

Real-time Locally Injective Volumetric Deformation

WENTAO LIAO, University of Science and Technology of China, China

RENJIE CHEN*, University of Science and Technology of China, China

YUCHEN HUA, University of Science and Technology of China, China

LIGANG LIU, University of Science and Technology of China, China

OFIR WEBER, Bar-Ilan University, Israel

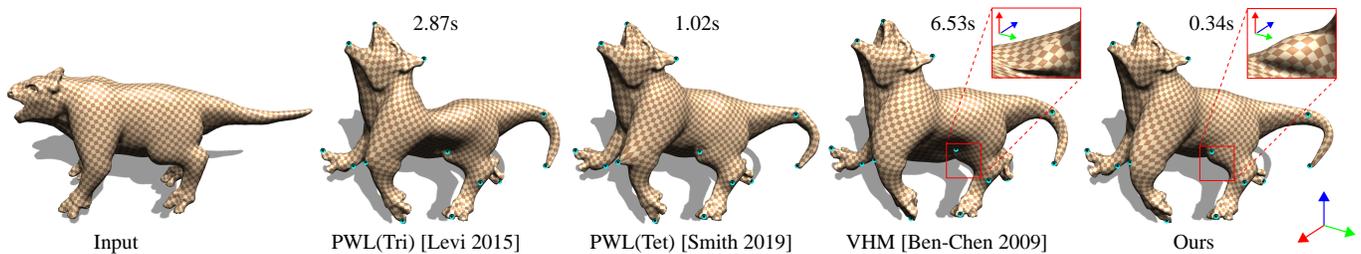


Fig. 1. A comparison of four state-of-the-art shape deformation methods. The listed execution times demonstrate the efficiency of our method compared to the others. Our method guarantees local injectivity and smoothness by construction, and it exhibits low distortion. The PWL (Tri) method is surface-based and is oblivious to the interior of the triangle mesh. As a result, it loses volume drastically at the center of the animal body. The PWL (Tet) method is only 3 times slower than ours. However, due to the use of a coarse tetrahedral mesh, the result is nonsmooth at the tail and foreleg. VHM is smooth, but is 19 times slower than our method. Moreover, similarly to PWL (Tri), it loses volume and fails to be locally injective at the left hind leg. This is more obvious when observed from another angle (shot from below) as shown in the zoom-ins (see also the accompanying video).

We present a highly efficient method for interactive volumetric meshless shape deformation. Our method operates within a low dimensional subspace of shape-aware C^∞ harmonic maps, and is the first method that is guaranteed to produce a *smooth* locally injective deformation in 3D. Unlike mesh-based methods in which local injectivity is enforced on tetrahedral elements, our method enforces injectivity on a sparse set of domain samples. The main difficulty is then to certify the map as locally injective throughout the *entire* domain. This is done by utilizing the Lipschitz continuity property of the harmonic basis functions. We show a surprising relation between the Lipschitz constant of the smallest singular value of the map Jacobian and the norm of the Hessian. We further carefully derive a Lipschitz constant for the Hessian, and develop a sufficient condition for the injectivity certification. This is done by utilizing the special structure of the harmonic basis functions combined with a novel regularization term that pushes the Lipschitz constants further down. As a result, the injectivity analysis can be performed on a relatively sparse set of samples. Combined with a parallel GPU-based implementation, our method can produce superior deformations

*Corresponding author

Authors' addresses: Wentao Liao, University of Science and Technology of China, China, lwt831@mail.ustc.edu.cn; Renjie Chen, University of Science and Technology of China, China, renjie.c@gmail.com; Yuchen Hua, University of Science and Technology of China, China, hyc12908@ustc.edu.cn; Ligang Liu, University of Science and Technology of China, China, lgliu@ustc.edu.cn; Ofir Weber, Bar-Ilan University, Israel, ofir.weber@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

0730-0301/2021/8-ART74 \$15.00

<https://doi.org/10.1145/3450626.3459794>

with unique quality guarantees at real-time rates which were possible only in 2D so far.

CCS Concepts: • **Computing methodologies** → **Computer graphics; Animation.**

Additional Key Words and Phrases: injective maps, harmonic maps, shape deformation, Newton method, GPU

ACM Reference Format:

Wentao Liao, Renjie Chen, Yuchen Hua, Ligang Liu, and Ofir Weber. 2021. Real-time Locally Injective Volumetric Deformation. *ACM Trans. Graph.* 40, 4, Article 74 (August 2021), 16 pages. <https://doi.org/10.1145/3450626.3459794>

1 INTRODUCTION

Shape deformation is central to computer graphics and geometry processing, and is widely used in modeling and animation applications. Material based simulations are used when accurate physical behavior is needed, producing hyper realistic animations with strong dynamic effects. These are popular in the cinema industry, where pure realism is sought. However, its complex setup and heavy computational burden make it infeasible for modeling and interactive scenarios.

In the last two decades, the geometry processing community has been striving to develop shape deformation methods which are physically plausible (as opposed to physically accurate), yet fast enough to be used in interactive applications. Our method belongs to this line of work. The overarching approach is to define an energy functional that aggregates a local geometric distortion measure throughout the deformation domain, and then augment it with a set of constraints. Some constraints steer the deformation based on the user guidance (e.g., positional constraints). Other constraints are

used to ensure quality (e.g., smoothness, bounded distortion, injectivity). This boils down to a variational problem which is discretized and solved as a finite dimensional constrained optimization problem. Such problems are typically nonlinear, nonconvex, and sometimes nonsmooth, and developing an efficient-and-robust solver for them is challenging.

Poranne and Lipman [2014] were the first to bridge the gap between meshless and mesh-based methods in the context of local injectivity in 2D. They bounded the distortion of a smooth planar map, on a set of collocation points, and then established a global distortion bound over the entire domain by estimating the maximal change in the distortion as a function of the distance away from the collocation points. This was achieved mathematically by using the Lipschitz constants of their radial basis functions (RBFs). Later, Chen and Weber [2015] suggested using shape-aware 2D harmonic basis functions (as opposed to RBFs), dramatically improving the deformation quality. They further derived the bounded distortion harmonic mapping theorem [Chen and Weber 2015, Theorem 4], which characterizes the maximal distortion of a planar harmonic map solely based on its boundary behavior. This reduces the dimensionality of the problem from two to one dimension. Hence, the collocation points are placed on the 1D boundary curve of a planar region rather than in its interior.

Chen and Weber [2017] further accelerated the deformation by designing a custom-made solver, utilizing the parallel processing power of a modern GPU. The robustness and real-time performance of their method motivated us to generalize their planar harmonic framework to the volumetric case. Nonetheless, the 3D case poses several mathematical and algorithmic challenges:

- (1) The deformation domain is one dimension higher.
- (2) The lack of suitable analog for the bounded distortion theorem of planar harmonic maps.
- (3) The difficulty of deriving Lipschitz constants for the singular values (of the Jacobian) of a 3D map (harmonic or not).

Challenge (3), which was first raised by [Poranne and Lipman 2014], prohibits a straightforward generalization to 3D. The reason is that in 2D, the singular values of the 2×2 Jacobian have a simple expression in terms of the Jacobian elements, while singular values of a 3×3 matrix are characterized as roots of high order polynomials, for which no straightforward formula exist [Kovalsky et al. 2014].

Our key contribution in this paper is the development of Lipschitz constants for the smallest (signed) singular value σ_3 , which are necessary for the generalization of [Poranne and Lipman 2014] to 3D, hence, overcoming Challenge (3). Positive σ_3 is equivalent to a positive map orientation and positivity throughout the domain allows us to certify the map as locally injective. The derivation of the Lipschitz constant is done thanks to several crucial observations we make in Sections 4 and 5. Remarkably, it turns out that it is possible to compute quite tight Lipschitz constants for σ_3 without forming its explicit expression. We do this by establishing a surprisingly simple connection between the Lipschitz continuity of σ_3 and the norm of the Hessian tensor of the map (Section 4).

In two-dimensions, the Radó-Kneser-Choquet theorem states that a harmonic map of a topological disk that maps its boundary homeomorphically to a unit disk (in fact to any convex shape) is a

diffeomorphism (hence bijective). It is unfortunate however, that in 3D, mapping a three-dimensional topological ball to a unit ball, fails to be a bijection [Laugesen 1996]. Chen and Weber [2015] proved that for locally injective harmonic maps, the maximal distortion is attained on the boundary. This is done by showing that the largest singular value, has a maximum principle, and the smallest singular value has a minimum principle. The immediate theoretical question that comes to mind is whether this holds in 3D as well. In Appendix D, we prove that σ_1 , the largest singular value is subharmonic, hence, has a 3D analogous maximum principle. Nevertheless, we failed to prove a minimum principle for σ_3 , the smallest singular value. [Laugesen 1996] does not shed a light on this question, and we were unable to find an answer in the literature. Yet, after significant investigation and experimentation, we were disappointed to learn that the answer for σ_3 is negative, which leads to challenge (2). Figure 2 illustrates a synthetic counterexample that we constructed.

Challenges (1) and (2) means that for volumetric harmonic maps, the dimension of the domain on which injectivity should be enforced is full, i.e., 3 as opposed to 1 in the planar case, implying that potentially, significantly more collocation points are required. To alleviate that, we promote as sparse as possible sampling (albeit still in 3 dimensions). This is done by 1) Encouraging smoothness 2) Deriving as tight as possible Lipschitz constants 3) Using an adaptive sampling strategy.

In Section 6.1, we introduce a novel smoothness energy term that promotes slow changes of σ_3 throughout the domain and encourages the (optimal) Lipschitz constants to be small. We further provide a novel reformulation for the Variational Harmonic Maps (VHM) [Ben-Chen et al. 2009] in a differential manner (Section 5.1). This makes the (derived) Lipschitz constants invariant to global affine transformations of the cage, making them tighter and closer to the optimal Lipschitz constants. Finally, we devise an adaptive octree-based sampling strategy to certify local injectivity (Section 7.2).

To meet the demand for real-time performance, we further perform a spectral dimensionality reduction of the deformation subspace (Section 6.2), and employ a dynamic weighting technique (Section 7.3), which tends to keep the number of Newton iterations low. Similarly to [Chen and Weber 2017], we implement the optimization on a GPU. The combination of all the above, adds up to a real-time system that can smoothly deform 3D shapes with superior quality. Our method is not only the first that guarantees 3D local injectivity in the smooth (meshless) setting, it is also significantly faster than the alternatives, supporting real-time deformations of complex 3D shapes.

2 PREVIOUS WORK

The literature on geometric shape deformation is voluminous and each method has its own traits. Some methods are suitable for planar deformation only [Chen and Weber 2015, 2017; Chen et al. 2013; Chien et al. 2016a; Igarashi et al. 2005; Levi and Weber 2016; Weber et al. 2009; Weber and Gotsman 2010] while other can be applied in 3D as well. These 3D methods are further split into surface-based methods [Levi and Gotsman 2015; Lipman et al. 2005; Sorkine and Alexa 2007; Sumner and Popović 2004] or spatial (volumetric) methods [Ben-Chen et al. 2009; Huang et al. 2006; Lipman et al. 2008]. The

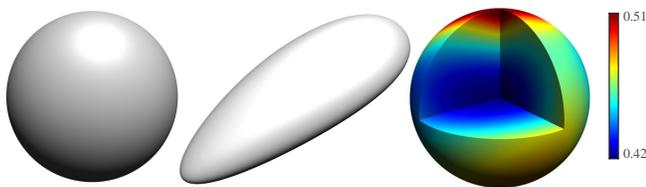


Fig. 2. A local minimum for σ_3 of a bijective harmonic map in \mathbb{R}^3 . (left) A unit ball source domain. (middle) The image of the harmonic map. (right) Color-coding of σ_3 in the ball's interior. The minimum is attained at the center. We provide the closed-form expression of this harmonic map in the supplemental material.

former treats the surface as a two-dimensional thin shell embedded in \mathbb{R}^3 , ignoring its volume. These methods typically assume that the input surface is represented as a manifold triangle mesh. On the other hand, the advantage of the spatial methods is that they are applied to a full dimensional subspace of \mathbb{R}^3 . Hence, any object embedded in this subspace, can be deformed regardless of its geometric representation (e.g., general polygonal meshes, point clouds, non-manifold inputs, etc.). Moreover, since the entire subspace is deformed rather than its boundary, these methods tend to better preserve volume.

Most spatial 3D methods discretize the interior of the shape using a tetrahedral mesh [Aharon et al. 2019; Aigerman and Lipman 2013; Chao et al. 2010; Kovalsky et al. 2014, 2015, 2016; Rabinovich et al. 2017; Smith et al. 2019]. The map is assumed to be continuous and linear on each tetrahedron. Such a piecewise linear (PWL) subspace is easy to construct, yet, the deformation is C^0 , hence nonsmooth, regardless of the amount of elements being used. Furthermore, the execution times grow dramatically when fine tetrahedral meshes are used as demonstrated in Figure 3.

