

Provably Good Planar Mappings

Roi Poranne Yaron Lipman
Weizmann Institute of Science

Abstract

The problem of planar mapping and deformation is central in computer graphics. This paper presents a framework for adapting general, smooth, function bases for building provably good planar mappings. The term “good” in this context means the map has no fold-overs (injective), is smooth, and has low isometric or conformal distortion.

Existing methods that use mesh-based schemes are able to achieve injectivity and/or control distortion, but fail to create smooth mappings, unless they use a prohibitively large number of elements, which slows them down. Meshless methods are usually smooth by construction, yet they are not able to avoid fold-overs and/or control distortion.

Our approach constrains the linear deformation spaces induced by popular smooth basis functions, such as B-Splines, Gaussian and Thin-Plate Splines, at a set of collocation points, using specially tailored convex constraints that prevent fold-overs and high distortion at these points. Our analysis then provides the required density of collocation points and/or constraint type, which guarantees that the map is injective and meets the distortion constraints over the *entire* domain of interest.

We demonstrate that our method is interactive at reasonably complicated settings and compares favorably to other state-of-the-art mesh and meshless planar deformation methods.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: meshless deformation, bijective mappings, bounded isometric distortion, conformal distortion

Links: [DL](#) [PDF](#)

1 Introduction

Space deformation is an important tool in graphics and image processing, with applications ranging from image warping and character animation, to non-rigid registration and shape analysis. The two-dimensional case has garnered a great deal of attention in recent years, as is evident from the abundance of literature on the subject. Virtually all methods attempt to find maps that possess three key properties: smoothness, injectivity and shape preservation. Furthermore, for the purpose of warping and posing characters, the method should have interactive performance. However,

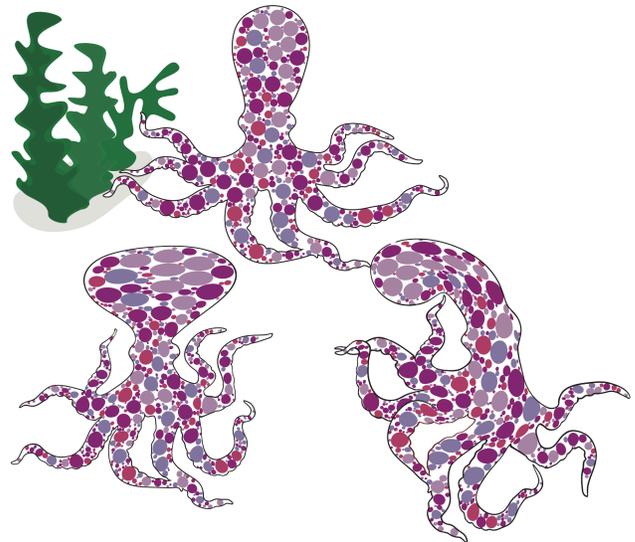


Figure 1: Our method is capable of generating smooth bijective maps with controlled distortion at interactive rates. Top row: source image. bottom row: examples of deformations.

there is no known method that possesses all of these properties. In this paper, we provide the theory and tools to generate maps that achieve all of these properties, including *interactivity*.

Previous deformation models can be roughly divided into mesh-based and meshless models. Mesh-based maps are predominantly constructed using linear finite elements, and are inherently not smooth, but can be made to look smooth by using highly dense elements. Although the methods for creating maps with controlled distortion exist, they are time-consuming, and dense meshes prohibit their use in an interactive manner. On the other hand, meshless maps are usually defined using smooth bases and hence are smooth themselves. Yet we are unaware of any known technique that ensures their injectivity and/or bounds on their distortion.

The goal of this work is to bridge this gap between mesh and meshless methods, by providing a generic framework for making any smooth function basis suitable for deformation. This is accomplished by enabling direct control over the distortion of the Jacobian during optimization, including preservation of orientation (to avoid flips). Our method generates maps by constraining the Jacobian on a dense set of “collocation” points, using an active-set approach. We show that only a sparse subset of the collocation points needs to be active at every given moment, resulting in fast performance, while retaining the distortion and injectivity guarantees. Furthermore, we derive a precise mathematical relationship between the density of the collocation points, the maximal distortion achieved on them, and the maximal distortion achieved everywhere in the domain of interest.



Figure 2: Several examples created with our method. The source of each group is on the left. Note the smooth deformations and the controlled distortion as can be visually assessed from the spheres texture.

2 Previous Work

There is a vast amount of previous work on planar warping and deformation, and it is impossible to provide a comprehensive list here. We therefore focus only on the previous work that is closely related to this paper.

2.1 Mesh-based deformations

The simplest form of mesh-based deformation is done by linearly interpolating the positions of the mesh vertices, which can cause arbitrary distortions and flips. Alexa et al. [2000] suggested controlling element distortion by individually interpolating triangles in an “as-rigid-as-possible” manner, using their polar decomposition. A generalization of this approach is made by considering the mesh in special “coordinates”, which capture the local shape of the mesh with some invariance property [Sheffer and Kraevoy 2004; Sorkine et al. 2004; Xu et al. 2005]. Other methods describe mesh deformations by discretizing a relevant variational problem directly over the mesh, using a finite-element or finite-difference perspective [Terzopoulos et al. 1987; Xu et al. 2005; Igarashi et al. 2005; Sorkine and Alexa 2007; Liu et al. 2008; Chao et al. 2010]. Recently, several mesh-based methods that explicitly control injectivity and distortion were introduced [Lipman 2012; Schüller et al. 2013; Chen et al. 2013]. However, these methods suffer from poor performance when the density of the mesh is high.

2.2 Meshless deformations

While the optimization front in mesh based methods has been developed extensively, most meshless based approaches still use only simple linear blending of basis functions to generate deformations. Instead, the effort in this field of research was put into defining better basis functions. Free-Form Deformation [Sederberg and Parry 1986] and Thin-Plate Spline (TPS) deformation [Bookstein 1989] were some of the earlier examples. Later, generalized barycentric coordinates (BC) were designed to be shape aware: for example, Mean Value Coordinates (MVC) [Floater 2003; Ju et al. 2005]. Different aspects of BC were subsequently improved [Joshi et al. 2007; Lipman et al. 2008; Weber et al. 2012; Li et al. 2013].

Related to our work, several researchers have developed methods that attain injectivity under some conditions. Floater and Kosinka [2010] proved that, for mappings between convex domains, the MVC achieves injectivity. Similarly, Kosinka and Barton [2010] provide conditions for injective maps constructed on cube-like domains by using Warren’s BC ([Warren 1996]). Recently, Schneider et al. [2013], constructed an algorithm that, by composing a series of MVC mappings, produces a bijection, up to pixel precision. Although their method is attractive, as it only uses the MVC basis, it produces mappings that are proven to be truly injective only on a finite set of points. In contrast, our method is more general and works with different basis functions and distortion measures, and the generated maps can be shown to satisfy the injectivity and distortion bounds over the entire domain.

Several methods suggest using the meshless basis functions in a framework similar to the mesh-based deformation framework, instead of directly controlling the coefficients of the basis function. This is done by sampling points inside the domain to discretize and minimize an energy. Adams et al. [2008] used *shape functions* as defined in [Fries T.-P. 2003] to find an interpolation between two shapes. Similarly, Ben-Chen et al. [2009] and Weber et al. [2012] used harmonic and biharmonic coordinates, respectively, but minimized an energy only on sampled points from the skeleton or boundary of the shape. Levi et al. [2013] used the so-called Interior RBF, but instead of sampling the domain, packed it with spheres. Their energy strives to retain the shape of the spheres as much as possible.

Our approach is similar to the ones mentioned above, in that we also use basis functions and optimize over a set of points. However, the previous methods have a considerable disadvantage: they cannot guarantee a bijection or limit the distortion. We overcome this limitation by first formulating an optimization problem that also bounds the distortion on a point set, based upon [Lipman 2012]. Since this does not guarantee that the distortion outside the point set is bounded, we develop sufficient conditions for the deformation to satisfy distortion bounds on the *entire domain*. Using these techniques, we are able to generate smooth maps that are not only injective, but are also sure to have bounded distortion.

Lastly, we mention that in the subdivision literature, injectivity of the characteristic map needs to be verified for proving C^1 continuity of the limit surface. Several related methods for analysing the Jacobian were suggested [Reif 1995; Peters and Reif 1998; Zorin 1998].

3 Method outline

Problem statement. We discuss the application of “handle”-based deformation. This scenario involves a user who wishes to smoothly deform a region-of-interest $\Omega \subset \mathbb{R}^2$ in the plane, e.g. part of an image or a 2D character, under an allowable amount of distortion and without fold-overs. The user drives the deformation by positioning handles inside the domain, and manipulating them in order to define positional constraints. Our algorithm will supply the map that conforms best to the handles, while not violating the distortion constraints at any point $\mathbf{x} = (x, y)$ inside Ω . The ideal deformation $\mathbf{f} : \Omega \rightarrow \mathbb{R}^2$ can be found as the solution to the general problem:

$$\min_{\mathbf{f}} E_{\text{pos}}(\mathbf{f}) + \lambda E_{\text{reg}}(\mathbf{f}) \quad (1)$$

$$\text{s.t. } D(\mathbf{f}; \mathbf{x}) \leq K_{\text{max}}, \forall \mathbf{x} \in \Omega \quad (2)$$

where E_{pos} is the positional constraints energy, E_{reg} is a regularization term, which controls the smoothness of the deformation, and $D(\mathbf{f}; \mathbf{x})$ is a measure of the distortion of \mathbf{f} at point \mathbf{x} (to be defined). Being infinite-dimensional with infinite number of constraints, the problem in (1)-(2) is intractable, so a simplification is required.

