ToTem NRS*f*M: Object-wise Non-Rigid Structure-from-Motion with a Topological Template

Agniva Sengupta[1] **Adrien Bartoli**[1,2]

[1] ENCOV, IGT, Institut Pascal
UMR6602 CNRS / Université Clermont Auvergne, France
[2] Department of Clinical Research and Innovation
Clermont-Ferrand University Hospital, France

Corresponding author: Agniva Sengupta, i.agniva+sengupta@gmail.com

October 7, 2023

International Journal of Computer Vision

Code and dataset at: github.com/agnivsen/ToTem-NRSfM

Abstract

We present a Non-Rigid Structure-from-Motion (NRS*f*M) method to reconstruct an object whose topology is known. We represent the topology by a 3D shape that weakly resembles the object, which we call a Topological Template (ToTem). The ToTem has two main differences with the template used in Shape-from-Template (S*f*T). First, the shape in the ToTem is not necessarily feasible for the object, whereas it must be in the S*f*T template. Second, the ToTem only models shape, excluding the classical texturemap representing colour in the S*f*T template. These two differences greatly alleviate the practical difficulty of constructing a template. However, they make the reconstruction problem challenging, as they preclude the use of strong deformation constraints between the template shape and the reconstruction and the possibility of directly establishing correspondences between the template and the images. Our method uses an isometric deformation prior and proceeds in four steps. First, it reconstructs point clouds from the images. Second, it aligns the ToTem to the point clouds. Third, it creates a coherent surface parameterisation. Fourth, it performs a global refinement, posed as a Bundle Adjustment (BA). We show experimentally that our method outperforms the existing methods for its isolated steps and NRS*f*M methods overall, in terms of 3D accuracy, ability to reconstruct the object's visible surface and ability to approximate the object's invisible surface.

Contents

1 INTRODUCTION

NRS*f*M is the problem of reconstructing the time-varying shape of a deforming scene as a set of point clouds, from image point correspondences [Bregler et al., 2000]. This is a difficult problem, as the number of unknown parameters is larger than the number of reprojection constraints, thus requiring priors, even when reconstructing a single object. The earliest prior used in NRS*f*M is the low-rank shape model [Bregler et al., 2000; Torresani et al., 2008]. It was followed by physics-based deformation priors, with isometry as the most successful one [Chhatkuli et al., 2014]. In spite of many advanced solution methods, NRS*f*M has had limited success due to insufficient accuracy in many cases. This is in sharp contrast to the rigid Structure-from-Motion (S*f*M) methods, as many sub-problems of the rigid case are considered solved [Faugeras and Luong, 2001; Hartley and Zisserman, 2004]. This is because rigidity is a strong prior, exerting important constraints on the reconstruction process. One important reason for the low accuracy of NRS*f*M is the lack of sufficiently strong priors. In contrast, S*f*T is a related method for deformable reconstruction which uses an object template as prior, comprising the object geometry, texturemap and deformation model [Bartoli et al., 2015; Perriollat et al., 2011; Salzmann and Fua, 2010]. Other classical methods such as Shape-from-Shading (S*f*S) and photometric stereo use additional priors related to the scene's lighting and the object's reflectance [Barron and Malik, 2014; Horn, 1975]. NRS*f*M is an attractive reconstruction method because it makes reasonable assumptions on the input data. A research challenge is thus to find new NRS*f*M priors with wide applicability and sufficient strength. We propose such a new prior in the form of the ToTem for object-wise NRS*f*M. The ToTem can be easily specified from basic knowledge about the object, yet is powerful enough to improve NRS*f*M. Its general definition is as follows.

Definition 1 (Topological Template)**.** *The ToTem is a shape prior in the form of a connected surface which has the observable object's topology. It is typically represented by an algebraic shape such as a sphere or by a triangular mesh.*

The ToTem shape may resemble the object shape, but this is not a hard requirement, for two reasons. First, the proposed reconstruction pipeline includes a final step depending only on the ToTem topology. This opens the possibility for the user to select a predefined ToTem, such as a sphere to reconstruct an inflated balloon. This makes the creation of a ToTem straightforward. Second, the notion of observability is important and refers to what can be seen of the object in the images, independently of its overall shape. For instance, imagine the camera is sufficiently close to the balloon, all it can observe is a topologically planar patch, making it desirable to use a plane as the ToTem, instead of a sphere. In most cases, the desired object's topology is minimal and easy to specify information for the user, as illustrated in figures 1 and 2.

We examine how the ToTem fits within the landscape of non-rigid 3D reconstruction methods including NRS*f*M and S*f*T in table 1. Existing NRS*f*M methods do not use a template. However, the use of a template in non-rigid 3D reconstruction is not new, forming the basis of S*f*T. The S*f*T template is made of a shape model and an appearance model. The shape model is, as in the proposed ToTem, an algebraic shape or a triangular mesh; and the appearance model is a texturemap. S*f*T then makes two strong assumptions. First, that the object shape can be reproduced by an isometric deformation of the template. Isometry is the distance-preserving property of deformations; it is the strongest and most popular mathematical deformation model in the context of 3D reconstruction, both in S*f*T [Perriollat et al., 2011; Salzmann and Fua, 2010] and NRS*f*M [Chhatkuli et al., 2017; Ji et al., 2017; Parashar et al., 2019]. Second, that the texturemap is matchable to the input images, which means that it must be of sufficient quality. Acquiring an S*f*T template may be impractical or infeasible in some applications, such as medical applications where patient and pathology-specific modifications to texture may not be available a priori, and automated food processing applications, where each object has a unique shape and texture. The proposed ToTem entirely resolves this issue. It however makes the 3D reconstruction problem different from and much more challenging than S*f*T.

We propose ToTem NRS*f*M, a novel formulation of NRS*f*M to exploit the topological prior. This formulation represents the reconstruction by coherently parameterised surfaces, which is the essential possibility brought by the knowledge of the object topology to the reconstruction process. The parameterisation is defined by a 2D space, called uv-space, and embedding functions, mapping points from the uv-space to the reconstructed 3D surfaces. The coherence is obtained by having each input point correspondence parameterised by a single point in uv-space. Concretely, for n images and m point correspondences, we estimate n 2D to 3D embedding functions and m 2D points in uv-space. The uv-space is classically used in computer graphics in order to texturemap a 3D mesh model, where the uv-space is aligned with texture images, and its mapping to the object mesh is obtained by means of local coordinates, typically barycentric coordinates drawn on the mesh's local triangles. The proposed representation can be regarded as a generative model of the observed point correspondences, with continuous surface extrapolation capability. This benefits NRS*f*M in three main ways. First, it facilitates the reconstruction of the object as a time-varying surface, as opposed to point clouds in standard NRS*f*M. One thus obtains a richer object representation for the downstream applications. Second, it improves the reconstruction accuracy, thanks to surface regularisation. Third, it provides hints on the shape of invisible object parts, which are not seen in any image and not reconstructed by existing NRS*f*M methods. Invisible parts are very common in images of volumetric objects; obtaining a hint about the shape of these parts would benefit applications in robotics and surgical image analysis, to name a few. To sum up, ToTem NRS*f*M is appealing, as it allows one to reconstruct full dense object models from a very light prior, without requiring model training and thus free of training data. An example of ToTem NRS*f*M in action is provided in figure 1, highlighting its full-surface reconstruction capability, including invisible parts, and the capability to leverage object topology to guide the reconstruction, as opposed to simple point-wise 3D reconstruction from the existing literature.

In developing ToTem NRS*f*M, the main difficulty is to relate the untextured and approximate shape prior to the

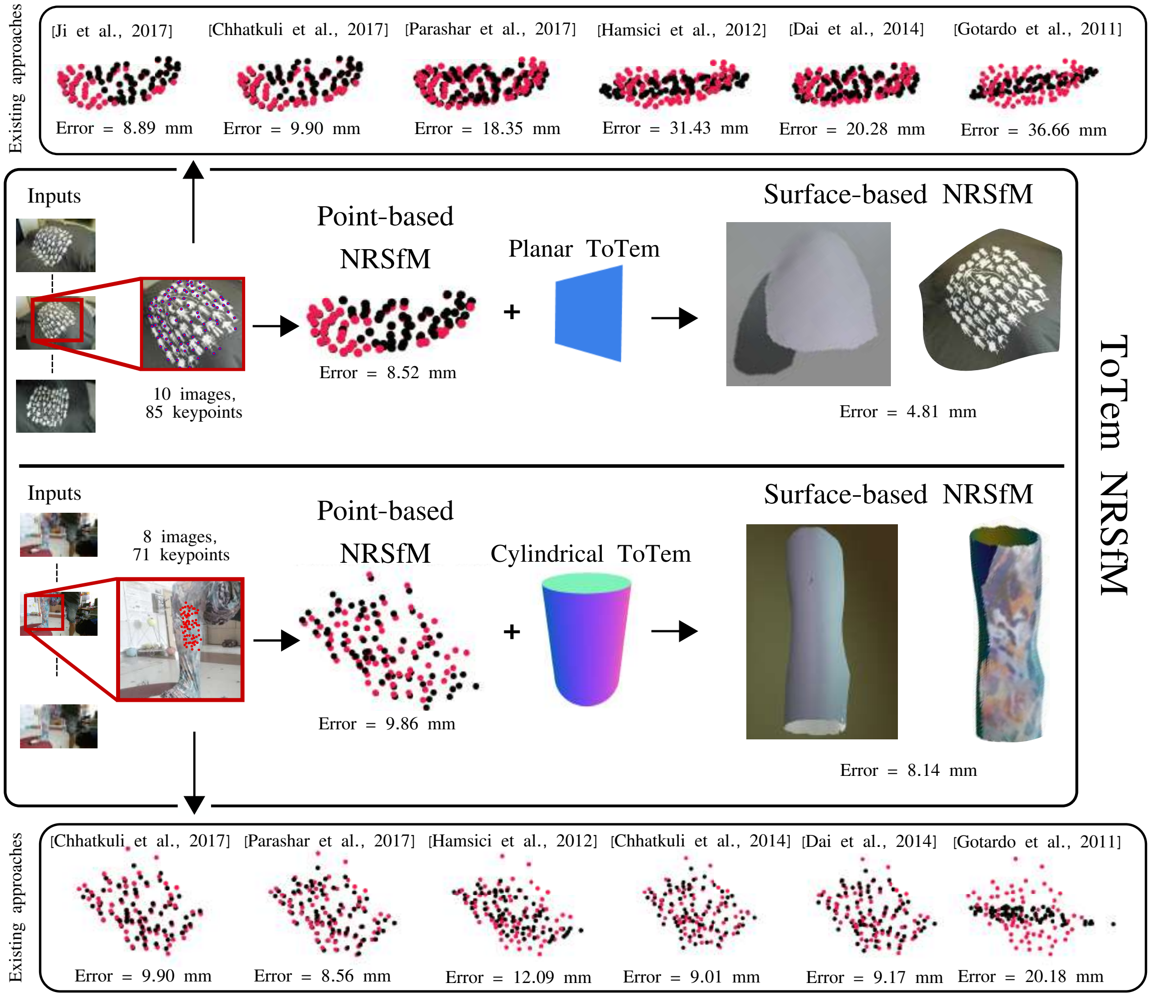


Figure 1: Two example reconstructions with comparisons. The top two rows show a deforming t-shirt; while our point-based NRS*f*M is already ahead of all existing approaches, the planar ToTem enables a very accurate dense reconstruction as well as smooth surface extrapolation to areas outside the boundary of the point correspondences. The bottom two rows show a legging being stretched, *i.e.*, a non-isometric case; by using the cylindrical ToTem, not only do we arrive at an accurate reconstruction, but we also generate a hypothesis for the unobserved surface of the leg, something beyond the capabilities of any existing method. The reported error metric is the 3D error, the red points are groundtruth and the black points are reconstructions; a deeper analysis is given in our experiments in section 8.

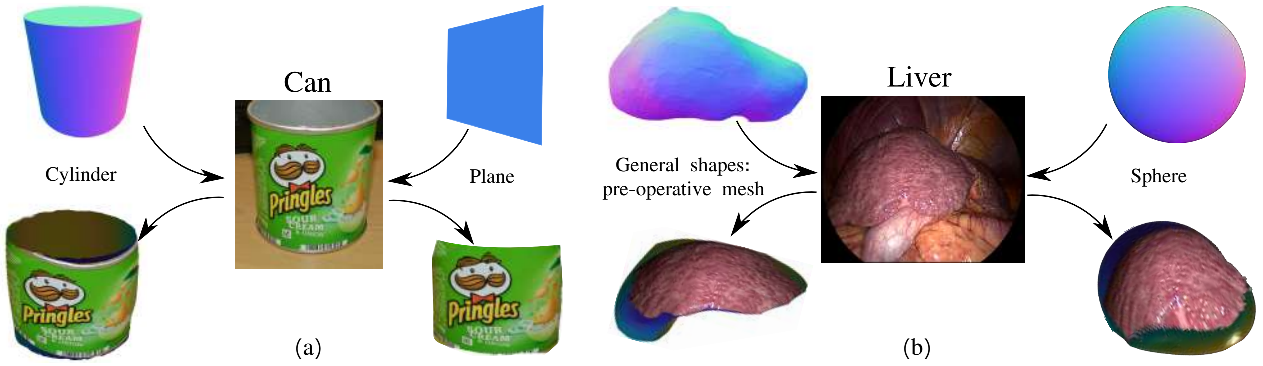


Figure 2: Two examples demonstrating the flexibility of ToTem NRS*f*M. (a) while the cylinder is an accurate ToTem for reconstructing this Pringles can, a planar ToTem does the job of dense surface reconstruction; the ability to reconstruct the unseen surface is however only available for the cylinder. (b) for reconstructing a deforming liver, a preoperative mesh model leads to an accurate reconstruction; in the absence of such a model, the knowledge that the human liver weakly resembles a sphere still allows our method to reconstruct the entire surface.

image point correspondences. This has connections with surface parameterisation, a problem for which there exist a substantial body of approaches. We propose an NRS*f*M method following an advanced pipeline with four main steps. The first three steps initialise the reconstruction model in a bottom-up manner and the last step performs a global refinement in a top-down manner. This pipeline is complex but rests on well-understood building blocks, for which efficient solution methods exist or are proposed. Step 1) computes an initial standard NRS*f*M reconstruction. Step 2) computes an initial individual alignment of the ToTem to each reconstructed point cloud. Step 3) computes an initial parameterised surface for each point cloud with independent embeddings and uv-space points. It then enforces coherence by computing common uv-space points. Step 4) refines the uv-space points and the embeddings in a BA manner.

We implement the proposed method under the common assumption of isometric object deformation. We study four possibilities for the ToTem: three algebraic shapes, namely the plane, the cylinder and the sphere, and general shapes. We have chosen these three algebraic shapes as their topology is commonly found in real-world objects. The proposed solution pipeline is extensible to other topologies. We assume without loss of generality that a uv-space has been computed for the ToTem, along with a flattening function, mapping points from the prior 3D shape to the uv-space. There often exist several such 2D parameterisations; for algebraic shapes such as a sphere, this comes in closed-form, while for general meshes, this may be obtained by conformal flattening [Sheffer et al., 2005]. We represent functions by polyharmonic splines, generalising the well-known Thin-Plate Spline (TPS) to higher dimensions. We model isometry as the approximate preservation of local inter-point distances. Along with the concurrent estimation of the inter-point distances with the reconstruction, this introduces flexibility and allows the model to cope with real deformations that do not strictly follow the isometric model. In addition to the proposed generative dense reconstruction model and the above pipeline, we bring four technical contributions required to implement the

	Strong priors	⟶	**Weak priors**
METHOD	**S*f*T**	**Proposed method**	**NRS*f*M**
Template:	S*f*T	ToTem	None
Texture	Required	—	—
Shape	Feasible	Topological	—
Deformation model	Required	Required	Required
Reconstruction	Full[†]	Full[†]	Partial[‡]

Table 1: Classification of non-rigid 3D reconstruction hypotheses and methods. S*f*T and NRS*f*M are two approaches lying at the extrema of the possible hypotheses: S*f*T uses strong hypotheses and NRS*f*M weak ones. We propose ToTem, an intermediate hypothesis to improve NRS*f*M when the strongest S*f*T template is unavailable.

[†] The reconstruction recovers the entire structure of the object, even those parts that are occluded from the camera

[‡] Only the visible parts of the object are reconstructed

method's steps. First, we propose a method for isometric NRS*f*M, used in step 1). We use an alternation scheme embedding the Maximum Depth Heuristic (MDH) principle to resolve the local convex-concave ambiguities using a novel automatic hyperparameter tuning scheme. In contrast, previous work solves a convex inextensible relaxation of this problem [Chhatkuli et al., 2017; Ji et al., 2017]. Second, we propose alignment methods for the cylinder and the sphere to a point cloud, used in step 2), using multi-step non-iterative optimisation, where each step is solved by a globally optimal method. Third, we propose a coherent parameterisation method for multiple 3D point clouds, used in step 3). Fourth, we propose a non-rigid surface-based adaptation of BA, used in step 4). This optimises all the model parameters and uses all the constraints with closed-form Jacobian matrices and sparsity aware accelerations, ensuring rapid convergence [Granshaw, 1980; Triggs et al., 1999].

We experimentally validate our method on various synthetic and real datasets, both quantitatively and qualitatively. The datasets cover all four types of ToTem shapes, namely the three algebraic shapes and general shapes. The choice of which ToTem to use for a given object with known topology is straightforward: if shape information is available, as in the surgical dataset, one should use this shape to form a general shape ToTem; otherwise, one should use the algebraic shape ToTem whose topology matches the object's.

2 RELATED WORK

We split our related work review into two parts: image-based non-rigid 3D reconstruction, with classical NRS*f*M and neural approaches, and parameterised surface fitting to point clouds.

2.1 Image-based Non-rigid 3D Reconstruction

We first discuss the classical NRS*f*M approach and then the neural approach.

2.1.1 The Non-Rigid Structure-from-Motion Approach

Research related to NRS*f*M started in the 00's, primarily using the low-rank shape model [Bregler et al., 2000; Dai et al., 2014; Kumar and Van Gool, 2022; Torresani and Bregler, 2002; Torresani et al., 2001]. This model is mainly effective at modelling mild deformations, explainable by a linear model. Physics-based deformation models then resolved this issue. In particular, the isometric model is widely applicable and yet provides sufficient constraints for reconstruction.

The isometric deformation model. The isometric deformation model has been widely used in Isometric NRS*f*M (Iso-NRS*f*M) [Chhatkuli et al., 2014] but also in related approaches including rigid S*f*M [Sabzevari et al., 2012], S*f*T [Chhatkuli et al., 2016; Östlund et al., 2012; Parashar et al., 2015; Salzmann and Urtasun, 2012] and for object shape reconstruction outside computer vision, for instance in molecular conformation [Biswas et al., 2008]. However, the isometric constraint suffers from two well-known drawbacks. The first drawback is that it does generally not uniquely resolve the surface but rather leads to a solution set [Dubrovina and Kimmel, 2011], containing multiple surfaces all obeying the constraints. In 3D reconstruction, this translates to convex-concave ambiguities. The second drawback is that the isometry constraint is an approximation, albeit generally a good one, as the surface of real objects do not exactly follow isometry, owing to the object's thickness.

The Maximum Depth Heuristic. In order to remedy the first drawback, a widely used heuristic is to reconstruct the deforming surface while simultaneously maximising the depth of the reconstructed surface from some reference point. The formulation has a unique solution, which is in general a plausible surface. This heuristic has been termed MDH [Perriollat et al., 2011; Salzmann and Fua, 2010] and Maximum-Leg Heuristics (MLH) [Ji et al., 2017]. We use MDH as a general name for both implementations. Maximum Depth Heuristic NRS*f*M (MDH-NRS*f*M) has been proven to be generally effective [Chhatkuli et al., 2017; Ji et al., 2017]. An explanation for this effectiveness is that it chooses the furthest solution surface from the camera, which is also the flattest surface.

Approximating isometry. In order to remedy the second drawback, approximate isometry constraints are commonly used. The way these constraints are formulated is intertwined with the mathematical framework. Specifically, two frameworks are found. The first framework, which we call 'algebraic', uses the point correspondences directly [Salzmann and Urtasun, 2012; Varol et al., 2009]. The second framework, which we call 'differential', uses the derivatives of the optic flow at the correspondences, at first-order [Chhatkuli et al., 2014] or at second-order [Parashar et al., 2017]. For both frameworks, isometry is approximated using a cost function attempting to minimise the variance of the distances across time in the reconstruction. The algebraic framework is simpler and more appealing in this respect. However, as it models the reconstruction by point clouds, one thus has to approximate the geodesic distance by a Euclidean one, introducing an extra approximation level. The algebraic framework does not lead to convex formulations and requires one to reconstruct the complete point set at once. In contrast, the differential framework is more complex, mainly owing to the computation of the optic flow. However, it allows one to resolve the normal at each correspondence individually in closed-form.

The convex inextensible relaxation. Convex relaxations are common for rigid scenes, specifically for camera pose, homography, and epipolar geometry estimation [Kahl and Henrion, 2007] and also for second-order differential NRS*f*M [Probst et al., 2019]. The inextensible relaxation of isometry, introduced for algebraic NRS*f*M in [Chhatkuli et al., 2017], constrains the inter-point Euclidean distances to remain lower or equal to the geodesic distances. This leads to the methods of inextensible Iso-NRS*f*M, where the constraints are inequalities of quadratic terms that can be posed as a convex Second-Order Cone Programming (SOCP). The inequalities were then modified to pose the problem as a convex Semi-Definite Programming (SDP), resulting in a tighter relaxation than SOCP [Ji et al., 2017], which is however substantially slower. Completed with the MDH, this results in an elegant convex solution method, coping with the first aforementioned drawback. The inextensible Iso-NRS*f*M methods however have limitations. First, they may have low accuracy, owing to the inextensible relaxation of the original isometric constraints. Second, they have a high error for data that depart from isometric deformations, producing an infeasible solution from the relaxed feasible set, a well-known drawback of convex relaxations [Kahl and Henrion, 2007; Li et al., 2015b]. This points to the necessity of a refinement technique exploiting the original isometric constraints.

2.1.2 The Neural Approach

The neural approach estimates depth from a single or a stream of monocular images using deep learning. A Convolutional Neural Network (CNN) is used in [Liu et al., 2015] to estimate the depth of general scenes, while a CNN is used along with Conditional Random Fields (CRF) to estimate the depth and surface normals in [Li et al., 2015a]. In contrast, a fully convolutional neural network is used in [Cao et al., 2017] to estimate depth ranges, essentially treating the problem as the classification of image pixels into bins of depth. Takmaz et al. [2020] recover the depth of a deforming scene using dense correspondences as input features by training a CNN using an As-Rigid-As-Possible (ARAP) hypothesis. Kong and Lucey [2019] cast NRSfM as a multi-layer block-sparse dictionary learning problem and demonstrate its equivalence with feed-forward DNN auto-encoders. Novotny et al. [2019] extract a 3D model from monocular images by learning a reconstruction function along with a canonicalisation function that enforces pose consistency. Recent detailed reviews of the Deep Convolutional Neural Network (DCNN) approach are given in [Bhoi, 2019; Khan et al., 2020]. An important drawback of the DCNN approach is the well-known lack of domain adaptation. To date, there are no publicly available neural models for NRSfM.

2.2 Surface Fitting to Point Clouds

The problem of parameterised surface fitting to a single point cloud has been largely studied. A vast body of methods is available in computer graphics and computational geometry for the closely related problem of surface reconstruction from a point cloud [Berger et al., 2017]. These methods however require high point density and quality, which are not met by the point clouds obtained by NRSfM. Within the applicable methods, we distinguish the rigid and the non-rigid cases. In the rigid case, the problem boils down to estimating the rigid-body motion between a template shape and a sparse point cloud. The base solution technique is Iterative Closest Point (ICP) [Besl and McKay, 1992; Castellani and Bartoli, 2020; Liu et al., 2020]. The problem is typically solved via non-convex optimisation [Bellekens et al., 2014], but there also exists techniques leading to the global minima [Low, 2004; Yang et al., 2015]. In the non-rigid case, the rigid-body motion is replaced by a deformation model. The amount of existing work is more restricted than in the rigid case and each method is generally specific to a deformation model and to a representation of the template shape. The problem is posed as non-linear optimisation with spatial constraints [Bernard et al., 2017; Russell et al., 2011]. However, [Russell et al., 2011] assumes the object is planar, and [Bernard et al., 2017] requires an RGB-D sensor. The case of SfT with a volumetric template is addressed in [Parashar et al., 2015], which however requires a template with a feasible shape and textural information. More recently, [Golyanik et al., 2020] has proposed a dynamic shape prior for NRSfM. This prior is represented by a low-rank shape basis constructed from an NRSfM reconstruction over a batch of images in an ‘acquisition’ step. It is then used to explain the deformation and pose for new images. The prior is however not updated and remains expressed in the form of point clouds. To sum up, the problem of computing surface parameterisation from a single point cloud and the computation of a point cloud model from multiple point clouds have been well-studied. However, the computation of a coherent parameterisation for multiple point clouds is still in need of solutions, both for simple algebraic and for complex non-parametric shape template priors.

3 MODEL AND METHOD OVERVIEW

We give the proposed surface-based model of ToTem NRSfM model, using figure 3 as an illustration. We first review the standard point-based model of NRSfM, which serves as basis. In ToTem NRSfM, each surface is obtained by a non-rigid transformation of the ToTem shape. We thus augment the standard model by introducing models for the ToTem and for the unknown surfaces, and their computational representation by polyharmonic splines. We finally present the pipeline of the proposed solution method.

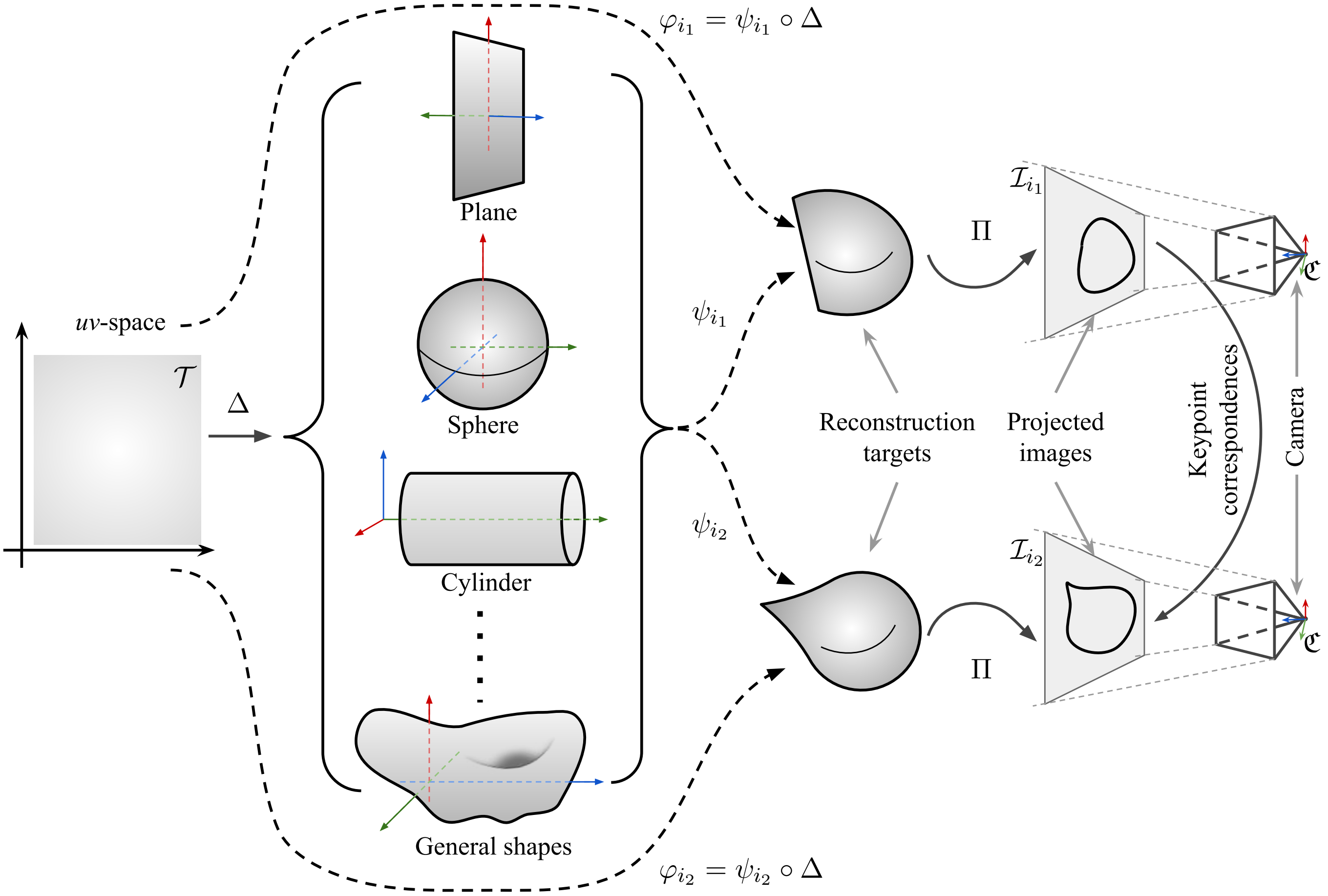


Figure 3: The proposed model of ToTem NRS*f*M is shown for $n = 2$ (a pair of views), where each deformed surface is defined as a composition of the $\mathbb{R}^2 \mapsto \mathbb{R}^3$ ToTem embedding Δ and an $\mathbb{R}^3 \mapsto \mathbb{R}^3$ non-rigid transformation ψ_i modelling object deformation and camera pose for all $i \in [1, n]$. Four examples of ToTem shapes are shown. The known transformations are represented by bold curves and the unknown ones are represented by dashed curves.

3.1 Standard Point-based Model

We consider m point correspondences between n images as inputs. For the i-th image $\mathcal{I}_i$, the j-th image point $\mathbf{q}_{i,j} = \left(\begin{smallmatrix} u_{i,j} & v_{i,j} \end{smallmatrix}\right)^\top \in \mathbb{R}^2$ is related to the unknown 3D point $\mathbf{Q}_{i,j} = \left(\begin{smallmatrix} X_{i,j} & Y_{i,j} & Z_{i,j} \end{smallmatrix}\right)^\top \in \mathbb{R}^3$ by the standard perspective projection function Π such that $\mathbf{q}_{i,j} = \Pi(\mathbf{Q}_{i,j})$ [Hartley and Zisserman, 2004]. The camera intrinsics are known, including the focal length (f_x, f_y) and principal point (c_x, c_y), in pixel units. Correspondences may or may not be visible across all images, which we model by a visibility matrix $\mathcal{V}$ whose (i, j)-th element $\mathcal{V}_{i,j}$ is 1 if $\mathbf{q}_{i,j}$ is visible and 0 otherwise. The unknown depth of a 3D point is denoted $\delta_{i,j} = \|\mathbf{Q}_{i,j}\|$, which represents its distance from the camera centre along the sightline. The homogeneous coordinates for any point such as $\mathbf{P}$ are denoted as $\tilde{\mathbf{P}}$. Each point $\mathbf{q}_{i,j}$ and $\mathbf{Q}_{i,j}$ are connected to their neighbours $\mathbf{q}_{i,q}$ and $\mathbf{Q}_{i,q}$ through a Nearest-Neighborhood Graph (NNG), represented by $\mathcal{N}$, derived using the strategy of [Chhatkuli et al., 2017] such that $q \in \mathcal{N}(j)$. The standard NRS*f*M setup is devoid of a representation of the object's surface, which we introduce next with the notion of ToTem and the deformation model.