Meshless methods on the other hand, avoid discretization of the volume, and typically use basis functions which are smooth in the interior of the domain [Adams et al. 2008; Floater et al. 2005; Huang et al. 2006; Ju et al. 2005; Li et al. 2013; Sederberg and Parry 1986; Thiery et al. 2014]. In particular, harmonic meshless methods use a linear combination of predefined C^∞ shape-aware harmonic basis functions, resulting in a naturally looking smooth maps. In [Ben-Chen et al. 2009; Lipman et al. 2008], the harmonic subspace is constructed by using a triangle mesh that fully encapsulates the deformation domain (the so-called cage) and integrating the fundamental solution of the Laplace equation on its elements. [Joshi et al. 2007] also uses harmonic basis functions, but discretizes using the finite element method. In [Jacobson et al. 2011; Weber et al. 2012b] biharmonic functions are used.

One of the most desirable properties of a shape deformation algorithm is its ability to guarantee that the map is injective (local injectivity is typically sufficient). Nonetheless, such a constraint is highly nonconvex. A plethora of methods emerged in the last decade to handle the local injectivity requirement using various techniques [Chen and Weber 2015, 2017; Chen et al. 2013; Chien et al. 2016a,b; Levi and Weber 2016; Lipman 2012; Schüller et al. 2013; Smith and Schaefer 2015; Weber and Gotsman 2010; Weber et al. 2012a]. In 3D, the research on local injectivity was focused so

far solely on (nonsmooth) mesh-based techniques. Local injectivity is obtained either by forcing an orientation preserving (non-flip) constraint on a per tetrahedron basis [Aigerman and Lipman 2013; Kovalsky et al. 2014, 2015] or by using the interior point method [Fu et al. 2015; Rabinovich et al. 2017; Smith et al. 2019] in which a barrier function goes to infinity when a tetrahedron collapses. Hence, the orientation of the tetrahedra is preserved.

To the best of our knowledge, our meshless method is the first method that produces smooth (C^∞) 3D maps that are *guaranteed* to be locally injective. Ensuring the injectivity and/or bounding the distortion of a meshless map is difficult since there is an infinite number of points on which the (nonconvex) constraints should hold.

3 BACKGROUND

3.1 Problem Statement

We describe a handle-based deformation framework that deforms a closed domain $\Omega \subset \mathbb{R}^3$ with a smooth (at least C^2) map $f : \Omega \rightarrow \mathbb{R}^3$. The map is obtained as the solution to the general constrained minimization problem:

$$\min_f E_{\text{dis}} + \lambda E_{P2P}, \quad (1a)$$

$$\text{s.t. } \sigma_3(p) > 0, \quad \forall p \in \Omega, \quad (1b)$$

where $\sigma_3(p)$ is the smallest *signed* singular value of the 3×3 Jacobian of f . E_{dis} measures the overall map distortion, and E_{P2P} is a *soft* positional constraints term. The condition $\sigma_3(p) > 0$ is equivalent to $\det(J(p)) > 0$ and implies that the orientation of the map at p is preserved. The map f is locally injective if and only if Condition (1b) holds.

We use the popular symmetric Dirichlet (SD) isometric measure [Smith and Schaefer 2015]: $E_{\text{iso}} = \sum_{i=1}^3 (\sigma_i^2 + \sigma_i^{-2})$, where σ_i is the i^{th} singular value. The overall distortion is defined by integrating the pointwise isometric measure E_{iso} over the domain Ω :

$$E_{\text{dis}} = \int_{\Omega} E_{\text{iso}}(p) dV. \quad (2)$$

The advantage of using E_{iso} is that it becomes infinite if $\sigma_3(p) = 0$ for some $p \in \Omega$, hence, in practice, a proper line search can be used to ensure Condition (1b) rather than explicitly enforcing it in the optimization. We solve the simpler unconstrained minimization with a projected Newton solver [Smith et al. 2019] which is needed due to the fact that the energy is nonconvex and its Hessian is not necessarily positive definite.

3.2 Harmonic Basis Functions

Due to their smoothness, low dimensionality, and shape-awareness, harmonic maps have been shown to be an excellent choice for shape deformation [Ben-Chen et al. 2009; Chen and Weber 2015, 2017; Chien et al. 2016a; Joshi et al. 2007; Lipman et al. 2008]. Here we briefly review the 3D harmonic subspace construction of [Ben-Chen et al. 2009].

A smooth function $h(x, y, z) : \Omega \rightarrow \mathbb{R}$ is called harmonic if it satisfies the Laplace's equation:

$$\Delta h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} + \frac{\partial^2 h}{\partial z^2} = 0,$$

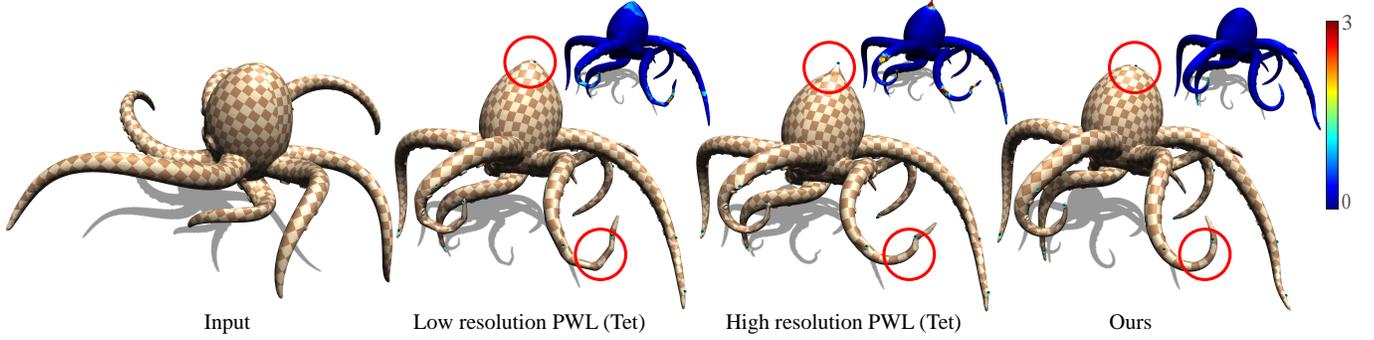


Fig. 3. A comparison of our meshless method with the mesh-based PWL (Tet) method [Smith et al. 2019] using two mesh resolutions. Execution time on the low (high) resolution tetrahedral mesh with 10,180 (93,911) vertices is 2.72s (110.39s). Notice the lack of smoothness in the low resolution result, and the concentration of the distortion near the handles in the high resolution result. Our method with 900 DOFs converges after 0.49 seconds and is 225 times faster compared to the high resolution PWL (Tet) result. The result is smooth and the distortion is distributed more evenly.

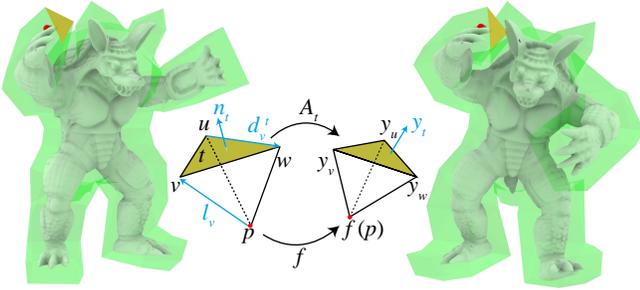


Fig. 4. Notations. On the left, a triangle mesh cage $\partial\Omega$ (in green) bounding Ω . One triangle (denoted t) with vertices u, v, w is highlighted (yellow) and zoomed-in at the middle along with its image triangle with vertices y_u, y_v, y_w . The point $p \in \Omega$ is mapped to $f(p)$. The vector coefficients y_{v_i} and y_{t_j} are defined on $\partial\Omega$. n_t is the unit normal vector of triangle t . l_v denotes the vector from p to v and d'_v denotes the vector opposite to v on t . A_t is the Jacobian of the affine transformation which maps the local coordinate system $\{d'_v, d'_u, n_t\}$ to $\{y_w - y_u, y_v - y_w, y_t\}$.

where Δ is the Laplacian operator. A map $f = (f^x(p), f^y(p), f^z(p))$, $p \in \mathbb{R}^3$ is called harmonic if f^x, f^y, f^z are harmonic functions. We assume that the boundary of Ω is a triangle mesh (the so-called cage) $M = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, with vertices \mathcal{V} , directed edges \mathcal{E} , and faces \mathcal{F} . We use the same linear subspace of harmonic maps that was used in [Ben-Chen et al. 2009; Lipman et al. 2008], namely:

$$f(p) = \sum_{v_i \in \mathcal{V}} y_{v_i} \phi_{v_i}(p) + \sum_{t_j \in \mathcal{F}} y_{t_j} \psi_{t_j}(p), \quad (3)$$

where $\phi_{v_i} : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $\psi_{t_j} : \mathbb{R}^3 \rightarrow \mathbb{R}$ are real-valued harmonic basis functions associated with vertex v_i and triangle t_j respectively. $y_{v_i} \in \mathbb{R}^3$ can be interpreted as the vertex positions of a deformed cage, and $y_{t_j} \in \mathbb{R}^3$ as “normal” vectors for the target cage (though these “normals” are orthogonal to its faces only for the identity map). See Figure 4 for notations. The Jacobian and the Hessian of f

are:

$$J(p) = \sum_{v_i \in \mathcal{V}} y_{v_i} \nabla \phi_{v_i}(p)^T + \sum_{t_j \in \mathcal{F}} y_{t_j} \nabla \psi_{t_j}(p)^T, \quad (4)$$

$$H(p) = \sum_{v_i \in \mathcal{V}} y_{v_i} \circ H_{\phi_{v_i}}(p) + \sum_{t_j \in \mathcal{F}} y_{t_j} \circ H_{\psi_{t_j}}(p). \quad (5)$$

Throughout the paper we use the product $v \circ M$ with $v \in \mathbb{R}^3$ and $M \in \mathbb{R}^{3 \times 3}$ to denote a tensor such that $(v \circ M)_{ijk} = v_i M_{jk}$. Similarly, the notation $M \circ v$ denotes a tensor such that $(M \circ v)_{ijk} = M_{ij} v_k$. The expressions of the basis functions ϕ_{v_i} and ψ_{t_j} , as well as their differentials, are given in [Ben-Chen et al. 2009]. We provide them in a slightly different form in Appendix A.

3.3 Local Injectivity in the Meshless Framework

As explained in Section 3.1, the local injectivity condition (Equation (1b)) is verified in the line search step rather than added explicitly to the optimization as a constraint. Nonetheless, in the meshless setting, there is an infinite number of points in Ω on which the condition should be verified. Since it is infeasible to verify the condition on an infinite set, in practice, Condition (1b) is verified on a finite set of samples \mathcal{I} in Ω , and then implied for the entire domain by using additional sufficient conditions. The idea is to estimate the maximal amount of change in σ_3 in-between the samples. To this end, the concept of Lipschitz continuity becomes handy.

Definition 3.1. A real function $g: \Omega \rightarrow \mathbb{R}$ is called L -Lipschitz continuous if there exists a constant $L \geq 0$, such that $\forall p, q \in \Omega$:

$$|g(p) - g(q)| \leq L \|p - q\|.$$

Any such L is referred to as a Lipschitz constant of g .

Intuitively, the rate of change of a Lipschitz continuous function never exceed L . In particular, a differentiable function $g(p)$ with a bounded derivative, is Lipschitz continuous with a tight Lipschitz constant $L = \sup_{p \in V} \|\nabla g(p)\|$ in any subdomain $V \subset \Omega$.

Let $V_{p_i}^r$ be a voxel (a cube) with center p_i and radius r , where r is the distance from p_i to one of the voxel’s corners. We partition Ω into voxels with identical radius r . Let \mathcal{I} be the set of all voxel centers. f is certified as locally injective over Ω if we can verify that

σ_3 is *strictly* positive in all the voxels $V_{p_i}^r$. Let us denote by $L_{p_i}^r$, the Lipschitz constant of σ_3 in $V_{p_i}^r$. For any point $p \in V_{p_i}^r$, the following inequality holds:

$$|\sigma_3(p_i) - \sigma_3(p)| \leq L_{p_i}^r \|p_i - p\| \leq r L_{p_i}^r,$$

which implies:

$$\sigma_3(p) \geq \sigma_3(p_i) - r L_{p_i}^r.$$

Hence, in order to ensure that $\sigma_3(p)$ is positive everywhere inside voxel $V_{p_i}^r$, it is sufficient to evaluate σ_3 at the voxel's center, p_i , and ensure that $\sigma_3(p_i) > r L_{p_i}^r$. To conclude, in order to certify the map f as locally injective, we first estimate the Lipschitz constants $L_{p_i}^r$ on all the voxels. We then evaluate $\sigma_3(p_i)$ at the voxel centers $p_i \in \mathcal{I}$, and verify that:

$$\sigma_3(p_i) > r L_{p_i}^r \quad \forall p_i \in \mathcal{I}. \quad (6)$$

The above condition is sufficient but not necessary. One straightforward way to avoid false conclusions regarding the lack of injectivity when the condition fails, is to reduce r . This however, results in more computations due to increased sampling density. Hence, it is desirable that $\sigma_3(p)$ will be as smooth (in the Lipschitz sense) as possible. In addition, we strive to derive Lipschitz constants which are as tight as possible.