Given a finite collection of basis functions $\mathcal{F} = \{f_i\}_{i=1}^n$, where $f_i : \Omega \rightarrow \mathbb{R}$, one can construct planar maps by linear combinations of the basis functions,

$$\mathbf{f}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^T = \sum_{i=1}^n \mathbf{c}_i f_i(\mathbf{x}), \quad (3)$$

where $\mathbf{c}_i = (c_i^1, c_i^2)^T \in \mathbb{R}^{2 \times 1}$ are column vectors. Such a map can be represented by a matrix $\mathbf{c} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n] \in \mathbb{R}^{2 \times n}$ containing the coefficients from eq. (3) as columns.

The bases mentioned above (and others) work very well for *interpolating* and *approximating* scalar functions in the plane, due to their regularity, approximation power and simplicity. Yet using this model as-is for building planar maps can, and often does, introduce arbitrary distortions and uncontrolled fold-overs, which renders this framework suboptimal for space warping and deformation. Nevertheless, we show how to constrain \mathbf{c} in the space $\mathbb{R}^{2 \times n}$ to provide a mechanism for constructing planar deformations with controllable distortion and without fold-overs.

Basis functions. Although the framework in eq. (3) is general, and can be used in theory with any basis function of choice, we chose to experiment with three popular function bases: B-Splines, Thin-Plate Splines (TPS), and Gaussians (see Table 1 on the following page). Nevertheless, the tools developed in this paper are general, and can be used to construct injective and distortion-controlled mappings from different bases as well.

Distortion. The distortion of a differentiable map \mathbf{f} at a point \mathbf{x} is defined to be some measure of how \mathbf{f} changes the metric at the vicinity of \mathbf{x} . Most distortion measures can be formulated using the Jacobian matrix,

$$J\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \partial_x u(\mathbf{x}) & \partial_y u(\mathbf{x}) \\ \partial_x v(\mathbf{x}) & \partial_y v(\mathbf{x}) \end{pmatrix},$$

of \mathbf{f} at point \mathbf{x} , and more specifically, its maximal and minimal singular values, which we denote by $\Sigma(\mathbf{f}; \mathbf{x})$ and $\sigma(\mathbf{f}; \mathbf{x})$, or simply $\Sigma(\mathbf{x})$ and $\sigma(\mathbf{x})$ when there is no risk of confusion. These values measure the extent to which the map stretches a local neighborhood near \mathbf{x} .

We denote the distortion measure of \mathbf{f} at \mathbf{x} by $D(\mathbf{f}, \mathbf{x}) = D(\mathbf{f}) = D(\Sigma(\mathbf{x}), \sigma(\mathbf{x}))$, where the greater D is, the greater the distortion. When $D(\mathbf{x}) = 1$ there is no distortion at all at \mathbf{x} . We make use of the two common measures of distortion: *isometric* and *conformal*. Isometric distortion measures the preservation of lengths and can be computed with $D_{\text{iso}}(\mathbf{x}) = \max\{\Sigma(\mathbf{x}), 1/\sigma(\mathbf{x})\}$. When $\Sigma(\mathbf{x}) = \sigma(\mathbf{x}) = 1$, and only then, $D_{\text{iso}}(\mathbf{x}) = 1$, which implies that \mathbf{f} is close to a rigid motion in the vicinity of \mathbf{x} . Conformal distortion, on the other hand, measures the change in angles that is introduced by the map \mathbf{f} and can be calculated with $D_{\text{conf}}(\mathbf{x}) = \Sigma(\mathbf{x})/\sigma(\mathbf{x})$. When $\Sigma(\mathbf{x}) = \sigma(\mathbf{x})$, $D_{\text{conf}}(\mathbf{x})$ reaches its lowest possible value of 1. This indicates that, locally, the map behaves like a similarity transformation (rigid motion with an isotropic scale).

Fold-overs. A continuously differential map \mathbf{f} is *locally injective* at a vicinity of a point \mathbf{x} if $\det J\mathbf{f}(\mathbf{x}) > 0$. To guarantee local injectivity, it suffices to ensure that $\sigma(\mathbf{x}) > 0$ for all $\mathbf{x} \in \Omega$, and $\det J\mathbf{f}(\mathbf{x}) > 0$ for a single point $\mathbf{x} \in \Omega$ (in fact, one point in each connected component of Ω). Indeed, since $\sigma(\mathbf{x}) > 0$, we know that $\det J\mathbf{f}(\mathbf{x}) \neq 0$, and since $\det J\mathbf{f}(\mathbf{x})$ is a continuous function of \mathbf{x} , it cannot change sign in a connected region. *Global injectivity* of a (proper) differential map $\mathbf{f} : \Omega \rightarrow \mathbb{R}^2$ that is locally injective is guaranteed if the domain is simply connected and \mathbf{f} , restricted to the boundary, is injective.

Collocation points and the active set method. Our goal is to control the distortion and local injectivity of the map \mathbf{f} over the domain Ω . To this end, we maintain a set of *collocation points* $\mathcal{Z} = \{\mathbf{z}_j\}_{j=1}^m \subset \Omega$, where we explicitly monitor and control the distortion and injectivity over. That is, we ensure that

$$D(\mathbf{z}_j) \leq K \quad , \quad \sigma(\mathbf{z}_j) > 0 \quad (4)$$

for all $j = 1, \dots, m$, where $K \geq 1$ is a parameter. Given these bounds on the set \mathcal{Z} we provide bounds on the distortion and injectivity of \mathbf{f} at all points $\mathbf{x} \in \Omega$.

To allow interactive rates, we use an *active set* method: The constraints are set only on a sparse subset, the active set, $\mathcal{Z}' \subset \mathcal{Z}$. Once a certain collocation point \mathbf{z} violates the desired bounds in eq. (4), it is added to the active set \mathcal{Z}' . Collocation points at which the distortion goes sufficiently below the desired bound are removed from the active set. See Figure 3 for an illustration. Implementation details are provided in Section 5. A similar idea was used in [Bommes et al. 2013], where the active set was termed *lazy constraints*.

It is possible to constrain the distortion at a collocation point \mathbf{z} by utilizing the simple observation that the Jacobian matrix of \mathbf{f} is linear in the variables \mathbf{c} ,

$$J\mathbf{f}(\mathbf{x}) = (\nabla u(\mathbf{x}), \nabla v(\mathbf{x}))^T = \sum_{i=1}^n \mathbf{c}_i \nabla f_i(\mathbf{x}),$$

and adapting the convexification approach of [Lipman 2012] to the meshless setting. Further details are in Section 5.

We recall that the definition of the *fill distance* $h(\mathcal{Z}, \Omega)$ of the collocation points in the domain is the furthest distance from \mathcal{Z} that can be achieved in Ω , namely,

$$h(\mathcal{Z}, \Omega) = \max_{\mathbf{x} \in \Omega} \min_{\mathbf{z} \in \mathcal{Z}} \|\mathbf{x} - \mathbf{z}\|. \quad (5)$$

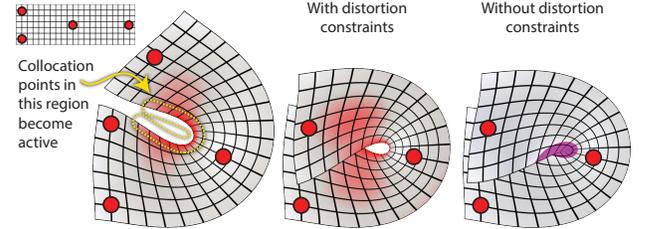


Figure 3: Illustration of our active set approach. As the bar bends, the distortion rises above a certain threshold, causing collocation points in the region to become active (left). These points prevent the bar from collapsing (middle). Excluding these points results in a map with singularities (right).

