Image-based deformable 3D reconstruction using differential geometry and cartan's connections

Shaifali Parashar

UFR DE MÉDECINE
ET DES PROFESSIONS PARAMÉDICALES
Université Clermont Auvergne

Année N° d'ordre

ECOLE DOCTORALE
DES SCIENCES POUR L'INGENIEUR

THÈSE

Présentée à l'Université Clermont Auvergne
pour l'obtention du grade de **DOCTEUR**
(Décret du 5 juillet 1984)

Specialité
COMPUTER VISION

Soutenue le
23 Novembre 2017

Shaifali Parashar

Image-based Deformable 3D Reconstruction using Differential Geometry and Cartan's Connections

---

Présidente	Marie-Odile Berger, Research Director, INRIA-Nancy, France
Rapporteurs	Mathieu Salzmann, Researcher, EPFL, Switzerland
	Alessio Del Bue, Researcher, IIT-Genova, Italy
Directeur de thèse	Adrien Bartoli, Professor, Université Clermont Auvergne, France
Co-encadrant	Daniel Pizarro, Lecturer, Universidad de Alcalá, Spain

EnCoV Institut Pascal
EnCoV, Institut Pascal, UMR 6602 CNRS,
Université Clermont Auvergne, SIGMA
Faculté de Médecine
28 Place Henri Dunant, Clermont-Ferrand
Tel: +33 4 73 17 81 23

To my family and Alok.

Acknowledgements

I would like to express my gratitude to my supervisors Adrien and Dani who always inspired and guided me during my PhD. With their immense knowledge about the field, they helped me a lot in looking for the right answers which led to my research. I am very thankful for their kindness, patience and a friendly assistance. I highly appreciate the fact that Adrien has always been available for a discussion. I am very grateful to Dani for his support. I have always found him insightful and his contributions are very significant in the development of this thesis. Both of them have been very friendly and caring, which has made this PhD a great learning experience for me.

I would also like to thank Toby for his guidance. His knowledge and perspective about research has helped me a lot in my research. It was a great pleasure to have spent time in this lab where everyone is so friendly and helpful. I have learnt a great deal from all of them and I will always cherish the memories of this time spent together.

In addition, I would like to thank the thesis jury reporters, Mathieu Salzmann and Alessio Del Bue for carefully reading the document and providing constructive remarks that has provided further encouragement to me for research in the field. I would also like to thank Marie-Odile Berger for her role of the President of the jury and her valuable comments.

I am also thankful to my parents and Alok for their unconditional love which helped a lot during these PhD years.

Abstract

Reconstructing the 3D shape of objects from multiple images is an important goal in computer vision and has been extensively studied for both rigid and non-rigid (or deformable) objects. Structure-from-Motion (SfM) is an algorithm that performs the 3D reconstruction of rigid objects using the inter-image visual motion from multiple images obtained from a moving camera. SfM is a very accurate and stable solution. Deformable 3D reconstruction, however, has been widely studied for monocular images (obtained from a single camera) and still remains an open research problem. The current methods exploit visual cues such as the inter-image visual motion and shading in order to formalise a reconstruction algorithm. This thesis focuses on the use of the inter-image visual motion for solving this problem. Two types of scenarios exist in the literature: *1)* Non-Rigid Structure-from-Motion (NRSfM) and *2)* Shape-from-Template (SfT). The goal of NRSfM is to reconstruct multiple shapes of a deformable object as viewed in multiple images while SfT (also referred to as template-based reconstruction) uses a single image of a deformed object and its 3D template (a textured 3D shape of the object in one configuration) to recover the deformed shape of the object.

We propose an NRSfM method to reconstruct the deformable surfaces undergoing isometric deformations (the objects do not stretch or shrink under an isometric deformation) using Riemannian geometry. This allows NRSfM to be expressed in terms of Partial Differential Equations (PDE) and to be solved algebraically. We show that the problem has linear complexity and the reconstruction algorithm has a very low computational cost compared to existing NRSfM methods. This work motivated us to use differential geometry and Cartan's theory of connections to model NRSfM, which led to the possibility of extending the solution to deformations other than isometry. In fact, this led to a unified theoretical framework for modelling and solving both NRSfM and SfT for various types of deformations. In addition, it also makes it possible to have a solution to SfT which does not require an explicit modelling of deformation. An important point is that most of the NRSfM and SfT methods reconstruct the thin-shell surface of the object. The reconstruction of the entire volume (the thin-shell surface and the interior) has not been explored yet. We propose the first SfT method that reconstructs the entire volume of a deformable object.

Resumé

La reconstruction 3D d'objets à partir de plusieurs images est un objectif important de la vision par ordinateur. Elle a été largement étudiée pour les objets rigides et non rigides (ou déformables). Le Structure-from-Motion (SfM) est un algorithme qui effectue la reconstruction 3D d'objets rigides en utilisant le mouvement visuel entre plusieurs images obtenues à l'aide d'une caméra en mouvement. Le SfM est une solution très précise et stable. La reconstruction 3D déformable a été largement étudiée pour les images monoculaires (obtenues à partir d'une seule caméra) mais reste un problème ouvert. Les méthodes actuelles exploitent des indices visuels tels que le mouvement visuel inter-image et l'ombrage afin de construire un algorithme de reconstruction. Cette thèse se concentre sur l'utilisation du mouvement visuel inter-image pour résoudre ce problème. Deux types de scénarios existent dans la littérature: *1)* le Non-Rigid Structure-from-Motion (NRSfM) et *2)* le Shape-from-Template (SfT). L'objectif du NRSfM est de reconstruire plusieurs formes d'un objet déformable tel qu'il apparaît dans plusieurs images, alors que le SfT (également appelé reconstruction à partir d'un modèle de référence) utilise une seule image d'un objet déformé et son modèle 3D de référence (une forme 3D texturée de l'objet dans une configuration) pour estimer la forme déformée de l'objet.

