Modeling of Personalized Anatomy using Plastic Strains

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

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:

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 $\Phi$ 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 as 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) [Barbić et al. 2009]. As shown in Section 3.8, none of these methods was able to capture the material locally stretches (or compresses) by large amounts; and this amount is different in different directions and varies spatially across the model.

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. Poisson’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.

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...
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 Magenenat-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 Magenenat-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 Magenenat-Thalmann 2010], which biases the output against volume growth. For bones, a similar idea was also presented in [Schmid et al. 2011; Schmid and Magenenat-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]. Kadlecik 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.

---

*Bohan Wang, George Matcuk, and Jernej Barbč*
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 $M$, such that the FEM equilibrium shape under $F_p$ 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 $3 \times 3$ matrix $F_p$ at each tetrahedron of $M$. For each specific deformed shape $x \in \mathbb{R}^{3m}$, one can define and compute the deformation gradient $F$ between $x$ and $x$ at each tet [Müller and Gross 2004]. The elastic deformation gradient $F_e$ can then be defined as $F_e = FF_p^{-1}$ [Bargteil et al. 2007] (see Figure 5).

Observe that for any shape $x$, there exists a corresponding plastic deformation gradient $F_p$ such that $x$ is the elastic equilibrium under $F_p$, namely $F_p = F$. This means that the space of all plastic deformation gradients $F_p$ is expressive enough to capture all shapes $x$. The elastic energy of a single tet is defined as

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

where $V$ is the rest volume of the tet under the plastic deformation $F_p$, and $\psi$ is the elastic energy density function. We have $V = |F_p| V_0$, where $V_0$ is the tet's volume in $M$, and $|F_p|$ is the determinant of the matrix $F_p$. Elastic forces equal $f_e(F_p, x) = d\mathcal{E}(F_p, x)/dx$. When solving our optimization problem to compute $F_p$ 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 $\psi$. 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 $3 \times 3$ 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 $\psi$, 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 $3 \times 3$ matrices. We can therefore model $F_p$ as a symmetric matrix and parameterize it using a vector $s \in \mathbb{R}^6$.

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

We model plasticity globally using a vector $s \in \mathbb{R}^{6m}$, where $m$ is the number of tets in $M$. 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-collinear 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 $x$ under the attachments and plastic strain $F_p$, so that the medical image observations are met as closely as possible,

$$\arg \min_{s, \mathbf{x}} \left\{ \| L s \|^2 + \alpha \mathcal{E}_{\text{AE}}(s) + \beta \mathcal{E}_{\text{L}}(\mathbf{x}) \right\},$$

subject to: $f_e(F_p(s), \mathbf{x}) + f_a(\mathbf{x}) = 0$, where $\alpha \geq 0$ and $\beta \geq 0$ are scalar trade-off weights, and $L$ is the plastic strain Laplacian. We define $L$ 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 $F_p$, 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 $F_p$ and under the attachment forces $f_a$. Attachments pull a material point embedded into a tet to a fixed world-coordinate location using spring forces; precise definitions of the attachment energy $\mathcal{E}_a$ and attachment forces $f_a$ are in Appendix F. It is important to emphasize that our output shapes are always in static equilibrium under the plastic strains $F_p$, and both this equilibrium shape $x$ and $F_p$ 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 $\mathcal{E}_a$ in Equation 3 in the next paragraph. The medical image energy measures how closely $\mathbf{x}$ matches the medical image constraints,