4 A LIPSCHITZ CONSTANT FOR σ_3

As discussed in Section 3.3, the Lipschitz constant $L_{p_i}^r$ of σ_3 for a given voxel $V_{p_i}^r$ can be any value greater than $\max_{p \in V_{p_i}^r} \|\nabla \sigma_3(p)\|$. This approach seems to be straightforward, however, $\sigma_3(p)$ is not necessarily smooth everywhere (for example the gradient is not well-defined at points where $\sigma_2(p) = \sigma_3(p)$). Moreover, even in regions where $\sigma_3(p)$ is smooth, it is, in general, impossible to compute the value of $\|\nabla \sigma_3\|$ analytically, because $\sigma_i, i = 1, 2, 3$ do not possess tractable closed-form expressions. This is in sharp contrast to the 2D case, where simple expressions for the singular values exist, allowing effective estimation of its Lipschitz constant [Chen and Weber 2017; Poranne and Lipman 2014].

To alleviate this difficulty, we searched for alternative ways for obtaining a Lipschitz constant of σ_3 without ever evaluating $\nabla \sigma_3$ or σ_3 explicitly. A key result in this work is the development of a surprisingly simple, but quite tight, Lipschitz constant for σ_3 . Moreover, this result can be used in order to compute a Lipschitz constant for all three singular values (though we didn't bound σ_1 and σ_2 in this work). In fact, one can even use it in higher dimensions, and it is applicable to any kind of C^2 map, not necessarily harmonic. We express this result in the following theorem. For brevity, we formulate the theorem only for dimension 3. Nonetheless, the generalization is trivial.

Theorem 4.1. Let $f : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be a C^2 map over a convex region with a 3×3 Jacobian J , and a 3rd order tensor Hessian H . Let $\|H\| = \sqrt{\sum_{i,j,k=1,2,3} H_{ijk}^2}$ denotes the ‘‘Frobenius’’ norm of the 3rd order tensor H .

Then, $\max_{v \in \Omega} \|H(v)\|$ is a Lipschitz constant for σ_1 , σ_2 , and σ_3 .

PROOF. We start by stating the following result by Golub and Van Loan [1996, Corollary 8.6.2]:

If A and E are in $\mathbb{R}^{m \times n}$, and σ_i is the i^{th} singular value $i = 1 \dots n$ then:

$$|\sigma_i(A + E) - \sigma_i(A)| \leq \sigma_1(E) = \|E\|_2. \quad (7)$$

By substituting $A = J(q) \in \mathbb{R}^{3 \times 3}$ and $A + E = J(p) \in \mathbb{R}^{3 \times 3}$, where $J(p)$ denotes the Jacobian of the map at p , and using the fact that the Frobenius norm of a matrix is always larger than its l_2 norm, we obtain:

$$|\sigma_i(p) - \sigma_i(q)| \leq \|J(p) - J(q)\|_F. \quad (8)$$

Equation (7) is stated for unsigned singular values. In Appendix B we generalize it to signed singular values which allows us to apply it to our σ_3 . Furthermore, since our map is harmonic, $J(p)$ is smooth, hence, by the multivariable mean value theorem [Pugh 2015]:

$$\|\text{vec}(J(p) - J(q))\| \leq \|p - q\| \max_{v \in \overline{pq}} \|\tilde{H}(v)\|_2, \quad (9)$$

where \overline{pq} is the line segment connecting p and q , $\text{vec}(M)$ is the vectorization of the matrix M (i.e. stacking the columns of M into a long vector), $\tilde{H} = [H^1, H^2, H^3]$ with H^i being the i^{th} slice of the tensor H . Let Ω be a convex region (e.g. a voxel). From Equations (8) and (9), and the fact that $\|J(p) - J(q)\|_F = \|\text{vec}(J(p) - J(q))\|$ and $\|\tilde{H}\|_2 \leq \|\tilde{H}\|_F = \|H\|$, we obtain:

$$\forall p, q \in \Omega, \quad |\sigma_i(p) - \sigma_i(q)| \leq \|p - q\| \max_{v \in \Omega} \|H(v)\|. \quad (10)$$

Hence, $\max_{v \in \Omega} \|H(v)\|$ is a Lipschitz constant for σ_i (in particular σ_3) in Ω . \square

5 BOUNDING $\|H\|$

In Section 4, we showed that obtaining a Lipschitz constant for σ_3 boils down to bounding $\|H\|$. In this section, we show how to compute an upper bound on the Hessian norm. Using Equation (5), and the fact that the ‘‘Frobenius’’ tensor norm is submultiplicative, the approach of Poranne and Lipman [2014] suggests the following straightforward upper bound:

$$\|H(p)\| \leq \sum_{v_i \in \mathcal{V}} \|y_{v_i}\| \|H_{\phi_{v_i}}(p)\|_F + \sum_{t_j \in \mathcal{F}} \|y_{t_j}\| \|H_{\psi_{t_j}}(p)\|_F. \quad (11)$$

With this, we can obtain a bound for $\|H\|$ by bounding the Hessians of the individual basis functions. While simple, this inequality is far from being tight and turned out to be quite useless in practice when a sparse sampling density is used. One reason for this is that, while $\|H\|$ is invariant to affine transformations of the target cage (coefficients y_{v_i} and y_{t_j} in Equation (3)), the right-hand side of Equation (11) isn't. Assume for instance, that the user wishes to translate the shape by a large amount. This simply amounts to addition of a constant to all the coefficients y_{v_i} . As a consequence, the right-hand side of Equation (11) will grow substantially. This makes no sense, as such a translation shouldn't affect σ_3 at all.

5.1 The Differential Map Representation

Our goal in this section is to develop, as tight as possible, bound for $\|H\|$, and as a consequence, a Lipschitz constant for σ_3 which is invariant to affine transformations of the cage. To this end, we develop an alternative novel ‘‘differential’’ representation of the

VHM barycentric mapping. By differential, we mean that the expression is a linear combination of the differentials of the y_{v_i} and y_{t_j} coefficients, rather than their actual values.

Consider an affine transformation that transforms a triangle t of the source cage to a triangle of the target cage. Moreover, assume that it transforms the normal n_t (representing the orthogonal space) to a vector y_t (the target “normal”). Notations and a schematic are given in Figure 4. Let A_t be the differential (a 3×3 matrix) of such an affine transformation:

$$A_t = \begin{bmatrix} y_v - y_w & y_w - y_u & y_t \\ d_u^t & d_v^t & n_t \end{bmatrix}^{-1} \in \mathbb{R}^{3 \times 3}, \quad (12)$$

where $d_u^t = v - w$, and $d_v^t = w - u$. We provide a small Lemma (the proof is in Appendix C) that will be handy later on.

Lemma 5.1. The linear part, A_t , of the affine transformation can be expressed alternatively as follows:

$$A_t = y_t n_t^T - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} \left(n_t \times d_{v_i}^t \right)^T, \quad (13)$$

where s_t is the signed area of triangle t .

Let t_1, t_2 be the two triangles adjacent to edges e and e' , where e and e' are opposite edges and $e \in t_1, e' \in t_2$. The notation δ_A^e stands for a 3×3 matrix such that $\delta_A^e = A_{t_1} - A_{t_2}$ (Similarly, $\delta_A^{e'} = A_{t_2} - A_{t_1}$). Intuitively, δ_A^e is the “differential” of A_t , or the second order “differential” of y_{v_i}, y_{t_j} . Using Lemma 5.1, we can obtain the following Theorem (see Appendix E for the proof) that allows us to substitute the straightforward representation of $H(p)$ (Equation (5)) with a “differential” representation.

Theorem 5.2. The Hessian tensor of the VHM harmonic map can be expressed using the following alternative representation:

$$H(p) = \frac{1}{2} \sum_{e \in \mathcal{E}} \left(\delta_A^e \circ \alpha_e(p) + \left(\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \beta_e(p) \right), \quad (14)$$

where $\alpha_e(p)$ and $\beta_e(p)$ are the vector functions α_v^t and β_v^t from Appendix A, just notated differently. Specifically, the subscript e corresponds to the edge opposite to vertex v in triangle t . Similarly, d_e is d_v^t written in a different form. The notation $[v]_{\times}$ stands for a 3×3 skew symmetric matrix, such that for any vector $w \in \mathbb{R}^3$, the matrix-vector product $[v]_{\times} w$ produces the vector $v \times w$.

5.2 A Tight Bound for $\|H\|$

We now utilize the differential representation of $H(p)$ (Theorem 5.2) in order to derive a tighter upper bound for $\|H\|$. From Equation (14) and the subadditivity of the tensor norm:

$$\|H(p)\| \leq \frac{1}{2} \sum_{e \in \mathcal{E}} \left\| \delta_A^e \circ \alpha_e + \left(\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \beta_e \right\|.$$

Note that the corresponding rows from δ_A^e and $\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times}$ are orthogonal to each other, hence, we have:

$$\begin{aligned} \|H(p)\| &\leq \frac{1}{2} \sum_{e \in \mathcal{E}} \sqrt{\left\| \delta_A^e \circ \alpha_e \right\|^2 + \left\| \left(\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \beta_e \right\|^2} \\ &\leq \frac{1}{2} \sum_{e \in \mathcal{E}} \sqrt{\left\| \delta_A^e \right\|_F^2 \|\alpha_e\|^2 + \left\| \delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right\|_F^2 \|\beta_e\|^2}. \end{aligned}$$

Furthermore, δ_A^e and $\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times}$ have the same norm, thus:

$$\begin{aligned} \|H(p)\| &\leq \frac{1}{2} \sum_{e \in \mathcal{E}} \left\| \delta_A^e \right\|_F \sqrt{\|\alpha_e\|^2 + \|\beta_e\|^2} \\ &= \frac{1}{2} \sum_{e \in \mathcal{E}} \left\| \delta_A^e \right\|_F \|\alpha_e - \beta_e\|. \end{aligned} \quad (15)$$

The equality is due to that α_e and β_e are orthogonal vectors (see Equation (42)). By denoting the vector $\gamma_e(p)$ as the difference $\alpha_e(p) - \beta_e(p)$, we finally obtain the Lipschitz constant for the i^{th} voxel:

$$L_{p_i}^r = \max_{p \in V_{p_i}^r} \frac{1}{2} \sum_{e \in \mathcal{E}} \left\| \delta_A^e \right\|_F \|\gamma_e(p)\|. \quad (16)$$

For every voxel, and every edge, we compute the constant $\Gamma_i^e = \frac{1}{2} \max_{p \in V_{p_i}^r} \|\gamma_e(p)\|$ once during preprocessing. Finally, the Lipschitz constant for the i^{th} voxel is given by:

$$L_{p_i}^r = \sum_{e \in \mathcal{E}} \Gamma_i^e \left\| \delta_A^e \right\|_F. \quad (17)$$

The advantages of having $\left\| \delta_A^e \right\|_F$ in Equation (17) are:

- (1) $\left\| \delta_A^e \right\|_F$ vanishes if f is a global affine map.
- (2) The scalar function $\|\gamma_e(p)\|$ is local and decays fast as a function of the distance from edge e .
- (3) $\left\| \delta_A^e \right\|_F$ is invariant to compositions of f with affine transformations.

5.3 Higher Order Estimation

While Equation (17) provides a fairly tight $L_{p_i}^r$, we can further tighten the estimation using the third order derivative of f . Given $p_i \in \mathcal{I}$ and $p \in V_{p_i}^r$, we have:

$$\begin{aligned} \|H(p)\| &\leq \|H(p_i)\| + \|H(p) - H(p_i)\| \\ &\leq \|H(p_i)\| + \max_{p \in V_{p_i}^r} \left\| J_f^3(p) \right\| r, \end{aligned} \quad (18)$$

where $J_f^3(p)$ is a 4th order tensor whose components are the 3rd order derivatives of f , i.e., the “Jacobian” of H :

$$J_f^3(p) = \frac{1}{2} \sum_{e \in \mathcal{E}} \left(\delta_A^e \circ \nabla \alpha_e + \left(\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \nabla \beta_e \right). \quad (19)$$

Similar to inequality (15), we can derive an upper bound for $\|J_f^3(p)\|$ from Equation (19):

$$\|J_f^3(p)\| \leq \frac{1}{2} \sum_{e \in \mathcal{E}} \left\| \delta_A^e \right\|_F \sqrt{\|\nabla \alpha_e\|^2 + \|\nabla \beta_e\|^2}. \quad (20)$$

Appendix H provides an upper bound for $\|\nabla \alpha_e\|^2 + \|\nabla \beta_e\|^2$. The important thing to note here is that this upper bound depends solely on the geometries of the source cage, and the voxel. Since these are fixed throughout the optimization, they are computed once during preprocessing.

To sum up, the Lipschitz constants of our algorithm are determined as follows. For each sample $p_i \in \mathcal{I}$, we evaluate $\|H(p_i)\|$ and $\max_{p \in V_{p_i}^r} \left\| J_f^3(p) \right\|$, then $L_{p_i}^r$ is defined as:

$$L_{p_i}^r = \|H(p_i)\| + \max_{p \in V_{p_i}^r} \left\| J_f^3(p) \right\| r. \quad (21)$$

Figure 5 (left graph) compares the different estimations of $L_{p_i}^r$ given by Equations (17) and (21). As anticipated, using the 3rd order derivatives leads to a significantly smaller $L_{p_i}^r$ which are close to the optimal ground truth bounds. The ground truth is estimated by computing the maximum of $\|\nabla\sigma_3\|$ over 1000 samples in each voxel. In addition to showing the superiority of the higher order bounds of Equation (21) over Equation (17), this experiment demonstrates the tightness of the bound obtained by Theorem 4.1.

