Shape-from-Template

Adrien Bartoli Yan Gérard François Chadebecq Toby Collins Daniel Pizarro

ISIT - ALCoV, Clermont Université, France

IEEE Transactions on Pattern Analysis and Machine Intelligence

Accepted January 2015

Abstract

We study a problem that we call *Shape-from-Template*, which is the problem of reconstructing the shape of a deformable surface from a single image and a 3D template. Current methods in the literature address the case of isometric deformations, and relax the isometry constraint to the convex inextensibility constraint, solved using the so-called maximum depth heuristic. We call these methods zeroth-order since they use image point locations (the zeroth-order differential structure) to solve the shape inference problem from a perspective image.

We propose a novel class of methods that we call first-order. The key idea is to use both image point locations and their first-order differential structure. The latter can be easily extracted from a warp between the template and the input image. We give a unified problem formulation as a system of PDEs for isometric and conformal surfaces that we solve *analytically*. This has important consequences. First, it gives the first *analytical algorithms* to solve this type of reconstruction problems. Second, it gives the first algorithms to solve for the *exact constraints*. Third, it allows us to study the *well-posedness* of this type of reconstruction: we establish that isometric surfaces can be reconstructed unambiguously and that conformal surfaces can be reconstructed up to a few discrete ambiguities and a global scale. In the latter case, the candidate solution surfaces are obtained analytically.

Experimental results on simulated and real data show that our isometric methods generally perform as well as or outperform state of the art approaches in terms of reconstruction accuracy, while our conformal methods largely outperform all isometric methods for extensible deformations.

Contents

1 Introduction

The reconstruction of 3D shape from monocular video data has been an extremely important research topic in computer vision for decades. It has been established that the scene shape may be recoverable from image motion. Crucially, the degree to which it can be uniquely recovered depends on the assumption made on the shape's (temporal) deformation and on the camera projection function to some extent. It has been established that if the shape is rigid (does not change over time) it is then uniquely recoverable, as well as camera motion and intrinsic parameters (Faugeras et al., 2001; Hartley and Zisserman, 2003) by rigid Shape-from-Motion (SfM). When the shape is deformable (changes over time) however, the problem is still largely open, since modeling deformation priors is substantially more difficult than using simple rigidity, and recovering time-varying shape is very weakly constrained as compared to recovering rigid shape. Efforts have been recently made by the community towards solving this problem known as Non-Rigid SfM (NRSfM) (Bregler et al., 2000; Del Bue, 2008; Russell et al., 2011; Torresani et al., 2008; Vicente and Agapito, 2012) under various assumptions and deformation priors.

We are here interested in a specific setup which has been known as *template-based deformable 3D reconstruction* (Brunet et al., 2010; Chhatkuli et al., 2014; Ecker et al., 2008; Gumerov et al., 2006; Malti et al., 2011a,b; Moreno-Noguer et al., 2009, 2010; Perriollat et al., 2011; Salzmann and Fua, 2009, 2011; Salzmann et al., 2007a,b, 2008; Shaji et al., 2010; Vicente and Agapito, 2012) and that we shall here call Shape-from-Template (SfT), as argued below. This setup is interesting since it resolves 3D shape from one input image only. In SfT, one wishes to reconstruct the apparent surface of an object for which a reference 3D shape –called the template– is available, and under a specific *deformation constraint*. Most of the recent work tackles the IsoPSfT problem (Isometric Perspective SfT): they make the assumption that the surface deforms isometrically and that the camera can be modeled by perspective projection. They also make the assumption that 3D to 2D point correspondences can be established between the template and the input image. We call these methods *zeroth-order methods*. Isometry implies that the geodesic between any two points on the surface remains constant; this is a non-convex constraint. The current literature proposes numerical methods to solve convex relaxations of the isometric deformation constraint. Isometry has also been combined with shading (Moreno-Noguer et al., 2010), the problem's dimensionality reduced by learning the admissible shape space (Moreno-Noguer et al., 2010) (originally proposed in (Salzmann et al., 2007c)) or stitching multiple local surface patches of low complexity (Salzmann and Fua, 2011). Some other works reconstruct elastic surfaces using shading (Moreno-Noguer et al., 2009), quasi-conformal deformations (Malti et al., 2011a) or temporal smoothness (Salzmann et al., 2007a).

Our work tackles the problem with a different modeling. Instead of using point correspondences, we assume that a warp can be established that maps points from the template to the input image. The literature on warp estimation is well-advanced. In our approach, the surface is detected in the input image and a warp may be computed following

the feature-based approach (Pilet et al., 2008), the pixel-based approach (Gay-Bellile et al., 2010) or a combination or both (Pizarro and Bartoli, 2012). In essence, a warp captures and densifies the 3D to 2D point correspondences used in other approaches. But in our approach, the warp needs not be dense or continuous, provided that it gives the first-order differential structure around each point to be reconstructed. We give a differential formulation of the shape inference problem and show that it leads to a class of *first-order methods*. This formulation constrains the surface to be continuously differentiable at each point to be reconstructed, which must thus not be located on a sharp fold or on a tear. It allows us to achieve three major steps forward:

- **Beyond isometric deformation.** We relax isometric deformation (distance-preserving) to conformal deformation (angle-preserving). While an isometry is also a conformity, the opposite is not always true. This way, we show that some types of extensible surfaces can be reconstructed from image motion only.

- **Beyond convex relaxations.** We solve the exact non-convex differential constraints without relaxing them, in both the isometric and conformal deformation cases.

- **Beyond numerical results.** We provide analytical solutions to the problem and prove that for the isometric deformation a unique solution exists, while for conformal deformations a finite number of solutions exists, each recoverable up to a global scale ambiguity. Despite the amount of recent literature concerned with solving SfT, very little effort has been devoted to analyzing the problem's well-posedness from a theoretical standpoint.

Our approach handles non-developable surfaces, modeled by non-flat templates, and thus applies to the surface of objects such as a balloon, a cushion and an organ, which cannot be physically flattened.

The present article is a fully rewritten extension of one of our conference articles (Bartoli et al., 2012). It features several significant novelties. First, our novel framework handles generic template parameterization (while our previous framework handles only conformal parameterizations). This is important since only quasi-conformal (and not exactly conformal) flattening can be computed from discrete shape representations such as meshes. The results obtained by our novel framework are independent of the actual flattening. This is because no assumptions are made on the type of flattening. This allows us to use an image of the object registered to a 3D model to serve as flattened template and 3D template, respectively. Second, we derive all our main results in a more general framework which clearly identifies the solutions as eigenvalues and eigenvectors of (2×2) matrices. We recently published a study of the solutions' stability with respect to the imaging geometry (Chhatkuli et al., 2014). This study shows that the depth solution is stable for a perspective camera but degrades as the camera becomes affine. We also used the principle of first-order shape inference methods in order to solve for the camera's focal length. Our first solution optimizes the shape's global isometricity by sampling the focal length (Bartoli and Collins, 2013). Our second solution solves for the focal length in closed-form by using a simplified camera model (Bartoli et al., 2013). Third, we identify that there

exist singular points in the conformal deformation case which correspond to surface patches whose normal coincides with the line of sight.

We demonstrate our algorithms and compare them to existing ones using simulated data and three real datasets. These experiments show that our algorithms compete with and often outperform state of the art approaches. They also show that 3D reconstruction for extensible surfaces *is possible* with the conformal deformation model.

Naming template-based monocular 3D reconstruction as Shape-from-Template. The scope of template-based monocular 3D reconstruction is to find the 3D shape of a deformable surface under a simple physics-based deformation model. This problem has been studied over the last decade, but still the class of methods has not so far been given a specific name. We believe that Shape-from-Template (SfT), which we use in this paper, would be an appropriate name. SfT is related to the Shape-from-X type of names, and emphasizes the fact that the problem leads to reconstructing shape. We noted that the acronym SfT also holds for Shape-from-Texture (Ikeuchi, 1984; Ohta et al., 1981). This is actually fortunate since one may easily realize that texton-based Shape-from-Texture is a very similar problem to Shape-from-Template: imagine that the texton is a small part of the template, and that the set of textons thus forms the template up to some planar rearrangement.

Notation. We use greek characters for functions (such as φ) and bold latin for vectors and matrices (such as $\mathbf{u}$). We write scalars with either greek or latin italic characters. We use $C^r(\mathcal{X},\mathcal{Y})$ for the space of r times continuously differentiable functions with domain $\mathcal{X}$ and range $\mathcal{Y}$. The first derivative operator is written $\mathbf{J}$ (such as $\mathbf{J}_\varphi$). We write $\mathbb{S}$ as the set of (2×2) Symmetric Positive Definite (SPD) matrices. We use λ_1 and λ_2 as two functions giving the eigenvalues of some (2×2) matrix, with $\lambda_1 \geq \lambda_2$, and ϵ_1 and ϵ_2 as two functions giving the associated eigenvectors in an orthonormal basis. For $\mathbf{A}\in\mathbb{S}$, we have $\mathbf{A}=\mathbf{P}\mathbf{\Sigma}\mathbf{P}^\top$ with $\mathbf{P}=(\epsilon_1(\mathbf{A}) \quad \epsilon_2(\mathbf{A}))$, $\mathbf{P}\mathbf{P}^\top=\mathrm{I}$ and $\mathbf{\Sigma}=\mathrm{diag}(\lambda_1(\mathbf{A}),\lambda_2(\mathbf{A}))$. Homogeneous coordinates are written with a tilde (such as $\tilde{\mathbf{u}}$, with $\tilde{\mathbf{u}}^\top \stackrel{\text{def}}{=} (\mathbf{u}^\top \ 1)$).

2 Problem Formulation

2.1 Geometric Modeling

Our problem modeling is illustrated in figure 1. The known template shape $\mathcal{T}\subset\mathbb{R}^3$ is related to the unknown deformed shape $\mathcal{S}\subset\mathbb{R}^3$ by an unknown continuously differentiable 3D deformation $\Psi\in C^1(\mathcal{T},\mathbb{R}^3)$. In other words, Ψ maps a point $\mathbf{P}\in\mathcal{T}$ to the corresponding point $\mathbf{Q}=\Psi(\mathbf{P})\in\mathcal{S}$. We assume that $\mathcal{T}$ and $\mathcal{S}$ share the same topology. Therefore, they also share a parameterization space $\mathcal{P}\subset\mathbb{R}^2$, which can be obtained by conformal flattening of $\mathcal{T}$, for instance. The known template embedding $\Delta\in C^1(\mathcal{P},\mathcal{T})$ is an invertible function that maps a point $\mathbf{p}\in\mathcal{P}$ to the 3D point $\mathbf{P}\in\mathcal{T}$. The unknown deformed embedding $\varphi\in C^1(\mathcal{P},\mathbb{R}^3)$ maps a point $\mathbf{p}\in\mathcal{P}$ to the 3D point $\mathbf{Q}\in\mathcal{S}$. It

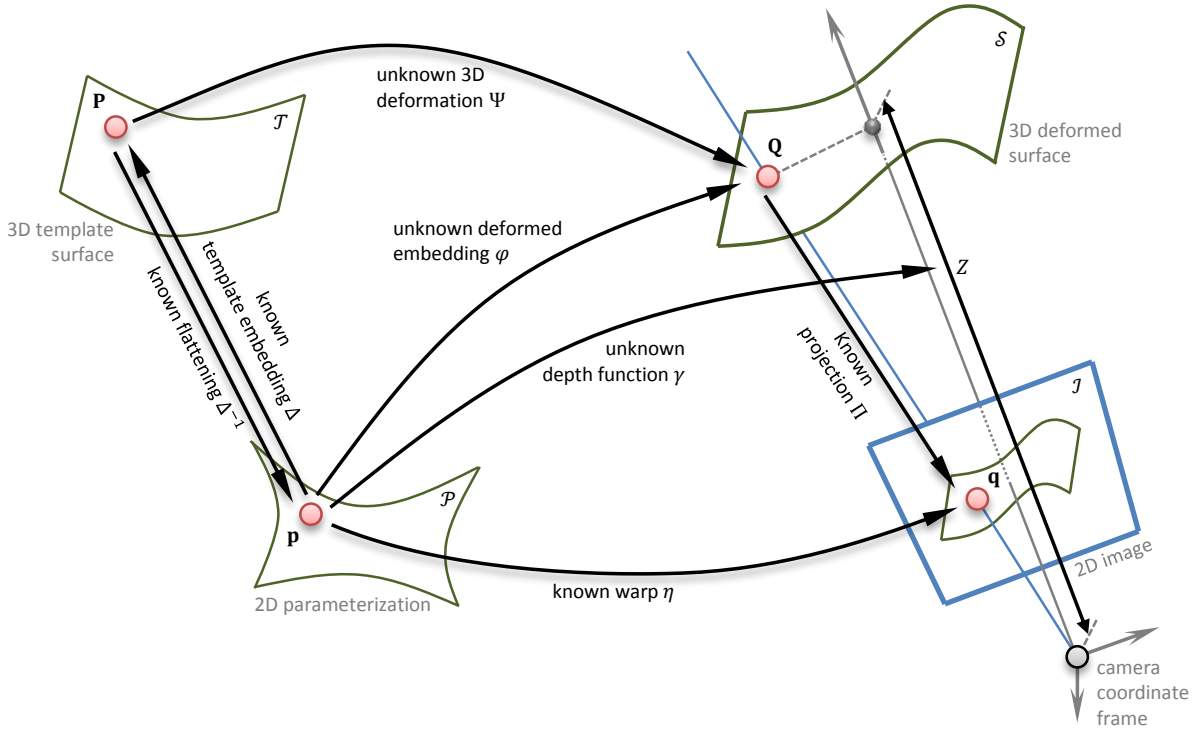


Figure 1: **Geometric modeling of SfT.** We propose to use a deformed embedding to model SfT. This function maps a 2D template parameterization obtained from the 3D template to the unknown 3D deformed surface. In our algorithms, any C^1 2D parameterization can be used.

is defined as:

$$\varphi \stackrel{\text{def}}{=} \Psi \circ \Delta. \tag{1}$$

It is thus clear that estimating φ and Ψ are two different ways to solve SfT.

The input image space is written $\mathcal{I} \subset \mathbb{R}^2$. The known camera operator $\Pi \in C^\infty(\mathbb{R}^3, \mathbb{R}^2)$ maps a point $\mathbf{Q} \in \mathcal{S}$ to $\mathbf{q} = \Pi(\mathbf{Q}) \in \mathcal{I}$ using perspective projection[1]. We assume that the camera has been intrinsically calibrated and that the effect of the intrinsics are 'undone' in the image. Writing $\mathbf{q}^\top = (x \quad y)$ and $\mathbf{Q}^\top = (X \quad Y \quad Z)$, the projection operator is thus simply:

$$\Pi(\mathbf{Q}) \stackrel{\text{def}}{=} \frac{1}{Z}\begin{pmatrix} X \\ Y \end{pmatrix}. \tag{2}$$

The known warp function is written $\eta \in C^1(\mathcal{P}, \mathcal{I})$. It maps a point $\mathbf{p} \in \mathcal{P}$ to $\mathbf{q} = \eta(\mathbf{p}) \in \mathcal{I}$. In our method, the warp function can be continuously defined in $\mathcal{P}$ or only at some sparse points, along with its first-order differential structure $\mathbf{J}_\eta$. Practical details on warp estimation are given in §8.

[1]Strictly speaking, Π is not defined at the camera centre, but if the observed surface would contain that point, it would anyway not be imaged.

