Controlling Singular Values with Semidefinite Programming

Shahar Z. Kovalsky* Noam Aigerman* Ronen Basri Yaron Lipman
Weizmann Institute of Science

Abstract

Controlling the singular values of n-dimensional matrices is often required in geometric algorithms in graphics and engineering. This paper introduces a convex framework for problems that involve singular values. Specifically, it enables the optimization of functionals and constraints expressed in terms of the extremal singular values of matrices.

Towards this end, we introduce a family of convex sets of matrices whose singular values are bounded. These sets are formulated using Linear Matrix Inequalities (LMI), allowing optimization with standard convex Semidefinite Programming (SDP) solvers. We further show that these sets are optimal, in the sense that there exist no larger convex sets that bound singular values.

A number of geometry processing problems are naturally described in terms of singular values. We employ the proposed framework to optimize and improve upon standard approaches. We experiment with this new framework in several applications: volumetric mesh deformations, extremal quasi-conformal mappings in three dimensions, non-rigid shape registration and averaging of rotations. We show that in all applications the proposed approach leads to algorithms that compare favorably to state-of-art algorithms.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry G.1.6 [Numerical Analysis]: Optimization;

Keywords: singular values, optimization, semidefinite programming, simplicial meshes

Links: DL PDF

1 Introduction

Linear transformations, or matrices, lie at the core of almost any numerical computation in science and engineering in general, and in computer graphics in particular.

Properties of matrices are often formulated in terms of their singular values and determinants. For example, the isometric distortion of a matrix, which can be formulated as its distance from the orthogonal transformation group, measures how close the singular values are to one; the condition number, or the conformal distortion of a matrix, is the ratio of its largest to smallest singular values; the stretch of a matrix, or its operator norm, is its largest singular value; a matrix is orientation preserving if its determinant is non-negative and is non-singular if its minimal singular value is positive.

*equal contributors

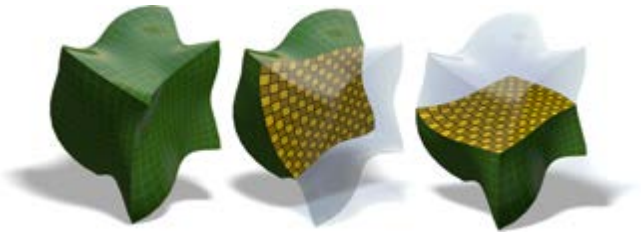


Figure 1: *The "most conformal" mapping of a volumetric cube subject to repositioning of its eight corners. Our framework minimizes the maximal conformal distortion in 3D, providing a unique glimpse to extremal quasiconformal maps in higher dimensions.*

The goal of this paper is to develop a convex framework for controlling singular values of square matrices of arbitrary dimension and, hence, facilitate applications in computer graphics that require optimizing functionals defined using singular values, or that require strict control over singular values of matrices.

The challenge in controlling singular values stems from their nonlinear and non-convex nature. In computer graphics, controlling singular values has received little attention while the focus was mostly on controlling specific functionals [Hormann and Greiner 2000; Sander et al. 2001; Floater and Hormann 2005], clamping singular values by means of projection [Wang et al. 2010], and controlling singular values in 2D [Lipman 2012]. Directly controlling singular values in higher dimensions than 2D is not straightforward. The difficulty in going beyond the two-dimensional case is demonstrated by the fact that the singular values of matrices in dimension three and higher are characterized as roots of polynomials of degree of at-least six, for which no analytic formula exists.

The key insight of this paper is a characterization of a complete collection of maximal convex subsets of $n \times n$ matrices with strict bounds on their singular values. By complete we mean that the union of these subsets covers the entire space of $n \times n$ matrices whose singular values are bounded, and maximal means that no other convex subset of $n \times n$ matrices with bounded singular values strictly contains any one of these subsets. These convex sets are formulated as Linear Matrix Inequalities (LMIs) and can be plugged as-is into a Semidefinite Program (SDP) solver of choice. Although SDP solvers are still not as mature as more classical convex optimization tools such as linear programming, and are lagging somewhat behind in terms of time-complexity, they are already efficient enough to enable many applications in computer graphics. Furthermore, regardless of any future progress in convex optimization, the maximality property of our subsets implies that one cannot hope to enlarge these subsets and stay within the convex regime.

Additionally, many problems require matrices that preserve orientation, i.e., matrices with non-negative determinant. This non-convex requirement is naturally addressed in our framework as well.

Our formulation can be used in a number of applications in geometry processing: volumetric mesh deformations, extremal quasiconformal mappings in three dimensions, non-rigid shape registration and averaging of rotations. We have experimented with these applications and show that in all cases our formulation leads to algorithms that compare favorably to state-of-art methods. Figure 1 depicts an example of an extremal quasiconformal deformation obtained with the proposed method.

2 Previous work

A number of studies in Computer Graphics deal with optimization related to singular values of matrices: [Lévy et al. 2002] propose Least-Squares Conformal Maps (LSCM), a convex functional that measures the distance of a 2×2 matrix from similarity, or, equivalently, minimizes the variance of the singular values; similarly, As-Rigid-As-Possible algorithms ([Alexa et al. 2000; Sorkine and Alexa 2007; Igarashi et al. 2005; Chao et al. 2010]) optimize a non-linear functional measuring the distance of a matrix to the rotation group, thus trying to minimize the distance of the singular values from 1. Works on surface parameterization have dealt with singular values of mappings to the plane, as these quantify desired properties. [Hormann and Greiner 2000] aim to minimize the ratio between singular values using a Frobenius norm relaxation; [Sorkine et al. 2002] flatten a cut mesh and then reassemble the pieces while bounding the extremal singular values; for surveys see [Floater and Hormann 2005; Hormann et al. 2007]. [Paillé and Poulin 2012] generate a volumetric parameterization that is as-conformal-as-possible via an adaptation of LSCM to 3D. [Freitag and Knupp 2002] improve tetrahedral meshes by reducing their aspect ratios. [Aigerman and Lipman 2013] project a given piecewise-linear map onto the space of bounded-distortion transformations.

Optimization problems involving singular values appear in other fields of research as well. [Polak and Wardi 1982] and later [Kiwiel 1986] consider problems in Control Theory that are expressed in terms of inequalities of singular values, however they optimize them in a local manner (*e.g.*, descent methods). [Maréchal and Ye 2009] and later [Lu and Pong 2011] address the problem of minimizing the condition number of positive semidefinite matrices. Their solution, however, is only applicable to PSD matrices.

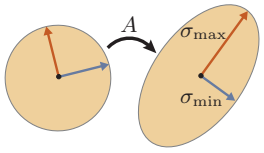
Projection based approaches have been considered for the optimization of problems involving singular values. For example, [Wang et al. 2010; Hernandez et al. 2013] employ an alternating projection approach to limit singular values. More generally, one could attempt to solve constrained minimization problems using gradient projection methods, which alternate between a gradient descent step that reduces the functional, and a projection step that enforces the constraints. However, projection on multiple constraints involving singular values is, in general, non-trivial, leading to a non-convex feasibility problem that should be solved at each step. In contrast, our method operates on convex subsets of these constraints; under proper initialization, both feasibility and monotonic reduction of the functional in each iteration are guaranteed.

[Lipman 2012] provides a characterization of bounded distortion and bounded isometry spaces for 2×2 matrices. His approach, which uses a second-order cone formulation, does not extend to higher dimension. Our paper, in contrast, characterizes similar spaces of $n \times n$ matrices in *any* dimension using linear matrix inequalities (LMIs), which are then used in Semidefinite Programming (SDP). In 2D our constraints can be shown to coincide with Lipman's bounded isometry characterization.

Solutions of applied problems with semidefinite programming date back to the work of [Goemans and Williamson 1995], who used it to approximate the MAX-CUT problem. [Vandenberghe and Boyd 1994; Boyd and Vandenberghe 2004] give a comprehensive overview of semidefinite programming. Recently, due to the continuing improvement in the efficiency of interior-point solvers, the popularity of SDP has increased. Examples of applications include manifold learning [Weinberger and Saul 2009], matrix completion [Candès and Recht 2009], estimation of rotations [Singer 2011] and surface reconstruction [Ecker et al. 2008]. It is however less commonly used in computer graphics; we are only aware of [Huang and Guibas 2013], which use SDP for jointly optimizing maps between a collection of shapes.

3 Preliminaries and problem statement

Definitions and notations. Let $A \in \mathbb{R}^{n\times n}$, and denote by $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_n(A)$ its singular values. We will also use the notation $\sigma_{\max} \triangleq \sigma_1$ and $\sigma_{\min} \triangleq \sigma_n$. Geometrically, $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ quantify, respectively, the largest and smallest change of Euclidean length induced by applying A to any vector (see inset). We say that a matrix A is *orientation preserving* if it satisfies $\det(A) \geq 0$. We use the notation $A \succeq 0$ to imply that A is a symmetric, positive semidefinite (PSD) matrix. Such an expression is called a linear matrix inequality (LMI) [Boyd and Vandenberghe 2004]. In the same manner, $A \succeq B$ implies that $A - B$ is PSD and thus for a scalar $c \in \mathbb{R}$, the equation $S \succeq cI$ implies that the eigenvalues of S are larger or equal to c. A semidefinite program (SDP) is a convex optimization problem formulated with LMI constraints and a linear objective. We note that any linear program, convex quadratic program and second-order cone program can be formulated as an SDP.



Goal and approach. Our goal in this paper is to characterize and provide an efficient algorithm for optimizing a class of problems formulated in terms of the minimal and maximal singular values of $n \times n$ matrices.

For example, consider the following toy problem:

$$\min_{A\in\mathbb{R}^{n\times n}} \quad \mathrm{f}(A) \tag{1a}$$

$$\text{s.t.} \quad \sigma_{\min}(A) \geq \Gamma^{-1} \tag{1b}$$

$$\sigma_{\max}(A) \leq \Gamma \tag{1c}$$

$$\det(A) \geq 0, \tag{1d}$$

for some constant $\Gamma \geq 1$. Intuitively, this problem describes the minimization of the functional $\mathrm{f}(A)$ under the constraint that the matrix A deviates by at most Γ from being a rotation. This is a non-convex problem even when f is convex, and we are unaware of any efficient algorithms for optimizing it in the general n-dimensional case.

Our goal is to present an algorithm for solving problems such as the one described above. More generally, we consider a broader class of problems in the form of the following *meta-problem*:

$$\min_{A\in\mathbb{R}^{n\times n}} \quad \mathrm{f}\left(A, \sigma_{\min}(A), \sigma_{\max}(A)\right) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{g}_i\left(A, \sigma_{\min}(A), \sigma_{\max}(A)\right) \leq 0\,, \quad i = 1, .., r \tag{2b}$$

$$\det(A) \geq 0, \tag{2c}$$

where $\mathrm{f}(A, x, y), \mathrm{g}_i(A, x, y)$ are convex functions that satisfy certain *monotonicity conditions* in x, y (as detailed in Section 5), and eq. (2c) ensures that A is orientation preserving.