Nous proposons une méthode de NRSfM pour reconstruire les surfaces déformables soumises à des déformations isométriques (les objets ne s'étirent pas ou ne se contractent pas sous une déformation isométrique) en utilisant la géométrie riemannienne. Cela permet d'exprimer le NRSfM en termes d'équations aux dérivées partielles et de le résoudre algébriquement. Nous montrons que le problème a une complexité linéaire et que l'algorithme de reconstruction proposé a un coût de calcul très bas comparé aux méthodes existantes de NRSfM. Ce travail nous a motivé à utiliser la géométrie différentielle et la théorie des connexions de Cartan pour modéliser le NRSfM, ce qui nous a permis d'étendre la solution à des déformations autres que l'isométrie. En fait, cela a conduit à un cadre théorique unifié pour modéliser et résoudre le NRSfM et le SfT pour différents types de déformations. Ce cadre permet également d'avoir une solution au SfT qui ne nécessite pas de modélisation explicite de la déformation. Un point important est que la plupart des méthodes de NRSfM et de SfT reconstruisent la surface de l'objet (hypothèse coque mince). La reconstruction de l'ensemble d'un volume (la surface et l'intérieur d'un objet) n'avait pas encore été explorée. Nous proposons la première méthode de SfT qui reconstruit le volume complet d'un objet déformable.

Contents

List of Figures

List of Tables

List of Abbreviations

ARAP As-Rigid-As-Possible

CS Christoffel Symbols

DCT Discrete Cosine Transformation

GS Global Smoothness

IL Infinitesimal Linearity

IP Infinitesimal Planarity

LLS Linear Least Squares

LR Local Rigidity

MDH Maximum Depth Heuristics

NRSfM Non-Rigid Structure-from-Motion

PDE Partial Differential Equations

RMSE Root Mean Square Error

SfM Structure-from-Motion

SfT Shape-from-Template

SIFT Scale Invariant Feature Transform

SOCP Second-order Cone Program

SVD Singular Value Decomposition

TPS Thin-Plate Splines

ToF Time-of-Flight

Chapter 1

Introduction

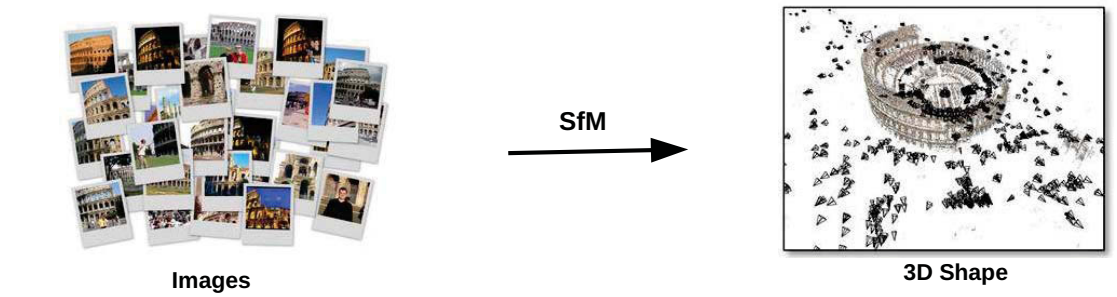
1.1 Background

An important task in 3D computer vision is to recover 3D information from 2D images obtained by the camera. This task is widely termed as 3D reconstruction. Although there are active image sensors such as the Kinect and Time-of-Flight (ToF) cameras available which can obtain the depth of the view under consideration, passive 3D reconstruction from images remains an interesting topic for researchers because the scope of 3D sensors is limited due to the various constraints of size, cost and accuracy. 3D reconstruction methods rely on various visual cues from images (such as shading, texture, silhouettes, contours and motion) in order to find 3D descriptors such as the depth map and local surface orientation (or normals) of the objects.

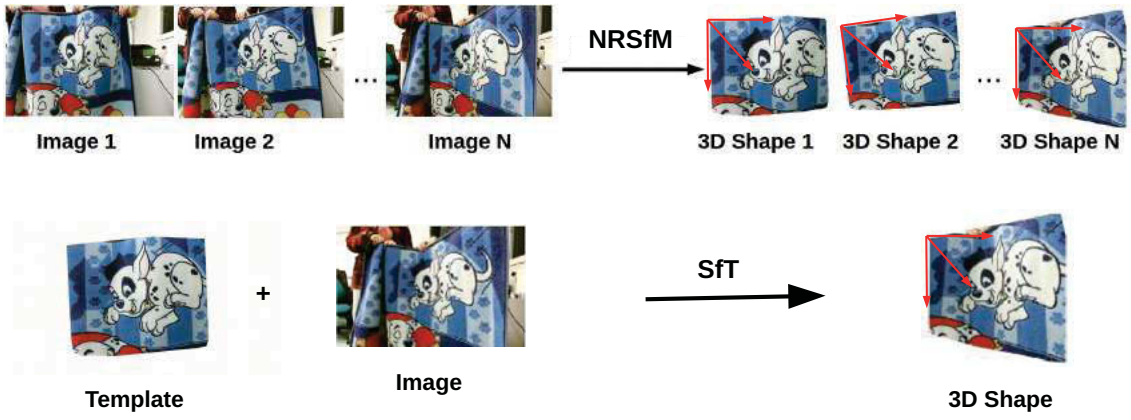
The objects found in nature can be roughly classified into rigid or non-rigid (or deformable) objects. The 3D reconstruction of rigid objects using motion cues, also known as Structure-from-Motion (SfM) [Hartley and Zisserman, 2000] (see figure 1.1a), has been widely studied for the past few decades and there are solutions available which are stable and accurate. SfM exploits the inter-image visual motion information in order to reconstruct 3D from multiple 2D images taken from different views of a rigid object. Rigidity allows the inter-image visual motion to be expressed in terms of the rotation and translation of the camera coordinate frames of the images. However, SfM cannot be extended to deformable objects as between any two images, the deformable object may undergo a deformation and therefore, the inter-image visual motion cannot be expressed only in terms of the camera rotation and translation.