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

where $S$ is the interpolation matrix that selects $Y_i$, namely $Sx = Y$. The function $\text{closestPoint}(x, z_i) \in \mathbb{R}^3$ computes the closest point to $z_i \in \mathbb{R}^3$ on the surface of the tet mesh with vertex positions $\mathbf{x}$.
With soft constraints, it is important to keep the spring coefficient tuned to reasonable values, and then find a static equilibrium $x$. As per the attachment energy $E_a$, we initially tried solving the optimization problem of Equations 3 and 4 without it. This seems natural, but actually did not work. Namely, without $E_a$, there is nothing in Equations 3 and 4 that forces the plastic strains to reasonable values. The optimizer is free to set $F_p$ to arbitrarily extreme values, and then find a static equilibrium $x$ under the attachment forces. In our outputs, we would see smooth nearly tet-collapsing plastic strains that result in a static equilibrium $x$ 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 $\mathcal{E}(F_p, x)$ 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, $E_a$. This keeps the expression in Equation 3 quadratic in $x$ and $F_p$, which we exploit in Section 3.4 for speed. Observe that $E_a$ behaves similarly to the elastic energy: if the plastic strain causes a rest shape that is far from the attachment targets, then both $E_a$ 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 $E_a$ 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 $a = 10^9$ and $\beta = 10^8$ in our examples. The weights $a$ and $\beta$ 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 model [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...
Modeling of Personalized Anatomy using Plastic Strains

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 \( x \) as a function of \( s \). The Gauss-Newton method uses the Jacobian \( J = dx/ds \), which models the change in the static equilibrium \( x \) 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 \( \text{closestPoint}(x, z) \) is a nonlinear function of \( x \), we can treat the barycentric coordinates of the closest point to \( z \) as fixed during one ICP iteration, and therefore \( \text{closestPoint}(x, z) \) becomes a linear function of \( x \). Therefore, \( E_{\text{Ag}}(x) \) can be seen as a quadratic function of \( x \), and so is \( E_a(x) \). We can thus rewrite Equations 3 and 4 as

\[
\begin{align*}
\arg\min_{x, s} & \, \frac{1}{2} ||Ls||^2 + \sum_{k=1}^{q+r+t} \frac{c_k}{2} ||A_k x + b_k||^2, \\
\text{s.t.} & \, f_{\text{net}}(s, x) = 0,
\end{align*}
\]  

where \( f_{\text{net}}(s, x) = f_e(F_p(s), x) + f_a(x) \) is the net force on the mesh, and constant matrices, vectors and scalars \( A_k, b_k, c_k \) are independent of \( s \) and \( x \) (we give them in Appendix F). The integer \( t \) denotes...
After each iteration, we update $x$ as $x_{\text{net}}$; we do this by performing eigen-decomposition of the symmetric matrix $F$, Vol. 1, No. 1, Article . Publication date: February 2021.

the initial guess for our full optimization as described above.

...is already differentially "baked" into Equation 8 via $\Delta x = J \Delta s$. At iteration $i$ of our Gauss-Newton method, given the previous iterates $x^i$ and $s^i$, we minimize a non-linearly constrained problem,

$$\arg\min_{x^{i+1}, \Delta s^i} \frac{1}{2} ||L(x^i + \Delta s^i)||^2 + \sum_{k=1}^{q+t+1} \frac{C_k}{2} ||A_k (x^i + J \Delta s^i) + b_k||^2, \quad (8)$$

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

After each iteration, we update $s^{i+1} = s^i + \Delta s^i$. Observe that Equation 8 does not depend on $x^{i+1}$, and that the constraint of Equation 9 is already differentially "baked" into Equation 8 via $\Delta x = J \Delta s$. We therefore first minimize Equation 8 for $\Delta s^i$, using unconstrained minimization; call the solution $\Delta s^i$. 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 $\Delta s^i$ so that the corresponding $F_\Delta$ is always positive-definite for each tet; we do this by performing eigen-decomposition of the symmetric matrix $F_{\Delta}$ 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 $\Delta s^i$. Specifically, for $\eta \geq 0$, we first solve Equation 7 with $s(\eta) = s^i + \eta \Delta s^i$ for $x = x(\eta)$ 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 $x = x(\eta)$ and $s = s(\eta)$. We perform the 1D line search for the optimal $\eta$ 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.

...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, \( \frac{\partial f_{\text{net}}}{\partial x} \) is square and invertible. Therefore, one can obtain a formula for \( J \) by differentiating Equation 7 with respect to \( s \),

\[
J = \left( \frac{\partial f_{\text{net}}}{\partial x} \right)^{-1} \frac{\partial f_{\text{net}}}{\partial s}.
\]

(10)

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

\[
H = L^2 + J^T \left( \sum_k c_k A_k^T A_k \right) J = L^2 + Z^T Z,
\]

(11)

Therefore, when the number of tet mesh elements \( m \) is large, it is not practically possible to compute \( H \), 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 \( Z \in \mathbb{R}^{3(q+r+t)} \times 6m \) is square and invertible. Therefore, one can obtain a formula for \( H \) by differentiating Equation 7 with respect to \( s \).