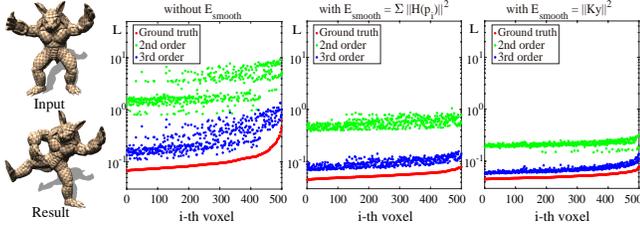


Fig. 5. A comparison of Lipschitz constants computed with Equations (17) and (21). In the left graph, no smoothness energy is used. In the middle, the VHM smoothness energy is used, while on the right, we use $E_{\text{smooth}} = \|Ky\|^2$. To avoid clutter, the plots include only the values for the 500 points with the largest 3rd order Lipschitz constants (21). The values are sorted according to the ground truth values (red).

6 SMOOTHNESS ENFORCEMENT

Condition (6) is a sufficient but not a necessary condition for local injectivity. In order to avoid cases in which Condition (6) fails, yet the map is injective, it is desirable to either have as small as possible Lipschitz constants, or use a smaller sampling radius r . Chen and Weber [2017] used dense sampling of the 1D boundary curve in order to evaluate the symmetric Dirichlet energy, which promotes injectivity by design. In contrast, we must sample the 3D space, and it is computationally infeasible to use high density sampling for a cubic domain. This implies that our sampling density needs to be significantly lower compared to the planar case. Note that even though harmonic mappings are C^∞ , their differentials may have large norm and can potentially cause the mapping to reverse orientation locally, in-between the samples. Hence, we need a strategy to encourage the mapping to be locally injective away from the samples.

Theorem 4.1 indicates that $\max_{p \in V_{p_i}^r} \|H(p)\|$ is a valid Lipschitz constant $L_{p_i}^r$. Therefore, it makes sense to enhance the smoothness of f , as measured by $\|H\|$, by introducing a regularization energy term, so that the obtained $L_{p_i}^r$ can be as small as possible:

$$E = E_{\text{dis}} + \lambda_1 E_{\text{P2P}} + \lambda_2 E_{\text{smooth}}. \quad (22)$$

λ_1 and λ_2 are user-defined weights which balance the three energy terms. Whereas Section 5.2 deals with obtaining a tight Lipschitz constant for σ_3 , in this section we strive to push down the *optimal* Lipschitz constant. The combination of these two steps leads to a Lipschitz constant that is as small as possible (Figure 5).

To see the effect of our smoothness energy, a comparison between the results with and without E_{smooth} is shown in Figure 6. Clearly

the smoothness energy makes it possible for our algorithm to successfully certify local injectivity with much sparser samplings. In contrast, without E_{smooth} , the optimization with the exact same set of samples, terminates prematurely due to failure of the injectivity certification step.

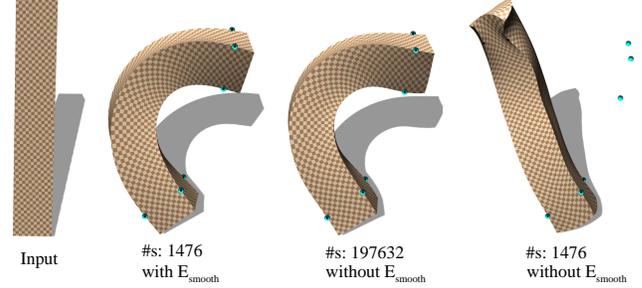


Fig. 6. A comparison of results computed with and without the smoothness energy term. The constructed harmonic mapping space has 450 DOFs. With the smoothness term included, only 1,476 samples are needed in order to produce the result and the runtime is 0.17s. Without the smoothness term, we can observe that the result (right) fails to satisfy the P2P constraints. This problem can be avoided by using a much denser sampling (197,632 samples). The result is visually identical, albeit, the execution time (12.2s) is X70 times longer.

6.1 The Smoothness Energy

We first rewrite Equations (3) and (4) as follows:

$$[f(p)]^T = D_p y \quad (23)$$

$$[J(p)]^T = J_p y, \quad (24)$$

where $D_p = [\Phi, \Psi]$, $\Phi \in \mathbb{R}^{1 \times |\mathcal{V}|}$ is a row vector with $\Phi_i = \phi_{v_i}$, $\Psi \in \mathbb{R}^{1 \times |\mathcal{F}|}$ is a row vector with $\Psi_j = \psi_{t_j}$. $J_p = [\nabla\Phi, \nabla\Psi]$ is the gradient of D_p . y is a $(|\mathcal{V}| + |\mathcal{F}|) \times 3$ matrix, concatenating y_{v_i} and y_{t_j} . It can be seen from Equation (13) that A_t is linear in y (since n_t and d_t^e are constants that depend solely on the source cage). Hence, $\delta_A^e = A_{t_1} - A_{t_2} \in \mathbb{R}^{3 \times 3}$ is also linear in y (t_1, t_2 are the adjacent triangles sharing edge e). Therefore, we construct a sparse $3|\mathcal{E}| \times (|\mathcal{V}| + |\mathcal{F}|)$ matrix K such that:

$$\begin{bmatrix} \delta_A^{e_1} & \delta_A^{e_2} & \dots & \delta_A^{e_{|\mathcal{E}|}} \end{bmatrix}^T = Ky. \quad (25)$$

Equation (14) shows that $H(p)$ is a “linear” combination of δ_A^e , hence there must exist a constant c , which is independent of p and y , such that for any point $p \in \Omega$, the following inequality holds:

$$\|H(p)\| \leq c \|Ky\|. \quad (26)$$

An estimation for the constant c is given in Appendix F. In order to push down $\|H(p)\|$ over the entire domain, we suggest to push the right-hand side of Equation (26) instead, therefore:

$$E_{\text{smooth}} = \|Ky\|^2. \quad (27)$$

Ben-Chen et al. [2009] proposed a different smoothness energy based on the maximum principle of harmonic functions. Their energy essentially integrates $\|H\|^2$ over $\partial\Omega$, the boundary of the volumetric domain. However, the integral is approximated with a sum

$\sum_i \|H_i\|^2$ over boundary samples, and optimizing it only guarantees that $\|H\|$ is small on these samples. In contrast, our smooth energy is independent of any sampling and promotes smoothness throughout the entire domain evenly (as the constant c is independent of p). Figure 5 compares the smoothness energy of Ben-Chen et al. with ours. It is evident that both smoothness energies encourage the Lipschitz constants to be smaller, albeit, our smoothness term is more effective in doing so.

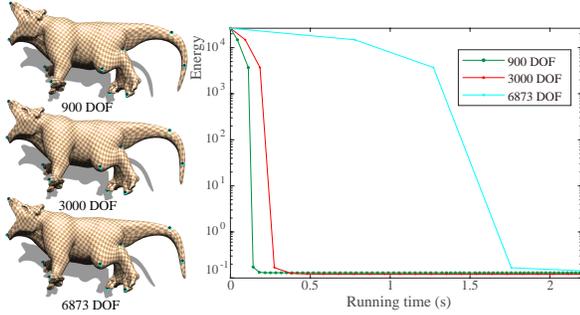


Fig. 7. Subspace dimensionality reduction on the animal shape. (right) The graphs illustrate that approximately, the same energy level is reached when the dimension of the subspace is reduced from 6,873 (a cage with 765 vertices and 1,526 triangles) to 900. The runtime on the other hand is reduced by 90%. (left) The actual deformations are also quite similar.

6.2 Spectral Subspace Dimensionality Reduction

Our optimization problem is formulated in the subspace spanned by the harmonic basis functions. Therefore, the dimension of this subspace is determined by the discretization of the cage. In order to capture the shape well, a high-resolution cage is required, which leads to a large number of DOFs. This severely affect the efficiency of our algorithm. Since the deformation complexity is often decoupled from the shape complexity, we opt to reduce the dimension of our subspace using spectral analysis. By doing so, we can greatly reduce the computational cost without dramatically sacrificing quality.

In order to promote smoothness, we first perform an eigenvalue decomposition of the Hessian of the smoothness term, $K^T K$, and find s eigenvectors $\Xi_s = [\xi_1, \xi_2, \dots, \xi_s]$ that correspond to the s smallest eigenvalues. We then restrict our variables y to be in the subspace spanned by Ξ_s such that $y = \Xi_s \tilde{y}$, where \tilde{y} becomes our new optimization variable. Figure 7 demonstrates that we can use much less DOFs without losing visual fidelity. This behavior is consistent across a wide range of models.

7 IMPLEMENTATION

The building blocks of our method are outlined in Algorithm 1. The main body of the loop includes a Newton iteration that computes the update direction, and a line search procedure that computes a proper step size along the update direction that, 1) satisfies the Wolfe condition [Nocedal and Wright 2006, Chapter 3.1], 2) certifies the map as locally injective (by checking Condition (6)). An energy decreasing step size always exist since the energy Hessian is forced to be positive definite. Moreover, since the previous step is assumed

Algorithm 1 Locally Injective Volumetric Deformation

Input: y ▷ initialization - locally injective map
 $p_i \in \mathcal{P} \subset \Omega, \quad q_i \in \mathbb{R}^3$ ▷ P2P constraints
Output: y_{opt} ▷ new locally injective map

```

1: PREPROCESS()
2:  $y_0 = y, E_0 = \text{COMPUTE\_ENERGY}(y_0), n = 0$ 
3: loop ▷ Newton iteration
4:    $(\nabla^2 E_n)^+ \delta y = -\nabla E_n$  ▷ solve linear system
5:    $(y_{n+1}, E_{n+1}, t) \leftarrow \text{LINE\_SEARCH}(E_n, y_n, \nabla E_n, \delta y)$ 
6:   if  $t \|\delta y\| \leq \epsilon$  then ▷ solver converged
7:     return  $y_{n+1}$ 
8:   else
9:      $n \leftarrow n + 1$ 
10:  end if
11: end loop

```

```

12: procedure LINE_SEARCH( $E_n, y_n, \nabla E_n, \delta y$ )
13:    $t \leftarrow 1$ 
14:   repeat
15:      $y_{n+1} \leftarrow y_n + t \delta y$ 
16:      $E_{n+1} = \text{COMPUTE\_ENERGY}(y_{n+1})$ 
17:     if  $E_{n+1} < E_n + ct \text{ DOT}(\delta y, \nabla E_n)$  then ▷ Wolfe condition
18:       if (6) holds for any  $p \in \mathcal{I}$  then ▷ locally injective
19:         return  $(y_{n+1}, E_{n+1}, t)$ 
20:       end if
21:     end if
22:      $t \leftarrow t/2$ 
23:   until MAXNUMSTEPS
24: end procedure

```

to be locally injective (hence the symmetric Dirichlet energy is finite), there exists a nonzero step size that corresponds to a locally injective map. Since we start with an injective map (the identity), by induction, the algorithm is guaranteed to maintain local injectivity at all times.

In the remaining part of this section, we discuss these steps in detail. To promote reproducibility, we provide a publicly available reference implementation (<https://github.com/lwt831/Real-time-Locally-Injective-Volumetric-Deformation>). Our algorithm is easily parallelized on the GPU by following the exact same principles presented in [Chen and Weber 2017, Appendix G].

7.1 The Gradient and the Hessian of the Energy

In order to minimize the objective function using Newton's method, the gradient and the Hessian of the energy E are required. While f and E_{iso} possess closed-form, as far as we know, the integrals in Equation (2) cannot be represented in such a manner. Consequently, we approximate them numerically:

$$E_{\text{dis}} = \sum_{p_i \in \mathcal{I}} w_i E_{\text{iso}}(p_i), \quad (28)$$

where w_i is the volume of the voxel centered at p_i . The soft positional constraints term is given by:

$$E_{P2P} = \sum_{p_i \in \mathcal{P}} \|f(p_i) - q_i\|^2, \quad (29)$$

where $\mathcal{P} \subset \Omega \subset \mathbb{R}^3$ is a set of user specified points and $q_i \in \mathbb{R}^3$ is the target position for the i^{th} handle. Plugging Equation (23) into Equation (29) and using matrix notations, we obtain:

$$E_{P2P} = \|Dy - Q\|^2, \quad (30)$$

where $D = [D_{p_1}^T, D_{p_2}^T, \dots, D_{p_{|\mathcal{P}|}}^T]^T$ such that $\forall p_i \in \mathcal{P}, [f(p_i)]^T = D_{p_i} y$, Q is the matrix stacking q_i^T . The expression for E_{smooth} is given in (27). Since E_{smooth} and E_{P2P} are both convex quadratic, it is trivial to derive their differentials:

$$\nabla_y E_{P2P} = \text{vec}(D^T(Dy - Q)), \quad \nabla_y^2 E_{P2P} = (D^T D) \otimes I_3, \quad (31)$$

$$\nabla_y E_{\text{smooth}} = \text{vec}(K^T K y), \quad \nabla_y^2 E_{\text{smooth}} = (K^T K) \otimes I_3, \quad (32)$$

where I_3 is the identity matrix in $\mathbb{R}^{3 \times 3}$, \otimes is the Kronecker product.