In the past decade, the deformable 3D reconstruction problem has been studied extensively in over a hundred research papers. Some of the methods combine motion with other visual cues to disambiguate the problem and make it well-posed. For example, [Liu-Yin et al., 2016; Moreno-Noguer et al., 2009; Varol et al., 2012b] combine shading with motion, [Gallardo et al., 2016, 2017] combine shading and contours with motion and [Choe and Kashyap, 1991; White and Forsyth, 2006] combine shading with texture. However, the existing solu-

tions are not close to SfM in terms of accuracy and stability. Deformable 3D reconstruction is an important problem to solve as it has a wide range of applications such as in the medical, sports, entertainment and advertising domains. Some applications are explored in augmented reality: *1)* [Smith et al., 2016] shows how to study the impact of a soft ball on various surfaces. This is useful in designing and testing sports equipments. *2)* [Haouchine et al., 2013, 2016; Koo et al., 2017; Maier-Hein et al., 2014] show how to augment the deformations of the body organs in order to aid minimally invasive medical surgeries. *3)* [Collins and Bartoli, 2015; Ngo et al., 2015] recover the deformation of the objects in real-time which can be useful in various industries. For example, they can be used by online shopping companies to enable the customers to try clothes and accessories virtually. Figure 1.2 shows some of these applications.



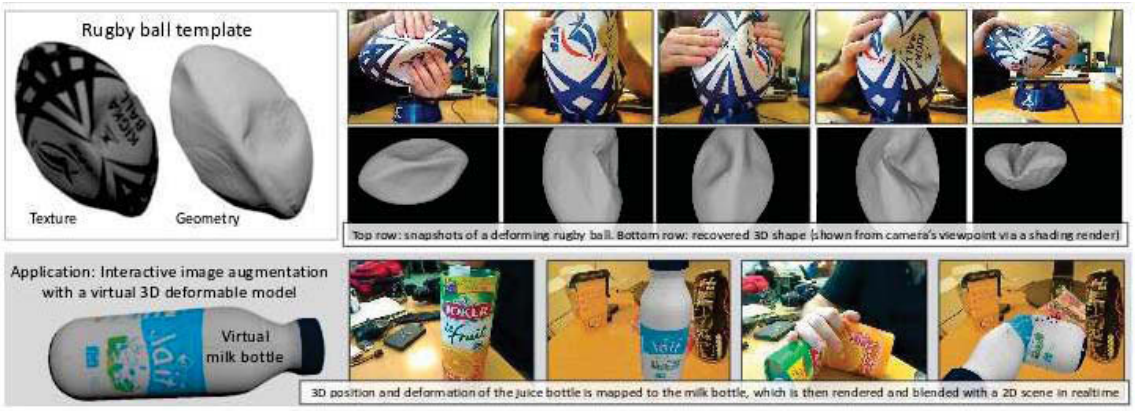
a) 3D reconstruction of rigid objects



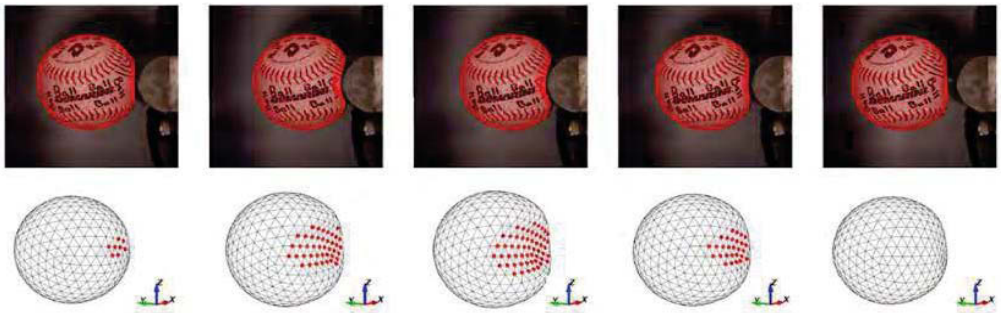
b) 3D reconstruction of non-rigid objects

Figure 1.1: 3D reconstruction methods. For rigid objects, SfM is a widely used method (Images taken from [Snavely et al., 2007]). The 3D reconstruction of deformable objects can be performed by either NRSfM or SfT methods.

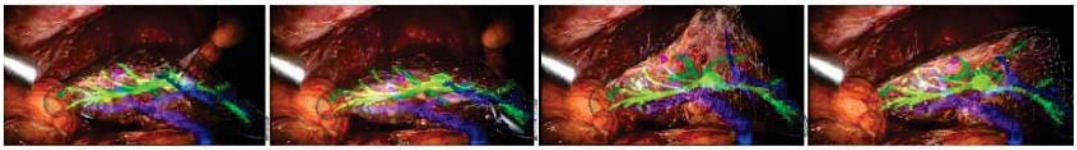
This thesis focuses on the monocular deformable 3D reconstruction methods that use motion as a visual cue in monocular imaging conditions. We now define the problem in detail and describe our contributions.



