Isometric Non-Rigid Shape-from-Motion with Riemannian Geometry Solved in Linear Time

Shaifali Parashar[1], Daniel Pizarro[2,1] and Adrien Bartoli[1]

[1]ISIT - CNRS/Université d'Auvergne, Clermont-Ferrand, France
[2]GEINTRA, Universidad de Alcalá, Alcalá de Henares, Spain

Abstract—We study Isometric Non-Rigid Shape-from-Motion (Iso-NRSfM): given multiple intrinsically calibrated monocular images, we want to reconstruct the time-varying 3D shape of a thin-shell object undergoing isometric deformations. We show that Iso-NRSfM is solvable from local warps, the inter-image geometric transformations. We propose a new theoretical framework based on the Riemmanian manifold to represent the unknown 3D surfaces as embeddings of the camera's retinal plane. This allows us to use the manifold's metric tensor and Christoffel Symbol (CS) fields. These are expressed in terms of the first and second order derivatives of the inverse-depth of the 3D surfaces, which are the unknowns for Iso-NRSfM. We prove that the metric tensor and the CS are related across images by simple rules depending only on the warps. This forms a set of important theoretical results. We show that current solvers cannot solve for the first and second order derivatives of the inverse-depth simultaneously. We thus propose an iterative solution in two steps. 1) We solve for the first order derivatives assuming that the second order derivatives are known. We initialise the second order derivatives to zero, which is an infinitesimal planarity assumption. We derive a system of two cubics in two variables for each image pair. The sum-of-squares of these polynomials is independent of the number of images and can be solved globally, forming a well-posed problem for $N \geq 3$ images. 2) We solve for the second order derivatives by initialising the first order derivatives from the previous step. We solve a linear system of $4N - 4$ equations in three variables. We iterate until the first order derivatives converge. The solution for the first order derivatives gives the surfaces' normal fields which we integrate to recover the 3D surfaces. The proposed method outperforms existing work in terms of accuracy and computation cost on synthetic and real datasets.

1 Introduction

One of the main problems in 3D computer vision is to reconstruct an object's 3D shape from multiple views. This has been solved specifically for the case of rigid objects from inter-image visual motion and is known as Structure-from-Motion (SfM) (Hartley and Zisserman, 2000). However, SfM breaks down for non-rigid objects. Two ways to exploit visual motion for non-rigid object reconstruction have been proposed: Shape-from-Template (SfT) (Bartoli et al., 2015; Perriollat et al., 2011; Salzmann and Fua, 2011) and Non-Rigid Shape-from-Motion (NRSfM) (Bregler et al., 2000; Chhatkuli et al., 2014; Gotardo and Martinez, 2011; Taylor et al., 2010; Torresani et al., 2008; Varol et al., 2009; Vicente and Agapito, 2012). The latter is a direct extension of SfM to the non-rigid case. The former, however, is not. Indeed, the inputs of SfT are a single image and the object's template, and its output is the object's deformed shape. The template is a very strong object-specific prior as it includes a reference shape, a texture map and a deformation model. Most SfT methods use the thin-shell isometric deformation model. This is because isometry is a very good approximation of the deformations of many objects. The inputs to NRSfM are multiple images and its output is the object's 3D shape for every image. In NRSfM, the rigidity constraint of SfM is replaced by constraints on the object's shape and deformation model. NRSfM methods were proposed with the low-rank shape basis (Gotardo and Martinez, 2011; Torresani et al., 2008), the trajectory basis (Akhter et al., 2009), isometry (Chhatkuli et al., 2014; Vicente and Agapito, 2012) and elasticity (Agudo et al., 2015). Existing methods suffer one or several limitations amongst solution ambiguities, low accuracy, ill-posedness, inability to handle missing data and high computation cost. NRSfM thus still exists as an open research problem.

We present a solution to NRSfM using the thin-shell isometric deformation model, that we hereinafter denote as Iso-NRSfM. Iso-NRSfM and the other Isometric NRSfM methods (Chhatkuli et al., 2014; Varol et al., 2009; Vicente and Agapito, 2012) recover the 3D shape in camera coordinates. This should not be seen as a limitation of these physics-based methods. Camera motion cannot be decoupled from deformations in these methods. The composition of camera motion (which is a global isometry) with an isometric mapping results in an isometric mapping. This means there is an infinite number of such decompositions. If other priors were introduced, such as rigid boundary conditions, a meaningful camera pose could be recovered. Statistical-model based NRSfM methods such as (Gotardo and Martinez, 2011; Torresani et al., 2008) recover shape and camera motion. However they use the assumption that the deformation follows a linear basis, which may yield substantial modeling errors in practice.

We model Iso-NRSfM using concepts from Riemannian geometry. Our framework relates the 3D shape to the inter-image warps. These may be computed from keypoint correspondences in several ways (Bookstein, 1989; Pizarro et al., 2016). In practice we only need to know the warps locally, though. More specifically, we need a point track and its

. code available at http://igt.ip.uca.fr/~ab/Research/index.html

local derivatives upto second order. We assume that they are known. We model the object's 3D shape for each image as a Riemmanian manifold and deformations as isometric mappings. We parametrise each manifold by embedding the corresponding retinal plane. This allows us to reason on advanced surface properties, namely the metric tensor and the CS, directly in retinal coordinates, and in relationship to the warps. These metric properties allow us to express the differential properties of surfaces, such as length, which are to be preserved under isometric deformations. We formulate Iso-NRSfM locally with five variables which are functions of the first and the second order derivatives of the inverse-depth of the surface undergoing deformation. We write the metric tensor and the CS in terms of these variables. We prove two new theorems showing that for isometric deformations, the metric tensor and the CS may be transferred between views using only the local warps. This limits the number of variables to only five for any number N of views. In (Parashar et al., 2016), we solved Iso-NRSfM by assuming that the surface can be approximated with a plane in the infinitesimal neighbourhood of each point. This is the assumption of infinitesimal planarity, which lets us get rid of the surface's second order derivatives in the expression of the CS. This limits the number of variables to only two. These variables correspond to the first order derivatives of surface's inverse depth. We obtained a system of two cubics in these two variables that involves the first and the second order derivatives of the warps. This system holds at each point. In this paper, which is an extended version of (Parashar et al., 2016), we also solve Iso-NRSfM without the assumption of infinitesimal planarity. Our solution is obtained in two steps. *1)* We solve for the first order derivatives assuming that the second order derivatives are known. Although this step seems similar to the solution with infinitesimal planarity (where we solve for first order derivatives assuming that the second order derivatives vanish), it is a general solution to which the solution with infinitesimal planarity is a special case. *2)* We solve for the second order derivatives with the first order derivatives obtained in the previous step. We obtain a system of $4N-4$ linear equations in three variables which is solved using Linear Least Squares (LLS). We iterate these two steps until the surface inverse-depth's first order derivatives converge. The solution gives an estimate of the metric tensor field, and thus of the surface's normal field, in all views. The shape is finally recovered by integrating the normal field for each view.

The proposed method has the following features. *1)* It has a linear complexity in the number of views and number of points. *2)* It uses a well-posed point-wise solution from $N \geq 3$ views, thus covering the minimum data case. *3)* It naturally handles missing data created by occlusions. *4)* It substantially outperforms existing methods in terms of complexity and accuracy, as we experimentally verified using synthetic and real datasets. *5)* Since the method is local, the normals of point correspondences on the surface are calculated independently of each other. Hence our methods can handle non-smooth deformations as well. In such cases, the object can be modeled as a collection of piecewise smooth regions. Our method could also be implemented with parallel programming easily.

Beyond the proposed method, we bring a new theoretical framework to NRSfM, allowing one to exploit the surface's metric tensor and CS in a simple and neat way. These properties have not been studied before for NRSfM. We discuss the state-of-the-art in section 2 and present our problem modelling in section 3, our reconstruction equations in section 4, our algorithms in section 5, our experiments in section 6 and finally conclusions in section 7.

2 Previous Work

Existing NRSfM methods can be divided into three main categories: *i)* object-wise, *ii)* piece-wise and *iii)* point-wise methods. *i)* solves for the entire object's shape at once. This group includes methods that assume a low-dimensional space of deformed shapes (Dai et al., 2014; Gotardo and Martinez, 2011; Hartley and Vidal, 2008; Torresani et al., 2008). These methods have been extensively studied in the recent years. They may use other constraints such as temporal smoothness (Gotardo and Martinez, 2011; Torresani et al., 2008) or point trajectory constraints (Akhter et al., 2009). They have been shown to be accurate for objects with a low number of deformation modes, such as a talking face and articulated objects. These methods require prior knowledge to set the number of shape bases, the kernel, and its parameters. Some improvements have been made for obtaining the basis size automatically, but there is no guarantee that a collection of shapes can be represented by a low number of shape bases accurately. These methods suffer in the presence of missing data and may present ambiguities (for instance due to the orthographic camera assumption). *i)* also includes methods using physics-based models such as isometry (Vicente and Agapito, 2012), elastic deformations (Agudo et al., 2015) and particle-based interactions (Agudo and Moreno-Noguer, 2015). (Vicente and Agapito, 2012) copes with missing data but involves costly non-convex optimisation, which requires a good initialisation. (Agudo and Moreno-Noguer, 2015; Agudo et al., 2015) require rigid motion at the beginning of the sequence so that the object's shape is reconstructed using rigid SfM. Those methods are thus also related to SfT. Recently, (Chhatkuli et al., 2016) proposed an object-wise method which solves NRSfM by using the Maximum Depth Heuristic (MDH) (Perriollat et al., 2011) for isometric deformations using convex optimisation. It handles missing data and very complex deformations. It reports good experimental results and hence it is important for us to compare with it. Our experiments show that (Chhatkuli et al., 2016) performs better than the other compared methods: its performance is very close to our solution to Iso-NRSfM using infinitesimal planarity. However, our iterative method performs better than all compared methods.

Methods in *ii)* and *iii)* are sometimes also called local methods. In piece-wise methods, one selects a simple model that approximates the shape of a small region of the surface. NRSfM is then solved for each region. This can be analytical for planes (Varol et al., 2009) and local rigid motions with both the orthographic (Taylor et al., 2010) and perspective (Russell et al., 2014) cameras. More complex models, such as the local quadratic models, require non-linear iterative optimisation. After solving for each region's approximate shape, a second step is to stitch all pieces

together, imposing some order of continuity in the surface. In (Russell et al., 2011) stitching is done using submodular optimisation. For some other models stitching can be solved by LLS (Chhatkuli et al., 2014). Piece-wise methods need segmenting the image domain in regions from which the local models are computed. Region segmentation may be costly and difficult to define optimally for general surfaces. This has a major impact in the efficiency and accuracy of these methods.

Point-wise methods replace local regions with infinitesimal regions, which allow one to describe NRSfM as a system of Partial Differential Equations (PDEs) involving differential properties of the shape and derivatives of the warps. (Chhatkuli et al., 2014) presents a point-wise solution for isometric NRSfM assuming that the surface is infinitesimally planar. It gives analytical solutions to compute the surface's twofold ambiguous normal at any point from a pair of views.

Point-wise methods form a promising solution for Iso-NRSfM. In principle, they allow one to overcome the complexity, missing data, and accuracy limitations of other methods. However, in practice, no theoretical framework and practical method were proposed which overcome these limitations. Our paper attempts to fill this gap by proposing a Riemannian framework to solve NRSfM accurately, using small globally solvable optimisation problems, and in time complexity linear in the number of images and points.

3 Mathematical Model

3.1 Notation

We use small-case Latin letters to denote scalars and small-case Greek letters to denote functions. Bold and small Latin letters denote 2D and 3D vectors. We use a subscript to index the images and a superscript to index the coordinates of a point. We use $\mathbf{J}$ to write the jacobians, $\mathbf{g}$ to denote the metric tensor and $\mathbf{\Gamma}$ to denote the CS matrix. We give our modelling for a pair of views. It straightforwardly generalises to any number of views. We consider two surfaces $\mathcal{M}_i$ and $\mathcal{M}_j$, which are represented by images $\mathcal{I}_i$ and $\mathcal{I}_j$. A point in $\mathcal{I}_i$ is denoted by $\mathbf{x}$ and the corresponding one in $\mathcal{I}_j$ by $\mathbf{y}$. We do this to avoid the subscripts in the equations. Similarly, a point on the surface $\mathcal{M}_i$ is denoted by $\mathbf{z}$ and the corresponding point on $\mathcal{M}_j$ by $\mathbf{w}$. We write $(k_1, k_2, k_3, k_4, k_5)$ as the variables that represent the first and second order derivatives of the inverse-depth of the surface $\mathcal{M}_i$, (p_1, p_2, p_3) which are known quantities on $\mathcal{M}_i$, written in terms of the second order derivatives and $(c_1, c_2, c_3, c_4, c_5, c_6)$ which represent the CS at $\mathcal{M}_i$. On $\mathcal{M}_j$, we write these expressions with a bar as $(\bar{k}_1, \bar{k}_2, \bar{k}_3, \bar{k}_4, \bar{k}_5)$, $(\bar{p}_1, \bar{p}_2, \bar{p}_3)$ and $(\bar{c}_1, \bar{c}_2, \bar{c}_3, \bar{c}_4, \bar{c}_5, \bar{c}_6)$.

3.2 General Model

Our model of NRSfM is shown in Fig. 1. We have N input images $\mathcal{I}_1, \ldots, \mathcal{I}_N$ that show the projection of different isometric deformations of the same surface. The registration between the pair of images $(\mathcal{I}_i, \mathcal{I}_j)$ is known and denoted by the functions η_{ij} and η_{ji}, called warps, with $\eta_{ij} = \eta_{ji}^{-1}$. In practice, we compute these warps from keypoint correspondences using (Pizarro et al., 2016). This choice is explained and justified by theorem 4. Abusing notation, we also use $\mathcal{I}_i$ to denote an image's retinal plane, with $\mathcal{I}_i \subset \mathbb{R}^2$. Surfaces in 3D are modeled as Riemannian manifolds. This

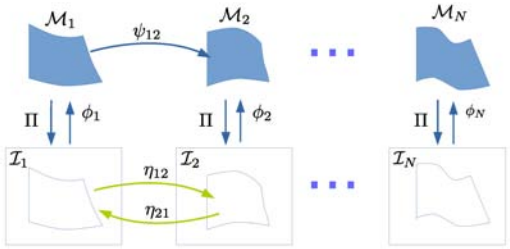

Fig. 1: The proposed model of NRSfM, where each surface $\mathcal{M}_i$ is a Riemannian manifold defined by embedding the corresponding retinal plane $\mathcal{I}_i$.

allows us to define lengths, angles and tangent planes on the surface (Lee, 1997). We denote $\mathcal{M}_i \subset \mathbb{R}^3$ the ith manifold, which can be seen as a two-dimensional subset embedded in 3D. We use the extrinsic definition of $\mathcal{M}_i$, where a function embeds a subset of the plane $\mathbb{R}^2$ into $\mathbb{R}^3$. With embedding functions, one can easily compute manifold characteristics (Lee, 2003) such as the metric tensor and the CS. However, these characteristics change according to the coordinate frame. We use the retinal plane $\mathcal{I}_i$ as coordinate frame for $\mathcal{M}_i$ and define as $\phi_i \in C^\infty(\mathcal{I}_i, \mathbb{R}^3)$ the image embedding for $\mathcal{M}_i$. We define as ψ_{ij} the isometric mapping between manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$.