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

\[
H = L^2 + J^T \left( \sum_k c_k A_k^T A_k \right) J = L^2 + Z^T Z,
\]

(11)

Therefore, when the number of tet mesh elements \( m \) is large, it is not practically possible to compute \( H \), 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 \( Z \in \mathbb{R}^{3(q+r+t)} \times 6m \) is square and invertible. Therefore, one can obtain a formula for \( H \) by differentiating Equation 7 with respect to \( s \). 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 \( \psi_i \) for all \( i = 1, \ldots, k \) (Figure 10).

(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 \( \psi_i \) for all \( i = 1, \ldots, k \).

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 \( H \) is a dense \( 6m \times 6m \) matrix. The column \( t_{\text{prep}} \) gives a common pre-processing time for both CG and our method.

<table>
<thead>
<tr>
<th>Example</th>
<th>6m</th>
<th>3(q + r + t)</th>
<th>t_{\text{prep}}</th>
<th>CG</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand muscle</td>
<td>237,954</td>
<td>1,143</td>
<td>17.3s</td>
<td>897.5s</td>
<td>9.3s</td>
</tr>
<tr>
<td>Hip bone</td>
<td>172,440</td>
<td>1,497</td>
<td>11.4s</td>
<td>408.9s</td>
<td>7.2s</td>
</tr>
<tr>
<td>Liver</td>
<td>259,326</td>
<td>1,272</td>
<td>24.4s</td>
<td>1486.7s</td>
<td>10.0s</td>
</tr>
</tbody>
</table>

Figure 10. Illustration of the Singular Lemma.
This solution can be found by solving the non-singular linear system
\[
\begin{bmatrix}
A & \psi_1 & \ldots & \psi_k \\
\psi_1^T & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\psi_k^T & 0 & \ldots & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_k
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\] (15)

All other solutions equal \(x + \sum_{i=1}^{k} \lambda_i \psi_i\) for some scalars \(\lambda_i \in \mathbb{R}\). (iii) For any scalars \(a_i \neq 0\), the matrix \(B = A + \sum_{i=1}^{k} a_i \psi_i \psi_i^T\) is invertible. If \(\psi_i\) are orthonormal vectors, then the solution to \(By = h\) equals \(y = x + \sum_{i=1}^{k} \frac{a_i}{a_i - \lambda_i} \psi_i\), where \(x\) and \(\lambda_i\) are solutions to Equation 15 with \(b = \text{proj}_{\text{span}(A)} h = h - \sum_{i=1}^{k} (\psi_i^T h) \psi_i\). 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 \(f_{\text{act}} = f_e\), and the equation \(f_e(F_p(s), x) = 0\) no longer has a unique solution \(x\) for a fixed plastic state \(s\). This can be intuitively easily understood: one can arbitrarily translate and rotate any elastic equilibrium shape \(x\) under the given plastic state \(s\); doing so produces another elastic equilibrium shape. The space of solutions \(x\) is 6-dimensional. This means that we can no longer uniquely solve Equation 7 for \(x\) during our line search of Section 3.4. Furthermore, the square tangent stiffness matrix

\[
K(F_p(s), x) = \frac{\partial f_e(F_p(s), x)}{\partial x}
\]

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 \(x\) under plasticity, is 6-dimensional. The six nullspace vectors are \(\psi_i := [e_i, e_1, \ldots, e_i]\), where \(e_i \in \mathbb{R}^3\) is the \(i\)-th standard basis vector, and \(\psi_{i+1} := [e_1 \times x_1, e_1 \times x_2, \ldots, e_i \times x_n]\), for \(i = 1, 2, 3\).