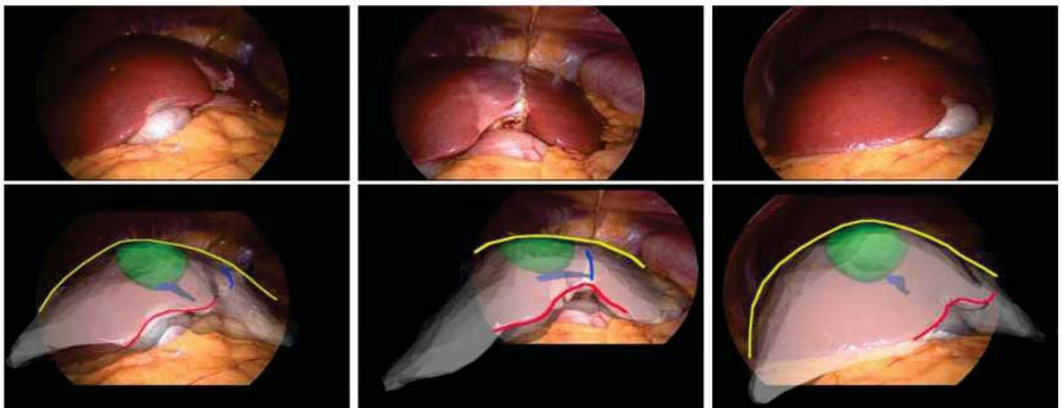
a) [Collins et al., 2015] shows an application for real-time SfT. The deformations of an object viewed in a single image can be transformed to other objects.



b) [Smith et al., 2016] shows an application for SfT. The reconstructed deformed ball can be used to study the impact of the ball on different surfaces. This can be used in sports industry to design and test equipments that are resistant to ball impact.



c) [Haouchine et al., 2013] shows an application for SfT in minimally invasive surgery. From the reconstructed surface, it estimates the deformation in depth and augments the liver (the wireframe) with a tumor (in purple), heptic vein (in blue) and portal vein (in purple).



d) [Koo et al., 2017] shows an application for SfT in minimally invasive surgery. It deforms a given model of liver and aligns it with the image obtained from laparoscopic camera. It uses additional cues of shading and contour for alignment. The tumor and vein are shown in green and blue respectively.

Figure 1.2: Some applications of the deformable 3D reconstruction methods.

1.2 3D Reconstruction of Deformable Objects

As mentioned earlier, the 3D reconstruction of rigid objects by SfM cannot be directly extended to deformable objects. As these objects may undergo deformations, the inter-image visual motion (strictly induced by the change in camera coordinate frame in the case of SfM) is now dependent on both camera motion and object deformation. Exploiting this visual motion (coupled with deformations) becomes a challenging task as the constraints are weaker in this case.

The goal of this thesis is to propose a general framework for modelling and solving deformable 3D reconstruction. At this point, we classify deformable objects as thin-shell (objects with an infinitesimal thickness, such as a piece of cloth, paper, etc.) and volumetric objects (objects with non-negligible thickness such as a sponge, cushion, etc.). Volumetric objects can be considered as thin-shell objects in cases they are represented by their outer-shells only but at the price of losing inner constraints. We now discuss the two categories of deformable 3D reconstruction problems that arise in computer vision.

1.2.1 Shape-from-Template

SfT (see figure 1.1b) is the generic name for a set of methods which perform the monocular 3D reconstruction of deformable objects using a 3D template of the object. It is also called template-based (or model-based) reconstruction in the literature. The inputs of SfT are a single image and the object's template, and its output is the object's deformed shape. The template (sometimes also called model) is a very strong object-specific prior as it includes a reference shape, a texture-map and a deformation model. Some SfT problems, such as the reconstruction of isometric thin-shell objects, have been extensively studied. Some of these methods are [Bartoli et al., 2015; Brunet et al., 2014; Chhatkuli et al., 2016b; Haouchine et al., 2014; Moreno-Noguer et al., 2009; Oswald et al., 2012; Perriollat et al., 2011; Salzmann and Fua, 2011; Vicente and Agapito., 2013]. SfT methods with real-time implementation are [Collins and Bartoli, 2015; Ngo et al., 2016].

Most of the SfT methods use the thin-shell isometric deformation model which implies that the geodesic distances between any two points on the object do not change due to the deformation. Isometry can be seen as local rigidity. Isometry is a very good approximation as most of the objects in nature undergo near-isometric deformations. Mathematically, it is also relatively easier to model isometry than other deformations. Our work focuses on isometry but we do explore other deformations as well and present a general modelling framework which makes it easier (and practical) to express various deformations.

1.2.2 Non-Rigid Structure-from-Motion

NRSfM (see figure 1.1b) is the generic name for a set of methods which perform the monocular 3D reconstruction of deformable objects from multiple images only. It is also called template-free (or model-free) reconstruction in the literature. The inputs to NRSfM are multiple images

and its output is the object's 3D shape for every image. In NRSfM, the rigidity constraint of SfM is replaced by constraints on the object's shape and deformation model. NRSfM methods were proposed initially with the low-rank shape basis [Bregler et al., 2000], the trajectory basis [Akhter et al., 2009], isometry [Chhatkuli et al., 2014; Varol et al., 2009; Vicente and Agapito, 2012], inextensibility [Chhatkuli et al., 2016a] and elasticity [Agudo et al., 2016]. Existing methods suffer from one or several limitations amongst solution ambiguities, low accuracy, ill-posedness, inability to handle missing data and high computation cost. NRSfM thus still exists as an open research problem.

Based on the modelling framework, existing NRSfM methods can be divided into two main categories: *1)* methods with statistics-based modelling and *2)* methods with physics-based modelling. physics-based modelling is the most recent. Most of the NRSfM methods use a statistics-based modelling. While statistics-based modelling does not take the object's nature into account, physics-based modelling is usually limited to thin-shell objects only.

1.2.3 Current Limitations

With this discussion, we want to emphasize the following limitations of existing SfT and NRSfM methods:

1) Methods with physics-based modelling are capable of handling more complex deformations than methods with statistics-based modelling and they proved to be very successful in SfT. However, physics-based modelling is seldom used in NRSfM.

2) Most of the existing thin-shell NRSfM methods work with the orthographic projection model which suffers from flip ambiguities. Therefore, the focus of the new techniques should be towards solving NRSfM using perspective projection as these solutions are more accurate.

3) Most of the methods deal with isometry or near-isometry which is relatively easier to model. Other deformations have been less explored.