2.2 The Parameterization Space

The parameterization space $\mathcal{P} \subset \mathbb{R}^2$ is obtained from the template shape $\mathcal{T}$ prior to running SfT. We distinguish two cases. The first case concerns developable surfaces, such as a piece of paper, which can be isometrically (and physically) flattened. This case has been the focus of many SfT methods in the literature. For these surfaces, the template embedding is a trivial function and may for instance be chosen as the identity. The second case concerns non-developable surfaces, such as a cushion, which cannot be physically flattened. We use rigid SfM to reconstruct the template shape, but any other shape reconstruction method could be used instead. Our SfT solution can use any continuously differentiable parameterization space. We have tried two solutions to form the parameterization space. The first one is to simply identify one of the images used in rigid SfM to the parameterization space. The main drawback of this solution is that some objects cannot be made entirely visible in a single perspective image. The second solution is to use quasi-conformal flattening. This is a very well-established technique in the computer graphics community, where it has been extensively used for texture-mapping purposes (Sheffer et al., 2005).

2.3 Differential Algebraic Modeling

The goal of SfT is to compute the 3D deformation Ψ or equivalently the embedding φ from image data which is here captured by the warp η. SfT is based on two types of constraints: the *reprojection constraint* and the *deformation constraint*. The former models compatibility with the image data and the latter models priors on the 3D deformation. We are here interested in isometric and conformal 3D deformations, which are first-order differential deformation priors that can both be written in terms of the deformation's local structure tensor $\mathbf{J}_{\Psi,\mathcal{T}}^{\top}\mathbf{J}_{\Psi,\mathcal{T}}$. The notation $\mathbf{J}_{\Psi,\mathcal{T}}$ refers to the first-order derivative operator acting on the tangent space[2] of $\mathcal{T}$. Isometric deformations preserve distances and are modeled by $\mathbf{J}_{\Psi,\mathcal{T}}^{\top}\mathbf{J}_{\Psi,\mathcal{T}} = \mathrm{I}$ while conformal deformations preserve angles and are modeled by $\mathbf{J}_{\Psi,\mathcal{T}}^{\top}\mathbf{J}_{\Psi,\mathcal{T}} \propto \mathrm{I}$ since local scaling varies isotropically. Both types of deformations can thus be written as $\mathbf{J}_{\Psi,\mathcal{T}}^{\top}\mathbf{J}_{\Psi,\mathcal{T}} = \beta\mathrm{I}$ by introducing a local scaling function $\beta \in C^1(\mathcal{T}, \mathbb{R}^{+*})$ with $\beta = 1$ in the isometric case and β unknown in the conformal case. The Isometric and Conformal Perspective SfT problems, respectively named as IsoPSfT and ConPSfT, can therefore be stated as:

IsoPSfT – Initial formulation

$$\text{Find } \Psi \in C^1(\mathcal{T}, \mathbb{R}^3) \qquad \text{s.t.} \qquad \begin{cases} \Pi \circ \Psi \circ \Delta &= \eta & \text{(reprojection constraint)} \\ \mathbf{J}_{\Psi,\mathcal{T}}^{\top}\mathbf{J}_{\Psi,\mathcal{T}} &= \mathrm{I} & \text{(deformation constraint)} \end{cases} \tag{3}$$

[2]For a smooth surface embedded in the 3D space the tangent space is a plane which linearly approximates the local shape.

ConPSfT – Initial formulation

$$\text{Find} \begin{cases} \Psi & \in & C^1(\mathcal{T}, \mathbb{R}^3) \\ \beta & \in & C^1(\mathcal{T}, \mathbb{R}^{+*}) \end{cases} \quad \text{s.t.} \quad \begin{cases} \Pi \circ \Psi \circ \Delta & = & \eta & \text{(reprojection constraint)} \\ \mathbf{J}_{\Psi,\mathcal{T}}^{\top} \mathbf{J}_{\Psi,\mathcal{T}} & = & \beta \mathrm{I} & \text{(deformation constraint)} \end{cases} \tag{4}$$

Each system in IsoPSfT and ConPSfT has five distinct PDEs, given by two matrix-valued PDEs, the reprojection and the deformation constraints. The former accounts for two PDEs, and the latter for three.

2.4 Overview of the Solutions' Derivation

As can be seen in their initial formulations, IsoPSfT and ConPSfT are first-order quadratic systems of PDEs. All systems of PDEs obtained by reformulating these formulations will be of the same kind. Our solutions' derivation follows a common framework for both problems with four main steps. The solution to IsoPSfT is obtained after the third step; the fourth step is specific to ConPSfT. Our goal is to reformulate both problems in such a way that the number of unknowns locally matches the number of independent PDEs. In the initial formulations, the systems of PDEs are not locally solvable: the number of local unknowns is greater than the number of PDEs. The first two steps we take are reformulations, which reduce the number of local unknowns, combine the reprojection and deformation constraints and introduce the first-order deformation constraint:

Step 1 (§3.1) This first reformulation substitutes the deformation function Ψ by the embedding function φ. This step prepares the next one by introducing the parameterization space in the formulation.

Step 2 (§3.2) This second reformulation substitutes the embedding function φ by the depth function γ. It allows us to combine the two matrix-valued PDEs forming each system into single matrix-valued PDEs, forming a system of three independent PDEs for each problem. Importantly, this combination uses the first-order derivatives of the reprojection constraint, and thus introduces the warp's first-order derivatives in the PDEs.

The resulting systems of PDEs are locally solvable for IsoPSfT, and ConPSfT to some extent. They are nonlinear, however, and deriving their analytical solutions requires two changes of variable:

Step 3 (§4) This first change of variable eliminates some of the nonlinear terms in the PDEs by introducing a novel unknown function θ and its first-order derivatives $\mathbf{J}_\theta$. It leaves three nonlinear terms in the formulation. Because the systems of PDEs are formed of three independent PDEs, the three nonlinear terms can be solved locally for IsoPSfT, as what we call a rank-1 equation of type 1 (§5). At this stage, we can show that the solution to θ, and thus to γ, φ and Ψ, is unique. ConPSfT however has a fourth term, the local scaling function, and can thus not be solved locally at this stage.

Step 4 (§6) This second change of variable eliminates function θ and its derivatives by introducing a novel unknown function μ. It turns out that only the first-order derivatives $\mathbf{J}_\mu$ of μ appear in the resulting PDEs, not μ itself.

The system of PDEs can thus be solved locally, as what we call a rank-1 equation of type 2 (§7). At this stage, we can show that the solution to $\mathbf{J}_\mu$ is two-fold ambiguous. This solution implies that the final algorithm must, once having solved for $\mathbf{J}_\mu$, integrate it to find μ, while coping with the local two-fold ambiguity, and only then proceed to recover γ, φ and Ψ.

Our methods solve the problems locally -or semi-locally for ConPSfT- and obtain numerically stable results. The price to pay for the local solutions is the use of the warp's first-order derivatives. Once the warp is computed, for instance from point correspondences, its first-order derivatives can be obtained analytically for most warp models such as the Thin-Plate Spline. This implies that the warp must be differentiable, at least locally around the points to be reconstructed. Our methods can thus not reconstruct sharp folds, nor can it deal with changes of topology.

3 Problem Reformulation

The main goals of reformulation is to introduce the depth function $\gamma \in C^1(\mathcal{P}, \mathbb{R})$, in place of the deformation function Ψ, and the first-order reprojection constraint. This is an essential step to then solve the problems locally. This is because the depth function is a simpler function to work with than the deformation function, essentially because its dimensions are lower, and because the resulting problem formulation lends itself to local solutions. We first introduce the embedding function, and then the depth function.

3.1 Formulation with the Embedding

The deformation function, used as main unknown, does not depend on the parameterization space. The depth function -whose domain is the parameterization space- can thus not be introduced directly, and requires us to first introduce the embedding function. This can be done quite directly since the deformation function can be expressed in terms of the embedding function φ using equation (1) as $\Psi = \varphi \circ \Delta^{-1}$. It is straightforward to rewrite the reprojection constraint in terms of φ as $\Pi \circ \varphi = \eta$, or simply $\varphi = \gamma\tilde{\eta}$.

A novel deformation constraint, expressed on the embedding function, and independent of the deformation function, is formed as follow. The idea is first to differentiate the embedding's definition, and then to eliminate the deformation function using its deformation constraint. Differentiating the embedding's definition (1) gives:

$$\mathbf{J}_\varphi \;=\; (\mathbf{J}_\Psi \circ \Delta)\, \mathbf{J}_\Delta.$$

Multiplying each side of the equation to the left by its transpose -so as to use the deformation constraint on Ψ- leads to:

$$\mathbf{J}_\varphi^\top \mathbf{J}_\varphi \;=\; \mathbf{J}_\Delta^\top (\mathbf{J}_\Psi \circ \Delta)^\top (\mathbf{J}_\Psi \circ \Delta)\, \mathbf{J}_\Delta \;=\; (\beta \circ \Delta)\, \mathbf{J}_\Delta^\top \mathbf{J}_\Delta \;=\; \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta,$$

with $\nu \stackrel{\text{def}}{=} \beta \circ \Delta$, $\nu \in C^1(\mathcal{P}, \mathbb{R}^{+*})$ being a local scaling function (known in the isometric case and unknown in the conformal case) and $\mathbf{J}_\Delta^\top \mathbf{J}_\Delta \in C^0(\mathcal{P}, \mathbb{S})$ is a known function depending on the chosen parameterization. The two SfT problems at hand can thus be rewritten as:

IsoPSfT and ConPSfT – Formulation with the deformed embedding φ

$$\text{Find} \begin{cases} \varphi \in C^1(\mathcal{P}, \mathbb{R}^3) \\ \left[\nu \in C^1(\mathcal{P}, \mathbb{R}^{+*}) \right] \end{cases} \quad \text{s.t.} \quad \begin{cases} \Pi \circ \varphi = \eta & \text{(reprojection constraint)} \\ \mathbf{J}_\varphi^\top \mathbf{J}_\varphi = \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta & \text{(deformation constraint)} \end{cases} \tag{5}$$

Note: the local scaling function ν is given by $\nu = 1$ for IsoPSfT and is unknown for ConPSfT.

3.2 Formulation with the Depth Function

From the formulation with the embedding function, we can now introduce the depth function. This can be done quite directly, using the projection function. Our reformulation combines the reprojection and deformation constraints. This is done by introducing the first-order reprojection constraint, which, importantly, increases the number of independent PDEs and facilitates the derivation of local solutions.

Using the definition (2) of the projection function, the reprojection constraint can be simply rewritten as:

$$\varphi = \gamma \tilde{\eta}. \tag{6}$$

Using the depth function γ as unknown thus enforces the reprojection constraint exactly, and our goal then becomes to find constraints on γ from the deformation constraint:

$$\mathbf{J}_\varphi^\top \mathbf{J}_\varphi = \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta. \tag{7}$$

This requires us to introduce the first-order reprojection constraint, obtained by differentiating the reprojection constraint (6):

$$\mathbf{J}_\varphi = \tilde{\eta} \mathbf{J}_\gamma + \gamma \mathbf{J}_{\tilde{\eta}}. \tag{8}$$

We note that:

$$\mathbf{J}_{\tilde{\eta}} = \begin{pmatrix} \mathbf{J}_\eta \\ 0\,0 \end{pmatrix}, \quad \mathbf{J}_{\tilde{\eta}}^\top \mathbf{J}_{\tilde{\eta}} = \mathbf{J}_\eta^\top \mathbf{J}_\eta \quad \text{and} \quad \tilde{\eta}^\top \mathbf{J}_{\tilde{\eta}} = \eta^\top \mathbf{J}_\eta.$$

Substituting $\mathbf{J}_\varphi$ from equation (8) into the deformation constraint (7) we obtain the *Perspective SfT PDE*:

$$\|\tilde{\eta}\|_2^2 \mathbf{J}_\gamma^\top \mathbf{J}_\gamma + \gamma^2 \mathbf{J}_\eta^\top \mathbf{J}_\eta + \gamma \left(\mathbf{J}_\gamma^\top \eta^\top \mathbf{J}_\eta + \mathbf{J}_\eta^\top \eta \mathbf{J}_\gamma \right) = \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta. \tag{9}$$

It leads to the following new formulation:

IsoPSfT and ConPSfT – Formulation with the depth function γ

$$\text{Find} \left\{ \begin{array}{rcl} \gamma & \in & C^1(\mathcal{P}, \mathbb{R}^{+*}) \\ \left[\nu \right. & \in & \left. C^1(\mathcal{P}, \mathbb{R}^{+*}) \right] \end{array} \right. \quad \text{s.t.} \quad \|\tilde{\eta}\|_2^2 \mathbf{J}_\gamma^\top \mathbf{J}_\gamma + \gamma^2 \mathbf{J}_\eta^\top \mathbf{J}_\eta + \gamma \left(\mathbf{J}_\gamma^\top \eta^\top \mathbf{J}_\eta + \mathbf{J}_\eta^\top \eta \mathbf{J}_\gamma \right) = \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta \tag{10}$$

Note: the local scaling function ν is given by $\nu = 1$ for IsoPSfT and is unknown for ConPSfT.

This new formulation is based on a first-order quadratic matrix-valued PDE in γ forming a system of three PDEs. In the case of IsoPSfT, this formulation suggests the existence of a local solution, obtained by neglecting the relationship between γ and $\mathbf{J}_\gamma$: there are three local unknowns -one for γ and two for $\mathbf{J}_\gamma$- and three PDEs. However, the system involves nonlinear terms in $\mathbf{J}_\gamma^\top \mathbf{J}_\gamma$, γ^2 and $\gamma \mathbf{J}_\gamma$. Our solution involves a change of variable required to cancel the term $\gamma \mathbf{J}_\gamma$. In the case of ConPSfT, this formulation suggests that no local solution can be found: there are four local unknowns -one for γ, two for $\mathbf{J}_\gamma$ and one for ν- and three PDEs. However, we show that with a second change of variable, a formulation with no zeroth-order term in the unknown function can be obtained, and solved locally for the unknown function's first-order partial derivatives. This final formulation shows that in ConPSfT there is locally no difference between depth and rescaling.