Note that problem (1) readily fits this framework with $\mathrm{f}(A, x, y) = \mathrm{f}(A)$, $\mathrm{g}_1(A, x, y) = -x + \Gamma^{-1}$, and $\mathrm{g}_2(A, x, y) = y - \Gamma$.

In this paper we present a generic iterative algorithm for solving instantiations of the meta-problem. In a nutshell, each iteration of the algorithm solves a semidefinite program (SDP). The algorithm is shown to be monotonically decreasing and optimal in the sense that each iteration considers the "largest" convex sub-problem of the non-convex meta-problem.

We demonstrate several interesting applications in geometry processing and computer graphics that can be formulated in terms of singular values of matrices, and claim that we expect to find other applications in the fields of computer graphics and vision.

4 Bounding singular values using LMI's

The key to successful optimization of the meta-problem (2) is understanding how to bound the maximal singular value of a matrix from above, and the minimal singular value from below. To that end, let us define two subsets of $n \times n$ matrices: first, the set of all matrices whose maximal singular value is at most a constant Γ,

$$\mathcal{I}^{\Gamma} = \left\{ A \in \mathbb{R}^{n \times n} \,|\, \sigma_{\max}(A) \leq \Gamma \right\}. \tag{3}$$

Second, the subset of orientation-preserving matrices whose smallest singular value is at least a constant $\gamma \geq 0$,

$$\mathcal{I}_{\gamma} = \left\{ A \in \mathbb{R}^{n \times n} \,|\, \sigma_{\min}(A) \geq \gamma,\, \det(A) \geq 0 \right\}. \tag{4}$$

Working with $\mathcal{I}^{\Gamma}, \mathcal{I}_{\gamma}$, as defined above, is not straightforward. These sets are characterized in terms of roots of high-order polynomials; namely, the characteristic polynomial of $A^T A$ and the determinant of A. As such, one cannot directly employ these definitions in an optimization framework.

As we see next, the set $\mathcal{I}^{\Gamma}$ is a convex set in $\mathbb{R}^{n \times n}$ and can be precisely reformulated as an LMI. In contrast, however, $\mathcal{I}_{\gamma}$ is not convex and introduces a challenge. Nonetheless, we show it is possible to characterize its *maximal convex subsets* using a surprisingly simple LMI.

4.1 Bounding $\sigma_{\max}$ from above

The constraint $\sigma_{\max}(A) \leq \Gamma$ can be readily written as a convex LMI ([Boyd and Vandenberghe 2004], Section 4.6.3). We briefly summarize this formulation below. Let

$$\mathcal{C}^{\Gamma} = \left\{ A \in \mathbb{R}^{n \times n} : \begin{pmatrix} \Gamma I & A \\ A^T & \Gamma I \end{pmatrix} \succeq 0 \right\}. \tag{5}$$

Then, $A \in \mathcal{C}^{\Gamma} \Leftrightarrow A^T A \preceq \Gamma^2 I \Leftrightarrow \sigma_{\max}(A) \leq \Gamma$, where the first equivalence is an immediate consequence of Schur's complement. We therefore conclude that $\mathcal{C}^{\Gamma} = \mathcal{I}^{\Gamma}$.

4.2 Bounding $\sigma_{\min}$ from below

The space $\mathcal{I}_{\gamma}$, defined by the constraints $\sigma_{\min}(A) \geq \gamma$ and $\det(A) \geq 0$, is non-convex and thus more challenging. A common approach for dealing with non-convex sets is to replace them with convex sets that contain them (*e.g.*, their convex hulls). In our case, such a type of convexification will include matrices whose minimal singular values are not properly bounded, thus significantly deviating from our set of interest. Instead, we suggest working with convex sets *contained in* $\mathcal{I}_{\gamma}$. Specifically, we introduce a family of *maximal subsets* of $\mathcal{I}_{\gamma}$, which furthermore covers the entire space $\mathcal{I}_{\gamma}$. This allows us to devise effective optimization procedures and guarantees that the constraints of the problem are satisfied.

Our basic formula for characterizing the maximal convex subsets of $\mathcal{I}_{\gamma}$ is as simple as

$$\frac{A + A^T}{2} \succeq \gamma I. \tag{6}$$

For an arbitrary $\gamma \geq 0$ we define

$$\mathcal{C}_{\gamma} = \left\{ A \in \mathbb{R}^{n \times n} \,|\, \frac{A + A^T}{2} \succeq \gamma I \right\}. \tag{7}$$

$\mathcal{C}_{\gamma}$ is defined in terms of an LMI, and so it is readily convex and can be directly used in SDP optimization. The optimization framework we propose relies on the next observations:

$\mathcal{C}_{\gamma}$ is indeed a *convex subset* of $\mathcal{I}_{\gamma}$, and

1. $\mathcal{C}_{\gamma}$ is large. It is of full dimension, extending the set of symmetric matrices with bounded eigenvalues.
2. Furthermore, it is *maximal* in $\mathcal{I}_{\gamma}$ and can be used to generate a family of maximal convex subsets that cover it.

These properties suggest that $\mathcal{C}_{\gamma}$ is a good choice for our optimization framework. In fact, it is an optimal choice in the convex regime. Next, we elaborate on the properties of this set. To that end, we first observe that $\mathcal{C}_{\gamma}$ admits two alternative representations that help shed light on its properties,

$$\mathcal{C}_{\gamma} = \left\{ A \,|\, x^T A x \geq \gamma,\ \text{for all } \|x\|_2 = 1,\ x \in \mathbb{R}^n \right\}, \tag{8}$$

and

$$\mathcal{C}_{\gamma} = \{ S \,|\, S \succeq \gamma I \} \oplus \left\{ E \,|\, E = -E^T \right\}, \tag{9}$$

where $\oplus$ denotes the (internal) direct sum operator. In other words, $\mathcal{C}_{\gamma}$ is the set of matrices whose symmetric part is PSD with eigenvalues no less than γ, and an arbitrary antisymmetric part. The equivalency between (8) and (9) is immediate, by noticing that the decomposition $A = S + E$ is unique and that $x^T E x = 0$. To see the equivalency to (7), let $A = S + E$ as in (9); clearly, A satisfies the condition of (7) as $A + A^T = 2S$. Conversely, if A satisfies (7), then by definition of PSD matrices $x^T(A + A^T)x \geq 2\gamma$, which implies that A satisfies the condition of (8). With this we can state our main results:

Theorem 1. *$\mathcal{C}_{\gamma}$ is a convex subset of $\mathcal{I}_{\gamma}$.*

Proof. As previously mentioned, $\mathcal{C}_{\gamma}$ is convex as it is expressed in terms of an LMI. To prove it is a subset of $\mathcal{I}_{\gamma}$ we need to show that if $A \in \mathcal{C}_{\gamma}$ then $\sigma_{\min}(A) \geq \gamma$ and $\det(A) \geq 0$.

First, notice that if x is the (unit norm) singular vector of A corresponding to its minimal singular value, then

$$\sigma_{\min}(A) = \|Ax\|_2 = \|x\|_2 \|Ax\|_2 \overset{\text{(CS)}}{\geq} \langle x, Ax \rangle = x^T A x \overset{(8)}{\geq} \gamma$$

where the inequality labeled by (CS) is due to the Cauchy-Schwartz inequality.

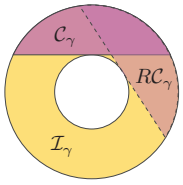
To see that $\det(A) \geq 0$, recall that it is the product of the eigenvalues λ_i of A. λ_i cannot be real and negative, or else it does not satisfy (8) as $x^T A x < 0 \leq \gamma$ for the corresponding eigenvector. Therefore, all the eigenvalues of A are either non-negative or complex, in which case they come in conjugate pairs, and so their product must be non-negative. □

Having established that $\mathcal{C}_{\gamma}$ is a convex subset of $\mathcal{I}_{\gamma}$, we seek to understand how large this set is. Definition (9) gives two immediate insights to this question: *(i)* $\mathcal{C}_{\gamma}$ is a set of full dimension, i.e. it has n^2 degrees of freedom, as the space of $n \times n$ matrices itself; *(ii)* it contains all $n \times n$ symmetric matrices with eigenvalues larger or equal to γ. Furthermore, it can be readily shown that $\mathcal{C}_{\gamma}$ contains all $n \times n$ rotation matrices with in-plane rotation angles $\theta_1, ..., \theta_k$ satisfying $|\theta_i| \leq \cos^{-1}(\gamma)$.

This suggests that $\mathcal{C}_\gamma$ is "rather large". Consequently, the question arises, whether it is the "largest" convex subset of $\mathcal{I}_\gamma$, in some sense. If the answer is no, it hypothetically means that one could optimize over larger pieces of $\mathcal{I}_\gamma$ and stay within the convex optimization regime. This will leave something to be desired. However, perhaps somewhat surprisingly, the answer is affirmative. As the following theorem shows (proven in the Appendix), $\mathcal{C}_\gamma$ is a maximal convex subset of $\mathcal{I}_\gamma$, meaning it cannot be added any other matrix from $\mathcal{I}_\gamma$ and stay convex.

Theorem 2. *$\mathcal{C}_\gamma$ is a* maximal *convex subset of $\mathcal{I}_\gamma$. That is, if another convex set $D \subset \mathcal{I}_\gamma$ satisfies $\mathcal{C}_\gamma \subseteq D$, then $\mathcal{C}_\gamma = D$.*

Orientation preserving matrices. Spaces of orientation preserving matrices are important in graphics, for example, in deformation and meshing applications [Schüller et al. 2013; Bommes et al. 2013]. Theorems 1 and 2 show that the convex space $\mathcal{C}_\gamma$ contains only orientation preserving matrices. Furthermore, they imply that $\mathcal{C}_0$, for $\gamma = 0$, is a maximal convex subset of the set of orientation preserving matrices, $\{A | \det(A) \geq 0\}$.



Covering $\mathcal{I}_\gamma$. The subset $\mathcal{C}_\gamma$ by itself does not cover the entire space $\mathcal{I}_\gamma$. However, rotated copies of $\mathcal{C}_\gamma$ can be used to cover $\mathcal{I}_\gamma$ in a natural manner, as the inset intuitively illustrates. The rotated copies of $\mathcal{C}_\gamma$ are completely equivalent to the original (unrotated) version $\mathcal{C}_\gamma$ except they cover different maximal pieces of the space $\mathcal{I}_\gamma$. The construction is simple: take an arbitrary $A \in \mathcal{I}_\gamma$, and let $A = RS$ be its *polar decomposition*[1]. Since $A \in \mathcal{I}_\gamma$, we necessarily have that R is a rotation and $S \succeq \gamma I$. Definition (9) implies that $S \in \mathcal{C}_\gamma$. Hence, $A \in R\mathcal{C}_\gamma$, where we denote $R\mathcal{C}_\gamma = \{RX \,|\, X \in \mathcal{C}_\gamma\}$. Since rotations preserve singular values and determinants, $R\mathcal{C}_\gamma \subset \mathcal{I}_\gamma$. Furthermore, $R\mathcal{C}_\gamma$ is also a maximal convex subset of $\mathcal{I}_\gamma$, for any rotation R. We therefore define a covering of $\mathcal{I}_\gamma$ via the family of its convex maximal subsets $R\mathcal{C}_\gamma$:

$$\mathcal{I}_\gamma = \bigcup_{R \in SO(n)} R\mathcal{C}_\gamma,$$

where $SO(n)$ denotes the $n \times n$ rotation matrices.