3.3 Image Embeddings

The projection and embedding models we use are illustrated in Fig. 1. We use the perspective camera as projection model. For a 3D point $\mathbf{z} = \begin{pmatrix} z^1 & z^2 & z^3 \end{pmatrix}^\top$ with $z^3 > 0$, we define perspective projection Π as:

$$\mathbf{x} = \Pi(\mathbf{z}) \qquad \mathbf{x} = \begin{pmatrix} \frac{z^1}{z^3} & \frac{z^2}{z^3} \end{pmatrix}^\top, \tag{1}$$

where $\mathbf{x}$ is the projected point's retinal coordinates. The image embeddings ϕ_i with $i \in [1, N]$ define the 'inverse' of perspective projection for a particular surface, as they map retinal coordinates to the 3D surface. They must satisfy the following identity:

$$\mathbf{x} = (\Pi \circ \phi_i)(\mathbf{x}). \tag{2}$$

Smooth functions that comply with equation (2) can be expressed with a depth function $\rho_i \in C^\infty(\mathcal{I}_i, \mathbb{R})$, where:

$$\phi_i(\mathbf{x}) = \rho_i(\mathbf{x}) \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top. \tag{3}$$

Alternatively, let $\alpha_i = {\rho_i}^{-1}$ be the inverse-depth function. This allows us to re-define the image embedding in equation (3) as:

$$\phi_i(\mathbf{x}) = \frac{1}{\alpha_i(\mathbf{x})} \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top. \tag{4}$$

As we show next, working with the inverse-depth for defining the image embedding has an important role while defining the differential properties of Iso-NRSfM.

3.4 Metric Tensors

The metric tensor (see appendix A for more details) is a differential quantity used to define lengths, angles and areas on the surface (Lee, 1997). In Fig. 2, the metric tensor of ϕ_i is denoted as $\mathbf{g}_{mn}[\phi_i]$. We use the standard Einstein tensor notation and thus $\mathbf{g}_{mn}[\phi_i]$ is a combined reference to all elements of the metric tensor, a 2×2 matrix in this case. According to the Einstein summation convention, the summation is done over the indices appearing twice in the expression. Also, the free indices in an expression (the ones that do not appear twice in the expression) can be seen as both the indexed element or the whole arrangement. The indices m and n refer to the components of the coordinate frame of ϕ_i. In Fig. 2, we have $\mathbf{z} = \phi_i(\mathbf{x})$ and:

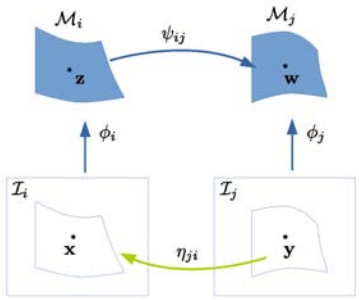

Fig. 2: Simplified notation for two images.

$$\mathbf{J}_{\phi_i} = \begin{pmatrix} \frac{\partial z^1}{\partial x^1} & \frac{\partial z^2}{\partial x^1} & \frac{\partial z^3}{\partial x^1} \\ \frac{\partial z^1}{\partial x^2} & \frac{\partial z^2}{\partial x^2} & \frac{\partial z^3}{\partial x^2} \end{pmatrix}^\top . \tag{5}$$

The metric tensor of ϕ_i is then:

$$\mathbf{g}_{mn}[\phi_i] = \mathbf{J}_{\phi_i}^\top \mathbf{J}_{\phi_i} = \frac{\partial z^s}{\partial x^m}\frac{\partial z^k}{\partial x^n}\delta_{sk}, \tag{6}$$

with δ_{sk} the Kronecker delta function. We remind that the summation in equation (6) is done over indices s and k. The inverse of the metric tensor is expressed with raised indices $\mathbf{g}^{mn}[\phi_i]$. Given the change of coordinates:

$$\mathbf{x} = \eta(\mathbf{y}) \quad \text{with} \quad \mathbf{y} = \begin{pmatrix} y^1 & y^2 \end{pmatrix}^\top , \tag{7}$$

the metric tensor of $\phi_i \circ \eta$ is obtained as:

$$\mathbf{g}_{st}[\phi_i \circ \eta] = \mathbf{J}_\eta^\top \mathbf{J}_{\phi_i}^\top \mathbf{J}_{\phi_i} \mathbf{J}_\eta = \frac{\partial x^m}{\partial y^s}\frac{\partial x^n}{\partial y^t}\mathbf{g}_{mn}[\phi_i]. \tag{8}$$

3.5 Christoffel Symbols (CS)

CS (see appendix B for more details) of the second kind are function arrays that describe several properties of a Riemannian manifold, such as the curvature tensor, the geodesic equations of curves and the parallel transport of vectors in surfaces (Lee, 1997). We denote the CS of embedding ϕ_i as $\mathbf{\Gamma}^p_{mn}[\phi_i]$. It is useful to represent the CS of ϕ_i as two 2×2 matrices $\mathbf{\Gamma}^1_{mn}[\phi_i]$ and $\mathbf{\Gamma}^2_{mn}[\phi_i]$, where the upper indices 1 and 2 make reference to the 2D image coordinates $\mathbf{x} = \begin{pmatrix} x^1 & x^2 \end{pmatrix}^\top$, where ϕ_i is defined. The CS are given by:

$$\mathbf{\Gamma}^p_{mn}[\phi_i] = \frac{1}{2}\mathbf{g}^{pl}[\phi_i]\left(\mathbf{g}_{lm,n}[\phi_i] + \mathbf{g}_{ln,m}[\phi_i] - \mathbf{g}_{mn,l}[\phi_i]\right), \tag{9}$$

where $\mathbf{g}_{lm,n} = \partial_n \mathbf{g}_{lm}$. Given a change of coordinates $\mathbf{x} = \eta(\mathbf{y})$, the CS in the new coordinates are given as:

$$\mathbf{\Gamma}^q_{st}[\phi_i \circ \eta] = \frac{\partial x^m}{\partial y^s}\frac{\partial x^n}{\partial y^t}\mathbf{\Gamma}^p_{mn}[\phi_i]\frac{\partial y^q}{\partial x^p} + \frac{\partial y^q}{\partial x^l}\frac{\partial^2 x^l}{\partial y^s \partial y^t}. \tag{10}$$

Note that even though CS are expressed with tensor notation, they are not tensors and thus equation (10) does not correspond to the way tensors change coordinates. The CS of the image embedding, defined via the inverse-depth in equation (4), has a special structure given in Theorem 1, whose proof is given in appendix C.

Theorem 1. *Let $\mathbf{x} \in \mathcal{I}_i$, then $\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]$ is given by:*

$$\begin{aligned} \mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] &= \frac{-1}{\alpha_i}\begin{pmatrix} 2\alpha_{i,1} & \alpha_{i,2} \\ \alpha_{i,2} & 0 \end{pmatrix} + \frac{(\alpha_i)^2 A_i}{D_i}\begin{pmatrix} \alpha_{i,11} & \alpha_{i,12} \\ \alpha_{i,12} & \alpha_{i,22} \end{pmatrix} \\ \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] &= \frac{-1}{\alpha_i}\begin{pmatrix} 0 & \alpha_{i,1} \\ \alpha_{i,1} & 2\alpha_{i,2} \end{pmatrix} + \frac{(\alpha_i)^2 B_i}{D_i}\begin{pmatrix} \alpha_{i,11} & \alpha_{i,12} \\ \alpha_{i,12} & \alpha_{i,22} \end{pmatrix}, \end{aligned} \tag{11}$$

where $\alpha_{i,k} = \dfrac{\partial \alpha_i}{\partial x^k}$, $\alpha_{i,nm} = \dfrac{\partial^2 \alpha_i}{\partial x^n \partial x^m}$ and:

$$\begin{aligned} A_i &= -x^1\alpha_i + \left(1 + \left(x^1\right)^2\right)\alpha_{i,1} + x^1 x^2 \alpha_{i,2} \\ B_i &= -x^2\alpha_i + \left(1 + \left(x^2\right)^2\right)\alpha_{i,2} + x^1 x^2 \alpha_{i,1} \\ D_i &= \left(\alpha_i - x^1\alpha_{i,1} - x^2\alpha_{i,2}\right)^2 + \left(\alpha_{i,1}\right)^2 + \left(\alpha_{i,2}\right)^2 . \end{aligned} \tag{12}$$

3.6 Commutativity under Isometry

Images and surfaces in Iso-NRSfM follow the commutative diagram shown in Fig. 2. Therefore,

$$\begin{aligned} \phi_j &= \psi_{ij} \circ \phi_i \circ \eta_{ji} \\ \mathbf{J}_{\phi_j} &= \mathbf{J}_{\psi_{ij}} \mathbf{J}_{\phi_i} \mathbf{J}_{\eta_{ji}}. \end{aligned} \tag{13}$$

The metric tensor of ϕ_j can be written according to equation (6). It is given by

$$\mathbf{J}_{\phi_j}^\top \mathbf{J}_{\phi_j} = \mathbf{J}_{\eta_{ji}}^\top \mathbf{J}_{\phi_i}^\top \mathbf{J}_{\psi_{ij}}^\top \mathbf{J}_{\psi_{ij}} \mathbf{J}_{\phi_i} \mathbf{J}_{\eta_{ji}} = \mathbf{J}_{\eta_{ji}}^\top \mathbf{J}_{\phi_i}^\top \mathbf{J}_{\phi_i} \mathbf{J}_{\eta_{ji}}. \tag{14}$$

The fact that mappings between manifolds are isometric ($\mathbf{J}_{\psi_{ij}}^\top \mathbf{J}_{\psi_{ij}} = \mathbf{I}_{3\times 3}$) allows us to derive the following fundamental result: *in Iso-NRSfM, both metric tensors and CS commute between surfaces with a change of variable given by the image warps.* This result is formalised with Theorem 2 and Corollary 1 proved in appendix C.

Theorem 2. *Let ψ_{ij} be an isometric mapping between the manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$, then $\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[\phi_i \circ \eta_{ji}]$ with $(i, j) \in [1, N] \times [1, N]$.*

Corollary 1. *Let ψ_{ij} be an isometric mapping between the manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$, then $\mathbf{\Gamma}^p_{mn}[\phi_j] = \mathbf{\Gamma}^p_{mn}[\phi_i \circ \eta_{ji}]$ with $(i, j) \in [1, N] \times [1, N]$.*

These results show that, given the metric tensor and the CS for one image embedding, they can be transferred to the rest of the embeddings using the warps, which are known entities in Iso-NRSfM. Note that this is not true in general when the mappings are non-isometric. This result establishes the ground rules for developing a local solution to Iso-NRSfM where the number of unknowns does not grow with the number of images. The main idea is to define the unknowns in a reference image and to use the warps to transfer all constraints into it.

3.7 Infinitesimal Planarity

Infinitesimal planarity refers to the assumption that a surface at each point is approximately planar in its infinitesimal neighbourhood. This is fundamentally different from piecewise planarity: in infinitesimal planarity, the surface is globally curved, but in an infinitesimal neighbourhood, it may be represented by a plane. In other words, each infinitesimal model agrees with the global surface at the point where infinitesimal planarity is assumed, but this agreement holds only at the zeroth order.

We study the differential properties of the image embedding when the surface is a plane. We then invoke infinitesimal planarity to extend these properties point-wise to non-planar surfaces. In this regard we present Theorem 3 and Corollary 2 proved in appendix C.

Theorem 3. *If $\mathcal{M}$ is a plane then its image embedding at $\mathbf{x} \in \mathcal{I}$ is $\phi(\mathbf{x}) = \beta(\mathbf{x})^{-1}(\mathbf{x} \quad 1)^\top$ with β a linear function.*

Corollary 2. *Let $\mathcal{M}$ be a plane and $\phi(\mathbf{x})$ the image embedding at $\mathbf{x} \in \mathcal{I}$, the CS $\mathbf{\Gamma}^p_{mn}[\phi(\mathbf{x})]$ are given by:*

$$\mathbf{\Gamma}^1_{mn}[\phi(\mathbf{x})] = \frac{1}{\beta(\mathbf{x})}\begin{pmatrix} -2\beta_1(\mathbf{x}) & -\beta_2(\mathbf{x}) \\ -\beta_2(\mathbf{x}) & 0 \end{pmatrix}$$
$$\mathbf{\Gamma}^2_{mn}[\phi(\mathbf{x})] = \frac{1}{\beta(\mathbf{x})}\begin{pmatrix} 0 & -\beta_1(\mathbf{x}) \\ -\beta_1(\mathbf{x}) & -2\beta_2(\mathbf{x}) \end{pmatrix}, \tag{15}$$

where $\beta_1(\mathbf{x}) = \dfrac{\partial \beta(\mathbf{x})}{\partial x^1}$ and $\beta_2(\mathbf{x}) = \dfrac{\partial \beta(\mathbf{x})}{\partial x^2}$.

Theorem 3 shows that the inverse-depth β of a planar surface is a linear function. Corollary 2 is derived from Theorem 1 and Theorem 3. It shows that the CS have a simplified structure under infinitesimal planarity, where at any point they have 3 degrees of freedom: β and its first order derivatives. Moreover this also shows that both the metric tensor and the CS share the same 3 unknowns.