	Definition of Δ and Δ^{-1}	Coordinates in $\mathcal{T}$	Properties of Δ	Shape properties	Orientation w.r.t. $\mathbb{R}^3$	Freedoms
Plane	$\Delta(x, y) = \begin{pmatrix} x & y & 0 \end{pmatrix}$, $\Delta^{-1}(X, Y, Z) = (X, Y)$	$x \in [-1, 1]$, $y \in [-1, 1]$	Isometric, $\Delta \in C^\infty$	Doubly ruled, non-compact	The plane of $z = 0$	Translation along $z = 0$, reflection along any plane $\perp$ to $z = 0$, rotation along axis $\perp$ to $z = 0$, π rotation about any point on $z = 0$
Cylinder	$\Delta(r, \theta) = \begin{pmatrix} \sin\theta & r & \cos\theta \end{pmatrix}$, $\Delta^{-1}(X, Y, Z) = \left(Y, \tan^{-1}\left(\frac{Z}{X}\right)\right)$	$r \in (0, L]$, $\theta \in (-\pi, \pi]$, where L = length of cylinder	Isometric, $\Delta \in C^\infty$	Ruled, non-compact	The cylinder axis is $(x, z) = 0$	Translation and rotation along $(x, z) = 0$, reflection along any plane $\perp$ to $(x, z) = 0$, π rotation along $(x, y) = 0$ and $(y, z) = 0$, reflection along $x = 0$, $y = 0$ or $z = 0$
Sphere	$\Delta(\theta_1, \theta_2) =$ $\begin{pmatrix} -\cos\theta_1\cos\theta_2 & -\sin\theta_1 & -\cos\theta_1\sin\theta_2 \end{pmatrix}$, $\Delta^{-1}(X, Y, Z) = \left(\sin^{-1}(-Y), \tan^{-1}\left(\frac{X}{Z}\right)\right)$	$\theta_1 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\theta_2 \in (-\pi, \pi]$	Conformal, $\Delta \in C^\infty$	Compact	Centred at the origin	Reflection along $x = 0$, $y = 0$ or $z = 0$, any rotation

Table 2: Summary of the embedding Δ and the flattening Δ^{-1} for the three considered algebraic shapes.

3.2 ToTem Shape Model and Representation

The ToTem is a 3D shape denoted $\mathcal{U} \subset \mathbb{R}^3$. It is represented by a uv-space denoted $\mathcal{T} \subset \mathbb{R}^2$ and an embedding $\Delta : \mathcal{T} \mapsto \mathcal{U}$ which maps a point $\mathbf{p} \in \mathcal{T}$ to $\Delta(\mathbf{p}) \in \mathcal{U}$. The embedding is almost everywhere smooth and regular. It is derived from the knowledge of the ToTem and then kept fixed for the rest of the reconstruction process. Each ToTem is represented in a canonical known pose. The ToTem with symmetries have freedoms in the canonical pose; for instance, the sphere does not define a canonical orientation. We ensure that these unconstrained degrees of freedom do not affect our reconstruction method. We elaborate on the derivation of the uv-space and Δ for algebraic and general shapes below. For points restricted to the ToTem shape, the embedding is invertible. It is denoted as $\Delta^{-1} : \mathcal{U} \mapsto \mathcal{T}$ for simplicity and named the flattening transformation.

Algebraic shapes. We consider three frequent shapes, the plane, the cylinder and the sphere [Hilbert and Cohn-Vossen, 2021]. The plane has a straightforward embedding whereas the cylinder and sphere require cylindrical and spherical coordinates respectively. For completeness, we give the expressions of Δ and Δ^{-1} and some relevant properties in table 2. The price to pay for parameterising the cylinder and sphere with a single chart is to admit singularities, where Δ is multi-valued on the surface of these ToTems. These singularities lie at $\theta = \pi$ for the cylinder and $\theta_2 = \pi$ for the sphere. We propose a singularity avoidance method in section 6.1.2.

General shapes. We consider that a general shape is given as a general triangular 3D mesh. We use conformal flattening [Sheffer et al., 2005] to obtain the uv-space $\mathcal{T}$, which returns the coordinates of each mesh vertex in uv-space. The mesh vertices thus form 2D-3D point correspondences, from which we can define an implementation for the embedding Δ and the flattening transformation Δ^{-1}. The flattening Δ^{-1} is used to compute the initial shape parameterisation in step 3) of our method. This step does not require differentiability and we thus implement Δ^{-1} using standard barycentric coordinates. In other words, for a point $\mathbf{Q} \in \mathcal{U}$ on some triangle of the ToTem shape, we find its barycentric coordinates with respect to its containing triangle and deduce its image $\mathbf{p} \in \mathcal{T}$ by applying the barycentric coordinates to the corresponding vertices in uv-space. The embedding Δ is used to perform BA in step 4) of our method. This step requires differentiability and we thus implement Δ by smooth interpolation of the 2D-3D vertex correspondences by a TPS [Bookstein, 1989].

3.3 Unknown Surface Set Model

Our prior is that the unknown surface $\mathcal{S}_i$ for image i can be obtained by a non-rigid transformation $\psi_i : \mathbb{R}^3 \mapsto \mathbb{R}^3$ of the ToTem, deforming it so that point $\Delta(\mathbf{p}) \in \mathcal{U}$ is mapped to $\psi_i\big(\Delta(\mathbf{p})\big) \in \mathcal{S}_i$. In order to take advantage of the uv-space provided by the ToTem prior, we introduce an embedding to represent the unknown surface $\mathcal{S}_i$ as $\varphi_i(\mathbf{p}) = \psi_i\big(\Delta(\mathbf{p})\big) \in \mathcal{S}_i$. The embedding $\varphi_i = \psi_i \circ \Delta$ is the composition of two smooth maps and is thus also smooth. Therefore, we obtain the tangent vectors as $\varphi_{i,u} = \frac{\partial \varphi_i}{\partial u}$ and $\varphi_{i,v} = \frac{\partial \varphi_i}{\partial v}$ and the surface normal from the normal operator $\eta[\varphi_i] = \frac{\varphi_{i,u} \times \varphi_{i,v}}{\|\varphi_{i,u} \times \varphi_{i,v}\|}$ [Kreyszig, 2013]. For any point $\mathbf{p} = (u, v) \in \mathcal{T}$, the corresponding surface normal is $\eta[\varphi_i](\mathbf{p})$. Extending these differential properties of the embedding φ_i, we arrive at the Gaussian curvature and the geodesic distance between points along the surface $\mathcal{S}_i$. A compendium of such properties relevant to the proposed method is given in appendix A. We consider $\psi_i : \mathbb{R}^3 \mapsto \mathbb{R}^3$ for $i = 1, \ldots, n$ as reconstruction unknowns, for which the next section gives a computational representation.

3.4 Non-rigid Transformation Representation

We require a computational representation of the functions representing non-rigid transformations. This includes the ToTem embedding $\Delta : \mathcal{T} \mapsto \mathbb{R}^3$, the non-rigid 3D transformations $\psi_i : \mathbb{R}^3 \mapsto \mathbb{R}^3$ and the deformable Generalized Procrustes Analysis (GPA) transformations $\varpi_i : \mathbb{R}^2 \mapsto \mathbb{R}^2$. For all these transformations, we use polyharmonic splines [Iske, 2004], which defines smooth, C^∞ functions, from an arbitrary source space dimension $d_s \in \{2, 3\}$ and an arbitrary target space dimension $d_t \in \{2, 3\}$. This representation can model the 3D deformation of arbitrary shapes and, hence can be used with any ToTem shape. It extends the well-known TPS [Bookstein, 1989], which is restricted to a 2D source space dimension, to arbitrary dimensions, and retains the desirable properties of the Radial Basis Function (RBF). We drop the image index i in the remainder of this section for clarity and define a general transformation ψ, which is later instantiated to represent the set of all n transformations $\{\psi_i\}$ and the other relevant transformations. For a 3D point $\mathbf{P} = \begin{pmatrix} X & Y & Z \end{pmatrix}^\top$, the map ψ is defined as:

$$\psi(\mathbf{P}) = \tilde{\mathbf{P}}^\top \mathbf{a} + \rho(\mathbf{P}, \mathbf{D})\mathbf{w}, \tag{1}$$

where $\{\mathbf{D}_k\} \in \mathbb{R}^3$ for all $k \in [1, l]$ are the source *control points*. ψ in equation (1) is divided into an affine part $\mathbf{a} \in \mathbb{R}^{4\times 3}$ representing the behaviour of ψ at infinity and a weighted sum over the RBF $\rho \in \mathbb{R}^l$ representing the nonaffine component [Bookstein, 1989], where $\mathbf{w} \in \mathbb{R}^{l\times 3}$ are the weights. We use the formulation $\rho(\mathbf{P}, \mathbf{D}_k) = \|\mathbf{P}-\mathbf{D}_k\|$ for the basis function of the k-th *control points* [Hu et al., 2009]. The conventional strategy for estimating the warp parameters $(\mathbf{a}, \mathbf{w})$ involves defining *target vertices* $\mathbf{C}$ which the surface produced by ψ must contain [Bartoli et al., 2010], such that:

$$\begin{bmatrix} \check{\mathbf{w}} \\ \check{\mathbf{a}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{K}_\lambda & \tilde{\mathbf{D}}^\top \\ \tilde{\mathbf{D}} & \mathbf{0}_{4\times 4} \end{bmatrix}^{-1}}_{\mathcal{D}^{-1}} \begin{bmatrix} \tilde{\mathbf{C}} \\ \mathbf{0} \end{bmatrix} = \varepsilon_\lambda \begin{bmatrix} \tilde{\mathbf{C}} \\ \mathbf{0} \end{bmatrix} \qquad \text{where} \qquad \Big(\mathbf{K}_\lambda\Big)_{k,k'} = \begin{cases} \lambda & \text{if } k = k' \\ \rho\big(\mathbf{D}_k, \mathbf{D}_{k'}\big) & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathbf{a}$ and $\mathbf{w}$ are obtained by truncating the fourth column of the 4×4 affine transformation matrix $\check{\mathbf{a}}$ and the $l \times 4$ warping coefficient matrix for the nonaffine deformation $\check{\mathbf{w}}$. Matrices $\tilde{\mathbf{D}} \in \mathbb{R}^{4\times l}$, $\tilde{\mathbf{C}} \in \mathbb{R}^{l\times 4}$ contain the homogeneous coordinates of the *control points* and *target vertices* respectively and $\mathcal{D}$ is termed as the *design matrix*. The transformation ψ can therefore be represented as:

$$\psi(\mathbf{P}) = \begin{bmatrix} \rho(\mathbf{P}, \mathbf{D}) & \tilde{\mathbf{P}}^\top \end{bmatrix} \varepsilon_\lambda \begin{bmatrix} \tilde{\mathbf{C}} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

Interestingly, ψ, although non-linear w.r.t. $\mathbf{P}$ and $\mathbf{D}$, remains linear w.r.t. $\mathbf{C}$ in equation (3).

3.5 Solution Method Overview

The proposed solution method integrates the surface-based model introduced in the above sections. The inputs are the image points $\{\mathbf{q}_{i,j}\}$ and their visibility $\{\mathcal{V}_{i,j}\}$ for $i \in [1, n]$, $j \in [1, m]$, as in standard NRS*f*M, the uv-space $\mathcal{T}$ and the ToTem embedding $\Delta : \mathcal{T} \mapsto \mathbb{R}^3$, computed following section 3.2. We use a neighbourhood structure $\mathcal{N}$ that connects points to define the deformation constraints. This structure is provided by an NNG obtained from the correspondences, following the strategy of [Chhatkuli et al., 2017]. We denote $q \in \mathcal{N}(j)$ to specify that the j-th and q-th points are connected. The proposed method follows a pipeline with four essential steps:

1. **Initial NRS*f*M reconstruction.** We use isometry, which we model as the least-squares preservation of the Euclidean distance between the unknown 3D points, to recover the depth of each input 2D point. This depends on a neighbourhood structure defining which point-pairs are considered. We use the MDH principle to guarantee that the formulation has a unique solution.

   Inputs: keypoint correspondences $\{\mathbf{q}_{i,j} \in \mathbb{R}^2\}$ and visibility $\{\mathcal{V}_{i,j} \in \{0, 1\}\}$ for all $i \in [1, n]$, $j \in [1, m]$, neighbourhood structure $\mathcal{N}$.

   Outputs: point-clouds $\{\mathbf{Q}_{i,j} \in \mathbb{R}^3\}$ for all $i \in [1, n]$, $j \in [1, m]$.

2. **Initial individual alignment of the ToTem to the reconstructed point-clouds.** We compute a rigid alignment of the ToTem to each reconstructed point-cloud. We provide an ad hoc solution method for each of the three algebraic shapes, consisting of sequential optimal solution steps, and a general solution method for general shapes.

   Inputs: point-clouds $\{\mathbf{Q}_{i,j}, \in \mathbb{R}^3\}$, visibility $\{\mathcal{V}_{i,j} \in \{0,1\}\}$ for all $i \in [1,n]$, $j \in [1,m]$, ToTem shape prior, neighbourhood structure $\mathcal{N}$.

   Outputs: rigid transformations $\{f_{P,i}\}$ for all $i \in [1,n]$.

3. **Initial surface reconstruction and coherent parameterisation.** We first compute independent uv-space coordinates for all 3D point-clouds, for which we enforce coherence using deformable GPA in the 2D uv-space. We then find the non-rigid 3D transformations ψ_i representing each of the unknown surfaces.

   Inputs: point cloud $\{\mathbf{Q}_{i,j} \in \mathbb{R}^3\}$, visibility $\{\mathcal{V}_{i,j} \in \{0,1\}\}$, rigid transformations $\{f_{P,i}\}$ for all $i \in [1,n]$ and $j \in [1,m]$, ToTem shape prior.

   Outputs: non-rigid transformations $\{\psi_i\}$ for all $i \in [1,n]$.

4. **Complete refinement by Bundle Adjustment.** We refine the uv-space coordinates and the set of non-rigid 3D transformations ψ_i. We pose the problem as the minimisation of a combination of the reprojection error, inextensibility and surface smoothing.

   Inputs: keypoint correspondences $\{\mathbf{q}_{i,j} \in \mathbb{R}^2\}$, point cloud $\{\mathbf{Q}_{i,j} \in \mathbb{R}^3\}$ and visibility $\{\mathcal{V}_{i,j} \in \{0,1\}\}$ for all $i \in [1,n]$, $j \in [1,m]$, neighbourhood structure $\mathcal{N}$, ToTem shape prior.

   Outputs: non-rigid transformations $\{\psi_i'\}$ for all $i \in [1,n]$.

Atypical of standard surface reconstruction methods, our method produces a set of parameterised surfaces as outputs, instead of a triangular mesh. However, a triangular mesh can be easily obtained for each surface using standard methods [Bernardini et al., 1999; Kazhdan and Hoppe, 2013], in a coherent manner, meaning that the mesh vertices would be physically corresponding points, if desired.

Topology is a weak prior, making it difficult to exploit in practice. The proposed method is based on the user-selected ToTem, which has a shape of the selected topology. It is thus important to understand how the ToTem shape, beyond the ToTem topology, may influence the reconstruction. In the above reconstruction pipeline, step 1 is independent of the ToTem, steps 2 and 3 are dependent on the ToTem shape and step 4 is dependent on the ToTem topology but independent of the ToTem shape. We achieve this independence by minimising terms expressed only on the differences between the reconstructed surfaces and refining the point parameterisation jointly. This theoretically guarantees that the final reconstruction does not depend on the ToTem shape.

4 ISOMETRIC NON-RIGID STRUCTURE-FROM-MOTION

Isometric NRSfM represents the first step of our reconstruction pipeline. We give the motivation for the proposed method, the problem statement, and the proposed solution method.

4.1 Motivation

Isometric NRSfM was relaxed to inextensible NRSfM and solved elegantly in [Chhatkuli et al., 2017; Ji et al., 2017] with convex relaxations. The inextensible and the convex relaxations involved in these formulations lead to a convex second-order or Positive Semi-Definite (PSD) cone, which admits a feasible set larger than the original problem formulation [d'Aspremont and Boyd, 2003]. The relaxed solution may thus be inaccurate, as confirmed by our experiments on real cases. Therefore, there exists the theoretical and practical necessity of refining the solutions from [Chhatkuli et al., 2017; Ji et al., 2017], based on the original isometric constraints. Isometric NRSfM is, unsurprisingly, non-trivial as it involves non-convex constraints and is ill-posed. We propose a well-posed formulation exploiting the true isometric constraints and the MDH.

4.2 Problem Statement

We begin the derivation of the proposed problem statement from the convex formulation of [Chhatkuli et al., 2017]. We assume full visibility of the points for the sake of simplicity throughout the derivation and introduce the visibility indicator in our final formulation. By denoting $\hat{\mathbf{q}}_{i,j}$ as the unit vector directed towards $\mathbf{Q}_{i,j}$, we have:

$$\begin{gathered}\min_{\{\delta_{i,j}\},\{d_{j,q}\}} -\sum_{i=1}^{n}\sum_{j=1}^{m}\delta_{i,j} \\ \text{s.t.} \quad \|\delta_{i,j}\hat{\mathbf{q}}_{i,j}-\delta_{i,q}\hat{\mathbf{q}}_{i,q}\| \leq d_{j,q}, \quad \sum_{j'=1}^{m}\sum_{q'\in\mathcal{N}(j')} d_{j',q'}=1, \quad \delta_{i,j}\geq 0, \quad \forall i\in[1,n], j\in[1,m], q\in\mathcal{N}(j).\end{gathered} \tag{4}$$

Minimising $-\delta_{i,j}$ maximises depth, therefore imparts MDH; the first constraint upper bounds the Euclidean approximation of geodesic distance $\|\delta_{i,j}\hat{\mathbf{q}}_{i,j}-\delta_{i,q}\hat{\mathbf{q}}_{i,q}\|$ by the estimated geodesic distance $d_{j,q}$ and the second constraint limits the scale of the reconstructed scene by mandating all geodesic distances to sum to one; these constraints allow the problem to be posed as SOCP. The problem formulation from [Ji et al., 2017] is an SDP reformulation of [Chhatkuli et al., 2017], hence equivalent to formulation (4) with a slightly less relaxed solution set due to the switch from SOCP to SDP. This reformulation replaces the inequality by its square, which is equivalent as both sides are positive, but cancels the square root of the Euclidean norm. It also introduces the square in the scale normalisation constraint.

Isometry can be introduced in formulation (4) by replacing the inequality constraint with equality, giving the following non-convex formulation, where we use the squared distance as in [Ji et al., 2017]:

$$\begin{aligned} &\min_{\{\delta_{i,j}\},\{d_{j,q}\}} -\sum_{i=1}^{n}\sum_{j=1}^{m}\delta_{i,j} \\ \text{s.t.} \quad & \|\delta_{i,j}\hat{\mathbf{q}}_{i,j}-\delta_{i,q}\hat{\mathbf{q}}_{i,q}\|^2 = d_{j,q}^2, \quad \sum_{j'=1}^{m}\sum_{q'\in\mathcal{N}(j')} d_{j',q'}^2 = 1, \quad \delta_{i,j}\geq 0, \quad \forall i\in[1,n], j\in[1,m], q\in\mathcal{N}(j). \end{aligned} \tag{5}$$

This formulation is however not generally solvable as exactly satisfying all isometry constraints prevents the structure from deforming, which is thus incompatible with the observed points. An interesting alternative is obtained by replacing the individual isometry constraints with their sum of squares and formulating the Lagrangian, using ℓ as the Lagrange multiplier:

$$\begin{aligned} &\min_{\{\delta_{i,j}\},\{d_{j,q}\}} -\sum_{i=1}^{n}\sum_{j=1}^{m}\delta_{i,j} + \ell\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{q\in\mathcal{N}(j)}\left(\|\delta_{i,j}\hat{\mathbf{q}}_{i,j}-\delta_{i,q}\hat{\mathbf{q}}_{i,q}\|^2 - d_{j,q}^2\right)^2 \\ &\text{s.t.} \quad \sum_{j'=1}^{m}\sum_{q'\in\mathcal{N}(j')} d_{j',q'}^2 = 1, \quad \delta_{i,j}\geq 0, \quad \forall i\in[1,n], j\in[1,m]. \end{aligned} \tag{6}$$

We observe that, by dropping the problematic isometry constraints, we thus obtain a maximally-isometric formulation, where ℓ can then be used to control the influence of isometry over depth maximisation. The depth maximisation term $\delta_{i,j}$ in equation (6) is linear but the isometry term $\left(\|\mathbf{Q}_{i,j}-\mathbf{Q}_{i,q}\|^2 - d_{j,q}^2\right)^2$ is quartic.

Remark 1. *Formulation (6) is a non-convex optimisation problem combining an MDH cost with a maximally-isometric cost, which is derived as a simple reformulation of [Chhatkuli et al., 2017; Ji et al., 2017]. Solving formulation (6) thus solves the formulation of [Chhatkuli et al., 2017; Ji et al., 2017] without the inextensible relaxation.*

We arrive at the problem statement by expanding formulation (6), introducing the visibility indicators, and a weight hyperparameter λ as:

$$\begin{aligned} &\min_{\{d_{j,q}\},\{\delta_{i,j}\}} \left(-\lambda\sum_{i=1}^{n}\sum_{j=1}^{m}\mathcal{V}_{i,j}\delta_{i,j} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{q\in\mathcal{N}(j)}\mathcal{V}_{i,j}\,\mathcal{V}_{i,q}\left(\delta_{i,j}^2+\delta_{i,q}^2-d_{j,q}^2-2\delta_{i,j}\delta_{i,q}\langle\hat{\mathbf{q}}_{i,j},\hat{\mathbf{q}}_{i,q}\rangle\right)^2\right) \\ &\text{s.t.} \quad \sum_{j'=1}^{m}\sum_{q'\in\mathcal{N}(j')} d_{j',q'}^2 = 1, \quad \{d_{j,q}\},\{\delta_{i,j}\}\geq 0, \quad \forall i\in[1,n], j\in[1,m], q\in\mathcal{N}(j). \end{aligned} \tag{7}$$

The weight λ has an important role, as it weights the MDH term with respect to the isometric term. We have chosen $\lambda = 1/\ell$ with $\lambda \geq 0$, because it allows one to entirely deactivate the MDH term, whereas the ℓ based parameterisation allows one to entirely deactivate the isometry term instead, which is not desirable as the formulation would degenerate and lead to a solution $\delta_{i,j} = +\infty$ and $d_{j,q}$ unconstrained. In contrast, as $\lambda \to +\infty$ the isometric term is given a lower influence, but is never deactivated. For $\lambda = 0$, only the isometric term is active, and the formulation is entirely isometric. It is meaningful but likely to have multiple discrete solutions, being subject to the typical convex-concave or 'flip' ambiguities of isometry. Therefore, choosing an appropriate value of λ is crucial in order to resolve the ambiguities with a limited bias to the isometric term. Classically, λ would be user-defined. However, we propose a method to choose it automatically, by following the principle that λ should be as small as possible but large enough to resolve all ambiguities. This principle is implemented in the proposed solution method given in the next section.

4.3 Solution Method

We solve the non-convex formulation (7) by an alternation scheme, successively solving for the depths and the geodesic distances. The method is given in algorithm 1.

4.3.1 Alternating Optimisation

We use the classical idea of Alternating Optimisation (AO) [Bezdek and Hathaway, 2002; Ortega and Rheinboldt, 2000]. Specifically, we use two rounds of AO in a nested manner. The first round, denoted AO_1, splits the parameters into geodesic distances $\{d_{j,q}\}$ and depths $\{\delta_{i,j}\}$, alternatively solving for one of the two sets of parameters while freezing the other. The second round, denote AO_2, splits the depth set, solving for the depth of a single 3D point while freezing the others. We denote the values at the r-th iteration of AO_1 as $(\cdot)^r$ and at the subsequent s-th iteration of AO_2 as $(\cdot)^{r,s}$; we thus only use s during AO_2. The initial estimate is thus denoted by $\{\delta^0_{i,j}\}$ and $\{d^0_{j,q}\}$.

4.3.2 Assumptions

Our method requires the following two assumptions.

Assumption 1. *The angle between the sight-lines of neighbouring points, i.e., the angle between vectors $\hat{\mathbf{q}}_{i,j}$ and $\hat{\mathbf{q}}_{i,q}$ for all $q \in \mathcal{N}(j)$, lies in the range $(0, \frac{\pi}{2})$.*

This assumption follows standard practice in NRS*f*M [Perriollat et al., 2011]. It ensures that the value of the inner product $\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q} \rangle$ lies in $(0, 1)$.

Assumption 2. *The geodesic distance between neighbouring point-pairs is smaller than the depth of these points, i.e., $\delta^2_{i,j}, \delta^2_{i,q} \geq d^2_{j,q}$ for all $q \in \mathcal{N}(j)$.*

This assumption is less standard than the first one but forms a mild requirement. It is satisfied for the vast majority of images of real-world objects with sufficient keypoint correspondences. We use it to guarantee that valid solutions are found; we give details in lemma 2.

4.3.3 Solving for the Geodesic Distances

We solve for the geodesic distances at iteration r while freezing the depths at their values at iteration $r-1$. Formulation (7) is thus rewritten as:

$$\begin{gathered}\min_{\{d^r_{j,q}\}} \Bigg(-\lambda \sum_{i=1}^{n}\sum_{j=1}^{m} \mathcal{V}_{i,j}\delta^{r-1}_{i,j} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{q\in\mathcal{N}(j)} \mathcal{V}_{i,j}\,\mathcal{V}_{i,q}\Big((\delta^{r-1}_{i,j})^2 + (\delta^{r-1}_{i,q})^2 - (d^r_{j,q})^2 \\ -2\delta^{r-1}_{i,j}\delta^{r-1}_{i,q}\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle\Big)^2\Bigg) \qquad \text{s.t.} \quad \sum_{j'=1}^{m}\sum_{q'\in\mathcal{N}(j')} (d^r_{j',q'})^2 = 1, \quad \{d^r_{j,q}\} \geq 0, \quad \forall j \in [1,m], q \in \mathcal{N}(j).\end{gathered} \tag{8}$$

We make two key observations. First, that each cost term only depends on a single geodesic distance. Second, the distances are only related to the normalisation constraint. However, a consequence of the AO scheme is that this constraint can be temporarily ignored. This constraint is important in the general formulation because it keeps the solution away from the trivial degenerate solution where all depths and distances would vanish. However, this degeneracy does not manifest in each individual round of AO. This is because, at these rounds, the depths are kept fixed and the distances cannot be set to zero without a cost increase. Therefore, the constraint can be safely ignored, provided that the depths are simply renormalised once the iteration is complete. Renormalisation is important, as it prevents the global reconstruction scale from drifting. It may increase the cost; this is normal, as the cost is scale-covariant, which is the property causing the degeneracy in the first place: as one increases or decreases the reconstruction scale, the cost also increases or decreases. This is a very common property of reconstruction problem statements in S*f*M and NRS*f*M. However, S*f*T escapes this ambiguity, as its template prescribes the object's geodesic distances, fixing the reconstruction scale. The problem in equation (8) can thus be rewritten as a series of smaller problems, one of which is to be solved for each geodesic distance, as:

$$\begin{gathered}\min_{d^r_{j,q}} \Bigg(-\lambda \sum_{i=1}^{n} \mathcal{V}_{i,j}\delta^{r-1}_{i,j} + \sum_{i=1}^{n} \mathcal{V}_{i,j}\,\mathcal{V}_{i,q}\Big((\delta^{r-1}_{i,j})^2 + (\delta^{r-1}_{i,q})^2 - (d^r_{j,q})^2 - 2\delta^{r-1}_{i,j}\delta^{r-1}_{i,q}\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle\Big)^2\Bigg) \\ \text{s.t.} \quad d^r_{j,q} \geq 0.\end{gathered} \tag{9}$$

We derive the solution to problem (9) by ignoring the inequality constraint and then show that the solution always satisfies the constraint. Upon differentiating the cost with respect to $d^r_{j,q}$ and nullifying, we arrive at:

$$d^r_{j,q} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big((\delta^{r-1}_{i,j})^2 + (\delta^{r-1}_{i,q})^2 - 2\delta^{r-1}_{i,j}\delta^{r-1}_{i,q}\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle\Big)}. \tag{10}$$

The properties of this solution are given by the next lemma.

Lemma 1. *There always exists a unique, real, and positive solution to equation (9) given by equation (10).*

Proof. We need to show that the operand of the square root in solution (10) is positive. This operand is a sum and we can thus show that each term of the sum is positive. This follows from:

$$\begin{aligned}(\delta_{i,j}^{r-1})^2 + (\delta_{i,q}^{r-1})^2 > 2\delta_{i,j}^{r-1}\delta_{i,q}^{r-1}\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle \qquad &\Leftrightarrow \qquad \frac{\delta_{i,j}^{r-1}}{\delta_{i,q}^{r-1}} + \frac{\delta_{i,q}^{r-1}}{\delta_{i,j}^{r-1}} > 2\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle \\ \Leftrightarrow \qquad \frac{(\delta_{i,j}^{r-1} + \delta_{i,q}^{r-1})}{2} &> \sqrt{\delta_{i,j}^{r-1}\delta_{i,q}^{r-1}}\sqrt{\frac{\left(\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle + 1\right)}{2}}.\end{aligned} \tag{11}$$

The left-hand-side in the second line of equation (11) is the arithmetic mean and the right-hand-side is the geometric mean multiplied by the term $\sqrt{\frac{(\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle+1)}{2}}$. Due to assumption 1, $\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle$ is within $(0, 1)$, therefore the term $\sqrt{\frac{(\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle+1)}{2}}$ is within $(\frac{1}{\sqrt{2}}, 1)$, *i.e.* always lower than one and positive. The fact that the arithmetic mean is always greater or equal to the geometric mean [Bhatia and Kittaneh, 2000] concludes the proof. □

4.3.4 Solving for the Depths

We solve for the depths at iteration r while freezing the geodesic distances at their previous values, already available for iteration r. Formulation (7) is thus rewritten as:

$$\begin{aligned}\min_{\{\delta_{i,j}^r\}} \Bigg(-\lambda \sum_{i=1}^{n}\sum_{j=1}^{m} \mathcal{V}_{i,j}\delta_{i,j}^r + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{q\in\mathcal{N}(j)} \mathcal{V}_{i,j}\,\mathcal{V}_{i,q}\Big((\delta_{i,j}^r)^2 + (\delta_{i,q}^r)^2 \\ -(d_{j,q}^r)^2 - 2\delta_{i,j}^r\delta_{i,q}^r\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle\Big)^2\Bigg) \qquad \text{s.t.} \quad \{\delta_{i,j}^r\} \geq 0, \quad \forall i \in [1, n], j \in [1, m].\end{aligned} \tag{12}$$

We observe that each cost term only depends on the depths from a single image. The problem is thus separable image-wise and can be rewritten as a series of n smaller problems:

$$\begin{aligned}\min_{\{\delta_{i,j}^r\}} \Bigg(-\lambda \sum_{j=1}^{m} \mathcal{V}_{i,j}\delta_{i,j}^r + \sum_{j=1}^{m}\sum_{q\in\mathcal{N}(j)} \mathcal{V}_{i,j}\,\mathcal{V}_{i,q}\Big((\delta_{i,j}^r)^2 + (\delta_{i,q}^r)^2 - (d_{j,q}^r)^2 - 2\delta_{i,j}^r\delta_{i,q}^r\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle\Big)^2\Bigg) \\ \text{s.t.} \quad \{\delta_{i,j}^r\} \geq 0, \quad \forall j \in [1, m].\end{aligned} \tag{13}$$

The formulation remains non-convex, involving a quartic cost, and cannot be solved in closed-form. We use a second round of AO, where we freeze all depths but one, using s as an iteration counter. We thus have:

$$\begin{aligned}\min_{\delta_{i,j}^{r,s}} \Bigg(-\lambda\mathcal{V}_{i,j}\delta_{i,j}^{r,s} + \sum_{q\in\mathcal{N}(j)} \mathcal{V}_{i,j}\,\mathcal{V}_{i,q}\Big((\delta_{i,j}^{r,s})^2 + (\delta_{i,q}^{r,s-1})^2 - (d_{j,q}^r)^2 - 2\delta_{i,j}^{r,s}\delta_{i,q}^{r,s-1}\langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle\Big)^2\Bigg) \\ \text{s.t.} \quad \delta_{i,j}^{r,s} \geq 0.\end{aligned} \tag{14}$$

Applying the first-order optimality condition, we obtain:

$$\begin{aligned}-\lambda + 2\sum_{q\in\mathcal{N}(j)} \Bigg(& 2(\delta^{r,s}_{i,j})^3 - 6(\delta^{r,s}_{i,j})^2 \delta^{r,s-1}_{i,q} \langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q}\rangle \\ & + 2\delta^{r,s}_{i,j}\Big((\delta^{r,s-1}_{i,q})^2(1+2(\langle \hat{\mathbf{q}}_{i,j},\hat{\mathbf{q}}_{i,q}\rangle)^2) - (d^r_{j,q})^2\Big) - 2\delta^{r,s-1}_{i,q}\langle\hat{\mathbf{q}}_{i,j},\hat{\mathbf{q}}_{i,q}\rangle\Big((\delta^{r,s-1}_{i,q})^2 - (d^r_{j,q})^2\Big)\Bigg) = 0.\end{aligned} \tag{15}$$

This is a cubic in $\delta^{r,s}_{i,j}$, whose properties are given in the next lemma.