Choosing the rotation of $\mathcal{C}_\gamma$. For a given optimization problem formulated with $\mathcal{I}_\gamma$, the choice of R determines over which convex piece $R\mathcal{C}_\gamma \subset \mathcal{I}_\gamma$ the optimization will be performed. Assume we aim to optimize a given convex functional over $\mathcal{I}_\gamma$, and assume we are also given an initial guess $A \in \mathcal{I}_\gamma$. We would like to carry out the optimization in some neighbourhood of A contained in $\mathcal{I}_\gamma$. There are infinitely many choices of $R \in SO(n)$ such that $R\mathcal{C}_\gamma$ contains such a neighbourhood, and we would like to choose the "best" one in some sense. A sensible choice would be to choose R such that $R\mathcal{C}_\gamma$ is *symmetric* with respect to A; *i.e.*, such that if a rotation of A is in the convex space, so is its inverse rotation. The next lemma shows that choosing R to be the rotation term of the polar decomposition of A (as discussed in the previous paragraph) satisfies exactly this property:

Lemma 1. *Let $A \in \mathcal{I}_\gamma$, with polar decomposition $A = RS$. Then a rotation matrix $Q \in SO(n)$ satisfies $QA \in R\mathcal{C}_\gamma$ if and only if $Q^T A \in R\mathcal{C}_\gamma$.*

We prove this Lemma in the Appendix. Therefore, given an "initial guess" A, we shall choose $R\mathcal{C}_\gamma$ where R is extracted from the polar decomposition of A.

[1] Here, polar decomposition $A = RS$ means $R \in SO(n)$ and $S = S^T$.

5 Optimization of the meta-problem

We are now ready to formulate our algorithm for optimization of the meta-problem (2) presented previously. First, we need to complete the definition of the meta-problem by specifying the so-called monotonicity conditions,

Definition 1. *A function $\mathrm{f}(A, x, y) : \mathbb{R}^{n\times n} \times \mathbb{R}^2_{\leq} \to \mathbb{R}$ is said to satisfy the* monotonicity condition *if it is monotonically decreasing in variable x and monotonically increasing in variable y, where $\mathbb{R}^2_{\leq} = \{(x, y) \,|\, 0 \leq x \leq y\}$.*

Our meta-problem was defined in (2) along with the requirement that f, g_i are convex functions and satisfy the *monotonicity conditions*. The motivation behind the monotonicity condition is that it precisely characterizes the problems that allow an equivalent formulation in terms of the spaces $\mathcal{I}_\gamma, \mathcal{I}^\Gamma$ as follows

$$\begin{aligned} \min \quad & \mathrm{f}(A, \gamma, \Gamma) && \text{(10a)} \\ \text{s.t.} \quad & \mathrm{g}_i\,(A, \gamma, \Gamma) \leq 0\,, \quad i = 1, .., r && \text{(10b)} \\ & A \in \mathcal{I}^\Gamma && \text{(10c)} \\ & A \in \mathcal{I}_\gamma && \text{(10d)} \end{aligned}$$

The equivalence of this problem to the meta-problem (2) is proved in the Appendix.

Eq. (10c) is a convex constraint as explained in Section 4.1 and can be equivalently replaced with the LMI $A \in \mathcal{C}^\Gamma$. Eq. (10d) is the only non-convex part in the formulation above and can be treated as detailed in Section 4.2 by an LMI of the form $A \in R\mathcal{C}_\gamma$; the rotation R determines which maximal convex subset of $\mathcal{I}_\gamma$ we use. This leads to the following convex problem:

$$\begin{aligned} \min \quad & \mathrm{f}(A, \gamma, \Gamma) && \text{(11a)} \\ \text{s.t.} \quad & \mathrm{g}_i\,(A, \gamma, \Gamma) \leq 0\,, \quad i = 1, .., r && \text{(11b)} \\ & A \in \mathcal{C}^\Gamma && \text{(11c)} \\ & A \in R\mathcal{C}_\gamma. && \text{(11d)} \end{aligned}$$

As described in Section 4.2, we initialize $R = R^{(0)}$ from an initial guess $A^{(0)}$ by looking at its polar decomposition $A^{(0)} = R^{(0)}S^{(0)}$. After solving the convex problem we reset R according to the polar decomposition of the minimizer A and re-optimize. In each iteration of the algorithm the maximal convex set $R\mathcal{C}_\gamma$ is chosen to be symmetric w.r.t. the last result. The algorithm is outlined in Algorithm 1.

Although Algorithm 1 is not guaranteed to find a global minimum of the (generally non-convex) meta-problem (2), the maximality of the convex spaces $R\mathcal{C}_\gamma$ assures that, in each iteration of the algorithm, we consider the largest possible part of the non-convex set of $n \times n$ matrices defined by Eq. (10d). This gives the algorithm the best chance of avoiding local minima while restricting the solution to the feasible set of the original meta-problem. Another benefit is that it allows the algorithm to take big steps toward convergence and in practice this algorithm usually requires about 10-20 iterations to converge.

Lastly, we note that Algorithm 1 is guaranteed to monotonically decrease the functional value in each iteration since (as discussed in Section 4.2) the set $R\mathcal{C}_\gamma$ is guaranteed to contain A if its polar decomposition is $A = RS$. Hence, in the notation of Algorithm 1 the previous solution $A^{(n-1)}$ is always feasible in the n'th iteration.

Algorithm 1: Optimization of the meta-problem

Input: Convex functions f, g_i as in eq. (2)
Initial guess $A^{(0)}$
Output: Minimizer (local) A

$A^{(1)} = \infty \cdot \mathbf{1}\mathbf{1}^T$; `// matrix with all entries` ∞
$n = 0$;
while $\|A^{(n+1)} - A^{(n)}\|_F > \varepsilon$ **do**
  Compute the polar decomposition $A^{(n)} = R^{(n)} S^{(n)}$;
  Solve SDP (11) with $R = R^{(n)}$;
  Set $A^{(n+1)}$ to be the minimizer;
  $n = n + 1$;
return $A = A^{(n+1)}$;

Note that Algorithm 1 requires the SDP (11) to be feasible for the rotation $R^{(0)}$, extracted from $A^{(0)}$. This is a limitation of the algorithm, however in many practical cases a feasible initial rotation is either available or can be computed by solving a feasibility problem using the same algorithm (e.g., in the spirit of phase I methods, [Boyd and Vandenberghe 2004], Section 11.4).

5.1 Meta-problem for a collection of matrices

The applications presented in the next section require optimizing the meta-problem over a *collection* of matrices $A_1, .., A_j$ rather than just a single matrix. This requires generalizing the meta-problem (2) and its optimization algorithm (Algorithm 1) to this setup. This generalization is rather straightforward and is explained in this section.

For the multiple-matrix meta-problem $A_1, .., A_m \in \mathbb{R}^{n \times n}$ we define f, g_i to include all matrices and their maximal and minimal singular values as arguments:

$$\mathrm{f}\big(\ldots, A_j, \sigma_{\min}(A_j), \sigma_{\max}(A_j), \ldots\big),$$

and similarly for g_i. As with the single matrix meta-problem, we require f, g_i to be convex functions that satisfy the *monotonicity condition* for each pair $\sigma_{\min}(A_j), \sigma_{\max}(A_j)$. The convex formulation (11) now takes the form:

$$\begin{aligned}
\min \quad & \mathrm{f}(...A_j, \gamma_j, \Gamma_j ...) && && \text{(12a)}\\
\text{s.t.} \quad & \mathrm{g}_i\left(...A_j, \gamma_j, \Gamma_j ...\right) \le 0\,, && i = 1, .., r && \text{(12b)}\\
& A_j \in \mathcal{C}^{\Gamma_j}\,, && j = 1, .., m && \text{(12c)}\\
& A_j \in R_j \mathcal{C}_{\gamma_j}\,, && j = 1, .., m && \text{(12d)}
\end{aligned}$$

where R_j are the rotations that define the maximal convex spaces used for each matrix A_j. Algorithm 2 provides a straightforward adaptation of Algorithm 1 to the multi-matrix case. Similarly to Algorithm 1, Algorithm 2 also requires feasible initial rotations R_j.

6 Applications

In this section we apply our framework to several problems in geometry processing and use Algorithms 1,2 for their optimization. We show that for many applications this approach achieves favorable or comparable results to the state-of-art.

Algorithm 2: Optimization of the multi-matrix meta-problem

Input: Convex functions f, g_i
Initial guess $\left\{A_j^{(0)}\right\}_{j=1}^m$
Output: Minimizer (local) $\{A_j\}_{j=1}^m$

$A_j^{(1)} = \infty \cdot \mathbf{1}\mathbf{1}^T, \quad j = 1..m;$
$n = 0$;
while $\max_j \left\|A_j^{(n+1)} - A_j^{(n)}\right\|_F > \varepsilon$ **do**
  Compute the polar decompositions $A_j^{(n)} = R_j^{(n)} S_j^{(n)}$;
  Solve SDP (12) with $R_j = R_j^{(n)}$;
  Set $\left\{A_j^{(n+1)}\right\}_{j=1}^m$ to be the minimizer;
  $n = n + 1$;
return $\left\{A_j^{(n+1)}\right\}_{j=1}^m$;

6.1 Simplicial maps of meshes

Several of the applications we explore optimize and constrain *simplicial maps* of 3-dimensional meshes. We first set a few definitions and then show how different functionals and constraints of interest in geometry processing can be formulated and optimized in our framework.

Notations. We consider *simplicial maps* of 3-dimensional meshes $\mathbf{M} = (\mathbf{V}, \mathbf{T})$, where $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n] \in \mathbb{R}^{3\times n}$ is a matrix whose columns are the vertices, and $\mathbf{F} = \{t_j\}_{j=1}^m$ is the set of tetrahedra (tets). We denote by $|t_j|$ the normalized volume of the j'th tet (so that $\sum |t_j| = 1$). A simplicial map $\Phi : \mathbf{M} \to \mathbb{R}^3$ is a continuous piecewise-affine map that is uniquely determined by setting the mapping of each vertex $\boldsymbol{u}_i = \Phi(\boldsymbol{v}_i)$. We will represent an arbitrary simplicial map Φ of the mesh $\mathbf{M}$ with a matrix $\mathbf{U} = [\boldsymbol{u}_1, .., \boldsymbol{u}_n] \in \mathbb{R}^{3\times n}$. The restriction of Φ to each tet $t_j \in \mathbf{T}$ is an affine map $\Phi|_{t_j}(\mathbf{x}) = A_j\mathbf{x} + \boldsymbol{\delta}_j$, where A_j can be defined in terms of the unknowns $\mathbf{U}$ via the following linear system:

$$A_j \left[\boldsymbol{v}_{j_1}\ \boldsymbol{v}_{j_2} \cdots \boldsymbol{v}_{j_4}\right] E = \left[\boldsymbol{u}_{j_1}\ \boldsymbol{u}_{j_2} \cdots \boldsymbol{u}_{j_4}\right] E, \qquad (13)$$

where $j_1, .., j_4$ denote the indices of the vertices of the j'th tet, and E is a (singular) centering matrix given by $E = I - \frac{1}{4}\mathbf{1}\mathbf{1}^T$. This enables us to express the matrices A_j as linear functions of the variables $\mathbf{U}$, which we compute at preprocess. We denote this relation via $A_j(\mathbf{U})$.