For the distortion energy E_{dis} , we consider $E_{\text{iso}}(p)$ at each $p \in \mathcal{I}$, whose gradient and Hessian are given as follows:

$$\nabla_y E_{\text{iso}} = \text{vec}(J_p^T \nabla J E_{\text{iso}}), \quad (33)$$

$$\nabla_y^2 E_{\text{iso}} = (J_p \otimes I_3)^T \nabla J^2 E_{\text{iso}} (J_p \otimes I_3), \quad (34)$$

where J_p is defined in Equation (24). The expressions for $\nabla J E_{\text{iso}}$ and $\nabla J^2 E_{\text{iso}}$ are given in Appendix G.

Note that $\nabla_y^2 E_{\text{iso}}$ is, in general, not positive semidefinite (PSD) since E_{iso} is not convex. Therefore, we follow the projected Newton approach and project the 9×9 matrix $\nabla J^2 E_{\text{iso}}$ to the closest (in Frobenius norm) PSD matrix $\nabla J^2 E_{\text{iso}}^+$ using the analytic solution from [Smith et al. 2019; Stomakhin et al. 2012]. We then, substitute $\nabla_y^2 E_{\text{iso}}$ with $(J_p \otimes I_3)^T \nabla J^2 E_{\text{iso}}^+ (J_p \otimes I_3)$ when $\nabla_y^2 E_{\text{iso}}$ is not PSD.

The cost to evaluate $\nabla J^2 E_{\text{iso}}$ and $\nabla_y^2 E_{\text{iso}}$ on the full sampling set \mathcal{I} can adversely impact the interactive performance of our method. To alleviate that, we adopt the strategy proposed by Chen and Weber [2017] in the 2D case. Namely, we introduce an additional set $\mathcal{H} \subset \mathcal{I}$ that contains significantly fewer samples, and approximate $\nabla^2 E_{\text{dis}}$ as:

$$\nabla^2 E_{\text{dis}} \approx \sum_{p_i \in \mathcal{H}} w_i \nabla^2 E_{\text{iso}}(p_i). \quad (35)$$

In all our experiments, we used $|\mathcal{H}| = 0.05|\mathcal{I}|$, which consistently reduced the execution times of our algorithm without compromising quality, as the example in Figure 8 shows.

In some rare cases, e.g. when the user adds the first P2P handle and starts deforming a shape from its rest pose, the Hessian of the full energy (22) can be positive semidefinite after the PSD projection. To address this, and to improve numerical stability for the linear solver, we add a small positive constant (10^{-5} in our implementation) to the diagonal of the Hessian matrix.

7.2 Adaptive Injectivity Certification Sampling

Our estimated Lipschitz constant is pretty tight for most moderate deformations, and the smoothness energy further pushes it down. Yet, for extreme deformations, denser 3D sampling is needed in

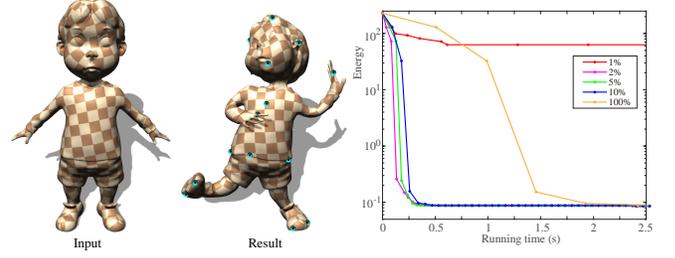


Fig. 8. Approximation of $\nabla^2 E_{\text{dis}}$. We show (right) the effect of different sampling ratios ($r_s = |\mathcal{H}|/|\mathcal{I}|$) on the optimization of the depicted boy deformation (left). We observed that with $r_s \geq 2\%$, our method consistently converges to the same final result. When using the full set of samples (orange curve), the execution time is dominated by the Hessian evaluation term. On the other hand, when r_s drops down to 1%, the progress of the Newton iterations significantly slows down due to early termination in the line search step. Using a ratio of $r_s = 5\%$ provides a sweet spot, leading to the fastest convergence.

order to prevent premature false termination of the local injectivity certification. To alleviate this without excessive density increase, we use a simple yet effective adaptive sampling strategy, by increasing the sampling density only at the problematic areas.

Assume that Ω is partitioned into a set of voxels with the same radius r , and denote each voxel by V_p^r with p being its center, all such centers form the sampling set \mathcal{I} . In the Newton iteration, we verify Condition (6) at all the voxels. If it fails on any of them, we subdivide that voxel and get eight sub-voxels with radius $r/2$. The centers of these new sub-voxels are then added to \mathcal{I} . This process is repeated until Condition (6) is satisfied on all voxels and the map is certified as locally injective. Alternatively, if a user-specified threshold for the maximal number of subdivision steps is reached, the map is assumed not to be locally injective. The Newton step size is then shortened, and further certification attempts are performed. For simplicity, we reuse the Lipschitz constant of the parent voxel for all its subvoxels and only benefit from reducing the voxel's radii. In a typical scenario, there is no need for more than 10% extra samples.

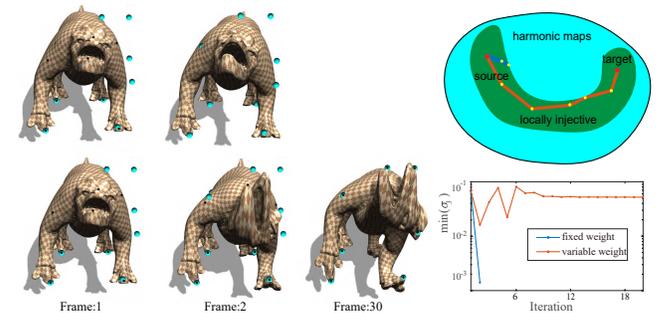


Fig. 9. Fixed vs. dynamically adjusted weights. The top left row shows the result using fixed weighting. The bottom row shows the deformation results with dynamic weighting. The top right schematics shows the optimization paths taken by the two approaches. The plot (bottom right) compares the lower bounds on σ_3 , obtained using the Lipschitz constant given in Equation (21).

7.3 Dynamic Weighting for Fast and Robust Deformation

The user interface of our system supports either direct specification of the target positions of the P2P handles, or continuous dragging of the handles toward its target position, which result in a highly responsive and interactive user experience. However, due to its non-convex nature, our optimization is not guaranteed to satisfy the P2P constraints under all circumstances. A small weight for the P2P energy term leads to a significant miss of the user constraints. On the other hand, a large weight results in a P2P energy term that overshadows the other terms, pulling the map toward the boundary of the feasible domain (the subspace of locally injective maps). Due to the interior point nature of the optimization, leaving the feasible domain is impossible and will be prevented by the line search step. Nonetheless, this slows down the algorithm and temper with robustness and convergence. This is illustrated by the blue path in Figure 9 top right. To avoid this, we propose a heuristic for dynamic weight adjustment such that a relatively “safe” path (marked red) within the feasible domain is taken.

Let the current weights for E_{P2P} and E_{smooth} in Equation (22) be λ_1 and λ_2 respectively. In each iteration, we adjust the weights by comparing E_{P2P} to a user-defined threshold E_0 . If $E_{P2P} \geq E_0$ we increase λ_2 by a multiplicative factor $\kappa_2 > 1$ and check the following condition:

$$\sigma_3(p) - rL_p^r \geq \epsilon, \quad (36)$$

which is a stronger version of Condition (6), with ϵ being a pre-defined constant. If Condition (36) does not hold, we reduce the influence of E_{P2P} by decreasing λ_1 by a multiplicative factor $\kappa_1 < 1$. If $E_{P2P} < E_0$, we acknowledge that the influence of E_{P2P} is small enough. We check Condition (6), and if it’s not satisfied, we enhance E_{smooth} by increasing λ_2 by the factor κ_2 . After the iteration converges, we reset λ_1 and λ_2 to their default values.

By taking the strategy above, we effectively “pull” the optimization direction towards the interior of the subspace of the locally injective maps in each iteration. This strategy turned out to be extremely effective in practice. The bottom right plot of Figure 9 shows that the estimated lower bound on σ_3 in the second iteration goes as low as 0.001, which indicates the proximity to the boundary of the feasible domain. Using the dynamic weighting scheme, we successfully prevent this from happening. This is also illustrated in the top left part of Figure 9, where fixed weights are used and the method terminates prematurely after two iterations.

8 EXPERIMENTAL RESULTS

For all the results presented in this paper, we use the following default parameters unless stated otherwise. We set the dimension of the reduced subspace to 900 (Section 6.2). The size of the sampling sets are $|\mathcal{I}| \approx 3 \times 10^4$ and $|\mathcal{H}| = 0.05|\mathcal{I}|$. The maximum number of subdivisions for the adaptive injectivity certification is 6. The weights for E_{P2P} and E_{smooth} are $\lambda_1 = 10^5$ and $\lambda_2 = 0.1$. All experiments are performed on a Windows 10 machine with an Intel i5-8500 CPU 3.0GHz, with 32GB RAM, and a NVIDIA TITAN RTX graphics card. The cages are automatically generated using [Sacht et al. 2015]. To avoid singularities in the harmonic basis functions ϕ_{v_i} and ψ_{t_j} which may cause problem with the injectivity certification (6), we ensure that there is some offset from the cage to the deforming



Fig. 10. Deformation of the boy. A comparison of our method with the other two volumetric methods using a coarse cage with 102 vertices. For PWL (Tet), the cage is tessellated into a tetrahedral mesh with 735 vertices (bottom left). 378 anchors are placed on the medial axis of the cage for VHM (bottom middle). 37,913 samples are scattered in the cage’s interior for our method (bottom right).

shape, and none of the voxels $V_{p_i}^r$ intersects the cage. We visualize the user specified P2P target positions q_i as cyan spheres, and the images of p_i (i.e., $f(p_i)$) as smaller black spheres. A small black sphere centered inside a larger cyan sphere indicates satisfaction of the P2P constraint.

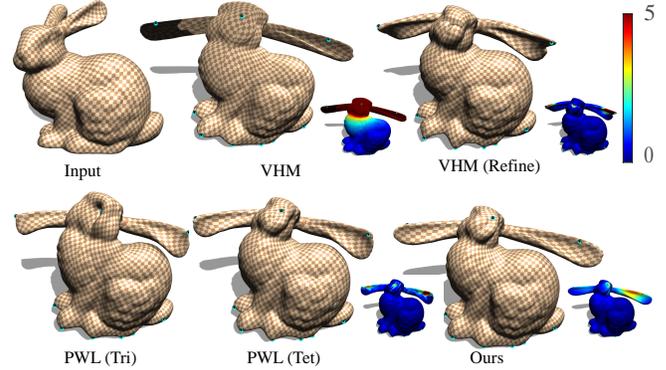


Fig. 11. Injectivity loss. Due to the extreme deformation, the VHM result fails to be locally injective, and depicts high distortion in the head of the bunny. We experimented with a modified VHM method such that the samples (so-called anchors) are taken from our method. Though improved, the result still fails to be locally injective.

We compare our method against three state-of-the-art methods. PWL (Tet) denotes the tetrahedral deformation method using the analytic Eigensystem [Smith et al. 2019] optimizing the symmetric Dirichlet energy. PWL (Tri) denotes the SR-ARAP surface deformation [Levi and Gotsman 2015], and VHM denotes [Ben-Chen et al. 2009]. To provide online user interaction with PWL (tet), we embed the deforming shape into a low resolution tetrahedron mesh, and use a barycentric mapping to represent the deformation. The superiority of our method can be observed by comparing the speed as well as the overall appearance, including smoothness, local-and-global volume preservation, and isometric distortion of the results produced with the different methods. The heat maps depict the volumetric symmetric Dirichlet energy (E_{iso}). We subtract 6 such that no isometric distortion corresponds to 0 on the side color bars. For the meshless methods, we evaluate the distortion on the vertices of

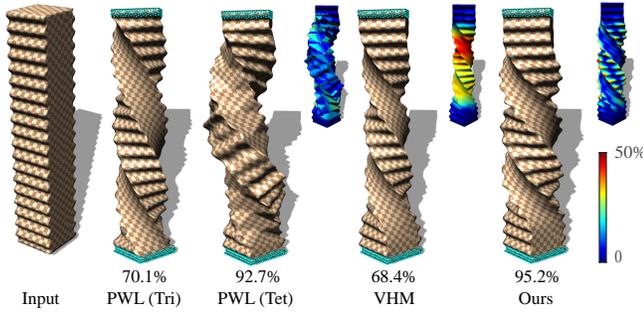


Fig. 12. Volume preservation of a twisting bar. The percentage above each method indicates the global volume ratio, defined as the quotient of the volume of the result and the volume of the source. The heat maps show the relative *local* volume change for the volumetric methods and is omitted for the surface-based method. Our method nicely preserves the volume globally as well as locally.

the mesh and interpolate linearly over triangles. PWL (Tri) colors are omitted since volumetric distortion is undefined for surfaces.

The degrees of freedom (DOF) of VHM are derived from the cage’s complexity (3 scalars for each vertex and additional 3 for each triangle). In PWL (Tet), we tessellated the cage using TetGen [Si 2015], and obtained a tetrahedral mesh in which each vertex corresponds to 3 DOF. We also tried tessellating some input meshes as opposed to its (much coarser) cages, albeit, this was prohibitively slow. For PWL (Tri), there are 3 DOF per vertex of the full resolution triangle mesh, with no DOF for the interior. Our method uses the same cage as VHM and PWL (Tet).