To the best of our knowledge, this fact of elastoplastic-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,

\[
\arg \min_{x,s,R} ||Lx||^2 + \alpha \mathcal{E}_{\text{fit}}(x)
\]

s.t. \(f_e(F_p(s), x) = 0\),

\[
\left( \sum_{j \in D} w_j x_j \right) - \sum_{j \in D} w_j X_j = t,
\]

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

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 \(M\) and 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 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 \(x, s\) while keeping \(R, t\) fixed and vice-versa (Figure 11, 12). Rigid transformations do not affect smoothness of \(s\) 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 \(s\). 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 \(K(F_p(s), x)\), as explained above, is now singular with a known 6-dimensional nullspace. In order to compute the Jacobian matrix \(J\) using Equation 10, we use our Singular Lemma (ii) (Section 3.5). Note that the right-hand side is automatically in the range of \(K\) 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 [Barbić and Zhao 2011], or first derivative with respect to each individual entry of \(F\) [Chao et al. 2010; Twigg and Kacic-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 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 \(S = \sqrt{FF^T}\). To compute the first-order derivatives, we start from \(F = RS\), and
differentiate,
\[
\frac{\partial F}{\partial F_i} = \frac{\partial R}{\partial F_i} S + R \frac{\partial S}{\partial F_i}, \quad \text{hence} \quad \frac{\partial R}{\partial F_i} = \left( \frac{\partial F}{\partial F_i} - R \frac{\partial S}{\partial F_i} \right) S^{-1},
\]
(21)

Therefore, we need to compute \( \partial S / \partial F_i \). We have
\[
F^T F = S^2, \quad \text{and thus} \quad \frac{\partial F^T F}{\partial F_i} = \frac{\partial S}{\partial F_i} S + S \frac{\partial S}{\partial F_i},
\]
(22)
i.e., this is the classic Sylvester equation for the unknown matrix \( \frac{\partial S}{\partial F} \) [Sylvester 1884]. The Sylvester equation \( AX + XB = C \) can be solved as
\[
\left( B^T \oplus A \right)^{-1} \text{vec}(X) = \text{vec}(C),
\]
(23)
where \( \oplus \) is the Kronecker sum of two matrices. In our case,
\[
\text{vec}(\frac{\partial S}{\partial F_i}) = (S \oplus S)^{-1} \text{vec}(\frac{\partial F^T F}{\partial F_i}).
\]
(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 \( W_1 x - d_1 = 0 \), 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 \( \text{Polar}(W_2 x - d_2) - \bar{R} = 0 \), 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
Bohan Wang, George Matcuk, and Jernej Barbič

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.

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.

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 \( M \); \#iter=number of ICP iterations; “time”=total computation time to compute the output shape; “attached” means whether the object is attached or not; \( \epsilon_{\text{init}} \) = number of ICP constraints; \( \epsilon_{\text{final}} \) = 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.

<table>
<thead>
<tr>
<th>Example</th>
<th>#vtx</th>
<th>#ele</th>
<th># markers</th>
<th># iter</th>
<th>time [min]</th>
<th>attached</th>
<th>( \epsilon_{\text{init}}[mm] )</th>
<th>( \epsilon_{\text{final}}[mm] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand muscle (min)</td>
<td>4,912</td>
<td>20,630</td>
<td>15</td>
<td>8</td>
<td>14.2</td>
<td>yes</td>
<td>0.53 / 2.22</td>
<td>0.06 / 0.14</td>
</tr>
<tr>
<td>Hand muscle (med)</td>
<td>6,260</td>
<td>31,737</td>
<td>32</td>
<td>12</td>
<td>12.8</td>
<td>yes</td>
<td>0.62 / 1.56</td>
<td>0.11 / 0.55</td>
</tr>
<tr>
<td>Hand muscle (max)</td>
<td>8,951</td>
<td>42,969</td>
<td>96</td>
<td>11</td>
<td>14.2</td>
<td>yes</td>
<td>3.35 / 12.82</td>
<td>0.11 / 0.34</td>
</tr>
<tr>
<td>Hand muscle (max-m)</td>
<td>7,552</td>
<td>34,966</td>
<td>151</td>
<td>18</td>
<td>20.3</td>
<td>yes</td>
<td>3.28 / 9.11</td>
<td>0.16 / 0.47</td>
</tr>
<tr>
<td>Hip muscle (Fig 13)</td>
<td>6,793</td>
<td>34,118</td>
<td>82</td>
<td>21</td>
<td>28.3</td>
<td>yes</td>
<td>7.41 / 21.27</td>
<td>0.39 / 1.85</td>
</tr>
<tr>
<td>Hip bone</td>
<td>6,796</td>
<td>28,740</td>
<td>499</td>
<td>34</td>
<td>49.2</td>
<td>no</td>
<td>4.12 / 14.27</td>
<td>0.25 / 1.30</td>
</tr>
<tr>
<td>Liver</td>
<td>11,392</td>
<td>43,221</td>
<td>424</td>
<td>80</td>
<td>128.3</td>
<td>no</td>
<td>9.00 / 33.68</td>
<td>0.21 / 4.81</td>
</tr>
<tr>
<td>Hand surface (Fig 15)</td>
<td>11,829</td>
<td>49,751</td>
<td>456</td>
<td>31</td>
<td>43.8</td>
<td>no</td>
<td>4.87 / 16.78</td>
<td>0.07 / 0.86</td>
</tr>
</tbody>
</table>