Modulus of continuity. One of the key aspects of this paper is the ability to ensure that the constructed maps via eq. (3) satisfy strict requirements of distortion and injectivity. This is achieved by estimating the change in the singular values functions $\sigma(\mathbf{x})$, $\Sigma(\mathbf{x})$ of the Jacobian $J\mathbf{f}(\mathbf{x})$ of the map \mathbf{f} . For this, the notion of the *modulus of continuity* becomes handy: It is a tool for measuring the rate of change of a function. Specifically, a function $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ is said to have a modulus of continuity ω , or in short, is ω -continuous, if it satisfies

$$|g(\mathbf{x}) - g(\mathbf{y})| \leq \omega(\|\mathbf{x} - \mathbf{y}\|), \quad \forall \mathbf{x}, \mathbf{y} \in \Omega, \quad (6)$$

where $\|\cdot\|$ denotes the Euclidean norm in \mathbb{R}^2 , and $\omega : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a continuous, strictly monotone function that vanishes at 0. Section 4 explains the computation of the modulus of continuity of the singular values functions $\sigma(\mathbf{x})$, $\Sigma(\mathbf{x})$ and describes how to use it for bounding the distortion of the map \mathbf{f} . We also make use of the modulus of continuity of maps (vector valued functions) $\mathbf{g} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where similarly to the scalar case, \mathbf{g} is ω -continuous if

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \leq \omega(\|\mathbf{x} - \mathbf{y}\|), \quad \forall \mathbf{x}, \mathbf{y} \in \Omega. \quad (7)$$

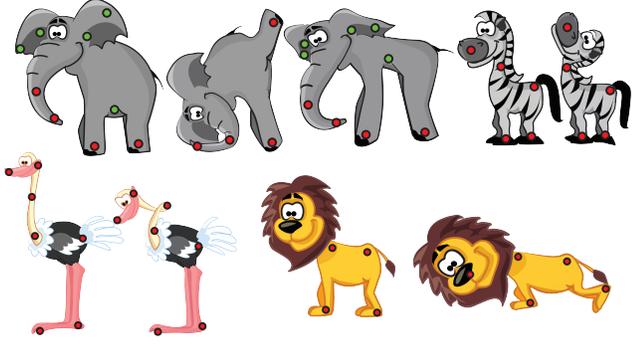


Figure 4: Several more examples created with our method. Due to the smoothness of the basis functions, our method is capable of handling pointwise constraints in a smooth and graceful manner.

4 Bounding the Distortion

The core of our approach lies in bounding the change in the distortion at a point as it gets further away from a collocation point $\mathbf{z} \in \mathcal{Z}$. We observe that, for many useful function bases \mathcal{F} , given the coefficients \mathbf{c} and the domain Ω , one can compute a modulus $\omega = \omega_{\Sigma, \sigma}$ such that the singular value functions $\Sigma(\mathbf{x}), \sigma(\mathbf{x})$ are ω -continuous. This, in turn, allows bounding the change in the singular values.

In this section we: (i) provide the general motivation for calculating the modulus ω of singular values; (ii) compute ω for the collection of basis functions used in this paper; (iii) show how ω can be used to bound the different distortion measures; and (iv) explore the different strategies for controlling the distortion of \mathbf{f} over Ω .

Why ω is useful? For example, to bound $\sigma(\mathbf{x})$ from below at all points $\mathbf{x} \in \Omega$ we assume that we have the bound $\sigma(\mathbf{z}) \geq \delta > 0$ at all collocation points $\mathbf{z} \in \mathcal{Z}$. Then, if $\sigma(\mathbf{x})$ is ω -continuous we have $|\sigma(\mathbf{x}) - \sigma(\mathbf{z})| \leq \omega(\|\mathbf{x} - \mathbf{z}\|)$ and therefore in particular $\sigma(\mathbf{x}) \geq \sigma(\mathbf{z}) - \omega(\|\mathbf{x} - \mathbf{z}\|) \geq \delta - \omega(\|\mathbf{x} - \mathbf{z}\|)$. Similarly, an upper bound to $\Sigma(\mathbf{x})$ can be found. This is described in the following lemma:

Lemma 1. *Let Σ and σ be ω -continuous functions, and let $\mathbf{z} \in \mathcal{Z}$ be some collocation point. Then for all points $\mathbf{x} \in \Omega$,*

$$\sigma(\mathbf{z}) - \omega(\|\mathbf{x} - \mathbf{z}\|) \leq \sigma(\mathbf{x}) \leq \Sigma(\mathbf{x}) \leq \Sigma(\mathbf{z}) + \omega(\|\mathbf{x} - \mathbf{z}\|).$$

Computing ω for different \mathcal{F} . Using Lemma 1 requires knowing the modulus of continuity of the singular value functions $\sigma(\mathbf{x}), \Sigma(\mathbf{x})$ of the map \mathbf{f} built using an arbitrary function basis \mathcal{F} . Although this task might seem daunting, we show that, surprisingly enough, for 2D maps, this problem can be reduced to the easier task of calculating the modulus of continuity of the *Jacobian* of the map \mathbf{f} , or equivalently, the modulus of continuity of the gradients ∇u and ∇v , as the following lemma asserts:

Lemma 2. *Let ∇u and ∇v be ω -continuous in Ω . Then both singular values functions Σ and σ are 2ω -continuous.*

The proof of this lemma is given in Appendix B. This lemma is used to compute a modulus of continuity $\omega = \omega_{\Sigma, \sigma}$ for the singular

values functions of a map \mathbf{f} defined via eq. (3). First, we note that

$$\begin{aligned} \|\nabla u(\mathbf{x}) - \nabla u(\mathbf{y})\| &\leq \sum_{i=1}^n |c_i^1| \|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \\ &\leq \sum_{i=1}^n |c_i^1| \omega_{\nabla f_i}(\|\mathbf{x} - \mathbf{y}\|) \\ &\leq \|\mathbf{c}\| \omega_{\nabla \mathcal{F}}(\|\mathbf{x} - \mathbf{y}\|), \end{aligned} \quad (8)$$

where $\omega_{\nabla f_i}$ is a modulus of continuity for the gradient of the basis function ∇f_i , $\omega_{\nabla \mathcal{F}}$ is a modulus function satisfying $\omega_{\nabla \mathcal{F}}(t) \geq \omega_{\nabla f_i}(t)$, for all $t \in \mathbb{R}^+$ and all $f_i \in \mathcal{F}$, and we use the matrix maximum-norm $\|\mathbf{c}\| = \max_{\ell \in \{1, 2\}} \sum_{i=1}^n |c_i^\ell|$. Equation (8) shows that the modulus of ∇u is $\omega_{\nabla u} = \|\mathbf{c}\| \omega_{\nabla \mathcal{F}}$. Similar arguments show that $\omega_{\nabla v} = \|\mathbf{c}\| \omega_{\nabla \mathcal{F}}$. Finally, Lemma 2 tells us that

$$\omega = 2 \|\mathbf{c}\| \omega_{\nabla \mathcal{F}}. \quad (9)$$

In order to use eq. (9) to bound the change in the singular value functions, the modulus of the gradient $\omega_{\nabla \mathcal{F}}$ for the function basis of interest needs to be known. In Table 1 we summarize the function bases that are used in this paper, as well as the moduli of their gradients, $\omega_{\nabla \mathcal{F}}$. In Appendix A we provide the derivations of these modulus functions. Note that the gradient modulus $\omega_{\nabla \mathcal{F}}$ of the TPS applies only locally to points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ such that $\|\mathbf{x} - \mathbf{y}\| \leq (1.25e)^{-1} \approx 0.29$. However, this is not a significant restriction, as the fill distance is always smaller in practice.

Basis	f_i	$\omega_{\nabla \mathcal{F}}(t)$
B-Splines	$\mathcal{B}_\Delta^{(3)}(x - x_i) \mathcal{B}_\Delta^{(3)}(y - y_i)$	$\frac{4}{3\Delta^2} t$
TPS	$\frac{1}{2}(\ \mathbf{x} - \mathbf{x}_i\ ^2) \ln(\ \mathbf{x} - \mathbf{x}_i\ ^2)$	$t(5.8 + 5 \ln t)$
Gaussians	$\exp\left(-\frac{\ \mathbf{x} - \mathbf{x}_i\ ^2}{2s^2}\right)$	$\frac{t}{s^2}$

Table 1: Function bases and the gradient modulus function.

Bounding isometric and conformal distortion. We show below how Lemma 1 and eq. (9) can be used to provide bounds on the isometric and/or conformal distortion, assuming such bounds are enforced at a set of collocation points \mathcal{Z} .

We start with isometric distortion and assume that at all collocation points $\mathbf{z} \in \mathcal{Z}$ we have $D_{\text{iso}}(\mathbf{z}) \leq K$, or equivalently,

$$\Sigma(\mathbf{z}) \leq K, \quad \sigma(\mathbf{z}) \geq \frac{1}{K}. \quad (10)$$

Denote for brevity $h = h(\mathcal{Z}, \Omega)$, the fill distance of \mathcal{Z} in Ω . Then using Lemma 1 we have for all points $\mathbf{x} \in \Omega$,

$$D_{\text{iso}}(\mathbf{x}) \leq \max \left\{ K + \omega(h), \frac{1}{K^{-1} - \omega(h)} \right\}. \quad (11)$$

This bound holds only when $K^{-1} > \omega(h)$, which implies that $\sigma(\mathbf{x}) > 0$, which in turn guarantees the injectivity of the map. Otherwise, $D_{\text{iso}}(\mathbf{x})$ cannot be bounded.

To bound the conformal distortion, we assume that all the collocation points $\mathbf{z} \in \mathcal{Z}$ satisfy a conformal distortion bound:

$$\Sigma(\mathbf{z}) \leq K\sigma(\mathbf{z}), \quad \sigma(\mathbf{z}) \geq \delta, \quad (12)$$

where the second constraint, with some constant $\delta > 0$, is used to avoid $\sigma(\mathbf{x}) = 0$, which may lead to loss of injectivity. Using Lemma 1 as above, for all $\mathbf{x} \in \Omega$,

$$D_{\text{conf}}(\mathbf{x}) \leq K \left(\frac{\delta + \omega(h)}{\delta - \omega(h)} \right), \quad \sigma(\mathbf{x}) \geq \delta - \omega(h). \quad (13)$$

where, as in the isometric case, $\delta > \omega(h)$ is required to hold.