4) Volumetric SfT has not been explored yet. There are some methods that recover the closed thin-shell of the object but a complete 3D reconstruction of a deformable object has not been achieved yet.

Now we discuss our contributions to SfT and NRSfM in detail.

1.3 Contributions

This thesis has three main contributions. Our first contribution is about solving NRSfM with the use of Riemannian geometry. This is a local formulation which means that the points on a surface can be reconstructed independently. The particles on an object are attached to each other and therefore the force causing deformation acts globally. However, the impact of the force is not necessarily uniform throughout the object which makes it interesting to study the deformations locally. This local formulation using Riemannian Geometry allows NRSfM to be expressed in terms of polynomial expressions whose variables are independent of the number of images under consideration. Our second contribution is about proposing a

modelling framework for NRSfM and SfT (using differential geometry) which is general and makes it convenient to handle various kinds of deformations. These solutions are obtained in terms of the differential or local quantities expressed at a surface, as a set of Partial Differential Equations (PDE) that hold at each point on the surface. In order to solve the PDE, we convert them to algebraic expressions by replacing the differentials in the PDE with algebraic variables. Given enough constraints, these algebraic equations yield a local solution. This solution can be obtained independently for each point. However, it may not always be possible to find such a solution. We discuss such conditions. The third contribution is a solution to SfT for volumetric objects.

To sum up, this thesis contributes in finding an answer to the following questions:

1) Is it possible to the extend the differential physics-based modelling of SfT [Bartoli et al., 2015] for isometric deformations to NRSfM? Can we solve NRSfM locally from a PDE formulation?

2) Is it possible to the extend the local formulation of NRSfM to deformations other than isometry?

3) Can we reconstruct the entire volume of a deformable object in a model-based scenario?

A fundamental assumption. Our framework relates the 3D shapes using the inter-image warps. These are the functions that register one image to another. Registering wide-baseline images can be accomplished by Scale Invariant Feature Transform (SIFT) [Lowe, 2004] which is a sparse-registration method. Dense or semi-dense registration can be achieved using SIFT Flow [Liu et al., 2011] and DeepFlow [Weinzaepfel et al., 2013] respectively. In order to register short-baseline images (for example, images from a video sequence), optical flow methods can be used. It usually gives a dense-registration. Some of the efficient methods are [Brox et al., 2004; Garg et al., 2013b; Sundaram et al., 2010]. These methods yield a dense registration. The first and higher order derivatives of the registration can be computed from the keypoint correspondences (obtained from the previous methods) using [Bookstein, 1989; Pizarro et al., 2016]. We make the assumption that in SfT and in NRSfM, the image registration can be established by using existing methods. Nevertheless, we only need to find these warps locally at each point. We use the first and the second-order derivatives of the warps in the first two contributions while in the third we only need the first order derivatives. The second-order derivatives are usually noisy, we correct them using Schwarps [Pizarro et al., 2016]. We show that the use of Schwarps is theoretically justified as well.

Contribution 1: Non-Rigid Structure-from-Motion using Riemannian geometry. We present a solution to NRSfM using the thin-shell isometric deformation model, that we hereinafter denote as Iso-NRSfM. We model Iso-NRSfM using concepts from Riemannian geometry.

We model the object's 3D shape for each image as a Riemannian manifold and deformations as isometric mappings. We parametrise each manifold by embedding the corresponding retinal plane. This allows us to reason on advanced surface properties, namely the metric

tensor and the Christoffel Symbols (CS), directly in retinal coordinates, and in relationship to the warps. These metric properties allow us to express the differential properties of surfaces, such as length, which are to be preserved under isometric deformations.

We formulate Iso-NRSfM locally with five variables which are functions of the first and the second-order derivatives of the inverse-depth of the surface undergoing deformation. We write the metric tensor and the CS in terms of these variables. We prove two new theorems showing that for isometric deformations, the metric tensor and the CS may be transferred between views using only the warps. This limits the number of variables to only five for any number N of views.

First, we solved Iso-NRSfM in [Parashar et al., 2016] (see figure 1.3) by assuming that the surface is planar in the infinitesimal neighbourhood of each point. This is the assumption of Infinitesimal Planarity (IP) which lets us get rid of the second-order derivatives in the expression of the CS. This limits the variables to only two. These variables correspond to the ratio of first-order derivatives of the inverse-depth function to the inverse-depth function. We obtained a system of two cubics in two variables that involve the first and the second-order derivatives of the warps. This system holds at each point on the surface.

Then, we extended the solution to Iso-NRSfM without the assumption of IP. Our solution is obtained in two steps. *1)* We solve for the first-order derivatives assuming that the second-order derivatives are known. This is initialised using the solution with IP. *2)* We solve for the second-order derivatives with the first-order derivatives obtained in the previous step. We obtain a system of $4N-4$ linear equations in three variables which is solved using Linear Least Squares (LLS). We iterate these two steps until the first-order derivatives converge. The solution gives an estimate of the metric tensor field, and thus of the surface's normal field, in all views. The shape is finally recovered by integrating the normal field for each view. The proposed method has the following features. *1)* It has a linear complexity in