Lemma 2. *For $\lambda \geq 0$ large enough, there is a unique, real, and positive solution to equation (15) for all points.*

Proof. The proof has two parts. We consider only one point; the result trivially generalises to multiple points by choosing the maximal value of λ over all points. First, we show that all the real solutions are positive. Second, we show that there exists λ_0 such that for any $\lambda \geq \lambda_0$ there is a unique real solution. We start with the first part by writing the cubic (15) as:

$$c_3(\delta^{r,s}_{i,j})^3 + c_2(\delta^{r,s}_{i,j})^2 + c_1\delta^{r,s}_{i,j} + c_0 = 0, \tag{16}$$

where:

$$\begin{gathered} c_3 = 4\,|\mathcal{N}(j)|, \qquad c_2 = -12\sum_{q=1}^{\mathcal{N}(j)} \delta^{r,s-1}_{i,q}\langle\hat{\mathbf{q}}_{i,j},\hat{\mathbf{q}}_{i,q}\rangle, \\ c_1 = 4\sum_{q=1}^{\mathcal{N}(j)}\left(\left(\delta^{r,s-1}_{i,q}(1+2\langle\hat{\mathbf{q}}_{i,j},\hat{\mathbf{q}}_{i,q}\rangle)\right)^2 - (d^r_{j,q})^2\right) \qquad \text{and} \\ c_0 = -4\sum_{q=1}^{\mathcal{N}(j)}\left(\delta^{r,s-1}_{i,q}\langle\hat{\mathbf{q}}_{i,j},\hat{\mathbf{q}}_{i,q}\rangle\Big((\delta^{r,s-1}_{i,q})^2 - (d^r_{j,q})^2\Big)\right) - \lambda. \end{gathered} \tag{17}$$

Recall that $|\mathcal{N}(j)| \in \mathbb{Z}^+$ is the number of visible neighbours for point j. Using $\lambda \geq 0$, $\delta^{r,s-1}_{i,q} > 0$ and assumptions 1 and 2, we have $c_3 > 0$, $c_2 < 0$, $c_1 \geq 0$ and $c_0 < 0$. The cubic has one or three real solutions, all positive. Indeed, following Descartes's rule of signs, as there are three sign changes across the ordered coefficients, there are exactly one or three positive solutions, and as there are no sign changes when negating the first and third coefficients, there cannot be negative solutions. In addition, because $c_0 \neq 0$, zero cannot be a solution.

We proceed with the second part of the proof. We observe that only c_0 depends on λ. The cubic discriminant is given by:

$$D = 18c_3c_2c_1c_0 - 4c_2^3c_0 + c_2^2c_1^2 - 4c_3c_1^3 - 27c_3^2c_0^2. \tag{18}$$

It forms a quadratic in λ whose leading term is $-27c_3^2\lambda^2$; as this leading coefficient is negative, the quadratic is represented as an upward pointing parabola. Recall that the cubic has only one real solution if and only if $D < 0$. We thus have two cases: either the parabola does not have a real root, in which case $D < 0$ for any value of λ, or the parabola has one repeated or two distinct real roots $\lambda_- \leq \lambda_+$, in which case $D < 0$ for $\lambda > \lambda_+$. □

4.3.5 Convergence Criteria and Analysis

We introduce function F as a shorthand for the cost we minimise in equation (7), to be used below in convergence analysis, given as:

$$\begin{aligned} F\Big(\{\delta_{i,j}^{r,s}\}, \{d_{j,q}^{r}\}\Big) &= -\lambda \sum_{i=1}^{n} \sum_{j=1}^{m} \mathcal{V}_{i,j} \delta_{i,j}^{r,s} \\ + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{q \in \mathcal{N}(j)} \mathcal{V}_{i,j}\, \mathcal{V}_{i,q} & \Big((\delta_{i,j}^{r,s})^2 + (\delta_{i,q}^{r,s})^2 - (d_{j,q}^{r})^2 - 2\delta_{i,j}^{r,s} \delta_{i,q}^{r,s} \langle \hat{\mathbf{q}}_{i,j}, \hat{\mathbf{q}}_{i,q} \rangle \Big)^2 . \end{aligned} \tag{19}$$

The convergence criteria for AO_1 and for AO_2 are, respectively:

$$\frac{1}{nm} |F\Big(\{\delta_{i,j}^{r,s}\}, \{d_{j,q}^{r}\}\Big) - F\Big(\{\delta_{i,j}^{r-1,s}\}, \{d_{j,q}^{r-1}\}\Big)| \leq \epsilon, \tag{20}$$

and:

$$\frac{1}{nm} |F\Big(\{\delta_{i,j}^{r,s}\}, \{d_{j,q}^{r}\}\Big) - F\Big(\{\delta_{i,j}^{r,s-1}\}, \{d_{j,q}^{r}\}\Big)| \leq \epsilon, \tag{21}$$

where ϵ is a small constant. The following lemma states that the proposed method converges.

Lemma 3. *For $\lambda \geq 0$ large enough, the proposed two-round AO converges to a local solution of problem (7).*

Proof. We use theorem 1 from [Hong et al., 2015], which we apply to AO_1 and AO_2. This is possible because 1) for both AO_1 and AO_2, the original feasible set is $\mathbb{R}_+$, which is unbounded from above but can be simply compactified by finding an upper-bound on the geodesic distance or depth from the isometric constraints, and 2) lemmas 1 and 2 show that for λ large enough both costs have a unique minimiser on the feasible set. □

4.3.6 Pseudo-code Implementation

We summarise the proposed maximally isometric NRS*f*M method by giving pseudo-code in algorithm 1, for which we explain the implementation details directly below.

The algorithm implements two main loops corresponding to the two rounds of AO. The outer loop from line 4 to line 19 implements AO_1; the inner loop from line 11 to line 17 implements AO_2. The geodesic distance solution in line 7 is a simple closed-form. The depth solution in line 15 is the closed-form solution for the single real cubic root [Kurosh, 2014]. We start with an initial MDH weight of zero in line 2 and increase it towards one as need be in the inner iteration in line 15 to ensure the resolution of all isometric ambiguities as they manifest, implementing a simple $\max$ operation with λ_+. Specifically, λ_+ is chosen as the largest real root of the quadratic discriminant (18) or as 0 if it does not have real roots. We fix the convergence threshold $\epsilon \leftarrow 1\text{e}{-12}$. The point clouds are eventually retrieved from the computed depths in line 20.

```
Algorithm 1: Isometric NRSfM
  Data: Image correspondences $\{\mathbf{q}_{i,j}\}$ and visibility indicators $\{\mathcal{V}_{i,j}\}$ for $i \in [1,n], j \in [1,m]$, NNG $\mathcal{N}$
  Result: Point clouds $\{\mathbf{Q}_{i,j}\}$ for $i \in [1,n], j \in [1,m]$
  Initialise the depths: $\{\delta^{0,0}_{i,j}\}$ for all $i \in [1,n]$, $j \in [1,m]$ from e.g. [Chhatkuli et al., 2017]
  Set the MDH weight: $\lambda \leftarrow 0$
  Set AO$_1$ iteration counter: $r \leftarrow 0$
  do
      Increase AO$_1$ iteration counter: $r \leftarrow r + 1$
      for $j \in [1,m]$, $q \in \mathcal{N}(j)$ do
          Solve for $d^r_{j,q}$ from equation (10)
      end
      for $i \in [1,n]$ do
          Set AO$_2$ iteration counter: $s \leftarrow 0$
          do
              Increase AO$_2$ iteration counter: $s \leftarrow s + 1$
              for $j \in [1,m]$ do
                  Increase MDH weight if required: $\lambda \leftarrow \max(\lambda, \lambda_+)$
                  Solve $\delta^{r,s}_{i,j}$ from problem (14)
              end
          while $|F\Big(\{\delta^{r,s}_{i,j}\}, \{d^r_{j,q}\}\Big) - F\Big(\{\delta^{r,s-1}_{i,j}\}, \{d^r_{j,q}\}\Big)| \leq nm\epsilon$;
      end
  while $|F\Big(\{\delta^{r,s}_{i,j}\}, \{d^r_{j,q}\}\Big) - F\Big(\{\delta^{r-1,s}_{i,j}\}, \{d^{r-1}_{j,q}\}\Big)| \leq nm\epsilon$;
  Retrieve the point clouds: $\|\mathbf{Q}_{i,j}\| \leftarrow \delta^{r,s}_{i,j}\, \tilde{\mathbf{q}}_{i,j}/\|\tilde{\mathbf{q}}_{i,j}\|$ for all $i \in [1,n], j \in [1,m]$
```

5 ToTem TO POINT CLOUD ALIGNMENT

Aligning the ToTem shape to the reconstructed point-clouds forms the second step of our reconstruction pipeline. We start by giving our strategy and the general problem statement, then adapt it to the three algebraic shapes, and finally to the general shapes, giving a solution method in each case. In particular, we present new methods consisting of multiple global optimal sub-steps to deal with the cylindrical and spherical ToTem cases.

5.1 Strategy and General Problem Statement

We compute the alignment of the ToTem shape to each point cloud independently. Theoretically, the alignment function is the deformation denoted ψ_i. Directly computing ψ_i, which has many degrees of freedom, would be very unstable because the point-clouds, resulting from NRSfM, are usually sparse and noisy. We thus need a more constrained model with as few degrees of freedom as possible. The similarity model is a scaled rigid model and thus has only 7 relevant degrees of freedom at most. It accounts for the global orientation with its rotational component, for the global position with its translational component, and for the global change in size by its scaling component. The similarity model is used to robustly initiate the parameterisation process and bootstraps the estimation of ψ_i in section 6.

We define the alignment problem for point-cloud $\mathbf{Q}_i = [\mathbf{Q}_{i,1}, \ldots, \mathbf{Q}_{i,m}]^\top \in \mathbb{R}^{m\times 3}$ with $i \in [1, n]$. We assume without loss of generality that all points are visible. The alignment function is denoted f_i with $f_i(\mathbf{Q_i}) = s_i \mathbf{R}_i \mathbf{Q}_i^\top + \mathbf{t}_i$, where $s_i > 0$ is the scale factor, $\mathbf{R}_i \in SO_3$ is a rotation matrix and $\mathbf{t}_i \in \mathbb{R}^3$ is the translation vector. Function f_i acts on the points of the ToTem shape $\mathcal{U}$ to produce an aligned shape $f_i(\mathcal{U})$. We define a cost function $\mathcal{E}_P$ that measures a fitting error between the point-cloud and the aligned shape as $\mathcal{E}_P\big(f_i(\mathcal{U}), \mathbf{Q}_i\big)$. We thus state the general alignment problem as:

$$\min_{s_i, \mathbf{R}_i, \mathbf{T}_i} \mathcal{E}_P\big(f_i(\mathcal{U}), \mathbf{Q}_i\big). \tag{22}$$

The cost function may for instance be the sum of squared distances between the points and the aligned shape, in which case a general solution is provided by the ICP approach. However, we choose the cost function on a case-by-case basis to facilitate the derivation of closed-form solutions for the three algebraic shapes. The key idea is that $f_i(\mathcal{U})$ is a shape of the same nature as $\mathcal{U}$ for the three algebraic shapes, so we reformulate the problem as a shape-fitting problem, from which we extract the $s_i, \mathbf{R}_i, \mathbf{t}_i$ parameters. For instance, a plane is transformed to a plane by the similarity f_i, so we fit the aligned plane directly, from which we extract the similarity parameters. A benefit of this approach is that it allows one to directly figure out which degrees of freedom of the similarity are not constrained by the fitted shape. For instance, plane fitting does not constrain the rotation around the plane normal.

5.2 Planar ToTem Alignment

Planar alignment is trivial, as it consists in fitting a plane to the point-cloud. We only have 3 degrees of freedom to resolve, namely 2 rotational and 1 translational. This is because the scale is irrelevant, as are the rotation about the plane normal and the in-plane translation. These 3 degrees of freedom are consistent with the 3 degrees of freedom of a plane in the 3D space.

We parameterise the aligned plane by its homogeneous coefficients $\mathbf{h}_i = (\mathbf{n}_i^\top \; d_i)^\top$ where $\mathbf{n}_i$, with $\|\mathbf{n}_i\| = 1$, is the normal vector and $|d_i|$ is some scalar. The classical point-to-plane distance minimisation problem is written as:

$$\min_{\mathbf{n}_i, d_i} \|\mathbf{Q}_i \mathbf{n}_i + d_i \mathbf{1}_{m\times 1}\|_2^2 \qquad \text{s.t.} \quad \|\mathbf{n}_i\|_2 = 1. \tag{23}$$

Differentiating equation (23) w.r.t. d_i and nullifying yields $d_i = \frac{(\mathbf{Q}_i \mathbf{n}_i)^\top \mathbf{1}_{m\times 1}}{m}$. As the centroid of $\mathbf{Q}_i$ is given by $\mathbf{C}_{i,\text{pln}} = \frac{\mathbf{Q}_i^\top \mathbf{1}_{m\times 1}}{m}$, we thus have that translating the coordinate frame by centring the point-cloud cancels d_i but leaves the normal vector unchanged. Therefore, the standard method of finding the singular vector associated with the smallest singular value by a Singular Value Decomposition (SVD) of the centred point-cloud matrix $\mathbf{Q}_i - \mathbf{C}_{i,\text{pln}}^\top$ directly gives $\mathbf{n}_i$. We then recover the parameters of f_i as follows. We set $s_i = 1$. The uv-space is a plane with normal $\mathbf{Z} = [0, 0, 1]^\top$. Hence, from $\mathbf{n}_i = \mathbf{R}_i \mathbf{Z}$, we know that the third column of $\mathbf{R}_i$ is $\mathbf{n}_i$. We thus find $\mathbf{R}_i$ by orthonormal completion, using the first factor $\mathbf{U}$ from an SVD of $\mathbf{n}_i$. The translation is then simply given by $\mathbf{T}_i = d_i\mathbf{Z}$.

5.3 Cylindrical ToTem Alignment

We propose a two-step closed-form solution to cylindrical alignment. We represent the cylinder by its axis and its radius, hence 5 degrees of freedom. Recall that a 3D line has 4 degrees of freedom. We only have 5 degrees of freedom to resolve, namely 1 scaling, 2 rotational, and 2 translational. This is because the rotation about the cylinder axis is irrelevant, as is the along-axis translation. These 5 degrees of freedom are consistent with the 5 degrees of freedom of a 3D cylinder.

We break down the cylinder axis into an orientation and a base point. We start with step 1), where we compute the axis' orientation and deduce the similarity's rotation. Fitting a cylinder to a point-cloud is a naturally ambiguous problem, should the point-cloud not possess a dominant direction. The worst case is trivially made of approximately isotropic point-clouds, where any cylinder orientation would fit equally well to a fair extent. These degenerate cases are however rare; we rule them out with the following widely applicable assumption.

Assumption 3. *The cylindrical shape is thin, meaning that the radius of the cylinder resembling the point cloud is smaller than its length.*

We can thus use the dominant point-cloud direction as cylinder axis orientation, denoted $\mathbf{\Lambda}_i$. A simple way to find it is by using the singular vector corresponding to the dominant singular value of $\mathbf{Q}_i$ from an SVD. The cylindrical ToTem is defined with an orientation along the positive Y-axis, given by vector $\mathbf{Y} = [0, 1, 0]^\top$. We proceed similarly to the planar case to retrieve $\mathbf{R}_i$, using $\mathbf{\Lambda}_i = \mathbf{R}_i\mathbf{Y}$ and an SVD of $\mathbf{\Lambda}_i$ to perform orthonormal completion. One could avoid requiring assumption 3 by testing the three main directions in turn and simply keeping the one yielding the lowest final fitting residual.

We proceed with step 2), where we compute the radius and axis' base point, and deduce the similarity's scale and translation. We rotate the point-cloud to align it to the ToTem in terms of orientation, defining $\mathbf{Q}'_i = \mathbf{R}_i^\top \mathbf{Q}_i$. Should it be perfectly cylindrical, the rotated point-cloud $\mathbf{Q}'_i$ would lie normal to the XZ-plane. The idea for determining the radius and base point is to orthogonally project the rotated point-cloud to the XZ-plane, giving points $\mathbf{q}'_i$. We then fit a base circle, representing the cylinder's profile, with radius $r_i \geq 0$ and centre $\boldsymbol{\mathcal{C}}_i = [C_x, C_y]^\top \in \mathbb{R}^2$. For a point to be on the circle, its distance to the centre must be equal to the radius. We turn this constraint into a cost function, by minimising the sum of squared differences of the squared radius and point to centre distance, as:

$$\min_{r_i, \boldsymbol{\mathcal{C}}_i} \sum_{j=1}^{m} \left(r_i^2 - \|\mathbf{q}'_{i,j} - \boldsymbol{\mathcal{C}}_i\|_2^2 \right)^2. \tag{24}$$

As the cost is quartic, the problem cannot be directly solved in closed-form with standard methods. We propose to substitute the variable $\varrho_i = r_i^2 - \|\boldsymbol{\mathcal{C}}_i\|_2^2$, obtaining:

$$\min_{\varrho_i, \boldsymbol{\mathcal{C}}_i} \sum_{j=1}^{m} \left(\left(\varrho_i + 2(\mathbf{q}'_{i,j})^\top \boldsymbol{\mathcal{C}}_i \right) - \|\mathbf{q}'_{i,j}\|_2^2 \right)^2 \quad \text{s.t.} \quad \varrho_i + \|\boldsymbol{\mathcal{C}}_i\|_2^2 \geq 0. \tag{25}$$

The additional constraint stems from the change of variable implying $\varrho_i^2 + \|\boldsymbol{\mathcal{C}}_i\|_2^2 = r_i^2 \geq 0$. The cost is now in the form of linear least-squares and can be expressed in matrix form as:

$$\left\| \begin{bmatrix} 1 & 2(\mathbf{q}'_{i,1})^\top \\ \vdots & \vdots \\ 1 & 2(\mathbf{q}'_{i,m})^\top \end{bmatrix} \begin{bmatrix} \varrho_i \\ \boldsymbol{\mathcal{C}}_i \end{bmatrix} - \begin{bmatrix} \|\mathbf{q}'_{i,1}\|_2^2 \\ \vdots \\ \|\mathbf{q}'_{i,m}\|_2^2 \end{bmatrix} \right\|_2^2 . \tag{26}$$

The constraint can be safely ignored; as the cost is convex, either the unconstrained global minimum is within the feasible set, and thus matches the constrained global minimum, or it is outside the feasible set, in which case the constrained global minimum is on the feasible set's boundary. This would be a degenerate case with $r_i = 0$, reducing the circle to a single point. Our implementation handles it, though we have not met the case in practice. We eventually solve the unconstrained problem with the matrix pseudo-inverse technique to recover ϱ_i and $\boldsymbol{\mathcal{C}}_i$, from which the radius is extracted as $r_i = \sqrt{\max(0, \varrho_i + \|\boldsymbol{\mathcal{C}}_i\|_2^2)}$. Recall that the ToTem cylinder has radius 1 and an axis parallel to axis Y passing by the origin. We thus set $s_i = 1/r_i$ and $\mathbf{t}_i = \mathbf{R}_i \, [C_x, 0, C_y]^\top$.

5.4 Spherical ToTem Alignment

We represent the sphere by its centre and its radius, hence 4 degrees of freedom. We only have 4 degrees of freedom to resolve, namely 1 scaling and 3 translational. This is because the rotation is irrelevant. These 4 degrees of freedom as consistent with the 4 degrees of freedom of a sphere. We denote the radius as $r_i \geq 0$ and the centre as $\boldsymbol{\mathcal{C}}_i \in \mathbb{R}^3$. We formulate the problem similarly to the base circle fitting formulation (24) and solve it with the same change of variable and constraint handling method. Recall that the ToTem sphere is zero-centred and of unit radius. We thus eventually set the parameters of f_i as $s_i = 1/r_i$, $\mathbf{R}_i = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix of appropriate size, and $\mathbf{t}_i = \boldsymbol{\mathcal{C}}_i$.

5.5 General Shape ToTem Alignment

The general case holds when the ToTem shape is not an algebraic shape but a general shape represented by a triangular mesh. The alignment with a point-cloud is part of a long-standing and extremely difficult problem. The problem is particularly difficult in the case at hand, owing to the point-cloud being a deformed (hence, non-rigid) version of the ToTem shape, and being subject to reconstruction error arising from the input data artefacts, including inaccurate correspondences, surface stretching violating isometry significantly and NRSfM ambiguities. There does not exist a universally agreed method to solve for the alignment under these conditions. In most practical applications, however, additional constraints are available to pre-align the ToTem shape and the point-cloud. Specifically, we use two types of constraints, whose specific uses are explained on a case-by-case basis in the experiments. The first type of constraints are the known correspondences between the ToTem shape and the images, which are naturally extended to correspondences with the point-clouds. This is the most common type of constraint, which exists for instance in surgical augmented reality applications in the form of anatomical landmarks. For instance, we perform an experiment with laparoscopic liver images where the ToTem shape is provided by a CT scan, and a few corresponding curves are identified by the physician [Rabbani et al., 2022]. These corresponding curves can be used to compute a pre-alignment which can then be refined by means of standard ICP [Besl and McKay, 1992] and its modern implementations such as the branch-and-bound method [Yang et al., 2015]. The second type of constraints is the assumption of small deformation. This typically occurs for stiff objects or objects undergoing strong environmental constraints. The alignment can be computed by means of ICP directly from a canonical identity initialisation.

6 PARAMETERISATION

Computing the parameterisation of the reconstructed point-clouds forms the third step of our reconstruction pipeline. The objective of parameterisation is to find, for each reconstructed point correspondence with index $j \in [1, m]$, a 2D point $\mathbf{p}_j \in \mathcal{T}$ in uv-space. This means that a consensus between the point-clouds must be found, which is challenging because, owing to the natural parameterisation freedom, there exist multiple equivalent parameterisations. For instance, the rotation is arbitrary for the spherical ToTem. We thus proceed in two steps, which parameterise each point-cloud and then resolve the parameterisation discrepancies. The first step computes an independent parameterisation for the point-cloud of each image, resulting in the 2D points $\{\mathbf{r}_{i,j} \in \mathcal{T}\}$ with $i \in [1, n]$, $j \in [1, m]$. The independent parameterisations are incoherent, which means that corresponding points across the point-clouds may have different uv-space coordinates. Beyond parameterisation freedom, this owes to several reasons, which we study in detail below, including the differences that exist between the ToTem shape and the actual object shape as reconstructed in the point-clouds. The second step unifies these independent parameterisations in a coherent manner, so that corresponding points have the same uv-coordinates, regardless of the considered image, resulting in the 2D points $\{\mathbf{p}_j \in \mathcal{T}\}$ with $j \in [1, m]$. These two steps are described in the next two sections,

6.1 Independent Image-wise Parameterisation

We compute an image-wise parameterisation by treating each point-cloud independently. In the next three sections, we give the general principle, the initialisation method, and the refinement method.

6.1.1 General Principle

By definition of the embedding Δ and the aligning similarity f_i, the uv-coordinates $\mathbf{r}_{i,j}$ must satisfy:

$$f_i\left(\Delta\left(\mathbf{r}_{i,j}\right)\right) = \mathbf{Q}_{i,j} + \mathbf{E}_{i,j}, \tag{27}$$

where $\mathbf{E}_{i,j} \in \mathbb{R}^3$ is an unknown vector which models reconstruction noise and object deformation. Inverting this equation to find $\mathbf{r}_{i,j}$ requires one to handle the fact that Δ^{-1} is only defined for 3D points on the ToTem shape $\mathcal{U}$. The key idea is to transfer the point-cloud $\mathbf{Q}_i$ to the ToTem shape in canonical pose using the inverse aligning similarity transformation f_i^{-1} as $\mathbf{Q}^a_{i,j} = f_i^{-1}(\mathbf{Q}_{i,j})$, find their closest point on the ToTem shape as $\mathbf{Q}^p_{i,j} = \mathcal{P}(\mathbf{Q}^a_{i,j})$, where $\mathcal{P}$ is the closest point operator, and compute the uv-space coordinates by the flattening transformation $\Delta^{-1} : \mathcal{U} \mapsto \mathcal{T}$. The uv-space coordinates are thus obtained as:

$$\mathbf{r}_{i,j} = \Delta^{-1}\left(\mathcal{P}\left(f_i^{-1}\left(\mathbf{Q}_{i,j}\right)\right)\right). \tag{28}$$

We instantiate this general procedure for each of the three algebraic shapes and the general shapes in turn.

6.1.2 Initialisation Method

Our initialisation method uses equation (28). Specifically, the aligning similarity f_i is computed as described in section 5 and the flattening transformation Δ is implemented as described in section 3.2, which is a closed-form for the algebraic shapes and exploits standard barycentric coordinates for the general shapes. Therefore, it only remains to define the closest point operator $\mathcal{P}$, for each of the algebraic shapes and the general shapes.

Algebraic shapes. The closest point operator is given by selecting the closest orthogonal projection. For the planar ToTem, the canonical plane is the XY-plane. The projection is thus:

$$\mathbf{Q}^p_{i,j} = [X^p_{i,j}, Y^p_{i,j}, Z^p_{i,j}]^\top = [X^a_{i,j}, Y^a_{i,j}, 0]^\top. \tag{29}$$

For the cylindrical ToTem, the canonical cylinder axis is the Y-axis of the coordinate frame and its radius is 1. The projection is thus:

$$\mathbf{Q}^p_{i,j} = [X^p_{i,j}, Y^p_{i,j}, Z^p_{i,j}]^\top = [\alpha X^a_{i,j}, Y^a_{i,j}, \alpha Z^a_{i,j}]^\top \quad \text{with} \quad \alpha = \frac{1}{\sqrt{(X^a_{i,j})^2 + (Z^a_{i,j})^2}}. \tag{30}$$

For the spherical ToTem, the canonical sphere is zero centred and its radius is 1. The projection is thus:

$$\mathbf{Q}^{p}_{i,j} = \alpha \mathbf{Q}^{a}_{i,j} \quad \text{with} \quad \alpha = \frac{1}{\|\mathbf{Q}^{a}_{i,j}\|_2}. \tag{31}$$

Singularity avoidance. For the cylindrical and spherical ToTems, some of the orthogonally projected points in $\{\mathcal{P}\left(f_i^{-1}\left(\mathbf{Q}_{i,j}\right)\right)\}$ from equations (30) and (31) may lie on or close to the singularity of these shape parameterisations, which, from table 2, lie at $\theta = \pi$ and $\theta_2 = \pi$ for the cylinder and sphere respectively. We propose a singularity avoidance correction, based on the fact that the reconstructed points $\{\mathbf{Q}_{i,j}\}$ are sparse. From this sparsity, there must exist empty spaces in the vicinity of the points where the singular strip or patch can be safely placed without points lying within a small distance ϵ_{s}. Unless the reconstructed points are dense, which they are certainly not, the existence of such 'empty spaces' is guaranteed. The cylindrical and spherical ToTems come equipped with redundant rotational degrees of freedom that allow us to search for such an empty region without affecting the overall problem setup, as shown in the last column of table 2. This is practically implemented by a simple line-search along θ or θ_2 for the cylinder and sphere respectively. The line-search terminates when no points in $\{\mathcal{P}\left(f_i^{-1}\left(\mathbf{Q}_{i,j}\right)\right)\}$ lie within an ϵ_{s} distance from the singularity. This line-search generates a small corrective rotational transformation g_i. The rigid alignment, orthogonal projection, and singularity avoidance correction can be written as $g_i\left(\mathcal{P}\left(f_i^{-1}\left(\mathbf{Q}_{i,j}\right)\right)\right) = \mathcal{P}\left(g_i\left(f_i^{-1}\left(\mathbf{Q}_{i,j}\right)\right)\right)$, since these transformations commute. With a slight abuse of notation, we denote the composition of transformations $g_i \circ f_i^{-1}$ simply as f_i^{-1} in the sequel.

General shapes. The general shape ToTem uses a triangular mesh representation. We compute the distance between the point $\mathbf{Q}^{a}_{i,j}$ and each triangle to select the closest triangle, in which we compute the closest point. Our method uses the triangle's supporting plane $\mathcal{A}$. It first computes the distance between $\mathbf{Q}^{a}_{i,j}$ and $\mathcal{A}$, which provides a lower-bound on the distance to the triangle, and uses it as a means to quickly eliminate the triangle from the search. For a triangle that passes this initial test, the method then projects $\mathbf{Q}^{a}_{i,j}$ on $\mathcal{A}$. If the projection falls inside the triangle, then it keeps it as the closest point; if it falls outside the triangle, then this means that the closest point must lie on one of the triangle's edges. In this case, the method projects $\mathbf{Q}^{a}_{i,j}$ to the lines supporting the edges and performs a similar test as for the supporting plane: if the projection falls within at least one edge, then the closest in-edge projection is kept as closest point; if it falls outside all edges, then this means that the closest point must be one of the triangle's vertices, and the closest vertex is chosen. This procedure is extremely fast and efficient.

Although the definition of general shapes does not involve singularities, the nature of conformal flattening requires an open patch, sometimes called a 'seam', that acts as the boundary of the flattened mesh. This open patch is a pseudo-singularity around which Δ remains undefined. This is handled in our approach by placing the open patch away from the viewing direction of the camera, ensuring that none of the reconstructed points fall close to the pseudo-singularity.