Controlling the distortion of \mathbf{f} . The bounds in eq. (11) and eqs. (13) relate the distortion of the map \mathbf{f} at all points in the domain Ω to the distortion K enforced on the collocation points \mathcal{Z} and the fill distance of the collocation points $h = h(\mathcal{Z}, \Omega)$. Using these relationships one can control the distortion of the map \mathbf{f} in one of three strategies:

1. Given \mathcal{Z} and the distortion bound K on its points, bound the maximal distortion K_{\max} of \mathbf{f} everywhere else in Ω .
2. Given the distortion bound K enforced at the points \mathcal{Z} and a desired distortion bound $K_{\max} > K$ everywhere in Ω , calculate the required fill distance h to achieve it.
3. Given \mathcal{Z} and a desired distortion bound $K_{\max} > 1$ everywhere in Ω , calculate the distortion bound K that should be enforced on \mathcal{Z} .

Strategy 1 can be accomplished directly from the bounds (11),(13). For strategy 2 we need to rearrange these equations: noting that ω^{-1} also monotonically increases we get

$$h_{\text{iso}} \leq \omega^{-1} \left(\min \left\{ K_{\max} - K, \frac{1}{K} - \frac{1}{K_{\max}} \right\} \right) \quad (14)$$

$$h_{\text{conf}} \leq \omega^{-1} \left(\delta \frac{K_{\max} - K}{K_{\max} + K} \right), \quad (15)$$

where h_{iso} and h_{conf} are the required fill distances to achieve isometric or conformal distortion K_{\max} , respectively. Note that the inequality for the conformal distortion in particular implies that $h_{\text{conf}} < \omega^{-1}(\delta)$ and hence by eq. (13) that $\sigma(\mathbf{x}) > 0$.

For strategy 3 we rearrange the bounds as follows,

$$K_{\text{iso}} \leq \min \left\{ K_{\max} - \omega(h), \frac{1}{\frac{1}{K_{\max}} + \omega(h)} \right\} \quad (16)$$

$$K_{\text{conf}} \leq K_{\max} \frac{\delta - \omega(h)}{\delta + \omega(h)}, \quad \delta_{\text{conf}} > \omega(h) \quad (17)$$

where $K_{\text{iso}}, K_{\text{conf}},$ and δ_{conf} are the distortion bound at the collocation points required to achieve the global isometric and conformal distortion K_{\max} . Note again, that $\sigma(\mathbf{x}) > 0$ is assured due to the inequality for δ_{conf} in eq. (13).

Non-convex domains and interior distances. It is often desirable to consider a non-convex domain Ω , endowed with an interior distance, and basis functions defined using this distance. The definition of the fill-distance and the modulus of continuity are changed accordingly. The analysis above can then be used as-is once the gradient modulus $\omega_{\nabla \mathcal{F}}$ is available, similarly to Table 1. In this paper we only provide the modulus $\omega_{\nabla \mathcal{F}}$ for the Euclidean distance-based basis functions listed in that table, leaving the analysis of other bases to future work.

We emphasize that in case the non-convex domain *is* endowed with the Euclidean distance, the analysis holds as-is for the basis functions from Table 1. This is due to the fact that these basis functions are defined everywhere in \mathbb{R}^2 and the modulus of their gradients is agnostic to the shape of the domain. To generate a set of collocation points with a prescribed Euclidean fill-distance in a non-convex domain it is enough to ask that the domain satisfies the *cone condition* (see e.g., [Wendland 2004], Definition 3.6), and to consider all the points from a surrounding uniform grid that fall inside the domain.

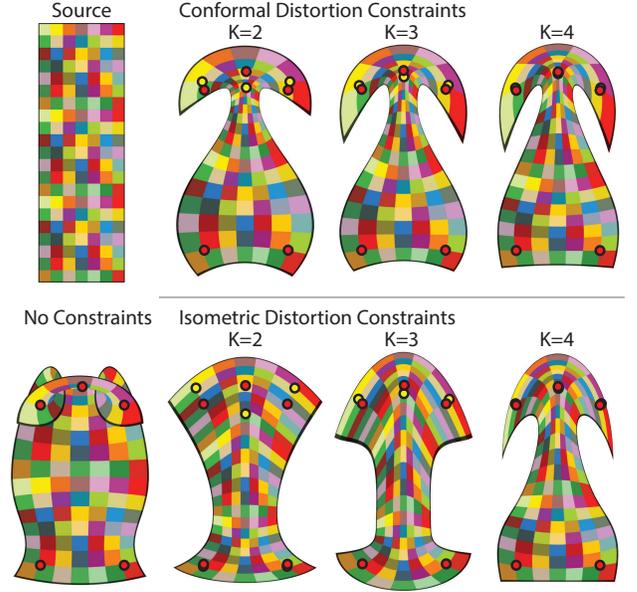


Figure 5: Deformation of a bar using various distortion constraints using B-Splines (see Section 6 for details).

5 Optimization and implementation details

In this section we describe the algorithm for calculating maps of the form of eq. (3), which conform to the positional constraints prescribed by the user, and satisfy distortion and injectivity requirements. This algorithm is summarized in Algorithm 1. The theory in Section 4 suggests replacing the optimization problem in (1)-(2) with the following:

$$\begin{aligned} \min_{\mathbf{c}} \quad & E_{\text{pos}}(\mathbf{f}) + \lambda E_{\text{reg}}(\mathbf{f}) \\ \text{s.t.} \quad & D(\mathbf{f}; \mathbf{z}) \leq K, \quad \forall \mathbf{z} \in \mathcal{Z}, \\ & \mathbf{f} = \sum_{i=1}^n \mathbf{c}_i f_i, \end{aligned} \quad (18)$$

where E_{pos} is the energy of the positional constraints that is changed during user interaction, E_{reg} is a regularization energy, $D = D_{\text{iso}}$ or $D = D_{\text{conf}}$ is the distortion type, and $K \geq 1$ is a user prescribed distortion bound. According to Section 4, for the correct choice of K and \mathcal{Z} , \mathbf{f} is guaranteed to be injective and have distortion smaller than K_{\max} . In the following, Eq. (18) is formulated as a Second-Order Cone Program (SOCP), which can be solved efficiently by an interior point method.

We remark here the positional constraints energy E_{pos} from eq. (18) can be replaced with hard constraints. In this case however, the problem may be infeasible due to the distortion bound, regardless of how the basis functions are chosen. This can occur if, for example, the isometric distortion is required to not exceed a value of K , but two handles are pulled apart by a factor greater than K . In an interactive session, this means that the deformation will *not update* until the handles are put back in acceptable positions, which can become a nuisance to the user.

Activation of constraints. During interaction, eq. (18) is solved constantly as the user manipulates the handles. At each optimization step, only a fraction of the collocation points is active, so removing the rest of the collocation point will not change the result, but will greatly reduce the computation time. In the following, we devise an algorithm that utilizes this fact, where collocation points may be inserted or removed from the active-set before each step.

The algorithm should make the interaction as smooth as possible; the distortion at any *deactivated* collocation point should not suddenly become significantly greater than K at any given step. Otherwise, at the next step, the point will become active, which will cause the deformation to "jump". Therefore, we opt to insert points into the active-set when the distortion on them is slightly below K . We assume that the collocation points are sampled on a dense rectangular grid. Before each optimization step, the distortion on each collocation point is measured, and the local maxima of the distortion are found. If a local maximum has a distortion greater than K_{high} for a specified $K_{\text{high}} \in [1, K]$, then that point is added to the active-set for the next optimization step. If any collocation point has distortion lower than K_{low} where $K_{\text{low}} \in [1, K_{\text{high}}]$, then that point is removed from the active-set. This ensures that the collocation points with the maximal distortion are always active, and hence all other collocation points must have distortion smaller than K . To further stabilize the process against fast movement of the handles by the user, we may keep a small subset of equally spread collocation points always active. In our implementation we used the default values $K_{\text{high}} = 0.1 + 0.9K$ and $K_{\text{low}} = 0.5 + 0.5K$. See Figure 6 for examples.

During an interactive session, potentially all of the collocation points can become active at once. However, this does not occur in practice, since only points that are above a threshold *and* are local maxima of the distortion can be activated. Thus, only a small number of isolated points will be activated at each iteration. The only scenario in which all collocation points are activated simultaneously is when the distortion is constant everywhere when it crosses the distortion bound threshold. This scenario is extremely unlikely due to nature of the deformation energy and the bases functions used.



Figure 6: Active-set visualization. The yellow dots represent the positions of the activated collocation points for the deformation shown. Note that some of the points remain activated throughout to stabilize the process.

Distortion and injectivity constraints. We explicitly constrain the points in the active-set according to eq. (10) or (12). This requires constraining the singular values of the Jacobian $J\mathbf{f}(\mathbf{z})$ for all $\mathbf{z} \in \mathcal{Z}$. We provide a new formulation to the convex second-order cone constraints described in [Lipman 2012], where the singular values of the Jacobian of the map \mathbf{f} are expressed in terms of the gradients of \mathbf{f} (i.e., ∇u and ∇v), which is compact and useful for proving Lemma 2 (see Appendix B).