4 First Change of Variable

The goal of this first change of variable is to eliminate the mixed terms, the products between the unknown function and its first-order derivatives. The resulting formulation is still nonlinear, but can be locally solved analytically. The new variable is a function θ which we define shortly. We first need to introduce $\varepsilon \stackrel{\text{def}}{=} \|\tilde{\eta}\|_2 \in C^1(\mathcal{P}, \mathbb{R}^{+*})$, implying:

$$\mathbf{J}_\varepsilon = \frac{1}{\varepsilon} \eta^\top \mathbf{J}_\eta \quad \text{and} \quad \varepsilon \mathbf{J}_\varepsilon = \eta^\top \mathbf{J}_\eta.$$

Substituting into the Perspective SfT PDE (9) we obtain:

$$\varepsilon^2 \mathbf{J}_\gamma^\top \mathbf{J}_\gamma + \gamma^2 \mathbf{J}_\eta^\top \mathbf{J}_\eta + \gamma \left(\mathbf{J}_\gamma^\top \mathbf{J}_\varepsilon + \mathbf{J}_\varepsilon^\top \mathbf{J}_\gamma \right) = \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta.$$

We now proceed to our first change of variable. We substitute γ for $\theta \in C^1(\mathcal{P}, \mathbb{R})$, with:

$$\theta \stackrel{\text{def}}{=} \gamma \varepsilon, \quad \gamma = \frac{\theta}{\varepsilon} \quad \text{and} \quad \mathbf{J}_\gamma = \frac{1}{\varepsilon^2} \left(\varepsilon \mathbf{J}_\theta - \theta \mathbf{J}_\varepsilon \right). \tag{11}$$

We note that $\varepsilon \geq 1$ and thus the change of variable always works. Function θ is defined as the product of the depth and ε, defined as the norm of $\tilde{\eta}$, the line of sight[3]. Therefore, as can be easily verified, $\theta = \|\varphi\|_2$, and may be interpreted as giving the distance of a surface's point to the camera centre. We show below that posing Perspective SfT problems with θ rather than γ allows one to avoid mixed terms involving the product of γ and $\mathbf{J}_\gamma$. Introducing θ gives the following rewriting of the Perspective SfT PDE:

$$\frac{1}{\varepsilon^2}\left(\varepsilon^2\mathbf{J}_\theta^\top\mathbf{J}_\theta + \theta^2\mathbf{J}_\varepsilon^\top\mathbf{J}_\varepsilon - \varepsilon\theta\left(\mathbf{J}_\theta^\top\mathbf{J}_\varepsilon + \mathbf{J}_\varepsilon^\top\mathbf{J}_\theta\right)\right) + \frac{\theta^2}{\varepsilon^2}\mathbf{J}_\eta^\top\mathbf{J}_\eta + \frac{\theta}{\varepsilon^2}\left(\varepsilon\mathbf{J}_\theta^\top\mathbf{J}_\varepsilon - \theta\mathbf{J}_\varepsilon^\top\mathbf{J}_\varepsilon + \varepsilon\mathbf{J}_\varepsilon^\top\mathbf{J}_\theta - \theta\mathbf{J}_\varepsilon^\top\mathbf{J}_\varepsilon\right) \;=\; \nu\mathbf{J}_\Delta^\top\mathbf{J}_\Delta.$$

Expanding and simplifying this equation leads to:

$$\mathbf{J}_\theta^\top\mathbf{J}_\theta + \frac{1}{\varepsilon^2}\left(\mathbf{J}_\eta^\top\mathbf{J}_\eta - \mathbf{J}_\varepsilon^\top\mathbf{J}_\varepsilon\right)\theta^2 \;=\; \nu\mathbf{J}_\Delta^\top\mathbf{J}_\Delta.$$

Replacing ε and $\mathbf{J}_\varepsilon$ by their definition, we finally arrive at:

$$\mathbf{J}_\theta^\top\mathbf{J}_\theta + \xi\theta^2 \;=\; \nu\mathbf{J}_\Delta^\top\mathbf{J}_\Delta \qquad \text{with} \qquad \xi \stackrel{\text{def}}{=} \frac{1}{\|\tilde{\eta}\|_2^2}\left(\mathbf{J}_\eta^\top\mathbf{J}_\eta - \frac{1}{\|\tilde{\eta}\|_2^2}\mathbf{J}_\eta^\top\eta\eta^\top\mathbf{J}_\eta\right). \tag{12}$$

We have that $\xi \in C^0(\mathcal{P}, \mathbb{S})$. First, because $\eta \in C^1(\mathcal{P}, \mathbb{R}^2)$ and $\|\tilde{\eta}\|_2^2 \geq 1$, it is clear that $\xi \in C^0(\mathcal{P}, \mathbb{R}^{2\times 2})$. Second, lemma 1 given in appendix A shows that $\xi(\mathbf{p}) \in \mathbb{S}$, for $\mathbf{p} \in \mathcal{P}$. Equation (12) shows how the distance of the surface's points to the camera centre must vary under isometric and conformal deformations to comply with a given warp. It represents a first-order quadratic PDE, as the original Perspective SfT PDE (9). However, it only depends on $\mathbf{J}_\theta^\top\mathbf{J}_\theta$ and θ^2, and is thus now in a shape which can be solved analytically. The new problem formulation is given by:

IsoPSfT and ConPSfT – Formulation with the θ function

$$\text{Find} \left\{ \begin{array}{rcl} \theta & \in & C^1(\mathcal{P}, \mathbb{R}^{+*}) \\ \left[\; \nu & \in & C^1(\mathcal{P}, \mathbb{R}^{+*}) \;\right] \end{array} \right. \qquad \text{s.t.} \qquad \mathbf{J}_\theta^\top\mathbf{J}_\theta + \xi\theta^2 \;=\; \nu\mathbf{J}_\Delta^\top\mathbf{J}_\Delta \tag{13}$$

Note: the local scaling function ν is given by $\nu = 1$ for IsoPSfT and is unknown for ConPSfT.

5 Isometric Solution

Equation (12) gives three constraints; the three unknowns in $\mathbf{J}_\theta$ and θ^2 can thus be solved for at each point independently. This last point is important to facilitate a fast implementation of the solution on a parallel architecture.

[3]We may interpret $\tilde{\eta} \propto \varphi$ as giving the coordinates of a point's line of sight.

5.1 Well-Posedness

Theorem 1 (Space of solutions for the IsoPSfT problem)**.** *The IsoPSfT problem (3) has a unique solution.*

Proof. We start from formulation (13) of the IsoPSfT problem. For a point $\mathbf{p} \in \mathcal{P}$, we define $\mathbf{u} \stackrel{\text{def}}{=} \mathbf{J}_\theta^\top(\mathbf{p}) \in \mathbb{R}^{2\times 1}$, $\mathbf{G} \stackrel{\text{def}}{=} \xi(\mathbf{p}) \in \mathbb{S}$, $z \stackrel{\text{def}}{=} \theta(\mathbf{p})^2 > 0$ and $\mathbf{K} \stackrel{\text{def}}{=} \mathbf{J}_\Delta^\top(\mathbf{p})\mathbf{J}_\Delta(\mathbf{p}) \in \mathbb{S}$. The PDE at point $\mathbf{p}$ is then rewritten as the following matrix equation:

$$\mathbf{u}\mathbf{u}^\top + z\mathbf{G} \;=\; \mathbf{K}.$$

We call this equation a rank-1 equation of type 1. Proposition 1 in appendix C shows that there is always a single solution for z, implying that function θ is uniquely defined. □

5.2 Algorithm

We derive our numerical algorithm based on the solution to the rank-1 equation of type 1 given in proposition 1 in appendix C. In practice we define a set of points $\mathcal{A}$ at which the deformed shape is to be reconstructed. This set depends on the nature of the warp and may thus be dense or sparse. Our algorithm is given in table 1.

▷ Inputs: warp η, template embedding Δ, point set $\mathcal{A} \subset \mathcal{P}$

▷ Outputs: deformed embedding φ for points in $\mathcal{A}$

- For $\mathbf{p} \in \mathcal{A}$ set

$$\varphi(\mathbf{p}) \leftarrow \sqrt{\lambda_2 \left(\mathbf{J}_\Delta^\top(\mathbf{p})\mathbf{J}_\Delta(\mathbf{p}) \left(\mathbf{J}_\eta^\top(\mathbf{p})\mathbf{J}_\eta(\mathbf{p}) - \frac{1}{\|\tilde{\eta}(\mathbf{p})\|_2^2} \mathbf{J}_\eta^\top(\mathbf{p})\eta(\mathbf{p})\eta(\mathbf{p})^\top \mathbf{J}_\eta(\mathbf{p}) \right)^{-1} \right)} \tilde{\eta}(\mathbf{p})$$

Table 1: **Algorithm implementing our analytical solution to IsoPSfT.** Each point may be reconstructed independently of the others. There is no restriction on the point set $\mathcal{A} \subset \mathcal{P}$, which may result of a sparse or dense discretization of the template to image registration.

6 Second Change of Variable

Our first change of variable allowed us to solve the IsoPSfT problem analytically. The ConPSfT problem requires a second change of variable to be solved: because the local scaling function ν is unknown, equation (12) cannot be solved locally (it gives three constraints and has four unknowns per point, the fourth one being the local scale factor). We thus substitute θ for $\mu \in C^1(\mathcal{P}, \mathbb{R})$ in equation (12), with:

$$\mu \;\stackrel{\text{def}}{=}\; \ln\theta, \qquad \theta \;=\; \exp(\mu) \qquad \text{and} \qquad \mathbf{J}_\theta \;=\; \exp(\mu)\mathbf{J}_\mu \;=\; \theta\mathbf{J}_\mu. \tag{14}$$

We note that $\theta > 0$ and thus the change of variable always works. Function μ is the natural logarithm of the distance between the surface's points and the camera centre. The fact that $\mathbf{J}_\theta = \theta\mathbf{J}_\mu$ is the fundamental property allowing us

to solve ConPSfT. This allows us to factor θ on the left-hand side of equation (12) as:

$$\mathbf{J}_\mu^\top \mathbf{J}_\mu \theta^2 + \xi\theta^2 \;=\; \nu \mathbf{J}_\Delta^\top \mathbf{J}_\Delta.$$

Once θ is factored, we can simply define a new variable ζ, combining the effect of the distance θ and function ν, essentially accounting for the conformal scale. Importantly, this means that in ConPSfT, the effect caused by the distance to the camera (translation along the line of sight of a surface path) and the effect caused by conformal rescaling cannot be distinguished. Defining $\zeta \stackrel{\text{def}}{=} \frac{\nu}{\theta^2} \in C^1(\mathcal{P}, \mathbb{R}^{+*})$, we arrive at:

$$\mathbf{J}_\mu^\top \mathbf{J}_\mu + \xi \;=\; \zeta \mathbf{J}_\Delta^\top \mathbf{J}_\Delta.$$

This is a first-order quadratic PDE in $\mathbf{J}_\mu$ and ζ. This equation indicates how the natural logarithm of the distance between the surface's points and the camera centre must vary under conformal deformations to comply with a given warp. It can be solved locally as it involves three unknowns and three PDEs. However, as we noted, only the combined effect of depth and conformal scaling can be recovered through function ζ. This suggests that the depth -and thus φ and Ψ- may not be directly recoverable, but require a further processing step of integrating the local estimates of $\mathbf{J}_\mu$ to first find μ. This of course introduces ambiguities in the solution.

ConPSfT – Formulation with the μ function

$$\text{Find} \quad \begin{cases} \mu \;\in\; C^1(\mathcal{P}, \mathbb{R}^{+*}) \\ \zeta \;\in\; C^1(\mathcal{P}, \mathbb{R}^{+*}) \end{cases} \qquad \text{s.t.} \qquad \mathbf{J}_\mu^\top \mathbf{J}_\mu + \xi \;=\; \zeta \mathbf{J}_\Delta^\top \mathbf{J}_\Delta \tag{15}$$

7 Conformal Solution

Equation (15) gives three constraints; the three unknowns in $\mathbf{J}_\mu$ and ζ can thus be solved for at each point independently. However, we show below that only the analog of the shape's normal function in the μ space may be recovered up to sign at this point, and that integration has to be carried out first to recover the deformed embedding φ.

7.1 Well-Posedness

Theorem 2 (Space of solutions for the ConPSfT problem)**.** *The ConPSfT problem (4) has a set of discrete solutions (at least two), each defined up to a global scale.*

Proof. We start from formulation (15) of the ConPSfT problem. For a point $\mathbf{p} \in \mathcal{P}$, we define $\mathbf{u} \stackrel{\text{def}}{=} \mathbf{J}_\mu^\top(\mathbf{p}) \in \mathbb{R}^{2\times 1}$, $\mathbf{G} \stackrel{\text{def}}{=} \mathbf{J}_\Delta^\top(\mathbf{p})\mathbf{J}_\Delta(\mathbf{p}) \in \mathbb{S}$, $z \stackrel{\text{def}}{=} \zeta(\mathbf{p}) > 0$ and $\mathbf{K} \stackrel{\text{def}}{=} \xi(\mathbf{p}) \in \mathbb{S}$. The PDE at point $\mathbf{p}$ is thus rewritten as the following

matrix equation:

$$-\mathbf{u}\mathbf{u}^\top + z\mathbf{G} \;=\; \mathbf{K}.$$

We call this equation a rank-1 equation of type 2. Proposition 2 in appendix D shows that there is always a single solution for z and two solutions for $\mathbf{u}$. Because $\mu \in C^1(\mathcal{P}, \mathbb{R})$, $\mathbf{J}_\mu \in C^0(\mathcal{P}, \mathbb{R}^{1\times 2})$. Therefore, each of the two entries of $\mathbf{J}_\mu$ may change sign only at those points where $\mathbf{J}_\mu = \mathbf{0}^\top$. In other words, the entries of $\mathbf{J}_\mu$ have a constant sign over connected components in the 2D parameterization separated by closed curves (or cutting the domain's boundary) defined by $\mathbf{J}_\mu = \mathbf{0}^\top$. The number b of such connected components depends on the deformed shape, and is usually small. The number of possible sign flips in the entries of $\mathbf{J}_\mu$, and thus of possible solutions to ConPSfT, is $n = 2^b \geq 2$. Each solution is defined up to an unknown global scale. This is because μ is obtained up to an unknown additive constant a, that carries over as a multiplicative factor to θ as $\exp(a+\mu) = \exp(a)\exp(\mu) = \exp(a)\theta$ and to γ as $\exp(a)\frac{\theta}{\varepsilon} = \exp(a)\gamma$. □

There are large differences in the number of solutions established in theorems 1 and 2, despite the strong similarities of the IsoPSfT and ConPSfT problems. These differences stem from the fact that in the conformal case, only the shape's gradient (transposed to some space with the two changes of variables) can be locally computed up to a two-fold ambiguity, because depth and conformal scale cannot be locally distinguished. In the isometric case however, while the same gradient may be computed, it is discarded since the depths (also transposed to some space with the first change of variable) can be locally computed uniquely.

7.2 Sign Ambiguities

Following the above discussion, sign ambiguities arise at the boundary of connected components where $\mathbf{J}_\mu = \mathbf{0}^\top$. These boundaries are curves made of *conformal singular points*. The surface's normal at these points passes at the camera centre, as we prove in appendix E. This is also equivalent to having $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$ and thus $\mathbf{A} \propto \mathrm{I}$ in the proof of proposition 2 in appendix D. In other words, this happens for $\mathbf{J}_\Delta^\top \mathbf{J}_\Delta \propto \xi$. It is then easy to show that conformal reparameterization does not change those singular points. Let $\tau \overset{\text{def}}{=} \|\mathbf{J}_\mu\|_2$, $\tau \in C^0(\mathcal{P}, \mathbb{R}^+)$. Connected components where the sign of both entries of $\mathbf{J}_\mu$ does not change simultaneously can thus be theoretically found using the fact that $\tau = 0$ at their boundaries and $\tau > 0$ in their interior.

7.3 Algorithm

Our algorithm has three key steps. It is summarized in table 2.