The multi-matrix meta-problem can be readily adapted for optimizing simplicial maps with functionals and constraints formulated in terms of singular values:

$$\begin{aligned}
\min_{\mathbf{U}\in\mathbb{R}^{3\times n}} \quad & \mathrm{f}(\mathbf{U}, ..., A_j, \sigma_{\min}(A_j), \sigma_{\max}(A_j), ...) && \text{(14a)}\\
\text{s. t.} \quad & A_j = A_j(\mathbf{U}) && \text{(14b)}\\
& \mathrm{g}_i(\mathbf{U}, ..., A_j, \sigma_{\min}(A_j), \sigma_{\max}(A_j), ...) \le 0 && \text{(14c)}\\
& \det(A_j) \ge 0. && \text{(14d)}
\end{aligned}$$

In turn, we use Algorithm 2 for its optimization. Unless noted otherwise, we initialize Algorithm 2 with the identity map.

Note that constraining $\sigma_{\min}(A_j) \ge \varepsilon > 0$, in conjunction with (14d), implies that $\det(A_j)$ is strictly positive, which guarantees injectivity in the interior of the j'th tet. Global or local injectivity of the resulting simplicial maps may be further guaranteed with some additional assumptions [Lipman 2014].

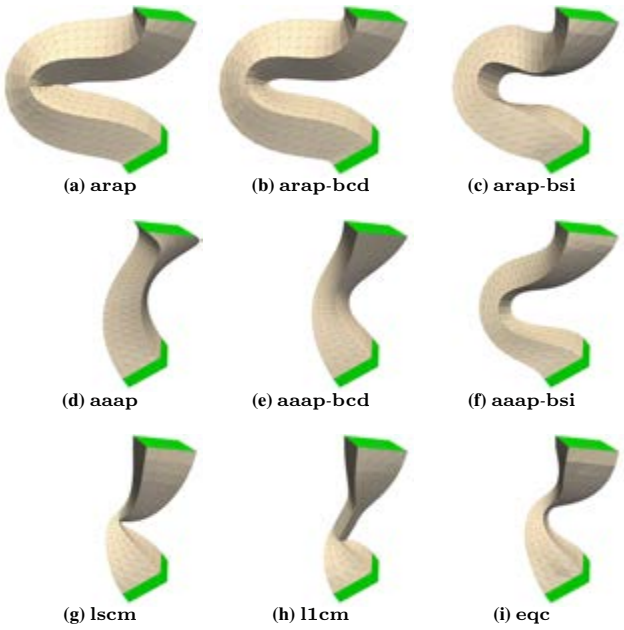


Figure 2: *Deformations obtained via optimizations formulated in terms of singular values. The green areas depict the positional constraints imposed on a volumetric bar. (a) optimizes the* **arap** *functional; (b),(c) the same functional while restricting either the conformal or scaled-isometric distortion. (c)-(e) repeats the comparison for the* **aaap** *functional. (g),(h) optimize the* **lscm** *functional and its ℓ_1 version* **l1cm**. *(i) shows the extremal quasiconformal deformation satisfying the constraints.*

We use two standard and popular functionals as a baseline for demonstrating our optimization framework:

1. *As-Rigid-As-Possible* (**arap**) energy [Alexa et al. 2000; Sorkine and Alexa 2007; Igarashi et al. 2005; Liu et al. 2008; Chao et al. 2010], defined as $\mathrm{f}_{\mathbf{arap}}(\mathbf{U}) = \sum_{j=1}^{m} \|A_j - R_j\|_F^2 |t_j|$, where $R_j \in SO(3)$ is the closest rotation to A_j.
2. *As-Affine-As-Possible* (**aaap**) smoothness energy $\mathrm{f}_{\mathbf{aaap}}(\mathbf{U}) = \sum_{t_i \backsim t_j} \|A_i - A_j\|_F^2 (|t_i| + |t_j|)$, where $t_i \backsim t_j$ implies two tets sharing a face.

The **aaap** functional is quadratic and convex, and hence fits into our meta-problem framework. The **arap** functional is not convex, however for fixed R_j it is quadratic and convex and fits into the meta-problem as well.

Both these functionals do not avoid flipping tets and may introduce arbitrarily high element distortion as shown in Figure 2: (a) shows an **arap** deformation result (we deform a bar, where the green areas depict the hard positional constraints used) which exhibits flipped tets and conformal distortion above 300, and (d) shows an **aaap** deformation result leading to conformal distortion above 8.

Constraints. Our first goal is to introduce spaces of 3D simplicial maps, that are orientation preserving (with no flipped tets) and have bounded amount of distortion. We express these in terms of constraints that involve singular values and demonstrate that optimizing functionals, such as **arap** or **aaap**, over these spaces produces plausible deformations. We have experimented with three flavors of spaces, for which we instantiate the meta-problem with the introduction of constraint functions g_i that satisfy the monotonicity condition:

1. *k-bounded isometry* (**bi**) maps forbid lengths to change by a factor greater than k. Namely, they satisfy $k^{-1} \leq \sigma_{\min}(A_j) \leq \sigma_{\max}(A_j) \leq k$. This formulates in our framework as the constraint functions $\mathrm{g}_{j,1}(\mathbf{U}) = \sigma_{\max}(A_j) - k$, and $\mathrm{g}_{j,2}(\mathbf{U}) = k^{-1} - \sigma_{\min}(A_j)$.
2. *k-bounded scaled isometry* (**bsi**) maps allow bounded k-isometric distortion with respect to a global isotropic scale $s > 0$. That is, $sk^{-1} \leq \sigma_{\min}(A_j) \leq \sigma_{\max}(A_j) \leq sk$. Taking s as a slack variable, this can be expressed as $\mathrm{g}_{j,1}(\mathbf{U}, s) = \sigma_{\max}(A_j) - sk$, and $\mathrm{g}_{j,2}(\mathbf{U}, s) = sk^{-1} - \sigma_{\min}(A_j)$.
3. *k-bounded conformal distortion* (**bcd**) maps forbid local length ratios to change by a factor greater than k. Thus, satisfying $\sigma_{\max}(A_j) \leq k\sigma_{\min}(A_j)$ which is expressed via $\mathrm{g}_j(\mathbf{U}) = \sigma_{\max}(A_j) - k\sigma_{\min}(A_j)$.

Figure 2 (b),(e) shows the result of optimizing the **arap**, **aaap** (resp.) restricted with the k-bounded conformal distortion, for $k = 2$. (c),(f) show the same functionals constrained with k-bounded scaled isometry. In both cases, the respective distortion in the final deformation is globally bounded by 2 and no tets are flipped.

Related work. Recent works have tackled similar problems. Schuller et al., [2013] introduced a barrier formulation to avoid flipping tets during optimization of similar energies, however their method is limited to the constraint $\det(A_j) \geq 0$ and cannot handle more elaborate singular value constraints. Aigerman et al., [2013] suggest an algorithm for projecting simplicial maps onto the set of bounded distortion maps, however this projection looks for a map close to an input initial map, and does not directly optimize a given energy. Our algorithm directly optimizes any convex energy over the space of bounded distortion maps. Table 1 compares the volumetric parameterization examples from Aigerman's paper to mappings achieved by minimizing the same energy (Dirichlet) using our algorithm, initialized by their results. Note that in all cases we decrease the Dirichlet energy of the map (we used the same bounds on the conformal distortion). See also Figure 3 for a visual comparison.

(a) *Our method* **(b)** *[Aigerman and Lipman 2013]*

Figure 3: *Volumetric parameterization – mapping a volume into a cube. Color encodes the Dirichlet energy per tet. Our approach achieves lower Dirichlet energy compared to that achieved by [Aigerman and Lipman 2013].*

	Verts	Tets	$\mathrm{f}_{\mathrm{our}}$	$\mathrm{f}_{\mathrm{aig}}$	$\#_{\mathrm{iter}}$
Duck	7k	13k	**10.4**	11.0	3
Max Plank	30k	40k	**11.0**	12.5	3
Hand	25k	41k	**10.3**	11.8	3
Sphinx	32k	43k	**3.8**	4.1	3
Bimba	32k	45k	**12.0**	13.1	4
Rocker	37k	60k	**26.3**	36.0	4

Table 1: *Volumetric parameterization – comparison to [Aigerman and Lipman 2013].* f_{our} *and* f_{aig} *are the Dirichlet energies of our and their solutions, and* $\#_{iter}$ *is the number of iterations our algorithm ran until convergence.*

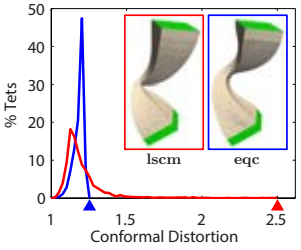
Functionals. Our framework further enables optimizing certain functionals that are formulated directly in terms of the singular values of the transformation matrices A_j. We explore several functionals that generalize *conformal* mappings to 3D:

1. *Least-Squares-Conformal-Maps* (**lscm**) [Lévy et al. 2002] can be generalized to any dimension by minimizing the spread of the singular values, *i.e.* the functional $\mathrm{f}_{\mathbf{lscm}}(\mathbf{U}) = \sum_{j=1}^{m} \left[\sigma_{\max}(A_j) - \sigma_{\min}(A_j)\right]^2 |t_j|$. This reduces to **lscm** in 2D, however it is no longer convex when considered in dimensions higher than two. Nonetheless, it is convex as a function of the singular values themselves and satisfies the monotonicity condition, and therefore can be optimized in the proposed framework.

2. *Sparse-Conformal-Maps* (**l1cm**) is an ℓ_1 version of the **lscm** functional defined by $\mathrm{f}_{\mathbf{l1cm}}(\mathbf{U}) = \sum_{j=1}^{n} \left[\sigma_{\max}(A_j) - \sigma_{\min}(A_j)\right] |t_j|$, which intuitively concentrates distortion in a sparse manner.

3. *Extremal Quasiconformal Distortion* (**eqc**) aims at minimizing the maximal conformal distortion and is defined via $\mathrm{f}_{\mathbf{eqc}}(\mathbf{U}) = \max_j \{\sigma_{\max}(A_j)/\sigma_{\min}(A_j)\}$. This functional is more challenging as it is not convex even when considered as a function of $\sigma_{\min}$ and $\sigma_{\max}$, however it is *quasi-convex* and satisfies the monotonicity condition. We show next that our framework can be extended to enable its optimization as well.