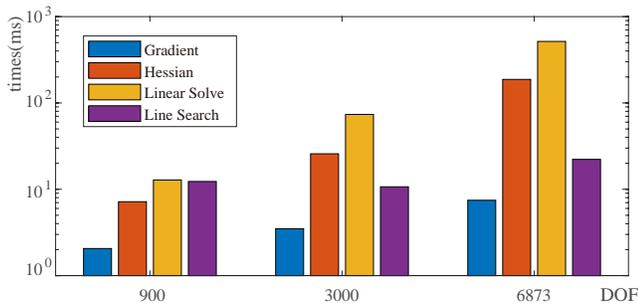


Fig. 13. Algorithm balance. We show how the runtime of a single iteration of our algorithm is spread across the four stages. Each group of bins corresponds to a different model with variable amount of DOF.

Figure 13 shows how the runtime of each Newton iteration is spread among the four stages, including gradient evaluation (Equation (33)), Hessian assembly-and-modification, linear solve, and line search. We observed that our solver is well-balanced among these four stages. Table 1 compares the execution times and iteration count to reach convergence on a variety of shapes. Our method has the shortest execution times and the least number of iterations consistently. Note that we ran PWL (Tet) on a low resolution mesh for better efficiency. PWL (Tri) operates on the original meshes

directly, therefore has the highest number of DOF, leading to the slowest execution times.

Our method and PWL (Tet) are designed to produce locally injective maps. In contrast, VHM, while aiming at that in spirit, develops foldovers quite often, especially in the vicinity of the P2P handles, as shown in Figure 11. These failures of VHM are identified by using our injectivity certification mechanism. For PWL (Tri), local injectivity is undefined within the 3D ambient space.

Our method is also effective in keeping the average-and-maximal distortion low, which is partly due to the newly introduced smoothness energy term. This is illustrated by the distortion heat maps in Figures 3,10,11,12, and 14. It is evident that the distortion in our results is kept relatively low even around the P2P handles. In contrast, we observed significant increase of distortion for both VHM and PWL (Tet) on the same problematic regions. Our method distributes the distortion more uniformly over the shape compared to the others.

Our method and PWL (Tet) are both effective in volume preservation, though it is not targeted explicitly. This is attributed to the use of the symmetric Dirichlet energy, which is evaluated over the entire domain rather than solely on the boundary [Chen and Weber 2015, 2017; Levi and Gotsman 2015] or along the medial axis [Ben-Chen et al. 2009]. This is illustrated in Figures 1, 10 and 11. A quantitative comparison of the volume preserving ability is given in Figure 12.

In Figure 3, we compare our method to [Smith et al. 2019] (PWL (Tet)) on a high resolution tetrahedral mesh. While both methods guarantee injectivity and optimize the symmetric Dirichlet energy, due to the harmonic subspace and our additional smoothness term, our method is free of artifacts. The piecewise linear nature of [Smith et al. 2019] is evident by the lack of smoothness when executed on a low resolution tetrahedral mesh. The execution time of PWL (Tet) on the high resolution mesh is 225 times slower compared to ours, and artifacts near the handles persist.

In Figure 11, we compare against a modified VHM method where the energy is evaluated on \mathcal{I} rather than on the medial axis solely. While the VHM result is improved with this dense and uniform sampling, it still fails to be locally injective.

We include an additional compilation of comparisons of the four methods in Figure 14. Finally, the accompanying video showcases some of these results, examines them from different point of views, and demonstrates the real-time performance of our method.

9 SUMMARY AND DISCUSSION

We have presented the *first* meshless framework for volumetric deformation with local injectivity guarantees. A novel generalization of the principle ideas from [Poranne and Lipman 2014] to the volumetric case is formed, where the main challenge lies in the development of a tight Lipschitz constant for the smallest singular value of the map Jacobian. We choose to follow Chen and Weber [2017] who demonstrated the many advantages of shape-aware harmonic basis functions as opposed to general RBFs. The main building block of our generalization (Section 4), however, can be extended to other function types, if one explicitly wishes to avoid smoothness, or when the interior of the domain should be deformed without affecting the boundary.

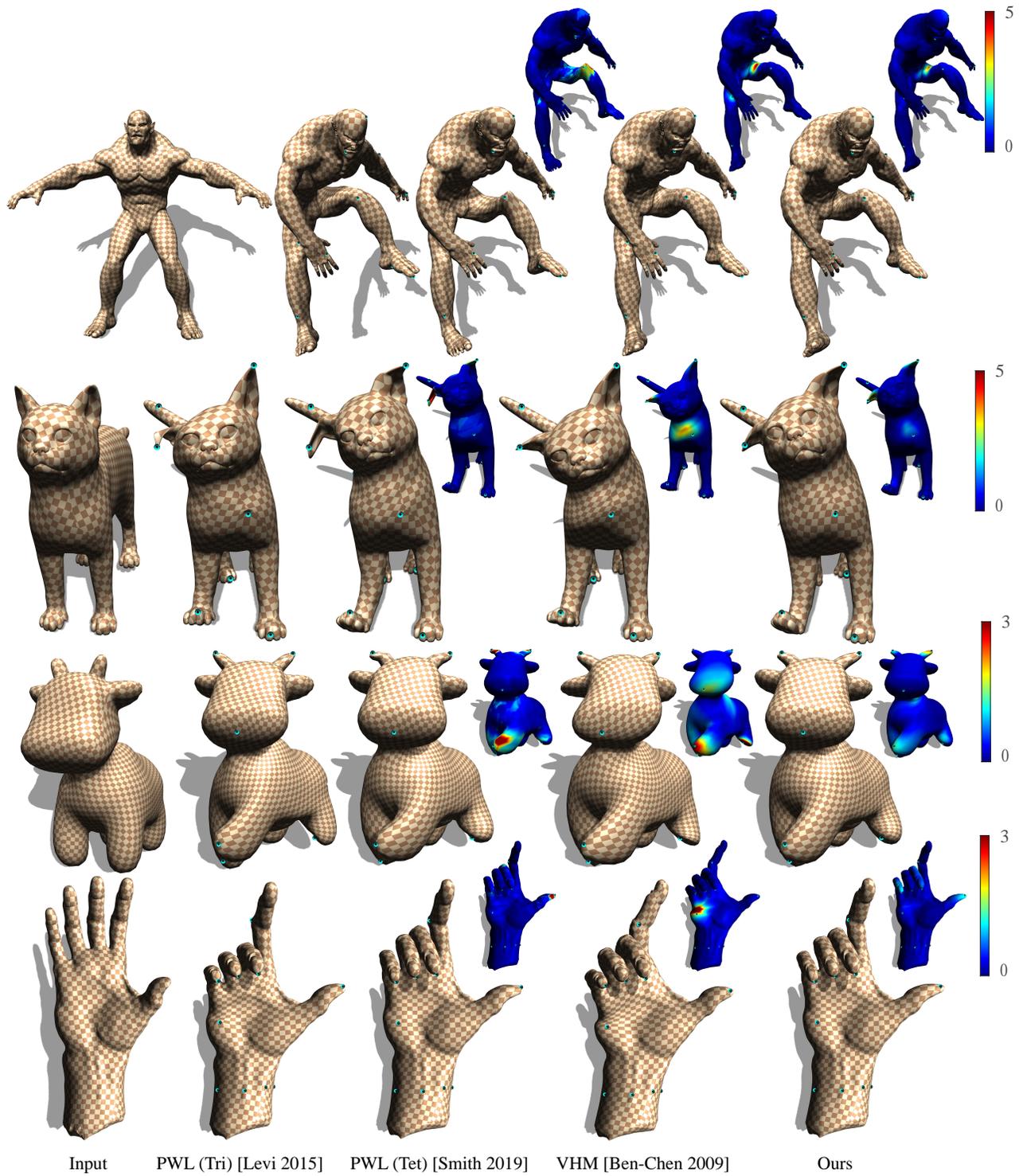


Fig. 14. Additional comparisons of our method with the state-of-the-art.

Table 1. Comparison of execution times and iteration count for each method to reach convergence when initialized with the identity map. We also report the DOF used in each method when deforming these shapes.

Model	PWL (Tri) [Levi 2015]			PWL (Tet) [Smith 2019]			VHM [Ben-Chen 2009]			Ours		
	DOF	time (sec)	#iter	DOF	time (sec)	#iter	DOF	time (sec)	#iter	DOF	time (sec)	#iter
animal	58,566	2.87	129	7,920	1.02	26	6,873	6.53	71	900	0.34	15
amardillo	497,862	25.34	131	8,316	0.47	13	1,464	1.05	36	900	0.21	10
bar	19,974	2.00	250	3,312	0.30	23	1,032	1.50	393	900	0.32	14
boy	39,996	3.20	194	8,208	0.41	12	906	0.76	145	900	0.28	10
bunny	79,371	1.68	51	21,123	2.21	18	9,375	10.55	50	900	0.35	14
beast	85,164	11.08	323	9,267	1.06	27	1,356	3.89	443	900	0.37	21
cat	105,870	7.14	165	8,370	0.50	12	906	3.01	204	900	0.31	9
hand	43,041	3.01	167	8,550	0.55	11	906	1.70	295	900	0.31	10
horse	59,550	10.05	407	5,727	0.40	16	447	0.54	124	300	0.19	11

Unlike [Chen and Weber 2015], we targeted local injectivity solely, without the ability to bound the distortion. Setting an explicit distortion bound should be straightforward by using semidefinite programming (SDP), based on the convexification presented in [Kovalsky et al. 2014].

Through experiments and comparisons, we demonstrated that our method is significantly faster than the existing methods, while producing superior results which are smooth, locally injective, and exhibit low isometric distortion.

9.1 Limitations and Future Work

Our method guarantees local injectivity, however, the space of locally injective maps is nonconvex, hence we cannot guarantee that the positional constraints will be satisfied while simultaneously guaranteeing injectivity. To the best of our knowledge, none of the existing methods (meshless or mesh-based) has this ability and developing such a method is highly desirable. For the space of locally injective *planar* harmonic maps, there exists a change of variables [Hefetz Fedida et al. 2017; Levi and Weber 2016] that transforms bijectively the nonconvex space of harmonic injective maps to a convex space (albeit, without positional constraints). The existence of such a transformation in the volumetric case is an open question.

On the theoretical side, it was disappointing to learn that a bounded distortion harmonic mapping theorem, analogous to the planar case, does not exist. In the future, it would be interesting to explore other mapping spaces for which the injectivity in the volume interior can be deduced solely based on the boundary behavior of the map. In particular, for certain harmonic maps, the minimum of σ_3 is attained on the boundary. Characterizing and identifying these maps is useful, and we intend to explore it.

Finally, our method is designed to produce locally injective maps. An avenue for future work would be to address a requirement for *global* injectivity.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. This research was partially funded by the Israel Science Foundation (grant No. 1949/19), the National Natural Science Foundation of China (62072422, 62025207), the National Key R&D Program of China (2020YFC1523102), the NSF of Anhui Province of China (2008085MF195) and the Zhejiang Lab (2019NB0AB03).