From Corollary 2 we find the following constraints over the elements of the CS:

$$\begin{aligned} \mathbf{\Gamma}^1_{22}[\phi(\mathbf{x})] &= \mathbf{\Gamma}^2_{11}[\phi(\mathbf{x})] = 0 \\ 2\mathbf{\Gamma}^2_{12}[\phi(\mathbf{x})] &= \mathbf{\Gamma}^2_{22}[\phi(\mathbf{x})] \\ \mathbf{\Gamma}^1_{11}[\phi(\mathbf{x})] &= 2\mathbf{\Gamma}^2_{12}[\phi(\mathbf{x})]. \end{aligned} \tag{16}$$

We derive Theorem 4 from equation (16). It shows that the warps must comply with the 2D Schwarzian derivatives (Sasaki and Yoshida, 2002), which are second order bilinear PDEs that arise in the field of projective differential geometry.

Theorem 4. *Given that $\mathcal{M}_i$ with $i \in [1, N]$ are planes, the registration warps η_{ij} with $(i, j) \in [1, N] \times [1, N]$ are point-wise solutions of the 2D Schwarzian equations.*

The 2D Schwarzian derivatives were used in (Pizarro et al., 2016) as a penalty to compute 'Schwarps', smooth warps that preserve the deformation's local projective structure. Schwarps were shown to improve accuracy in both SfT and NRSfM with respect to other smoothing penalties based on the bending energy. Theorem 4 theoretically justifies our choice to use Schwarps for computing our image warps. Nonetheless our method can also be used with any means to compute the local image warps.

4 Reconstruction Equations

We study local solutions to Iso-NRSfM, based on the differential properties derived in the previous section. We show that Iso-NRSfM can be posed as a non-linear PDE system and that we can find non-holonomic solutions of this system. We do not deal with boundary conditions in the PDE as we find algebraic solutions of the system in terms of the non-holonomic variables. This follows the same path as (Bartoli et al., 2015) for finding local solutions in Iso-SfT.

For planes, there is a unique linear relationship between the metric tensor and the CS, which is why Iso-NRSfM is solvable under the assumption of infinitesimal planarity. Corollary 2 shows that both of them can be expressed in terms of the first order derivatives of the inverse-depth of the surface only. We explore this relationship for non-planar surfaces, where we need the first and the second order derivatives of the inverse-depth of the surface to express the metric tensor and the CS. We argue that there is no uniqueness in the relationship between the metric tensor and the CS anymore, and therefore, there is not a unique solution to Iso-NRSfM locally. Then, we propose to solve Iso-NRSfM by solving for the first and the second order derivatives separately.

4.1 Relating the Metric Tensor and the CS

For a non-planar surface, the CS at $\mathbf{z} \in \mathcal{M}_i$ are given by equation (11). We define them as:

$$\mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] = \begin{pmatrix} c_1 & c_3 \\ c_3 & c_5 \end{pmatrix} \quad \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] = \begin{pmatrix} c_2 & c_4 \\ c_4 & c_6 \end{pmatrix}, \tag{17}$$

where c_1, c_2, c_3, c_4, c_5 and c_6 are expressed in terms of the first and second order derivatives of $\alpha_i(\mathbf{x})$ defined in equation (4). The expressions in equation (11) are:

$$\begin{aligned} c_1 &= -2k_1 + k_3 A_i & c_2 &= k_3 B_i \\ c_3 &= -k_2 + k_4 A_i & c_4 &= -k_1 + k_4 B_i \\ c_5 &= k_5 A_i & c_6 &= -2k_2 + k_5 B_i, \end{aligned} \tag{18}$$

with:

$$\begin{aligned} A_i &= -x^1 + \left(1 + \left(x^1\right)^2\right) k_1 + x^1 x^2 k_2 \\ B_i &= -x^2 + \left(1 + \left(x^2\right)^2\right) k_2 + x^1 x^2 k_1 \\ D_i &= \left(1 - x^1 k_1 - x^2 k_2\right)^2 + \left(k_1\right)^2 + \left(k_2\right)^2, \end{aligned} \tag{19}$$

where $k_1 = \dfrac{\alpha_{i,1}}{\alpha_i}$, $k_2 = \dfrac{\alpha_{i,2}}{\alpha_i}$, $k_3 = \dfrac{\alpha_{i,11}}{\alpha_i D_i}$, $k_4 = \dfrac{\alpha_{i,12}}{\alpha_i D_i}$ and $k_5 = \dfrac{\alpha_{i,22}}{\alpha_i D_i}$. The jacobian and hence the metric tensor at $\mathbf{z}$ can be written in terms of (k_1, k_2). Our goal is to find a relationship between the metric tensor parametrised with (k_1, k_2) and the CS $(c_1, c_2, c_3, c_4, c_5, c_6)$. Having such a relationship, we can formulate a system of equations exploiting the transfer of variables in the CS and metric tensor from one surface to another. From c_1 and c_2 in equation (18), we can write:

$$\frac{c_1 + 2k_1}{c_2} = \frac{A_i}{B_i}. \tag{20}$$

Similarly, from c_5 and c_6 in equation (18), we can write:

$$\frac{c_5}{c_6 + 2k_2} = \frac{A_i}{B_i}. \tag{21}$$

From c_3 and c_4 in equation (18), we find $\frac{A_i}{B_i} = \frac{c_3+k_2}{c_4+k_1}$. We substitute $\frac{A_i}{B_i}$ in equations (20) and (21), we obtain:

$$\begin{aligned}(c_1+2k_1)(c_4+k_1) &= c_2(c_3+k_2)\\ (c_6+2k_2)(c_3+k_2) &= c_5(c_4+k_1).\end{aligned} \tag{22}$$

From the first expression in equation (22), we find $k_2 = \frac{(c_1+2k_1)(c_4+k_1)}{c_2} - c_3$ and substitute it in the second expression. We obtain the following quartic in k_1:

$$\begin{aligned}&(c_4+k_1)\left(8k_1^3+8\left(c_1+c_4\right)k_1^2\right.\\ &+2\left(c_1\left(c_1+4c_4\right)+c_2\left(c_6-2c_3\right)\right)k_1\\ &\left.+2c_1^2c_4+c_1c_2\left(c_6-2c_3\right)-c_2^2c_5\right)=0.\end{aligned} \tag{23}$$

This gives up to four possible solutions to k_1, which means that there is not a unique relationship between the CS and the metric tensor.

In the solution with the infinitesimal planarity assumption, we had obtained a system of two cubics in two variables for each pair of views. Combining equation (22) with equation (18), we can express (k_1, k_2) in terms of the CS $(c_1, c_2, c_3, c_4, c_5, c_6)$ as a rational expression of degree two. This gives a system of two polynomials of degree 8 in 6 variables for each pair of views. Existing solvers such as (Henrion and Lasserre, 2003) cannot solve such high degree polynomial systems. We conclude that the first and second order derivatives of $\alpha_i(\mathbf{x})$ cannot be solved jointly via estimating the CS. However, we see that the expressions of the CS in equation (18) are linear in terms of (k_1, k_2) and (k_3, k_4, k_5). By assuming (k_3, k_4, k_5) to be known, we can find a unique relationship between the metric tensor and the CS and vice versa. Therefore, splitting the problem in two steps of solving for the first and the second order derivatives of $\alpha_i(\mathbf{x})$ separately leads to a solution to Iso-NRSfM.

4.2 Solving for the First Order Derivatives

We assume that the second order derivatives of $\alpha_i(\mathbf{x})$ are known. They can be assumed to be zero (as in the case of the infinitesimal planarity assumption) or they can be obtained by the method we describe next. We show how to solve for the first order derivatives of $\alpha_i(\mathbf{x})$. We also show that this solution has a similar structure as the solution to Iso-NRSfM under infinitesimal planarity. We first select a pair of surfaces $(\mathcal{M}_i, \mathcal{M}_j)$ (see Fig. 2) and a point $\mathbf{x} = (x^1, x^2)^\top \in \mathcal{I}_i$. We evaluate $\mathbf{\Gamma}^p_{mn}[\phi_i]$ at $\mathbf{x}$, namely $\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]$. According to equation (11), it is given by:

$$\begin{aligned}\mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] &= \begin{pmatrix}-2k_1+A_ip_1 & -k_2+A_ip_2\\ -k_2+A_ip_2 & A_ip_3\end{pmatrix}\\ \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] &= \begin{pmatrix}B_ip_1 & -k_1+B_ip_2\\ -k_1+B_ip_2 & -2k_2+B_ip_3\end{pmatrix},\end{aligned} \tag{24}$$

where $k_1 = \frac{\beta_1(\mathbf{x})}{\beta(\mathbf{x})}$ and $k_2 = \frac{\beta_2(\mathbf{x})}{\beta(\mathbf{x})}$. The expressions (p_1, p_2, p_3) are functions of second order derivatives of $\alpha_i(\mathbf{x})$ and therefore, they are known. A_i and B_i are linear expressions in (k_1, k_2) according to equation (12). Next we compute $\mathbf{J}_{\phi_i}$ in terms of (k_1, k_2):

$$\mathbf{J}_{\phi_i}(\mathbf{x}) = \frac{1}{\beta(\mathbf{x})}\begin{pmatrix}1-k_1x^1 & -k_2x^1\\ -k_1x^2 & 1-k_2x^2\\ -k_1 & -k_2\end{pmatrix}. \tag{25}$$

By substitution of equation (25) in equation (6) we have:

$$\begin{aligned}\mathbf{g}_{11}[\phi_i(\mathbf{x})] =& \frac{1}{\beta(\mathbf{x})^2}\left({k_1}^2+\left(k_1x^1-1\right)^2+\left(k_1x^2\right)^2\right)\\ \mathbf{g}_{12}[\phi_i(\mathbf{x})] =& \frac{1}{\beta(\mathbf{x})^2}\left(k_1k_2\left(1+\left(x^1\right)^2+\left(x^2\right)^2\right)-k_2x^1-k_1x^2\right)\\ \mathbf{g}_{22}[\phi_i(\mathbf{x})] =& \frac{1}{\beta(\mathbf{x})^2}\left({k_2}^2+\left(k_2x^1\right)^2+\left(k_2x^2-1\right)^2\right).\end{aligned} \tag{26}$$

We define $\mathbf{G}_{mn}[\phi_i(\mathbf{x})] = \beta(\mathbf{x})^2\mathbf{g}_{mn}[\phi_i(\mathbf{x})]$, which only depends on (k_1, k_2). Let $\mathbf{x} = \eta_{ji}(\mathbf{y})$. We use equation (10) and Corollary 1 to compute $\mathbf{\Gamma}^k_{mn}[\phi_j(\mathbf{y})] = \mathbf{\Gamma}^k_{mn}[(\phi_i \circ \eta_{ji})(\mathbf{y})]$ as:

$$\begin{aligned}\mathbf{\Gamma}^1_{mn}[(\phi_i\circ\eta_{ji})(\mathbf{y})] &= \begin{pmatrix}-2\bar{k}^1+A_j\bar{p}_1 & -\bar{k}^2+A_j\bar{p}_2\\ -\bar{k}^2+A_j\bar{p}_2 & A_j\bar{p}_3\end{pmatrix}\\ \mathbf{\Gamma}^2_{mn}[(\phi_i\circ\eta_{ji})(\mathbf{y})] &= \begin{pmatrix}B_j\bar{p}_1 & -\bar{k}^1+B_j\bar{p}_2\\ -\bar{k}^1+B_j\bar{p}_2 & -2\bar{k}^2+B_j\bar{p}_3\end{pmatrix},\end{aligned} \tag{27}$$

where according to equation (10), $(\bar{k}_1, \bar{k}_2)$ are linear combinations of (k_1, k_2) and $(\bar{p}_1, \bar{p}_2, \bar{p}_3)$, which are known. A_j and B_j are linear expressions in $(\bar{k}_1, \bar{k}_2)$ according to equation (12). From equation (26) one can find $\mathbf{g}_{st}[\phi_j(\mathbf{y})]$ in function of $(\bar{k}_1, \bar{k}_2)$, and thus in function of (k_1, k_2).

Alternatively, from equation (8) and using the definition of $\mathbf{G}_{mn}[\phi_i(\mathbf{x})]$ and $\mathbf{G}_{mn}[\phi_j(\mathbf{y})]$ we have the following identity:

$$\frac{1}{\beta(\mathbf{x})^2}\mathbf{G}_{st}[\phi_j(\mathbf{y})] = \frac{1}{\beta(\mathbf{y})^2}\frac{\partial x^m}{\partial y^s}\frac{\partial x^n}{\partial y^t}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]. \tag{28}$$

We cancel $\beta(\mathbf{x})$ and $\beta(\mathbf{y})$ by converting system (28) into the following two equations:

$$\begin{aligned}&\mathbf{G}_{11}[\phi_j(\mathbf{y})]\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)\\ -&\mathbf{G}_{12}[\phi_j(\mathbf{y})]\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)=0\\ &\mathbf{G}_{22}[\phi_j(\mathbf{y})]\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)\\ -&\mathbf{G}_{12}[\phi_j(\mathbf{y})]\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)=0.\end{aligned} \tag{29}$$

We recall that both $\mathbf{G}_{mn}[\phi_i(\mathbf{x})]$ and $\mathbf{G}_{st}[\phi_j(\mathbf{y})]$ are only functions of (k_1, k_2) and $\mathbf{x}$.

Equation (29) is a system of two cubics in variables (k_1, k_2) modeling Iso-NRSfM for manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$ at point $\mathbf{x} \in \mathcal{I}_i$. We denote the two equations as $\mathcal{Q}_{ij}(\mathbf{x}, \eta_{1j}(\mathbf{x}), k_1, k_2)$. By keeping the first index as the reference manifold, for instance $i = 1$, and obtaining the polynomials for the rest of the views we have $2N - 2$ polynomial equations in two variables $\mathcal{Q}_1(k_1, k_2) = \{\mathcal{Q}_{1j}(\mathbf{x}, \eta_{1j}(\mathbf{x}), k_1, k_2)\}_{j=2}^n$. The solution (k_1, k_2) to the polynomial system $\mathcal{Q}_1(k_1, k_2)$ at the point $\mathbf{x} = \mathbf{x}_1$ allows us to reconstruct the metric tensor, the CS and the tangent plane for point $\mathbf{x}_1$ in view $\mathcal{I}_1$. Using equation (25) we can reconstruct $\mathbf{J}_{\phi_1}(\mathbf{x}_1)$ up to an unknown scale $\beta(\mathbf{x}_1)^{-1}$. It is not necessary to recover this scale to estimate a unit normal, which is computed by taking the cross product of the two columns of $\mathbf{J}_{\phi_1}(\mathbf{x}_1)$ and normalising.