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 (~10 – 20 landmarks and ~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
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.

![Comparison to [Huang et al. 2019]](image)

**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, 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,

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

(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 $\alpha$, and compared them (Figure 23) to our method in 1D.

$$
\min_{f_p} \int_0^1 \frac{\partial^2 f_p}{\partial x^2} dt + \alpha \left( \left( \frac{x}{4} - \frac{1}{4} \right)^2 + \left( \frac{x}{2} - \frac{3}{2} \right)^2 \right)
+ \beta \int_0^1 \left( \frac{\partial f_p}{\partial x} \right)^2 dt
$$

(26)

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

(27)

**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 representative choice. It also has fewest artefacts in Figure 10 of the shape deformation survey [Botsch and Sorkine 2008].
Observe that, in the same vein as in 3D, we can decompose $\dot{\mathbf{x}} = f_e \mathbf{f}_e + f_p \mathbf{f}_p + \beta$, whereby scalar functions $f_e$ and $f_p$ and the $\beta$ 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.

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 $3 \times 4$ affine transformation $A = [F\ b]$ 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 Sorkine 2008], 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).

**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 wiggly artifacts.

**4.6 Comparison to ICP methods**

**Fig. 25. Comparison to $E_{bend}$.** We deformed the surface using the ShapeOp library. Here, $\alpha$ 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 $\alpha$ (such as, for example, $\alpha = 10^4$) cannot satisfy the constraints; and we therefore use $\alpha = 10^5$ and $\alpha = 10^6$ 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). Wiggly artefacts can be observed in the $E_{bend}$ method on the muscle.

Penalizing the differences of affine transformations between neighboring vertices...
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 $E_{\text{bend}}$ 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.

Fig. 27. **Comparison to $E_{\text{bend}}$ 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, $E_{\text{bend}}$ 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 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.

ACKNOWLEDGMENTS
This research was sponsored in part by NSF (IIS-1911224), USC Annenberg Fellowship to Bohan Wang, Bosch Research and Adobe Research.

CONFLICT OF INTEREST STATEMENT
The hip bone and muscle template meshes were obtained at Ziva Dynamics. Jernej Barbic 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.

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.).