6.1.3 Refinement Method

Refining the image-wise parameterisation is done by leaving the similarity approximation f_i and returning to the original deformation function ψ_i. With this model, one can introduce more advanced priors to better constrain the parameterisation. This is conceptually simple but complex to realise, for two main reasons. First, an ambiguity exists between the parameterisation and the deformation. In other words, to some extent, a point can be moved in the uv-space and the 3D deformation adapted to produce a similar prediction in the NRS*f*M reconstruction space. This ambiguity can be resolved by using strong priors. Second, the priors must be specifically designed: as our goal is to refine the parameterisation, the priors should be essentially placed to constrain the deformation and not the parameterisation. However, the isometric deformation prior is not applicable in the image-wise case. We deal with these two challenges by a specific data term and two deformation regularisation priors. We thus evolve equation (27) to:

$$\psi_i\left(\Delta\left(\mathbf{r}_{i,j}\right)\right) = \mathbf{Q}_{i,j} + \mathbf{E}'_{i,j}, \tag{32}$$

where $\mathbf{E}'_{i,j} \in \mathbb{R}^3$ is an unknown vector which models only reconstruction noise, whilst object deformation is now covered by ψ_i. We thus introduce a cost function g to refine the parameterisation and estimate ψ_i jointly as follows:

$$\min_{\{\mathbf{r}_{i,j}\},\mathbf{C}_i} g(\{\mathbf{r}_{i,j}\},\mathbf{C}_i) \quad \text{with} \quad g(\{\mathbf{r}_{i,j}\},\mathbf{C}_i) = g_{\text{data}}(\{\mathbf{r}_{i,j}\},\mathbf{C}_i) + \kappa_0\, g_{\text{def0}}(\{\mathbf{r}_{i,j}\},\mathbf{C}_i) + \kappa_1\, g_{\text{def1}}(\{\mathbf{r}_{i,j}\},\mathbf{C}_i), \tag{33}$$

where $\kappa_0, \kappa_1 > 0$ are hyperparameters and $\mathbf{C}_i$ represents the target control points of ψ_i, following the polyharmonic model of section 3.4. The cost function has three terms, which we explain one by one. The data term g_{data} measures the closeness of the points $\psi(\Delta(\mathbf{r}_{i,j}))$ predicted by the model and the points $\mathbf{Q}_{i,j}$ from the NRS*f*M point-cloud. In order to preserve flexibility, we project the error vector to the surface normal, as in the standard ICP cost, as:

$$g_{\text{data}}\left(\{\mathbf{r}_{i,j}\},\mathbf{C}_i\right) = \sum_{j=1}^{m} \Big(\left(\psi_i\left(\Delta\left(\mathbf{r}_{i,j}\right)\right) - \mathbf{Q}_{i,j}\right)^\top \eta[\psi_i \circ \Delta]\left(\mathbf{r}_{i,j}\right) \Big)^2 . \tag{34}$$

Recall that η is the normal operator, giving the surface normal at the point $\psi_i(\Delta(\mathbf{r}_{i,j}))$. The two other terms are regularisation terms, which constrain the deformation to remain small, by penalising deviation from the dominant similarity transformation f_i. The first regularisation term acts at the zeroth differential order, directly on the point position, as:

$$g_{\text{def0}}\left(\{\mathbf{r}_{i,j}\},\mathbf{C}_i\right) = \sum_{j=1}^{m} \left\|\psi_i\left(\Delta\left(\mathbf{r}_{i,j}\right)\right) - f_i\left(\Delta\left(\mathbf{r}_{i,j}\right)\right)\right\|_2^2 . \tag{35}$$

The second regularisation term acts at the first differential order, on the normals, as:

$$g_{\text{def1}}\left(\{\mathbf{r}_{i,j}\}, \mathbf{C}_i\right) = \sum_{j=1}^{m} \left\|\eta[\psi_i \circ \Delta]\left(\mathbf{r}_{i,j}\right) - \eta[f_i \circ \Delta]\left(\mathbf{r}_{i,j}\right)\right\|_2^2. \tag{36}$$

The cost function g is a sum of squares and the problem is thus a nonlinear least-squares minimisation. We set the hyperparameters in the range $\kappa_0 \in (0, 0.1)$ and $\kappa_1 \in (0.1, 0.25)$. We use the Levenberg-Marquardt (LM) algorithm with f_i as the initial solution, using the property that a polyharmonic spline can exactly reproduce a similarity transformation. We achieve this initialisation by setting the target control points by transferring the fixed source control points as $\mathbf{C}_i = f_i(\mathbf{D}_i)$.

6.2 Coherent Multi-image Parameterisation

In an ideal setting, the independent parameterisations $\{\mathbf{r}_{i,j} \in \mathcal{T}\}$ should perfectly match, satisfying $\mathbf{r}_{1,j} = \cdots = \mathbf{r}_{n,j}$, for $j \in [1, m]$. In other words, a point from the j-th correspondence should have the same parameterisation regardless of the point-cloud $i \in [1, n]$ from which this parameterisation is computed. If this property held, we could simply combine the independent parameterisations to retrieve the coherent parameters $\{\mathbf{p}_j \in \mathcal{T}\}$, for instance using averaging or robust averaging to handle noise. In practice however, this property does not hold and it is of key importance to understand the reasons why, in order to devise a correct method to compute the coherent parameterisation. The first reason is noise, which manifests itself in the reconstructed point-clouds and may have a strong impact in the parameters as it propagates through the independent parameterisation process. The second reason is the unresolved degrees of freedom in the alignment process. This mainly occurs for the algebraic ToTem shapes, which we analyse in detail in the previous section. For instance, the sphere rotation is arbitrary, which may cause strong parameter offsets across the images. The third reason is the deformation. As the alignment is rigid, it does not perfectly match the ToTem shape to the point-cloud, resulting in parameterisations with dependencies on the actual deformation state of the point-cloud, which varies across the images.

The main consequence of these three reasons is that the independent parameterisations must be unified by means of registration with a deformable transformation model. Concretely, we have the n independent parameterisations $\{\mathbf{r}_{i,j}\}$ in uv-space, each represented by a point-cloud of size m, and we search for a coherent parameterisation $\{\mathbf{p}_j\}$ representative of these point-clouds. We propose to minimise the distance between the coherent parameterisation $\{\mathbf{p}_j\}$ and each of the independent parameterisations $\{\mathbf{r}_{i,j}\}$ up to unknown deformable transformations, accounting for the above explained discrepancies. We identify this problem as an instance of the deformable GPA problem, to which a practical closed-form solution was recently given [Bai and Bartoli, 2022], where the transformations are represented by TPS.

7 SURFACE-BASED NON-RIGID BUNDLE ADJUSTMENT

The surface-based global refinement in a bundle adjustment manner forms the fourth step of our reconstruction pipeline. We first give the motivation and problem statement, then an efficient optimisation scheme.

7.1 Motivation and Problem Statement

The goal of bundle adjustment is to minimise a cost modelling the data and the priors in a principled manner. We propose a formulation adapted to deformable objects with known topology, which uses and refines the coherent parameterisation achieved by the previous steps. The key difference with the previous step, which already refines the parameterisation, is that bundle adjustment does it coherently for the whole set of images, allowing one to use the reprojection error and to re-introduce the isometric deformation prior and surface smoothing whilst taking advantage of the proposed embedding and deformation framework.

We start from 3D point prediction obtained from the model of section 3 as:

$$\mathbf{Q}_{i,j} = \psi_i(\Delta(\mathbf{p}_j)). \tag{37}$$

By combining with the camera projection function Π, we obtain the prediction of the j-th point in the i-th image $\mathbf{q}_{i,j}$ as:

$$\hat{\mathbf{q}}_{i,j} = \Pi(\psi_i(\Delta(\mathbf{p}_j))). \tag{38}$$

In this model, the unknowns are the parameterisation $\{\mathbf{p}_j\}$ and the deformation function $\{\psi_i\}$; in analogy to classical rigid BA, these are respectively the structure and the motion. We thus formulate surface-based BA as a minimisation problem over the point parameterisation in uv-space $\{\mathbf{p}_j\}$, $j \in [1, m]$ and the 3D deformation function parameters $\{\mathbf{C}_i\}$, $i \in [1, n]$, as:

$$\min_{\{\mathbf{p}_j\}, \mathbf{C}_i} h(\{\mathbf{p}_j\}, \mathbf{C}_i), \tag{39}$$

where h is a compound cost function, where we use three terms with hyperparameters $\nu_0, \nu_1 > 0$, as:

$$h(\{\mathbf{p}_i\}, \mathbf{C}_i) = h_{\text{rep}}(\{\mathbf{p}_j\}, \mathbf{C}_i) + \nu_0 \, h_{\text{iso}}(\{\mathbf{p}_j\}, \mathbf{C}_i) + \nu_1 \, h_{\text{smooth}}(\{\mathbf{p}_j\}, \mathbf{C}_i). \tag{40}$$

The first cost term h_{rep} is the data term, which relates the model to the data, using the standard reprojection error, as:

$$h_{\text{rep}}(\{\mathbf{p}_j\}, \mathbf{C}_i) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \mathbf{q}_{i,j} - \Pi\Big(\psi_i\big(\Delta(\mathbf{p}_j)\big)\Big) \right\|^2. \tag{41}$$

The second cost term h_{iso} represents the isometric deformation prior. We follow the standard way of penalising the deviation of inter-point Euclidean distances from the geodesic distances. In our formulation, however, the geodesic distances are estimated indirectly, as the effect of optimising the parameterisation, causing the points to move on the surface of $\mathcal{U}$. For a point-pair (j, q), the geodesic distance between $\Delta(\mathbf{p}_j)$ and $\Delta(\mathbf{p}_q)$ is thus obtained by a function of the parameters as $\mathfrak{g}(\mathbf{p}_j, \mathbf{p}_q)$. This allows us to define the isometric cost by the least-squares difference of inter-point distances as:

$$h_{\text{iso}}(\{\mathbf{p}_j\}, \mathbf{C}_i) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{q \in \mathcal{N}(j)} \left(\sqrt{\|\psi_i\left(\Delta\left(\mathbf{p}_j\right)\right) - \psi_i\left(\Delta\left(\mathbf{p}_q\right)\right)\|^2 + \epsilon} - \sqrt{\mathfrak{g}(\mathbf{p}_j, \mathbf{p}_q)^2 + \epsilon} \right)^2 . \tag{42}$$

We use ϵ as a small constant, whose effect is to smooth the cost function, which we consistently choose as $\epsilon = 10^{-2}$. For the planar and cylindrical ToTem, $\mathfrak{g}(\mathbf{p}_j, \mathbf{p}_q)$ is trivially the Euclidean distance in the uv-space while for the spherical ToTem it is the standard great circle length [Pressley, 2010], check appendix A for details. For the general ToTem shapes, there does not exist a closed-form for the geodesic distance. We thus compute it numerically from the shortest path found by the A* algorithm, using a standard efficient method [Calla et al., 2019].

The first cost term h_{smooth} represents the surface smoothness prior. It acts as a regulariser, designed to smooth the reconstructed surfaces. We use the classical Gaussian smoothing approach [Horn and Weldon, 1986]. We implement it by densely sampling the surface by b points, using $b = 10^4$ points. We choose these points in uv-space with parameterisation $\{\mathbf{p}_r\}$, $r \in [1, b]$ by uniformly sampling the convex-hull of $\{\mathbf{p}_j\}$ in $\mathbb{R}^2$. As for the NNG, we use $\mathcal{N}'$ to denote the neighbouring points operator, where $s \in \mathcal{N}'(r)$ means that $\mathbf{p}'_r$ and $\mathbf{p}'_s$ are neighbours. This leads to the following smoothing term:

$$h_{\text{smooth}}(\{\mathbf{p}_j\}, \mathbf{C}_i) = \sum_{i=1}^{n} \sum_{r=1}^{b} \sum_{s \in \mathcal{N}'(r)} \|\psi_i(\Delta(\mathbf{p}'_r)) - \psi_i(\Delta(\mathbf{p}'_s))\|^2. \tag{43}$$

Gaussian smoothing is simple and effective but has a known downside, which is the uncontrolled shrinkage of the surface. This is trivially avoided in the compound cost, thanks to the other two cost terms which do not share this side effect.

7.2 An Efficient Separable Implementation

The proposed formulation (39) is a non-linear least-squares problem, which, as usual in BA, can be handled by means of the LM method [Triggs et al., 1999]. This iterative method is guaranteed to converge to a local minimum, is generally fast, and handles the problem parameterisation ambiguities by damping. In theory, an off-the-shelf LM implementation [Kelley, 1999] would be sufficient. This would, however, be highly inefficient in terms of numerical stability and computation cost, as discussed below. We describe the structure of the LM update and then the proposed adaptation to surface-based non-rigid BA.

Structure of the LM update. We define the parameter vector as $\boldsymbol{\xi} \in \mathbb{R}^{2m+3nm}$ with

$$\boldsymbol{\xi} = \left(\mathbf{p}_1^\top, \ldots, \mathbf{p}_m^\top, \mathrm{vec}(\mathbf{C}_1)^\top, \ldots, \mathrm{vec}(\mathbf{C}_n)^\top\right)^\top \tag{44}$$

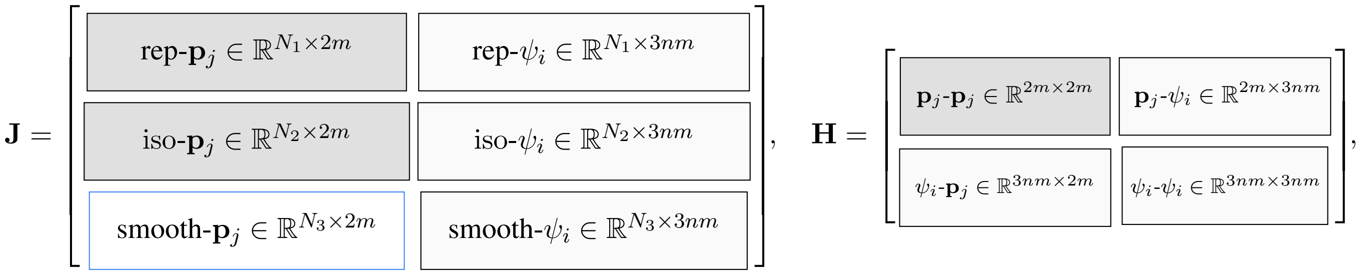
where vec is matrix vectorisation. We define the residual vector $\mathbf{E}_{\mathrm{BA}} \in \mathbb{R}^N$ as the stacking of the residuals from the three cost terms h_{rep}, h_{iso} and h_{smooth}, such that $\mathbf{E}_{\mathrm{BA}}^\top \mathbf{E}_{\mathrm{BA}} = h(\{\mathbf{p}_j\}, \mathbf{C}_i)$, where the total number of residuals $N = N_1 + N_2 + N_3$ is the sum of the number of residuals for each cost term. A typical LM update $\boldsymbol{\xi}'$ is then given by solving the augmented normal equations:

$$(\mathbf{H} + \lambda_{\mathrm{LM}} \mathbf{I})\, \boldsymbol{\xi}' = \mathbf{J}^\top \mathbf{E}_{\mathrm{BA}} \quad \text{with} \quad \mathbf{H} = \mathbf{J}^\top \mathbf{J}, \tag{45}$$

where $\mathbf{J} \in \mathbb{R}^{N \times (2m+3nm)}$ is the Jacobian matrix of $\mathbf{E}_{\mathrm{BA}}$, $\mathbf{H} \in \mathbb{R}^{(2m+3nm)\times(2m+3nm)}$ the Gauss-Newton approximation of the Hessian matrix and λ_{LM} the damping factor.

The main computational bottleneck of standard LM is solving the normal equations (45). For the standard solver, this computation is prohibitive when the number of images or keypoints are large, requiring hours or days for $n \sim 50$ and $m \sim 100$ on a modern desktop computer. We present an adapted method following the classical BA strategies to exploit the sparsity pattern of the Hessian matrix [Triggs et al., 1999] and specific approximations of the non-rigid transformations.

Block structure of the Jacobian and Hessian matrices. The Jacobian and Hessian matrices possess strong block structures. The Jacobian matrix has six blocks arising out of the partial derivatives of the three cost terms with respect to the point parameterisation $\{\mathbf{p}_j\}$ and to the control points of ψ_i, $\{\mathbf{C}_i\}$. We denote these blocks as $*$-$\dagger$, with $* \in \{\mathrm{rep}, \mathrm{iso}, \mathrm{smooth}\}$ and $\dagger \in \{\mathbf{p}_j, \psi_i\}$. For instance, the top-left hand corner block is rep-$\mathbf{p}_j$. The Hessian has four blocks arising out of the interactions between the parameters. We denote these blocks are $\dagger$-$\dagger$. For instance, the top-left hand corner block is $\mathbf{p}_j$-$\mathbf{p}_j$. These blocks can be shown pictorially as:



$$\mathbf{J} = \begin{bmatrix} \boxed{\mathrm{rep}\text{-}\mathbf{p}_j \in \mathbb{R}^{N_1 \times 2m}} & \boxed{\mathrm{rep}\text{-}\psi_i \in \mathbb{R}^{N_1 \times 3nm}} \\ \boxed{\mathrm{iso}\text{-}\mathbf{p}_j \in \mathbb{R}^{N_2 \times 2m}} & \boxed{\mathrm{iso}\text{-}\psi_i \in \mathbb{R}^{N_2 \times 3nm}} \\ \boxed{\mathrm{smooth}\text{-}\mathbf{p}_j \in \mathbb{R}^{N_3 \times 2m}} & \boxed{\mathrm{smooth}\text{-}\psi_i \in \mathbb{R}^{N_3 \times 3nm}} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \boxed{\mathbf{p}_j\text{-}\mathbf{p}_j \in \mathbb{R}^{2m \times 2m}} & \boxed{\mathbf{p}_j\text{-}\psi_i \in \mathbb{R}^{2m \times 3nm}} \\ \boxed{\psi_i\text{-}\mathbf{p}_j \in \mathbb{R}^{3nm \times 2m}} & \boxed{\psi_i\text{-}\psi_i \in \mathbb{R}^{3nm \times 3nm}} \end{bmatrix},$$

where the darker background indicates denser blocks, the lighter background indicates sparser blocks and the blue outline indicates an all-zero block, arising because h_{smooth} is independent of $\{\mathbf{p}_j\}$ in equation (43).

First approximation: fixing the points' basis coefficients in the polyharmonic spline. The first main expensive operation in the problem setup that significantly slows down the solution method is the computation of $\psi_i(\Delta(\mathbf{p}_j))$. This computation is expensive because it must be repeated mn times for a single cost evaluation and because the polyharmonic spline ψ_i is applied to a moving source point $\Delta(\mathbf{p}_j)$, implying that the polyharmonic spline's basis coefficients cannot be precomputed. We propose to fix the source point to $\mathbf{R}_j = \Delta(\mathbf{p}_j^0)$, where $\{\mathbf{p}_j^0\}$ is the initial estimate so that the basis coefficients become constant. We thus define new transformations ϑ_i parameterised by m' control points $\mathbf{C}'_i$ such that $\vartheta_i(\mathbf{R}_j) \approx \psi_i(\Delta(\mathbf{p}_j))$. Therefore, the surface-based non-rigid BA formulation (39) is reformulated to:

$$\min_{\{\mathbf{p}_j\},\mathbf{C}'_i} h(\{\mathbf{p}_j\}, \mathbf{C}_{\vartheta,i}), \tag{46}$$

and $\psi_i(\Delta(\mathbf{p}_j))$ in equations (41) and (42) are replaced by $\vartheta_i(\mathbf{R}_j)$. The new transformations are initialised such that $\vartheta_i(\mathbf{R}_j) = \psi_i(\Delta(\mathbf{p}_j^0))$. Throughout the optimisation, $\{\mathbf{R}_j\}$ remains unmodified, since it is now dissociated from $\Delta(\mathbf{p}_j)$. However, $\Delta(\mathbf{p}_j)$ is still used to determine the geodesic distances and the reprojection error. The proposed approximation has two beneficial side-effects: it allows one to control the number of control points m', which we reduce compared to the number of correspondences m, and it sparsifies the Jacobian and Hessian matrices, as it zeroes the iso-$\mathbf{p}_j$ block.

Second approximation: using the Euclidean approximation of geodesic distances. The second main expensive operation in the problem setup is, for the general ToTem shapes, the computation of the geodesic distances by numerical methods. While stable, this is costly and prevents the analytic computation of the Jacobian matrix' entries. We thus propose to approximate the geodesic distance by the Euclidean distance, simply using $\mathfrak{g}(\mathbf{p}_j, \mathbf{p}_q) = \|\Delta(\mathbf{p}_j) - \Delta(\mathbf{p}_q)\|$ for general ToTem shapes.

Sparsity-aware solution of the normal equations. With the proposed approximations, the new sparsity structure of the Jacobian and Hessian matrices becomes:

$$\mathbf{J}_\vartheta = \begin{bmatrix} \boxed{\text{rep-}\mathbf{p}_j \in \mathbb{R}^{N_1 \times 2m}} & \boxed{\text{rep-}\vartheta_i \in \mathbb{R}^{N_1 \times 3nm'}} \\ \boxed{\text{iso-}\mathbf{p}_j \in \mathbb{R}^{N_2 \times 2m}} & \boxed{\text{iso-}\vartheta_i \in \mathbb{R}^{N_2 \times 3nm'}} \\ \boxed{\text{smooth-}\mathbf{p}_j \in \mathbb{R}^{N_3 \times 2m}} & \boxed{\text{smooth-}\vartheta_i \in \mathbb{R}^{N_3 \times 3nm'}} \end{bmatrix}, \mathbf{H}_\vartheta = \begin{bmatrix} \boxed{\mathbf{p}_j\text{-}\mathbf{p}_j \in \mathbb{R}^{2m \times 2m}} & \boxed{\mathbf{p}_j\text{-}\vartheta_i \in \mathbb{R}^{2m \times 3nm'}} \\ \boxed{\vartheta_i\text{-}\mathbf{p}_j \in \mathbb{R}^{3nm' \times 2m}} & \boxed{\vartheta_i\text{-}\vartheta_i \in \mathbb{R}^{3nm' \times 3nm'}} \end{bmatrix},$$

The dense iso-$\mathbf{p}_j$ is zeroed, owing to the use of the new transformations $\{\vartheta_i\}$, independent of the parameterisation $\{\mathbf{p}_j\}$. Although this does not completely nullify any of the four blocks of the Hessian matrix, it significantly reduces the computation time. The efficient solution method is obtained by means of the Schur complement [Triggs et al., 1999], observing that the ϑ_i-ϑ_i block is block-diagonal. By naming its individual diagonal blocks as ϑ_1-ϑ_1 to ϑ_n-ϑ_n, the structure is refined as:

$$\mathbf{H}_\vartheta = \begin{bmatrix} \boxed{\mathbf{p}_j\text{-}\mathbf{p}_j \in \mathbb{R}^{2m\times 2m}} & \boxed{\mathbf{p}_j\text{-}\vartheta_1 \in \mathbb{R}^{2m\times 3m'}} & \cdots & \boxed{\mathbf{p}_j\text{-}\vartheta_n \in \mathbb{R}^{2m\times 3m'}} \\ \boxed{\vartheta_1\text{-}\mathbf{p}_j \in \mathbb{R}^{3m'\times 2m}} & \boxed{\vartheta_1\text{-}\vartheta_1 \in \mathbb{R}^{3m'\times 3m'}} & & \\ \vdots & & \ddots & \\ \boxed{\vartheta_n\text{-}\mathbf{p}_j \in \mathbb{R}^{3m'\times 2m}} & & & \boxed{\vartheta_n\text{-}\vartheta_n \in \mathbb{R}^{3m'\times 3m'}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{\vartheta,1,1} & \mathbf{H}_{\vartheta,1,2} \\ \mathbf{H}_{\vartheta,1,2}^\top & \mathbf{H}_{\vartheta,2,2} \end{bmatrix}.$$

As the ϑ_1-ϑ_1 to ϑ_n-ϑ_n blocks may be ill-conditioned owing to the parameterisation freedom, we use the generalised Schur complement [Carlson et al., 1974] based on with pseudo-inverses, leading to:

$$(\mathbf{H}_\vartheta)^{-1} = \begin{bmatrix} \mathbf{H}_{\vartheta,2,2} & \mathbf{H}_{\vartheta,1,2} \\ \mathbf{H}_{\vartheta,1,2}^\top & \mathbf{H}_{\vartheta,1,1} \end{bmatrix}^{-1} =$$

$$\begin{pmatrix} \mathbf{H}^\dagger_{\vartheta,2,2} + \mathbf{H}^\dagger_{\vartheta,2,2}\mathbf{H}_{\vartheta,1,2}(\mathbf{H}_{\vartheta,1,1} - \mathbf{H}^\top_{\vartheta,1,2}\mathbf{H}^\dagger_{\vartheta,2,2}\mathbf{H}_{\vartheta,1,2})^{-1}\mathbf{H}^\top_{\vartheta,1,2}\mathbf{H}^\dagger_{\vartheta,2,2} & -\mathbf{H}^\dagger_{\vartheta,2,2}\mathbf{H}_{\vartheta,1,2}(\mathbf{H}_{\vartheta,1,1} - \mathbf{H}^\top_{\vartheta,1,2}\mathbf{H}^\dagger_{\vartheta,2,2}\mathbf{H}_{\vartheta,1,2})^{-1} \\ -(\mathbf{H}_{\vartheta,1,1} - \mathbf{H}^\top_{\vartheta,1,2}\mathbf{H}^\dagger_{\vartheta,2,2}\mathbf{H}_{\vartheta,1,2})^{-1}\mathbf{H}^\top_{\vartheta,1,2}\mathbf{H}^\dagger_{\vartheta,2,2} & (\mathbf{H}_{\vartheta,1,1} - \mathbf{H}^\top_{\vartheta,1,2}\mathbf{H}^\dagger_{\vartheta,2,2}\mathbf{H}_{\vartheta,1,2})^{-1} \end{pmatrix}, \tag{47}$$

where $\mathbf{H}^\dagger_{\vartheta,2,2}$ is computed efficiently by pseudo-inverting each of its diagonal blocks; the leading cost is thus the inversion of n matrices of size $3m' \times 3m'$.

8 EXPERIMENTAL RESULTS

We present the results of the proposed NRS*f*M approach for object-wise reconstruction of deformable objects with ToTem. We evaluate the different steps of the proposed pipeline individually and the complete proposed pipeline. First, we evaluate the point-based NRS*f*M results obtained using the method from section 4, which is the first step of the pipeline. Second, we evaluate the similarity-based alignment results obtained using the methods from section 5, which is the second step of the pipeline. Third, we evaluate the independent parameterisation results obtained using the methods from section 6.1, which is the initial part of the third step of the pipeline. Fourth, we perform an ablation study to evaluate the importance of BA from section 7, which is the fourth step of the pipeline. Fifth, we evaluate the complete pipeline thoroughly.

8.1 Point-based NRSfM

We validate the proposed point-based NRSfM method from section 4 by comparing it to representative methods from the literature. The proposed point-based NRSfM method is denoted uNRSfM-x, where x refers to the initialisation method. We use the following monikers for the baseline and initialisation methods (hence substituting x): MDH-SOCP for [Chhatkuli et al., 2017], MLH-SDP for [Ji et al., 2017], IsoH for [Chhatkuli et al., 2014], IsoSfM-InfP and IsoSfM-G for the infinitesimal planarity and general methods of [Parashar et al., 2017], PriorFree-K for [Dai et al., 2014], LowRank-K for [Gotardo and Martinez, 2011] and Learning-K-rIK and Learning-K-aIK for the rotation invariant kernel and affine SfM kernel methods from [Hamsici et al., 2012].

Benchmark real datasets. We test uNRSfM on standard benchmark datasets: the Hulk and White Cartoon T-shirt (WCT) datasets from [Chhatkuli et al., 2014], Kinect Paper (KP) dataset from [Varol et al., 2009], and the cushion dataset from [Chhatkuli et al., 2016]. We compare all three datasets with all available baseline methods. We present two versions of uNRSfM: one initialised from MDH-SOCP giving uNRSfM-MDH-SOCP and one initialised from MLH-SDP giving uNRSfM-MLH-SDP. The results are summarised in table 3, where we report the accuracy in terms of two metrics, Root Mean Square Error (RµSE) and Mean Euclidean Error (µEC), with all values in mm. All four methods, namely MDH-SOCP, MLH-SDP, uNRSfM-MDH-SOCP and uNRSfM-MLH-SDP, use the notion of NNG to decide nearest neighbours, as described in section 4; therefore the number of neighbours n_K is an important parameter for the result. We repeat the experiments for all n_K up to about 40 and report the best accuracy; the n_K corresponding to the best accuracy is mentioned (in brackets) throughout table 3.

We observe that uNRSfM-MDH-SOCP and uNRSfM-MLH-SDP clearly dominate the results for the Hulk and WCT datasets. For the Hulk dataset, uNRSfM-MDH-SOCP and uNRSfM-MLH-SDP beat the best-performing baseline method, MLH-SDP, by 1.5% and 0.69%. For the WCT dataset, uNRSfM-MLH-SDP surpasses the best-performing baseline method, MLH-SDP, by 13.49% and 14.14% in terms of RµSE and µEC respectively. For the KP dataset, uNRSfM-MDH-SOCP lags behind MLH-SDP by 1.23% in terms of RµSE, but leads by 1.04% in terms of µEC. The cushion dataset has few views, with just four images. The reconstruction accuracy is generally lower than for the other three datasets. IsoSfM-G, PriorFree-K, and LowRank-K fail on this dataset. uNRSfM-MLH-SDP is the best-performing method, leading MLH-SDP by a relatively small 1.38% accuracy in terms of RµSE, but 5.07% in terms of µEC.

Simulated data. We additionally validate the effect of uNRSfM using an isometric data simulator based on the developable surface generator from [Perriollat and Bartoli, 2013]. We generate 12 images with 42 keypoints on a 7×6 grid on a developable surface with random rulings and deformation, subject these surfaces to random roto-translation in the camera view and project them to the image plane with random intrinsics. Additionally, we add

		uNRS*f*M-MDH-SOCP	uNRS*f*M-MLH-SDP	MDH-SOCP	MLH-SDP	IsoH	IsoS*f*M-InfP	IsoS*f*M-G	PriorFree-K	LowRank-K	Learning-K-*r*IK	Learning-K-*a*IK
Hulk	RμSE ↓	**1.861** (10)	1.876 (10)	2.374 (11)	1.889 (11)	17.279	11.100	11.252	16.587	34.608	27.272	19.336
	μEC ↓	1.434	**1.365**	1.676	1.479	15.441	9.644	9.813	14.995	32.179	25.297	18.328
WCT	RμSE↓	7.72 (18)	**5.04** (16)	9.64 (18)	5.72 (17)	26.400	21.454	21.541	25.70	106.89	74.41	66.28
	μEC ↓	6.28	**3.96**	7.78	4.52	23.3	18.69	18.78	22.91	95.24	69.58	62.78
KP	RμSE↓	3.756 (21)	3.758 (21)	4.578 (23)	**3.710** (21)	14.275	11.305	11.343	13.40	30.612	12.480	13.257
	μEC ↓	**3.349**	3.367	4.067	3.384	12.514	10.129	10.149	12.125	28.406	11.291	11.872
Cushion	RμSE↓	24.398 (35)	**22.299** (35)	25.669 (41)	22.607 (40)	26.517	36.023	x	x	x	30.255	30.790
	μEC ↓	15.886	**13.806**	17.172	14.507	20.346	29.062	x	x	x	23.529	24.14

Table 3: Results on the Hulk, WCT, KP, and cushion datasets, compared across the baseline methods. Darker cells indicate better accuracy with the best accuracy in a row marked in bold. All values are in mm. For uNRS*f*M-MDH-SOCP, uNRS*f*M-MLH-SDP, MDH-SOCP, and MLH-SDP, the number in brackets indicates the optimal number of neighbours, cells with 'x' indicate failure to produce results.

a point-wise random translation, added to each 3D keypoint on the simulated surface, with a standard deviation σ. Therefore, with increasing σ, the generated surface shows increasingly random deviation from isometry. We increase σ in 20 steps from 0 to 0.02 Arbitrary Units (*au*), where the largest diagonal of the generated surface is 0.5 *au*, and repeat the experiment at each noise step 60 times. For comparison, we chose MDH-SOCP and MLH-SDP as baseline methods, since they represent the two best methods from the baseline on the benchmark datasets in table 3. The results are shown in figure 4. We observe that uNRS*f*M-MLH-SDP is consistently the best-performing method. Interestingly, uNRS*f*M-MDH-SOCP and MLH-SDP produce similar results but the improvement of uNRS*f*M-MDH-SOCP over MDH-SOCP is clear and significant.