Finding $\mathbf{J}_\mu$ up to sign. This is a straightforward step, that we implement based on the solution to the rank-1 equation of type 2 given in proposition 2 in appendix D. In practice, we evaluate the solution independently at each

▷ Inputs: warp η, template embedding Δ, point set $\mathcal{B} \subset \mathcal{P}$

▷ Outputs: number of solutions n, deformed embeddings φ_i, $i = 1, \dots, n$ for points in $\mathcal{B}$

- **Find $\mathbf{J}_\mu$ up to sign**

  For $\mathbf{p} \in \mathcal{B}$

  Compute the Cholesky factor $\mathbf{V} \in \mathbb{R}^{2\times 2}$ such that $\mathbf{J}_\Delta^\top(\mathbf{p})\mathbf{J}_\Delta(\mathbf{p}) = \mathbf{V}\mathbf{V}^\top$

  Set $\mathbf{A} \leftarrow \frac{1}{\|\tilde{\eta}(\mathbf{p})\|_2^2}\mathbf{V}^{-1}\left(\mathbf{J}_\eta^\top(\mathbf{p})\mathbf{J}_\eta(\mathbf{p}) - \frac{1}{\|\tilde{\eta}(\mathbf{p})\|_2^2}\mathbf{J}_\eta^\top(\mathbf{p})\eta(\mathbf{p})\eta(\mathbf{p})^\top\mathbf{J}_\eta(\mathbf{p})\right)\mathbf{V}^{-\top}$

  Set $\mathbf{J}_\mu(\mathbf{p}) \leftarrow \sqrt{\lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})}\mathbf{V}\epsilon_2(\mathbf{A})$ (where $\epsilon_2(\mathbf{A})$ is sign-normalized)

- **Find connected components $\mathcal{C}_1, \dots, \mathcal{C}_b$**

  Instantiate function $\hat{\tau} \in C^2(\mathcal{P}, \mathbb{R}^+)$ by fitting a Thin-Plate Spline to $\{\|\mathbf{J}_\mu(\mathbf{p})\|_2\}$, $\mathbf{p} \in \mathcal{B}$

  Use $\text{sign}(\nabla_u\hat{\tau})$ and $\text{sign}(\nabla_v\hat{\tau})$ to label connected components in $\mathcal{U}(\mathbf{p})$ and $\mathcal{V}(\mathbf{p})$, $\mathbf{p} \in \mathcal{B}$

  Merge components whose boundary does not satisfy $\lambda_1\left(\frac{\partial^2\hat{\tau}}{\partial\mathbf{p}^2}\right) \geq 0$ in $\mathcal{U}$ and $\mathcal{V}$

  Blend $\mathcal{U}$ and $\mathcal{V}$ to $\mathcal{C}$ by keeping all possible components

  Suppress components in $\mathcal{C}$ whose support is smaller than 10 px

  Set $b \leftarrow \text{size}(\mathcal{C})$ and $n \leftarrow 2^b$

- **Integrate each connected component and combine them to form φ_i, $i = 1, \dots, n$**

  For $s = 1, \dots, b$

  Integrate $\mathbf{J}_\mu$ in the least-squares sense to find $\mu_s(\mathbf{p})$, $\mathbf{p} \in \mathcal{C}_s$

  For $\mathbf{p} \in \mathcal{C}_s$, Set $\theta_s(\mathbf{p}) \leftarrow \exp(\mu_s(\mathbf{p}))$

  For $i = 1, \dots, n$

  Set $\{\delta_1, \dots, \delta_b\} \leftarrow \text{bin}(i-1)$ (where bin forms the binary representation of an integer)

  Compute $d_2, \dots, d_b$ so that $\{d_1\theta_1^{\delta_1}\frac{1}{\varepsilon}\tilde{\eta}, \dots, d_b\theta_b^{\delta_b}\frac{1}{\varepsilon}\tilde{\eta}\}$ best agree in the least-squares sense, with $d_1 = 1$

  For $\mathbf{p} \in \mathcal{B}$, Set $\varphi_i(\mathbf{p}) \leftarrow d_s\theta_s^{\delta_s}(\mathbf{p})\frac{1}{\varepsilon(\mathbf{p})}\tilde{\eta}(\mathbf{p})$ for $\mathbf{p} \in \mathcal{C}_s$

Table 2: **Algorithm implementing our analytical solution to ConPSfT.** While $\mathbf{J}_\mu$ may be computed up to sign at each point independently, computing the embedding φ requires integrating over the whole surface. Here $\mathcal{B} \subset \mathcal{P}$ is a dense discretization for which we use the pixel grid in practice.

point of a dense set $\mathcal{B} \subset \mathcal{P}$. This solution involves computing the least eigenvector $\epsilon_2(\mathbf{A})$ of some matrix $\mathbf{A} \in \mathbb{S}$. The sign of this eigenvector is undetermined and we arbitrarily sign-normalize it by ensuring that its element with largest magnitude is positive, to avoid discontinuities.

Finding connected components $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_b\}$ with non-zero τ. This part of our algorithm is illustrated in figure 2. As we saw in §7.2, the theorerical criterion which characterizes the boundary between two neighboring components in $\mathcal{P}$ is $\tau = 0$, where τ is a known non-negative function. In practice this theoretical criterion is never exactly met due to noise. Our first attempt to overcome this issue has been to define a threshold $c \in \mathbb{R}^+$, with the hope that $\tau \leq c$ would allow us to detect the boundaries. In practice, this does not work: the criterion turns out to be extremely sensitive to the value of the threshold c, for which a typical value does not seem to exist. We now show how the theoretical criterion $\tau = 0$ can be relaxed to a practical criterion. The relaxation handles noise, does not require one to define a threshold, and guarantees that a superset of the true connected components is found. We first reconstruct a continuous version of τ called $\hat{\tau}$. For that purpose, we sample τ at a dense grid of points and form $\hat{\tau}$ as the Thin-Plate Spline interpolant (Duchon, 1976). The theoretical criterion simply becomes $\hat{\tau} = 0$. Because $\hat{\tau}$ is a continuous differentiable and non-negative function, $\hat{\tau} = 0$ corresponds to valleys of the $\hat{\tau}$ function, where $\hat{\tau}$ reaches zero perfectly. In the presence of noise, this is this last part of the criterion which is not met. We thus make a first step of relaxation by detecting all the valleys of the $\hat{\tau}$ function, including valleys for which $\hat{\tau} > 0$. Mathematically, we have that $\hat{\tau} = 0$ implies $\nabla\hat{\tau} = \mathbf{0}$, $\lambda_1(\mathcal{H}) \geq 0$ and $\lambda_2(\mathcal{H}) = 0$, where $\mathcal{H} \stackrel{\text{def}}{=} \frac{\partial^2 \hat{\tau}}{\partial \mathbf{p}^2}$. The first two conditions, $\nabla\hat{\tau} = \mathbf{0}$ and $\lambda_1(\mathcal{H}) \geq 0$, characterize valleys. The third condition, $\lambda_2(\mathcal{H}) = 0$, comes from the fact that the curvature of $\hat{\tau}$ must vanish along the valley representing the boundary. This, again, is a theoretical condition, which is not met in practice: it implies that $\hat{\tau}$ has a strictly constant value along the boundary which, in the presence of noise, does obviously not happen. Our second step of relaxation is to simply ignore this constraint, in order to avoid defining a threshold on the value of $\lambda_2(\mathcal{H})$. It means that the value of $\hat{\tau}$ may oscillate along the boundary, which is exactly what is expected to happen in the presence of noise.

In practice, our criterion for detecting the boundaries of connected components is thus $\nabla\hat{\tau} = \mathbf{0}$ and $\lambda_1(\mathcal{H}) \geq 0$. We implement it as follows. The condition $\nabla\hat{\tau} = \mathbf{0}$ is equivalent to $\nabla_u\hat{\tau} = 0$ and $\nabla_v\hat{\tau} = 0$. We use each of these two equations to define two sets of components, namely $\mathcal{U}$ and $\mathcal{V}$, using respectively $\text{sign}(\nabla_u\hat{\tau})$ and $\text{sign}(\nabla_v\hat{\tau})$ to define boundaries. We then merge regions in $\mathcal{U}$ and $\mathcal{V}$ whose boundary is a local maximum by checking $\lambda_1(\mathcal{H}) \leq 0$. Finally, we blend $\mathcal{U}$ and $\mathcal{V}$ in a single component set $\mathcal{C}$ by keeping all possible components. Components with small support are merged with their neighbours, and we set the number of components to $b \leftarrow \text{size}(\mathcal{C})$. This algorithm ensures that $\mathcal{P} = \cup_{s=1}^{b} \mathcal{C}_s$. In practice, we have found that it does a very good job at finding the connected components.

Integrating and combining connected components. Let $\delta_s \in \{-1; 1\}$ be a constant but unknown multiplicative factor of $\mathbf{J}_\mu$ on $\mathcal{C}_s$, representing the two-fold sign ambiguity. It is clear that because there are b connected

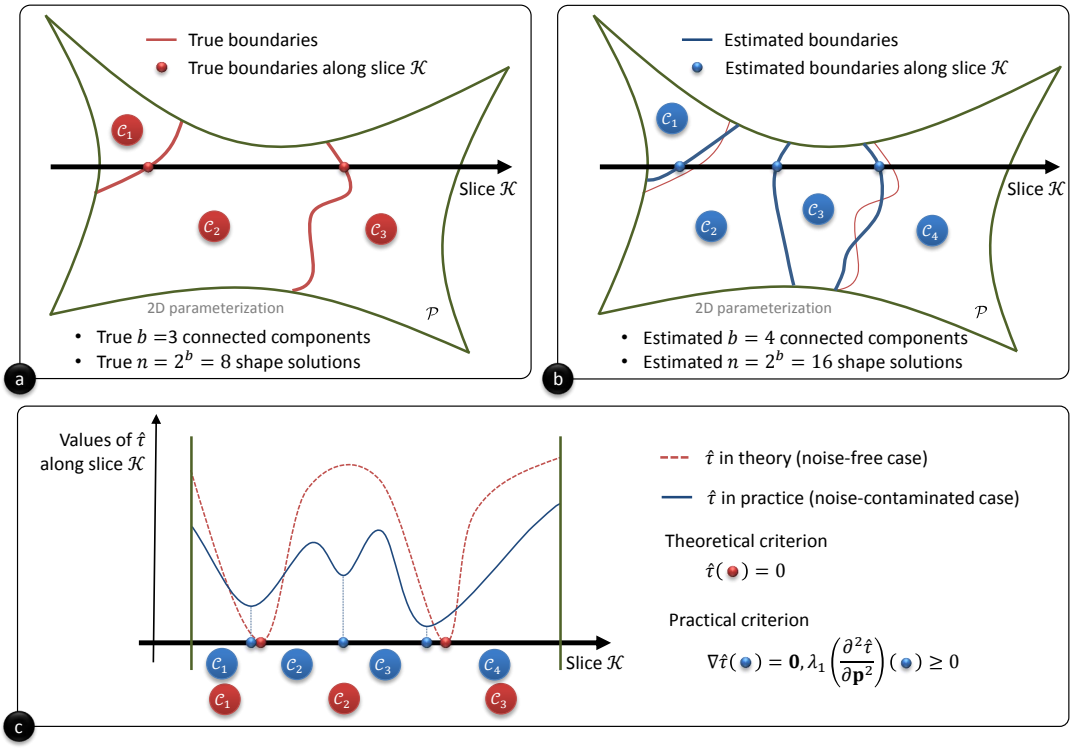


Figure 2: **Finding connected components in ConPSfT.** (a) shows a case where the 2D parameterization is theoretically divided up in 3 connected components, leading to 8 solutions. (b) shows an example of result on this case, where our practical criterion leads to the detection of 4 connected components, and thus to 16 solutions. Our algorithm based on this criterion guarantees that the 3 true connected components are, up to noise, included into the 4 detected ones. (c) uses a 1D slice of the 2D parameterization to illustrate the theoretical and practical criteria. We observe that the theoretical criterion $\hat{\tau} = 0$, which would apply well to a noise-free estimate of $\hat{\tau}$ (in dashed red), cannot be used on a noise-contaminated estimate of $\hat{\tau}$ (in full blue). It would not indeed find any of the connected components. Numerical tests also showed that one cannot find a constant threshold on the value of $\hat{\tau}$ to cope with noise. Our practical criterion $\nabla\hat{\tau} = \mathbf{0}$ and $\lambda_1\left(\frac{\partial^2\hat{\tau}}{\partial\mathbf{p}^2}\right) \geq 0$ is a relaxation of the theoretical criterion, which does not involve setting a threshold value, and still applies well to noise-contamined data. Our experimental results show the number of detected connected components is always small, being typically $b = 3$.

components this leaves $n = 2^b$ possible combinations and as many solutions to φ. A straightforward way to compute all solutions would be to loop through all of the n possible sign combinations, and solve n times for φ. Solving for φ from $\mathbf{J}_\mu$ requires one to first integrate $\mathbf{J}_\mu$ to find μ and then undo the two changes of variables. A computationally much faster strategy is to first solve for $\bar{\varphi}_s$, defined as the restriction of φ to component $\mathcal{C}_s \subset \mathcal{P}$, for $s = 1, \ldots, b$, and combine these component-wise estimates a posteriori. The cost of this is basically equivalent to the cost of solving for φ for only one sign configuration, and is therefore n times faster than the straightforward approach. This is because assembling the component-wise estimates for some sign configuration requires one to solve for only $b - 1$ scalars through simple linear least-squares. This step is illustrated in figure 3, and explained in detail below.

Let μ_s be the restriction of μ to component $\mathcal{C}_s \subset \mathcal{P}$. We obtain μ_s up to an additive constant $c_s \in \mathbb{R}$ by integrating $\mathbf{J}_\mu$ on $\mathcal{C}_s$ in the least-squares sense, by solving $\arg\min_{\mu_s} \int_{\mathcal{C}_s} \|\mathbf{J}_{\mu_s} - \mathbf{J}_\mu\|_2^2 \, \mathrm{d}\mathbf{p}$. This is the most expensive part of the computation, which is solved only once for each component. We thus end up with function $\delta_s \mu_s + c_s$ on component $\mathcal{C}_s$. Using the second change of variable (14) we define $\theta_s \stackrel{\text{def}}{=} \exp(\mu_s)$ and $d_s \stackrel{\text{def}}{=} \exp(c_s)$, which combine as $d_s \theta_s^{\delta_s}$. Using the first change of variable (11) and then the reprojection constraint (6), we arrive at $\bar{\varphi}_s = d_s \theta_s^{\delta_s} \frac{1}{\varepsilon} \tilde{\eta}$. We fix the arbitrary global scale by setting $d_1 = 1$. For each of the n possible sign configurations we then find the scales $d_2, \ldots, d_b$ which maximize the agreement of the component-wise embeddings at the components' boundaries. In practice, we solve the following linear least-squares problem:

$$\arg \min_{d_2,\ldots,d_b} \sum_{s=1}^{b} \sum_{\substack{s'=1 \\ s' \neq s}}^{b} \int_{\mathcal{C}_s \cap \mathcal{C}_{s'}} \|d_s \bar{\varphi}_s - d_{s'} \bar{\varphi}_{s'}\|_2^2 \mathrm{d}\mathbf{p} \qquad \text{with} \qquad d_1 = 1.$$

This part is computationally cheap since b is usually small. It is solved n times to compute the n possible solutions $\varphi_1, \ldots, \varphi_n$ for all combinations of the signs $\delta_1, \ldots, \delta_b$.

8 Experimental Results

We here compare our algorithms with existing zeroth-order algorithms. We use simulated and real data.

8.1 Compared Algorithms

The compared algorithms are listed below. The first batch of algorithms are zeroth-order convex initialization algorithms based on the inextensibility relaxation and the maximum depth heuristic:

- **PeIso** an iterative solution tightening upper bounds on depth from point pairs (Perriollat et al., 2011)
- **SaIso** a convex solution (Salzmann and Fua, 2009)
- **BrIso** an SOCP solution (Brunet et al., 2010)

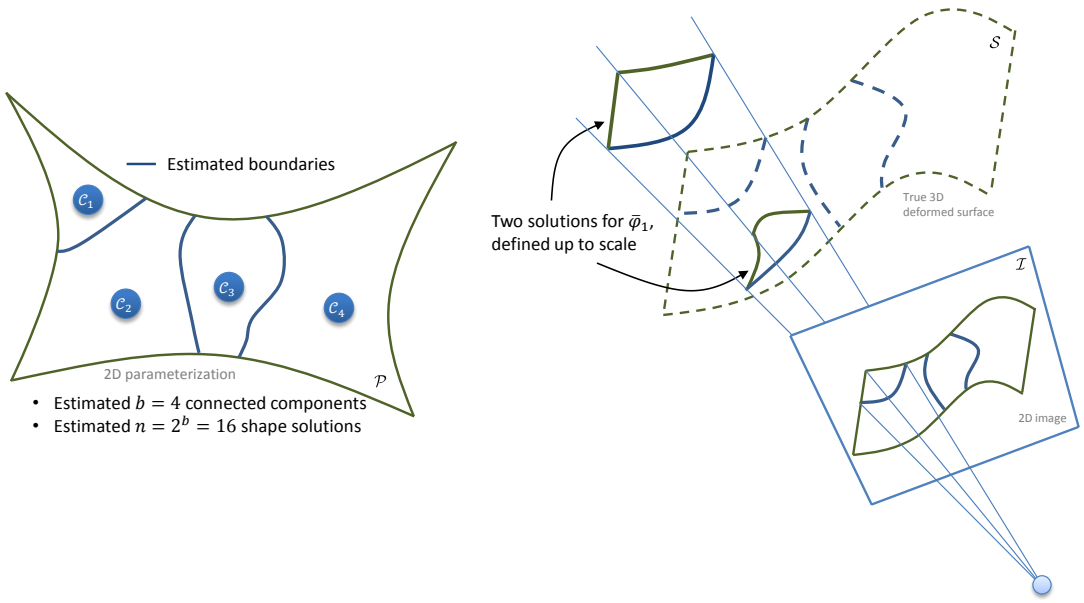


Figure 3: **Combining the connected components in ConPSfT.** The left part shows an example of 4 estimated connected components. The right part illustrates, on component $\mathcal{C}_1$, that after integration, the restriction $\bar{\varphi}_1$ of the embedding function φ to $\mathcal{C}_1$, is recovered up to a two-fold ambiguity and up to scale. The 16 up to scale solutions for the embedding φ are obtained by forming all possible configurations of two-fold component-wise ambiguities, and computing the best scale for each component so that the recovered embeddings agree as best possible along the components' boundaries, here the blue curves between $\mathcal{C}_1/\mathcal{C}_2$, $\mathcal{C}_2/\mathcal{C}_3$, and $\mathcal{C}_3/\mathcal{C}_4$.