\[ \frac{\partial E}{\partial x_i} = V \frac{\partial F_e}{\partial x_i} = VP \frac{\partial F_e}{\partial x_i}, \quad (32) \]
\[ \frac{\partial^2 E}{\partial x_i \partial x_j} = V \frac{\partial^2 F_e}{\partial x_i \partial x_j} = \frac{\partial V}{\partial x_i} \left( \frac{\partial}{\partial x_j} \frac{\partial F_e}{\partial x_i} \right) + V \frac{\partial F_e}{\partial x_i} \frac{\partial^2 F_e}{\partial x_j \partial x_i} + VP \frac{\partial^2 F_e}{\partial x_i \partial x_j} + \frac{\partial^2 V}{\partial x_i \partial x_j}, \quad (33) \]
\[ \frac{\partial^2 E}{\partial x_i \partial x_j} = \frac{V}{\partial x_i} \left( \frac{\partial}{\partial x_j} \frac{\partial F_e}{\partial x_i} \right) + V \frac{\partial F_e}{\partial x_i} \frac{\partial^2 F_e}{\partial x_j \partial x_i} + VP \frac{\partial^2 F_e}{\partial x_i \partial x_j} + \frac{\partial^2 V}{\partial x_i \partial x_j} \frac{\partial F_e}{\partial x_i} \frac{\partial F_e}{\partial x_j}, \quad (34) \]
\[ \frac{\partial^2 E}{\partial x_i \partial x_j} = \frac{V}{\partial x_i} \left( \frac{\partial}{\partial x_j} \frac{\partial F_e}{\partial x_i} \right) + V \frac{\partial F_e}{\partial x_i} \frac{\partial^2 F_e}{\partial x_j \partial x_i} + VP \frac{\partial^2 F_e}{\partial x_i \partial x_j} + \frac{\partial^2 V}{\partial x_i \partial x_j} \frac{\partial F_e}{\partial x_i} \frac{\partial F_e}{\partial x_j}, \quad (35) \]

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 \( \partial^2 E / (\partial x_i \partial x_j) \).

Finally, we have
\[ \begin{align*}
&\frac{\partial^2 E}{\partial x_i \partial x_j} = \frac{\partial^2 V}{\partial x_i \partial x_j} + \frac{\partial^2 F_e}{\partial x_i \partial x_j}, \\
&V \left( \frac{\partial}{\partial x_i} \frac{\partial F_e}{\partial x_j} \right) + \frac{\partial V}{\partial x_i} \left( \frac{\partial}{\partial x_j} \frac{\partial F_e}{\partial x_i} \right) + VP \frac{\partial^2 F_e}{\partial x_i \partial x_j} + \frac{\partial^2 V}{\partial x_i \partial x_j} \frac{\partial F_e}{\partial x_i} \frac{\partial F_e}{\partial x_j},
\end{align*} \]

The quantities \( \psi, P \) 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 \( P \) and \( s \). Because we want to directly optimize \( s \), we also need the derivatives of \( \partial F_e(s, x) \) with respect to \( s \). From Equation 2 we can see that \( F_e \) is linearly dependent on \( s \). Therefore, so we can define a matrix \( Y \) such that \( \text{vec}(F_p) = Ys \).

**C. Proof of Singular Lemma**

Statement (i) follows from well-known linear algebra facts \( R(A) = N(A^T)^\perp \) and \( \dim(N(A)) + \dim(R(A)) = p \), and the symmetry of \( A \). As per (ii), \( A \) maps \( R(A) \) into itself, and no vector from \( R(A) \) maps to zero, hence the restriction of \( A \) to \( R(A) \) 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
\[ \min_x \frac{1}{2} x^T Ax - b^T x \]

s.t. \( \psi_i^T x = 0 \) for all \( i = 1, \ldots, k \).

When expressed using Lagrange multipliers, this gives Equation 15. Suppose \( x = n + r \) is another solution and \( n \in N(A) \) and \( r \in R(A) \). 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 \( 0 = B(n + r) = Ar + \sum_{i=1}^k \lambda_i \psi_i^T n \psi_i \). Observe that the first summand is in \( R(A) \) and the second in \( N(A) \). 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 \( (A + \sum_{i=1}^k \alpha_i \psi_i^T \psi_i) (x + \sum_{i=1}^k \lambda_i \psi_i) \).