8.2 Similarity-based Alignment

We validate the proposed similarity-based alignment methods from section 5 by comparing them to the existing shape alignment method of [Torr and Zisserman, 2000]. We randomly generate between 10 and 200 3D points on a plane, a cylinder, and a sphere. The plane has a largest diagonal of 4 *au*, the cylinder has a random radius lower than 1 *au* and the sphere is of unit radius. To these 3D points, we add a random noise of standard deviation 0.1 *au* for the plane and 1 *au* for the cylinder and sphere, since a large noise on planar points creates a random blob point-cloud for which

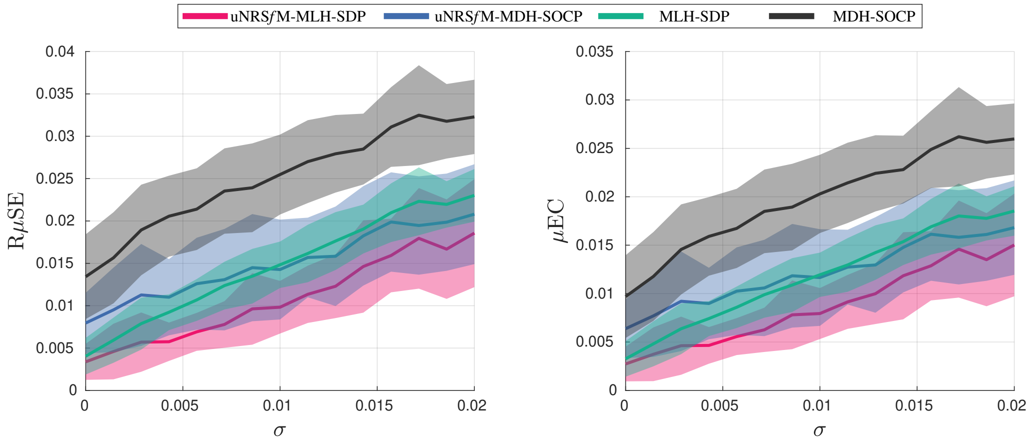


Figure 4: Comparison of uNRS*f*M-MLH-SDP and uNRS*f*M-MDH-SOCP with MLH-SDP and MDH-SOCP on synthetic isometric data, where σ denotes the measure of deviation from isometry. All values are in *au*.

plane fitting holds no physical meaning. We randomly roto-translate the noisy data into random locations in $\mathbb{R}^3$.

Finally, to evaluate the accuracy, we inverse transform the noisy point-clouds to the origin using the estimated f_i and compute an alignment error. This alignment error metric has three variants. For the plane, the alignment error metric is the Mean Point-to-Plane Error (μP$_2$P) distance between the re-aligned point-cloud and the groundtruth. For the cylinder, we define some random test points along the axis of the noisy, cylindrical point-cloud. We use f_i^{-1} to bring these test points on the axis to align to the groundtruth. We eliminate the translation along $y = 0$ line by centring this line along the Y-axis. There cannot be any reflection or rotation by π, since the roto-translations we used to perturb the simulated data are small. Finally, we define the alignment error metric as the RμSE between these aligned points on the axis and the groundtruth. For the sphere, the alignment error metric is the RμSE between the re-aligned sphere with f_i^{-1} and the groundtruth centroid.

We repeat the experiments 10^4 times for each algebraic shape. The results are shown in figure 5. The proposed methods are ahead of [Torr and Zisserman, 2000] for all three algebraic shapes. The error is approximately doubled for [Torr and Zisserman, 2000] compared to the proposed methods for the plane and cylinder, and has a factor of about eight for the sphere.

8.3 Independent Shape Parameterisation

We validate the proposed independent shape parameterisation methods from section 6.1, highlighting the benefits of using the ToTem for surface reconstruction as opposed to generic surface reconstruction methods. We name the proposed methods ToTemFit followed by 'P', 'C', or 'S' for the three algebraic shapes. We do not evaluate the method for the general ToTem shapes, as it is formed of standard methods depending on the context of the application. We proceed by generating an aligned point-cloud corresponding to $\{f_i^{-1}(\mathbf{Q}_{i,j})\}$. We randomly draw points on a plane, cylinder, and sphere and split the points into training and testing batches. We add random noise with standard deviation σ, fit the ToTem to the training point-cloud and reconstruct the surface. We then evaluate the reconstruction accuracy on the testing point-cloud, using μP$_2$P as metric. As a baseline, we use standard surface reconstruction techniques: the Ball-Pivot algorithm (BP) [Bernardini et al., 1999] and Poisson Surface Reconstruction (PSR) [Kazhdan et al., 2006]. The results are shown in figure 6a and sample qualitative results are shown in figures 6b, 6c, and 6d. For clean

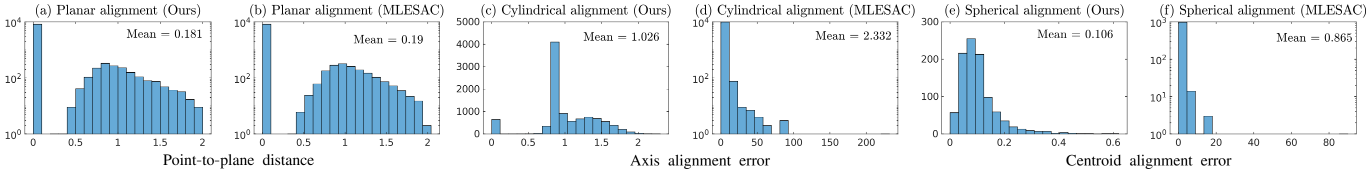


Figure 5: Comparison of the proposed similarity alignment methods with MLESAC [Torr and Zisserman, 2000]. The x-axis of the histograms are: (a), (b) the μP$_2$P between re-aligned point-cloud and the groundtruth plane, (c), (d) alignment error defined on the axis of the cylinder, and (e), (f) RμSE between the centroid of the aligned cylinder and the groundtruth. All the values are in *au*.

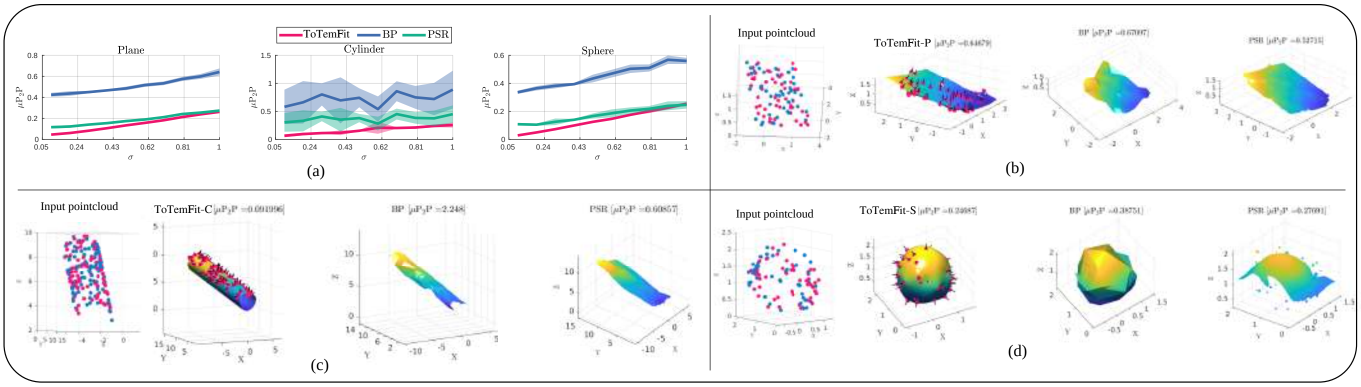


Figure 6: Comparison of the proposed parameterisation methods ToTemFit-P, ToTemFit-C, and ToTemFit-S with generic surface reconstruction methods BP and PSR. (a) quantitative results, all the values in *au*. (b), (c), and (d) sample qualitative results, with the training point-clouds in blue and the testing point-clouds in red. The normals are shown for visualisation on the results of ToTemFit-P, ToTemFit-C, and ToTemFit-S.

data with very low σ, both ToTemFit-P and ToTemFit-S have a consistent advantage over PSR while BP remains a distant third. As $\sigma \to 1$, the advantage is lost because the point-cloud becomes too noisy to distinctly benefit from the ToTem, therefore ToTemFit devolves to accuracies comparable with generic surface fitting through PSR. For ToTemFit-C however, the advantage over PSR and BP is distinctive, even for high σ.

We show a second set of validation, this time using ToTemFit-G, to show that the resemblance of the ToTem to the object's shape does not affect the reconstruction accuracy significantly in the visible parts of the object surrounding the observed keypoints. We use a real dataset from [Casillas-Perez et al., 2019], which is a balloon being squeezed across 8 images with 29 sparse keypoints correspondences; we term it the Squeezed Balloon (SB) dataset, whose sample input images are shown in figures 17 and 18. The general mesh used for this reconstruction is a triangulated sphere differing from the algebraic spherical ToTem. We deform it using free-form deformations over ten steps into a dumbbell-shaped object by randomly squeezing the equator of the sphere, as shown in figure 7(b). Therefore, the object is spherical while this set of general ToTems starts from a perfect sphere and gradually drifts away from the true shape. Upon reconstructing the shape with ToTemFit-G starting from uNRSfM-MDH-SOCP and uNRSfM-MLH-SDP, we observe that the drift away from sphericity does not significantly affect the RµSE evaluated on the visible surface of the balloon, see figure 7(a). The variance of accuracy due to the change of ToTem shape is lower than $\pm 2.5\ mm$. The minor differences in accuracy observable in figure 7a as one sweeps across the ten deformed configurations of this general ToTem stem both from the difference in the rigid alignment of the deformed shapes and the effect of the ToTem itself. To further highlight the resilience of accuracy to the ToTem shape, we perform another repeated randomised test by bootstrapping ToTemFit-G from the groundtruth 3D keypoints with additive random noise from a uniform distribution $U(-10\sqrt{3}, 10\sqrt{3})\ mm$. Each experiment is repeated over ten times. The final results, overlaid on figure 7(a), show an almost horizontal mean curve and very similar standard deviation across the ToTem shapes. Qualitative results for a sample image frame of SB are given in the last row of figure 7(b).

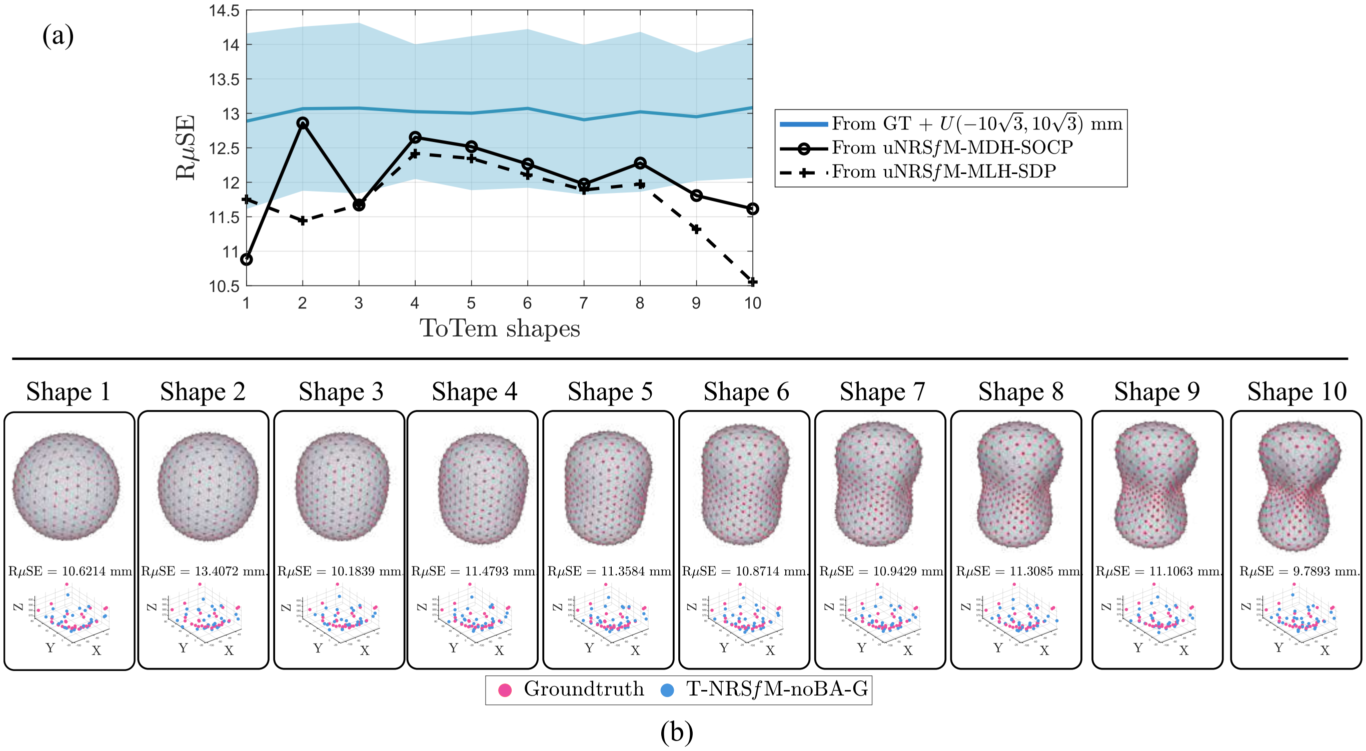


Figure 7: Results with ToTemFit-G, starting from a triangulated mesh model resembling the spherical ToTem and being gradually deformed by squeezing the centre into a dumbbell-shaped mesh. All values are in mm; the mesh deformation is done over ten steps, Shapes 1 through 10, as shown in the first row of (b). The tests were done on a spherical object, a balloon being squeezed in the SB dataset from [Casillas-Perez et al., 2019]. (a) We initialise ToTemFit-G from uNRSfM-MDH-SOCP and uNRSfM-MLH-SDP and report the reconstruction accuracy of the visible points. We also show randomised tests obtained by adding a uniform noise to ground-truth and initialising ToTemFit-G from these noisy 3D points. The bottom row of (b) shows the reconstructed and visible keypoints on the surface of the reconstructed ToTem w.r.t. the groundtruth for the sample frame 3 of SB for the results from uNRSfM-MLH-SDP.

8.4 Bundle Adjustment Ablation

We validate the importance of the BA method from section 7 by an ablation study. We measure the 3D reconstruction accuracy as RμSE, the reprojection error, and the reconstructed to groundtruth model distance μP_2P. We use four synthetic object mesh models, representing a deforming plane, a flower vase, a human face, and a liver to demonstrate the planar, cylindrical, spherical, and general shapes respectively, as shown in figure 8. Each model is synthetically deformed into 11 distinct configurations using the method of [Joshi et al., 2007]. These deformed configurations are split into training and testing regions. Importantly, the testing region involves the entire surface, including parts completely invisible in the training region. We randomly sample 6 point-clouds with 100 points each from the training set and perform reconstruction, following the pipeline without and with BA, termed ToTem NRSfM no BA (T-NRSfM-noBA) and ToTem NRSfM (T-NRSfM), respectively. We repeat the experiment 500 times for each ToTem and evaluate RμSE and reprojection error on the training samples and μP_2P on the testing samples. The quantitative results are shown in figure 9. We observe a reduction of μP_2P from T-NRSfM-noBA to T-NRSfM, showing an improvement in accuracy brought by BA, for the planar and cylindrical shapes. In contrast, μP_2P remains practically unchanged for the spherical shape and slightly degrades for the general shape. For the algebraic shapes, this result is not surprising, as the deviation of the simulated plane and deforming flower vase from the planar and cylindrical ToTem are mild, while this is not the case for the deforming human face, the sphere to face transformation is significantly stronger. For the general shape, the effect of the slight degrading in μP_2P can be qualitatively verified in figure 13. The 3D RμSE follows a similar pattern as μP_2P. Importantly, the BA improves the reprojection accuracy significantly for all ToTems. We observe that μP_2P remains significantly better without and with BA than the baseline method PSR in all cases. Sample qualitative results for the 11 deformed configurations are shown in figures 10, 11, 12, and 13 for the planar, cylindrical, spherical and general shapes respectively.

8.5 ToTem NRS*f*M on Real Data

Planar topology. Our first batch of results on real data are obtained from the previously used benchmark datasets of Hulk, KP, and WCT using the planar ToTem. In the absence of fully dense groundtruth, we use the distance of the groundtruth points from the dense surface reconstruction with T-NRS*f*M-P as an indirect indicator of the accuracy of the entire surface reconstruction, and qualitative visualisation, as shown in figure 14.

Non-planar topologies. We now focus on objects with a non-planar topology. We define a Dense Mean Point-to-Plane Error (d-μP_2P) metric for evaluating the accuracy of our dense surface reconstruction on objects with self-occlusion, which are all the non-planar objects in our experiments. As baseline methods, we consider the NRS*f*M methods with PSR as the state-of-the-art for dense reconstruction. We then measure the μP_2P distance between the mesh obtained from PSR and the groundtruth sparse 3D points, re-aligned with each other. For T-NRS*f*M, we already have the densely sampled points from $\{\psi_i\}$, we apply PSR on these points and proceed exactly in the same manner as for the baseline methods. Importantly, we compare against all baseline methods, the ones that have been left out are the ones that failed to produce any result.

The next two reconstruction results are from the dataset proposed in [Casillas-Perez et al., 2019]. The first dataset is reconstructed with the cylindrical ToTem, as it shows a legging being stretched across 8 images with 71 keypoints correspondences; we term it the Stretched Leggings (SL) dataset. The data clearly violates isometry, therefore forms an important validation of the efficacy of our proposed ToTem NRS*f*M on data not following isometry. We show some sample reconstruction results for two frames in figure 16 while a qualitative comparison across all the baseline methods for these two frames is given in figure 18(a). Importantly, figure 20(a) shows the quantitative results across all frames, where the proposed T-NRS*f*M-C outperforms all baseline methods in the highly stretched configuration, thanks to the cylindrical ToTem. In terms of d-μP_2P, the closest baseline method is PriorFree-K, but T-NRS*f*M-C is ahead by 8.22% of accuracy.

The second dataset is SB. This dataset also clearly violates isometry. We show sample reconstruction results for two frames in figure 17 while a qualitative comparison across all the baseline methods for these two frames is given in figure 18(b). Importantly, figure 20(b) shows the quantitative results across all frames, where T-NRS*f*M-S outperforms the baseline methods on all frames thanks to the spherical ToTem to recover the shape sphericity, something which has been clearly missed by all baseline methods. T-NRS*f*M-S beats the closest baseline method, MLH-SDP, by 97.43% accuracy in terms of d-μP$_2$P, showing the efficacy of spherical ToTem in this case.

Our validation for general ToTem shapes is done on the cushion dataset from [Chhatkuli et al., 2016]. This dataset has four images with 80 correspondences. We utilise a coarse triangulated mesh as ToTem, obtained by triangulating and downsampling a mesh from the groundtruth of the first image frame. The ToTem thus obtained has just 138 vertices. Given the shape of the cushion, the rigid alignment problem, as discussed in section 5.5, is solved by utilising standard rigid ICP. We show the qualitative output of T-NRS*f*M-G in figure 19. The mesh model allows us to capture the curvature of the cushion much better than possible with any of the existing methods, as verified quantitatively in figure 20(c). The closest baseline method, in terms of d-μP$_2$P, is MLH-SDP, but T-NRS*f*M-G is ahead by 39.55%, a significant lead.

Endomapper dataset. As a validation of T-NRS*f*M on a challenging use-case, we use the Endomapper dataset from [Azagra et al., 2022]. This dataset consists of real colonoscopic image sequences with camera calibration. However, this dataset does not have feature correspondences, so we pick a sample subset also used in [Sengupta and Bartoli, 2021] and add correspondences to 10 images, ending up with 84 keypoints tracked across these images. The dataset does not have groundtruth and we can thus only report qualitative results. We first run T-NRS*f*M-P, whose results are shown in figure 15. We observe that the reconstructed structure is overly flat and not appropriate for any further usage. We then run T-NRS*f*M-C. As the colon segment visible in this image sequence ends up in a constricted region with restricted visible depth, it breaks assumption 3. In other words, the observed ‘cylinder’ is wider than its depth and we thus chose the right axis in the cylinder fitting step. We observe that the reconstructed structure forms a qualitatively appropriate approximation of the true shape and could be used in downstream applications such as navigation.

9 CONCLUSION

We have presented ToTem NRS*f*M, an NRS*f*M method that exploits a topological template. As a result, it can reconstruct the surface, as opposed to point clouds in classical NRS*f*M, of a deforming object, for its observed and invisible parts. This forms the first NRS*f*M method to exploit a topological prior, forming an important intermediate between S*f*T methods using a strong prior model and NRS*f*M methods using primarily a deformation prior. The topological prior is available for a wide variety of real-life objects, whose topology is known but whose exact shape, required for the S*f*T template, is unknown. Our experimental results show the strength of our method in reconstructing challenging shapes, outperforming existing methods in accuracy and in the ability to approximate the invisible object parts.

We plan to explore three prominent directions for future work. The first direction is to generalise the notion of topological template, which we simply defined as a shape which shares its genus with the observed object. The genus thus acts as a shape invariant. We plan to define a series of templates by using different invariants to unify the existing approaches and discover new ones. In particular, S*f*T's invariant is the geodesic length and classical rigid pose's invariant is the Euclidean length. The second direction is to refine the use of MDH in resolving isometric convex-concave ambiguities. In the proposed method, this is done by a weight applied to the MDH term, encompassing the depth of all points. We plan to use a per-point weight, which would activate depth maximisation as and when needed only to resolve the ambiguities. The third direction is to extend the method to multi-object reconstruction. The lead idea is to perform the first step, point-based NRS*f*M, and search for shapes using a predefined collection of ToTems, then running the remaining steps of the pipeline for each identified cluster. Lastly, the proposed concept of ToTem NRS*f*M could be implemented in different ways, obtained by designing alternative solution pipelines.

Acknowledgements

This work received funding from the FET-Open grant 863146 EndoMapper, under the European Union's Horizon 2020 research and innovation program. We thank David Casillas-Perez for sharing and explaining the dataset from [Casillas-Perez et al., 2019].

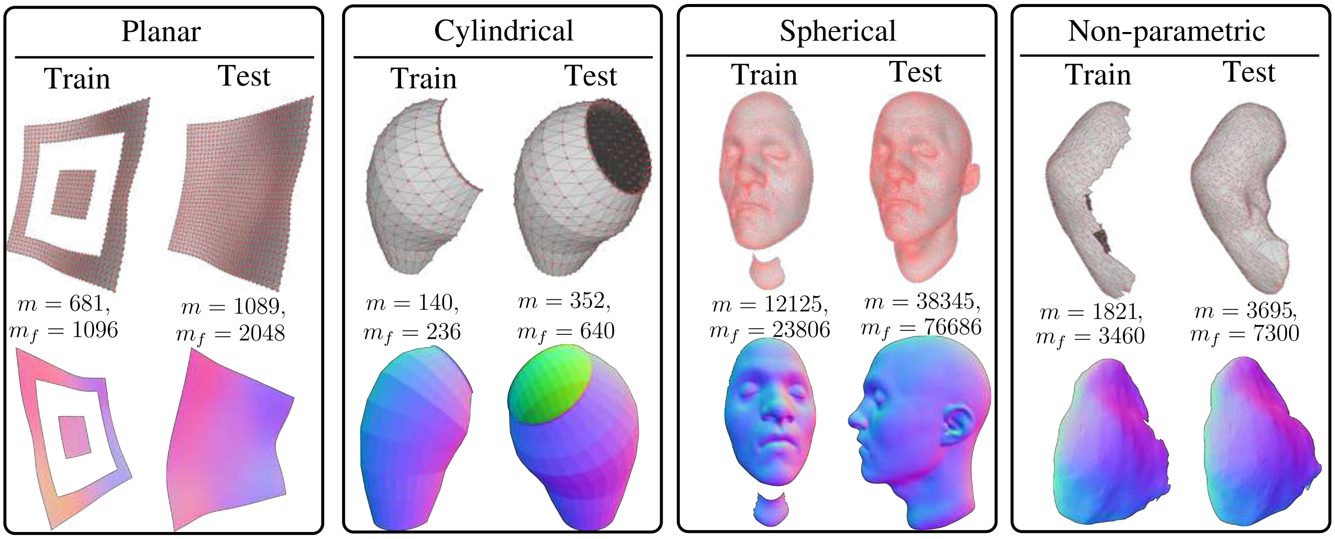
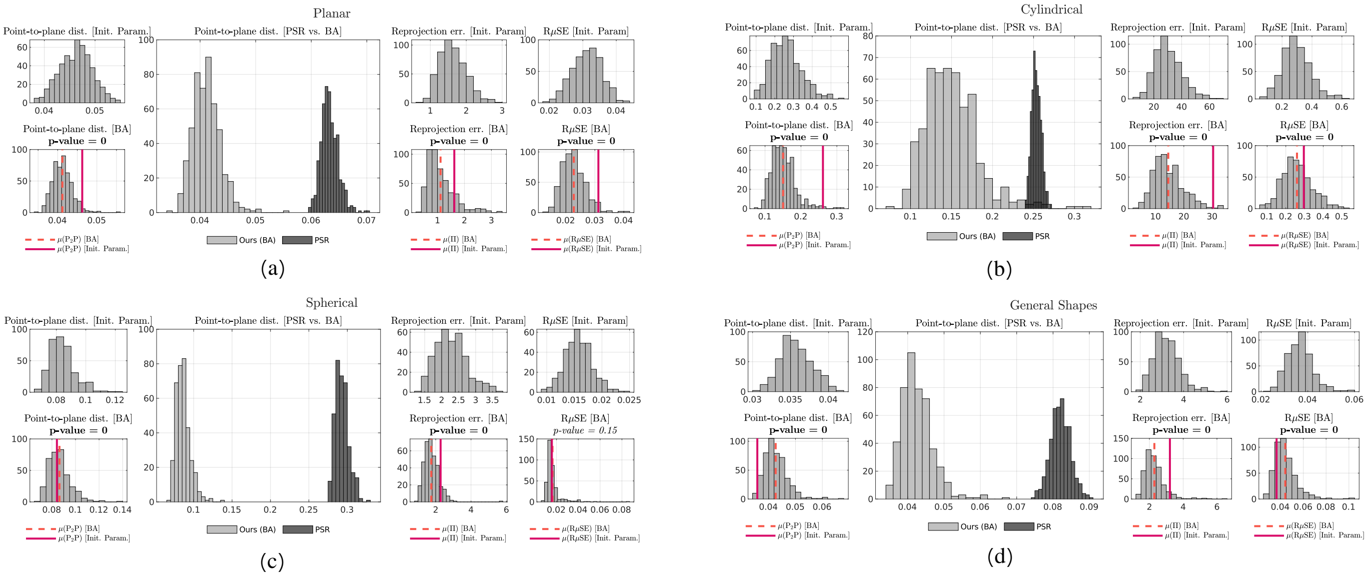


Figure 8: Experimental setup for the bundle adjustment ablation study. Each input mesh is split into a training and a testing region. The testing surface clearly includes parts of the mesh occluded in the training surface. The reconstruction process uses only the training samples. We give the number of keypoints m and the number of faces m_f for each mesh.



(a) (b) (c) (d)

Figure 9: Quantitative results for the BA ablation study, shown for the planar, cylindrical, spherical, and general shapes in (a), (b), (c), and (d) respectively, with all values in *au*. In the results following T-NRSfM, the mean of the histograms from T-NRSfM-noBA and T-NRSfM are displayed in bold and dashed red lines respectively, while the p-value for the null hypothesis between the T-NRSfM-noBA and T-NRSfM distributions is shown between the two rows of histograms.

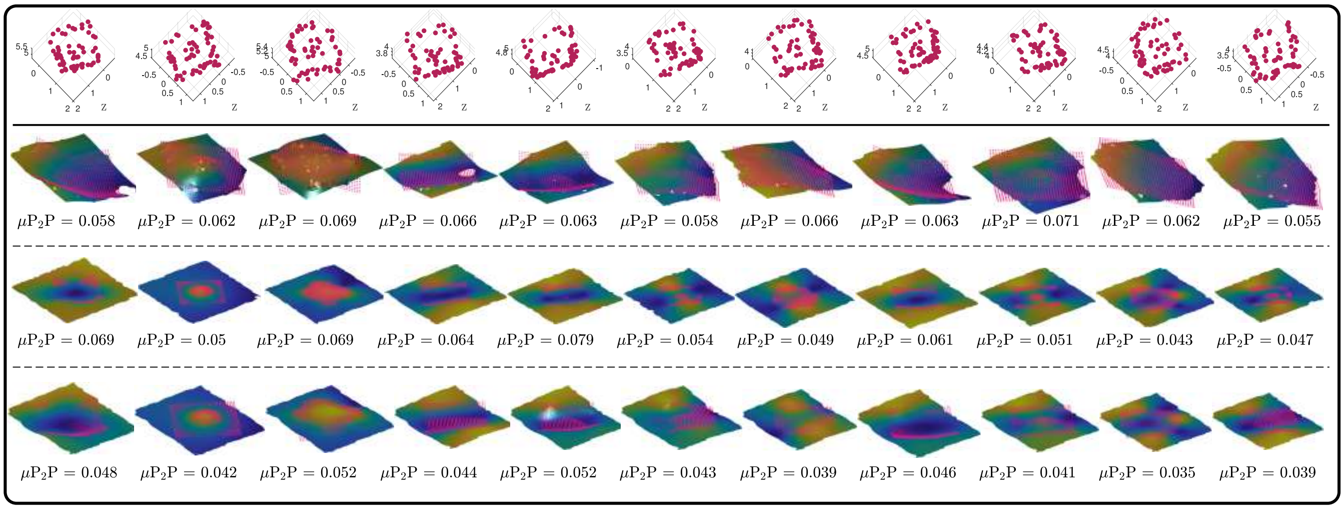

Input
PSR
μP$_2$P = 0.058 μP$_2$P = 0.062 μP$_2$P = 0.069 μP$_2$P = 0.066 μP$_2$P = 0.063 μP$_2$P = 0.058 μP$_2$P = 0.066 μP$_2$P = 0.063 μP$_2$P = 0.071 μP$_2$P = 0.062 μP$_2$P = 0.055
T-NRSfM-noBA-P
μP$_2$P = 0.069 μP$_2$P = 0.05 μP$_2$P = 0.069 μP$_2$P = 0.064 μP$_2$P = 0.079 μP$_2$P = 0.054 μP$_2$P = 0.049 μP$_2$P = 0.061 μP$_2$P = 0.051 μP$_2$P = 0.043 μP$_2$P = 0.047
T-NRSfM-P
μP$_2$P = 0.048 μP$_2$P = 0.042 μP$_2$P = 0.052 μP$_2$P = 0.044 μP$_2$P = 0.052 μP$_2$P = 0.043 μP$_2$P = 0.039 μP$_2$P = 0.046 μP$_2$P = 0.041 μP$_2$P = 0.035 μP$_2$P = 0.039

Figure 10: Results for the planar ToTem. The first row shows the sampled keypoints, the second row shows the reconstruction using PSR, the third row shows our reconstruction with T-NRS*f*M-noBA-P, and the fourth row shows our reconstruction with T-NRS*f*M-P. The µP$_2$P evaluated on the testing regions are given below each reconstructed surface, with all values in *au*.