The second batch of algorithms are the proposed first-order analytical initialization algorithms:

- **AnIso** the isometric solution with algorithm in table 1
- **AnCon** the conformal solution with algorithm in table 2

Finally, we have a gold standard, the non-analytical and non-convex solution minimizing reprojection error of a smooth surface:

- **ReIso** isometric refinement (Brunet et al., 2010)
- **ReCon** an extension of **ReIso** to conformal refinement

8.2 Simulated Data

Simulation setup. We simulate a calibrated pin-hole camera with intrinsic parameters $\mathrm{diag}(500, 500, 1)$ (where 500 is thus the focal length in px) with VGA image resolution (640×480 px). This camera observes a deformable surface whose template has size 100×100 mm. We randomly draw m points on the surface and create their image, with default value $m = 50$, to which we add centred Gaussian noise with variance σ px, with default value $\sigma = 1$ px. The simulated surface is located approximately 1 m away from the camera. We investigate two kinds of deformations: isometric and conformal. Examples are shown in figure 4. In the former case we use a paper model (Perriollat and

Bartoli, 2013) to create an isometric embedding of the data points lying on the 2D square template. In the latter case we apply a 2D conformal deformation of the template prior to an isometric embedding. This deformation is generated by changing the template's rectangular boundary with conformal interpolation inside it. The target boundary is obtained by blending the original template boundary with a circle. The blending is controlled by a parameter $w \in \mathbb{R}$ with $0 \leq w \leq 1$ which allows us to choose the amount of conformity in the generated surface: $w = 0$ creates an isometric deformation while $w = 1$ creates a highly conformal deformation. The default value is chosen as $w = 0.6$. For our methods **AnIso** and **AnCon**, which require a warp as input, we fit a Thin-Plate Spline warp using a standard method (Bookstein, 1989). For all experiments, we measure the 3D error in mm as the average distance between the simulated and the reconstructed 3D points. For each configuration we keep the average over 100 trials. For conformal surfaces, because the absolute scale cannot be reconstructed, we proceed to a best-fit least-squares scale adjustment before computing the 3D error. We use the lowest 3D error over the multiple solutions returned by **AnCon**. To keep the comparison fair, we also proceed to this scale adjustment on the results of the isometric algorithms, when used on conformal surfaces.

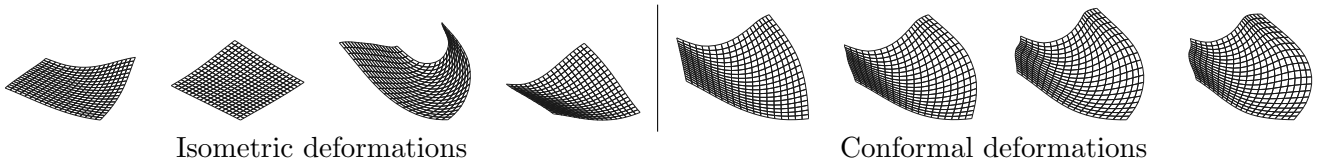


Figure 4: **Examples of simulated surfaces.** The four isometric deformation examples are all different, while the conformal ones illustrate the effect of the parameter w controlling the amount of conformity, with s chosen, from left to right, as 0, 0.25, 0.5 and 0.75.

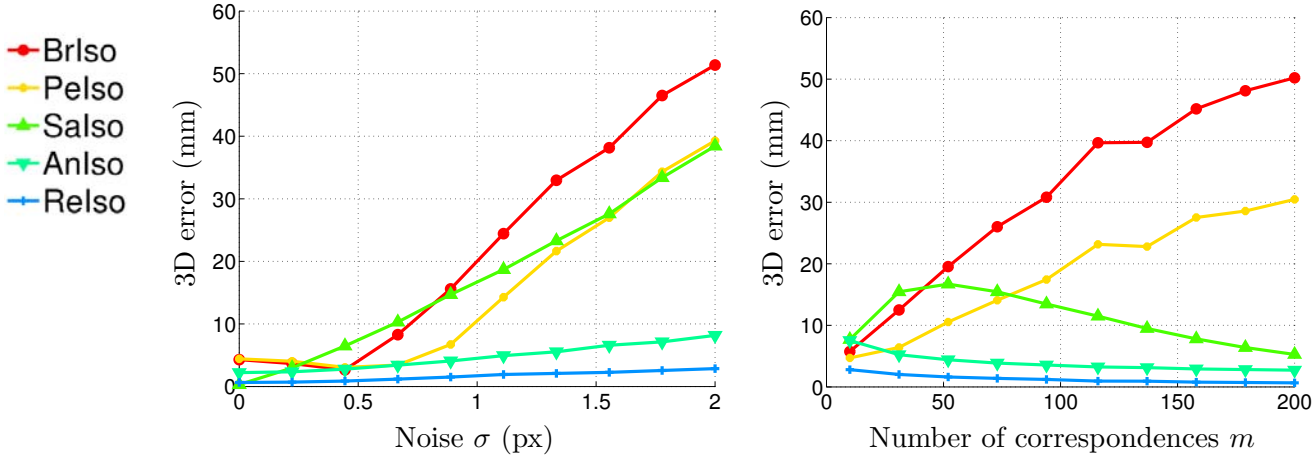


Figure 5: **Experimental results using simulated isometric deformations – noise and number of point correspondences.** The 3D error in mm is shown as a function of noise on the point correspondences and of the number of point correspondences for simulated isometric deformations and isometric SfT methods.

Isometric deformations. Results for isometric deformations and isometric methods are shown in figure 5. The accuracy of all methods degrades when noise increases. In the absence of noise, **SaIso** rivals **ReIso**, both achieving a very small error. The other methods' accurary is within 5 mm. For a moderate amount of noise, namely less than 0.25 px, all methods provide an accurate shape, with 3D errors lower than 5 mm, and **SaIso** quickly degrading. When noise increases beyond 0.5 px, the accuracy of **BrIso** significantly degrades. The same phenomenon can be observed for **PeIso** when noise approaches 1 px. Eventually, **BrIso**, **SaIso** and **PeIso** end up with large errors for a 2 px noise, of several dozens of mm. We observe than **AnIso** does not have a breakdown in accuracy as noise increases, simply degrading linearly. It ends up with a 3D error of less than 10 mm for 2 px noise. It is clear that **ReIso** gives a significant improvement to its initial solution, showing the same behaviour as **AnIso**, but reaching a 3D error of less than 5 mm for 2 px noise. The accuracy of **BrIso** and **PeIso** decreases when the number of correspondences increases. This is due to their use of a neighborhood including a fixed number of points to express the inextensibility constraints. When the point density increases, the neighborhood size shrinks, thereby weakening the constraints. The accuracy of **SaIso** degrades up to about 50 point correspondences, peaking at a 3D error of about 18 mm, and then steadily improves, with a 3D error smaller than 10 mm for more than about 130 point correspondences. Finally, we observe that the accuracy of **AnIso** and **ReIso** is consistently improved by increasing the number of point correspondences. Both methods seem to stabilize at about 4 mm and 1 mm in 3D error, respectively.

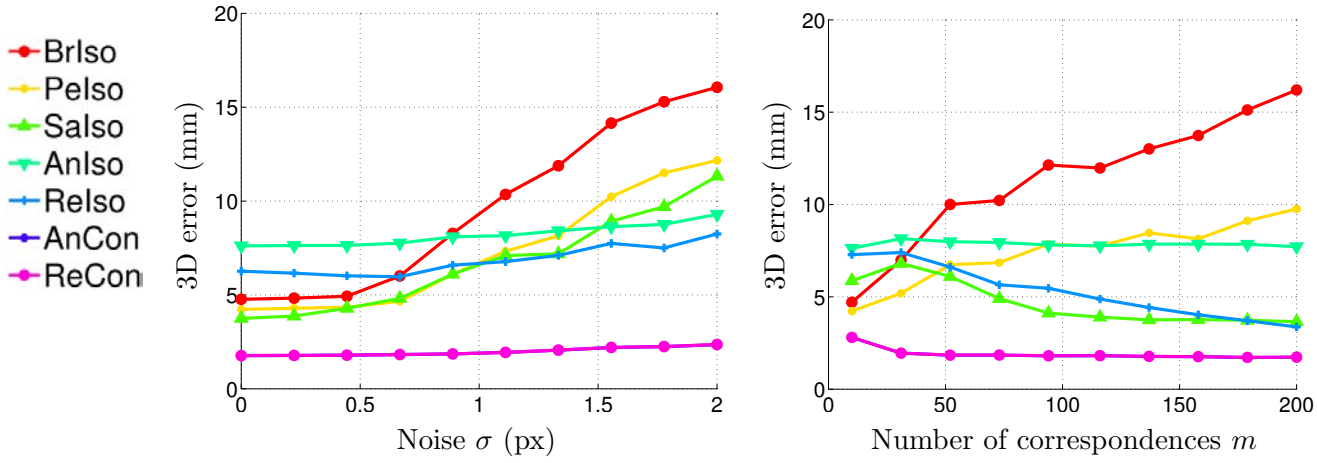


Figure 6: **Experimental results using simulated conformal deformations – noise and number of point correspondences.** The 3D error in mm is shown as a function of noise on the point correspondences and of the number of point correspondences for simulated conformal deformations and isometric and conformal SfT methods.

Conformal deformations. Results for conformal deformations and isometric and conformal methods are shown in figure 6. As expected, conformal methods perform much better than isometric methods. The accuracy of all methods degrades when noise increases. For less than 1 px noise, **BrIso**, **PeIso** and **SaIso** are more accurate than **AnIso** and **ReIso**. The first three however quickly degrade beyond 1 px noise while the last two degrade steadily and linearly. Because the simulated surface is here not isometric, relaxing the isometric constraint as **BrIso**, **PeIso** and **SaIso** do

performs better than enforcing it more strongly, as **AnIso** and **ReIso** do. However, as noise increases, the isometric constraint becomes useful to resolve the surface, causing the crossover between the two groups of methods. The two conformal methods **AnCon** and **ReCon** are indistinguishable. Their 3D error is around 2 mm for zero noise, and linearly increases to 3 mm for 2 px noise. When the number of point correspondences increases, we observe the same behaviour as in the isometric case, with the difference that **SaIso** performs slightly better than **ReIso**, stabilizing at a 4 mm 3D error. Conformal methods still perform very well, and stabilize at a 3 mm 3D error beyond 30 point correspondences.

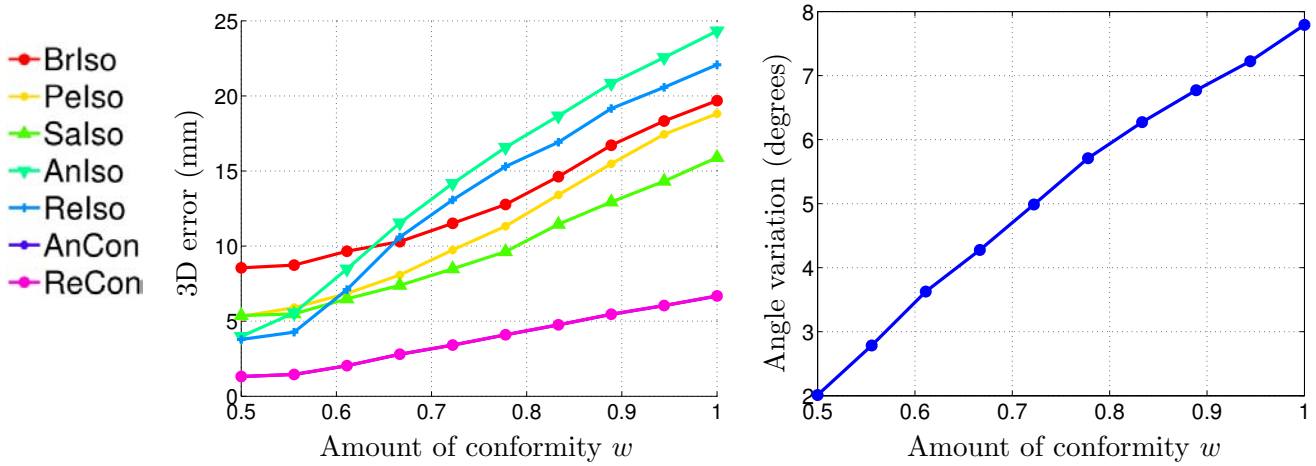


Figure 7: **Experimental results using simulated conformal deformations – amount of conformity.** (left) 3D error in mm is shown as a function of the amount of conformity for simulated conformal surfaces and isometric and conformal SfT methods. (right) Non-conformity due to surface discretization is shown as the average angle variation (in degrees) of the simulated surfaces as a function of conformity.

More results for conformal deformations and isometric and conformal methods are shown in figure 7. The accuracy of all methods degrade with the amount of conformity. While **AnIso** and **ReIso** perform better than the other isometric methods, they degrade quicker. Again, this is explained by the fact that they enforce isometry as opposed to relaxing it as **BrIso**, **PeIso** and **SaIso** do. Conformal methods perform much better than isometric methods. However, their accuracy linearly degrades with the amount of conformity. This may seem counterintuitive, and the reason has to be found in the discrete mesh used to simulate the surface. Because of discretization, conformity is only approximated, and the larger the amount of conformity, the stronger the approximation. We quantify this effect by computing the average angle difference in degrees between the original and conformally deformed meshes. This can be interpreted as a deviation from conformity, and is clearly visible in figure 7.

Finally, we monitor the number of solutions returned by **AnCon**, as a function of the number of correspondences and noise level, and give its average and standard deviation in the next table:

Number of correspondences m	Noise σ (px) 0	0.5	1	1.5	2
10	6.68 ± 3.13	7.38 ± 3.26	8.30 ± 5.69	8.22 ± 4.87	8.72 ± 5.39
50	8.08 ± 1.26	8.16 ± 1.12	8.44 ± 1.95	9.12 ± 3.10	10.44 ± 5.77
100	8.00 ± 0.00	8.00 ± 0.00	8.16 ± 1.12	8.92 ± 3.29	8.60 ± 2.22
200	8.00 ± 0.00	8.00 ± 0.00	8.08 ± 0.80	8.08 ± 0.80	8.48 ± 1.90

We observe that the typical number of solutions found is $n = 8$, meaning that the typical number of connected components is $b = 3$. Logically, the standard deviation increases with noise and decreases with the number of correspondences.