We solve system $\mathcal{Q}_1(k_1, k_2)$ by finding the values of (k_1, k_2) that minimise the sum-of-squares of all polynomials in the system. This optimisation is solved globally using

moment based convex optimisation (Henrion and Lasserre, 2003). The cost function's complexity is independent of the number of views N. Given (k_1, k_2), we calculate $(\bar{k}_1, \bar{k}_2)$ by using equation (10) at each point.

Notice that the structure of the CS given in equation (15) for planes is very similar to equation (24) with (p_1, p_2, p_3) as zeros. This shows that the solution to Iso-NRSfM with the infinitesimal planarity assumption is a special case of this solution. We express the system with zero second derivatives as $\mathcal{P}_1(k_1, k_2)$, which is solved in a similar way as $\mathcal{Q}_1(k_1, k_2)$.

4.3 Solving for the Second Order Derivatives

We now show how to solve for the second order derivatives of $\alpha_i(\mathbf{x})$, assuming that the first order derivatives of $\alpha_i(\mathbf{x})$ are known from the previous step. The expressions for the CS in equation (24) become linear in the second order derivatives of $\alpha_i(\mathbf{x})$. This means that $\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]$ is a linear function of (k_3, k_4, k_5). Given that $\mathbf{x} = \eta_{ji}(\mathbf{y})$ and equation (10), $\mathbf{\Gamma}^p_{mn}[(\phi_i \circ \eta_{ji})(\mathbf{y})]$ is given by:

$$\begin{aligned}\mathbf{\Gamma}^1_{mn}[(\phi_i \circ \eta_{ji})(\mathbf{y})] &= \begin{pmatrix} \bar{c}_1 & \bar{c}_3 \\ \bar{c}_3 & \bar{c}_5 \end{pmatrix} \\ \mathbf{\Gamma}^2_{mn}[(\phi_i \circ \eta_{ji}(\mathbf{y}))] &= \begin{pmatrix} \bar{c}_2 & \bar{c}_4 \\ \bar{c}_4 & \bar{c}_6 \end{pmatrix},\end{aligned} \tag{30}$$

where $(\bar{c}_1, \bar{c}_2, \bar{c}_3, \bar{c}_4, \bar{c}_5, \bar{c}_6)$ are expressed as a linear combination of (k_3, k_4, k_5). Therefore, at $\mathcal{M}_j$, $(\bar{k}_3, \bar{k}_4, \bar{k}_5)$ are given by the following expressions:

$$\begin{aligned}\bar{k}_3 &= \frac{\left(\bar{c}_1 + 2\bar{k}_1\right) A_j + \bar{c}_2 B_j}{A_j^2 + B_j^2} \\ \bar{k}_4 &= \frac{\left(\bar{c}_3 + \bar{k}_2\right) A_j + \left(\bar{c}_4 + \bar{k}_1\right) B_j}{A_j^2 + B_j^2} \\ \bar{k}_5 &= \frac{\bar{c}_5 A_j + \left(\bar{c}_6 + 2\bar{k}_2\right) B_j}{A_j^2 + B_j^2}.\end{aligned} \tag{31}$$

These expressions show that $(\bar{k}_3, \bar{k}_4, \bar{k}_5)$ can be expressed as a linear combination of (k_3, k_4, k_5).

In order to solve for the second order derivatives of $\alpha_i(\mathbf{x})$, we differentiate the first order reconstruction equations (29). The expressions are given in equation (32). The derivatives of $\mathbf{G}_{mn}[\phi_i(\mathbf{x})]$ in equation (29) are given in equation (33). Equation (33) shows that the derivatives are linear functions of (k_3, k_4, k_5). Using equation (31), the reconstruction equations (32) form a linear system in three variables only, which can be solved using LLS. Therefore, for each pair of manifolds $(\mathcal{M}_i, \mathcal{M}_j)$ at point $\mathbf{x} \in \mathcal{I}_i$, equation (32) is a system of four linear equations in variables (k_3, k_4, k_5). We denote the four equations as $\mathcal{S}_{ij}(\mathbf{x}, \mathbf{y}, k_3, k_4, k_5)$. Fixing the i^{th} manifold as a reference, for instance $i = 1$, we obtain $4N - 4$ linear equations in three variables which are written as $\mathcal{S}_1(k_3, k_4, k_5) = \{\mathcal{S}_{1j}(\mathbf{x}, \eta_{1j}(\mathbf{x}), k_3, k_4, k_5)\}_{j=2}^{n}$. The solution (k_3, k_4, k_5) to the linear system $\mathcal{S}_1(k_3, k_4, k_5)$ for the point $\mathbf{x} = \mathbf{x}_1$ gives the second derivatives of $\alpha_i(\mathbf{x})$. We use them to compute a better estimate of the CS in equation (11). Using equation (31), we can obtain $(\bar{k}_3, \bar{k}_4, \bar{k}_5)$ using (k_3, k_4, k_5).

$$\begin{aligned}
&\frac{\partial \mathbf{G}_{11}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&+\mathbf{G}_{11}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0 \\
&\frac{\partial \mathbf{G}_{11}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&+\mathbf{G}_{11}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0 \\
&\frac{\partial \mathbf{G}_{22}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&+\mathbf{G}_{22}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0 \\
&\frac{\partial \mathbf{G}_{22}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&+\mathbf{G}_{22}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) \\
&-\mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0.
\end{aligned} \tag{32}$$

5 Algorithms

We describe our solutions to Iso-NRSfM based on the theoretical results from the previous sections. First, we describe the algorithm for solving Iso-NRSfM with the infinitesimal planarity assumption and then we describe the algorithm for the general solution. This uses the solution with the infinitesimal planarity assumption as initialisation. The inputs to our system are N images of a deforming object and their inter-image warps with respect to the first image η_{j1} and η_{1j}. The outputs of our system are the 3D points and normals corresponding to the point correspondences for the N images. In our formulation, our goal is to find the jacobian of the image embedding. The normals are then obtained from the jacobian and the 3D points are calculated by integrating the normal field as in (Chhatkuli et al., 2014).

5.1 Solution under Infinitesimal Planarity

With the assumption of infinitesimal planarity, we can write the metric tensor of equation (24) and the CS of equation (26) on a manifold $\mathcal{M}_i$ in terms of two variables which correspond to the first order derivatives of the inverse-depth β of

$$
\begin{aligned}
\frac{\partial \mathbf{G}_{11}[\phi_i(\mathbf{x})]}{\partial x^1} &= 2k_1\left(x^1k_1-1\right)+2\left(\epsilon k_1-x^1\right)\left(k_3D-{k_1}^2\right)\\
\frac{\partial \mathbf{G}_{12}[\phi_i(\mathbf{x})]}{\partial x^1} &= k_2\left(2x^1k_1-1\right)+\left(\epsilon k_2-x^2\right)\left(k_3D-{k_1}^2\right)\\
&\quad+\left(\epsilon k_1-x^1\right)\left(k_4D-k_1k_2\right)\\
\frac{\partial \mathbf{G}_{22}[\phi_i(\mathbf{x})]}{\partial x^1} &= 2x^1{k_2}^2+2\left(\epsilon k_2-x^2\right)\left(k_4D-k_1k_2\right)\\
\frac{\partial \mathbf{G}_{11}[\phi_i(\mathbf{x})]}{\partial x^2} &= 2x^2{k_1}^2+2\left(\epsilon k_1-x^1\right)\left(k_4D-k_1k_2\right)\\
\frac{\partial \mathbf{G}_{12}[\phi_i(\mathbf{x})]}{\partial x^2} &= k_1\left(2x^2k_2-1\right)+\left(\epsilon k_2-x^2\right)\left(k_4D-k_1k_2\right)\\
&\quad+\left(\epsilon k_1-x^1\right)\left(k_5D-{k_2}^2\right)\\
\frac{\partial \mathbf{G}_{22}[\phi_i(\mathbf{x})]}{\partial x^2} &= 2k_2\left(x^2k_2-1\right)+2\left(\epsilon k_2-x^2\right)\left(k_5D-{k_2}^2\right),\\
\text{with } D &= \left(1-x^1k_1-x^2k_2\right)^2+{k_1}^2+{k_2}^2\\
\text{and } \epsilon &= 1+\left(x^1\right)^2+\left(x^2\right)^2.
\end{aligned}
\tag{33}
$$

the image embedding ϕ_i. We can also write the metric tensor and the CS on the rest of the images in terms of the variables in the first image (equations (8) and (10) respectively), which leads to two variables for N images. We solve the system of two cubics in two variables of equation (29) for all images by minimising the sum-of-squares of the polynomials. The algorithm takes the following steps:

Inputs: Warps η_{j1}, $j \in [2, N]$.

1) Find point correspondences. Select a grid of points on the first image and using the warps η_{1j}, find the corresponding grid of points on the rest of the images. We evaluated our method on a 20×20 grid of points.

2) Find (k_1, k_2). Evaluate the polynomial $\mathcal{P}_1(k_1, k_2)$ and solve by minimising the sum of squares using (Henrion and Lasserre, 2003). This gives (k_1, k_2). Find $(\bar{k}_1, \bar{k}_2)$ in terms of (k_1, k_2) and the η_{j1} warps, $j \in [2, N]$, using equation (10).

3) Find normals and 3D points. Find the jacobian in terms of (k_1, k_2) using equation (25). Compute normals by taking the cross-product of the jacobian's columns and normalising it. Use the method in (Chhatkuli et al., 2014) to recover the 3D surfaces by integrating the normal fields.

Outputs: Points and normals on 3D surfaces.

5.2 General Solution

We still have the metric tensors on a manifold $\mathcal{M}_i$ in terms of two variables but the CS are now written in terms of five variables, the first and the second order derivatives of the inverse-depth α_i of the image embedding ϕ_i. We iteratively solve for the first and second order derivatives alternatively until the first order derivatives of $\alpha_i(\mathbf{x})$ converge. Our algorithm is:

Inputs: Warps η_{j1}, $j \in [2, N]$.

1) Find point correspondences. This step is the same as step *1)* from the previous algorithm.

2) Initialise (k_1, k_2) *using the solution under infinitesimal planarity.* Run step *2)* from the previous algorithm.

3) Find the second order derivatives (k_3,k_4,k_5). Evaluate the linear system $\mathcal{S}_1(k_3, k_4, k_5)$ and solve using LLS. This gives (k_3, k_4, k_5). Find $(\bar{k}_3, \bar{k}_4, \bar{k}_5)$ using equation (31).

4) Find (k_1, k_2). Evaluate the sum-of-squares polynomials of the system $\mathcal{Q}_1(k_1, k_2)$ and find (k_1, k_2) by minimising it using (Henrion and Lasserre, 2003). Find $(\bar{k}_1, \bar{k}_2)$ in terms of (k_1, k_2) and the warps η_{j1}, $j \in [2, N]$, using equation (10).

5) Repeat steps 3) and 4) until the solution to (k_1, k_2) *converges.* The maximum number of iterations is set to 5.

6) Find normals and 3D points. Run step *3)* from the previous algorithm.

Outputs: Points and normals on 3D surfaces.

5.3 Complexity Analysis

We discuss the complexity of the algorithm under the infinitesimal planarity assumption and in the general case. For both solutions, we assume that the warps are provided. We calculate the warps using schwarps (Pizarro et al., 2016), that impose the Schwarzian equations (16) as a soft constraint. However we would like to point out that we do not require warps between all possible pairs of images in the sequence. We use a reference view and thus require the computation of $N-1$ warps only, not N^2.

5.3.1 Solution with infinitesimal planarity

The solution to Iso-NRSfM under infinitesimal planarity solves only one sextic polynomial for N images. This polynomial is formed by computing the sum-of-squares of $2(N-1)$ cubic polynomials (29). Forming this sextic polynomial has a linear complexity but solving it is independent of N.

5.3.2 General case

The solution to the general case solves for the first and the second order derivatives of $\alpha_i(\mathbf{x})$ in parts. The approach for solving for the first order derivatives of $\alpha_i(\mathbf{x})$ is similar to the solution under infinitesimal planarity assumption. It also solves one sextic polynomial and therefore, the solution is independent of N. For the second order derivatives of $\alpha_i(\mathbf{x})$, we obtain $4(N-1)$ linear equations and they are solved using LLS. Therefore, there is a linear complexity in forming these equations and solving them as well.

6 Experimental Results

We tested Iso-NRSfM on synthetic and real datasets. Fig. 3 shows some images from real datasets on which the methods were evaluated. For quantitative comparison, we measured the normal error (mean difference between computed and ground truth normals in degrees) and the depth error (mean difference between computed and ground truth 3D coordinates in mm). We denote Iso-NRSfM with infinitesimal planarity as **infP** and as **iso** otherwise. We compared our method with six other NRSfM methods: **diffH** (Chhatkuli et al., 2014), **mdhI** (Chhatkuli et al., 2016), **kerF** (Gotardo and Martinez, 2011), **plaH** (Varol et al., 2009), **pieceI** (Taylor et al., 2010), **inextI** (Vicente and Agapito, 2012). All the codes for these methods were obtained from the authors' websites except **plaH** which we re-implemented.

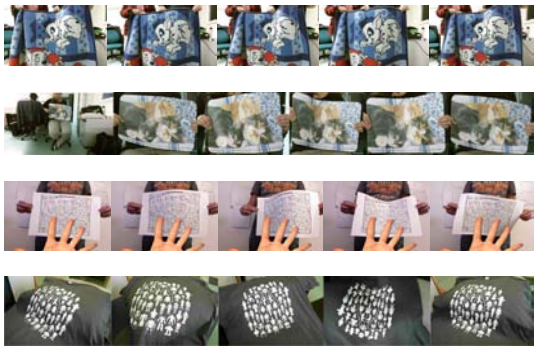

Fig. 3: Some images of the rug, table mat, kinect paper and tshirt datasets. The five rightmost images of the table mat dataset are zoomed in to improve visibility.