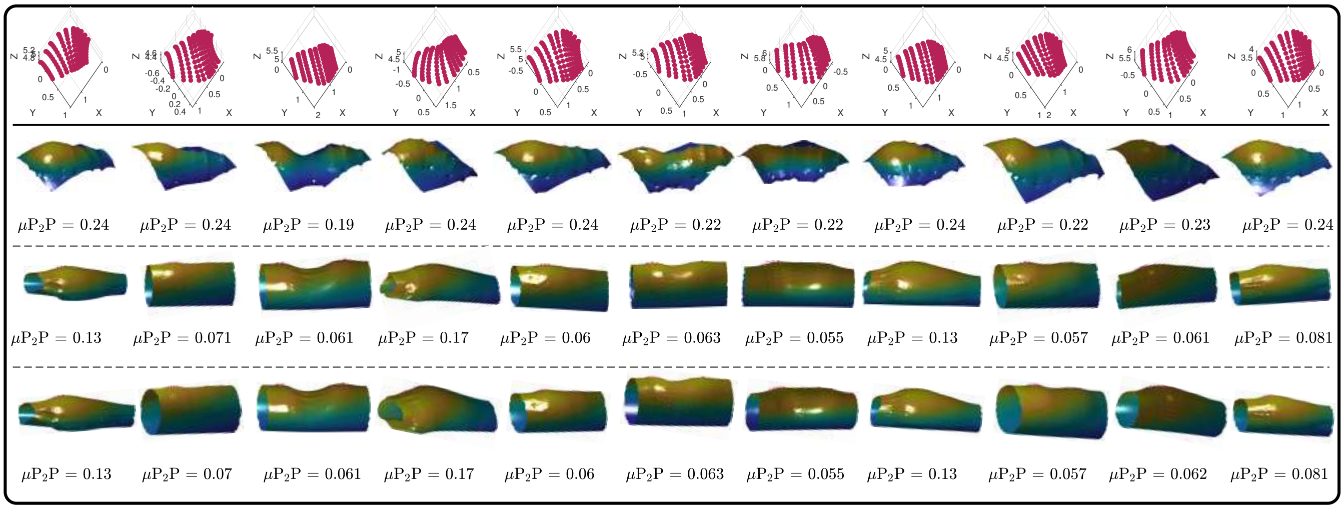

Input
PSR
μP$_2$P = 0.24 μP$_2$P = 0.24 μP$_2$P = 0.19 μP$_2$P = 0.24 μP$_2$P = 0.24 μP$_2$P = 0.22 μP$_2$P = 0.22 μP$_2$P = 0.24 μP$_2$P = 0.22 μP$_2$P = 0.23 μP$_2$P = 0.24
T-NRSfM-noBA-C
μP$_2$P = 0.13 μP$_2$P = 0.071 μP$_2$P = 0.061 μP$_2$P = 0.17 μP$_2$P = 0.06 μP$_2$P = 0.063 μP$_2$P = 0.055 μP$_2$P = 0.13 μP$_2$P = 0.057 μP$_2$P = 0.061 μP$_2$P = 0.081
T-NRSfM-C
μP$_2$P = 0.13 μP$_2$P = 0.07 μP$_2$P = 0.061 μP$_2$P = 0.17 μP$_2$P = 0.06 μP$_2$P = 0.063 μP$_2$P = 0.055 μP$_2$P = 0.13 μP$_2$P = 0.057 μP$_2$P = 0.062 μP$_2$P = 0.081

Figure 11: Results for the cylindrical ToTem. The first row shows the sampled keypoints, the second row shows the reconstruction using PSR, the third row shows our reconstruction with T-NRS*f*M-noBA-C, and the fourth row shows our reconstruction with T-NRS*f*M-C. The µP$_2$P evaluated on the testing regions are given below each reconstructed surface, with all values in *au*.

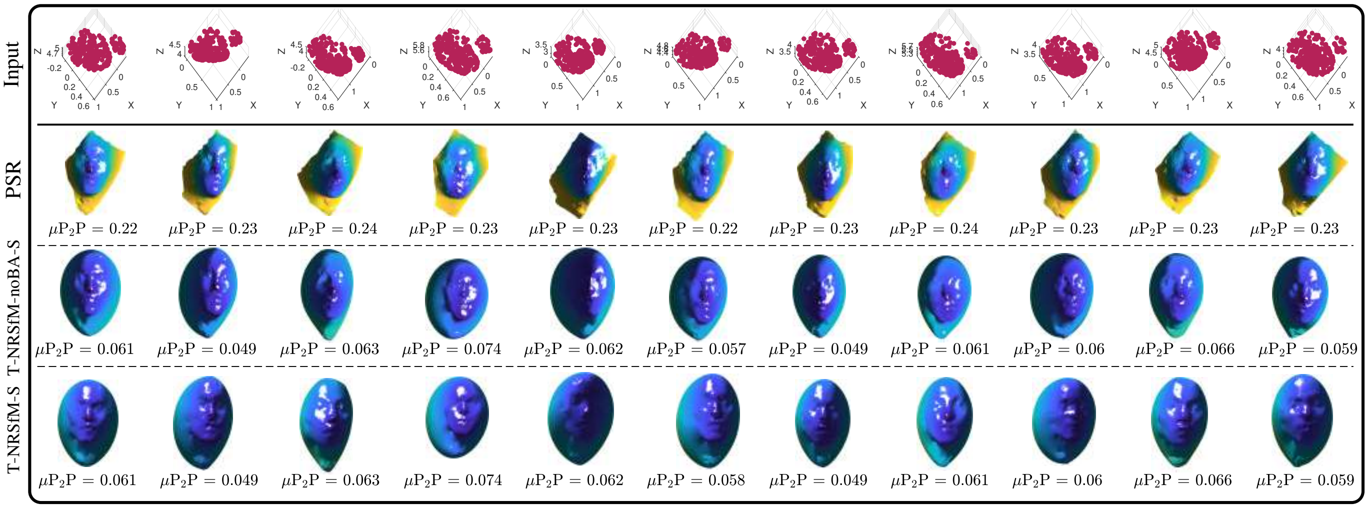

Input
PSR
$\mu P_2P = 0.22$ $\mu P_2P = 0.23$ $\mu P_2P = 0.24$ $\mu P_2P = 0.23$ $\mu P_2P = 0.23$ $\mu P_2P = 0.22$ $\mu P_2P = 0.23$ $\mu P_2P = 0.24$ $\mu P_2P = 0.23$ $\mu P_2P = 0.23$ $\mu P_2P = 0.23$
T-NRSfM-noBA-S
$\mu P_2P = 0.061$ $\mu P_2P = 0.049$ $\mu P_2P = 0.063$ $\mu P_2P = 0.074$ $\mu P_2P = 0.062$ $\mu P_2P = 0.057$ $\mu P_2P = 0.049$ $\mu P_2P = 0.061$ $\mu P_2P = 0.06$ $\mu P_2P = 0.066$ $\mu P_2P = 0.059$
T-NRSfM-S
$\mu P_2P = 0.061$ $\mu P_2P = 0.049$ $\mu P_2P = 0.063$ $\mu P_2P = 0.074$ $\mu P_2P = 0.062$ $\mu P_2P = 0.058$ $\mu P_2P = 0.049$ $\mu P_2P = 0.061$ $\mu P_2P = 0.06$ $\mu P_2P = 0.066$ $\mu P_2P = 0.059$

Figure 12: Results for the spherical ToTem. The first row shows the sampled keypoints, the second row shows the reconstruction using PSR, the third row shows our reconstruction with T-NRS*f*M-noBA-S, and the fourth row shows our reconstruction with T-NRS*f*M-S. The μP_2P evaluated on the testing regions are given below each reconstructed surface, with all values in *au*.

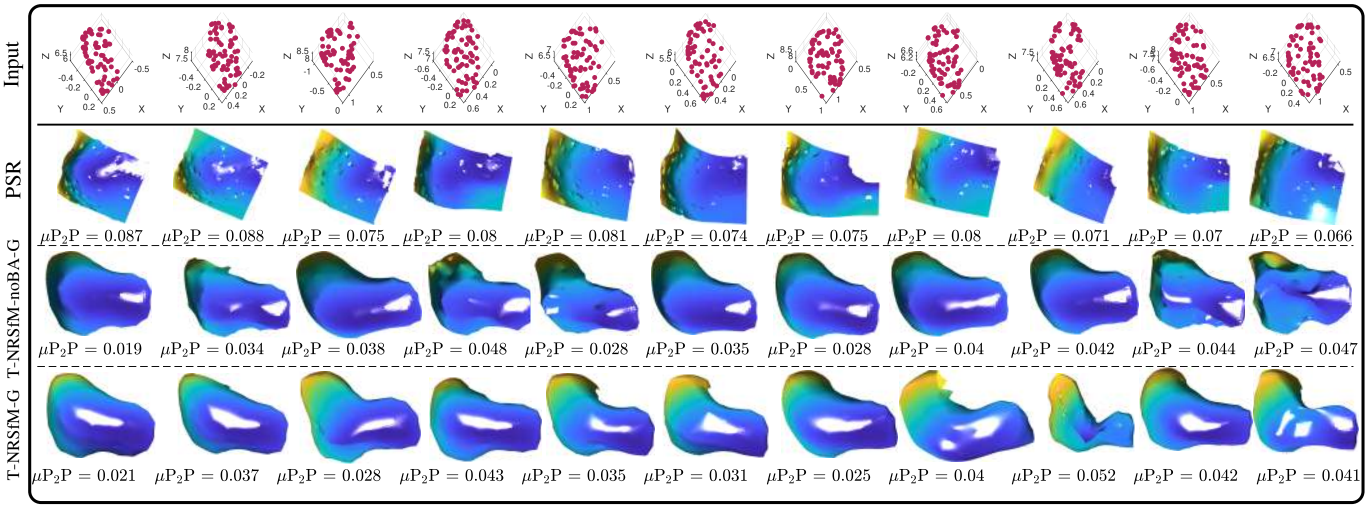

Input
PSR
$\mu P_2P = 0.087$ $\mu P_2P = 0.088$ $\mu P_2P = 0.075$ $\mu P_2P = 0.08$ $\mu P_2P = 0.081$ $\mu P_2P = 0.074$ $\mu P_2P = 0.075$ $\mu P_2P = 0.08$ $\mu P_2P = 0.071$ $\mu P_2P = 0.07$ $\mu P_2P = 0.066$
T-NRSfM-noBA-G
$\mu P_2P = 0.019$ $\mu P_2P = 0.034$ $\mu P_2P = 0.038$ $\mu P_2P = 0.048$ $\mu P_2P = 0.028$ $\mu P_2P = 0.035$ $\mu P_2P = 0.028$ $\mu P_2P = 0.04$ $\mu P_2P = 0.042$ $\mu P_2P = 0.044$ $\mu P_2P = 0.047$
T-NRSfM-G
$\mu P_2P = 0.021$ $\mu P_2P = 0.037$ $\mu P_2P = 0.028$ $\mu P_2P = 0.043$ $\mu P_2P = 0.035$ $\mu P_2P = 0.031$ $\mu P_2P = 0.025$ $\mu P_2P = 0.04$ $\mu P_2P = 0.052$ $\mu P_2P = 0.042$ $\mu P_2P = 0.041$

Figure 13: Results for the general shapes ToTem. The first row shows the sampled keypoints, the second row shows the reconstruction using PSR, the third row shows our reconstruction with T-NRS*f*M-noBA-G, and the fourth row shows our reconstruction with T-NRS*f*M-G. The μP_2P evaluated on the testing regions are given below each reconstructed surface, with all values in *au*.

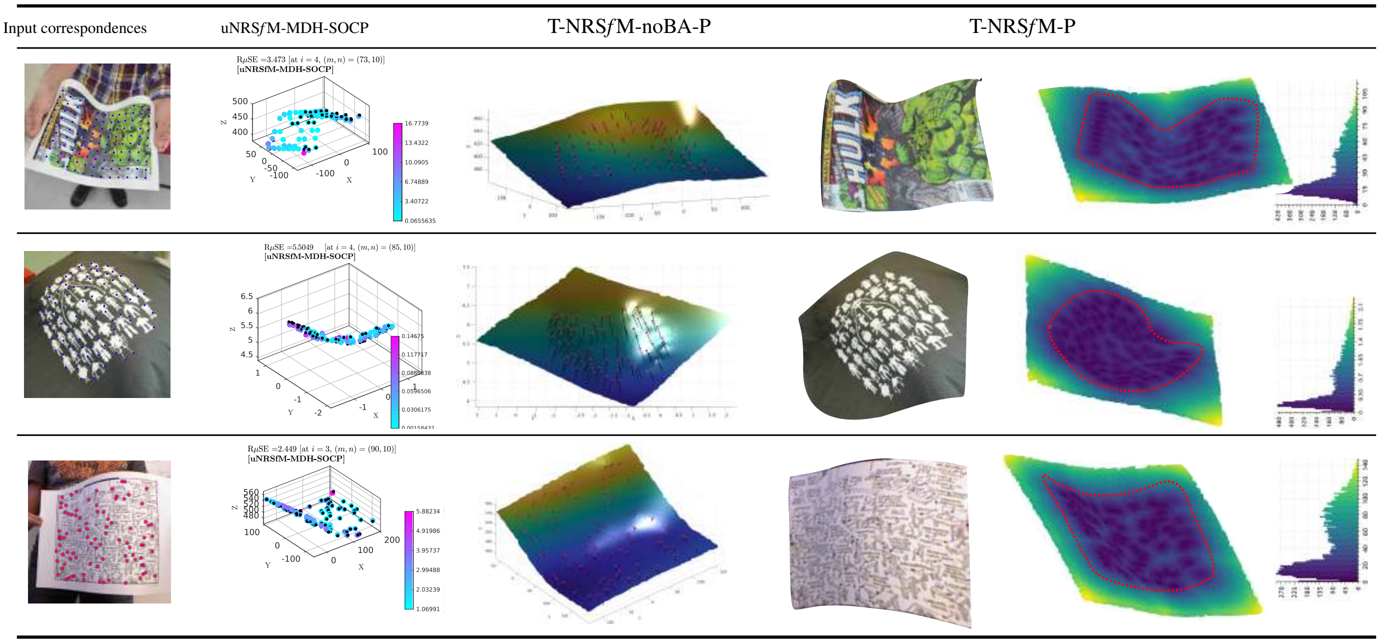


Figure 14: Results from reconstruction of the benchmark datasets Hulk, WCT and KP with planar ToTem. The first column shows the input correspondences overlaid on the images, the second column shows our 3D reconstruction with uNRS*f*M, groundtruth keypoints are in black while our reconstruction output is colour coded according to Euclidean distance from groundtruth (all values in mm), the third column shows dense reconstruction result from T-NRS*f*M-noBA-P, the surface normals evaluated on the keypoints are shown, the fourth column shows the dense reconstruction result with T-NRS*f*M-P with the surface texture overlaid on the reconstructed surface, the fifth column shows the colour coded distance between the groundtruth of the sparse keypoints and the densely sampled, reconstructed surface from T-NRS*f*M-P, the approximate region of interest on the surface around the matched keypoints is highlighted in red.

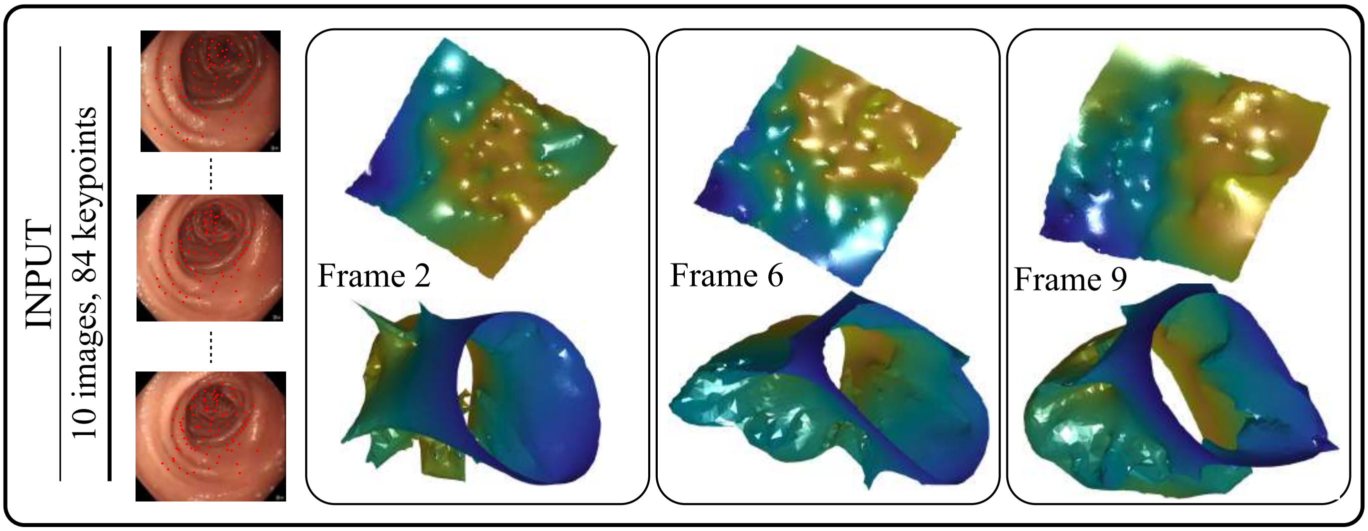


Figure 15: Qualitative reconstruction results of the colonic surface from the colonoscopy dataset [Azagra et al., 2022] using T-NRS*f*M-P (top row) and T-NRS*f*M-C (bottom row). The T-NRS*f*M-C results, although clearly a mere approximation of the true shape, form a significantly more usable output than the T-NRS*f*M-P results.

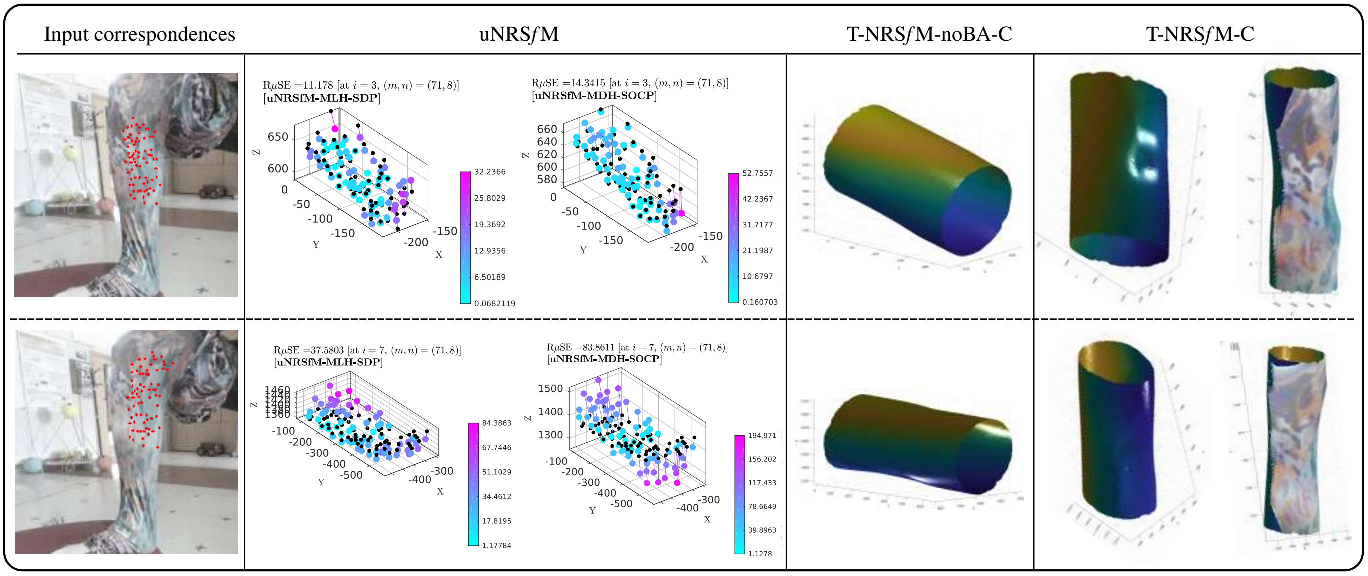


Figure 16: We present two representative samples of our reconstruction for the SL dataset, the first column shows the input correspondences, the second and third columns show uNRSfM-MLH-SDP and uNRSfM-MDH-SOCP respectively, the fourth column shows T-NRSfM-noBA-C and the fifth column show T-NRSfM-C, all values in mm.

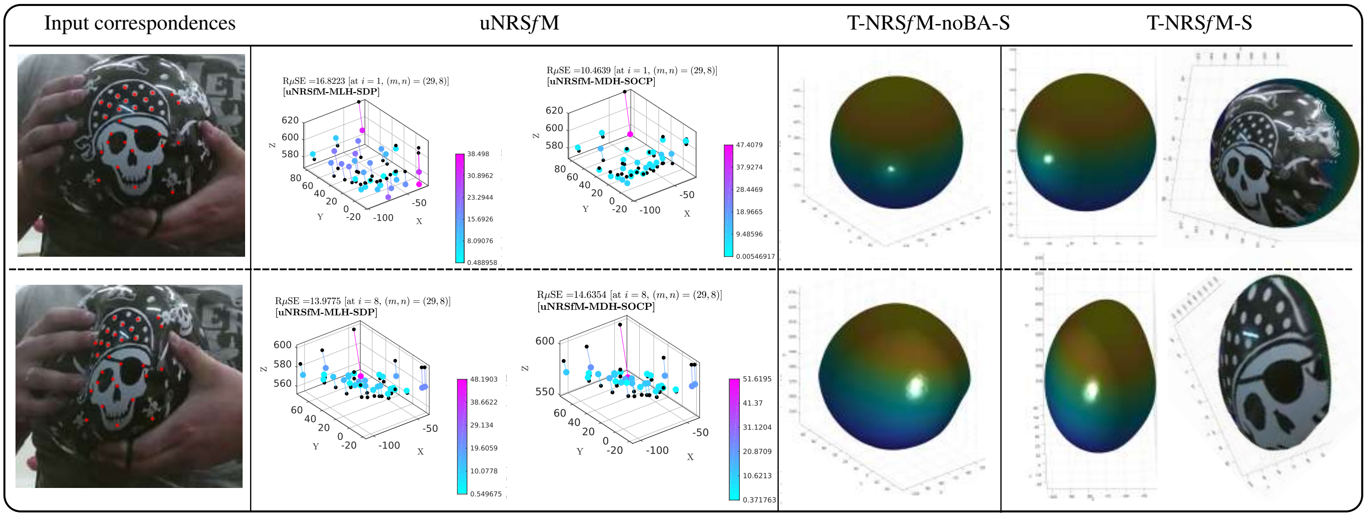


Figure 17: We present two representative samples of our reconstruction for the SB dataset; the first column shows the input correspondences, the second and third column show uNRSfM-MLH-SDP and uNRSfM-MDH-SOCP respectively, the fourth column shows T-NRSfM-noBA-S and the fifth column shows T-NRSfM-S, all values in mm. While the second frame ($n = 8$) slightly over-estimates the curvature of the ball, this still remains a better reconstruction than all baseline methods, as verifiable in figure 18.

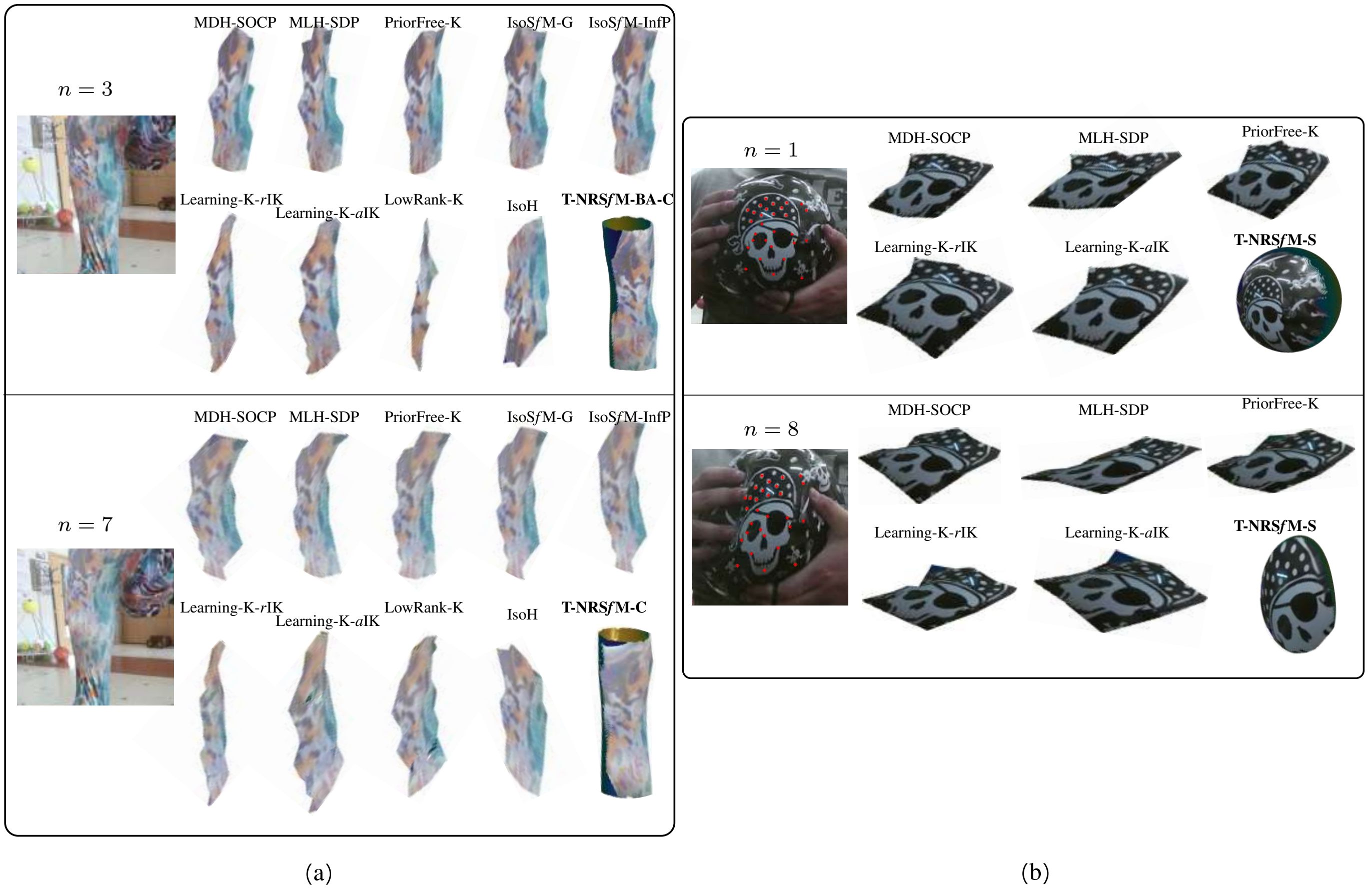


Figure 18: Qualitative comparison of the results from T-NRSfM-C and T-NRSfM-S with surface reconstructed by using PSR on baseline methods for the two frames, each of (a) SL, and (b) SB, datasets shown in figures 16 and 17. Being challenging datasets with strong stretching/shearing, all baseline methods produce surfaces that are either nearly planar or curved in a completely wrong direction.

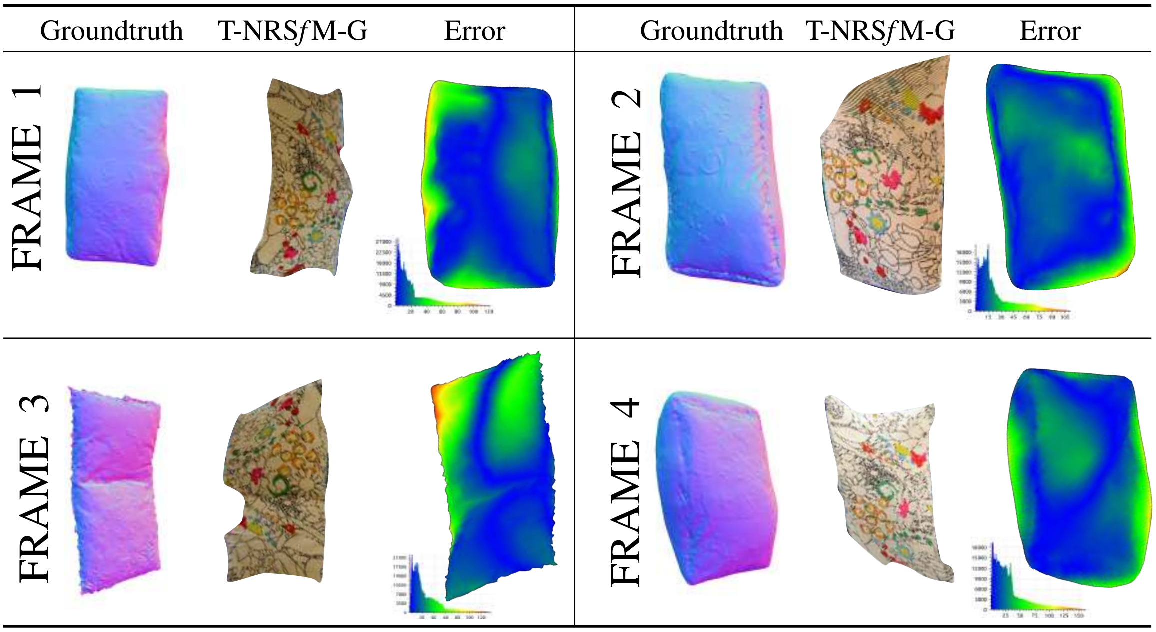
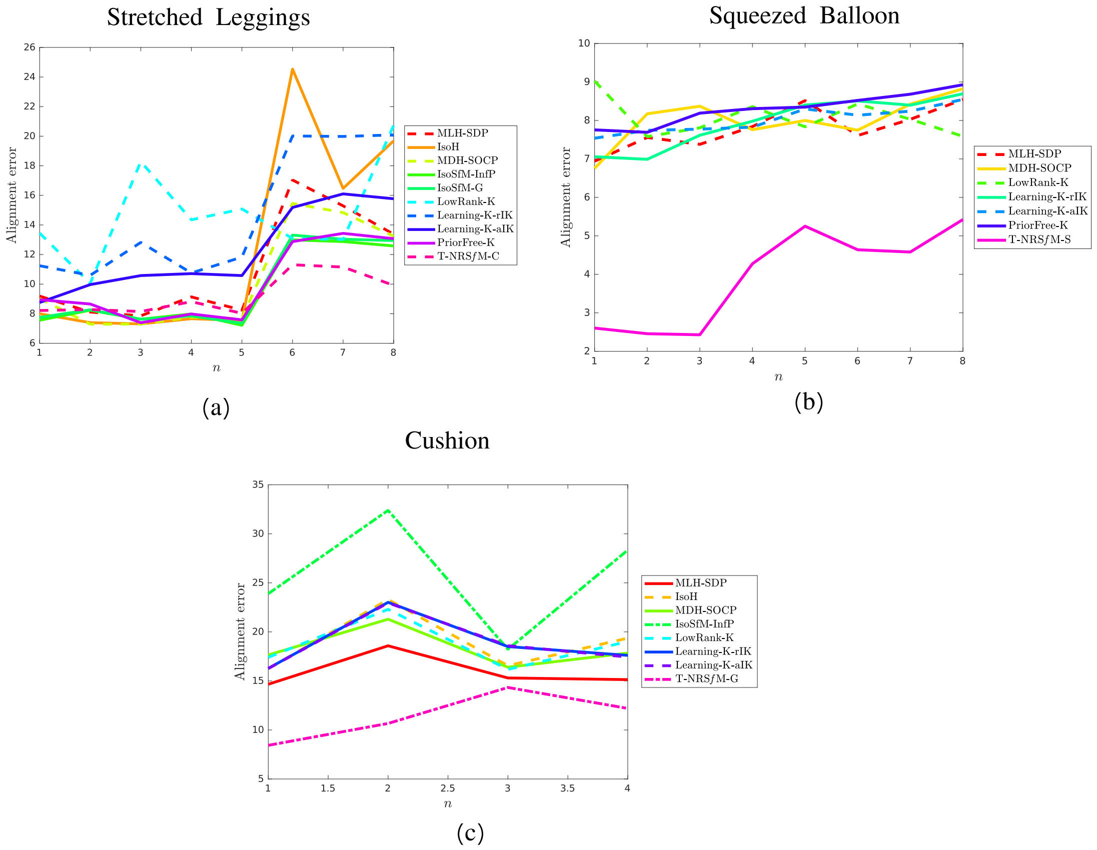


Figure 19: The four image frames of the cushion dataset, visualised together. The first image for each frame is the groundtruth shape, the second image is the texture-mapped result of T-NRSfM-G, the third image is the alignment error, obtained by densely sampling the groudtruth and the reconstructed surface from T-NRSfM-G and computing the distance between the nearest points on the aligned point-clouds, all values in mm.



