = 0 \text{ for all } i = 1, \ldots, k. \tag{46}$$

When expressed using Lagrange multipliers, this gives Equation 15. Suppose $x = n + r$ is another solution and $n \in \mathcal{N}(A)$ and $r \in \mathcal{R}(A)$. Then $b = Ax = Ar$ and hence r is the unique solution from Equation 15. The vector n can be an arbitrary nullspace vector, proving the last statement of (ii). As per (iii), suppose we have $0 = B(n+r) = Ar + \sum_{i=1}^k \frac{\lambda_i}{\alpha_i} (\psi_i^T n) \psi_i$. Observe that the first summand is in $\mathcal{R}(A)$ and the second in $\mathcal{N}(A)$. Hence, $B(n + r)$ can only be zero if both summands are zero. $Ar = 0$ implies $r = 0$. The second summand can only be zero if $n \perp \psi_i$ for each i, which implies that $n = 0$. Hence, B is invertible. The last statement of (iii) can be verified by expanding $\left(A + \sum_{i=1}^k \alpha_i \psi_i \psi_i^T \right) \left(x + \sum_{i=1}^k \frac{\lambda_i}{\alpha_i} \psi_i \right)$. ∎

D PROOF OF NULLSPACE LEMMA

We are trying to prove that $\mathbf{K}(\mathbf{F_p}(\mathbf{s}), \mathbf{x})$ is 6-dimensional for any $\mathbf{x}$ that solves $\mathbf{f_e}(\mathbf{s}, \mathbf{x}) = 0$. First, if $\mathbf{x}$ is a solution, then translating all vertices of the object by the same constant 3-dimensional vector is also a solution. This means that vector $\psi_i := (e_i, e_i, \ldots, e_i)$ is in the nullspace of $\mathbf{K}$, where $e_i \in \mathbb{R}^3$ is the i-th standard basis vector, for $i = 1, 2, 3$. Now, suppose we rotate the object with an infinitesimal rotation $X \mapsto X + e_i \times X$. Observe that for general plastic strains $\mathbf{s}$, the elastic forces in each individual tet are not zero even in the equilibrium $\mathbf{x}$; but the
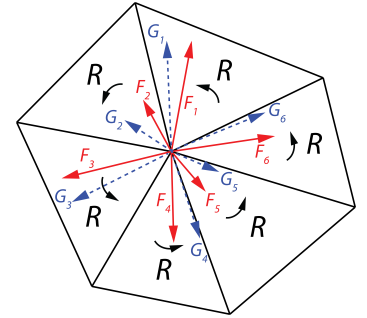

Fig. 28. **Illustration of the nullspace proof.** The original forces F_i sum to zero. We have $G_i = RF_i$; note that R is the same for all tets. Therefore, the rotated forces G_i also sum to zero. Hence there is no change in the internal elastic force under a rotation, i.e., infinitesimal rotations are in the nullspace of $\mathbf{K}$.

contributions of elastic forces on a tet mesh vertex from all adjacent tets sum to zero. As we rotate the object, the forces contributed by adjacent tets to a specific tet mesh vertex rotate by the same rotation in each tet. Therefore, as these forces sum to zero, they continue to sum to zero even under the rotation (see Figure 28). This means that the vector of infinitesimal displacements $\psi_{3+i} := [e_i \times x_1, e_i \times x_2, \ldots, e_i \times x_n]$ induced by the infinitesimal rotation is in the nullspace of $\mathbf{K}$, for each $i = 1, 2, 3$. Here, x_i are the components of $\mathbf{x} = [x_1, x_2, \ldots, x_n]$. The vectors ψ_i, $i = 1, 2, 3, 4, 5, 6$, form the nullspace of $\mathbf{K}$. ∎

Finally, we inform the reader that the nullspace of $\mathbf{K}(\mathbf{F_p}(\mathbf{s}), \mathbf{x})$ is only 3-dimensional if $\mathbf{x}$ is *not* an elastic equilibrium. In this case,

only translations are in the nullspace. Infinitesimal rotations are not in the nullspace because under an infinitesimal rotation, the non-zero elastic forces $\mathbf{f_e}$ rotate, i.e., they do not remain the same. The assumption of $\mathbf{x}$ being the equilibrium shape is therefore crucial (and is satisfied in our method).