REFERENCES

- Adams Bart, Ovsjanikov Maks, Wand Michael, Seidel Hans-Peter, and Guibas Leonidas J. 2008. Meshless Modeling of Deformable Shapes and Their Motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) (SCA 2008). Eurographics Association, Goslar, DEU, 77–86.
- Aharon Ido, Chen Renjie, Zorin Denis, and Weber Ofir. 2019. Bounded Distortion Tetrahedral Metric Interpolation. *ACM Transactions on Graphics* 38, 6 (2019), Article 182, 17 pages.
- Aigerman Noam and Lipman Yaron. 2013. Injective and bounded distortion mappings in 3D. *ACM Transactions on Graphics* 32, 4 (2013), 106.
- Ben-Chen Mirela, Weber Ofir, and Gotsman Craig. 2009. Variational harmonic maps for space deformation. *ACM Transactions on Graphics* 28, 3 (2009), Article 34, 11 pages.
- Chao Isaac, Pinkall Ulrich, Sanan Patrick, and Schröder Peter. 2010. A simple geometric model for elastic deformations. *ACM Transactions on Graphics* 29, 4 (2010), Article 38, 6 pages.
- Chen Renjie and Weber Ofir. 2015. Bounded distortion harmonic mappings in the plane. *ACM Transactions on Graphics* 34, 4 (2015), Article 73, 12 pages.
- Chen Renjie and Weber Ofir. 2017. GPU-Accelerated Locally Injective Shape Deformation. *ACM Transactions on Graphics* 36, 6 (2017), Article 214, 13 pages.
- Chen Renjie, Weber Ofir, Keren Daniel, and Ben-Chen Mirela. 2013. Planar shape interpolation with bounded distortion. *ACM Transactions on Graphics* 32, 4 (2013), Article 108, 11 pages.
- Chien Edward, Chen Renjie, and Weber Ofir. 2016a. Bounded Distortion Harmonic Shape Interpolation. *ACM Transactions on Graphics* 35, 4 (2016), Article 105, 15 pages.
- Chien Edward, Levi Zohar, and Weber Ofir. 2016b. Bounded Distortion Parametrization in the Space of Metrics. *ACM Transactions on Graphics* 35, 6 (2016), Article 215, 16 pages.
- Floater Michael S., Kós Géza, and Reimers Martin. 2005. Mean value coordinates in 3D. *Computer Aided Geometric Design* 22, 7 (2005), 623–631.
- Fu Xiao-Ming, Liu Yang, and Guo Baining. 2015. Computing Locally Injective Mappings by Advanced MIPS. *ACM Transactions on Graphics* 34, 4, Article 71 (2015), 12 pages.
- Golub Gene H. and Van Loan Charles F. 1996. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA.
- Hefetz Fedida Eden, Chien Edward, and Weber Ofir. 2017. Fast Planar Harmonic Deformations with Alternating Tangential Projections. *Computer Graphics Forum* 36, 5 (2017), 175–188. Proceedings of Symposium on Geometry Processing 2017.
- Huang Jin, Shi Xiaohan, Liu Xinguo, Zhou Kun, Wei Li-Yi, Teng Shang-Hua, Bao Hujun, Guo Baining, and Shum Heung-Yeung. 2006. Subspace Gradient Domain Mesh Deformation. *ACM Transactions on Graphics* 25, 3 (2006), 1126–1134.
- Igarashi Takeo, Moscovich Tomer, and Hughes John F. 2005. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics* 24, 3 (2005), 1134–1141.
- Jacobson Alec, Baran Ilya, Popović Jovan, and Sorkine Olga. 2011. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics* 30, 4 (2011), Article 78, 8 pages.
- Joshi Pushkar, Meyer Mark, DeRose Tony, and Green Brian. 2007. Harmonic coordinates for character articulation. *ACM Transactions on Graphics* 26, 3 (2007), 71.
- Ju Tao, Schaefer Scott, and Warren Joe. 2005. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics* 24, 3 (2005), 561–566.
- Kovalsky Shahar Z., Aigerman Noam, Basri Ronen, and Lipman Yaron. 2014. Controlling Singular Values with Semidefinite Programming. *ACM Transactions on Graphics* 33, 4, Article 68 (2014), 13 pages.
- Kovalsky Shahar Z., Aigerman Noam, Basri Ronen, and Lipman Yaron. 2015. Large-scale bounded distortion mappings. *ACM Transactions on Graphics* 34, 6 (2015), Article 191.
- Kovalsky Shahar Z., Galun Meirav, and Lipman Yaron. 2016. Accelerated Quadratic Proxy for Geometric Optimization. *ACM Transactions on Graphics* 35, 4 (2016),

- Article 134.
- Laugesen Richard Snyder. 1996. Injectivity can fail for higher-dimensional harmonic extensions. *Complex Variables, Theory and Application: An International Journal* 28, 4 (1996), 357–369.
- Levi Z. and Gotsman C. 2015. Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (2015), 264–277.
- Levi Zohar and Weber Ofir. 2016. On the convexity and feasibility of the bounded distortion harmonic mapping problem. *ACM Transactions on Graphics* 35, 4 (2016), Article 106, 15 pages.
- Li Xian-Ying, Ju Tao, and Hu Shi-Min. 2013. Cubic mean value coordinates. *ACM Transactions on Graphics* 32, 4 (2013), Article 126, 10 pages.
- Lipman Yaron. 2012. Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics* 31, 4 (2012), 108.
- Lipman Yaron, Levin David, and Cohen-Or Daniel. 2008. Green Coordinates. *ACM Transactions on Graphics* 27, 3 (2008), 10 pages.
- Lipman Yaron, Sorkine Olga, Levin David, and Cohen-Or Daniel. 2005. Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics* 24, 3 (2005), 479–487.
- Nocedal Jorge and Wright Stephen. 2006. *Numerical Optimization*. Springer Science & Business Media, New York.
- Poranne Roi and Lipman Yaron. 2014. Provably good planar mappings. *ACM Transactions on Graphics* 33, 4 (2014), 76.
- Pugh Charles C. 2015. *Real Mathematical Analysis (2nd Ed.)*. Springer, Cham, USA.
- Rabinovich Michael, Poranne Roi, Panozzo Daniele, and Sorkine-Hornung Olga. 2017. Scalable Locally Injective Mappings. *ACM Transactions on Graphics* 36, 2, Article 16 (2017), 16 pages.
- Sacht Leonardo, Vouga Etienne, and Jacobson Alec. 2015. Nested Cages. *ACM Transactions on Graphics* 34, 6, Article 170 (2015), 14 pages.
- Schüller Christian, Kavan Ladislav, Panozzo Daniele, and Sorkine-Hornung Olga. 2013. Locally injective mappings. *Computer Graphics Forum* 32, 5 (2013), 125–135.
- Sederberg Thomas W. and Parry Scott R. 1986. Free-Form Deformation of Solid Geometric Models. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1986)*. Association for Computing Machinery, New York, NY, USA, 151–160.
- Si Hang. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Softw.* 41, 2, Article 11 (2015), 36 pages.
- Smith Breannan, Goes Fernando De, and Kim Theodore. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Transactions on Graphics* 38, 1, Article 3 (2019), 15 pages.
- Smith Jason and Schaefer Scott. 2015. Bijective Parameterization with Free Boundaries. *ACM Transactions on Graphics* 34, 4 (2015), Article 70, 9 pages.
- Sorkine Olga and Alexa Marc. 2007. As-rigid-as-possible Surface Modeling. In *Proceedings of the Symposium on Geometry Processing*. Eurographics, Switzerland, 109–116.
- Stomakhin Alexey, Howes Russell, Schroeder Craig, and Teran Joseph M. 2012. Energetically Consistent Invertible Elasticity. In *Eurographics Conference on Computer Animation (Lausanne, Switzerland) (EUROSCA'12)*. Eurographics Association, Goslar, DEU, 25–32.
- Summer Robert W and Popović Jovan. 2004. Deformation transfer for triangle meshes. *ACM Transactions on Graphics* 23, 3 (2004), 399–405.
- Thiery Jean-Marc, Tierny Julien, and Boubekeur Tamy. 2014. Jacobians and Hessians of mean value coordinates for closed triangular meshes. *The Visual Computer* 30, 9 (2014), 981–995.
- Weber Ofir, Ben-Chen Mirela, and Gotsman Craig. 2009. Complex Barycentric Coordinates with Applications to Planar Shape Deformation. *Computer Graphics Forum* 28, 2 (2009), 587–597.
- Weber Ofir and Gotsman Craig. 2010. Controllable conformal maps for shape deformation and interpolation. *ACM Transactions on Graphics* 29, 4 (2010), Article 78, 11 pages.
- Weber Ofir, Myles Ashish, and Zorin Denis. 2012a. Computing Extremal Quasiconformal Maps. *Computer Graphics Forum* 31, 5 (2012), 1679–1689.
- Weber Ofir, Poranne Roi, and Gotsman Craig. 2012b. Biharmonic Coordinates. *Computer Graphics Forum* 31, 8 (2012), 2409–2422.

A THE HARMONIC BASIS FUNCTIONS

Using the notations given in Figure 4, one can express $\phi_v(p)$ and $\psi_t(p)$ as follows:

$$\phi_v = \sum_{t \in N(v)} \frac{1}{2s_t} (l_u \times l_w) \cdot P_t, \quad (37)$$

$$\psi_t = - \sum_{v \in t} \frac{C_v^t}{\|d_v^t\|} ((l_u \times l_w) \cdot n_t) - \frac{3}{st} \omega_t \text{vol}_t, \quad (38)$$

where $N(v)$ is the set of triangles neighboring vertex v . The area of triangle t is denoted s_t . We denote by vol_t the signed volume of the tetrahedron formed by triangle t and the point p . The signed solid angle at p is $4\pi\omega_t$. The functions $C_v^t(p) \in \mathbb{R}$, and $P_t(p) \in \mathbb{R}^3$ are:

$$C_v^t = \frac{1}{4\pi} \log \left(\frac{\|l_u\| + \|l_w\| + \|d_v^t\|}{\|l_u\| + \|l_w\| - \|d_v^t\|} \right),$$

$$P_t = n_t \times \sum_{v \in t} \frac{d_v^t}{\|d_v^t\|} C_v^t - \omega_t n_t.$$

The gradients of Equation (37) and (38) are given by:

$$\nabla \phi_v = \sum_{t \in N(v)} \frac{1}{2s_t} P_t \times d_v^t, \quad (39)$$

$$\nabla \psi_t = -P_t. \quad (40)$$

To obtain the Hessian of ϕ_v and ψ_t , we first compute the Jacobian matrix of P_t :

$$J(P_t) = \sum_{v \in t} \left(\beta_v^t \left(n_t \times \frac{d_v^t}{\|d_v^t\|} \right)^T - \alpha_v^t n_t^T \right), \quad (41)$$

where $\alpha_v^t, \beta_v^t \in \mathbb{R}^3$ are row vectors:

$$\alpha_v^t = \frac{\|l_u\| + \|l_w\|}{2\pi \left((\|l_u\| + \|l_w\|)^2 - \|d_v^t\|^2 \right)} \left(\frac{l_u}{\|l_u\|} \times \frac{l_w}{\|l_w\|} \right), \quad (42)$$

$$\beta_v^t = \frac{\|d_v^t\|}{2\pi \left((\|l_u\| + \|l_w\|)^2 - \|d_v^t\|^2 \right)} \left(\frac{l_u}{\|l_u\|} + \frac{l_w}{\|l_w\|} \right).$$

Finally, the Hessians of ϕ_v and ψ_t are:

$$H_{\phi_v} = \sum_{t \in N(v)} \frac{1}{2s_t} J(P_t) [d_v^t]_{\times}, \quad (43)$$

$$H_{\psi_t} = -J(P_t). \quad (44)$$

B EXTENSION TO SIGNED SINGULAR VALUES

By convention, the signed (smallest) singular value, σ_m^s , of a $m \times m$ matrix A is defined as $\sigma_m^s(A) = \text{sign}(\det(A))\sigma_m(A)$. In the following, we prove that given any $A, E \in \mathbb{R}^{m \times m}$:

$$|\sigma_m^s(A + E) - \sigma_m^s(A)| \leq \|E\|_2.$$

PROOF. We consider the following cases:

(1) $\sigma_m^s(A + E)\sigma_m^s(A) = 0$.

In this case either $\sigma_m^s(A + E) = 0$ or $\sigma_m^s(A) = 0$, hence:

$$|\sigma_m^s(A + E) - \sigma_m^s(A)| = |\sigma_m(A + E) - \sigma_m(A)| \leq \|E\|_2.$$

(2) $\sigma_m^s(A + E)\sigma_m^s(A) > 0$.

In this case $\sigma_m^s(A + E)$ has the same sign as $\sigma_m^s(A)$, therefore:

$$|\sigma_m^s(A + E) - \sigma_m^s(A)| = |\sigma_m(A + E) - \sigma_m(A)| \leq \|E\|_2.$$

(3) $\sigma_m^s(A + E)\sigma_m^s(A) < 0$.

For any λ_0 we have:

$$\begin{aligned} & |\sigma_m^s(A + E) - \sigma_m^s(A)| \\ &= |\sigma_m^s(A + E) - \sigma_m^s(A + \lambda_0 E) + \sigma_m^s(A + \lambda_0 E) - \sigma_m^s(A)| \\ &\leq |\sigma_m^s(A + E) - \sigma_m^s(A + \lambda_0 E)| + |\sigma_m^s(A + \lambda_0 E) - \sigma_m^s(A)|, \end{aligned}$$

where the last inequality is due to the triangle inequality. Since $\sigma_m^s(A + E)\sigma_m^s(A) < 0$, we know that $\sigma_m^s(A + E)$ and $\sigma_m^s(A)$ have different signs. By construction, $\sigma_m^s(M)$ is a continuous function of the elements of M . Therefore, by the intermediate value theorem, there exists $\lambda_0 \in [0, 1]$ such that $\sigma_m^s(A + \lambda_0 E) = 0$. Hence, we can apply case (1) twice:

$$\begin{aligned} & |\sigma_m^s(A + E) - \sigma_m^s(A + \lambda_0 E)| + |\sigma_m^s(A + \lambda_0 E) - \sigma_m^s(A)| \\ & \leq (1 - \lambda_0) \|E\|_2 + \lambda_0 \|E\|_2 = \|E\|_2. \end{aligned}$$

□

C PROOF FOR LEMMA 5.1 - AFFINE MAP

PROOF. To prove the lemma, we need to show that the right-hand side of Equation (13) maps the vectors $\{d_u^t, d_v^t, n_t\}$ to $\{y_v - y_w, y_w - y_u, y_t\}$. Since n_t is orthogonal to $n_t \times d_{v_i}^t$, we have:

$$\begin{aligned} & \left(y_t n_t^T - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} (n_t \times d_{v_i}^t)^T \right) n_t \\ & = y_t (n_t^T n_t) - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} \left((n_t \times d_{v_i}^t)^T n_t \right) \\ & = y_t. \end{aligned}$$