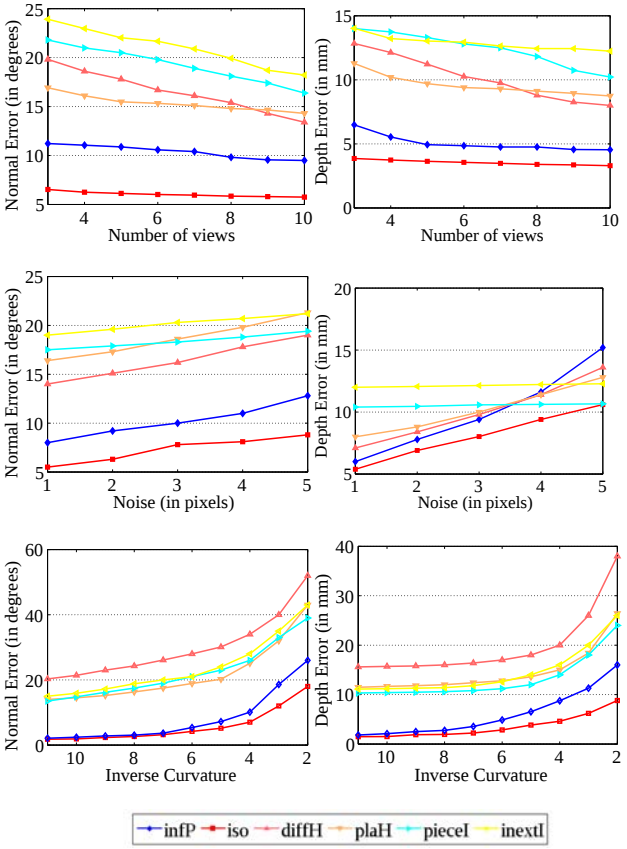

Fig. 4: Synthetic data experiments. Average normal and depth errors with respect to number of views, noise and curvature. Best viewed in colour.

6.1 Experiments with Synthetic Data

We simulated random views of a cylindrical surface deforming isometrically. The image size is $640p \times 480p$ and the focal length is $400p$. We tracked 400 points. We compared all methods by varying the number of views and noise in the images. **kerF** needs a temporal sequence, 10 views are not enough for reconstruction especially for short-baseline viewpoints. It therefore did not do well. Also, **mdhI** needs the views to be very different and therefore, it also gave very bad results on this sequence. (Chhatkuli et al., 2016) mentioned that their method fails on such sequences. The results are shown in Fig. 4. The results are obtained after averaging the errors over 50 trials (the default is $1p$ noise and 10 views).

6.1.1 *Varying the Number of Views*

infP gives a very good reconstruction for three views which improves when more images are added. **iso** performs much better than **infP** as it does not assume infinitesimal planarity; this helps in reconstructing the high curvature deformations more accurately. The errors of **iso** are almost half of **infP**. **plaH** and **diffH** give higher errors as compared to **infP** and **iso** with **plaH** being better than **diffH**. However, **diffH** improves faster with the number of views and gives better results than **plaH** for 8-10 views. The performance of these methods is off by almost 5 degrees as compared with **infP**. **pieceI** and **inextI** gives decent results only with 8-10 views but their performance is worst amongst the compared methods. Overall, **infP** and **iso** consistently show lower errors than all other methods.

6.1.2 *Varying Noise*

For the 10 images of the synthetic dataset, we observe that all methods degrade linearly when noise varying between 1-5 pixels is added. **inextI** and **pieceI** show a good tolerance to noise, even though their performance is worse than all other methods. **diffH** and **plaH** give a slightly better performance than **inextI** and **pieceI**. Their performance degrades faster with noise as compared to **inextI** and **pieceI**. Even though the normal errors of **diffH** and **plaH** are comparable to **inextI** and **pieceI**, their depth errors are lower because they smooth normals while calculating the depth. **infP** and **iso** give best performance with noise. **iso** performs better than **infP** and degrades more slowly than **infP**. The normal error for **infP** and **iso** is almost half as compared to other methods.

We also made an experiment to study the influence of noise on the warps. This is because we use warps to represent the image transformation and that estimating these could be reducing some of the noise applied to the correspondences. We simulated two synthetic images (I1 and I2) and added a 1-5 pixels noise to point correspondences in I2. We computed the schwarps (using B-splines) between I1 and I2 using our default setup (30 control points and a fixed value for the hyperparameter controlling the schwarps smoothing). We then computed the standard deviation of the pixel noise after computing the warps. The result is given in the last row of table 1, where we can see that the amount of noise is almost unaffected. We did the same experiment with fewer control points, which resulted in improved noise values (see the table 1). However, we found that using fewer points has an impact on the final accuracy as they degrade the derivatives, which directly influence the output of our method. We conclude that the warps do not remove the noise from the point correspondences.

6.1.3 *Varying Curvature*

We simulated a cylindrical surface with a varying radius. The curvature is the inverse of the radius. We simulated 10 surfaces with the radius varying from 2 to 11 and used 10 different views of each surface for the experiment. We see that the performance of **iso** and **infP** is best amongst the compared methods. Their performance is very similar when

Noise (in pixels)→ B-spline control points	1	2	3	4	5
10	.53	.92	1.55	1.94	2.46
20	.99	2.00	2.94	3.93	4.90
30	1.00	2.06	3.02	4.07	5.14

TABLE 1: Performance of warps in noisy conditions. The warps reduce the noise only when fewer control points are used, but then significantly degrade their derivatives.

the surfaces are almost flat or less bent. As the curvature of the surfaces increase, we can see that **iso** performs much better than **infP**. **iso** handles high deformations better than **infP** because it estimates the second order derivatives of the surface while **infP** assumes them to be zero. **diffH** and **plaH** need wide baseline views with different deformations, therefore, their performance is worse or comparable with **inextI** and **pieceI**.

6.2 Experiments with Real Data

We conducted experiments with the four datasets shown in Fig. 3 and performed two types of experiments. The tshirt dataset is a wide baseline dataset and the rest of them are short baseline datasets. Our observations are summarised below.

6.2.1 Experiments on Short Sequences

The rug, table mat and kinect paper datasets are long sequences (159, 60 and 191 images with 350, 300 and 1500 point correspondences) and the t-shirt is a short dataset (10 images, 85 point correspondences). The long sequences are uniformly reduced by picking 10 images at regular intervals in the sequence. The results are shown in Fig. 5. The results for the tshirt dataset are averaged over 20 trials of randomly sampled images. The figure clearly shows that **iso** works best among the compared methods while the performance of **mdhI** and **infP** is quite good as well.

6.2.1.1 **Rug dataset**: **iso** and **infP** have the best performance on this dataset; **iso** being better than **infP**. **mdhI** improves with the number of views and gives comparable results to **infP** for 7-10 views. **diffH** and **plaH** show a worse performance than **mdhI**, with **plaH** being better. **inextI** and **pieceI** are orthographic methods and therefore, they did not do well on this dataset as it has too much perspective in the deformations. **inextI**'s depth error is comparable to **plaH** but normal error is much higher. This indicates flattening of surfaces during the reconstruction.

6.2.1.2 **Table mat dataset**: **iso** has the best performance on this dataset. **infP** and **mdhI** show a similar performance in this dataset, and they are closest to **iso** as compared with other methods. **diffH** and **plaH** show a worse performance than **mdhI**, with **plaH** being a little bit better than **diffH**. **inextI** and **pieceI** are orthographic methods and therefore, they did not do well.

6.2.1.3 **Kinect paper dataset**: **mdhI** has the best performance on this dataset. **infP** and **iso** are very close to **mdhI**. **inextI** and **pieceI** have the worst normal error as compared to other methods. Their errors are almost twice as the best performing methods **mdhI**, **infP** and **iso**. **diffH** and **plaH** show better performance in terms of normals as compared to **inextI** and **pieceI**. The performance of **diffH**, **plaH**, **inextI** and **pieceI** in terms of depth error is similar.

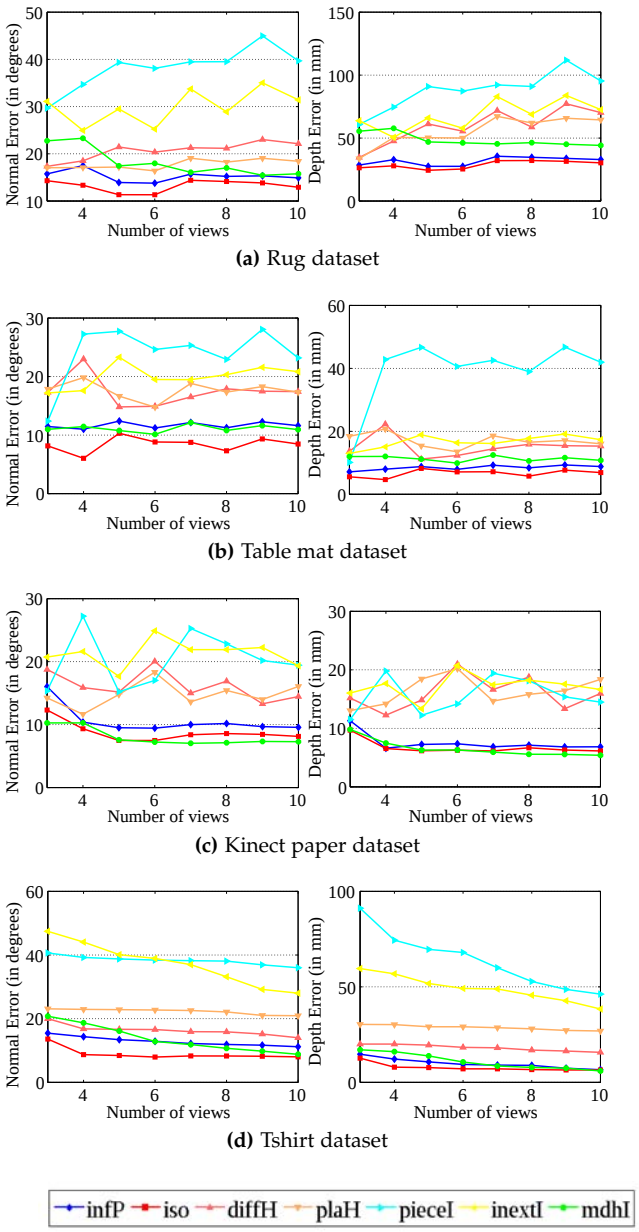

Fig. 5: Experiments on short sequences. The average normal and depth error for each experiment with number of views varying between 3-10 is shown. The views of the rug, table mat and kinect paper datasets are selected by uniform sampling the long sequences. The views of the tshirt dataset are selected by randomly sampling the dataset and the results are averaged over 20 trials. Best viewed in colour.

6.2.1.4 **Tshirt dataset**: **iso** has the best performance on this dataset with **infP** and **mdhI** being very close to **iso**. **diffH** is slightly worse than **infP** and **mdhI**. **plaH** follows a similar trend as **diffH**, but its performance is worse. **inextI** and **pieceI** have poor results on this dataset because they cannot handle such deformations.

6.2.1.5 **Summary of experiments on short sequences**: **iso** and **infP** give the best performance on the rug and table mat datasets while **mdhI** gives best results on the kinect paper dataset. It is important to note that **iso** and **infP** converge quickly as compared to the rest of the

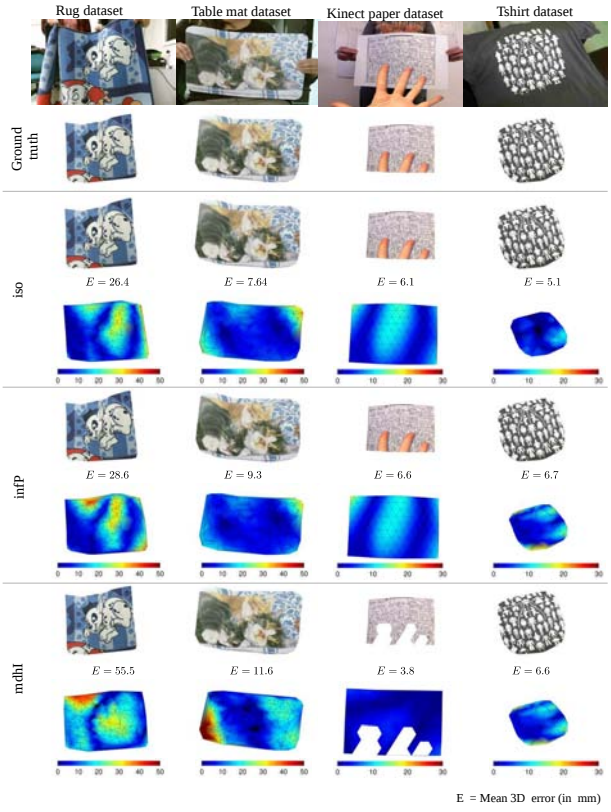

Fig. 6: Reconstruction error maps and renderings for the rug, table mat, kinect paper and tshirt datasets. We remind that **mdhI** reconstructs only the visible part of the surface. Therefore, the rendering and error map for the kinect paper dataset is broken for this method.

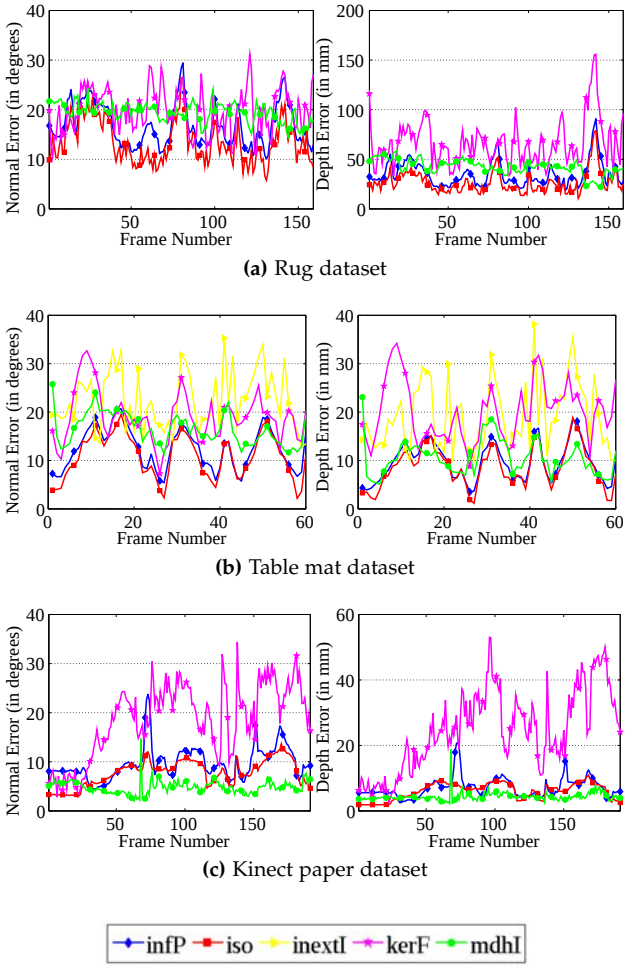

Fig. 7: Experiments on long sequences. The average normal and depth error for each frame is shown. The experiment with **inextI** is only performed for the table mat dataset. Best viewed in colour.

methods. They show very good results for as few as three views. **iso** converges much quicker than **infP**; the errors seem to have been stabilised with four views only. Fig. 6 shows 3D reconstruction error maps for the rug, table mat, kinect paper and tshirt datasets. We showed results for **iso**, **infP** and **mdhI** only because they are the most competitive methods amongst the compared methods. In case of occlusion, **iso** and **infP** can reconstruct the entire surface as long as the surface is visible in at least three images, therefore, even if the kinect paper dataset is occluded by a hand, it can be reconstructed by them. However, **mdhI** reconstructs only the visible part of each surface. Therefore we observe the occlusion in the reconstruction and rendering as well.