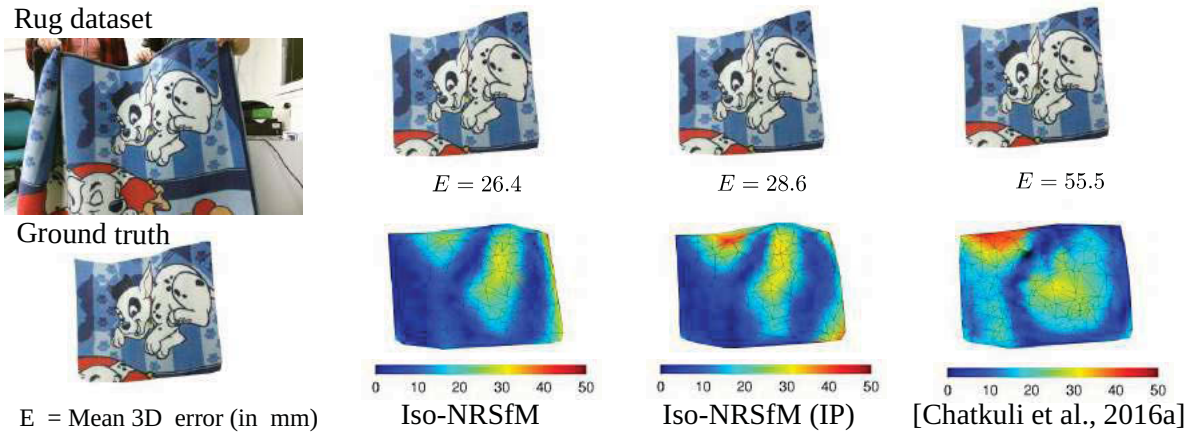


Figure 1.3: Comparison of Iso-NRSfM (with and without IP) with [Chhatkuli et al., 2016a].

the number of views and number of points. *2)* It uses a well-posed point-wise solution from $N \geq 3$ views, thus covering the minimum data case. *3)* It naturally handles missing data created by occlusions. *4)* It substantially outperforms existing methods in terms of speed

and accuracy, as we experimentally verified using synthetic and real datasets.

Contribution 2: a unified framework for the 3D reconstruction of deformable objects using Cartan's connections. Unlike SfM which is modelled using algebraic projective geometry, there is no consensus on the modelling framework of NRSfM yet. We present a modelling framework for NRSfM using the differential geometry of surfaces. In mathematics, differential geometry is the basis to study the properties of curves and surfaces. Recently, [Fabbri and Kimia, 2016] proposed to solve SfM using the differential geometry of 3D curves. [Fabbri, 2010] proposed pose estimation and camera calibration using differential geometry of 3D curves. However, it is not widely used in modelling surfaces for deformable 3D reconstruction. Recently, [Bartoli et al., 2015] proposed solutions to SfT using differential geometry. These solutions are analytic and therefore, they are very fast and need not be initialised. The success of our first contribution where we solve NRSfM with Riemannian geometry (which is a special case of differential geometry) is a motivation for us to use differential geometry to propose a general framework to model NRSfM and SfT. Riemannian geometry is limited to isometric (geodesic-distances preserving) and conformal (angles preserving) deformations whereas differential geometry is more general and can model a wider range of deformations.

This framework can therefore handle a wide variety of deformation models (including isometry) in a convenient way and therefore is a practical approach towards NRSfM. We model surfaces as smooth manifolds [Lee, 2003] and extract differential properties of the surfaces using differential geometry. Our work is essentially based on Cartan's theory of connections [Cartan, 1923, 1924, 1926] devised using the differential geometry of smooth manifolds and the theory of moving frames [Cartan, 1937]. The connections were at first formalised as the entities that enable movement along the curves as a parallel transport i.e., the orientation of a vector on the curve or surface does not change when it moves in a closed curve. This is known as a Levi-Civita connection [Lee, 1997]. Cartan generalised the idea of connections as the entities that transport tangent plane vectors along the curve. Cartan's connections are not limited to parallel transport along the curves and therefore, they are more generic. In this thesis, we always work with Cartan's connections.

A moving frame is defined as a local frame of reference defined at a point on a surface (or a manifold). The differential properties of the surfaces such as lengths, angles and areas can be described using the moving frames. The connections are derived using the moving frame and its derivatives. They are related to the first, second and the third fundamental forms of the surfaces [O'Neill, 2006]. Cartan proved that connections are necessary and sufficient to study the properties of 3D surfaces. From these properties, we derive differential constraints on the surfaces that lead to a solution to NRSfM and SfT. We solve these constraints algebraically.

We use moving frames and connections to design a modelling framework for the study of thin-shell deformable objects. Our framework has the following characteristics.

1) Our framework relies on the Infinitesimal Linearity (IL) assumption [Kock, 2010]. Under this assumption, any smooth deformation may be considered to be linear in the in-

finitesimal neighbourhood of a point while globally it could still be non-linear. This allows us to express the moving frame and the connections in terms of two variables (the first-order derivatives of the inverse-depth) only.

2) We prove a theorem stating that connections on any two surfaces can be related to each other for any kind of smooth (IL) deformation they undergo. This allows the number of variables to be only two for any number of views used in the reconstruction.

3) We express the physical properties of surfaces (such as lengths, angles and areas) locally in terms of the moving frames. We express deformation constraints as the physical properties they preserve. For example, isometry preserves both lengths and angles. We express isometric deformation constraints as the preservation of lengths and angles defined using moving frames. We express constraints for other deformations such as conformal (angles made by any three points on a surface do not change under deformation) and equiareal (areas are preserved under deformations). We propose a deformation that is a combination of anisotropic scaling (along surfaces' frame-basis) and a conformal deformation. We call it the skewless deformation. We explain it further in chapter 5.

4) These physical properties are related by the image warps across surfaces.