(a)

(b)

(c)

Figure 20: We show the d-μP$_2$P metric across all the frames in the three non-planar datasets: (a) SL reconstruction accuracy, for all the baseline methods, deteriorates as the stretching increases in frames 6, 7, and 8, but with cylindrical ToTem, T-NRSfM-C manages to mitigate the accuracy degradation, (b) for the SB dataset, T-NRSfM-S, due to its spherical prior, manages to outperform other methods by a significant margin, and (c) in the cushion dataset, T-NRSfM-G outperforms the compared methods in estimating the curvature of the cushion. All values in mm.

APPENDICES

A Geometry of the Topological Templates

We describe some basic geometric properties of the parametric shapes in the following paragraphs.

A.1 Planar

We denote the 2D template with $\mathbf{p} = (x, y) \in \mathbb{R}^2$. Therefore the map $\Delta : \mathbb{R}^2 \mapsto \mathbb{R}^3$ is trivial and given in table 2. The next warp ψ is defined as:

$$\psi_i(\mathbf{P}) = \begin{pmatrix} X & Y & Z & 1 \end{pmatrix} \underset{4\times 3}{\mathbf{a}} + \underset{1\times l}{\rho(\mathbf{P}, \mathbf{D})} \underset{l\times 3}{\mathbf{w}}, \tag{48}$$

and the combined warp is:

$$\varphi_i(x, y) = \psi_i \circ \Delta = \Delta(x, y)\mathbf{a} + \rho\Big(\Delta(x, y), \mathbf{D}\Big)\mathbf{w}. \tag{49}$$

Therefore, the derivatives of the warps are:

$$\begin{aligned} \varphi_x(x, y) &= \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \mathbf{a} + \frac{\partial}{\partial x}\Big(\rho\big(\Delta(x, y), \mathbf{D}\big)\Big)\mathbf{w} \\ \varphi_y(x, y) &= \begin{pmatrix} 0 & 1 & 0 & 1 \end{pmatrix} \mathbf{a} + \frac{\partial}{\partial y}\Big(\rho\big(\Delta(x, y), \mathbf{D}\big)\Big)\mathbf{w}, \end{aligned} \tag{50}$$

where for the l-th element of ρ we have:

$$\frac{\partial}{\partial x}\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big) = \frac{X - D_l^X}{\rho^l\big(\Delta(x, y), \mathbf{D}\big)} \qquad \text{and} \qquad \frac{\partial}{\partial y}\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big) = \frac{Y - D_l^Y}{\rho^l\big(\Delta(x, y), \mathbf{D}\big)}. \tag{51}$$

Proceeding similarly, the second derivatives of the warps are given by:

$$\begin{gathered} \varphi_{xx}^l(x, y) = \frac{\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big)^2 - \big(X - D_l^X\big)^2}{\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big)^2}\mathbf{w} \qquad \varphi_{yy}^l(x, y) = \frac{\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big)^2 - \big(Y - D_l^Y\big)^2}{\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big)^2}\mathbf{w} \\ \varphi_{xy}^l(x, y) = \varphi_{yx}^l(x, y) = \frac{\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big)^2 - \Big(\big(X - D_l^X\big) + \big(Y - D_l^Y\big)\Big)^2}{\Big(\rho^l\big(\Delta(x, y), \mathbf{D}\big)\Big)^2}\mathbf{w}. \end{gathered} \tag{52}$$

A.2 Cylindrical

Following the definition of Δ from table 2, we obtain ψ as:

$$\psi_i(\mathbf{P}) = \begin{pmatrix} X & Y & Z & 1 \end{pmatrix} \underset{4\times3}{\mathbf{a}} + \underset{1\times l}{\rho(\mathbf{P},\mathbf{D})} \underset{l\times3}{\mathbf{w}}. \tag{53}$$

$$\varphi(r,\theta) = \begin{pmatrix} \sin\theta & r & \cos\theta & 1 \end{pmatrix} \mathbf{a} + \rho\Big(\Delta(r,\theta),\mathbf{D}\Big)\mathbf{w} \tag{54}$$

For the case of cylindrical surfaces, we elaborate the computation of the tangent vectors and surface normal of ψ_i with some verbosity. Assuming:

$$\check{\mathbf{a}} = \begin{pmatrix} \alpha_1 & \alpha_5 & \alpha_9 & \alpha_{13} \\ \alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \\ \alpha_3 & \alpha_7 & \alpha_{11} & \alpha_{15} \\ \alpha_4 & \alpha_8 & \alpha_{12} & \alpha_{16} \end{pmatrix} \quad \text{and} \quad \check{\mathbf{w}} = \begin{pmatrix} \beta_{11} & \dots & \beta_{14} \\ \vdots & \ddots & \vdots \\ \beta_{l1} & \dots & \beta_{l4} \end{pmatrix}, \tag{55}$$

we can compute:

$$\tilde{\varphi}_r = \frac{\partial\varphi}{\partial r} = \begin{pmatrix} \alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \end{pmatrix} + \begin{pmatrix} \frac{r-D^{\mathrm{Y}}{}_1}{\rho^1(\Delta(r,\theta),\mathbf{D})} & \dots & \frac{r-D^{\mathrm{Y}}{}_l}{\rho^l(\Delta(r,\theta),\mathbf{D})} \end{pmatrix} \check{\mathbf{w}}, \tag{56}$$

and ignoring α_{14} from equation (56), we obtain:

$$\varphi_r^\top = \begin{pmatrix} \alpha_2 & \alpha_6 & \alpha_{10} \end{pmatrix} + \begin{pmatrix} \lambda_1^r & \lambda_2^r & \lambda_3^r \end{pmatrix}, \tag{57}$$

where the assumption that the fourth column of $\check{\mathbf{a}}$ and $\check{\mathbf{w}}$ are all zeros gives us $\mathbf{a}$ and $\mathbf{w}$ respectively. Similarly:

$$\varphi_\theta^\top = \begin{pmatrix} (\alpha_1\cos\theta - \alpha_3\sin\theta) & (\alpha_5\cos\theta - \alpha_7\sin\theta) & (\alpha_9\cos\theta - \alpha_{11}\sin\theta) \end{pmatrix} + \begin{pmatrix} \lambda_1^\theta & \lambda_2^\theta & \lambda_3^\theta \end{pmatrix}, \tag{58}$$

where:

$$\begin{pmatrix} \lambda_1^\theta & \lambda_2^\theta & \lambda_3^\theta \end{pmatrix} = \begin{pmatrix} \frac{D^{\mathrm{Z}}{}_1\sin\theta - D^{\mathrm{X}}{}_1\cos\theta}{\rho^1(\Delta(r,\theta),\mathbf{D})} & \dots & \frac{D^{\mathrm{Z}}{}_l\sin\theta - D^{\mathrm{X}}{}_l\cos\theta}{\rho^l(\Delta(r,\theta),\mathbf{D})} \end{pmatrix} \check{\mathbf{w}}, \tag{59}$$

since α_{13} and α_{15} are zeros. Given that $\mathbf{N} = \begin{pmatrix} N^X & N^Y & N^Z \end{pmatrix} = \varphi_r \times \varphi_\theta$, we have:

$$N^X = (\alpha_6 + \lambda_2^r)(\alpha_9\cos\theta - \alpha_{11}\sin\theta + \lambda_3^\theta) - (\alpha_{10} + \lambda_3^r)(\alpha_5\cos\theta - \alpha_7\sin\theta + \lambda_2^\theta) \tag{60}$$

$$N^Y = -(\alpha_2 + \lambda_1^r)(\alpha_9\cos\theta - \alpha_{11}\sin\theta + \lambda_3^\theta) + (\alpha_{10} + \lambda_3^r)(\alpha_1\cos\theta - \alpha_3\sin\theta + \lambda_1^\theta) \tag{61}$$

$$N^Z = (\alpha_2 + \lambda_1^r)(\alpha_5 \cos\theta - \alpha_7 \sin\theta + \lambda_2^\theta) - (\alpha_6 + \lambda_2^r)(\alpha_1 \cos\theta - \alpha_3 \sin\theta + \lambda_1^\theta), \tag{62}$$

where $\mathbf{N}$ is the un-normalized vector along the direction of the surface normal, evaluated at (r, θ). Consequently, the unit normal is expressed as $\hat{\mathbf{N}} = \frac{\varphi_r \times \varphi_\theta}{\|\varphi_r \times \varphi_\theta\|}$. We denote the function for unit normal computation as $\mathcal{N}_i(\cdot)$, which takes the j-th coordinate from the flattened template (r_j, θ_j) and computes the unit normal, given by $\mathcal{N}_i(r_j, \theta_j) = \begin{pmatrix} \hat{N}^X_{i,j} & \hat{N}^Y_{i,j} & \hat{N}^Z_{i,j} & 1 \end{pmatrix}$ for the i-th image frame.

A.3 Spherical

For spherical objects, beginning from the initial mapping Δ described in table 2, the maps ψ and φ are obtained following the same form as equations (1) and (54) respectively. The normals to the spherical surface, which are the equivalent of equations (56) to (62), are summarised below. The derivatives of φ are given as:

$$\tilde{\varphi}_{\theta_1} = \begin{pmatrix} \sin\theta_1 \cos\theta_2 & -\cos\theta_1 & \sin\theta_1 \sin\theta_2 \end{pmatrix} \mathbf{a} + \frac{\partial \rho\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_1} \mathbf{w} \tag{63}$$

$$\tilde{\varphi}_{\theta_2} = \begin{pmatrix} \cos\theta_1 \sin\theta_2 & 0 & -\cos\theta_1 \cos\theta_2 \end{pmatrix} \mathbf{a} + \frac{\partial \rho\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_2} \mathbf{w}, \tag{64}$$

where the l-th element of $\frac{\partial \rho\big(\Delta(r,\theta),\mathbf{D}\big)}{\partial \theta_1}$ and $\frac{\partial \rho\big(\Delta(r,\theta),\mathbf{D}\big)}{\partial \theta_2}$ are given respectively as:

$$\frac{\partial \rho^l\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_1} = \frac{\Big(\Delta(\theta_1, \theta_2) - \mathbf{D}_l\Big)^\top \Delta_{\theta_1}}{\rho^l\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)} \quad \text{and} \quad \frac{\partial \rho^l\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_2} = \frac{\Big(\Delta(\theta_1, \theta_2) - \mathbf{D}_l\Big)^\top \Delta_{\theta_2}}{\rho^l\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}, \tag{65}$$

where $\Delta_{\theta_1} = \frac{\partial \Delta}{\partial \theta_1}$ and $\Delta_{\theta_2} = \frac{\partial \Delta}{\partial \theta_2}$ following standard convention. Similar to section A.2, the unit normals can be obtained as $\hat{\mathbf{N}} = \frac{\varphi_{\theta_1} \times \varphi_{\theta_2}}{\|\varphi_{\theta_1} \times \varphi_{\theta_2}\|}$. From equation (65), the second derivatives of the warps can be computed as:

$$\begin{aligned} \tilde{\varphi}_{\theta_1\theta_1} &= \begin{pmatrix} \cos\theta_1 \cos\theta_2 & \sin\theta_1 & \cos\theta_1 \sin\theta_2 \end{pmatrix} \mathbf{a} + \frac{\partial^2 \rho\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_1^2} \mathbf{w} \\ \tilde{\varphi}_{\theta_2\theta_2} &= \begin{pmatrix} \cos\theta_1 \cos\theta_2 & 0 & \cos\theta_1 \sin\theta_2 \end{pmatrix} \mathbf{a} + \frac{\partial^2 \rho\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_2^2} \mathbf{w} \\ \tilde{\varphi}_{\theta_1\theta_2} = \tilde{\varphi}_{\theta_2\theta_1} &= \begin{pmatrix} -\sin\theta_1 \sin\theta_2 & 0 & \sin\theta_1 \cos\theta_2 \end{pmatrix} \mathbf{a} + \frac{\partial}{\partial \theta_1} \Big(\frac{\partial \rho\big(\Delta(\theta_1, \theta_2), \mathbf{D}\big)}{\partial \theta_2} \Big) \mathbf{w}, \end{aligned} \tag{66}$$

where:

$$
\begin{aligned}
\frac{\partial^2 \rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)}{\partial\theta_1^2} &= \frac{\Big(\rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)\Big)^2\Big((\Delta(\theta_1,\theta_2)-\mathbf{D}_l)^\top\Delta_{\theta_1\theta_1}+\Delta_{\theta_1}^\top\Delta_{\theta_1}\Big)-(\Delta(\theta_1,\theta_2)-\mathbf{D}_l)^\top\Delta_{\theta_1}}{\Big(\rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)\Big)^3}\\
\frac{\partial^2 \rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)}{\partial\theta_2^2} &= \frac{\Big(\rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)\Big)^2\Big((\Delta(\theta_1,\theta_2)-\mathbf{D}_l)^\top\Delta_{\theta_2\theta_2}+\Delta_{\theta_2}^\top\Delta_{\theta_2}\Big)-(\Delta(\theta_1,\theta_2)-\mathbf{D}_l)^\top\Delta_{\theta_2}}{\Big(\rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)\Big)^3}\\
&\frac{\partial}{\partial\theta_1}\Big(\frac{\partial\rho^l\big(\Delta(\theta_2),\mathbf{D}\big)}{\partial\theta_2}\Big)=\frac{\partial}{\partial\theta_2}\Big(\frac{\partial\rho^l\big(\Delta(\theta_1),\mathbf{D}\big)}{\partial\theta_2}\Big)=\\
&\frac{\Big(\rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)\Big)^2\Big((\Delta(\theta_1,\theta_2)-\mathbf{D}_l)^\top\Delta_{\theta_1\theta_2}+\Delta_{\theta_1}^\top\Delta_{\theta_2}\Big)-(\Delta(\theta_1,\theta_2)-\mathbf{D}_l)^\top(\Delta(\theta_1,\theta_2)-\mathbf{D}_l)\Delta_{\theta_1}^\top\Delta_{\theta_2}}{\Big(\rho^l\big(\Delta(\theta_1,\theta_2),\mathbf{D}\big)\Big)^3}.
\end{aligned}
\tag{67}
$$

A.4 General Shapes

We have a combination of two warps, the first one being:

$$
\Delta(x,y)=\begin{pmatrix}x & y & 0 & 1\end{pmatrix}\mathbf{a}_\Delta+\rho(\mathbf{p},\mathbf{D})\mathbf{w}_\Delta=\begin{pmatrix}X & Y & Z\end{pmatrix}=\mathbf{P}, \tag{68}
$$

where the pre-defined warp parameters $(\mathbf{a}_\Delta, \mathbf{w}_\Delta)$ are obtained using conformal flattening of some mesh using the method of [Sheffer and de Sturler, 2001; Sheffer et al., 2005]. The variables $(\mathbf{a}_\Delta, \mathbf{w}_\Delta)$ are computed offline and only once per object model. This is followed by the second warp for deformation of the canonical model, given by:

$$
\psi_i(\mathbf{P})=\begin{pmatrix}X & Y & Z & 1\end{pmatrix}\underset{4\times3}{\mathbf{a}}+\underset{1\times l}{\rho(\mathbf{P},\mathbf{D})}\underset{l\times3}{\mathbf{w}} \qquad \text{and} \qquad \varphi_i(x,y)=\psi_i\circ\Delta. \tag{69}
$$

We are interested in the derivatives of the warps:

$$
\varphi_x=\begin{pmatrix}\frac{\partial\mathbf{P}}{\partial x} & 0\end{pmatrix}+\frac{\partial\rho(\mathbf{P},\mathbf{D})}{\partial x}\mathbf{w} \qquad\qquad \varphi_y=\begin{pmatrix}\frac{\partial\mathbf{P}}{\partial y} & 0\end{pmatrix}+\frac{\partial\rho(\mathbf{P},\mathbf{D})}{\partial y}\mathbf{w}. \tag{70}
$$

Thereafter, we have:

$$
\begin{pmatrix}\frac{\partial\mathbf{P}}{\partial x}\\ \frac{\partial\mathbf{P}}{\partial y}\end{pmatrix}=\begin{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}\mathbf{a}_\Delta+\frac{\partial\rho(\mathbf{p},\mathbf{D})}{\partial x}\mathbf{w}_\Delta\\ \begin{pmatrix}0 & 1 & 0 & 0\end{pmatrix}\mathbf{a}_\Delta+\frac{\partial\rho(\mathbf{p},\mathbf{D})}{\partial y}\mathbf{w}_\Delta\end{pmatrix}=\begin{pmatrix}\frac{\partial X}{\partial x} & \frac{\partial Y}{\partial x} & \frac{\partial Z}{\partial x}\\ \frac{\partial X}{\partial y} & \frac{\partial Y}{\partial y} & \frac{\partial Z}{\partial y}\end{pmatrix}, \quad \begin{pmatrix}\frac{\partial\rho^l(\mathbf{p},\mathbf{D})}{\partial x}\\ \frac{\partial\rho^l(\mathbf{p},\mathbf{D})}{\partial x}\end{pmatrix}=\begin{pmatrix}\frac{x-D_l^X}{\rho^l(\mathbf{p},\mathbf{D})}\\ \frac{y-D_l^Y}{\rho^l(\mathbf{p},\mathbf{D})}\end{pmatrix}\forall l\in[1,n], \tag{71}
$$

and:

$$\begin{pmatrix} \frac{\partial \rho^l(\mathbf{P},\mathbf{D})}{\partial x} \\ \frac{\partial \rho^l(\mathbf{P},\mathbf{D})}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{(X-D_l^X)\frac{\partial X}{\partial x}+(Y-D_l^Y)\frac{\partial Y}{\partial x}+(Z-D_l^Z)\frac{\partial Z}{\partial x}}{\rho^l(\mathbf{P},\mathbf{D})} \\ \frac{(X-D_l^X)\frac{\partial X}{\partial y}+(Y-D_l^Y)\frac{\partial Y}{\partial y}+(Z-D_l^Z)\frac{\partial Z}{\partial y}}{\rho^l(\mathbf{P},\mathbf{D})} \end{pmatrix} \qquad \forall l \in [1, n]. \tag{72}$$

Following similar geometric properties, the next frame describes an example of the computation of length along geodesic, specifically for the spherical template:

Geodesic distances for **spherical** topology

The distance along the surface of the sphere is computed as follows. For two arbitrary, neighbouring points $\mathbf{p}_j = (\theta_{1,j}, \theta_{2,j})$ and $\mathbf{p}_q = (\theta_{1,q}, \theta_{2,q})$, we define a unit vector from $\mathbf{p}_j$ to $\mathbf{p}_q$ as $\mathbf{n}_{j,q} = \frac{\mathbf{p}_j - \mathbf{p}_q}{d_p} = \begin{pmatrix} n_{\theta_1} \\ n_{\theta_2} \end{pmatrix}$ where $d_p = \|\mathbf{p}_j - \mathbf{p}_q\|$ and describe a line connecting $\mathbf{p}_j$ and $\mathbf{p}_q$ as $\mathbf{p}(t) = \mathbf{p}_j + \mathbf{n}_{j,q}t$. Thereafter, the geodesic distance on the surface of a sphere is given as:

$$d_{j,q} = \int_0^{d_p} \sqrt{\mathcal{E}_\Delta \Big(\frac{d}{dt}(\mathbf{p}_{\theta_1}(t))\Big)^2 + 2\mathcal{F}_\Delta \frac{d}{dt}(\mathbf{p}_{\theta_1}(t))\frac{d}{dt}(\mathbf{p}_{\theta_2}(t)) + \mathcal{G}_\Delta \Big(\frac{d}{dt}(\mathbf{p}_{\theta_2}(t))\Big)^2}, \tag{73}$$

where $\mathcal{E}_\Delta$, $\mathcal{F}_\Delta$ and $\mathcal{G}_\Delta$ are obtained from the *first* fundamental form of Δ and evaluates to $\mathcal{E}_\Delta = 1$, $\mathcal{F}_\Delta = 0$ and $\mathcal{G}_\Delta = \cos^2\theta_1$ for our choice of Δ for spherical topology. This leads to the following outcome for the value of $d_{j,q}$:

$$d_{j,q} = d_p\sqrt{n_{\theta_1}^2 + \cos^2\theta_1 n_{\theta_2}^2}. \tag{74}$$

B Jacobian Matrices

We now describe the derivation of analytic Jacobian matrices for all cost functions used in sections 6.2 and 7.

B.1 Initial Parameterised Reconstruction

The minimisation of equation (33) is done analytically and the Jacobian matrix relating the change of g_{data}, g_{def0} and g_{def1} w.r.t. $\xi_I \in \mathbb{R}^{1\times 3l}$, the stacked optimisation parameters, are expressed by the terms:

$$\mathbf{J}_P = \frac{\partial g_{\text{data}}}{\partial \xi_I}, \quad \mathbf{J}_N = \frac{\partial g_{\text{def1}}}{\partial \xi_I} \quad \text{and} \quad \mathbf{J}_B = \frac{\partial g_{\text{def0}}}{\partial \xi_I}. \tag{75}$$

Expanding the Jacobian matrices for the j-th row of the matrices $\mathbf{J}_P$ and $\mathbf{J}_N$ (corresponding to the j-th row of the error vectors), we obtain the following expressions:

$$\begin{aligned}\underbrace{\mathbf{J}_{P,j}}_{1\times 3l} =& \frac{\partial}{\partial \xi_I}\Big(\big(\varphi(\mathbf{p}_j) - \mathbf{Q}_j\big)\cdot \eta\big(\varphi(\mathbf{p}_j)\big)\Big)\\ =& \frac{\partial}{\partial\xi_I}\Bigg(\overbrace{\Big(\underbrace{\begin{bmatrix}\rho\big(\Delta(r_j,\theta_j),\mathbf{D}\big) & \tilde{\mathbf{P}}_j\end{bmatrix}\mathcal{D}^{-1}\begin{bmatrix}\tilde{\mathbf{C}}\\ \mathbf{0}\end{bmatrix}}_{1\times4} - \underbrace{\tilde{\mathbf{Q}}_j}_{1\times4}\Big)}^{\boldsymbol{\mu}_j}\overbrace{\underbrace{\begin{bmatrix}\hat{N}_j^X & \hat{N}_j^Y & \hat{N}_j^Z & 1\end{bmatrix}^\top}_{4\times1}}^{\hat{\mathbf{N}}_j}\Bigg)\\ =& \frac{\partial}{\partial\xi_I}(\boldsymbol{\mu}_j\hat{\mathbf{N}}_j) = \underbrace{\frac{\partial\boldsymbol{\mu}_j}{\partial\xi_I}}_{1\times12l}\underbrace{(\hat{\mathbf{N}}_j\otimes\mathbb{1}_{3l})}_{12l\times3l} + \underbrace{\boldsymbol{\mu}_j}_{1\times4}\underbrace{\frac{\partial\hat{\mathbf{N}}_j}{\partial\xi_I}}_{4\times3l},\end{aligned} \tag{76}$$

where:

$$\begin{aligned}\underbrace{\frac{\partial\boldsymbol{\mu}_j}{\partial\xi_I}}_{1\times12l} &= \frac{\partial}{\partial\xi_I}\Big(\begin{bmatrix}\rho\big(\Delta(r_j,\theta_j),\mathbf{D}\big) & \tilde{\mathbf{P}}_j\end{bmatrix}\mathcal{D}^{-1}\begin{bmatrix}\tilde{\mathbf{C}}\\ \mathbf{0}\end{bmatrix} - \tilde{\mathbf{Q}}_j\Big)\\ &= \underbrace{\begin{bmatrix}\rho\big(\Delta(r_j,\theta_j),\mathbf{D}\big) & \tilde{\mathbf{P}}_j\end{bmatrix}}_{1\times(l+4)}\underbrace{\mathcal{D}^{-1}}_{(l+4)\times(l+4)}\underbrace{\begin{bmatrix}\mathbb{1}_l & \mathbb{0}_{l\times3l} & \mathbb{1}_l & \mathbb{0}_{l\times3l} & \mathbb{1}_l & \mathbb{0}_{l\times3l}\\ \multicolumn{6}{c}{\mathbb{0}_{4\times12l}}\end{bmatrix}}_{(l+4)\times12l},\end{aligned} \tag{77}$$

and $\frac{\partial\hat{\mathbf{N}}_j}{\partial\xi_I}$ can be obtained from:

$$\begin{aligned}\underbrace{\frac{\partial\hat{\mathbf{N}}_j}{\partial\xi_I}}_{4\times3l} =& \frac{\partial}{\partial\xi_I}\begin{bmatrix}\hat{N}_j^X & \hat{N}_j^Y & \hat{N}_j^Z & 1\end{bmatrix}^\top\\ = \frac{1}{\|\tilde{\mathbf{N}}_j\|}\begin{bmatrix}\frac{\partial N_j^X}{\partial\xi_I} & \frac{\partial N_j^Y}{\partial\xi_I} & \frac{\partial N_j^Z}{\partial\xi_I} & 0\end{bmatrix}^\top - & \frac{\begin{bmatrix}N_j^X & N_j^Y & N_j^Z & 1\end{bmatrix}^\top\Big(N_j^X\frac{\partial N_j^X}{\partial\xi_I} + N_j^Y\frac{\partial N_j^Y}{\partial\xi_I} + N_j^Z\frac{\partial N_j^Z}{\partial\xi_I}\Big)}{(\|\tilde{\mathbf{N}}_j\|)^3}.\end{aligned} \tag{78}$$

Thereafter, we are left with the terms $\frac{\partial N_j^X}{\partial\xi_I}$, $\frac{\partial N_j^Y}{\partial\xi_I}$ and $\frac{\partial N_j^Z}{\partial\xi_I}$ which are obtained from equations (56) and (58) by differentiating the expression for the un-normalised normal vector:

$$\begin{aligned}\frac{\partial}{\partial\xi_I}\Big(\begin{bmatrix}N_j^X & N_j^Y & N_j^Z\end{bmatrix}\Big) =& \frac{\partial}{\partial\xi_I}(\varphi_r\times\varphi_\theta) = \frac{\partial}{\partial\xi_I}\big([\varphi_r]_\times\varphi_\theta\big) = \frac{\partial}{\partial\xi_I}\big([\varphi_r]_\times\big)\big(\varphi_\theta\otimes\mathbb{1}_{3l}\big) + [\varphi_r]_\times\frac{\partial\varphi_\theta}{\partial\xi_I}\\ =& \begin{bmatrix}0 & -\frac{\partial\varphi_{r,3}}{\partial\xi_I} & \frac{\partial\varphi_{r,2}}{\partial\xi_I}\\ \frac{\partial\varphi_{r,3}}{\partial\xi_I} & 0 & -\frac{\partial\varphi_{r,1}}{\partial\xi_I}\\ -\frac{\partial\varphi_{r,2}}{\partial\xi_I} & \frac{\partial\varphi_{r,1}}{\partial\xi_I} & 0\end{bmatrix}\big(\varphi_\theta\otimes\mathbb{1}_{3l}\big) + [\varphi_r]_\times\begin{bmatrix}\frac{\partial\varphi_{\theta,1}}{\partial\xi_I}\\ \frac{\partial\varphi_{\theta,2}}{\partial\xi_I}\\ \frac{\partial\varphi_{\theta,3}}{\partial\xi_I}\end{bmatrix}.\end{aligned} \tag{79}$$

From equation (56), we have:

$$
\begin{aligned}
\frac{\partial \tilde{\varphi}_r}{\partial \xi_I} &= \frac{\partial}{\partial \xi_I}\left(\begin{bmatrix} \alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \end{bmatrix} \right) + \frac{\partial}{\partial \xi_I}\left(\begin{bmatrix} \frac{r - {D^{\mathrm{Y}}}_1}{\rho^1(\Delta(r,\theta),\mathbf{D})} & \dots & \frac{r - {D^{\mathrm{Y}}}_l}{\rho^l(\Delta(r,\theta),\mathbf{D})} \end{bmatrix} \bar{\varepsilon}_\lambda \tilde{\mathbf{C}} \right) \\
&= \frac{\partial}{\partial \xi_I}\left(\begin{bmatrix} \alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \end{bmatrix} \right) + \begin{bmatrix} \frac{r - {D^{\mathrm{Y}}}_1}{\rho^1(\Delta(r,\theta),\mathbf{D})} & \dots & \frac{r - {D^{\mathrm{Y}}}_l}{\rho^l(\Delta(r,\theta),\mathbf{D})} \end{bmatrix} \bar{\varepsilon}_\lambda \begin{bmatrix} \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l}. \end{bmatrix}
\end{aligned} \tag{80}
$$

Similarly:

$$
\begin{aligned}
\frac{\partial \tilde{\varphi}_\theta}{\partial \xi_I} &= \frac{\partial}{\partial \xi_I}\left(\begin{bmatrix} (\alpha_1 \cos\theta - \alpha_3 \sin\theta) \\ (\alpha_5 \cos\theta - \alpha_7 \sin\theta) \\ (\alpha_9 \cos\theta - \alpha_{11} \sin\theta) \\ (\alpha_{13} \cos\theta - \alpha_{15} \sin\theta) \end{bmatrix}^\top \right) + \begin{bmatrix} \frac{{D^{\mathrm{Z}}}_1 \sin\theta - {D^{\mathrm{X}}}_1 \cos\theta}{\rho^1(\Delta(r,\theta),\mathbf{D})} \\ \vdots \\ \frac{{D^{\mathrm{Z}}}_l \sin\theta - {D^{\mathrm{X}}}_l \cos\theta}{\rho^l(\Delta(r,\theta),\mathbf{D})} \end{bmatrix}^\top \bar{\varepsilon}_\lambda \begin{bmatrix} \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta \\ -\sin\theta \end{bmatrix} \frac{\partial}{\partial \xi_I}\left(\begin{bmatrix} \alpha_1 & \alpha_5 & \alpha_9 & \alpha_{13} \\ \alpha_3 & \alpha_7 & \alpha_{11} & \alpha_{15} \end{bmatrix} \right) + \begin{bmatrix} \frac{{D^{\mathrm{Z}}}_1 \sin\theta - {D^{\mathrm{X}}}_1 \cos\theta}{\rho^1(\Delta(r,\theta),\mathbf{D})} \\ \vdots \\ \frac{{D^{\mathrm{Z}}}_l \sin\theta - {D^{\mathrm{X}}}_l \cos\theta}{\rho^l(\Delta(r,\theta),\mathbf{D})} \end{bmatrix}^\top \bar{\varepsilon}_\lambda \begin{bmatrix} \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} \end{bmatrix}.
\end{aligned} \tag{81}
$$

Equations (80) and (81) leave us with some more unknown terms in the form of $\frac{\partial \alpha_1}{\partial \xi_I}, \dots, \frac{\partial \alpha_{15}}{\partial \xi_I}$. However, these values can be easily obtained by differentiating equation (55) as:

$$
\frac{\partial \breve{\mathbf{a}}}{\partial \xi_I} = \frac{\partial}{\partial \xi_I}\left(\begin{bmatrix} \alpha_1 & \alpha_5 & \alpha_9 & \alpha_{13} \\ \alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \\ \alpha_3 & \alpha_7 & \alpha_{11} & \alpha_{15} \\ \alpha_4 & \alpha_8 & \alpha_{12} & \alpha_{16} \end{bmatrix} \right) = \left(\tilde{\mathbf{D}} \mathbf{K}_\lambda^{-1} \tilde{\mathbf{D}}^\top \right)^{-1} \tilde{\mathbf{D}} \mathbf{K}_\lambda^{-1} \begin{bmatrix} \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} \end{bmatrix}. \tag{82}
$$

Going back to equation (75), the Jacobian matrix $\mathbf{J}_N$ can be expressed as:

$$
\underbrace{\mathbf{J}_{N,j}}_{1\times 3l} = \frac{\left(\hat{N}_j^X - \eta^X(\Delta(\mathbf{p}_j))\right)\frac{\partial \hat{N}_j^X}{\partial \xi_I} + \left(\hat{N}_j^Y - \eta^Y(\Delta(\mathbf{p}_j))\right)\frac{\partial \hat{N}_j^Y}{\partial \xi_I} + \left(\hat{N}_j^Z - \eta^Z(\Delta(\mathbf{p}_j))\right)\frac{\partial \hat{N}_j^Z}{\partial \xi_I}}{\left\|\eta\big(\varphi(\mathbf{p}_j)\big) - \eta\big(\Delta(\mathbf{p}_j)\big)\right\|}. \tag{83}
$$

The second Jacobian matrix from equation (75), $\mathbf{J}_B$, is a straightforward partial differentiation of equation (35).