Also notice that $(n_t \times d_i^t)^T d_j^t$ is the triple product of the vector set $\{n_t, d_i^t, d_j^t\}$ ($i, j \in \{u, v, w\}$), which denotes $(6 \times)$ the signed volume of the tetrahedron spanned by this set, hence we have:

$$\begin{aligned} & \left(y_t n_t^T - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} (n_t \times d_{v_i}^t)^T \right) d_u^t = y_v - y_w, \\ & \left(y_t n_t^T - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} (n_t \times d_{v_i}^t)^T \right) d_v^t = y_w - y_u, \end{aligned}$$

which completes the proof. □

D PROOF FOR THE SUBHARMONICITY OF σ_1

We have $\nabla \sigma_1 = \sum_{i=1}^9 \nabla \tilde{J}_i \frac{\partial \sigma_1}{\partial \tilde{J}_i}$, where \tilde{J} is the vectorized representation of matrix J (i.e. $\tilde{H} = [\nabla \tilde{J}_1, \nabla \tilde{J}_2, \dots, \nabla \tilde{J}_9]$). By applying the product rule, we get:

$$\nabla^2 \sigma_1 = \sum_{i=1}^9 \nabla^2 \tilde{J}_i \frac{\partial \sigma_1}{\partial \tilde{J}_i} + \sum_{i=1}^9 \nabla \tilde{J}_i (\nabla \frac{\partial \sigma_1}{\partial \tilde{J}_i})^T. \quad (45)$$

With the chain rule, we get $\nabla \frac{\partial \sigma_1}{\partial \tilde{J}_i} = \sum_{j=1}^9 \nabla \tilde{J}_j \frac{\partial^2 \sigma_1}{\partial \tilde{J}_i \partial \tilde{J}_j}$. Plugging it into Equation (45), we obtain:

$$\nabla^2 \sigma_1 = \sum_{i=1}^9 \nabla^2 \tilde{J}_i \frac{\partial \sigma_1}{\partial \tilde{J}_i} + \sum_{i,j=1}^9 \nabla \tilde{J}_i \nabla \tilde{J}_j^T \frac{\partial^2 \sigma_1}{\partial \tilde{J}_i \partial \tilde{J}_j}. \quad (46)$$

Therefore,

$$\Delta \sigma_1 = \underbrace{\sum_{i=1}^9 \Delta \tilde{J}_i \frac{\partial \sigma_1}{\partial \tilde{J}_i}}_I + \text{Trace} \left(\underbrace{\sum_{i,j=1}^9 \nabla \tilde{J}_i \nabla \tilde{J}_j^T \frac{\partial^2 \sigma_1}{\partial \tilde{J}_i \partial \tilde{J}_j}}_{II} \right). \quad (47)$$

Notice that \tilde{J}_i is a component of the derivative of f . Since f is harmonic, so is each component of its derivative, which means that

part I in Equation (47) vanishes. For part II , we first rewrite it using vector notations:

$$\sum_{i,j=1}^9 \nabla \tilde{J}_i \nabla \tilde{J}_j^T \frac{\partial^2 \sigma_1}{\partial \tilde{J}_i \partial \tilde{J}_j} = \tilde{H} \nabla^2 \sigma_1 \tilde{H}^T. \quad (48)$$

By plugging $E = \sigma_1$ into the expression from Appendix G, we obtain the 9 eigenvalues of $\nabla^2 \sigma_1$: $0, 0, 0, 0, 0, \frac{1}{\sigma_1 - \sigma_2}, \frac{1}{\sigma_1 + \sigma_2}, \frac{1}{\sigma_1 - \sigma_3}, \frac{1}{\sigma_1 + \sigma_3}$. Since σ_1 is the largest singular value, both in magnitude and absolute value, the 4 fractions are all positive. Hence, $\nabla^2 \sigma_1$ is positive semi-definite and $\tilde{H} \nabla^2 \sigma_1 \tilde{H}^T \geq 0$. Finally, we conclude that the Laplacian of σ_1 is positive, hence, it is a subharmonic function.

E THE DIFFERENTIAL REPRESENTATION OF THE JACOBIAN AND HESSIAN OF f

We prove below Theorem 5.2. As a side result, we also have a differential representation for the map Jacobian (though we do not utilize it directly in this work). Plugging Equation (39)-(40) into Equation (4), we have:

$$\begin{aligned} J(p) & = \sum_{t \in \mathcal{F}} \left(\frac{1}{2s_t} \sum_{v \in t} y_v (P_t \times d_v^t)^T - y_t P_t^T \right) \\ & = \sum_{t \in \mathcal{F}} \omega_t \left(y_t n_t^T - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} (n_t \times d_{v_i}^t)^T \right) \\ & \quad + \sum_{t \in \mathcal{F}} \sum_{v \in t} C_v^t \left(y_t n_t^T - \frac{1}{2s_t} \sum_{v_i \in t} y_{v_i} (n_t \times d_{v_i}^t)^T \right) \left[\frac{d_v^t}{\|d_v^t\|} \right]_{\times}. \end{aligned}$$

Using Lemma 5.1 we express J as a blend of matrices A_t :

$$J(p) = \sum_{t \in \mathcal{F}} \left(\omega_t A_t + \left(\sum_{v \in t} C_v^t A_t \left[\frac{d_v^t}{\|d_v^t\|} \right]_{\times} \right) \right). \quad (49)$$

To compute $H(p)$, we take the derivative of Equation (49):

$$H(p) = \sum_{t \in \mathcal{F}} \left(A_t \circ \nabla \omega_t + \sum_{v \in t} \left(A_t \left[\frac{d_v^t}{\|d_v^t\|} \right]_{\times} \circ \nabla C_v^t \right) \right). \quad (50)$$

Following [Ben-Chen et al. 2009], we get:

$$\nabla \omega_t = \sum_{v \in t} \alpha_v^t. \quad (51)$$

Again, by taking the derivative of C_v^t we have:

$$\nabla C_v^t = \frac{1}{4\pi} \nabla \log \left(\frac{\|l_u\| + \|l_w\| + d_v^t}{\|l_u\| + \|l_w\| - d_v^t} \right) = \beta_v^t. \quad (52)$$

Plugging Equation (51)-(52) into Equation (50), we have:

$$H(p) = \sum_{t \in \mathcal{F}} \sum_{v \in t} \left(A_t \circ \alpha_v^t + A_t \left[\frac{d_v^t}{\|d_v^t\|} \right]_{\times} \circ \beta_v^t \right). \quad (53)$$

The summation can be rewritten as follows,

$$H(p) = \sum_{e \in \mathcal{E}} \left(A_e \circ \alpha_e + \left(A_e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \beta_e \right), \quad (54)$$

where A_e denotes the matrix A_t with $e \in t$. Next, let e' be the opposite edge of e , it follows that $\alpha_e = -\alpha_{e'}$, $\beta_e = \beta_{e'}$ and $d_e = -d_{e'}$

and the Hessian can be written as:

$$H(p) = \sum_{e \in \mathcal{E}} \left(-A_{e'} \circ \alpha_e - \left(A_{e'} \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \beta_e \right). \quad (55)$$

Finally, we combine Equation (54) and (55), and express the Hessian as a blend of matrix differences δ_A^e :

$$H(p) = \frac{1}{2} \sum_{e \in \mathcal{E}} \left(\delta_A^e \circ \alpha_e + \left(\delta_A^e \left[\frac{d_e}{\|d_e\|} \right]_{\times} \right) \circ \beta_e \right). \quad (56)$$

F INEQUALITY OF THE HESSIAN MODULUS

Here, we provide an estimation of the constant c such that Equation (26) holds. By Equation (25), we have:

$$\|Ky\| = \sqrt{\sum_{e \in \mathcal{E}} \|\delta_A^e\|^2} \quad (57)$$

Using the inequality of means, it follows that:

$$\frac{1}{n} \sum_{e \in \mathcal{E}} \|\delta_A^e\| \leq \sqrt{\frac{1}{n} \sum_{e \in \mathcal{E}} \|\delta_A^e\|^2}. \quad (58)$$

Combining Equation (57) and (58), we get:

$$\frac{1}{\sqrt{n}} \sum_{e \in \mathcal{E}} \|\delta_A^e\| \leq \|Ky\|. \quad (59)$$

Now, let's take:

$$c(p) = \frac{\sqrt{n}}{2} \max_{e \in \mathcal{E}} \sqrt{\|\alpha_e\|^2 + \|\beta_e\|^2}. \quad (60)$$

Substitute Equation (60) into inequality (15), and it follows that:

$$\|H(p)\| \leq c(p) \frac{1}{\sqrt{n}} \sum_{e \in \mathcal{E}} \|\delta_A^e\| \leq c(p) \|Ky\|. \quad (61)$$

If we take $c = \max_{p \in \Omega} c(p)$, then the following inequality holds:

$$\|H(p)\| \leq c \|Ky\|. \quad (62)$$

G EXPRESSIONS FOR $\nabla_J E_{\text{iso}}$ AND $\nabla_J^2 E_{\text{iso}}$

Following [Stomakhin et al. 2012], the expression of $\nabla_J E_{\text{iso}}$ and $\nabla_J^2 E_{\text{iso}}$ are given as follows (to avoid clutter, E substitutes E_{iso}):

$$\begin{aligned} \nabla_J E &= \sum_{i=1}^3 (V_{(i)} \otimes U_{(i)}) \frac{\partial E}{\partial \sigma_i}, \\ \nabla_J^2 E &= Q \Lambda Q^T, \end{aligned} \quad (63)$$

where $Q = (V \otimes U)P$, $J = USV^T$ is the SVD decomposition of J , P is a permutation matrix that maps column (1, 2, 3, 4, 5, 6, 7, 8, 9) to (1, 5, 9, 2, 4, 6, 8, 7, 3). Λ is a block diagonal matrix whose diagonal blocks are Λ_0 , Λ_1 , Λ_2 and Λ_3 , and Λ_0 is a 3×3 matrix with entries:

$$(\Lambda_0)_{ij} = \frac{\partial^2 E}{\partial \sigma_i \partial \sigma_j}, \quad i, j = 1, 2, 3,$$

while each Λ_i ($i = 1, 2, 3$) is a 2×2 matrix:

$$\Lambda_i = \begin{bmatrix} \frac{\partial E}{\partial \sigma_i} & -\frac{\partial E}{\partial \sigma_i} \\ -\frac{\partial E}{\partial \sigma_i} & \frac{\partial E}{\partial \sigma_i} \end{bmatrix} \begin{bmatrix} \sigma_j & -\sigma_i \\ -\sigma_i & \sigma_j \end{bmatrix}^{-1},$$

where $i = 1, 2, 3$, $j = (i \bmod 3) + 1$.

H UPPER BOUNDS FOR $\|\nabla \alpha_e\|^2 + \|\nabla \beta_e\|^2$ IN V_p^r

Let us introduce the following simplified notations:

$$m = \|l_u\| + \|l_w\|, \quad n = \|l_u\| \|l_w\|, \quad d = \|d_e\| = \|d_e\|.$$

Using a symbolic math software, we obtained the following expressions:

$$\|\nabla \alpha_e\|^2 = \frac{\sin^2 \theta \left(2n^2 (1 + \cos \theta) + m^2 (\|l_u\|^2 + \|l_w\|^2) \right) + 2m^2 d^2}{4\pi^2 n^2 (m^2 - d^2)^2}, \quad (64)$$

$$\|\nabla \beta_e\|^2 = \frac{2m^2 d^2 \cos \theta + 4m^2 d^2 - 2n \sin^2 \theta}{4\pi^2 n^2 (m^2 - d^2)^2}, \quad (65)$$

where θ denotes the angle between the vectors l_u and l_w , as shown in Figure 15. Consider the triangle formed by (q, u, w) . Obviously

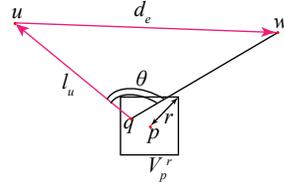


Fig. 15. Notations. $d_e = w - u$ is an edge vector on the cage. V_p^r is a voxel with center p and radius r . The point q lies inside the voxel, forming the triangle (q, u, w) .

$\|l_u\| \sin \theta$ is its height towards edge \overline{qw} , therefore smaller than d , and similarly $\|l_w\| \sin \theta < d$. Using Equation (64) we get:

$$\|\nabla \alpha_e\|^2 \leq \frac{1}{\pi^2 (m^2 - d^2)^2} + \frac{m^2 d^2}{\pi^2 n^2 (m^2 - d^2)^2},$$

which implies that:

$$\max_{q \in V_p^r} \|\nabla \alpha_e\|^2 \leq \frac{1}{\pi^2 (h_e^2 - d^2)^2} \left(1 + \frac{h_e^2 d^2}{(\|l_1\| - r)^2 (\|l_2\| - r)^2} \right), \quad (66)$$

where $h_e = \min_{q \in V_p^r} \|l_u\| + \|l_w\|$, $l_1 = \overrightarrow{pu}$ and $l_2 = \overrightarrow{pw}$.

Similarly, by taking $\theta = 0$, we obtain an upper bound for $\|\nabla \beta_e\|^2$:

$$\|\nabla \beta_e\|^2 \leq \frac{3m^2 d^2}{2\pi^2 (m^2 - d^2)^2 n^2},$$

which implies that

$$\max_{t \in V_p^r} \|\nabla \beta_e\|^2 \leq \frac{3h_e^2 d^2}{2\pi^2 (h_e^2 - d^2)^2 (\|l_1\| - r)^2 (\|l_2\| - r)^2}. \quad (67)$$