6.2.2 Experiments on Long Sequences

The rug, table mat and kinect paper datasets are long sequences with 60, 159 and 193 images. Since our method can easily handle a large number of images, it is important to show results on large sequences by considering all images in the dataset. A limitation of current NRSfM methods is that they cannot handle a large number of views. Also, several NRSfM methods such as **diffH** and **plaH** (Chhatkuli et al., 2014; Varol et al., 2009) reconstruct the reference image only and are computationally expensive to recover the other shapes. **kerF** reconstructs the entire image set in one execution and therefore, we compare our method with **kerF** on long sequences. The cat dataset is a relatively short sequences (60 images) therefore, we added the results of **inextI** for this dataset on the entire sequence. Fig. 7 shows the comparison of our method with others. It is very clearly visible that our method performs much better than the compared methods. **mdhI** and **infP** show good results as well. Table 2 summarises the results of these methods.

6.2.2.1 **Rug dataset**: **iso** gives the best results among the compared methods on this dataset. **infP**'s performance is slightly worse than **iso**. **mdhI** shows better performance than **kerF**, but it is worse than **iso** and **infP**.

6.2.2.2 **Table mat dataset**: **iso** has the best performance with **infP** being very close to it. **mdhI** shows better performance than **kerF**, but both of them are worse than **iso** and **infP**. **pieceI** has the worst performance amongst the compared methods. **diffH** and **plaH** need to compute homographies between image pairs, therefore, they grow non-linearly with the number of views. For 60 images, the execution time goes upto 45 min for a single reconstruction. Therefore, we did not compare with them. **pieceI** breaks on this sequence, therefore we did not include it. The comparison with **inextI** is done only for this sequence as it is a relatively smaller sequence. One must also note that **inextI, pieceI** grow with the number of views and point correspondences, therefore, they are not very efficient with a large number of views.

Fig. 8: Images $(10, 20, 30, 40, 50)$ of a partially stretched rubber like surface. The first image has the least deformation and the last one has the most.

	Rug		Table mat		Kinect paper	
Methods	E_n	E_d	E_n	E_d	E_n	E_d
iso	12.9	27.2	10.5	8.2	7.8	6.1
infP	16.7	34.9	12.3	9.6	9.6	7.1
mdhI	18.8	42.3	16.4	10.5	4.8	4.4
kerF	20.0	66.6	19.0	20.0	18.7	24.8
inextI	-	-	22.4	19.0	-	-

TABLE 2: Summary of experiments on long sequences. The average normal (E_n) and depth error (E_d), are measured in degrees and mm respectively and shown over the entire sequences.

6.2.2.3 **Kinect paper dataset**: **mdhI** shows the best results on this dataset. **iso** and **infP** have similar performance but they are worse than **mdhI**. **kerF** has the worst performance; it is at least twice as worse than the rest of the methods. It is because this sequence has outliers; and therefore the performance of **kerF** is affected. One must note that **iso**, **infP** and **kerF** reconstruct the occluded part of the paper while **mdhI** only recovers the visible paper in each frame.

6.2.2.4 **Summary of experiments on long sequences** : Table 2 summarises the results of the compared methods on the rug, table mat and kinect paper datasets. **iso** and **infP** give the best performance on rug and the table mat datasets while **mdhI** gives the best results on the kinect paper dataset. **kerF** gives decent results on the rug and table mat datasets but does not do well on the kinect paper dataset. However, its performance is always worse than **iso**, **infP** and **mdhI**. **inextI** was only compared on the table mat dataset; it gives the worst results as compared to the other methods.

6.2.2.5 **Summary of experiments**: In the experiments that we performed, we observed that **iso** and **infP** show the best results amongst all compared methods. **mdhI** shows a good performance. Its results are comparable to ours for the tshirt and kinect paper datasets. This method is based on the maximum depth heuristic (MDH), it usually requires a lot of images with different viewpoints to give good results. **diffH** and **plaH** are based on homography decomposition. They suffer from ambiguities in the normals. They disambiguate the normals assuming the smoothness of the surfaces which is not a strong assumption to make. These methods work well with wide baseline datasets. **kerF** is a method based on statistical modeling designed to work for video sequences. It needs a good estimation of the radius of the kernel in which the similarities between the two shapes are measured. **pieceI** and **inextI** are methods based on orthographic projection. They suffer from convex-concave flip ambiguities.

Experiment	1	2	3	4	5
No. of images	10	20	30	40	50
infP	19.1	26.6	29.3	32.7	36.1
iso	14.1	22.7	27.0	28.3	34.8

TABLE 3: Mean shape error (in degrees) for the experiment with partially stretched surface. The error increases with the number of images as the deformation increases. **iso** performs better than **infP**.

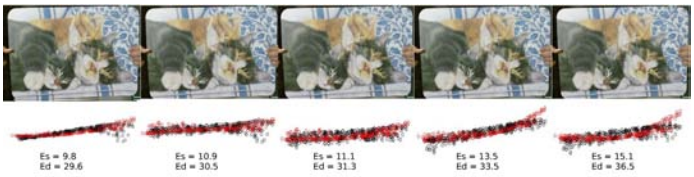

Fig. 9: Experiment with an almost stationary object. The first five images of the table mat sequence are used. The reconstruction of **iso** is shown in red. The ground truth is indicated with black. E_s represents the mean shape error (in degrees). E_d represents the mean depth error (in mm). The performance of **iso** is almost the same as **infP** on these five images.

6.3 Experiment with an Elastic Object

Our methods model deformations with isometry. In case of non-isometric deformations, theorem 2 and corollary 1 do not hold and therefore, there is no meaningful theoretical solution guaranteed. However, we solve Iso-NRSfM by finding a set of CS that minimise the sum of squares of polynomials in equation (29). Therefore, we made an experiment to test Iso-NRSfM with objects deforming non-isometrically in order to test the limitation for our methods **infP** and **iso**. We used the partially stretched surface dataset introduced in (Ozgur and Bartoli, 2016). It consists of 50 shapes of an elastic surface partially stretched from its longest side in a sequential order. The images are shown in Fig. 8. We made five experiments on this dataset. These experiments include 10, 20, 30, 40 and 50 images respectively. The experiment with 10 images has the least elastic deformation (this can be seen in Fig. 8) and the one with 50 images has the most elastic deformation. For each experiment, we calculate the shape error for each image. The mean shape error calculated over the entire image set in each experiment increases about linearly with the degree of extension. Therefore, we see that the method does not collapse completely for non-isometric deformations but gracefully degrades. Table 3 summarises the results of this experiment.

6.4 Computation Time Comparison

We compared the performance of our methods with the others in terms of computation time on a standard computer with 8GB RAM. Ours and the rest of the methods are implemented in MATLAB. **infP** takes $\approx$ 10 seconds for any number of images. **iso** is initialised with **infP**, and the solution to the first and the second order derivatives of $\alpha_i(\mathbf{x})$ is found iteratively upto 5 iterations. Solving for the first order derivatives takes a similar duration as **infP** while solving for second order derivatives and integrating normals have a linear complexity but they are very fast. We made an experiment with 10, 30 and 60 views. We observed that the computation time of **iso** ($\approx$ 60 seconds) and **infP** ($\approx$10 seconds) is almost the same (very small increase) in the three experiments. **mdhI** and **kerF** also are very fast but

	infP	iso	mdhI	kerF	plaH	diffH	pieceI	inextI
10	11.2	51	**8.9**	14.1	65	86	180	210
30	**13.1**	52.4	21.3	26.3	3090	2280	850	1299
60	**14.8**	55.1	65.8	44.8	-	-	-	-

TABLE 4: Comparison of computation time (in seconds) for 10, 30 and 60 views. The best performing method is highlighted in bold. **infP** and **iso** show a much lower increase in computation time while **plaH**, **diffH**, **pieceI** and **inextI** show a drastic increase. They could not be evaluated for 60 views.

the computation time increases significantly as the number of images increases. **plaH**, **diffH**, **pieceI** and **inextI** show a drastic increase in computation time on changing the number of images from 10 to 30. We did not compute these timings for 60 images. Table 4 summarises the results.

6.5 Experiment with an Almost Stationary Object

We made an experiment with an almost stationary object. In the table mat dataset, we picked the first 5 frames which are shown in Fig. 9. The mat is almost stationary. We observed that our methods **iso** and **infP** did get a decent reconstruction for these datasets. The errors are higher as compared to the experiments with the full sequence, as expected. Fig. 9 shows the images and the reconstruction. The results are shown for **iso**. The performance of **infP** was almost the same as **iso**. This shows that the solution to Iso-NRSfM is well-posed. There is always a solution for $N \geq 3$, as long as the images are not exactly the same.

7 Conclusions

We proposed a theoretical framework for modelling and solving NRSfM locally for surfaces deforming isometrically. It uses Riemannian manifolds and applies to the minimal and redundant cases of $N \geq 3$ views. Unlike existing methods, the proposed method has only five variables to solve for N views. Therefore, it can handle a large number of views. The complexity is linear, which is a substantial improvement from the current state-of-the-art. We tested our method on datasets with wide-baseline and short-baseline viewpoints, large and small deformations. Our results show that the proposed methods consistently give significantly better results than the state-of-the-art methods even for as few as three views. For future work, we will explore the possibility of extending this framework to non-isometric deformations.

Acknowledgements. This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE, the Spanish Ministry of Economy, Industry and Competitiveness under project ARTEMISA (TIN2016-80939-R).

References

A. Agudo and F. Moreno-Noguer. Simultaneous pose and non-rigid shape with particle dynamics. In *CVPR*, 2015.

A. Agudo, F. Moreno-Noguer, B. Calvo, and J. Montiel. Sequential non-rigid structure from motion using physical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1–1, 2015.

I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*, 2009.

A. Bartoli, Y. Gerard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, 2015.

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

C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, 2000.

A. Chhatkuli, D. Pizarro, and A. Bartoli. Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In *BMVC*, 2014.

A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. Inextensible non-rigid shape-from-motion by second-order cone programming. In *CVPR*, 2016.

Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014.

P. Gotardo and A. Martinez. Kernel non-rigid structure from motion. In *ICCV*, 2011.

R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *ECCV*. 2008.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000. ISBN 0521623049.

D. Henrion and J. B. Lasserre. Gloptipoly: Global optimization over polynomials with matlab and sedumi. *ACM Transactions on Mathematical Software (TOMS)*, 29(2):165–194, 2003.

J. Lee. *Riemannian manifolds : an introduction to curvature.* Springer, 1997. ISBN 0-387-98322-8.

J. Lee. *Introduction to Smooth Manifolds.* Springer, 2003. ISBN 0-387-95448-1.

E. Ozgur and A. Bartoli. Particle-SfT: a Provably-Convergent, Fast Shape-from-Template Algorithm. *International Journal of Computer Vision*, 123(2):184–205, 2016.

S. Parashar, D. Pizarro, and A. Bartoli. Isometric non-rigid shape-from-motion in linear time. In *CVPR*, 2016.

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

D. Pizarro, R. Khan, and A. Bartoli. Schwarps: Locally projective image warps based on 2D schwarzian derivatives. *International Journal of Computer Vision*, 119(2):93–109, 2016.

C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011.

C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3D reconstruction of dynamic scenes. In *ECCV*. 2014.

M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.

T. Sasaki and M. Yoshida. Schwarzian derivatives and uniformization. *CRM Proc Lecture Notes AMS*, (672):271–286, 2002.

J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, 2010.

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

A. Varol, M. Salzmann, E. Tola, and F. P. Template-free monocular reconstruction of deformable surfaces. In *CVPR*, 2009.

S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *ECCV*, 2012.

8 Biographies

Shaifali Parashar received her Msc degree in Computer Vision from the University of Burgundy in 2014. She is currently a PhD candidate in Computer Vision at Université d'Auvergne under Prof. Adrien Bartoli and Dr. Daniel Pizarro. Her research interest is non-rigid 3D reconstruction.

Daniel Pizarro Pérez has been an Associate Professor at the Universidad de Alcalá (Spain) since 2012. He is a member of the GEINTRA group and an invited member in the ALCoV-ISIT group at Université d'Auvergne. His research interests are in Computer Vision, including image registration, deformable reconstruction and their applications to Minimally Invasive Surgery.

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.

APPENDIX A
METRIC TENSORS

In order to describe a physical surface, one must define a coordinate system where measurements like lengths, angles and areas can be defined. The metric tensor (Lee, 1997) is a function which is defined on a physical surface to obtain these measurements. We use a simple example to develop a better understanding of the metric tensor. For this purpose, we consider an infinitesimal vector $\vec{v}$ in the Euclidean 3-space.