Influence of the warp's regularization weight. Our methods **AnIso** and **AnCon** rely on a warp -here a Thin-Plate Spline- computed to smoothly interpolate the point correspondences. It is standard in warp computation to balance a data term and a regularizer (Bookstein, 1989). The former measures how well the warp interpolates the point correspondences using the transfer error (the sum of squared distances between the points observed in the image, and the points transferred from the template). The regularizer measures how smooth is the warp using the integral of the Frobenius norm of its second derivatives. Both terms are linear least-squares and can be easily minimized. However, their combination is a tradeoff, governed by the regularization weight, a positive factor applied to the regularizer. Choosing the regularization weight too low causes overfitting, while choosing it too high causes underfitting. Because the warp is the main input to our analytical methods, we monitore the influence of the regularization weight on the reconstruction 3D error for the default isometric and conformal simulation setups. Our results are shown in figure 8. We observe that the empirically optimal regularization weight is 1, for both **AnIso** on isometric deformations and **AnCon** on conformal deformations. A small improvement is observed for **AnIso** on conformal deformations by choosing 10^3 for the regularization weight. More importantly, we observe that there is a substantial tolerance in choosing the regularization weight: **AnIso** and **AnCon** are both stable and give close to optimal results for a regularization weight chosen between 10^{-2} and 10^4. In practice we use the value of 1 in all our experiments.

Computation time. We give statistics on the computation time required by the different methods for a varying number of points in table 3. These times are measured with Matlab implementations, running on an Intel Core i7-3770 CPU at 3.40GHz. We can clearly see that the two refinement algorithms, **ReIso** and **ReCon**, are much slower that all the other algorithms, taking on average between 20 and 25 seconds per run, independently of the number of points. This is due to the fact that they model the surface densely, so as to globally enforce the isometric and conformal deformation constraints, respectively. In the initialization algorithms, **SaIso** and our conformal algorithm **AnCon** reach the same order of running time, approximately 1.5 seconds, and are quite independent of the number

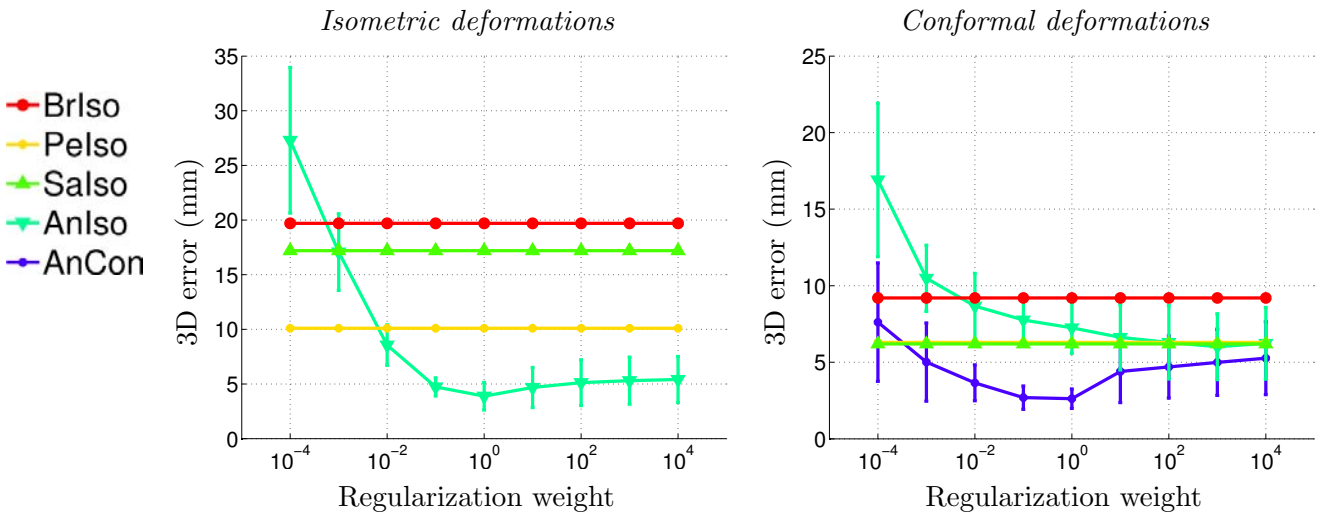


Figure 8: **Experimental results using simulated isometric and conformal deformations – regularization weight.** 3D error in mm is shown as a function of the regularization weight for simulated isometric and conformal deformations. The vertical bars represent the standard deviation.

of points. There is then a group of two isometric initialization algorithms, **BrIso** and **PeIso**, with run times of 0.25 to 1.83 seconds for the former, and 0.14 to 0.22 seconds for the latter, both increasing with the number of points. Finally, the fastest algorithm is **AnIso**. We add the computation time required to compute the input warp, so as to make the comparison fair with the other algorithms, requiring, to the exception of **AnCon**, only point correspondences. This results in run times between 1.2×10^{-3} to 1.5×10^{-2} seconds on average, and increasing with the number of points. Our algorithm **AnIso** is thus much faster than the other isometric initialization algorithms, namely, two orders of magnitude faster than **BrIso** and **PeIso**, and between two and three orders of magnitude faster than **SaIso**.

Average computation time (ms)

	Number of points			
Method	10	50	100	200
BrIso	245.34	462.71	707.05	1834.16
PeIso	145.80	181.97	195.03	224.51
SaIso	1426.19	1539.45	1412.17	1573.00
AnIso	0.26	1.49	2.90	5.14
ReIso	21895.45	24424.24	24346.94	22208.08
AnCon	1525.79	1775.88	1736.10	1439.95
ReCon	19820.33	22207.36	22118.35	19927.76
Warp	0.96	2.60	4.89	10.07

Table 3: **Computation time.** Average computation time for all methods for a varying number of points, in ms. The computation time of warp estimation from the point correspondences is also given on the last row.

8.3 Real Data

We show algorithm comparison results on three datasets. Each dataset is different of the others in terms of the observed surface characteristics, and it thus important on its own. For each dataset, keypoint correspondences are established using a specific method. In all cases, a Thin-Plate Spline warp is then fitted (Bookstein, 1989).

8.3.1 CVLab's Paper Dataset

This dataset is provided by the CVLab (Varol et al., 2012). It is captured with the Kinect sensor, providing some groundtruth measurement of the shape for every frame, and allowing one to monitor 3D error. It shows a piece of paper being gently bent in front of the camera without occlusions. Results are shown in figure 9.

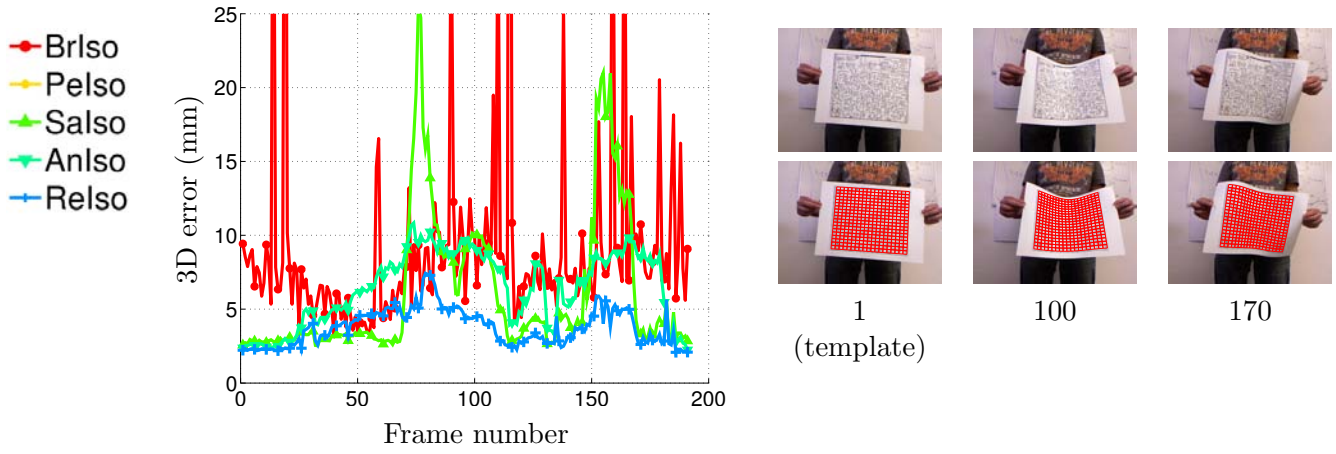


Figure 9: **Experimental results using real isometric deformations – CVLab's paper dataset.** (left) 3D error in mm with respect to groundtruth is shown for all frames for isometric SfT methods. Note that the error is off the y axis for **PeIso**. (right) The original images and the estimated warp.

The template is defined from the first frame of the dataset, where the surface is shown approximately planar. Note that because the surface's groundtruth shape is provided, we could use any other frame and build a 3D template for our algorithms, though this would not allow us to run **PeIso** and **BrIso**, which both require a flat template. We clean the point correspondences provided with the images which contain false correspondences with our robust warp estimation method (Pizarro and Bartoli, 2012). On average, 1026 point correspondences per image are kept. The 3D error is computed as the average distance between these points on the groundtruth shape and their reconstructed positions. The following table gives 3D error statistics extracted from all frames for the paper dataset:

Method	**BrIso**	**PeIso**	**SaIso**	**AnIso**	**ReIso**
Average 3D error (mm)	9.29	92.48	**6.17**	6.47	3.79
Standard deviation (mm)	8.50	51.70	4.95	**2.44**	1.22

We observe that **PeIso** fails on every images of this dataset. **BrIso** gives results which reach the same accuracy as **SaIso**, **AnIso** and **ReIso** though it fails on some images, degrading its average 3D error and its general reliability.

The three other isometric methods –**SaIso**, **AnIso** and **ReIso**– all give sensible results. We observe that while the average 3D errors for **SaIso** and **AnIso** are very similar, and slightly better for **SaIso**, **AnIso** has approximately half the standard deviation of **SaIso**, and is thus the best initialization method. This is also clearly visible on the corresponding 3D error graph. Finally, we observe that **ReIso** has a significantly lower 3D error, and approximately half the standard deviation of **AnIso**.

8.3.2 The Cushion Dataset

We propose this new dataset which uses a 3D template of an object undergoing close to isometric deformations. We build the 3D template using dense SfM from multiple images of a cushion shown in figure 10. Specifically, we use the commercial software Photoscan. We keep both the reconstructed 3D template and one of the input images so as to be able to easily match a deformed input image against the template. We repeat this procedure for three

Figure 10: **Experimental results using real isometric deformations – the cushion dataset, construction of the template.** The left part shows some of the 20 images we use to reconstruct the 3D template shown on the right part with SfM.

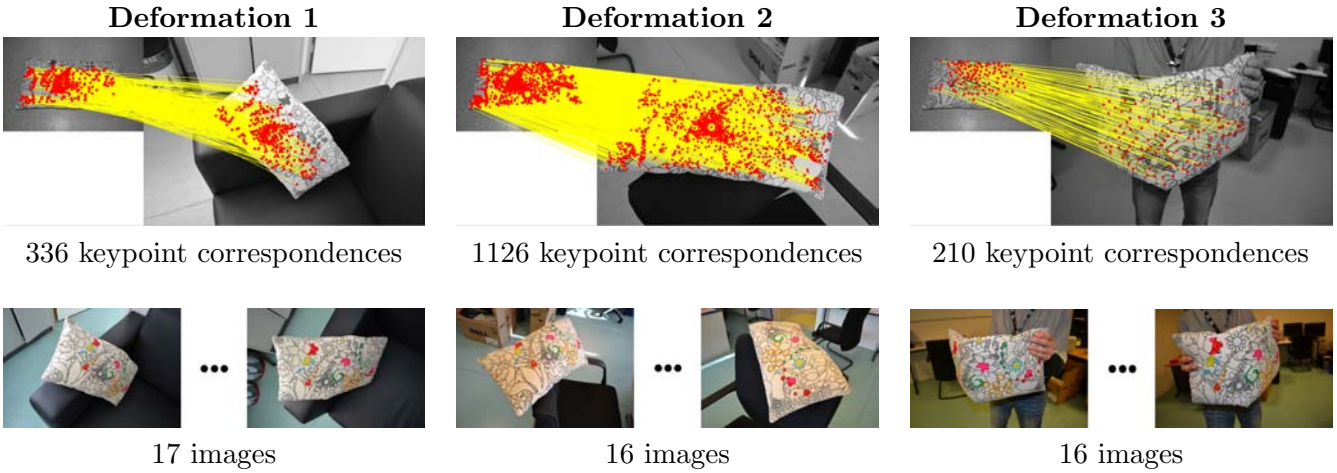


Figure 11: **Experimental results using real isometric deformations – the cushion dataset, input images and correspondences.** (top) A representative image for each of the three deformations is shown, overlaid with the automatically established keypoint correspondences to a template image, used as the flattened template. (bottom) Each deformation is photographed from 17, 16 and 16 viewpoints respectively, giving a total of 49 differents images.

sets of isometric deformations of the cushion. This way, we obtain groundtruth using dense SfM, and test isometric

SfT procedures using only one of the input frames while being able to measure 3D error. Concretely, a total of 49 images is available, and we use all of them in turn to test SfT and to be able to report statistical results. Some selected of these inputs images are shown in figure 11. We establish point correspondences automatically, using SIFT (Lowe, 2004) followed by our robust correspondence filtering method (Pizarro and Bartoli, 2012). These are shown in figure 11 for three selected images. The computed image warps are shown in figure 12 for the same three images. We then proceed to apply the reconstruction algorithms. For our methods, the template image is here used as a flattening of the template. This is possible since our algorithm can use any type of flattening, and not only conformal flattening. Each or several of the images used to reconstruct the template by dense SfM, and thus having a depth associated to each pixel, can therefore be used as a flattened template, provided they together cover all parts of the object's surface. The 3D error is computed as the average distance between the groundtruth position of these point correspondences previously obtained by dense SfM, and their reconstructed positions. We run three methods on this dataset: **SaIso**, **AnIso** and **ReIso**. The other methods, **PeIso** and **BrIso**, only work for a flat template, and could thus not be applied on the cushion dataset.

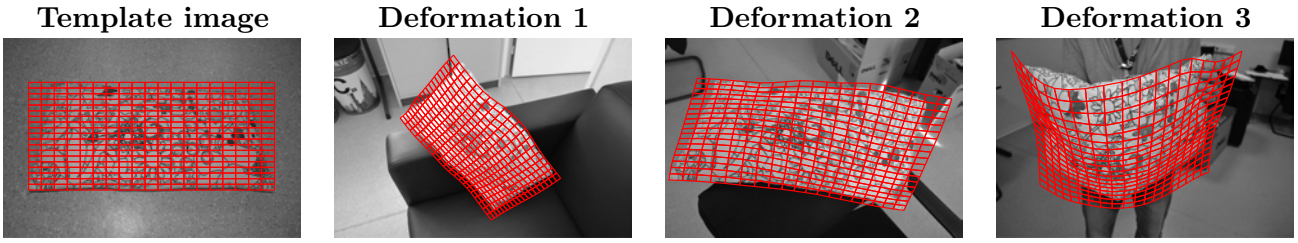


Figure 12: **Experimental results using real isometric deformations – the cushion dataset, the computed image warps.** (right) A representative image for each of the three deformations is shown overlaid with a grid representing the warp to the template image (left) computed from keypoint correspondences. For deformation 3, there are no correspondences at the bottom of the cushion. In this area, the warp is thus controlled by the smoothing term in the warp estimation method (Bookstein, 1989).