Figure 2 (g)-(i) shows deformations of a bar with these functionals. **lscm** and **l1cm** strive to minimize deviation from conformality, in the sense of minimizing the deviation from Cauchy-Riemann-type equations. **eqc** directly minimizes the maximal conformal distortion. The inset shows two distributions of conformal distortion, highlighting the difference between the **lscm** and **eqc** solutions: the **eqc** achieves much lower *maximal* conformal distortion than the **lscm** solution (as indicated by the triangles). Another interesting aspect is that **eqc** achieves almost constant conformal distortion, with most tets having distortion just below the maximal value. Although this behavior is well understood for extremal quasiconformal maps in 2D (see *e.g.*, [Weber et al. 2012]), we are unaware of any results of this kind for extremal quasiconformal maps in 3D. This optimization tool can be used to gain a first glimpse of these fascinating maps. Figures 1,4 depict a few extremal quasiconformal maps computed with our method (fully described below), by placing point constraints on a volume and moving them around. Note that although we only optimize the maximal conformal distortion, the minimizers are highly regular. This regularity is not trivial and indicates that this problem has an interesting underlying structure.



Minimizing maximal conformal distortion. Let us provide more details on the optimization of the **eqc** functional described above, as it deviates from our general framework. The core idea is to use its quasi-convex structure. For a fixed k, we consider the following optimization problem:

$$\min_{\mathbf{U}\in\mathbb{R}^{3\times n}} \quad \tau$$
$$\text{s.t.} \quad \sigma_{\max}(A_j) \le k\sigma_{\min}(A_j) + \tau, \qquad j = 1..m$$

with additional linear constraints on some of the columns of $\mathbf{U}$. For example, positional constraints of the form $\boldsymbol{u}_i = \boldsymbol{w}_i$.

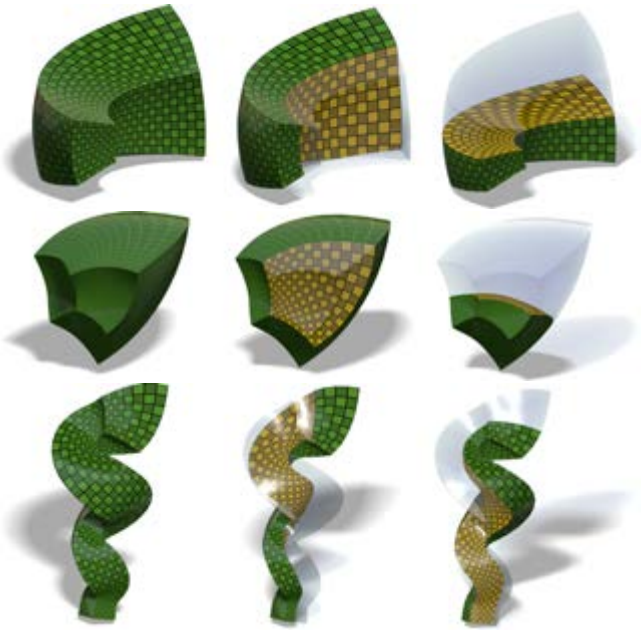


Figure 4: *Extremal quasiconformal mappings (**eqc**). Volumetric deformations that minimize the maximal conformal distortion.*

This can be interpreted as a k-**bcd** feasibility problem, where one seeks a map with maximal conformal distortion k. In fact, if a solution with $\tau < 0$ is found, it is guaranteed to have maximal conformal distortion strictly below k; this follows by noticing that $\frac{\sigma_{\max}}{\sigma_{\min}} \le k + \frac{\tau}{\sigma_{\min}} < k$ for $\tau < 0$.

For a fixed $k \ge 1$, this problem can be cast into our framework (14) with the choice

$$\mathrm{f}(\tau, \mathbf{U}, ...) = \tau \tag{15a}$$
$$\mathrm{g}_j(\tau, \mathbf{U}, ...) = \sigma_{\max}(A_j) - k\sigma_{\min}(A_j) - \tau, \tag{15b}$$

for $j = 1, \ldots, m$. These functions are convex in $\sigma_{\min}$ and $\sigma_{\max}$ and satisfy the monotonicity conditions.

We therefore run Algorithm 2 with eqs. (15), starting with $k >> 1$ (we used $k = 50$). Once a solution with $\tau < 0$ is found, we reset $k = \max_j \{\sigma_{\max}(A_j)/\sigma_{\min}(A_j)\}$ and reiterate Algorithm 2 until k has converged. Once an initial feasible result is found, each such iteration is guaranteed to be feasible, with monotonically decreasing maximal conformal distortion. See Figure 4 for examples of extremal quasiconformal mappings.

6.2 Non-Rigid ICP

We use our framework to introduce an alternative deformation model to a non-rigid Iterative Closest Point (ICP) framework. We suggest to directly control the deformation in terms of the maximal isometric distortion. We demonstrate how this leads to a more robust version of non-rigid ICP, producing favorable results compared to a baseline algorithm. Additional technical details on our implementation are provided in the supplementary material.

Non-rigid ICP [Allen et al. 2003; Brown and Rusinkiewicz 2007; Huang et al. 2008; Li et al. 2008] is a popular variant of the classical ICP algorithm [Besl and McKay 1992; Rusinkiewicz and Levoy 2001]. It aims to find a mapping Φ that registers two deformable surfaces $\mathcal{S}$ and $\mathcal{T}$ embedded in 3D.

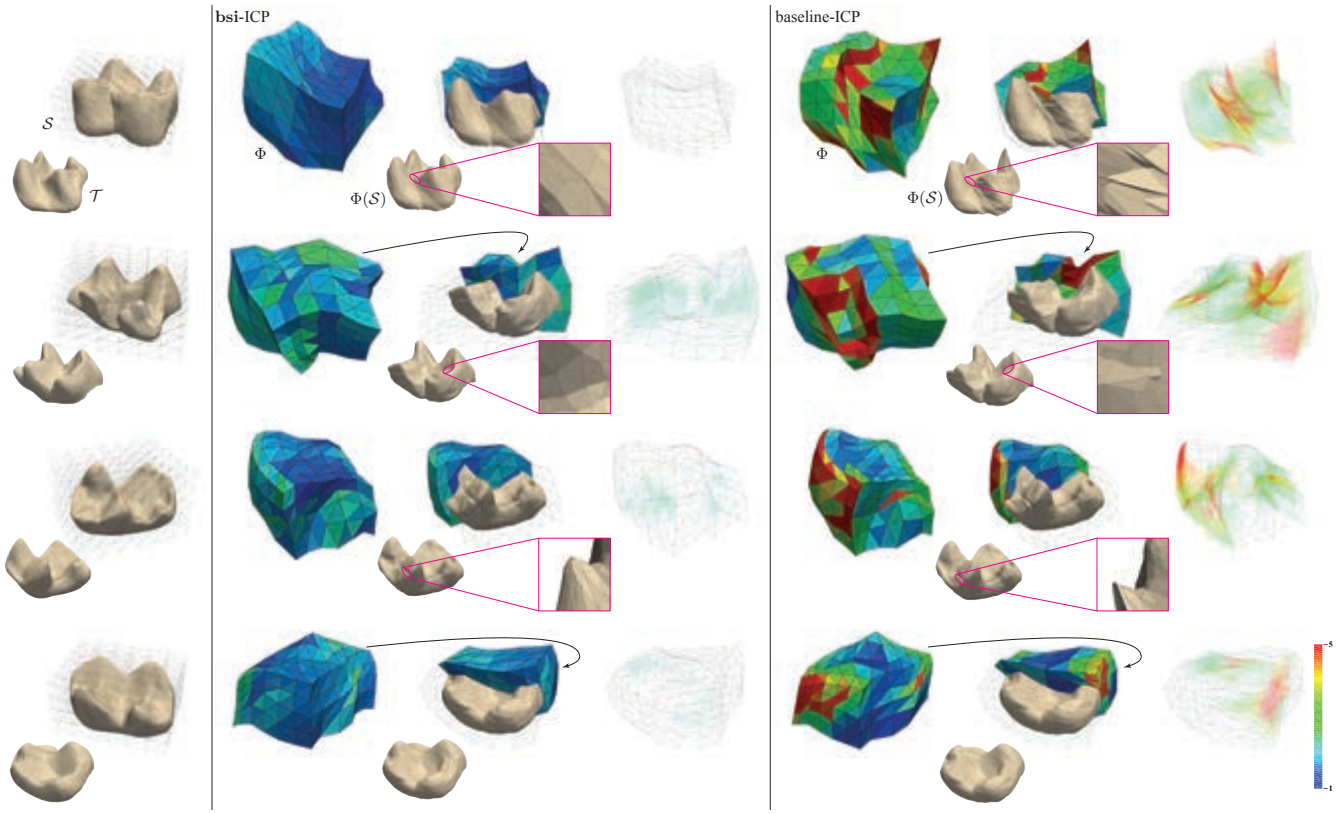


Figure 5: *Volumetric deformations induced by fitting bone surfaces. Left – source surface enclosed in volumetric tetrahedral mesh and target surface. Middle – deformed bone surface and induced volumetric deformation using the* **bsi***-ICP algorithm. Right – results obtained with the baseline algorithm. Color encodes isometric distortion.* **bsi***-ICP guarantees bounded isometric distortion and injectivity. The baseline algorithm, in contrast, tends to introduce high isometric distortion and to create artifacts on the deformed surface.*

Deformation model. Inspired by [Sumner et al. 2007; Li et al. 2008], we use a deformable tetrahedral mesh to model volumetric deformations of the source mesh $\mathcal{S}$. A deformation of the volume Φ then naturally induces a deformation of the source surface, which we denote by $\Phi(\mathcal{S})$.

For each point $\mathbf{p} \in \Phi(\mathcal{S})$ we compute the closest point $\mathbf{p}' \in \mathcal{T}$ and vice-versa for $\mathbf{q} \in \mathcal{T}$ compute its closest $\mathbf{q}' \in \Phi(\mathcal{S})$. We then define a fitting energy by

$$\mathrm{f}^2_{\mathbf{fit}}(\Phi) = \sum_{\mathbf{p}\in\Phi(\mathcal{S})} w_{\mathbf{p}} \left\|\mathbf{p} - \mathbf{p}'\right\|^2 + \sum_{\mathbf{q}\in\mathcal{T}} w_{\mathbf{q}} \left\|\mathbf{q} - \mathbf{q}'\right\|^2 \qquad (16)$$

where $w_{\mathbf{p}}$ is determined by the resemblance of the Heat Kernel Signatures (HKS) [Sun et al. 2009] of $\mathbf{p}$ and its closest point $\mathbf{p}'$ on $\Phi(\mathcal{S})$; $w_{\mathbf{q}}$ is defined similarly (see supplementary material).

We use an auxiliary tetrahedral mesh $\mathbf{M} = (\mathbf{V}, \mathbf{T})$ to define the deformation model. The deformation is then simply $\Phi = \Phi_{\mathbf{U}}$, a simplicial volumetric map defined in terms of $\mathbf{U} \in \mathbb{R}^{3\times n}$, as described in subsection 6.1.