If we use a Cartesian coordinate system, we can define the length ds of $\vec{v}$ using Pythagoras' law for distances as:

$$ds^2 = dx^2+dy^2+dz^2 = \begin{pmatrix} dx & dy & dz \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}, \tag{34}$$

where dx, dy and dz represent the components of $\vec{v}$ in x, y and z coordinates respectively and the identity matrix in equation (34) is the metric tensor. The metric tensor is denoted as $\mathbf{g}$. The identity metric tensor implies that the distances remain constant as one moves along the coordinate frame. Now if we measure the length of the same infinitesimal vector $\vec{v}$ in a spherical coordinate system (r, θ, ϕ), we have:

$$\begin{aligned} ds^2 &= dr^2 + r^2 d\theta^2 + r^2 \sin^2\theta d\phi^2 \\ &= \begin{pmatrix} dr & d\theta & d\phi \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & r^2 \sin^2\theta \end{pmatrix} \begin{pmatrix} dr \\ d\theta \\ d\phi \end{pmatrix}, \end{aligned} \tag{35}$$

Here, the metric tensor $\mathbf{g}$ is not the identity and changes at each point. This is necessary in order to measure the same distances at various locations in the coordinate frame. Fig. 10 shows two points A and B represented on a spherical coordinate system. Moving these points to A' and B', in the direction of the $r-$coordinate, will also change the distance between them. However, this does not happen in a Cartesian coordinate system. On further elaboration, we see that the

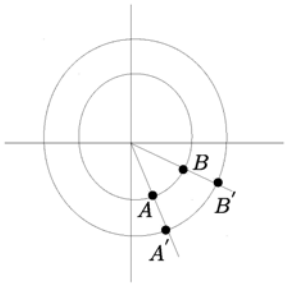

Fig. 10: Translating points A and B in spherical coordinates.

change of coordinates from Cartesian to spherical (described by the transformation function f), results in the following expression for the metric tensor:

$$(x, y, z) = f(r, \theta, \phi) \quad \mathbf{g} = \mathbf{J}_f^T \mathbf{J}_f \tag{36}$$

In this case $\mathbf{g}$ depends only on $\mathbf{J}_f$ as the metric tensor in the Cartesian coordinate system is the identity.

Moving on to the application of this theory in our work, we have a surface $\mathcal{M}_i$ (see Fig. 2) undergoing an isometric deformation which leads to $\mathcal{M}_j$. A point $\mathbf{z} \in \mathcal{M}_i$ becomes $\mathbf{w} \in \mathcal{M}_j$. Since ψ_{ij} is an isometric deformation, the infinitesimal distances around $\mathbf{w}$ will be the same as that of $\mathbf{z}$, as both of them represent the same point on different isometric surfaces. This is analogous to equations (34) and (35).

APPENDIX B
CHRISTOFFEL SYMBOLS (CS)

In the previous section we saw that the metric tensor in a Cartesian coordinate system (equation (34)) is written as an identity matrix, but this may not be true for the metric tensors in other coordinate systems (equation (35)). In such coordinate systems, since the metric tensor keeps on changing with the coordinates, we can define its change using CS. Therefore, CS is a set of numbers which are the components of vectors that represent the change in metric tensor. They are defined as the CS of first kind ($\mathbf{\Gamma}_{cab}$) and the CS of second kind ($\mathbf{\Gamma}^d_{ab}$) related by the expression $\mathbf{\Gamma}_{cab} = \mathbf{g}_{cd}\mathbf{\Gamma}^d_{ab}$. Therefore it is very easy to recover the CS of one kind given the other ones. In our work, we found that the expressions of the CS of second kind were simpler and therefore, we use only these. In the Cartesian coordinate system, the metric tensor is the identity and therefore, all the CS of second kind are zero. However, in the spherical coordinate system, the metric tensor is variable and the CS of second kind are:

$$\begin{aligned} &\mathbf{\Gamma}^r = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -r & 0 \\ 0 & 0 & -r\sin^2\theta \end{pmatrix} \quad \mathbf{\Gamma}^\theta = \begin{pmatrix} 0 & r^{-1} & 0 \\ r^{-1} & 0 & 0 \\ 0 & 0 & -\sin\theta\cos\theta \end{pmatrix} \\ &\mathbf{\Gamma}^\phi = \begin{pmatrix} 0 & 0 & r^{-1} \\ 0 & 0 & \cot\theta \\ r^{-1} & \cot\theta & 0 \end{pmatrix}. \end{aligned} \tag{37}$$

They are expressed in terms of the metric tensor and its first order derivatives. Therefore, we can define them at various surfaces and use them in our framework, just like the metric tensor. The CS are given by:

$$\mathbf{\Gamma}^p_{mn} = \frac{1}{2}\mathbf{g}^{pl}\left(\mathbf{g}_{lm,n} + \mathbf{g}_{ln,m} - \mathbf{g}_{mn,l}\right), \tag{38}$$

where $\mathbf{g}_{lm,n} = \partial_n \mathbf{g}_{lm}$ and $\mathbf{g}^{pl} = (\mathbf{g}_{pl})^{-1}$. We write the derivatives of the metric tensor for the spherical coordinate system as:

$$\begin{aligned} &\partial_r \mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2r & 0 \\ 0 & 0 & 2r\sin^2\theta \end{pmatrix} \quad \partial_\theta \mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & r^2\sin 2\theta \end{pmatrix} \\ &\partial_\phi \mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{39}$$

Given $\mathbf{g}^{pl} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^{-2} & 0 \\ 0 & 0 & r^{-2}\sin^{-2}\theta \end{pmatrix}$ and $\mathbf{g}_{lm,n}$ (obtained in equation (39)), we obtain the CS of the spherical coordinate

system given in equation (37) using equation (38). For example, $\mathbf{\Gamma}^r_{\theta\theta}$, according to equation (38) is given by

$$\begin{aligned}\mathbf{\Gamma}^r_{\theta\theta} &= \frac{1}{2}\mathbf{g}^{rr}\left(\mathbf{g}_{r\theta,\theta}+\mathbf{g}_{r\theta,\theta}-\mathbf{g}_{\theta\theta,r}\right)\\ &+\frac{1}{2}\mathbf{g}^{r\theta}\left(\mathbf{g}_{\theta\theta,\theta}+\mathbf{g}_{\theta\theta,\theta}-\mathbf{g}_{\theta\theta,\theta}\right)\\ &+\frac{1}{2}\mathbf{g}^{r\phi}\left(\mathbf{g}_{\phi\theta,\theta}+\mathbf{g}_{\phi\theta,\theta}-\mathbf{g}_{\theta\theta,\phi}\right)\\ &=\frac{1}{2}\left(0+0-2r\right)+0+0=-r.\end{aligned} \tag{40}$$

APPENDIX C
THEOREMS AND THEIR PROOFS

Theorem 1. *Let $\mathbf{x}\in\mathcal{I}_i$, then $\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]$ is given by:*

$$\begin{aligned}\mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] &= \frac{-1}{\alpha_i}\begin{pmatrix}2\alpha_{i,1} & \alpha_{i,2}\\ \alpha_{i,2} & 0\end{pmatrix}+\frac{\left(\alpha_i\right)^2 A_i}{D_i}\begin{pmatrix}\alpha_{i,11} & \alpha_{i,12}\\ \alpha_{i,12} & \alpha_{i,22}\end{pmatrix}\\ \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] &= \frac{-1}{\alpha_i}\begin{pmatrix}0 & \alpha_{i,1}\\ \alpha_{i,1} & 2\alpha_{i,2}\end{pmatrix}+\frac{\left(\alpha_i\right)^2 B_i}{D_i}\begin{pmatrix}\alpha_{i,11} & \alpha_{i,12}\\ \alpha_{i,12} & \alpha_{i,22}\end{pmatrix},\end{aligned} \tag{41}$$

where $\alpha_{i,k}=\dfrac{\partial\alpha_i}{\partial x^k}$, $\alpha_{i,nm}=\dfrac{\partial^2\alpha_i}{\partial x^n\partial x^m}$ and:

$$\begin{aligned}A_i &= -x^1\alpha_i+\left(1+\left(x^1\right)^2\right)\alpha_{i,1}+x^1x^2\alpha_{i,2}\\ B_i &= -x^2\alpha_i+\left(1+\left(x^2\right)^2\right)\alpha_{i,2}+x^1x^2\alpha_{i,1}\\ D_i &= \left(\alpha_i-x^1\alpha_{i,1}-x^2\alpha_{i,2}\right)^2+\left(\alpha_{i,1}\right)^2+\left(\alpha_{i,2}\right)^2.\end{aligned} \tag{42}$$

Proof. From the definition of $\phi_i(\mathbf{x})$ in equation (4), we can write the Jacobian matrix of $\phi_i(\mathbf{x})$ as:

$$\mathbf{J}_{\phi_i(\mathbf{x})}=\frac{1}{\alpha_i^2}\begin{pmatrix}\alpha_i-x^1\alpha_{i,1} & -x^1\alpha_{i,2}\\ -x^2\alpha_{i,1} & \alpha_i-x^2\alpha_{i,2}\\ -\alpha_{i,1} & -\alpha_{i,2}\end{pmatrix}. \tag{43}$$

Next we compute the metric tensor by substituting the Jacobian matrix from equation (43) in equation (6). The metric tensor is given by

$$\begin{aligned}\mathbf{g}_{11}[\phi_i(\mathbf{x})] &= \frac{1}{\alpha_i^4}\left(\epsilon^2\left(\alpha_{i,1}\right)^2+\left(\alpha_i\right)^2-2x^1\alpha_i\alpha_{i,1}\right)\\ \mathbf{g}_{12}[\phi_i(\mathbf{x})] &= \frac{1}{\alpha_i^4}\left(\epsilon^2\alpha_{i,1}\alpha_{i,2}-x^1\alpha_i\alpha_{i,2}-x^2\alpha_i\alpha_{i,1}\right)\\ \mathbf{g}_{22}[\phi_i(\mathbf{x})] &= \frac{1}{\alpha_i^4}\left(\epsilon^2\left(\alpha_{i,2}\right)^2+\left(\alpha_i\right)^2-2x^2\alpha_i\alpha_{i,2}\right).\end{aligned} \tag{44}$$

where $\epsilon^2=1+\left(x^1\right)^2+\left(x^2\right)^2$. The inverse of metric tensor is given by

$$\begin{aligned}\mathbf{g}^{11}[\phi_i(\mathbf{x})] &= \frac{\mathbf{g}_{22}[\phi_i(\mathbf{x})]}{\det(\mathbf{g}[\phi_i(\mathbf{x})])}=\frac{\left(\alpha_i\right)^8\mathbf{g}_{22}[\phi_i(\mathbf{x})]}{D_i}\\ \mathbf{g}^{12}[\phi_i(\mathbf{x})] &= \frac{-\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{\det(\mathbf{g}[\phi_i(\mathbf{x})])}=\frac{-\left(\alpha_i\right)^8\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{D_i}\\ \mathbf{g}^{22}[\phi_i(\mathbf{x})] &= \frac{\mathbf{g}_{11}[\phi_i(\mathbf{x})]}{\det(\mathbf{g}[\phi_i(\mathbf{x})])}=\frac{\left(\alpha_i\right)^8\mathbf{g}_{11}[\phi_i(\mathbf{x})]}{D_i}.\end{aligned} \tag{45}$$

The derivatives of the metric tensor are given by

$$\begin{aligned}\mathbf{g}_{11,1}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})]+\frac{2E_i\alpha_{i,11}}{\left(\alpha_i\right)^4}\\ \mathbf{g}_{12,1}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{12}[\phi_i(\mathbf{x})]-\frac{H_i}{\left(\alpha_i\right)^4}+\frac{E_i\alpha_{i,12}+F_i\alpha_{i,11}}{\left(\alpha_i\right)^4}\\ \mathbf{g}_{22,1}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{22}[\phi_i(\mathbf{x})]+\frac{2L_i}{\left(\alpha_i\right)^4}+\frac{2F_i\alpha_{i,12}}{\left(\alpha_i\right)^4}\\ \mathbf{g}_{11,2}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,2}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})]+\frac{2H_i}{\left(\alpha_i\right)^4}+\frac{2E_i\alpha_{i,12}}{\left(\alpha_i\right)^4}\\ \mathbf{g}_{12,2}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,2}}{\alpha_i}\mathbf{g}_{12}[\phi_i(\mathbf{x})]-\frac{L_i}{\left(\alpha_i\right)^4}+\frac{E_i\alpha_{i,22}+F_i\alpha_{i,12}}{\left(\alpha_i\right)^4}\\ \mathbf{g}_{22,2}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,2}}{\alpha_i}\mathbf{g}_{22}[\phi_i(\mathbf{x})]+\frac{2F_i\alpha_{i,22}}{\left(\alpha_i\right)^4},\\ \text{where } E_i &= \left(1+\left(x^1\right)^2+\left(x^2\right)^2\right)\alpha_{i,1}-x^1\alpha_i,\\ F_i &= \left(1+\left(x^1\right)^2+\left(x^2\right)^2\right)\alpha_{i,2}-x^2\alpha_i,\\ H_i &= x^2\left(\alpha_{i,1}\right)^2+\alpha_i\alpha_{i,2}-x^1\alpha_{i,1}\alpha_{i,2},\\ \text{and } L_i &= x^1\left(\alpha_{i,2}\right)^2\alpha_{i,1}\alpha_i+-x^2\alpha_{i,1}\alpha_{i,2}.\end{aligned} \tag{46}$$

According to equation (9), the CS are given by

$$\begin{aligned}&\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]=\\ &\frac{1}{2}\mathbf{g}^{p1}[\phi_i(\mathbf{x})]\left(\mathbf{g}_{1m,n}[\phi_i(\mathbf{x})]+\mathbf{g}_{1n,m}[\phi_i(\mathbf{x})]-\mathbf{g}_{mn,1}[\phi_i(\mathbf{x})]\right)+\\ &\frac{1}{2}\mathbf{g}^{p2}[\phi_i(\mathbf{x})]\left(\mathbf{g}_{2m,n}[\phi_i(\mathbf{x})]+\mathbf{g}_{2n,m}[\phi_i(\mathbf{x})]-\mathbf{g}_{mn,2}[\phi_i(\mathbf{x})]\right).\end{aligned} \tag{47}$$