E SECOND DERIVATIVE OF POLAR DECOMPOSITION

To compute the second-order derivatives, we differentiate

$$\frac{\partial F}{\partial F_i} = \frac{\partial R}{\partial F_i}S + R\frac{\partial S}{\partial F_i}, \tag{47}$$

$$\frac{\partial^2 F}{\partial F_i \partial F_j} = \frac{\partial^2 R}{\partial F_i \partial F_j}S + \frac{\partial R}{\partial F_i}\frac{\partial S}{\partial F_j} + \frac{\partial R}{\partial F_j}\frac{\partial S}{\partial F_i} + R\frac{\partial^2 S}{\partial F_i \partial F_j}, \tag{48}$$

$$\frac{\partial^2 R}{\partial F_i \partial F_j} = \left(-R\frac{\partial^2 S}{\partial F_i \partial F_j} - \frac{\partial R}{\partial F_j}\frac{\partial S}{\partial F_i} - \frac{\partial R}{\partial F_i}\frac{\partial S}{\partial F_j}\right)S^{-1}. \tag{49}$$

To compute $\partial^2 R/(\partial F_i \partial F_j)$, we need to compute $\partial^2 S/(\partial F_i \partial F_j)$ first. This can be derived in the same way as for $\partial S/\partial F_i$. Starting from Equation 22, we have

$$\frac{\partial^2 F^T F}{\partial F_i \partial F_j} = \frac{\partial^2 S}{\partial F_i \partial F_j}S + \frac{\partial S}{\partial F_i}\frac{\partial S}{\partial F_j} + \frac{\partial S}{\partial F_j}\frac{\partial S}{\partial F_i} + S\frac{\partial^2 S}{\partial F_i \partial F_j}. \tag{50}$$

We can now solve a similar Sylvester equation

$$\text{vec}(\frac{\partial^2 S}{\partial F_i \partial F_j}) = (S \oplus S)^{-1}\text{vec}(C), \tag{51}$$

$$C = \frac{\partial^2 F^T F}{\partial F_i \partial F_j} - \frac{\partial S}{\partial F_i}\frac{\partial S}{\partial F_j} - \frac{\partial S}{\partial F_j}\frac{\partial S}{\partial F_i}. \tag{52}$$

F FORMULAS FOR $\mathbf{A_k}, \mathbf{b_k}, \mathbf{c_k}$ (EQUATION 6)

Let the attachment k be embedded into a tetrahedron t_k with barycentric weights $\left[w_1^k, w_2^k, w_3^k, w_4^k\right]$. We have

$$\mathbf{A_k} = \begin{bmatrix} w_1^k I_3 & w_2^k I_3 & w_3^k I_3 & w_4^k I_3 \end{bmatrix} S^k \in \mathbb{R}^{3\times 3n} \tag{53}$$

$$\mathbf{b_k} = -y_k \in \mathbb{R}^3, \tag{54}$$

where $y_k \in \mathbb{R}^3$ is the attachment's target position, $I_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix, and $S^k \in \mathbb{R}^{12\times 3n}$ is a selection matrix that selects the positions of vertices of t_k. The scalar $\mathbf{c_k}$ is the weight of the attachment k. Equivalent formulas apply to landmarks and ICP markers. The attachment energy and forces are,

$$\mathcal{E}_a(\mathbf{x}) = \frac{1}{2}\sum_k \mathbf{c_k}\|\mathbf{A_k x} + \mathbf{b_k}\|^2, \tag{55}$$

$$f_a(\mathbf{x}) = \frac{d\mathcal{E}_a}{d\mathbf{x}} = \sum_k \mathbf{c_k}\mathbf{A_k^T}(\mathbf{A_k x} + \mathbf{b_k}). \tag{56}$$

G MEETING CONSTRAINTS BY SCALING THE ELASTIC STIFFNESS

In this section, we demonstrate that scaling the elastic stiffness cannot be used to "better" meet the constraints. We illustrate this on a simple toy problem of linear elasticity with linear constraints. Denote the vertex displacements by $u \in \mathbb{R}^{3n}$ and the stiffness matrix by $K \in \mathbb{R}^{3n\times 3n}$.

First, consider scaling the stiffness ($K \to \alpha K$) when using hard constraints:

$$\min_u\ < Ku, u > \tag{57}$$

$$\text{s.t. } Cu = b \tag{58}$$

v.s.

$$\min_u\ < (\alpha K)u, u > \tag{59}$$

$$\text{s.t. } Cu = b \tag{60}$$

produces same u for any scalar $\alpha > 0$.

Similarly, consider scaling stiffness ($K \to \alpha K$) when using soft constraints:

$$\min_u\ < (\alpha K)u, u > +\beta\|Cu - b\|^2 \tag{61}$$

produces same u as

$$\min_u\ < Ku, u > +\beta/\alpha\|Cu - b\|^2. \tag{62}$$

Therefore, scaling the stiffness does not provide any more expressive shape deformation power than tweaking β in:

$$\min_u\ < Ku, u > +\beta\|Cu - b\|^2. \tag{63}$$

As one tweaks β in the above, one can choose between (A) meeting the constraints well (when $\beta >> 1$), or (B) produce smooth deformation (when $\beta << 1$). In option (A), one obtains sharp non-smooth deformation ("spikes"), and in option (B) one does not meet the constraints. In the middle territory, one doesn't meet either of these goals well, as seen in Figure 2,f. So, in summary, merely increasing the elastic stiffness does not help with better satisfying the constraints.