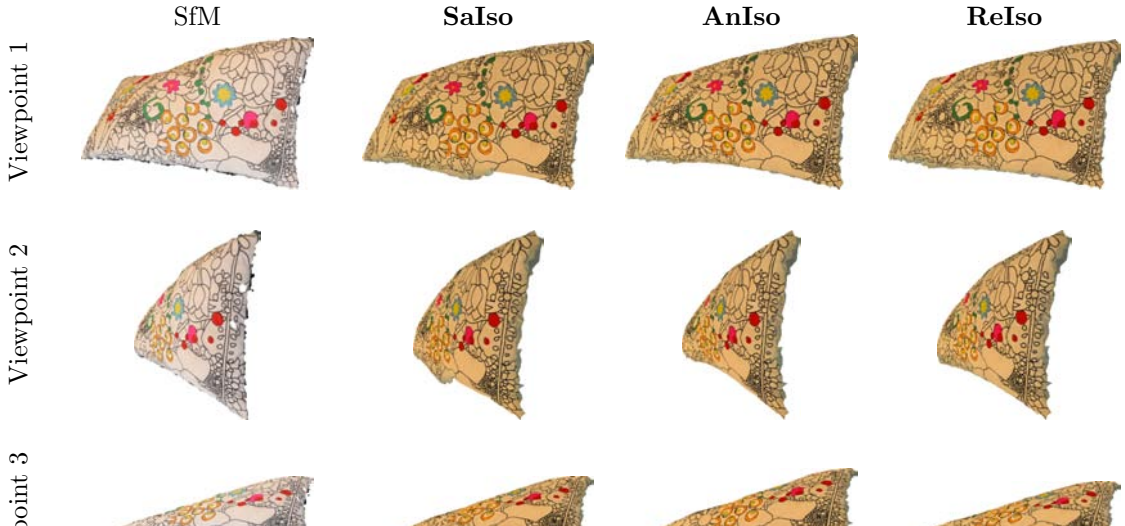
Qualitative and quantitative results are shown in figures 13, 14 and 15, for the three deformations and the three compared methods, as well as groundtruth. The 3D models' snapshots are generated from the same set of viewpoints to allow one to easily assess their visual similarity. The visual difference between groundtruth and the three compared methods is very small, and indistinguishable in some cases and for some viewpoints. Quantitatively, we observe that **SaIso** and **AnIso** give very similar results, **SaIso** being more accurate for Deformation 1, and less accurate for Deformation 2 and Deformation 3. We also observe that **ReIso** gives a significant reduction of the 3D error.

8.3.3 The Balloon Dataset

We propose this new dataset which uses a 3D template of an object undergoing close to conformal deformations. We build the 3D template by fitting a sphere to a set of points obtained using SfM from multiple images of a balloon shown in figure 16. We then gently deform the balloon by pressing onto it, and track features using KLT (Shi and

Average 3D error ...	11.3 mm	17.1 mm	6.6 mm
Standard deviation ...	2.3 mm	3.0 mm	1.3 mm

Figure 13: **Experimental results using real isometric deformations – the cushion dataset, the reconstructed 3D models for Deformation 1.** The 3D error is computed with respect to SfM, which is here the groundtruth, for 17 images.



Average 3D error ...	13.9 mm	10.0 mm	4.0 mm
Standard deviation ...	2.8 mm	1.9 mm	1.1 mm

Figure 14: **Experimental results using real isometric deformations – the cushion dataset, the reconstructed 3D models for Deformation 2.** The 3D error is computed with respect to SfM, which is here the groundtruth, for 16 images.

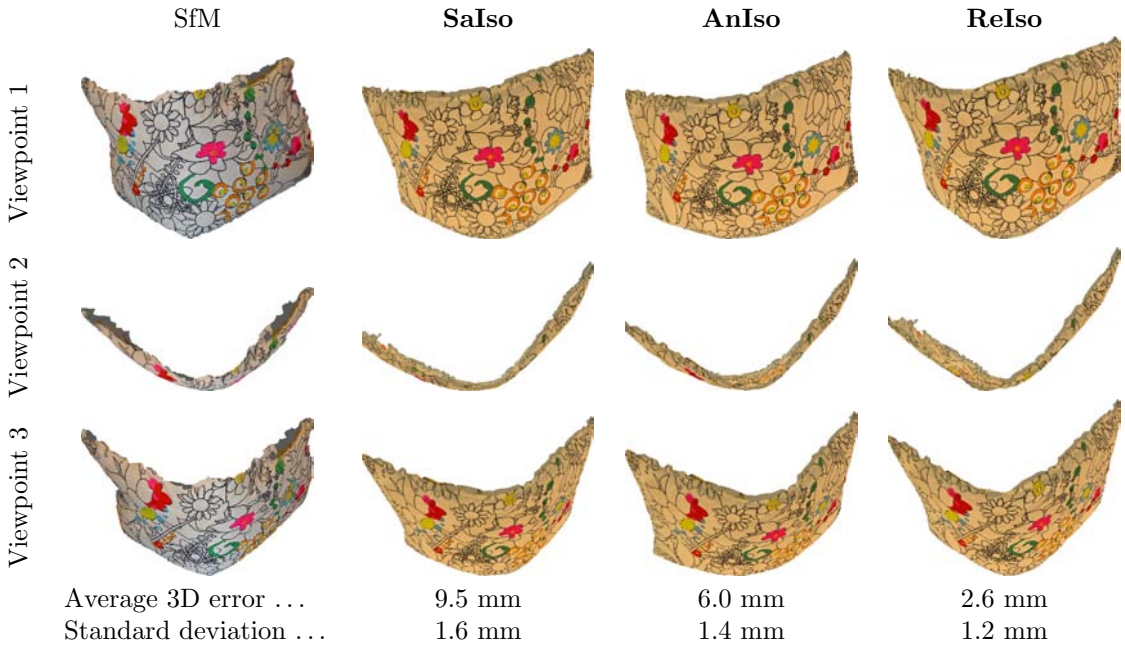


Figure 15: **Experimental results using real isometric deformations – the cushion dataset, the reconstructed 3D models for Deformation 3.** The 3D error is computed with respect to SfM, which is here the groundtruth, for 16 images.

Figure 16: **Experimental results using real conformal deformations – the balloon dataset, construction of the template.** The left part shows some of the 8 images we used to reconstruct the 3D template shown on the right part with SfM.

Tomasi, 1994) through the deformation video. This way, 262 points are successfully tracked. We finally hold the balloon still, and compute groundtruth for the end of the video using dense SfM. The point correspondences and the estimated warp are shown in figure 17. We run five methods on this dataset: the two conformal ones **AnCon** and **ReCon**, and the three isometric ones which handle a 3D template, **SaIso**, **AnIso** and **ReIso**. The 3D error is computed as the average distance to the point correspondences' groundtruth position previously obtained by dense SfM. The 3D error computation is combined with least-squares recovery of the 3D reconstruction's scale. This is because in conformal deformations the scale is not recoverable. The conformal method returns eight solutions; we keep the one closest to groundtruth.

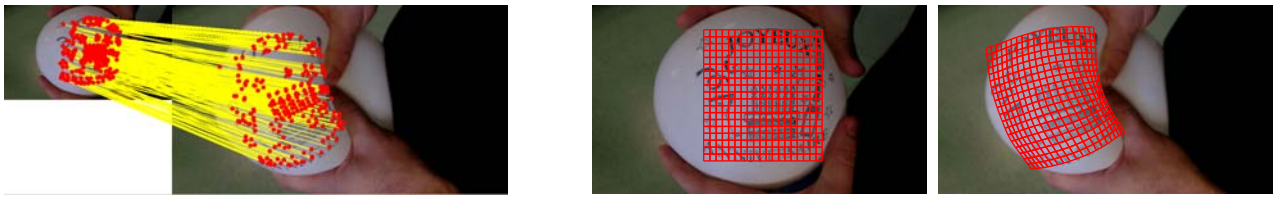


Figure 17: **Experimental results using real conformal deformations – the balloon dataset, the input image.** (left) The input image is shown overlaid with automatically established point correspondences to the template image, used as the flattened template. (right) The corresponding image warp, represented by a grid.

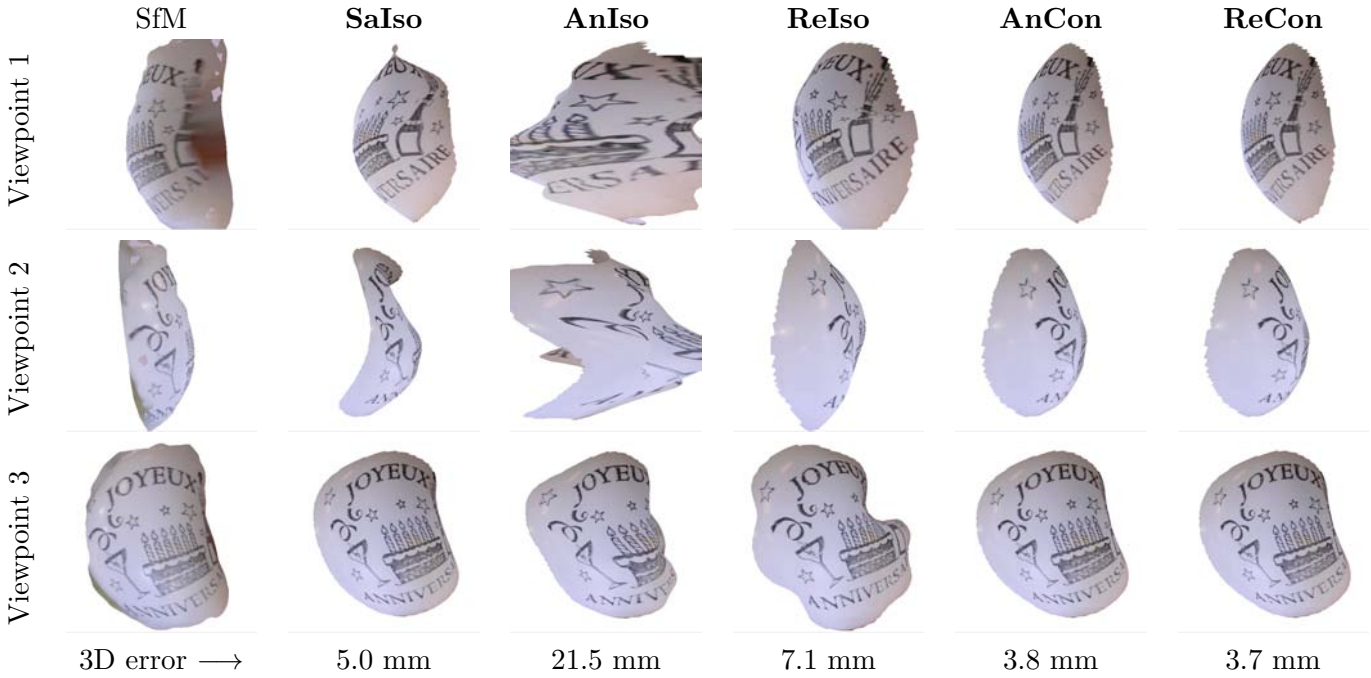


Figure 18: **Experimental results using real conformal deformations – the balloon dataset, the reconstructed 3D models.** The 3D error is computed with respect to SfM, which is here the groundtruth.

Qualitative and quantitative results are shown in figure 18 for the five compared methods, as well as groundtruth. As for the cushion, the snapshots are generated from the same set of viewpoints. There are important discrepancies between the different methods, both visually and in terms of 3D error. It is visually clear that isometric methods fail,

while both conformal methods give a satisfying result. **SaIso** reconstructs a large part of the balloon well, which keep the average 3D error to 5 mm, though it fails on the top and right hand side parts, as can be seen from viewpoints 1 and 2 for the former, and viewpoint 3 for the latter. **AnIso** largely fails, with a 3D error of 21.5 mm, because of the isometric local modeling assumption being largely violated. **ReIso** also fails to recover a satisfying shape, and, as expected, gives a surface that looks like a cloth sheet wrapped around a solid body, reflecting the fact that it enforces isometry to a very good extent. **AnCon** and **ReCon** give visually indistinguishable results, with a 3D error of 3.8 mm and 3.7 mm respectively. Both shapes are visually very close to groundtruth.

9 Discussion and Conclusion

We have proposed a novel class of methods to tackle the Shape-from-Template problem. They are called first-order methods since they exploit the first-order differential structure between the template and the input image. This allowed us to derive proofs of existence and analytical solutions to the problem, while state of the art focused on zeroth-order relaxations. We showed that for an isometric deformation, depth can be obtained pointwise, while for a conformal deformation only some shape gradient in a transformed space can be obtained locally, and leads to several discrete shape solutions after integration.

From the Perspective SfT PDE (9) it seems clear that one may be able to derive constraints on the warp rather than on the 3D shape. This may be achieved by plugging the analytical solutions back into this equation. This should lead to some second-order quartic polynomial PDE on the warp giving two constraints per point in the isometric case, and one constraint per point in the conformal case. While this could be used to improve warp estimation by including 3D consistency earlier in the process, this also tells us that there is a spare constraint which we have not yet made use of, and which may be used to extend the framework to more flexible deformations including local shearing and anisotropy, and thus to general first-order differential deformation priors. The properties, existence of solutions, ambiguities and algorithms for each possibility will require a dedicated and careful study.

In terms of accuracy, the proposed isometric method competes with the previous best (Salzmann and Fua, 2009) and outperforms others (Brunet et al., 2010; Perriollat et al., 2011). It is though the lightest method in terms of computation, simply requiring one to apply an analytical formula at each point. Our experimental results show that our isometric algorithm runs several orders of magnitude faster than state of the art algorithms. All isometric methods break down in the presence of an elastic deformation, while the proposed conformal method performs well. Nonlinear refinement brings substantial improvement to the result of isometric methods but marginal improvement to the result of our conformal method. Studying this fact from a theoretical standpoint forms an interesting topic for future work. Generalizing conformal deformations to locally anisotropic deformations would also form an interesting model to study. Similarly, solving the problem of finding the conformal solution in the discrete set of solutions

returned by our conformal method using extra visual cues such as shading would be a very interesting and important topic for future work, which may be tackled with a graph-based formulation.

Acknowledgements. We thank the authors of (Salzmann and Fua, 2009) for the implementation of their reconstruction method and the authors of (Varol et al., 2012) for their dataset. This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE.

References

A. Bartoli and T. Collins. Template-based isometric deformable 3D reconstruction with sampling-based focal length self-calibration. In *International Conference on Computer Vision and Pattern Recognition*, 2013.

A. Bartoli, Y. Gérard, F. Chadebecq, and T. Collins. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In *International Conference on Computer Vision and Pattern Recognition*, 2012.

A. Bartoli, D. Pizarro, and T. Collins. A robust analytical solution to isometric shape-from-template with focal length calibration. In *International Conference on Computer Vision*, 2013.

F. L. Bookstein. Principal warps: Thin-Plate Splines and the decomposition of deformations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *International Conference on Computer Vision and Pattern Recognition*, 2000.

F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. In *Asian Conference on Computer Vision*, 2010.

A. Chhatkuli, D. Pizarro, and A. Bartoli. Stable template-based isometric 3D reconstruction in all imaging conditions by linear least-squares. In *International Conference on Computer Vision and Pattern Recognition*, 2014.

A. Del Bue. A factorization approach to structure from motion with shape priors. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Analyse Numérique*, 10:5–12, 1976.

A. Ecker, K. Kutulakos, and A. Jepson. Semidefinite programming heuristics for surface reconstruction ambiguities. In *European Conference on Computer Vision*, 2008.

O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 32(1):87–104, January 2010.

N. A. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. 3D structure recovery and unwarping surfaces applicable to planes. *International Journal of Computer Vision*, 66(3):261–281, 2006.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

K. Ikeuchi. Shape from regular patterns. *Artificial Intelligence*, 22(1):49–75, 1984.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

A. Malti, A. Bartoli, and T. Collins. Template-based conformal shape-from-motion from registered laparoscopic images. In *Conference on Medical Image Understanding and Analysis*, 2011a.

A. Malti, A. Bartoli, and T. Collins. A pixel-based approach to template-based monocular 3D reconstruction of deformable surfaces. In IEEE *International Workshop on Dynamic Shape Capture and Analysis at ICCV*, 2011b.

F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D stretchable surfaces from single images in closed form. In *International Conference on Computer Vision and Pattern Recognition*, 2009.

F. Moreno-Noguer, J. Porta, and P. Fua. Exploring ambiguities for monocular non-rigid shape estimation. In *European Conference on Computer Vision*, 2010.

Y. Ohta, K. Maenobu, and T. Sakai. Obtaining surface orientation from texels under perspective projection. In *International Joint Conference on Artificial Intelligence*, 1981.