Using the metric tensor and its derivative from equations (44) and (46) in the expression for the CS given in equation (47), gives the result in equation (41). For example, $\mathbf{\Gamma}^1_{11}[\phi_i(\mathbf{x})]$ is given by

$$\begin{aligned}\mathbf{\Gamma}^1_{11}[\phi_i(\mathbf{x})] &= \frac{1}{2}\mathbf{g}^{11}[\phi_i(\mathbf{x})]\left(\mathbf{g}_{11,1}[\phi_i(\mathbf{x})]\right)\\ &+\frac{1}{2}\mathbf{g}^{12}[\phi_i(\mathbf{x})]\left(2\mathbf{g}_{21,1}[\phi_i(\mathbf{x})]-\mathbf{g}_{11,2}[\phi_i(\mathbf{x})]\right)\\ &=\frac{\left(\alpha_i\right)^8\mathbf{g}_{22}[\phi_i(\mathbf{x})]}{D_i}\left(-\frac{2\alpha_{i,1}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})]+\frac{E_i\alpha_{i,11}}{\left(\alpha_i\right)^4}\right)\\ &-\frac{\left(\alpha_i\right)^8\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{D_i}\left(-\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{12}[\phi_i(\mathbf{x})]-\frac{2H_i}{\left(\alpha_i\right)^4}\right)\\ &-\frac{\left(\alpha_i\right)^8\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{D_i}\left(\frac{F_i\alpha_{i,11}}{\left(\alpha_i\right)^4}+\frac{2\alpha_{i,2}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})]\right)\\ &=\frac{-2\alpha_{i,1}}{\alpha_i}+\frac{\left(\alpha_i\right)^2A_i}{D_i}.\end{aligned} \tag{48}$$

□

Theorem 2. *Let ψ_{ij} be an isometric mapping between the manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$, then $\mathbf{g}_{mn}[\phi_j]=\mathbf{g}_{mn}[\phi_i\circ\eta_{ji}]$ with $(i,j)\in[1,N]\times[1,N]$.*

Proof. We first write ϕ_j in terms of ϕ_i using the isometric mapping ψ_{ij}:

$$\phi_j=\psi_{ij}\circ\phi_i\circ\eta_{ji}. \tag{49}$$

From equations (8) and (49) we have:

$$\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[(\psi_{ij}\circ\phi_i)\circ\eta_{ji}] = \frac{\partial x^s}{\partial y^m}\frac{\partial x^t}{\partial y^n}\mathbf{g}_{st}[\psi_{ij}\circ\phi_i]. \quad (50)$$

By definition, isometric mappings do not change the local metric and so $\mathbf{g}[\psi_{ij}\circ\phi_i] = \mathbf{g}[\phi_i]$, which applied to equation (50) gives:

$$\mathbf{g}_{mn}[\phi_j] = \frac{\partial x^s}{\partial y^m}\frac{\partial x^t}{\partial y^n}\mathbf{g}_{st}[\phi_i]. \quad (51)$$

Identifying equation (8) with equation (51) gives the sought equality $\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[\phi_i\circ\eta_{ji}]$. □

Corollary 1. *Let ψ_{ij} be an isometric mapping between the manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$, then $\mathbf{\Gamma}^p_{mn}[\phi_j] = \mathbf{\Gamma}^p_{mn}[\phi_i\circ\eta_{ji}]$ with $(i,j)\in[1,N]\times[1,N]$.*

Proof. As described in equation (9), $\mathbf{\Gamma}^p_{mn}[\phi_j]$ is a function of $\mathbf{g}_{mn}[\phi_j]$ and its derivatives. From Theorem 2 we have that $\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[\phi_i\circ\eta_{ji}]$. By multiplying this expression in both sides by $\mathbf{g}^{mn}[\phi_j]$ we have:

$$\mathbf{g}^{mn}[\phi_j]\mathbf{g}_{mn}[\phi_j] = \mathbf{g}^{mn}[\phi_j]\mathbf{g}_{mn}[\phi_i\circ\eta_{ji}] = \delta_{mn}, \quad (52)$$

from which we deduce that $\mathbf{g}^{mn}[\phi_j] = \mathbf{g}^{mn}[\phi_i\circ\eta_{ji}]$. Also, by differentiating $\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[\phi_i\circ\eta_{ji}]$ on both sides we have:

$$\partial_l\mathbf{g}_{mn}[\phi_j] = \partial_l\mathbf{g}_{mn}[\phi_i\circ\eta_{ji}], \quad (53)$$

giving $\mathbf{g}_{mn,l}[\phi_j] = \mathbf{g}_{mn,l}[\phi_i\circ\eta_{ji}]$. By substitution of these identities in equation (51) we obtain:

$$\mathbf{\Gamma}^p_{mn}[\phi_j] = \frac{1}{2}\mathbf{g}^{pl}[\phi_i\circ\eta_{ji}](\mathbf{g}_{lm,n}[\phi_i\circ\eta_{ji}] + \mathbf{g}_{ln,m}[\phi_i\circ\eta_{ji}] - \mathbf{g}_{mn,l}[\phi_i\circ\eta_{ji}]),$$

and thus the equality $\mathbf{\Gamma}^p_{mn}[\phi_j] = \mathbf{\Gamma}^p_{mn}[\phi_i\circ\eta_{ji}]$ holds. □

Theorem 3. *If $\mathcal{M}$ is a plane then its image embedding at $\mathbf{x}\in\mathcal{I}$ is $\phi(\mathbf{x}) = \beta(\mathbf{x})^{-1}(\mathbf{x}\quad 1)^\top$ with β a linear function.*

Proof. Suppose $\mathcal{M}$ is a plane described by the equation $\mathbf{n}^\top\mathbf{z}+d=0$, where $\mathbf{z} = \begin{pmatrix}z^1 & z^2 & z^3\end{pmatrix}^\top$ and $\mathbf{n}$ is the plane's normal. From equation (3), the embedding is expressed with a depth function $\phi(\mathbf{x}) = \rho(\mathbf{x})\begin{pmatrix}\mathbf{x} & 1\end{pmatrix}^\top$. By combining the depth parametrisation with the plane equation, we have:

$$\mathbf{n}^\top\rho(\mathbf{x})\begin{pmatrix}\mathbf{x} & 1\end{pmatrix}^\top + d = 0, \quad (54)$$

from which we compute ρ as:

$$\rho(\mathbf{x}) = \frac{-d}{\mathbf{n}^\top\begin{pmatrix}\mathbf{x} & 1\end{pmatrix}^\top}. \quad (55)$$

By defining $\beta(\mathbf{x}) = (\rho(\mathbf{x}))^{-1}$, ϕ is written as:

$$\phi(\mathbf{x}) = \beta(\mathbf{x})^{-1}(\mathbf{x}\quad 1)^\top. \quad (56)$$

□

Corollary 2. *Let $\mathcal{M}$ be a plane and $\phi(\mathbf{x})$ the image embedding at $\mathbf{x}\in\mathcal{I}$, the CS $\mathbf{\Gamma}^p_{mn}[\phi(\mathbf{x})]$ are given by:*

$$\mathbf{\Gamma}^1_{mn}[\phi(\mathbf{x})] = \frac{1}{\beta(\mathbf{x})}\begin{pmatrix}-2\beta_1(\mathbf{x}) & -\beta_2(\mathbf{x})\\ -\beta_2(\mathbf{x}) & 0\end{pmatrix}$$
$$\mathbf{\Gamma}^2_{mn}[\phi(\mathbf{x})] = \frac{1}{\beta(\mathbf{x})}\begin{pmatrix}0 & -\beta_1(\mathbf{x})\\ -\beta_1(\mathbf{x}) & -2\beta_2(\mathbf{x})\end{pmatrix}, \quad (57)$$

where $\beta_1(\mathbf{x}) = \frac{\partial\beta(\mathbf{x})}{\partial x^1}$ and $\beta_2(\mathbf{x}) = \frac{\partial\beta(\mathbf{x})}{\partial x^2}$.

Proof. From the definition of $\phi(\mathbf{x})$ in equation (56), we can write the Jacobian matrix of $\phi(\mathbf{x})$ as:

$$\mathbf{J}_\phi(\mathbf{x}) = \frac{1}{\beta(\mathbf{x})^2}\begin{pmatrix}\beta(\mathbf{x}) - x^1\beta_1(\mathbf{x}) & -x^1\beta_2(\mathbf{x})\\ -x^2\beta_1(\mathbf{x}) & \beta(\mathbf{x}) - x^2\beta_2(\mathbf{x})\\ -\beta_1(\mathbf{x}) & -\beta_2(\mathbf{x})\end{pmatrix}. \quad (58)$$

Using equation (6), the metric tensor at $\phi(\mathbf{x})$ can be written as $\mathbf{J}^\top_{\phi(\mathbf{x})}\mathbf{J}_{\phi(\mathbf{x})}$. The expression is given by

$$\begin{aligned}\mathbf{g}_{11}[\phi(\mathbf{x})] &= \frac{1}{\beta^4}\left(\epsilon^2(\beta_1)^2 + \beta^2 - 2x^1\beta\beta_1\right)\\ \mathbf{g}_{12}[\phi(\mathbf{x})] &= \frac{1}{\beta^4}\left(\epsilon^2\beta_1\beta_2 - x^1\beta\beta_2 - x^2\beta\beta_1\right)\\ \mathbf{g}_{22}[\phi(\mathbf{x})] &= \frac{1}{\beta^4}\left(\epsilon^2(\beta_2)^2 + \beta^2 - 2x^2\beta\beta_2\right).\end{aligned} \quad (59)$$

where $\epsilon^2 = 1 + (x^1)^2 + (x^2)^2$. The derivatives of the metric tensor are given by:

$$\begin{aligned}\mathbf{g}_{11,1}[\phi(\mathbf{x})] &= -\frac{4\beta_1}{\beta}\mathbf{g}_{11}[\phi(\mathbf{x})]\\ \mathbf{g}_{12,1}[\phi(\mathbf{x})] &= -\frac{4\beta_1}{\beta}\mathbf{g}_{12}[\phi(\mathbf{x})] - \frac{H}{(\beta)^4}\\ \mathbf{g}_{22,1}[\phi(\mathbf{x})] &= -\frac{4\beta_1}{\beta}\mathbf{g}_{22}[\phi(\mathbf{x})] + \frac{2L}{(\beta)^4}\\ \mathbf{g}_{11,2}[\phi(\mathbf{x})] &= -\frac{4\beta_2}{\beta}\mathbf{g}_{11}[\phi(\mathbf{x})] + \frac{2H}{(\beta)^4}\\ \mathbf{g}_{12,2}[\phi(\mathbf{x})] &= -\frac{4\beta_2}{\beta}\mathbf{g}_{12}[\phi(\mathbf{x})] - \frac{L}{(\beta)^4}\\ \mathbf{g}_{22,2}[\phi(\mathbf{x})] &= -\frac{4\beta_2}{\beta}\mathbf{g}_{22}[\phi(\mathbf{x})].\end{aligned} \quad (60)$$

where $H = x^2(\beta_1)^2 + \beta\beta_2 - x^1\beta_1\beta_2$ and $L = x^1(\beta_2)^2\beta_1\beta + -x^2\beta_1\beta_2$. Note that there are no second order derivatives in the above expression because they vanish in the case of planes. This leads to the CS in equation (57). □

Theorem 4. *Given that $\mathcal{M}_i$ with $i\in[1,N]$ are planes, the registration warps η_{ij} with $(i,j)\in[1,N]\times[1,N]$ are point-wise solutions of the 2D Schwarzian equations.*

Proof. The elements of the CS for $\mathcal{M}_i$ with $i = [1,\dots,N]$ have the form of (15), and thus must comply with the following algebraic constraints:

$$\begin{aligned}\mathbf{\Gamma}^1_{22}[\phi_i] &= \mathbf{\Gamma}^2_{11}[\phi_i] = 0\\ 2\mathbf{\Gamma}^2_{12}[\phi_i] &= \mathbf{\Gamma}^2_{22}[\phi_i]\\ \mathbf{\Gamma}^1_{11}[\phi_i] &= 2\mathbf{\Gamma}^2_{12}[\phi_i].\end{aligned} \quad (61)$$

From Corollary 1 we have $\mathbf{\Gamma}^p_{mn}[\phi_j] = \mathbf{\Gamma}^p_{mn}[\phi_i\circ\eta_{ji}]$. Now we use equation (10) to compute $\mathbf{\Gamma}^p_{nm}[\phi_i\circ\eta_{ji}]$ from $\mathbf{\Gamma}^p_{nm}[\phi_i]$ given in equation (57). Given that $\mathbf{x} = \eta_{ji}(\mathbf{y})$, we write

$$\begin{aligned}&\mathbf{\Gamma}^p_{nm}[\phi_i\circ\eta_{ji}] =\\ &\partial_1 y^p\left(-2\frac{\beta_{i,1}}{\beta_i}\partial_m x^1\partial_n x^1 - \frac{\beta_{i,2}}{\beta_i}\left(\partial_m x^1\partial_n x^2 + \partial_m x^2\partial_n x^1\right)\right) +\\ &\partial_2 y^p\left(-2\frac{\beta_{i,2}}{\beta_i}\partial_m x^2\partial_n x^2 - \frac{\beta_{i,1}}{\beta_i}\left(\partial_m x^1\partial_n x^2 + \partial_m x^2\partial_n x^1\right)\right) +\\ &\partial_1 y^p\partial^2_{mn}x^1 + \partial_2 y^p\partial^2_{mn}x^2.\end{aligned} \quad (62)$$

By forcing conditions in equation (61) in $\mathbf{\Gamma}[\phi_i \circ \eta_{ji}]$ we obtain the following four second order PDEs only in η_{ji}

$$\begin{aligned}
\left(\partial^2_{11}x^1\right)\left(\partial_1 x^2\right) - \left(\partial^2_{11}x^2\right)\left(\partial_1 x^1\right) &= 0 \\
\left(\partial^2_{22}x^1\right)\left(\partial_2 x^2\right) - \left(\partial^2_{22}x^2\right)\left(\partial_2 x^1\right) &= 0 \\
\left(\partial_{11}x^1\right)\left(\partial_2 x^2\right) - \left(\partial_{11}x^2\right)\left(\partial_2 x^2\right) & \\
+2\left(\left(\partial_{12}x^1\right)\left(\partial_1 x^2\right) - \left(\partial_{12}x^2\right)\left(\partial_1 x^1\right)\right) &= 0 \\
\left(\partial_{22}x^1\right)\left(\partial_1 x^2\right) - \left(\partial_{22}x^2\right)\left(\partial_1 x^1\right) & \\
+2\left(\left(\partial_{12}x^1\right)\left(\partial_2 x^2\right) - \left(\partial_{12}x^2\right)\left(\partial_2 x^1\right)\right) &= 0.
\end{aligned} \tag{63}$$

These are the 2D Schwarzian equations introduced in (Pizarro et al., 2016), where point-wise projective warps were investigated. □