We use either *(i)* a tetrahedral mesh enclosing the surface $\mathcal{S}$ (for space warping) or *(ii)* a mesh enclosed by the surface $\mathcal{S}$ (for articulation), see inset. In the first case, we encode each surface point $\mathbf{p} \in \mathcal{S}$ by its barycentric coordinates inside the relevant tet of $\mathbf{M}$. In the second case, the deformation mesh $\mathbf{M}$ does not necessarily contain $\mathcal{S}$, as seen in the inset; in this case we encode $\mathbf{p}$ as a linear combination of nearby vertices of $\mathbf{M}$, using a linear moving least squares approximation (additional details in the supplementary). In both cases, the deformed surface $\Phi(\mathcal{S})$ is represented as a linear function of the variables $\mathbf{U}$.

Optimization of baseline non-rigid ICP. We first describe the baseline algorithm, to which we compare our algorithm. This algorithm seeks to find a deformation Φ that minimizes

$$\mathrm{f}(\Phi) = \lambda_{\mathrm{f}}\mathrm{f}_{\mathbf{fit}}(\Phi) + \lambda_{\mathrm{s}}\mathrm{f}_{\mathbf{smooth}}(\Phi) + \lambda_{\mathrm{r}}\mathrm{f}_{\mathbf{rigid}}(\Phi), \qquad (17)$$

where $\mathrm{f}_{\mathbf{smooth}}$ and $\mathrm{f}_{\mathbf{rigid}}$ regularize the deformation. For the smoothness term $\mathrm{f}_{\mathbf{smooth}}(\Phi)$ we use the **aaap** energy, and for $\mathrm{f}_{\mathbf{rigid}}$ we use the **arap** energy, which penalizes for deviations from rigidity. (Both **aaap** and **arap** energies are defined in Section 6.1.)

Note that $\mathrm{f}(\Phi)$ is a convex quadratic function of the variables $\mathbf{U}$. It is optimized, following a standard ICP approach, by alternating between the following two steps:

1. For each $\mathbf{p} \in \Phi(\mathcal{S})$ compute the closest point $\mathbf{p}' \in \mathcal{T}$ and vice-versa for $\mathbf{q} \in \mathcal{T}$ compute its closest $\mathbf{q}' \in \Phi(\mathcal{S})$.
2. Optimize $\mathrm{f}(\Phi)$ given in equation (17).

In order to allow the surface $\mathcal{S}$ to gradually deform and fit to the target surface $\mathcal{T}$, the coefficient λ_{r} of the rigidity term is decreased with each iteration. Thus, allowing increasing levels of deformation. We choose to set $\lambda_{\mathrm{r}}^{(n)} = \lambda_{\mathrm{r}}^{\max}/\delta^n$ in the n'th iteration for $\delta > 1$, until reaching a minimal value $\lambda_{\mathrm{r}}^{\min}$.

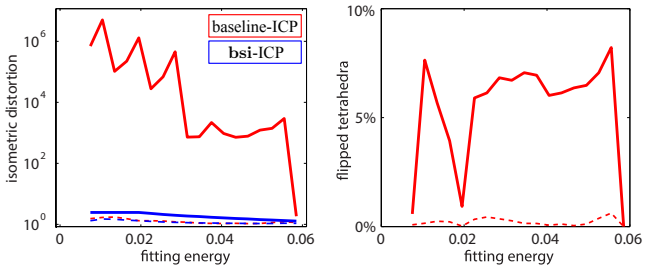


Figure 6: *The average and maximal isometric distortion (left) and number of flipped tetrahedra (right) obtained with the baseline and* **bsi***-ICP algorithms when applied to the anatomical surface dataset. Solid lines mark maximal values and dashed lines average values. The baseline algorithm tends to introduce high isometric distortions.*

Non-rigid ICP with bounded isometric distortion. Finding the balance between the different terms of eq. (17) is not straightforward and is usually resolved heuristically, as suggested above. Specifically, it is unclear how to set λ_{r} to allow only a certain amount of deformation. Furthermore, popular deformation energies, such as the **arap** energy, often concentrate isometric distortion unevenly, resulting in strong volumetric distortion and possibly non-injective maps. Consequently, the deformed surface $\Phi(\mathcal{S})$ suffers from the same problems as well. Thus, difficult fine-tuning may be required in order to approach state-of-the-art performance.

Instead, we suggest to simply replace the rigidity term in the functional (17) with the k-bounded scaled isometry constraint (**bsi**). Then, increasing k in each iteration of the algorithm directly controls the maximal isometric distortion allowed for Φ, thus avoiding the question of balancing the different energy terms.

Therefore, step (2) of the baseline ICP algorithm is replaced with the minimization of a simpler functional:

$$\mathrm{f}(\Phi) = \lambda_{\mathrm{f}}\mathrm{f}_{\mathbf{fit}}(\Phi) + \lambda_{\mathrm{s}}\mathrm{f}_{\mathbf{smooth}}(\Phi),$$

subject to the constraint that Φ is k-bounded scaled isometry. Optimization is performed using Algorithm 2, as described in section 6.1. The bound k is linearly increased $k^{(n)} = 1 + n\Delta$, until reaching a maximal value $k_{\max}$. In particular, for $k = 1$ the model reduces to the classical ICP algorithm, as the only simplicial maps Φ with scaled isometric distortion of 1 are global similarity transformations. Thus, the algorithm gradually transitions from classical ICP to non-rigid ICP. We denote this algorithm as **bsi**-ICP.

Anatomical surfaces dataset. In the first experiment we compared the baseline non-rigid ICP to the **bsi**-ICP algorithm on three datatsets of anatomical surfaces (bones) taken from [Boyer et al. 2011] which include 217 pairs of surfaces extracted from volumetric CT scans. The motivation here is to achieve well-behaved volumetric deformations that best fit the surfaces.

Figure 5 shows the result of the baseline ICP compared to our **bsi**-ICP on a few sample pairs of surfaces. Figure 6 depicts the tradeoff between the fitting energy and the amount of distortion and flipped tets over the entire dataset. It summarizes the results obtained with the two algorithms, where common parameters are set the same. Note that our **bsi**-ICP achieves similar fitting energy while maintaining a much lower distortion than the baseline and without introducing any flipped tets.

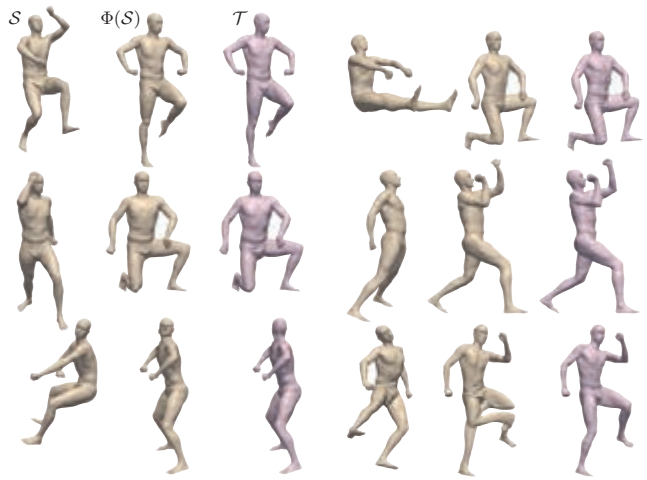


Figure 7: **bsi***-ICP applied to pairs of SCAPE models. Each triplet shows the source $\mathcal{S}$, its deformed version $\Phi(\mathcal{S})$ and target $\mathcal{T}$. Our approach successfully registers significant non-rigid deformations, with only an initial rigid alignment as input. It may however fail (bottom row) when the Euclidean closest point leads to bad alignment.*

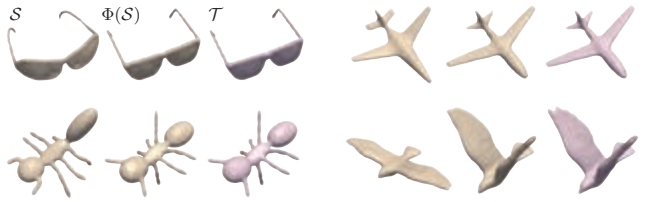


Figure 8: **bsi***-ICP applied to pairs of SHREC models. Each triplet shows the source $\mathcal{S}$, its deformed version $\Phi(\mathcal{S})$ and target $\mathcal{T}$.*

Other models. We have also tested our **bsi**-ICP algorithm on different models from the SCAPE [Anguelov et al. 2005] and SHREC 2007 [Giorgi et al. 2007] datasets. These models are more challenging for ICP-type algorithms due to the large changes of pose (SCAPE) and shape (SHREC). Nevertheless, we found that in many cases merely initializing the **bsi**-ICP with a reasonable rigid motion is enough to achieve good fitting results, as we demonstrate next.

Figure 9 shows the deformation sequence for **bsi**-ICP (top row) and the baseline algorithm (bottom row) for a pair of SCAPE models. Note that the bounded-isometric deformation model better preserves the shape of the model during deformation and at the end result. Figure 7 shows a collection of results of the **bsi**-ICP algorithm on pairs of SCAPE models. Note that the algorithm is able to reproduce rather large deformations with only an initial rigid alignment as input. Bottom row shows failure cases, in which wrong correspondences, due to the use of Euclidean closest point matching, led to bad alignment.

The SHREC dataset is extremely challenging as inter-class surfaces introduce large shape variability and a simple deformation model (*i.e.*, volumetric deformations of an auxiliary mesh $\mathbf{M}$) no longer well-represents the deformation between arbitrary pairs. Nevertheless, Figure 8 shows that in some cases the **bsi**-ICP achieves pleasing results with pairs of the same class.

Figure 9: *Deformation sequence of a pair of SCAPE models. The sequence, from left to right, shows the deformation $\Phi(\mathcal{S})$ of the source surface $\mathcal{S}$ towards the target surface $\mathcal{T}$. Our* **bsi**-*ICP (top) is compared to the baseline algorithm (bottom). Directly bounding the isometric distortion of the deformation better preserves the shape of the model during the deformation and at the end result.*

6.3 Averaging rotations

In this last application we exemplify how our framework applies to different types of problems than optimization of simplicial maps. We chose the classical problem of averaging of rotations; that is, given a set of rotations $R^1, ..., R^k \in SO(3)$ and non-negative weights $w_1, ..., w_k$ that sum up to one, we want to calculate a rotation R^* that plays the role of their weighted average. One way to define an average is via the Karcher mean, which generalizes the Euclidean mean to the manifold case [Karcher 1977]:

$$R^* = \operatorname*{argmin}_{R \in SO(3)} \sum_{j=1}^{k} w_j \operatorname{dist}\left(R, R^j\right)^2, \tag{18}$$

where $\operatorname{dist}\left(R, R^j\right)$ is the geodesic distance between the two rotations in the rotation manifold $SO(3)$.

Methods for approximating the Karcher mean on either the manifolds of rotations or PSD matrices have been studied in [Rentmeesters and Absil 2011; Jeuris et al. 2012]. These usually use local gradient or Newton methods, while taking advantage of the log-exp maps, and typically require fine tuning (*e.g.*, of line search step size). In computer graphics, [Alexa 2002] defined averages of transformations by exploiting the linear structure at the tangent space (using the log and exp maps). [Rossignac and Vinacua 2011] consider the interpolation of pairs of affine transformations; they further determine the conditions on which this interpolation is stable. We show that the problem of averaging rotations can be cast into our framework, producing approximations to the weighted Karcher mean, without computing the log or exp of any transformations. We start by showing how to approximate geodesics on the rotation group and then extend it to the weighted Karcher mean of several rotations, eq. (18).