This theoretical framework leads to local solutions to deformable 3D reconstruction in terms of PDE which we solve algebraically (see figure 1.4 for more details). This framework represents surfaces analytically. Therefore, it is very easy to change surface definition which makes it very easy for this framework to adapt for different representations. By ex-

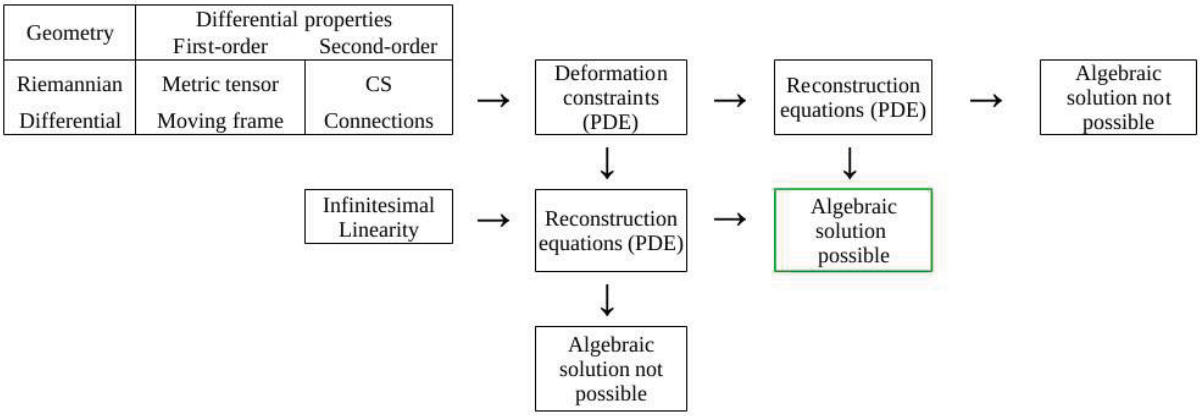


Figure 1.4: A broad overview of the problem. Moving frames and connections are the generalisation of the concepts of metric tensors and CS from Riemannian geometry. We use them to express differential constraints in terms of PDE. The manipulation of these constraints with or without the IL assumption leads to reconstruction equations that are also PDE. IL is not necessary to find these equations, however we use it to simplify the problem. These reconstruction equations may or may not have an algebraic solution. Here, an algebraic solution implies that the equations can be solved locally at each point. In this thesis, we solve the equations with possible algebraic (or local) solution.

pressing the deformation constraints in terms of the moving frame and using our theorem of transfer of connections, we present solutions to deformable 3D reconstruction for various deformations like isometry, conformity, skewless or equiareal. The solution to NRSfM under *a)* isometric/conformal deformations is given by solving a system of two cubics, *b)* skewless

deformations is given by solving a system of two septics (exploiting the first and second-order derivatives of the warps) in terms of two variables only.

We show that Iso-NRSfM (our first contribution) can be obtained using this framework as well. In this solution, we chose isometry to be solved as conformity as it makes the problem simpler to solve.

Our framework is directly extended to SfT. The existing solutions to isometric and conformal SfT [Bartoli et al., 2015] can be derived using this framework. We obtain an SfT solution to *a)* isometric/conformal and equiareal deformation as two linear expressions, *b)* skewless deformation as a system of two cubics in terms of two variables. These expressions exploit the first and the second-order derivatives of the warps.

Due to our theorem of transfer of connections across smooth surfaces using the inter-image warps, we propose a solution to SfT which is independent of the deformation constraints and only imposes deformation to be locally smooth. SfT has previously been solved under the assumption of smooth deformation in [Bartoli and Özgür, 2016]. We compare it with our results of SfT. [Salzmann et al., 2007] discussed that such a solution is not well-posed, however, we show that our solution to smooth deformations is well-posed. We discuss the reasons which make it well-posed.

Summing up, the proposed framework has the following features. *1)* It is a unified modelling framework for NRSfM (and SfT) using differential geometry assuming IL, which can be extended to various deformations. *2)* It brings the solutions to isometric/conformal and skewless NRSfM as a set of two cubic and septic polynomials respectively in terms of two variables for any number of views. *3)* It brings the solutions to isometric/conformal and equiareal SfT using a linear system of two equations in two variables only. *4)* It brings the solution to skewless SfT by solving a cubic system of two equations in two variables only. *5)* It also brings a solution to SfT for smooth deformations which is also a system of two linear equations in two variables only.

Contribution 3: Shape-from-Template for volumetric objects. As discussed earlier, existing SfT methods are thin-shell SfT essentially, as they are designed for thin objects such as a piece of paper. However, while thin-shell SfT handles thicker objects such as the woggle of figure 1.5 or a foam ball, it does not fully exploit the strong constraints induced by the object's non-empty interior.

We bring SfT one step further by introducing volumetric SfT, defined as an SfT method which uses a deformation constraint on the object's outer surface and interior. An example is shown in figure 1.5. Volumetric SfT reconstructs the object's interior deformation, which is not reconstructed by thin-shell SfT, and reconstructs the object's outer surface more accurately than thin-shell SfT thanks to the stronger deformation constraint it uses. Volumetric SfT is challenging as only the front part of the object's surface is visible in the image: the object's back surface and interior have to be inferred with no direct visual observations.

It is important to note that strictly speaking, isometry leads to rigidity in volumes. Only rigid volumetric objects can preserve geodesic-distances while undergoing deformation. We