We define two vectors, $J_S\mathbf{f}(\mathbf{x})$ and $J_A\mathbf{f}(\mathbf{x})$, corresponding to the similarity and anti-similarity parts of $J\mathbf{f}(\mathbf{x})$, as follows

$$\begin{aligned} J_S\mathbf{f}(\mathbf{x}) &= \frac{\nabla u(\mathbf{x}) + \mathcal{I}\nabla v(\mathbf{x})}{2} \\ J_A\mathbf{f}(\mathbf{x}) &= \frac{\nabla u(\mathbf{x}) - \mathcal{I}\nabla v(\mathbf{x})}{2} \end{aligned} \quad (19)$$

Here \mathcal{I} is the counter-clockwise rotation 2×2 matrix by $\pi/2$. It can be shown (see e.g. [Lehto and Virtanen 1973], ch. 1.9, p. 49) that the singular values of $J\mathbf{f}(\mathbf{x})$ can then be expressed as

$$\begin{aligned} \Sigma(\mathbf{x}) &= \|J_S\mathbf{f}(\mathbf{x})\| + \|J_A\mathbf{f}(\mathbf{x})\| \\ \sigma(\mathbf{x}) &= \left| \|J_S\mathbf{f}(\mathbf{x})\| - \|J_A\mathbf{f}(\mathbf{x})\| \right|. \end{aligned} \quad (20)$$

The requirement (10) for the isometric distortion can be written in terms of $J_S\mathbf{f}$ and $J_A\mathbf{f}$, which are linear in \mathbf{c} . Eq. (10) then becomes

$$\|J_S\mathbf{f}(\mathbf{x}_i)\| + \|J_A\mathbf{f}(\mathbf{x}_i)\| \leq K \quad (21)$$

$$\|J_S\mathbf{f}(\mathbf{x}_i)\| - \|J_A\mathbf{f}(\mathbf{x}_i)\| \geq \frac{1}{K}, \quad (22)$$

where eq. (21) can be transformed into convex cone constraints,

$$\|J_S\mathbf{f}(\mathbf{x}_i)\| \leq t_i \quad (23a)$$

$$\|J_A\mathbf{f}(\mathbf{x}_i)\| \leq s_i \quad (23b)$$

$$t_i + s_i \leq K, \quad (23c)$$

where t_i, s_i are auxiliary variables. However, trying to apply a similar transformation to eq. (22) will result in the non-convex cone-complement constraint,

$$\|J_S\mathbf{f}(\mathbf{x}_i)\| \geq r_i, \quad (24)$$

for an auxiliary r_i . Following Lipman's [2012] approach, eq. (24) can be convexified by introducing the notion of *frames*. A frame is a unit vector \mathbf{d}_i used to replace eq. (24) by

$$J_S\mathbf{f}(\mathbf{x}_i) \cdot \mathbf{d}_i \geq r_i. \quad (25)$$

Eq. (25) is a half plane that is contained in the cone-complement of eq. (24) (see inset). Using (25), we can replace (22) with

$$J_S\mathbf{f}(\mathbf{x}_i) \cdot \mathbf{d}_i - s_i \geq \frac{1}{K}, \quad (26)$$

noting that r_i is actually redundant. We also note that this constraint forces the determinant to be positive.

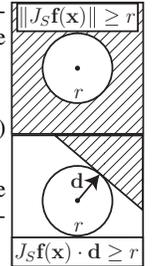
The optimal choice of \mathbf{d}_i at a certain optimization step depends on the value of $J_S\mathbf{f}(\mathbf{x}_i)$ at the previous step. We would like the boundary of the half plane defined by \mathbf{d}_i to be as far away as possible from $J_S\mathbf{f}(\mathbf{x}_i)$ of the previous step to allow maximum maneuverability for the next step. This is achieved by setting

$$\mathbf{d}_i = J_S\mathbf{f}(\mathbf{x}_i) / \|J_S\mathbf{f}(\mathbf{x}_i)\| \quad (27)$$

after each step. For the conformal distortion case we write the constraints as in [Lipman 2012] in our notation:

$$\|J_A\mathbf{f}(\mathbf{x}_i)\| \leq \frac{K-1}{K+1} J_S\mathbf{f}(\mathbf{x}_i) \cdot \mathbf{d}_i \quad (28a)$$

$$\|J_A\mathbf{f}(\mathbf{x}_i)\| \leq J_S\mathbf{f}(\mathbf{x}_i) \cdot \mathbf{d}_i - \delta. \quad (28b)$$



Initialization of the frames. In [Lipman 2012], the frames had to be picked correctly to guarantee feasibility. However, here, matters are simpler. Firstly, by using soft positional constraints we ensure that a solution always exists. Although the choice of frames may not be optimal in the first iteration, it will improve in subsequent steps. Secondly, the interaction usually starts from a *rest pose*, so the trivial solution has the identity as the Jacobian for each collocation point, and hence satisfies any distortion bound. To include the trivial solution in the feasible set we set the frames to be $\mathbf{d}_i = (1, 0)$ for all \mathbf{d}_i .

Deformation energies and positional constraints. The energy $E_{\text{pos}}(\mathbf{f})$ for the positional constraints in eq. (18) is defined by

$$E_{\text{pos}}(\mathbf{f}) = \sum_l \|\mathbf{f}(\mathbf{p}_l) - \mathbf{q}_l\| = \sum_l \left\| \sum_{i=1}^n \mathbf{c}_i f_i(\mathbf{p}_l) - \mathbf{q}_l \right\| \quad (29)$$

where $\{\mathbf{p}_l\}_{l=1}^{n_l}$ and $\{\mathbf{q}_l\}_{l=1}^{n_l}$ are the source and target positions of the handles. We choose to this energy instead of the more common quadratic energy since it is more natural in the SOCP setting, although a quadratic energy can be used as well (by adding another cone constraint). Minimizing this energy is equivalent to minimizing,

$$\begin{aligned} \min \quad & \sum_l r_l \\ \text{s.t.} \quad & \left\| \sum_{i=1}^n \mathbf{c}_i f_i(\mathbf{p}_l) - \mathbf{q}_l \right\| \leq r_l, \forall l \end{aligned} \quad (30)$$

where r_l are auxiliary variables. Eq. (30) is an SOCP, which can be combined with the distortion constraints of the previous paragraph.

As for the regularization energy E_{reg} , we use a combination of two common functional: the biharmonic energy, E_{bh} and the ARAP energy, E_{arap} . The biharmonic energy is defined by

$$E_{\text{bh}}(\mathbf{f}) = E_{\text{bh}}(u, v) = \iint_{\Omega} \|Hu(\mathbf{x})\|_F^2 + \|Hv(\mathbf{x})\|_F^2 dA, \quad (31)$$

where Hu and Hv are the Hessians of u and v , respectively, which is a quadratic form in \mathbf{c} . Once \mathbf{c} is taken out of the integral, the integration can be done by numerical quadrature. The ARAP energy is defined by the standard sum,

$$E_{\text{arap}}(\mathbf{f}) = \sum_{s=1}^{n_s} \|\mathbf{J}\mathbf{f}(\mathbf{r}_s) - \mathbf{Q}(\mathbf{r}_s)\|_F^2, \quad (32)$$

where $\{\mathbf{r}_s\}_{s=1}^{n_s}$ are a set of equally spread pre-defined points, and $\mathbf{Q}(\mathbf{r}_s)$ is the closest rotation matrix to $\mathbf{J}\mathbf{f}(\mathbf{r}_s)$. Due to the non-convexity of eq. (32), incorporating it in an optimization problem usually requires a local-global approach in order to solve it (see [Liu et al. 2008]). Using the frames, it can be seen that eq. (32) can be solved via the quadratic functional,

$$E_{\text{arap}}(\mathbf{f}) = \sum_{s=1}^{n_s} (\|J_A \mathbf{f}(\mathbf{x})\|_F^2 + \|J_S \mathbf{f}(\mathbf{x}) - \mathbf{d}_s\|_F^2) \quad (33)$$

where \mathbf{d}_s is the frame at \mathbf{r}_s .

Algorithm 1: Provably good planar mapping

Input:

- Set of positional constraints $\{\mathbf{p}_l\}_{l=1}^{n_l}$ and $\{\mathbf{q}_l\}_{l=1}^{n_l}$
- Set of basis functions $f_i \in \mathcal{F}$
- Grid of collocation points $\mathcal{Z} = \{\mathbf{z}_j\}_{j=1}^m$
- Distortion type and bound on collocation points $K \geq 1$

Output:

- Deformation \mathbf{f}

Initialization:

if first step **then**

- Precompute $f_i(\mathbf{z})$ and $\nabla f_i(\mathbf{z})$ for all $\mathbf{z} \in \mathcal{Z}$.
- Set $\mathbf{d}_i = (1, 0)$ for all \mathbf{d}_i .
- Initialize empty active set \mathcal{Z}' .
- Initialize set \mathcal{Z}'' with farthest point samples.

Evaluate $D(\mathbf{z})$ for $\mathbf{z} \in \mathcal{Z}$.