Discretization of geodesics on $SO(3)$. Constant speed geodesics $\Upsilon : [0, 1] \to SO(3)$ on $SO(3)$, seen as a Riemannian manifold, can be formulated in a variational form as critical points of the energy functional

$$\mathrm{f}(\Upsilon) = \int_0^1 \left\| \dot{\Upsilon}(t) \right\|_F^2 dt. \tag{19}$$

In the discrete case, we subdivide the unit interval into equal-length segments $0 = t_0 < t_1 < ... < t_n = 1$, where $\Delta t = t_{i+1} - t_i = 1/n$ and consider the piecewise linear curve $\Upsilon = [R_0, R_1, ..., R_n]$. Observing that $\dot{\Upsilon}(t) = n\,(R_{i+1} - R_i)$ for $t \in (t_{i+1}, t_i)$, we calculate $\mathrm{f}(\Upsilon)$ using eq. (19):

$$\mathrm{f}(\Upsilon) = \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} \left\| \dot{\Upsilon}(t) \right\|_F^2 dt = n \sum_{i=0}^{n-1} \|R_{i+1} - R_i\|_F^2 . \tag{20}$$

Note that this discretization satisfies two desirable properties, similarly to the continuous case: *(i)* $\operatorname{length}(\Upsilon)^2 = \left[\sum_{i=1}^{n-1} \|R_{i+1} - R_i\|_F\right]^2 \leq n \sum_{i=1}^{n-1} \|R_{i+1} - R_i\|_F^2 = \mathrm{f}(\Upsilon)$, and *(ii)* if Υ is of constant speed, that is $\|R_{i+1} - R_i\|_F = c$, then $\operatorname{length}(\Upsilon)^2 = \mathrm{f}(\Upsilon)$. We note that $\operatorname{length}(\Upsilon)$ is a discrete approximation to $\operatorname{dist}\left(R^0, R^n\right)$.

Therefore, we can calculate geodesics on $SO(3)$ between two rotations G_a and G_b by minimizing $\mathrm{f}(\Upsilon)$ subject to the constraint that $R_0 = G_a$, $R_n = G_b$, and $R_i \in SO(3)$. The latter constraint is not convex, as the rotation group is not a convex set. However, since our functional $\mathrm{f}(\Upsilon)$ is contractive, it is sufficient to constrain $\sigma_{\min}(R_i) \geq 1$. This leads to the following optimization problem:

$$\begin{aligned} \min \quad & \mathrm{f}(\Upsilon) && && \text{(21a)} \\ \text{s.t.} \quad & \sigma_{\min}(R_i) \geq 1, && i = 1, ..., n-1 && \text{(21b)} \\ & R_0 = G_a, \quad R_n = G_b. && && \text{(21c)} \end{aligned}$$

Note that $\mathrm{f}(\Upsilon)$ is a convex quadratic function in the matrices R_i and the constraint $\sigma_{\min}(R_i) \geq 1$ can be easily realized in our framework. Hence, we can optimize (21) using Algorithm 2. We initialize R_i with the linear interpolant $R_i = (1 - t_i)G_a + t_i G_b$. Empirically, we have observed that this minimization results in a piecewise linear curve Υ of a constant speed; moreover, it precisely reproduces the geodesic in $SO(3)$ at times t_i (as can be computed, *e.g.*, with SLERP [Shoemake 1985]). Below is a result of such an optimization:

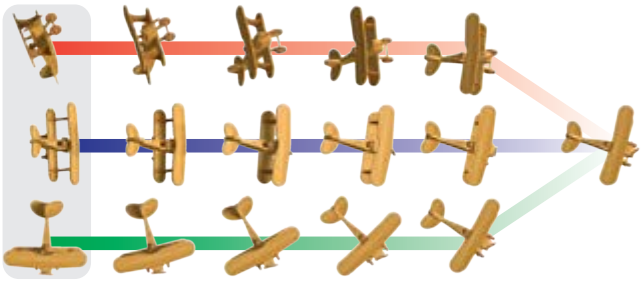


Figure 10: *Approximate Karcher mean. The rotation on the right approximates the Karcher mean (with equal weights) of the three rotations given in the left column. Each row illustrates a geodesic.*

Karcher mean. We proceed to optimizing the weighted Karcher mean, eq. (18). Recall that we aim to compute the weighted average of the rotations $R^1, \ldots, R^k$. To this end, we employ the geodesic discretization by defining k piecewise linear curves $\Upsilon^j = [R_0^j, R_1^j, ..., R_n^j]$, where $R_0^j = R^j$. We then optimize

$$\min_{R, R_i^j \in \mathbb{R}^{3\times 3}} \quad \sum_{j=1}^{k} w_j \mathrm{f}(\Upsilon^j) \tag{22a}$$

$$\text{s.t.} \quad \sigma_{\min}(R_i^j) \geq 1, \qquad \forall i, j \tag{22b}$$

$$R_n^j = R, \quad R_0^j = R^j. \qquad \forall j \tag{22c}$$

Following the observations above, constant speed minimizers of (21) satisfy $\mathrm{f}(\Upsilon^j) = \mathrm{length}(\Upsilon^j)^2 \approx \mathrm{dist}\left(R^j, R\right)^2$, and therefore the minimizer R of problem (22) is our approximation of the weighted Karcher mean.

As before, this problem fits into our optimization framework and can be solved with Algorithm 2. We initialize the algorithm in two steps: first, we solve (22) with $n = 1$ (single segment geodesics) with R initialized as the Euclidean centroid of $R^1, ..., R^k$; then, we initialize each of the geodesics $R^j \to R$ by optimizing (21). We note that the Karcher mean on the Rotation group $SO(3)$ is unique if all rotations $R^1, ..., R^k$ belong to a ball (on the manifold) of diameter at most π [Rentmeesters and Absil 2011]. The inset shows the discrete Karcher energy eq. (22a) as a function of the number of segments n used in each geodesic. As expected, the energy is increasing and converging. (Recall that the discrete piecewise linear curves "short-cut" the rotation manifold.) Figure 10 shows the result of optimizing (22). The rotation on the right hand side is the approximate Karcher mean R^*, and each row illustrates the geodesic $R^j \to R^*$.

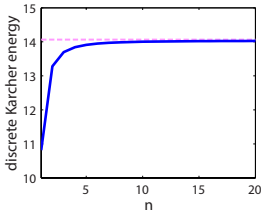
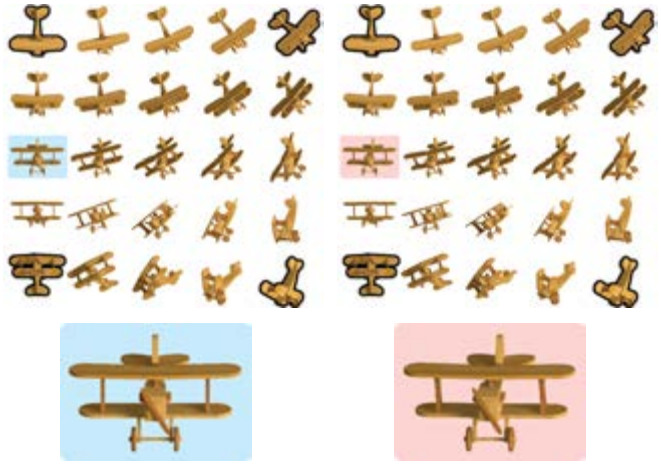


Figure 11 shows an application of the weighted Karcher mean for exploring rotations. In this case, the input are the four rotations in the corners (highlighted with solid borders); different weighted combinations of these rotations, with our approximation and [Alexa 2002], are shown on a grid. Note that Alexa's averaging, although mathematically elegant, does not produce exact geodesics on the borders of the square; namely, it deviates from the in-plane rotation, as emphasized in the blow-up. [Rossignac and Vinacua 2011] can also be used to produce similar output (BiSAM), however, unlike our averaging their approach is limited to generating tensor-product patterns.

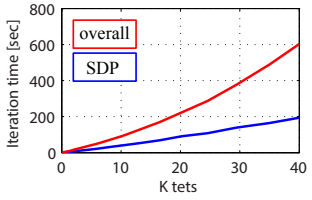


(a) *Our method* **(b)** *[Alexa 2002]*

Figure 11: *Exploring rotations – different weighted combinations of the 4 fixed rotations in the corners (top). Comparing* **(a)** *our approximate Karcher mean result with* **(b)** *Alexa's averaging. The latter does not produce exact geodesics on the boundaries of the grid, as seen in the blowup (bottom).*

7 Implementation details

We implemented our algorithm in Matlab, using YALMIP for the modeling of semidefinite programs [Löfberg 2004] and MOSEK [Andersen and Andersen 1999] for its optimization. All timings were measured on a single core of a 3.50GHz Intel i7. The inset shows typical runtimes of a single iteration of Algorithm 2, used for **bcd**-constrained deformations of tetrahedral meshes of various sizes. Roughly half the time is spent on the semidefinite optimization (MOSEK), while the rest is an overhead spent on problem setup (YALMIP); a more efficient implementation can significantly reduce this overhead. We further note that our SDP model is quite untypical (*e.g.*, has extremely many low-dimensional LMI constraints, much more than the number of variables); thus, standard SDP solvers may be non-optimal. Typical overall optimization time in several of the applications in the paper: Computation of Karcher mean with 5 links took 2 seconds (Figure 10); volumetric parameterization converged in 3-4 iterations, which took 28 minutes for the Max Plank model with 40k tets (Figure 3); extremal quasiconformal deformation of a cube with 16.5k tets converged in 11 iteration which took 46 minutes (Figure 1); non-rigid ICP registration took less than 20 minutes for each pair of the anatomical surface dataset (Figure 5) and 1 hour for a pair of SCAPE models (Figure 9).



8 Concluding remarks

In this paper, we have developed a framework for optimizing a family of problems formulated in terms of the minimal and maximal singular values of matrices. We use linear matrix inequality constraints to characterize maximal convex subsets of the set of orientation preserving matrices whose singular values are bounded. This leads to an effective convex optimization framework for an entire class of highly non-convex problems, and, in turn, to a single algorithm that applies to a variety of geometry processing problems. We apply this method to a collection of problems in computer graphics, and expect to find more applications in related fields.

As of the present time, the main limitation of the proposed framework is its time complexity. SDP solvers still lag behind simpler conic solvers and optimization time may be considerable, as described above. Nevertheless, we believe that a customized SDP solver, tailored to the structure of problems that arise in computer graphics, can be designed and has the potential for significant speed-up. We plan this as future work.

Acknowledgements This research was supported in part by the European Research Council (ERC Starting Grant SurfComp, Grant No. 307754), U.S.-Israel Binational Science Foundation, Grant No. 331/10, by the Israel Science Foundation Grants No. 1284/12 and 764/10, I-CORE program of the Israel PBC and ISF (Grant No. 4/11), the Israeli Ministry of Science, and by the Citigroup Foundation. The authors would like thank Gilles Tran for the airplane and ladybug models, Johan Löfberg for providing and supporting Yalmip, Ethan Fetaya for helpful discussions, and the anonymous reviewers for their useful comments and suggestions.