M. Perriollat and A. Bartoli. A computational model of bounded developable surfaces with application to image-based 3D reconstruction. *Computer Animation and Virtual Worlds*, 24(5):459–476, September 2013.

M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision*, 95(2):124–137, November 2011.

J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, February 2008.

D. Pizarro and A. Bartoli. Feature-based non-rigid surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, March 2012.

C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *International Conference on Computer Vision and Pattern Recognition*, 2011.

M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *International Conference on Computer Vision and Pattern Recognition*, 2009.

M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 33(5), May 2011.

M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-D tracking. In *International Conference on Computer Vision*, 2007a.

M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *International Conference on Computer Vision and Pattern Recognition*, 2007b.

M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for nonrigid 3D shape recovery. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1–7, August 2007c.

M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. In *European Conference on Computer Vision*, 2008.

A. Shaji, A. Varol, L. Torresani, and P. Fua. Simultaneous point matching and 3D deformable surface reconstruction. In *International Conference on Computer Vision and Pattern Recognition*, 2010.

A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. ABF++: Fast and robust angle based flattening. In SIGGRAPH, 2005.

J. Shi and C. Tomasi. Good features to track. In *International Conference on Computer Vision and Pattern Recognition*, 1994.

L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, May 2008.

A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *International Conference on Computer Vision and Pattern Recognition*, 2012.

S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *European Conference on Computer Vision*, 2012.

Biographies

Adrien Bartoli has held the position of Professor of Computer Science at Université d'Auvergne since fall 2009. He leads the ALCoV (Advanced Laparoscopy and Computer Vision) research group, member of CNRS and Université d'Auvergne, at ISIT. His main research interests include image registration and Shape-from-X for rigid and non-rigid environments, with applications to computer-aided endoscopy.

Yan Gérard has been an Associate Professor in Mathematics and Computer Science at Université d'Auvergne since 2001 and a member of ALCoV since 2011. His work has mostly been theoretical, in the fields of digital geometry, discrete tomography and more recently medical imaging.

François Chadebecq is a PhD candidate in Computer Vision at Université d'Auvergne under Dr. Christophe Tilmant and Prof. Adrien Bartoli with a dual affiliation to the ComSee group at Institut Pascal and ALCoV. His research interests include 3D reconstruction and its applications in colonoscopy.

Toby Collins received the MSc degree in Artificial Intelligence at the University of Edinburgh (first in class) in 2005. In 2006 he began his PhD in Computer Vision at the University of Edinburgh. Since 2009 he has been a full-time research fellow in ALCoV. His research interests include nonrigid shape analysis, registration and reconstruction, AR for deformable surfaces and computer assisted intervention.

Daniel Pizarro Pérez received the PhD degree in Electrical Engineering in 2008 from the University of Alcala. In 2005-2012 he was an Assistant Professor and member of the GEINTRA group at the University of Alcala. Since 2013 he is an Associate Professor at Université d'Auvergne and member of ALCoV. His research interests are in optimization and Computer Vision, including image registration and deformable reconstruction, and their application to Minimally Invasive Surgery.

A Positive Definiteness of G

Lemma 1. *Let* $\mathbf{W} \in \mathbb{R}^{2\times 2}$, $\mathbf{q} \in \mathbb{R}^2$ *and* $\tilde{\mathbf{q}}^\top \stackrel{\text{def}}{=} (\mathbf{q}^\top \quad 1)$. *Given that* $\det(\mathbf{W}) \neq 0$ *and* $\|\mathbf{q}\|_2^2 \neq 0$ *we have:*

$$\mathbf{G} \stackrel{\text{def}}{=} \frac{1}{\|\tilde{\mathbf{q}}\|_2^2}\left(\mathbf{W}^\top\mathbf{W} - \frac{1}{\|\tilde{\mathbf{q}}\|_2^2}\mathbf{W}^\top\mathbf{q}\mathbf{q}^\top\mathbf{W}\right) \in \mathbb{S},$$

and:

$$\mathbf{G} = \mathbf{V}\mathbf{V}^\top \quad \textit{with} \quad \mathbf{V} \stackrel{\text{def}}{=} \frac{1}{\|\mathbf{q}\|_2}\frac{1}{\|\tilde{\mathbf{q}}\|_2^2}\mathbf{W}^\top\begin{pmatrix}\mathbf{q}_\perp & \mathbf{q}\end{pmatrix}\begin{pmatrix}\|\tilde{\mathbf{q}}\|_2 & \\ & 1\end{pmatrix},$$

where $\mathbf{q}_\perp \stackrel{\text{def}}{=} (-q_2 \quad q_1)^\top$.

Proof. For $\mathbf{q} = \mathbf{0}$ we have $\mathbf{G} = \frac{1}{\|\tilde{\mathbf{q}}\|_2^2}\mathbf{W}^\top\mathbf{W} \in \mathbb{S}$. For $\mathbf{q} \neq 0$, the proof is trivial by expanding $\mathbf{V}\mathbf{V}^\top$ and checking equality with $\mathbf{G}$. □

B Eigendecomposition of $\mathbf{A} - \lambda_j(\mathbf{A})\,\mathrm{I}$ for $\mathbf{A} \in \mathbb{S}$ and $j \in \{1,2\}$

Lemma 2. *The eigenvalues of* $\mathbf{A} - \lambda_j(\mathbf{A})\,\mathrm{I}$ *are given by* $\lambda_i(\mathbf{A} - \lambda_j(\mathbf{A})\,\mathrm{I}) = \lambda_i(\mathbf{A}) - \lambda_j(\mathbf{A})$, *giving:*

$$\begin{cases} \lambda_1(\mathbf{A} - \lambda_1(\mathbf{A})\,\mathrm{I}) &= 0 \\ \lambda_2(\mathbf{A} - \lambda_1(\mathbf{A})\,\mathrm{I}) &= \lambda_2(\mathbf{A}) - \lambda_1(\mathbf{A}) \leq 0 \\ \lambda_1(\mathbf{A} - \lambda_2(\mathbf{A})\,\mathrm{I}) &= \lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A}) \geq 0 \\ \lambda_2(\mathbf{A} - \lambda_2(\mathbf{A})\,\mathrm{I}) &= 0. \end{cases} \tag{16}$$

We also have that:

$$\begin{cases} \operatorname{rank}(\mathbf{A} - \lambda_j(\mathbf{A})\,\mathrm{I}) = 1 & \textit{if } \lambda_1(\mathbf{A}) \neq \lambda_2(\mathbf{A}) \\ \mathbf{A} - \lambda_j(\mathbf{A})\,\mathrm{I} = \mathbf{0} & \textit{otherwise,} \end{cases}$$

that $\mathbf{A} - \lambda_1(\mathbf{A})\,\mathrm{I}$ *is symmetric negative semi-definite and that* $\mathbf{A} - \lambda_2(\mathbf{A})\,\mathrm{I}$ *is symmetric positive semi-definite. The eigenvectors are the same as those of matrix* $\mathbf{A}$*:*

$$\epsilon_i(\mathbf{A} - \lambda_j(\mathbf{A})\mathrm{I}) = \epsilon_i(\mathbf{A}) \tag{17}$$

Proof. We replace $\mathbf{A}$ by its eigendecomposition in $\mathbf{A} - \lambda_j(\mathbf{A})\mathrm{I}$:

$$\mathbf{A} - \lambda_j(\mathbf{A})\mathrm{I} = \mathbf{P}\ \mathrm{diag}(\lambda_1(\mathbf{A}), \lambda_2(\mathbf{A}))\ \mathbf{P}^\top - \lambda_j(\mathbf{A})\mathrm{I}$$

Because $\mathbf{P}\mathbf{P}^\top = \mathrm{I}$ we can factorize it as:

$$\mathbf{P}\ \mathrm{diag}(\lambda_1(\mathbf{A}) - \lambda_j(\mathbf{A}), \lambda_2(\mathbf{A}) - \lambda_j(\mathbf{A}))\ \mathbf{P}^\top,$$

from which we can trivially derive all the properties given in the lemma. □

C Solution of the Rank-1 Equation of Type 1, $\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}$

Proposition 1 (Rank-1 equation, type 1)**.** *Let $\mathbf{G} \in \mathbb{S}$ and $\mathbf{K} \in \mathbb{S}$ be two known matrices. The following matrix equation defines three constraints on three unknowns in $\mathbf{u} \in \mathbb{R}^{2\times 1}$ and $z \in \mathbb{R}^+$:*

$$\mathbf{u}\mathbf{u}^\top + z\mathbf{G} \ = \ \mathbf{K}. \tag{18}$$

Equation (18) has always a unique solution for z and one or two solutions for $\mathbf{u}$. Let $\mathbf{V} \in \mathbb{R}^{2\times 2}$ be an upper triangular full rank matrix obtained from the Cholesky decomposition $\mathbf{G} = \mathbf{V}\mathbf{V}^\top$ and set $\mathbf{A} \leftarrow \mathbf{V}^{-1}\mathbf{K}\mathbf{V}^{-\top}$. The solution for z is given by $z = \lambda_2(\mathbf{A})$. The two solutions for $\mathbf{u}$ are given by $\mathbf{u} = \pm\sqrt{\lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})}\mathbf{V}\epsilon_1(\mathbf{A})$. They both vanish if $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$. We note that $\lambda_i(\mathbf{A}) = \lambda_i(\mathbf{K}\mathbf{G}^{-1})$, for $i \in \{1, 2\}$.

Proof. Because $\mathbf{G} \in \mathbb{S}$, we can always compute its Cholesky decomposition and rewrite equation (18) as:

$$\mathbf{d}\mathbf{d}^\top + z\mathrm{I} \ = \ \mathbf{A}, \tag{19}$$

with $\mathbf{d} \stackrel{\text{def}}{=} \mathbf{V}^{-1}\mathbf{u}$ and $\mathbf{A} \stackrel{\text{def}}{=} \mathbf{V}^{-1}\mathbf{K}\mathbf{V}^{-\top}$, $\mathbf{A} \in \mathbb{S}$.

Since $\mathbf{d}\mathbf{d}^\top = \mathbf{A} - z\mathrm{I}$ is rank-1 positive semi-definite, equation (19) is equivalent to:

$$\lambda_1(\mathbf{A} - z\mathrm{I}) \ \geq \ 0 \tag{20}$$

$$\lambda_2(\mathbf{A} - z\mathrm{I}) \ = \ 0. \tag{21}$$

Equation (21) implies $\det(\mathbf{A} - z\mathrm{I}) = 0$ and so $\exists j \in \{1, 2\}$, $z = \lambda_j(\mathbf{A})$. Equation (20) and lemma 2 then imply that z has a single solution given by $z = \lambda_2(\mathbf{A})$.

Multiplying equation (19) by $\mathbf{d}$ to the right gives us:

$$\|\mathbf{d}\|_2^2\mathbf{d} \ = \ (\mathbf{A} - z\mathrm{I})\mathbf{d}.$$

This equation implies $\|\mathbf{d}\|_2^2 = \lambda_1(\mathbf{A} - z\mathrm{I}) = \lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})$. The last equality follows from lemma 2. This equation also implies $\mathbf{d} \propto \epsilon_1(\mathbf{A} - z\mathrm{I}) = \epsilon_1(\mathbf{A})$, where again, the last equality follows from lemma 2. We thus obtain $\mathbf{d} =$

$\pm\sqrt{\lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})}\epsilon_1(\mathbf{A})$ and:

$$\mathbf{u} \;=\; \pm\sqrt{\lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})}\mathbf{V}\epsilon_1(\mathbf{A}).$$

It means that $\mathbf{u}$ has two opposite solutions, except for $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$ for which both solutions vanish. □

D Solution of the Rank-1 Equation of Type 2, $-\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}$

Proposition 2 (Rank-1 equation, type 2)**.** *Let $\mathbf{G} \in \mathbb{S}$ and $\mathbf{K} \in \mathbb{S}$ be two known matrices. The following matrix equation defines three constraints on three unknowns in $\mathbf{u} \in \mathbb{R}^{2\times 1}$ and $z \in \mathbb{R}^+$:*

$$-\mathbf{u}\mathbf{u}^\top + z\mathbf{G} \;=\; \mathbf{K}. \tag{22}$$

Equation (18) has always a unique solution for z and one or two solutions for $\mathbf{u}$. Let $\mathbf{V} \in \mathbb{R}^{2\times 2}$ be an upper triangular full rank matrix obtained from the Cholesky decomposition $\mathbf{G} = \mathbf{V}\mathbf{V}^\top$ and set $\mathbf{A} \leftarrow \mathbf{V}^{-1}\mathbf{K}\mathbf{V}^{-\top}$. The solution for z is given by $z = \lambda_1(\mathbf{A})$. The two solutions for $\mathbf{u}$ are given by $\mathbf{u} = \pm\sqrt{\lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})}\mathbf{V}\epsilon_2(\mathbf{A})$. They both vanish if $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$.

Proof. The proof is very similar to the proof of proposition 2. The conditions are now:

$$\lambda_1(\mathbf{A} - z\mathrm{I}) \;=\; 0 \tag{23}$$

$$\lambda_2(\mathbf{A} - z\mathrm{I}) \;\leq\; 0, \tag{24}$$

from which we get $z = \lambda_1(\mathbf{A})$ using lemma 2 and $\mathbf{u} = \pm\sqrt{\lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A})}\mathbf{V}\epsilon_2(\mathbf{A})$. □

E The Normals of Conformal Singular Points Pass at the Camera Centre

Proof. Let $\omega \in C^0(\mathcal{P}, \mathbb{R}^3)$, $\|\omega\|_2 = 1$ be the surface normal function. Those points whose normal passes at the camera centre are characterized by:

$$\omega \;\propto\; \tilde{\eta}.$$

The columns of $\mathbf{J}_\varphi$ form a basis for the tangent plane, and we can thus rewrite this equation as:

$$\mathbf{J}_\varphi^\top\omega \;=\; \mathbf{J}_\varphi^\top\tilde{\eta} \;=\; \mathbf{0}.$$

Because $\varphi = \gamma\tilde{\eta}$:

$$\mathbf{J}_\varphi \;=\; \begin{pmatrix}\eta\\ 1\end{pmatrix}\mathbf{J}_\gamma + \gamma\begin{pmatrix}\mathbf{J}_\eta\\ 0\;0\end{pmatrix}. \tag{25}$$

We start from $\mathbf{J}_\mu = 0$, which characterizes conformal singular points. Because $\theta > 0$, and $\mathbf{J}_\theta = \theta \mathbf{J}_\mu$, this condition is equivalent to $\mathbf{J}_\theta = 0$ and to:

$$\mathbf{J}_\gamma \;=\; -\frac{1}{\varepsilon^2}\theta \mathbf{J}_\varepsilon \;=\; -\frac{\theta}{\varepsilon^3}\eta^\top \mathbf{J}_\eta.$$

Replacing in equation (25) we obtain:

$$\mathbf{J}_\varphi \;=\; -\frac{\theta}{\varepsilon^3}\begin{pmatrix}\eta\\1\end{pmatrix}\eta^\top \mathbf{J}_\eta + \gamma\begin{pmatrix}\mathbf{J}_\eta\\0\;0\end{pmatrix} \;=\; \left(-\frac{\theta}{\varepsilon^3}\begin{pmatrix}\eta\\1\end{pmatrix}\eta^\top + \gamma\begin{pmatrix}\mathrm{I}\\0\;0\end{pmatrix}\right)\mathbf{J}_\eta.$$

Transposing and multiplying to the right by $\tilde{\eta}$ we obtain:

$$\mathbf{J}_\eta^\top\left(-\frac{\theta}{\varepsilon}+\gamma\right)\eta \;=\; \mathbf{J}_\eta^\top\left(-\gamma+\gamma\right)\eta \;=\; \mathbf{0},$$

which completes the proof. □