**D. Proof of Nullspace Lemma**

We are trying to prove that \( K(F_p(s, x)) \) is 6-dimensional for any \( x \) that solves \( E(s, x) = 0 \). First, if \( x \) 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 \( \psi := (\epsilon_1, \epsilon_2, \ldots, \epsilon_5) \) is in the nullspace of \( K \), where \( \epsilon_i \in \mathbb{R}^3 \) is the \( i \)-th standard basis vector, for \( i = 1, 2, 3 \). Now, suppose we rotate the object with an infinitesimal rotation \( X \mapsto X + \epsilon_1 \times X. \) Observe that for general plastic strains \( s \), the elastic forces in each individual tet are not zero even in the equilibrium \( x \); but the 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 21). This means that the vector of infinitesimal displacements \( \psi_{s,i} := [\epsilon_1 \times x_1, \epsilon_1 \times x_2, \ldots, \epsilon_1 \times x_n] \) induced by the infinitesimal rotation is in the nullspace of \( K \), for each \( i = 1, 2, 3 \). Here, \( x_j \) are the components of \( x = [x_1, x_2, \ldots, x_n] \). The vectors \( \psi_i \), \( i = 1, 2, 3, 4, 5, 6 \), form the nullspace of \( K \).

Finally, we inform the reader that the nullspace of \( K(F_p(s, x)) \) is only 3-dimensional if \( x \) 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

\[
\frac{\partial^2 F}{\partial F_i \partial F_j} = \frac{\partial^2 F}{\partial F_i \partial F_j} S + \frac{\partial^2 S}{\partial F_i \partial F_j} + \frac{\partial F}{\partial F_i} \frac{\partial S}{\partial F_j} + \frac{\partial F}{\partial F_j} \frac{\partial S}{\partial F_i} + \frac{\partial^2 S}{\partial F_i \partial F_j}.
\]

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

\[
\frac{\partial^2 F T F}{\partial F_i \partial F_j} = \frac{\partial^2 S}{\partial F_i \partial F_j} S + \frac{\partial S}{\partial F_i} \frac{\partial S}{\partial F_j} + \frac{\partial S}{\partial F_j} \frac{\partial S}{\partial F_i} + \frac{\partial^2 S}{\partial F_i \partial F_j}.
\]

We can now solve a similar Sylvester equation

\[
\text{vec}(\frac{\partial^2 S}{\partial F_i \partial F_j}) = (S \circ S)^{-1} \text{vec}(C),
\]

\[
C = \frac{\partial^2 F T F}{\partial F_i \partial F_j} - \frac{\partial S}{\partial F_i} \frac{\partial S}{\partial F_j} - \frac{\partial S}{\partial F_j} \frac{\partial S}{\partial F_i}.
\]

F  FORMULAS FOR \( A_k, b_k, c_k \) (EQUATION 6)

Let the attachment \( k \) be embedded into a tetrahedron \( t_k \) with barycentric weights \( [w^k_1, w^k_2, w^k_3, w^k_4] \). We have

\[
A_k = \begin{bmatrix} w^k_1 I_3 & w^k_2 I_3 & w^k_3 I_3 & w^k_4 I_3 \end{bmatrix} S_k \in \mathbb{R}^{3 \times 3n}
\]

\[
b_k = -y_k \in \mathbb{R}^3,
\]

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 \( c_k \) is the weight of the attachment \( k \). Equivalent formulas apply to landmarks and ICP markers. The attachment energy and forces are,

\[
E_a(x) = \frac{1}{2} \sum_k c_k ||A_k x + b_k||^2,
\]

\[
f_a(x) = \frac{dE_a}{dx} = \sum_k c_k A_k^T (A_k x + b_k).
\]

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 \( (K \rightarrow K u) \) when using hard constraints:

\[
\begin{align*}
\min_{u} & < Ku, u > \\
\text{s.t.} & Cu = b
\end{align*}
\]

vs.

\[
\begin{align*}
\min_{u} & < (K u) u, u > \\
\text{s.t.} & Cu = b
\end{align*}
\]

produces same \( u \) for any scalar \( a > 0 \).

Similarly, consider scaling stiffness \( (K \rightarrow K u) \) when using soft constraints:

\[
\begin{align*}
\min_{u} & < (K u) u, u > + \beta ||Cu - b||^2 \\
\text{s.t.} & Cu = b
\end{align*}
\]

produces same \( u \) as

\[
\min_{u} < Ku, u > + \beta ||Cu - b||^2.
\]

Therefore, scaling the stiffness does not provide any more expressive shape deformation power than tweaking \( \beta \) in:

\[
\min_{u} < Ku, u > + \beta ||Cu - b||^2
\]

As one tweaked \( \beta \) 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.