References

AIGERMAN, N., AND LIPMAN, Y. 2013. Injective and bounded distortion mappings in 3d. *ACM Trans. Graph. 32*, 4, 106–120.

ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. Proc. SIGGRAPH, 157–164.

ALEXA, M. 2002. Linear combination of transformations. *ACM Trans. Graph. 21*, 3 (July), 380–387.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph. 22*, 3 (July), 587–594.

ANDERSEN, E. D., AND ANDERSEN, K. D. 1999. *The MOSEK interior point optimization for linear programming: an implementation of the homogeneous algorithm*. Kluwer Academic Publishers, 197–232.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (July), 408–416.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2 (Feb.), 239–256.

BOMMES, D., CAMPEN, M., EBKE, H.-C., ALLIEZ, P., AND KOBBELT, L. 2013. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph. 32*, 4 (July), 98:1–98:12.

BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.

BOYER, D. M., LIPMAN, Y., ST. CLAIR, E., PUENTE, J., PATEL, B. A., FUNKHOUSER, T., JERNVALL, J., AND DAUBECHIES, I. 2011. Algorithms to automatically quantify the geometric similarity of anatomical surfaces. *Proceedings of the National Academy of Sciences 108*, 45, 18221–18226.

BROWN, B. J., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-d scans. *ACM Trans. Graph. 26*, 3 (July).

CANDÈS, E. J., AND RECHT, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics 9*, 6, 717–772.

CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph. 29*, 4, 38.

ECKER, A., JEPSON, A. D., AND KUTULAKOS, K. N. 2008. Semidefinite programming heuristics for surface reconstruction ambiguities. In *ECCV 2008*. Springer, 127–140.

FLOATER, M. S., AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Springer, 157–186.

FREITAG, L. A., AND KNUPP, P. M. 2002. Tetrahedral mesh improvement via optimization of the element condition number. *International Journal for Numerical Methods in Engineering 53*, 6, 1377–1391.

GIORGI, D., BIASOTTI, S., AND PARABOSCHI, L., 2007. Shape retrieval contest 2007: Watertight models track.

GOEMANS, M. X., AND WILLIAMSON, D. P. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM 42*, 6 (Nov.), 1115–1145.

HERNANDEZ, F., CIRIO, G., PEREZ, A. G., AND OTADUY, M. A. 2013. Anisotropic strain limiting. In *Proc. of Congreso Español de Informática Gráfica*.

HORMANN, K., AND GREINER, G. 2000. MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*. Vanderbilt University Press, 153–162.

HORMANN, K., LÉVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice. In *ACM SIGGRAPH 2007 Courses*, ACM, New York, NY, USA, SIGGRAPH '07.

HUANG, Q., AND GUIBAS, L. 2013. Consistent shape maps via semidefinite programming. *Proc. Eurographics Symposium on Geometry Processing 32*, 5, 177–186.

HUANG, Q.-X., ADAMS, B., WICKE, M., AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. In *Proc. Eurographics Symposium on Geometry Processing*, 1449–1457.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph. 24*, 3 (July), 1134–1141.

JEURIS, B., VANDEBRIL, R., AND VANDEREYCKEN, B. 2012. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39*, 379–402.

KARCHER, H. 1977. Riemannian center of mass and mollifier smoothing. *Comm. pure and applied mathematics 30*, 5, 509–541.

KIWIEL, K. 1986. A linearization algorithm for optimizing control systems subject to singular value inequalities. *IEEE Transactions on Automatic Control 31*, 7, 595–603.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. 21*, 3 (July), 362–371.

LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Proc. Eurographics Symposium on Geometry Processing 27*, 5.

LIPMAN, Y. 2012. Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph. 31*, 4, 108.

LIPMAN, Y. 2014. Bijective mappings of meshes with boundary and the degree in mesh processing. *SIAM J. Imaging Sci., to appear*.

LIU, L., ZHANG, L., XU, Y., GOTSMAN, C., AND GORTLER, S. J. 2008. A local/global approach to mesh parameterization. *Proc. Eurographics Symposium on Geometry Processing 27*, 5, 1495–1504.

LÖFBERG, J. 2004. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*.

LU, Z., AND PONG, T. K. 2011. Minimizing condition number via convex programming. *SIAM J. Matrix Analysis Applications 32*, 4, 1193–1211.

MARÉCHAL, P., AND YE, J. J. 2009. Optimizing condition numbers. *SIAM Journal on Optimization 20*, 2, 935–947.

PAILLÉ, G.-P., AND POULIN, P. 2012. As-conformal-as-possible discrete volumetric mapping. *Computers & Graphics 36*, 5, 427–433.

POLAK, E., AND WARDI, Y. 1982. Nondifferentiable optimization algorithm for designing control systems having singular value inequalities. *Automatica 18*, 3, 267 – 283.

RENTMEESTERS, Q., AND ABSIL, P.-A. 2011. Algorithm comparison for karcher mean computation of rotation matrices and diffusion tensors. In *Proc. European Signal Processing Conference*, EURASIP, 2229–2233.

ROSSIGNAC, J., AND VINACUA, A. 2011. Steady affine motions and morphs. *ACM Trans. Graph. 30*, 5 (Oct.), 116:1–116:16.

RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the ICP algorithm. In *Int. Conf. 3D Digital Imaging and Modeling*.

SANDER, P. V., SNYDER, J., GORTLER, S. J., AND HOPPE, H. 2001. Texture mapping progressive meshes. Proc. SIGGRAPH, 409–416.

SCHÜLLER, C., KAVAN, L., PANOZZO, D., AND SORKINE-HORNUNG, O. 2013. Locally injective mappings. *Proc. Eurographics Symposium on Geometry Processing 32*, 5, 125–135.

SHOEMAKE, K. 1985. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph. 19*, 3 (July), 245–254.

SINGER, A. 2011. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis 30*, 1, 20 – 36.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proc. Eurographics Symposium on Geometry Processing*, 109–116.

SORKINE, O., COHEN-OR, D., GOLDENTHAL, R., AND LISCHINSKI, D. 2002. Bounded-distortion piecewise mesh parameterization. In *Proc. Conference on Visualization '02*, VIS '02, 355–362.

SUMNER, R. W., SCHMID, J., AND PAULY, M. 2007. Embedded deformation for shape manipulation. *ACM Trans. Graph. 26*, 3.

SUN, J., OVSJANIKOV, M., AND GUIBAS, L. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. Eurographics Symposium on Geometry Processing*, 1383–1392.

VANDENBERGHE, L., AND BOYD, S. 1994. Semidefinite programming. *SIAM Review 38*, 49–95.

WANG, H., O'BRIEN, J., AND RAMAMOORTHI, R. 2010. Multi-resolution isotropic strain limiting. In *ACM SIGGRAPH Asia 2010*, 156:1–156:10.

WEBER, O., MYLES, A., AND ZORIN, D. 2012. Computing extremal quasiconformal maps. *Computer Graphics Forum 31*, 5, 1679–1689.

WEINBERGER, K. Q., AND SAUL, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res. 10* (June), 207–244.

Appendix

Proof of maximality. Assume towards contradiction that there exists a convex set $\mathcal{D}$ such that $\mathcal{C}_\gamma \subsetneq \mathcal{D} \subset \mathcal{I}_\gamma$. Then, let $B \in \mathcal{D} \backslash \mathcal{C}_\gamma$, and let $B = S + E$ be its decomposition into a sum of a symmetric and skew-symmetric matrices. Let $S = U\Lambda U^T$ be the spectral decomposition of S, with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. Then $B \notin \mathcal{C}_\gamma$ implies that $\lambda_n < \gamma$. Below we find a matrix $C \in \mathcal{C}_\gamma$ for which $\frac{B+C}{2} \notin \mathcal{I}_\gamma$, which by convexity entails $\mathcal{D} \not\subset \mathcal{I}_\gamma$, in contradiction.

We select C to have the form $C = U\Delta U^T - E$ with a diagonal matrix $\Delta = \operatorname{diag}(\delta_1, \ldots, \delta_n)$ whose entries are set as follows: $\delta_i = 1 + 2\gamma + |\lambda_i|$ for $i = 1, \ldots, n-1$ and $\delta_n = \gamma$. Clearly, all the diagonal entries $\delta_i \geq \gamma$ and so $C \in \mathcal{C}_\gamma$. However,

$$\frac{B+C}{2} = U\frac{(\Lambda + \Delta)}{2}U^T,$$

and the diagonal entries of $\frac{\Lambda+\Delta}{2}$ satisfy $\frac{\lambda_i+\delta_i}{2} > \gamma \geq 0$ for $i = 1, \ldots, n-1$ and $\frac{\lambda_n+\delta_n}{2} < \gamma$. Consequently, the latter entry is either negative, in which case the product of the diagonal values, and hence the determinant, is negative, or it is non-negative and strictly smaller than γ, in which case $\sigma_{\min} < \gamma$, therefore $\frac{B+C}{2} \notin \mathcal{I}_\gamma$ in contradiction.

Proof of Lemma 1. Suppose $QA \in R\mathcal{C}_\gamma$. Recall that $A = RS$. The definition of $\mathcal{C}_\gamma$ then implies that $R^TQRS + SR^TQ^TR \succeq 2\gamma I$. Multiplying by R^TQ^TR from left and its transpose from right gives $SR^TQR + R^TQ^TRS \succeq 2\gamma I$, which implies that $Q^TA \in R\mathcal{C}_\gamma$.

Meta-problem equivalency. Following, we prove that the meta-problem (2) is equivalent to formulation (10), expressed in terms of $\mathcal{I}_\gamma$,$\mathcal{I}^\Gamma$:

Suppose A^* is optimal in (2) with $a^* = \mathrm{f}(A^*, \sigma_{\min}(A^*), \sigma_{\max}(A^*))$. Let $\gamma = \sigma_{\min}(A^*)$ and $\Gamma = \sigma_{\max}(A^*)$. Clearly (A^*, γ, Γ) is feasible in (10) with the same functional value.

Now, let $(B^*, \gamma^*, \Gamma^*)$ be optimal in (10) with $b^* = \mathrm{f}(B^*, \gamma^*, \Gamma^*)$. This implies that $\sigma_{\min}(B^*) \geq \gamma^*$, $\det(B^*) \geq 0$ and $\sigma_{\max}(B^*) \leq \Gamma^*$. This, along with the monotonicity conditions, implies that B^* is feasible in (2). Moreover, $\mathrm{f}(B^*, \sigma_{\min}(B^*), \sigma_{\max}(B^*)) \leq \mathrm{f}(B^*, \gamma^*, \Gamma^*) = b^*$.

In order to conclude the proof, we need to show that B^* is in fact optimal in (2). Assume, towards contradiction, that B' is feasible in (2) with $\mathrm{f}(B') < b^*$. By the first part of the proof, $(B', \sigma_{\min}(B'), \sigma_{\max}(B'))$ is optimal in (10) with $\mathrm{f}(B', \sigma_{\min}(B'), \sigma_{\max}(B')) < b^*$, in contradiction to the optimality of $(B^*, \gamma^*, \Gamma^*)$.