Find the set \mathcal{Z}_{max} of local maxima of $D(\mathbf{z})$

foreach $\mathbf{z} \in \mathcal{Z}_{\text{max}}$ such that $D(\mathbf{z}) > K_{\text{high}}$ **do**

└ insert \mathbf{z} to \mathcal{Z}'

foreach $\mathbf{z} \in \mathcal{A}$ such that $D(\mathbf{z}) < K_{\text{low}}$ **do**

└ Remove \mathbf{z} from \mathcal{Z}'

Optimization:

Solve the problem in (18) using the SOCP formulation to find \mathbf{c} .

Use the constraints from eq. (23) and eq. (26) for the isometric case, and those of eq. (28) in the conformal case on the collocation points in $\mathcal{Z}' \cup \mathcal{Z}''$. Use energies from eq. (30), (31), and/or (33).

Postprocessing:

Compute \mathbf{f} using \mathbf{c} and \mathcal{F} .

Update \mathbf{d}_i using eq. (27).

Return:

- Deformation \mathbf{f}
-

6 Results

Software implementation and timing. We have implemented an interactive software using Algorithm 1. We used Mosek [Andersen and Andersen 1999] for solving the optimization problem, and Matlab for updating the active-set. In addition, we used an external OpenGL application to interact and show the deformation. We used a machine with Intel i7 CPU clocked at 3.5 GHz. The included video shows an interactive session using our software. Figure 7 shows timings for solving the optimization problem using Mosek as a function of the number of basis functions and the size of the active-set. For this range, which covers the results in this paper, the time complexity exhibits linear behaviour. In all of our experiments we used a 200^2 grid of collocation points during interaction, and after being satisfied with the results switched to higher grid resolutions using Strategy 2 to guarantee the bounds on the distortion.

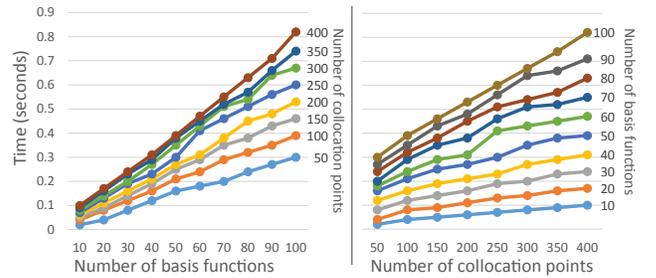


Figure 7: Graphs showing the time required for the optimization as a function of the number of active collocation points and the number of basis function. Note the linear behaviour.

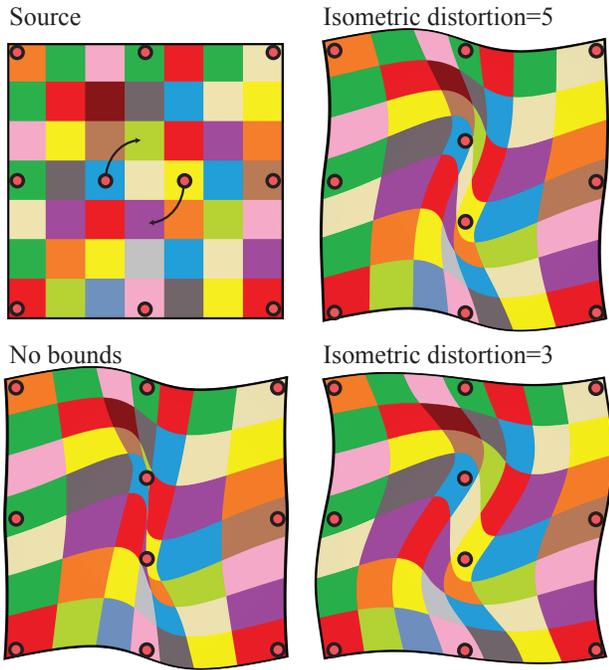


Figure 8: Deformation of a square using TPS. Note the fold-over that caused the subsquare in the middle to disappear in the unconstrained case, and compare to the bounded isometric distortion results on the right.

Parameters and function bases. Our approach is quite versatile as the different function bases, and the distortion type and bound already attain a large variety of different results. We present a set of examples that we believe should advocate the use of our approach.

In Figure 5 we show an example of a deformation of a bar using a 6 by 6 tensor product of uniform cubic B-Splines using the energy $E_{\text{reg}} = E_{\text{pos}} + 10^{-2}E_{\text{arap}}$. Note that for the lower values of K , the positional constraints cannot be satisfied. Also note that with no distortion constraints, the deformation creates two singularities, which were unintended and undesired. Using strategy 3 we found that in order to achieve injectivity for all cases, it was enough to check the distortion on a grid of size 3000^2 . For this grid we found that for $K = 2, 3, 4$, the maximal distortion was guaranteed to be smaller than 3.2, 10 and 49 in the isometric case respectively, and 14, 35 and 33 in the conformal case.

In Figure 8 we show additional examples of deformations of a square to demonstrate the effectiveness of our method for warping. Using TPS this time, with 25 bases positioned on a grid, we rotated two points in the middle while keeping some points on the boundary fixed. We used the smoothness energy $E_{\text{pos}} + 10^{-1}E_{\text{bh}}$. In this case, the unconstrained map resulted in a fold-over that made the sub-square in the middle completely disappear, while the constrained maps stayed bijective. The required grid size that provides the injectivity certification for $K = 5$ in this example was slightly less than 6000^2 . For this resolution, the computation shows that for $K = 3$, the maximal distortion everywhere is smaller than 7.

Mesh-based vs. Meshless. We compared our results with the results of previous similar mesh-based methods. In Figure 9 we show a bird image deformed with a variant of the ARAP method of Igarashi et al. [2005] as implemented in Adobe Photoshop. We compare this result to the meshless approach using ARAP energy

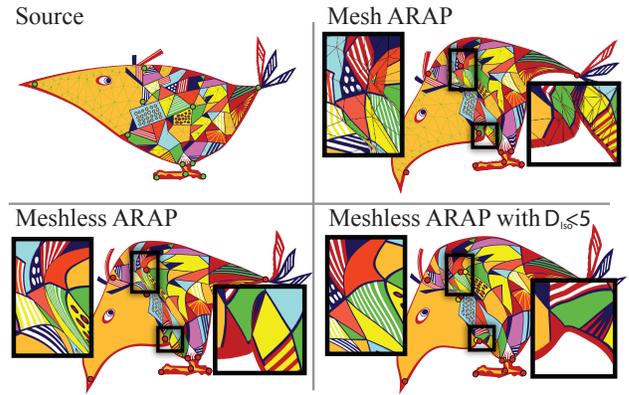


Figure 9: Deformation of a bird drawing. The unconstrained mesh deformation resulted in an unpleasant cusp, while the yellow triangle in the unconstrained meshless deformation (in the blow up) almost vanished. The constrained meshless deformation avoided these problems.

with and without the distortion constraints. One of the main difficulties with mesh-based ARAP, which can be seen in Figure 9, is that when the object is forced to undergo a deformation that is not close to being locally rigid, cusps with fold-overs appear near the handles. This cannot happen when the basis functions are smooth, but even then the ARAP functional creates fold-overs and unbounded distortion. This is rectified by incorporating the distortion constraints. In this example, the required grid size to ensure injectivity was also 6000^2 .

In Figure 10 we deform a disk (see inset) using the method of [Schüller et al. 2013] and [Lipman 2012] and compare it to ours using $D_{\text{iso}} = 5$. Both mesh-based methods guarantee injectivity, as our method does, but it is clearly seen they lack smoothness, in contrast to our approach. In this example, a grid of 2000^2 collocation points proves the map is injective. By evaluating the distortion on 4000^2 points we show that the maximal isometric distortion is smaller than 10.

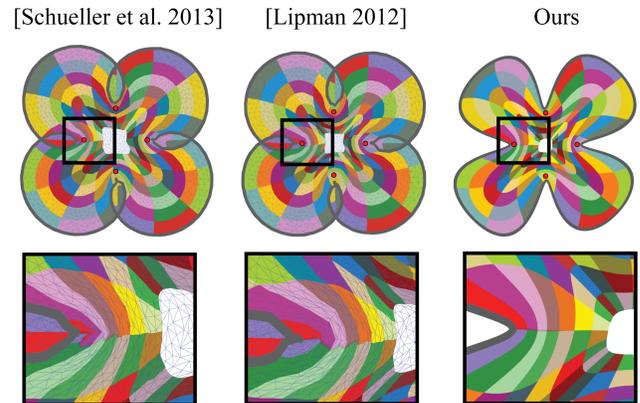


Figure 10: Deformation of a disk. Note the lack of smoothness in both mesh-based methods.