B.2 Global Refinement

We now discuss the details of the Jacobian matrices involved in the global refinement process from equation (40), the Jacobian matrices are given for the original problem, not the efficient one, since the accelerated solutions are easy to derive from the full expanded ones. A single row of the first Jacobian matrix relating the change of h_{rep} to $\boldsymbol{\xi}$ is given by:

$$\underbrace{\mathbf{J}_{R,g}}_{1\times(2m+3nl)} = \frac{\partial h_{\text{rep}}}{\partial \boldsymbol{\xi}} = \frac{\partial}{\partial \boldsymbol{\xi}}\Big(\|\mathbf{q}_{i,j} - \Pi\big(\varphi_i(\mathbf{p}_j)\big)\|\Big) = \begin{bmatrix} \frac{\partial h_{\text{rep}}}{\partial \kappa} & \frac{\partial h_{\text{rep}}}{\partial \boldsymbol{\omega}_1} & \dots & \frac{\partial h_{\text{rep}}}{\partial \boldsymbol{\omega}_n} \end{bmatrix}, \tag{84}$$

where $\boldsymbol{\kappa}$ and $\boldsymbol{\omega}$ are vectorised uv-coordinates and control handles respectively. Considering the first term and the second set of terms separately, we start with the first term of equation (39):

$$\begin{aligned} \frac{\partial h_{\text{rep}}}{\partial \kappa} = \frac{\partial}{\partial \kappa}\Big(\|\mathbf{q}_{i,j} - \Pi\big(\varphi_i(\mathbf{p}_j)\big)\|\Big) = \frac{-1}{\|\mathbf{q}_{i,j} - \Pi\big(\varphi_i(\mathbf{p}_j)\big)\|}\Bigg(\bigg(\Big((\mathbf{q}_j)_x - \Pi\big(\varphi_i(\mathbf{p}_j)\big)_x\Big)\frac{\partial}{\partial \kappa}\Big(\Pi\big(\varphi_i(\mathbf{p}_j)\big)_x\Big)\bigg) \\ +\bigg(\Big((\mathbf{q}_j)_y - \Pi\big(\varphi_i(\mathbf{p}_j)\big)_y\Big)\frac{\partial}{\partial \kappa}\Big(\Pi\big(\varphi_i(\mathbf{p}_j)\big)_y\Big)\bigg)\Bigg), \end{aligned} \tag{85}$$

assuming the suffix $(\cdot)_x$ or $(\cdot)_y$ gives the x or y coordinate of the 2D point. Given that $\mathbf{Q}_{i,j} = \varphi_i(\mathbf{p}_j)$ and its homogeneous coordinates are given as $\tilde{\mathbf{Q}}_{i,j}$, we differentiate both sides of the perspective projection equation w.r.t. κ to obtain:

$$\begin{aligned} \underbrace{\frac{\partial}{\partial \kappa}\Big(\Pi\big(\varphi_i(\mathbf{p}_j)\big)\Big)}_{2\times 2m} = \frac{\partial}{\partial \kappa}\left(\begin{bmatrix} \Pi\big(\varphi_i(\mathbf{p}_j)\big)_x \\ \Pi\big(\varphi_i(\mathbf{p}_j)\big)_y \end{bmatrix}\right) = \frac{\partial}{\partial \kappa}\left(\begin{bmatrix} \frac{f_x}{(\mathbf{Q}_{i,j})_Z} & 0 & 0 & c_x \\ 0 & \frac{f_y}{(\mathbf{Q}_{i,j})_Z} & 0 & c_y \end{bmatrix} f_i^{-1}\tilde{\mathbf{Q}}_{i,j}\right) \\ = \underbrace{\frac{\partial}{\partial \kappa}\left(\begin{bmatrix} \frac{f_x}{(\mathbf{Q}_{i,j})_Z} & 0 & 0 & c_x \\ 0 & \frac{f_y}{(\mathbf{Q}_{i,j})_Z} & 0 & c_y \end{bmatrix}\right)}_{2\times 8m}\underbrace{\big(f_i^{-1}\tilde{\mathbf{Q}}_{i,j}\otimes \mathbb{1}_{2m}\big)}_{8m\times 2m} + \left(\underbrace{\begin{bmatrix} \frac{f_x}{(\mathbf{Q}_{i,j})_Z} & 0 & 0 & c_x \\ 0 & \frac{f_y}{(\mathbf{Q}_{i,j})_Z} & 0 & c_y \end{bmatrix}}_{[2\times 4]}\underbrace{f_i^{-1}}_{4\times 4}\underbrace{\frac{\partial \tilde{\mathbf{Q}}_{i,j}}{\partial \kappa}}_{4\times 2m}\right). \end{aligned} \tag{86}$$

Given that $\frac{\partial}{\partial \kappa}\big(\frac{f_x}{(\mathbf{Q}_{i,j})_Z}\big) = -\frac{f_x}{(\mathbf{Q}_{i,j})_Z^2}\frac{\partial(\mathbf{Q}_{i,j})_Z}{\partial \kappa}$, to compute equation (86), we need the expression for the term $\frac{\partial(\mathbf{Q}_{i,j})_Z}{\partial \kappa}$, which can be obtained from:

$$\begin{aligned} \underbrace{\frac{\partial \tilde{\mathbf{Q}}_{i,j}^{\top}}{\partial \kappa}}_{1\times 8m} = \begin{bmatrix} \frac{\partial(\mathbf{Q}_{i,j})_X}{\partial \kappa} & \frac{\partial(\mathbf{Q}_{i,j})_Y}{\partial \kappa} & \frac{\partial(\mathbf{Q}_{i,j})_Z}{\partial \kappa} & \mathbb{0}_{1\times 2m} \end{bmatrix} = \frac{\partial}{\partial \kappa}\left(\begin{bmatrix} \underbrace{\rho\big(\Delta(r_j, \theta_j), \mathbf{D}\big)}_{[1\times l]} & \tilde{\mathbf{Q}}_{i,j}^{\top} \end{bmatrix} \varepsilon_{\lambda,i} \begin{bmatrix} \tilde{\mathbf{C}}_i \\ \mathbf{0} \end{bmatrix}\right) \\ = \underbrace{\begin{bmatrix} \frac{\partial}{\partial \kappa}\Big(\rho\big(\Delta(r_j, \theta_j), \mathbf{D}\big)\Big) & \frac{\partial \sin\theta_j}{\partial \kappa} & \frac{\partial r_j}{\partial \kappa} & \frac{\partial \cos\theta_j}{\partial \kappa} & 0 \end{bmatrix}}_{1\times 2m(l+4)} \underbrace{\left(\left(\varepsilon_{\lambda,i}\begin{bmatrix} \tilde{\mathbf{C}}_j \\ \mathbf{0} \end{bmatrix}\right)\otimes \mathbb{1}_{2m}\right)}_{2m(l+4)\times 8m}, \end{aligned} \tag{87}$$

and the four terms inside the first $1 \times 2m(l+4)$ matrix of equation (87), for the k-th element of the source points $\mathbf{D}$, is given by the expressions in the frames below.

For **planar** topology

This is the simplest case where $\frac{\partial}{\partial\kappa}\Big(\rho\big(\Delta(x_j, y_j), \mathbf{D}_k\big)\Big)$ is given by equation (51) and $\frac{\partial(\mathbf{Q}_{i,j})}{\partial\kappa}$ is trivial, simply obtained by populating ones and zeros in the right sequence.

For **cylindrical** topology

$$\frac{\partial}{\partial\kappa}\Big(\rho\big(\Delta(r_j, \theta_j), \mathbf{D}_k\big)\Big) = \frac{-1}{\rho\big(\Delta(r_j, \theta_j), \mathbf{D}_k\big)} \begin{bmatrix} \underset{[1\times(j-1)]}{\mathbf{0}} & \Big(r_j - D_k^{\mathrm{Y}}\Big) & \underset{[1\times(m-1)]}{\mathbf{0}} & \Big(D_k^{\mathrm{Z}} \sin\theta_j - D_k^{\mathrm{X}} \cos\theta_j\Big) & \underset{[1\times(m-j)]}{\mathbf{0}} \end{bmatrix} \tag{88}$$

and

$$\frac{\partial \sin\theta_j}{\partial\kappa} = \begin{bmatrix} \underset{1\times(m+j-1)}{\mathbf{0}} & \cos\theta_j & \underset{1\times(m-j)}{\mathbf{0}} \end{bmatrix} \qquad \frac{\partial r_j}{\partial\kappa} = \begin{bmatrix} \underset{1\times(j-1)}{\mathbf{0}} & 1 & \underset{1\times(2m-j)}{\mathbf{0}} \end{bmatrix}$$
$$\frac{\partial \cos\theta_j}{\partial\kappa} = -\begin{bmatrix} \underset{1\times(m+j-1)}{\mathbf{0}} & \sin\theta_j & \underset{1\times(m-j)}{\mathbf{0}} \end{bmatrix}. \tag{89}$$

For **spherical** topology

Equation (87) is repurposed as:

$$\frac{\partial\tilde{\mathbf{Q}}_{i,j}^{\top}}{\partial\kappa} = \underbrace{\begin{bmatrix} \frac{\partial}{\partial\kappa}\Big(\rho\big(\Delta(\theta_{1,j}, \theta_{2,j}), \mathbf{D}\big)\Big) & \frac{\partial}{\partial\kappa}\big(-\cos\theta_{1,j}\cos\theta_{2,j}\big) & \frac{\partial}{\partial\kappa}\big(-\sin\theta_{1,j}\big) & \frac{\partial}{\partial\kappa}\big(-\cos\theta_{1,j}\sin\theta_{2,j}\big) & 0 \end{bmatrix}}_{1\times 2m(l+4)} \mathcal{K}_{\varepsilon}, \tag{90}$$

where $\mathcal{K}_{\varepsilon}$ is the $2m(l+4)\times 8m$ sized matrix on the right of equation (87). Expanding the terms for the k-th element of the source point $\mathbf{D}$:

$$\frac{\partial}{\partial\kappa}\Big(\rho\big(\Delta(\theta_{1,j}, \theta_{2,j}), \mathbf{D}_k\big)\Big) = \frac{-1}{\rho\big(\Delta(\theta_{1,j}, \theta_{2,j}), \mathbf{D}_k\big)} \begin{bmatrix} \ldots & \Big(\frac{\partial\mathbf{Q}_{i,j}}{\partial\theta_{1,j}}\Big)^{\top}\mathbf{D}_k & \ldots & \Big(\frac{\partial\mathbf{Q}_{i,j}}{\partial\theta_{2,j}}\Big)^{\top}\mathbf{D}_k & \ldots \end{bmatrix}, \tag{91}$$

where:

$$\begin{bmatrix} \frac{\partial\mathbf{Q}_{i,j}}{\partial\theta_{1,j}} & \frac{\partial\mathbf{Q}_{i,j}}{\partial\theta_{2,j}} \end{bmatrix} = \begin{bmatrix} \Delta_{\theta_{1,j}} & \Delta_{\theta_{2,j}} \end{bmatrix} = \begin{bmatrix} \sin\theta_{1,j}\cos\theta_{2,j} & \cos\theta_{1,j}\sin\theta_{2,j} \\ -\cos\theta_{1,j} & 0 \\ \sin\theta_{1,j}\sin\theta_{2,j} & -\cos\theta_{1,j}\cos\theta_{2,j} \end{bmatrix}, \tag{92}$$

and the other three partial derivatives of equation (90) are trivial.

Moving to the second set of terms in equation 84:

$$\frac{\partial h_{\text{rep}}}{\partial \boldsymbol{\omega}_i} = \frac{\partial}{\partial \boldsymbol{\omega}_i}\Big(\|\mathbf{q}_{i,j} - \Pi\big(\varphi_i(\mathbf{p}_j)\big)\|\Big) = \frac{-1}{\|\mathbf{q}_{i,j} - \Pi\big(\varphi_i(\mathbf{p}_j)\big)\|}\Bigg(\bigg(\Big((\mathbf{q}_j)_x - \Pi\big(\varphi_i(\mathbf{p}_j)\big)_x\Big)\frac{\partial}{\partial \boldsymbol{\omega}_i}\Big(\Pi\big(\varphi_i(\mathbf{p}_j)\big)_x\Big)\bigg)$$
$$+\bigg(\Big((\mathbf{q}_j)_y - \Pi\big(\varphi_i(\mathbf{p}_j)\big)_y\Big)\frac{\partial}{\partial \boldsymbol{\omega}_i}\Big(\Pi\big(\varphi_i(\mathbf{p}_j)\big)_y\Big)\bigg)\Bigg), \tag{93}$$

where:

$$\frac{\partial}{\partial \boldsymbol{\omega}_i}\Big(\Pi\big(\varphi_i(\mathbf{p}_j)\big)\Big) = \frac{\partial}{\partial \boldsymbol{\omega}_i}\Bigg(\begin{bmatrix} \frac{f_x}{(\mathbf{Q}_{i,j})_Z} & 0 & 0 & c_x \\ 0 & \frac{f_y}{(\mathbf{Q}_{i,j})_Z} & 0 & c_y \end{bmatrix}\Bigg)\Big(f_i^{-1}\tilde{\mathbf{Q}}_{i,j} \otimes \mathbb{1}_{3l}\Big) + \Bigg(\begin{bmatrix} \frac{f_x}{(\mathbf{Q}_{i,j})_Z} & 0 & 0 & c_x \\ 0 & \frac{f_y}{(\mathbf{Q}_{i,j})_Z} & 0 & c_y \end{bmatrix} f_i^{-1}\frac{\partial \tilde{\mathbf{Q}}_{i,j}}{\partial \boldsymbol{\omega}_i}\Bigg). \tag{94}$$

Once again, to compute equation (94), we need the expression for the term $\frac{\partial (\mathbf{Q}_{i,j})_Z}{\partial \boldsymbol{\omega}_i}$, which can be obtained from:

$$\underbrace{\frac{\partial \tilde{\mathbf{Q}}_{i,j}^{\top}}{\partial \boldsymbol{\omega}_i}}_{1\times 12l} = \begin{bmatrix} \frac{\partial (\mathbf{Q}_{i,j})_X}{\partial \boldsymbol{\omega}_i} & \frac{\partial (\mathbf{Q}_{i,j})_Y}{\partial \boldsymbol{\omega}_i} & \frac{\partial (\mathbf{Q}_{i,j})_Z}{\partial \boldsymbol{\omega}_i} & \mathbb{0}_{1\times 3l} \end{bmatrix}$$
$$= \begin{bmatrix} \underbrace{\rho\big(\Delta(r_j, \theta_j), \mathbf{D}\big)}_{1\times l} & \sin\theta_j & r_j & \cos\theta_j & 1 \end{bmatrix} \varepsilon_{\lambda,i} \underbrace{\begin{bmatrix} \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} & \mathbb{1}_l & \mathbb{0}_{l\times 3l} \\ \multicolumn{6}{c}{\mathbb{0}_{4\times 12l}} \end{bmatrix}}_{(l+4)\times 12l}. \tag{95}$$

Obviously, when the indices of $\tilde{\mathbf{Q}}$ and $\boldsymbol{\omega}$ do not match, the rows of the Jacobian matrix are all zeroes, *i.e.* $\frac{\partial \tilde{\mathbf{Q}}_{i_1,j}^{\top}}{\partial \boldsymbol{\omega}_{i_2}} = \mathbf{0}_{1\times 12l}$ when $i_1 \neq i_2$. Our next objective is to compute the Jacobian matrix relating the change of h_{iso} to $\boldsymbol{\xi}$, providing us with:

$$\mathbf{J}_{I,g} = \frac{\partial h_{\text{iso}}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \frac{\partial h_{\text{iso}}}{\partial \kappa} & \frac{\partial h_{\text{iso}}}{\partial \boldsymbol{\omega}} \end{bmatrix}. \tag{96}$$

The two terms on the right most matrix of equation (96) are expanded below:

$$\frac{\partial h_{\text{iso}}}{\partial \kappa} = \frac{1}{\zeta_1}\Bigg(\Big((\mathbf{Q}_{i,j})_X - (\mathbf{Q}_{i,q})_X\Big)\Big(\frac{\partial (\mathbf{Q}_{i,j})_X}{\partial \kappa} - \frac{\partial (\mathbf{Q}_{i,q})_X}{\partial \kappa}\Big) + \Big((\mathbf{Q}_{i,j})_Y - (\mathbf{Q}_{i,q})_Y\Big)\Big(\frac{\partial (\mathbf{Q}_{i,j})_Y}{\partial \kappa} - \frac{\partial (\mathbf{Q}_{i,q})_Y}{\partial \kappa}\Big)$$
$$+\Big((\mathbf{Q}_{i,j})_Z - (\mathbf{Q}_{i,q})_Z\Big)\Big(\frac{\partial (\mathbf{Q}_{i,j})_Z}{\partial \kappa} - \frac{\partial (\mathbf{Q}_{i,q})_Z}{\partial \kappa}\Big)\Bigg) - \frac{1}{\zeta_2}\frac{\partial d_{j,q}^2}{\partial \kappa}, \tag{97}$$

where:

$$\frac{\partial d_{j,q}^2}{\partial \kappa} = 2\begin{bmatrix} 0 & \dots & (r_j - r_q) & \dots & -(r_j - r_q) & \dots & 0 & \dots & (\theta_j - \theta_q) & \dots & -(\theta_j - \theta_q) & \dots & 0 \end{bmatrix}. \tag{98}$$

For **planar** topology

Using a similar expansion, it can be shown that for the planar topology, we have:

$$\frac{\partial d_{j,q}^2}{\partial \kappa} = 2\begin{bmatrix}0 & \dots & (x_{1,j}-x_{1,q}) & \dots & -(x_{1,j}-x_{1,q}) & \dots & 0 & \dots & (x_{2,j}-x_{2,q}) & \dots & -(x_{2,j}-x_{2,q}) & \dots & 0\end{bmatrix}. \tag{99}$$

For **spherical** topology

It can also be shown that for the spherical ToTem we have:

$$\frac{\partial d_{j,q}^2}{\partial \kappa} = 2\begin{bmatrix}0 & \dots & (\theta_{1,j}-\theta_{1,q}) & \dots & -(\theta_{1,j}-\theta_{1,q}) & \dots & 0 & \dots & (\theta_{2,j}-\theta_{2,q}) & \dots & -(\theta_{2,j}-\theta_{2,q}) & \dots & 0\end{bmatrix}. \tag{100}$$

The other term from equation (96) is:

$$\begin{aligned}\frac{\partial h_{\text{iso}}}{\partial \boldsymbol{\omega}} = \frac{1}{\zeta_1}\Bigg(\Big((\mathbf{Q}_{i,j})_X - (\mathbf{Q}_{i,q})_X\Big)\Big(\frac{\partial(\mathbf{Q}_{i,j})_X}{\partial \boldsymbol{\omega}} - \frac{\partial(\mathbf{Q}_{i,q})_X}{\partial \boldsymbol{\omega}}\Big) \\ +\Big((\mathbf{Q}_{i,j})_Y - (\mathbf{Q}_{i,q})_Y\Big)\Big(\frac{\partial(\mathbf{Q}_{i,j})_Y}{\partial \boldsymbol{\omega}} - \frac{\partial(\mathbf{Q}_{i,q})_Y}{\partial \boldsymbol{\omega}}\Big)+\Big((\mathbf{Q}_{i,j})_Z - (\mathbf{Q}_{i,q})_Z\Big)\Big(\frac{\partial(\mathbf{Q}_{i,j})_Z}{\partial \boldsymbol{\omega}} - \frac{\partial(\mathbf{Q}_{i,q})_Z}{\partial \boldsymbol{\omega}}\Big)\Bigg).\end{aligned} \tag{101}$$

References

Azagra, P., Sostres, C., Ferrandez, Á., Riazuelo, L., Tomasini, C., Barbed, O. L., Morlana, J., Recasens, D., Batlle, V. M., Gómez-Rodríguez, J. J., et al. (2022). Endomapper dataset of complete calibrated endoscopy procedures. *arXiv preprint arXiv:2204.14240*. 44, 49

Bai, F. and Bartoli, A. (2022). Procrustes analysis with deformations: A closed-form solution by eigenvalue decomposition. *International Journal of Computer Vision*, pages 1–27. 32

Barron, J. T. and Malik, J. (2014). Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687. 4

Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T., and Pizarro, D. (2015). Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118. 4

Bartoli, A., Perriollat, M., and Chambon, S. (2010). Generalized thin-plate spline warps. *International Journal of Computer Vision*, 88(1):85–110. 15

Bellekens, B., Spruyt, V., Berkvens, R., and Weyn, M. (2014). A survey of rigid 3d pointcloud registration algorithms. In *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pages 8–13. 11

Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., and Silva, C. T. (2017). A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library. 11

Bernard, F., Schmidt, F. R., Thunberg, J., and Cremers, D. (2017). A combinatorial solution to non-rigid 3d shape-to-image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1009. 11

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359. 16, 40

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics. 11, 28

Bezdek, J. C. and Hathaway, R. J. (2002). Some notes on alternating optimization. In *AFSS international conference on fuzzy systems*, pages 288–300. Springer. 19

Bhatia, R. and Kittaneh, F. (2000). Notes on matrix arithmetic–geometric mean inequalities. *Linear Algebra and Its Applications*, 308(1-3):203–211. 21

Bhoi, A. (2019). Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*. 11

Biswas, P., Toh, K.-C., and Ye, Y. (2008). A distributed sdp approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM Journal on Scientific Computing*, 30(3):1251–1277. 9

Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585. 14, 15

Bregler, C., Hertzmann, A., and Biermann, H. (2000). Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 690–696. IEEE. 4, 9

Calla, L. A. R., Perez, L. J. F., and Montenegro, A. A. (2019). A minimalistic approach for fast computation of geodesic distances on triangular meshes. *Computers & Graphics*, 84:77–92. 34

Cao, Y., Wu, Z., and Shen, C. (2017). Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182. 10

Carlson, D., Haynsworth, E., and Markham, T. (1974). A generalization of the schur complement by means of the moore–penrose inverse. *SIAM Journal on Applied Mathematics*, 26(1):169–175. 37

Casillas-Perez, D., Pizarro, D., Fuentes-Jimenez, D., Mazo, M., and Bartoli, A. (2019). Equiareal shape-from-template. *Journal of Mathematical Imaging and Vision*, 61:607–626. 41, 42, 43, 45

Castellani, U. and Bartoli, A. (2020). 3d shape registration. *3D Imaging, Analysis and Applications*, pages 353–411. 11

Chhatkuli, A., Pizarro, D., and Bartoli, A. (2014). Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In *British Machine Vision Conference*. 4, 9, 38

Chhatkuli, A., Pizarro, D., Bartoli, A., and Collins, T. (2016). A stable analytical framework for isometric shape-from-template by surface integration. *IEEE transactions on pattern analysis and machine intelligence*, 39(5):833–850. 9, 38, 44

Chhatkuli, A., Pizarro, D., Collins, T., and Bartoli, A. (2017). Inextensible non-rigid structure-from-motion by second-order cone programming. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2428–2441. 5, 8, 9, 10, 12, 15, 17, 18, 24, 38

Dai, Y., Li, H., and He, M. (2014). A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107:101–122. 9, 38

Dubrovina, A. and Kimmel, R. (2011). Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures. *Advances in Adaptive Data Analysis*, 3(01n02):203–228. 9

d'Aspremont, A. and Boyd, S. (2003). Relaxations and randomized methods for nonconvex qcqps. *EE392o Class Notes, Stanford University*, 1:1–16. 17

Faugeras, O. and Luong, Q.-T. (2001). *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press. 4

Golyanik, V., Jonas, A., Stricker, D., and Theobalt, C. (2020). Intrinsic dynamic shape prior for dense non-rigid structure from motion. In *2020 International Conference on 3D Vision (3DV)*, pages 692–701. IEEE. 11

Gotardo, P. F. and Martinez, A. M. (2011). Kernel non-rigid structure from motion. In *2011 International Conference on Computer Vision*, pages 802–809. IEEE. 38

Granshaw, S. (1980). Bundle adjustment methods in engineering photogrammetry. *The Photogrammetric Record*, 10(56):181–207. 8

Hamsici, O. C., Gotardo, P. F., and Martinez, A. M. (2012). Learning spatially-smooth mappings in non-rigid structure from motion. In *European Conference on computer vision*, pages 260–273. Springer. 38

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition. 4, 12

Hilbert, D. and Cohn-Vossen, S. (2021). *Geometry and the Imagination*, volume 87. American Mathematical Soc. 13

Hong, M., Razaviyayn, M., Luo, Z.-Q., and Pang, J.-S. (2015). A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Processing Magazine*, 33(1):57–77. 23

Horn, B. (1975). Determining shape from shading. *The Psychology of Computer Vision*. 4

Horn, B. K. and Weldon, E. (1986). Filtering closed curves. *IEEE transactions on pattern analysis and machine intelligence*, (5):665–668. 34

Hu, Y., Zhou, M., and Wu, Z. (2009). A dense point-to-point alignment method for realistic 3d face morphing and animation. *International Journal of Computer Games Technology*, 2009. 15

Iske, A. (2004). *Multiresolution methods in scattered data modelling*, volume 37. Springer Science & Business Media. 14

Ji, P., Li, H., Dai, Y., and Reid, I. (2017). " maximizing rigidity" revisited: a convex programming approach for generic 3d shape reconstruction from multiple perspective views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 929–937. 5, 8, 9, 10, 17, 18, 38

Joshi, P., Meyer, M., DeRose, T., Green, B., and Sanocki, T. (2007). Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71–es. 42

Kahl, F. and Henrion, D. (2007). Globally optimal estimates for geometric reconstruction problems. *International Journal of Computer Vision*, 74:3–15. 10

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0. 40

Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13. 16

Kelley, C. T. (1999). *Iterative methods for optimization*. SIAM. 34

Khan, F., Salahuddin, S., and Javidnia, H. (2020). Deep learning-based monocular depth estimation methods—a state-of-the-art review. *Sensors*, 20(8):2272. 11

Kong, C. and Lucey, S. (2019). Deep interpretable non-rigid structure from motion. *arXiv preprint arXiv:1902.10840*. 10

Kreyszig, E. (2013). *Differential geometry*. Courier Corporation. 14

Kumar, S. and Van Gool, L. (2022). Organic priors in non-rigid structure from motion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 71–88. Springer. 9

Kurosh, A. G. (2014). *Lectures in general algebra*. Elsevier. 23

Li, B., Shen, C., Dai, Y., Van Den Hengel, A., and He, M. (2015a). Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127. 10

Li, L. et al. (2015b). *Selected applications of convex optimization*, volume 103. Springer. 10

Liu, F., Shen, C., Lin, G., and Reid, I. (2015). Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039. 10

Liu, Y., Pears, N., Rosin, P. L., and Huber, P. (2020). *3D imaging, analysis and applications*. Springer. 11

Low, K.-L. (2004). Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4(10):1–3. 11

Novotny, D., Ravi, N., Graham, B., Neverova, N., and Vedaldi, A. (2019). C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7688–7697. 10

Ortega, J. M. and Rheinboldt, W. C. (2000). *Iterative solution of nonlinear equations in several variables*. SIAM. 19

Östlund, J., Varol, A., Ngo, D. T., and Fua, P. (2012). Laplacian meshes for monocular 3d shape recovery. In *ECCV (3)*, pages 412–425. 9

Parashar, S., Pizarro, D., and Bartoli, A. (2017). Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2442–2454. 9, 38

Parashar, S., Pizarro, D., and Bartoli, A. (2019). Local deformable 3d reconstruction with cartan's connections. *IEEE transactions on pattern analysis and machine intelligence*, 42(12):3011–3026. 5

Parashar, S., Pizarro, D., Bartoli, A., and Collins, T. (2015). As-rigid-as-possible volumetric shape-from-template. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 891–899. 9, 11

Perriollat, M. and Bartoli, A. (2013). A computational model of bounded developable surfaces with application to image-based three-dimensional reconstruction. *Computer Animation and Virtual Worlds*, 24(5):459–476. 38

Perriollat, M., Hartley, R., and Bartoli, A. (2011). Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision*, 95(2):124–137. 4, 5, 9, 19

Pressley, A. N. (2010). *Elementary differential geometry*. Springer Science & Business Media. 34

Probst, T., Paudel, D. P., Chhatkuli, A., and Gool, L. V. (2019). Convex relaxations for consensus and non-minimal problems in 3d vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10233–10242. 10

Rabbani, N., Calvet, L., Espinel, Y., Le Roy, B., Ribeiro, M., Buc, E., and Bartoli, A. (2022). A methodology and clinical dataset with ground-truth to evaluate registration accuracy quantitatively in computer-assisted laparoscopic liver resection. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 10(4):441–450. 28

Russell, C., Fayad, J., and Agapito, L. (2011). Energy based multiple model fitting for non-rigid structure from motion. In *CVPR 2011*, pages 3009–3016. IEEE. 11

Sabzevari, R., Del Bue, A., and Murino, V. (2012). Multi-view photometric stereo using semi-isometric mappings. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 547–554. IEEE. 9

Salzmann, M. and Fua, P. (2010). Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944. 4, 5, 9

Salzmann, M. and Urtasun, R. (2012). Beyond feature points: Structured prediction for monocular non-rigid 3d reconstruction. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pages 245–259. Springer. 9

Sengupta, A. and Bartoli, A. (2021). Colonoscopic 3d reconstruction by tubular non-rigid structure-from-motion. *International Journal of Computer Assisted Radiology and Surgery*, 16(7):1237–1241. 44

Sheffer, A. and de Sturler, E. (2001). Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with computers*, 17(3):326–337. 56

Sheffer, A., Lévy, B., Mogilnitsky, M., and Bogomyakov, A. (2005). Abf++: fast and robust angle based flattening. *ACM Transactions on Graphics (TOG)*, 24(2):311–330. 7, 13, 56

Takmaz, A., Paudel, D. P., Probst, T., Chhatkuli, A., Oswald, M. R., and Van Gool, L. (2020). Unsupervised monocular depth reconstruction of non-rigid scenes. *arXiv preprint arXiv:2012.15680*. 10

Torr, P. H. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156. 39, 40

Torresani, L. and Bregler, C. (2002). Space-time tracking. In *European Conference on Computer Vision*, pages 801–812. Springer. 9

Torresani, L., Hertzmann, A., and Bregler, C. (2008). Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892. 4

Torresani, L., Yang, D. B., Alexander, E. J., and Bregler, C. (2001). Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE. 9

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (1999). Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer. 8, 34, 35, 36

Varol, A., Salzmann, M., Tola, E., and Fua, P. (2009). Template-free monocular reconstruction of deformable surfaces. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1811–1818. IEEE. 9, 38

Yang, J., Li, H., Campbell, D., and Jia, Y. (2015). Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254. 11, 28