Shape aware bases. The previous results show deformations using the Euclidean distance-based function bases provided in Table 1. For non-convex domains endowed with an internal distance, it is better to use bases that are *shape aware*, namely, their influence obeys internal distances. Many possibilities exist in the literature, e.g., the shape function used in [Adams et al. 2008], or any smooth set of generalized barycentric coordinates. In our experiments we tested shape aware variation of Gaussians, which is achieved by simply replacing the norm in their definition with the shortest distance function. Figure 11 shows such a deformation and compare it again to [Schüller et al. 2013]. Although their method does not allow fold-overs to occur, cusps can still be seen where the handles are. Figures 1,2,4 also demonstrate deformations with this basis function. To provide a proof of injectivity and/or bounded distortion for these examples the modulus of the gradients of the Gaussian shape-aware functions, $\omega_{\nabla\mathcal{F}}$, should be calculated. Although straightforward, it is cumbersome to compute it in general, and we defer it to future work. Lastly, we note that the deformations are as smooth as the basis functions. Using exact shortest distances (which is done here for simplicity) in a non-convex polygonal domain will have discontinuous derivatives at certain points in the domain, but nevertheless produce visually pleasing results.

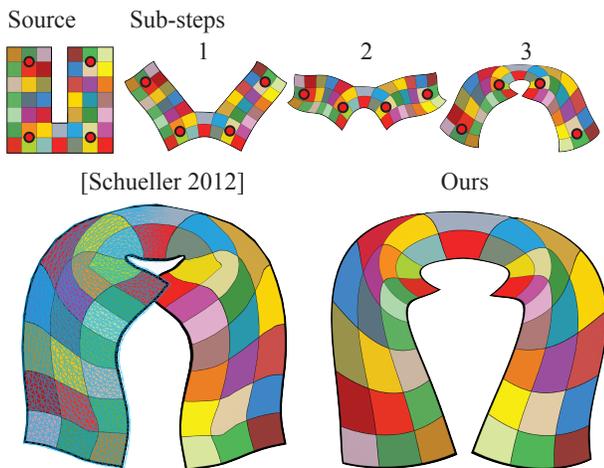


Figure 11: Deformation using 40 shape aware Gaussians and isometric distortion constraint of $K_{iso} \leq 3$. Top row shows intermediate positions of the handles and the respective deformations. Bottom row compares the final deformation with [Schüller et al. 2013] (using the same handle positions, not shown). Note the cusps that occur at the handles when using Schüller’s method.

7 Discussion

This paper presents a framework for making general smooth basis functions suitable for planar deformations. The framework is demonstrated with three popular function bases and the algorithm is shown to allow interactive deformations. The paper provides theory that allows establishing guarantees of injectivity and bounds of isometric and/or conformal distortion.

Our theory and bounds rely on the simple expressions of the singular values of the Jacobian. These expressions are true only in the case of two dimensional domains, and therefore our method is not trivially extended to three dimensions. However, this is the only missing requirement for the transition into higher dimensions, since other key ingredients, such as the use of collocation points and active set remains the same. If this gap can be bridged, we believe that smooth maps with controlled distortion can also be generated in 3D using our approach.

One limitation of the convexification approach we used occurs when enforcing hard positional constraints: if the problem is reported as infeasible in this case, one cannot tell whether this is due to the non-convex problem being infeasible or to the frames not being chosen correctly. This limitation is alleviated either by using soft positional constraints as explained in the paper, or by looking for appropriate frames using some other method (e.g., taking the global unconstrained minimum of the functional and extracting frames from it). In any case the question of feasibility when hard constraints are used is still an open problem.

While the map computation is done interactively, proving its injectivity and bounding its distortion may require a more time, especially if the bounds are strict and/or the deformation is strong. This is due to the very dense grids required by the theoretical bounds. Although the task is simple - all that is required is evaluate the distortion on every point of the grid - our current *serial* implementation allows supporting this computation interactively only on medium sized grids (40k points). However, we believe that using GPU to evaluate the distortion on the collocation points in parallel may enable interactive rates for large grids.

Our results show that, by using a generic SOCP solver (Mosek), our approach can handle a few hundreds of active collocation points and basis function at interactive rates. However, there may be situations where thousands or more are required. Developing a specialized solver for this task can allow such large problems to be solved quickly.

One long standing problem, is how to find bijections between fixed domains. One approach is to use barycentric coordinates, e.g. [Floater and Kosinka 2010; Schneider et al. 2013]. We wish to explore the possibility of using our method with certain barycentric coordinates as basis function to find maps between domains.

Acknowledgements

We would like to thank Christian Schüller and Olga Sorkine-Hornung for providing the code and support for “Locally Injective Mappings” and Nave Zelzer for the external OpenGL viewer. We would also like to thank Eitan Grinspun for his valuable feedback. This work was funded by the European Research Council (ERC Starting Grant “SurfComp”), the Israel Science Foundation (grant No. 1284/12) and the I-CORE program of the Israel PBC and ISF (Grant No. 4/11). The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

References

- ADAMS, B., OVSJANIKOV, M., WAND, M., SEIDEL, H.-P., AND GUIBAS, L. J. 2008. Meshless modeling of deformable shapes and their motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA ’08, 77–86.
- ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH ’00, 157–164.
- 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.

- BEN-CHEN, M., WEBER, O., AND GOTSMAN, C. 2009. Variational harmonic maps for space deformation. In *ACM SIGGRAPH 2009 Papers*, ACM, New York, NY, USA, SIGGRAPH '09, 34:1–34:11.
- BOMMES, D., CAMPEN, M., EBKE, H.-C., ALLIEZ, P., AND KOBELT, L. 2013. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* 32, 4 (July), 98:1–98:12.
- BOOKSTEIN, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 6 (June), 567–585.
- CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph.* 29, 4 (July), 38:1–38:6.
- CHEN, R., WEBER, O., KEREN, D., AND BEN-CHEN, M. 2013. Planar shape interpolation with bounded distortion. *ACM Trans. Graph.* 32, 4 (July), 108:1–108:12.
- FLOATER, M., AND KOSINKA, J. 2010. On the injectivity of wachspress and mean value mappings between convex polygons. *Advances in Computational Mathematics* 32, 2, 163–174.
- FLOATER, M. S. 2003. Mean value coordinates. *Comput. Aided Geom. Des.* 20, 1 (Mar.), 19–27.
- FRIES T.-P., M. H. G. 2003. Classification and overview of mesh-free methods. Tech. Rep. 2003-03, TU Brunswick, Germany.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.* 24, 3 (July), 1134–1141.
- JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (July).
- JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (July), 561–566.
- KOSINKA, J., AND BARTON, M. 2010. Injective shape deformations using cube-like cages. *Computer-Aided Design and Applications* 7, 3, 309–318.
- LEHTO, O., AND VIRTANEN, K. 1973. *Quasiconformal mappings in the plane*. Grundlehren der mathematischen Wissenschaften. Springer.
- LEVI, Z., AND LEVIN, D. 2013. Shape deformation via interior rbf. *IEEE Transactions on Visualization and Computer Graphics* 99, PrePrints, 1.
- LI, X.-Y., JU, T., AND HU, S.-M. 2013. Cubic mean value coordinates. *ACM Transactions on Graphics* 32, 4, 126:1–10.
- LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. *ACM Trans. Graph.* 27, 3 (Aug.), 78:1–78:10.
- LIPMAN, Y. 2012. Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph.* 31, 4 (July), 108:1–108:13.
- LIU, L., ZHANG, L., XU, Y., GOTSMAN, C., AND GORTLER, S. J. 2008. A local/global approach to mesh parameterization. In *Proceedings of the Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '08, 1495–1504.
- PETERS, J., AND REIF, U. 1998. Analysis of algorithms generalizing b-spline subdivision. *SIAM J. Numer. Anal.* 35, 2 (Apr.), 728–748.
- REIF, U. 1995. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design* 12, 2, 153 – 174.
- SCHNEIDER, T., HORMANN, K., AND FLOATER, M. S. 2013. Bijective composite mean value mappings. *Computer Graphics Forum* 32, 5 (July), 137–146. Proceedings of SGP.
- SCHÜLLER, C., KAVAN, L., PANOZZO, D., AND SORKINE-HORNUNG, O. 2013. Locally injective mappings. *Computer Graphics Forum (proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing)* 32, 5, 125–135.
- SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '86, 151–160.
- SHEFFER, A., AND KRAEVOY, V. 2004. Pyramid coordinates for morphing and deformation. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2Nd International Symposium*, IEEE Computer Society, Washington, DC, USA, 3DPVT '04, 68–75.
- SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, 109–116.
- SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, 175–184.
- TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '87, 205–214.
- WARREN, J. 1996. Barycentric coordinates for convex polytopes. In *Advances in Computational Mathematics* 6, 97–108.
- WEBER, O., PORANNE, R., AND GOTSMAN, C. 2012. Biharmonic coordinates. *Comput. Graph. Forum* 31, 8, 2409–2422.
- WENDLAND, H. 2004. *Scattered Data Approximation*. Cambridge University Press.
- XU, D., ZHANG, H., WANG, Q., AND BAO, H. 2005. Poisson shape interpolation. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, ACM, New York, NY, USA, SPM '05, 267–274.
- ZORIN, D. 1998. *A Method for Analysis of C1-continuity of Subdivision Surfaces*. Technical report (Stanford University. Computer Systems Laboratory). Computer Systems Laboratory, Stanford University.

Appendix A

In this appendix we compute the modulus $\omega_{\nabla\mathcal{F}}$ of the basis functions used in this paper as detailed in Table 1. Excluding Thin-Plate Splines, the modulus of all the basis functions (B-Splines and Gaussians) is of the standard Lipschitz type: $\omega_{\nabla\mathcal{F}}(t) = Lt$, where $L > 0$ is called the Lipschitz constant.

We start by formulating a small lemma that would be of help in calculating the modulus for the Lipschitz case:

Lemma 3. *Let $g : \Omega \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ be a C^2 function, where Ω is a convex domain. Then ∇g is L -Lipschitz with*

$$L \leq \max_{\mathbf{x} \in \Omega} \|\mathbf{H}g(\mathbf{x})\|_2, \quad (34)$$

where $\mathbf{H}g$ is the Hessian matrix of g and $\|\mathbf{H}g(\mathbf{x})\|_2$ denotes the operator 2-norm of the $\mathbf{H}g(\mathbf{x})$.

Proof. Let $\mathbf{x}, \mathbf{y} \in \Omega$ and let $\gamma(t) = (1-t)\mathbf{x} + t\mathbf{y}$, $t \in [0, 1]$ be the straight path between \mathbf{x} and \mathbf{y} , with constant speed and length $\|\mathbf{x} - \mathbf{y}\|$. Then

$$\nabla g(\mathbf{y})^T - \nabla g(\mathbf{x})^T = \int_0^1 \mathbf{H}g(\gamma(t)) \cdot \gamma'(t) dt, \quad (35)$$

Hence,

$$\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\| = \left\| \int_0^1 \mathbf{H}g(\gamma(t)) \cdot \gamma'(t) dt \right\| \quad (36)$$

$$\leq \int_0^1 \|\mathbf{H}g(\gamma(t))\|_2 \cdot \|\gamma'(t)\| dt \quad (37)$$

$$\leq \|\mathbf{x} - \mathbf{y}\| \max_{\mathbf{x} \in \Omega} \|\mathbf{H}g(\mathbf{x})\|_2. \quad (38)$$

□

B-splines. We use tensor product of uniform cubic B-splines in this paper. The uniform univariate cubic B-spline $\mathcal{B}^{(3)}(x)$ is classically defined on the interval $[0, 4]$, with knot spacing equal to one by,

$$\begin{cases} \frac{1}{6}x^3 & 0 \leq x \leq 1 \\ \frac{1}{6}(x^3 - 4(x-1)^3) & 1 \leq x \leq 2 \\ \frac{1}{6}((4-x)^3 - 4(x-3)^3) & 2 \leq x \leq 3 \\ \frac{1}{6}(4-x)^3 & 3 \leq x \leq 4 \end{cases} \quad (39)$$

Uniform cubic B-splines on longer knot vectors are just shifted copies of $\mathcal{B}^{(3)}(x)$. For our purposes, we use scaled versions of $\mathcal{B}^{(3)}(x)$, which we define by $\mathcal{B}_{\Delta}^{(3)}(x) = \mathcal{B}^{(3)}(\frac{x}{\Delta})$. The tensor product is then defined by $\mathcal{B}_{\Delta}^{(3)}(x)\mathcal{B}_{\Delta}^{(3)}(y)$. By direct computation we can see that $\|\mathbf{H}f_i(\mathbf{x})\|_2 \leq \frac{4}{3\Delta^2}$ for all $\mathbf{x} \in \mathbb{R}^2$ and by Lemma 3 we have that $\omega_{\nabla\mathcal{F}}(t) = \frac{4}{3\Delta^2}t$.

Gaussians are examples of radial basis functions [Wendland 2004] defined via $f_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2s^2}\right)$, where $\mathbf{x}_i \in \Omega$ are called centers, and $s > 0$ is a constant controlling the width of the Gaussian. Without loosing generality we assume $\mathbf{x}_i = \mathbf{0}$, and, again, direct calculation shows that $\|\mathbf{H}f_i(\mathbf{x})\|_2 \leq \frac{1}{s^2}$ for all $\mathbf{x} \in \mathbb{R}^2$, and therefore by Lemma 3 we have $\omega_{\nabla\mathcal{F}}(t) = \frac{1}{s^2}t$.

Thin-Plate Splines (TPS) are also radial basis functions defined via $f_i(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_i\|^2 \ln \|\mathbf{x} - \mathbf{x}_i\|^2$. The gradients are $\nabla f_i(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_i) [1 + \ln \|\mathbf{x} - \mathbf{x}_i\|^2]$. We show that the situation with TPS is slightly more complicated than with the other

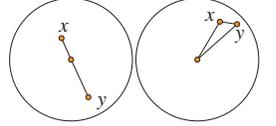
bases described above and that their gradients ∇f_i are *locally almost Lipschitz* in the sense that the modulus function is of the form $\omega_{\nabla\mathcal{F}}(t) = t(5.8 + 5 \ln t)$, for $0 \leq t \leq (1.25e)^{-1} \approx 0.29$. That is, it applies only when $\|\mathbf{x} - \mathbf{y}\| \leq (1.25e)^{-1}$. However, this limitation is not important as our fill-distance is in practice always (much) smaller than this constant, and therefore we always apply the modulus of continuity to smaller distances. We mention that although we expected this result to be known, we could not find it or an equivalent result in existing literature so we prove it here.

We assume w.l.o.g. that $\mathbf{x}_i = \mathbf{0}$, set $\tau = 1.25$ and recall that we assume $\|\mathbf{x} - \mathbf{y}\| \leq (\tau e)^{-1}$. First,

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\| + \|\mathbf{x} \ln \|\mathbf{x}\|^2 - \mathbf{y} \ln \|\mathbf{y}\|^2\|. \quad (40)$$

Focusing on the second term,

we observe two cases: (i) $\max\{\|\mathbf{x}\|, \|\mathbf{y}\|\} \leq \tau \|\mathbf{x} - \mathbf{y}\|$ (e.g., left in inset); and (ii) $\max\{\|\mathbf{x}\|, \|\mathbf{y}\|\} > \tau \|\mathbf{y} - \mathbf{x}\|$ (e.g., right in inset).



For case (i), note that the function $g(t) = t |\ln t^2|$ is monotonically increasing for $t \in [0, e^{-1}]$ and therefore we can replace $\|\mathbf{x}\|, \|\mathbf{y}\|$ with $\tau \|\mathbf{x} - \mathbf{y}\| \leq e^{-1}$, as follows,

$$\begin{aligned} \|\mathbf{x} \ln \|\mathbf{x}\|^2 - \mathbf{y} \ln \|\mathbf{y}\|^2\| &\leq \|\mathbf{x}\| |\ln \|\mathbf{x}\|^2| + \|\mathbf{y}\| |\ln \|\mathbf{y}\|^2| \\ &\leq 2\tau \|\mathbf{x} - \mathbf{y}\| |\ln \tau^2 \|\mathbf{x} - \mathbf{y}\|^2|. \end{aligned}$$

Combining this with eq. (40) we get after rearranging,

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\| \left[1 + 2\tau |\ln \tau^2| + 2\tau |\ln(\|\mathbf{x} - \mathbf{y}\|^2)| \right].$$

For case (ii), the bound (38) in the proof of Lemma 3 can be refined a bit: Instead of taking the maximum of $\|\mathbf{H}w(\mathbf{x})\|$ over all $\mathbf{x} \in \Omega$, one can take the maximum over all the points $\{\gamma(t) | t \in [0, 1]\}$. In our case, this is the straight line $(1-t)\mathbf{x} + t\mathbf{y}$. We assume that $\max\{\|\mathbf{x}\|, \|\mathbf{y}\|\} > \tau \|\mathbf{x} - \mathbf{y}\|$, and therefore it can be shown that $\|(1-t)\mathbf{x} + t\mathbf{y}\| \geq (\tau - 1)\|\mathbf{x} - \mathbf{y}\|$ for all $t \in [0, 1]$ (e.g., by bounding the distance $\|((1-t)\mathbf{x} + t\mathbf{y}) - \mathbf{y}\|$). A direct computation shows that $\|\mathbf{H}f_i(\mathbf{x})\|_2 \leq 3 + |\ln \|\mathbf{x}\|^2|$. Therefore,

$$\|\mathbf{H}f_i(\mathbf{x})\|_2 \leq 3 + |\ln(\tau - 1)^2 \|\mathbf{x} - \mathbf{y}\|^2|,$$

on the line $(1-t)\mathbf{x} + t\mathbf{y}$, $t \in [0, 1]$. The refined version described above of the bound (38) implies that

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\| \left[3 + |\ln(\tau - 1)^2| + |\ln \|\mathbf{x} - \mathbf{y}\|^2| \right].$$

Plugging in $\tau = 1.25$ and combining the two cases we get:

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\| \left[5.8 + 5 |\ln \|\mathbf{x} - \mathbf{y}\| \right].$$

Appendix B

We prove Lemma 2:

Lemma 2. *Let ∇u and ∇v be ω -continuous in Ω . Then both singular values functions Σ and σ are 2ω -continuous.*

Proof. The key to the proof is in eqs. (19) and (20). From (19) we find that a modulus of continuity for both vector valued functions (maps) $J_{\mathcal{S}}f$ and $J_{\mathcal{A}}f$ is ω , by noting that the sum (or difference) of ω -continuous maps is 2ω -continuous. By the triangle inequality their norms are also ω -continuous. Using eq. (20) we thus reach our conclusion. □