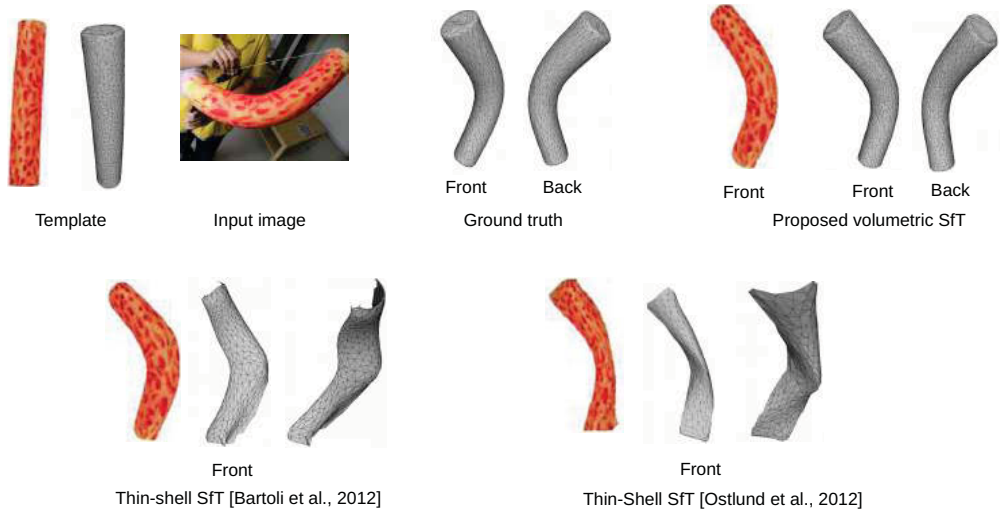


Figure 1.5: Volumetric SfT versus thin-shell SfT. Existing methods are thin-shell SfT. They use deformation constraints on the object's surface. For instance, [Bartoli et al., 2012] uses isometric constraints on the object's visible (front) surface and reconstructs the object partially, while [Östlund et al., 2012] uses isometric constraints on the object's whole closed outer surface and reconstructs it entirely. Volumetric SfT uses deformation constraints on the object's surface *and* interior. This greatly improves reconstruction accuracy and facilitates reconstruction of the object's interior. In this example, the thin-shell SfT methods [Bartoli et al., 2012; Östlund et al., 2012] reach a 3D error of 20 mm and 13 mm respectively on the visible surface, while the proposed volumetric SfT method reaches a 3D error of 7 mm. It reconstructs the non-visible (back) surface, for which no visual data is available, with a 3D error of 17 mm.

propose to instantiate volumetric SfT using the As-Rigid-As-Possible (ARAP) deformation model (a relaxation of isometry), which has been used extremely successfully in Computer Graphics [Sorkine and Alexa., 2007; Zhang et al., 2010]. The ARAP model maximises local rigidity while penalising stretching, sheering and compression. More specifically, ARAP has been widely used to perform mesh editing of animated characters [Zhou et al., 2005; Zollhöfer et al., 2012] because the resulting deformations locally preserve the object's structure.

Contrary to thin-shell SfT, volumetric SfT is largely unexplored. Recently, [Innmann et al., 2016] proposed a method that reconstructs the closed thin-shell surface of the deformable object in real-time. This method is named "VolumeDeform" however it does not reconstruct the interior of the object. The closest method to volumetric SfT is perhaps [Vicente and Agapito., 2013], where SfT has been combined with silhouette-based reconstruction. However this method requires stronger image cues, including silhouette and point correspondences, and recovers two-way ambiguous shape solutions. In contrast, we solve volumetric SfT without restricting the topology of the object and using the perspective camera. By using ARAP, our method preserves the object's interior structure while jointly reconstructing the deformation of the object's full outer surface and interior, as illustrated in figure 1.6. ARAP volumetric SfT involves solving a non-convex constrained variational optimisation problem. We discretise the object's volume and relax the constraints to convert the variational problem into an unconstrained non-linear least-squares optimisation problem. This problem can then be solved with standard numerical solvers such as Levenberg-Marquardt. We propose two

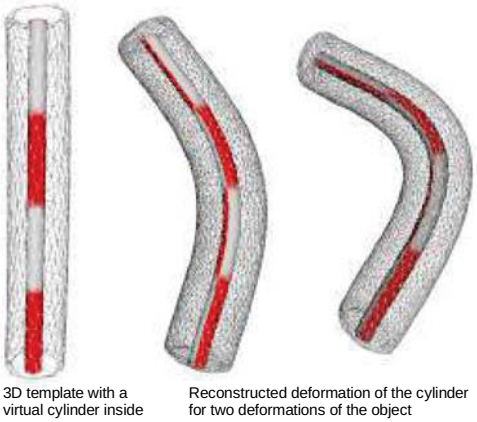


Figure 1.6: As opposed to thin-shell SfT, volumetric SfT reconstructs the object's interior deformation. In this example using the data from figure 1.5, a virtual cylinder is placed inside the woggle's template. It is then deformed using the deformation reconstructed by volumetric SfT to aid visualization of the object's reconstructed interior deformation. The second deformation is the one shown in figure 1.5.

initialisation methods. These methods use isometric thin-shell SfT and propagate the result through the object's volume.

The proposed contribution has the following features. *1)* We show that isometry in volumes is essentially a local rigidity or inextensibility constraint. *2)* We solve volumetric SfT in two steps: initialisation and refinement. *3)* We propose two methods for initialisation. *4)* We perform refinement in two ways: minimising L1 and L2 norms. *5)* Experimental results on synthetic and real data show that volumetric SfT improves accuracy to a large extent compared to state-of-the-art thin-shell SfT methods.

Thesis layout. We have divided this thesis into 7 chapters. We discuss the state of the art in chapter 2, mathematical preliminaries in chapter 3. Chapters 4, 5 and 6 give our first, second and third contributions. Chapter 7 presents our conclusions and perspectives for future work.

Chapter 2

Related Work

In this chapter we discuss the existing works on SfT and NRSfM that use motion as a visible cue in two sections. We sub-categorise these methods based on the constraints they use. Most of these methods are designed for thin-shell objects but we also discuss the works that are related to volumetric objects.

2.1 Shape-from-Template

The SfT methods were introduced much later than NRSfM but they evolved quickly. Now there are stable and real-time SfT methods [Collins and Bartoli, 2015; Ngo et al., 2016]. In general, SfT uses a 3D template of a thin-shell object. This is a very strong prior and makes SfT a better-posed problem than NRSfM. We classify current SfT methods into two categories: initialisation and refinement methods. The initialisation methods are the ones which achieve a fast solution to SfT using deformation constraints. They do not employ a heavy non-convex optimisation to minimise a cost which consists of a set of constraints such as deformation, smoothness or reprojection which is the case with refinement methods. Refinement methods are computationally expensive but more accurate. However, they need to be initialised. The performance of these methods depends on the accuracy of initialisation. A good initialisation can significantly reduce their computation time. We recall that the current SfT solutions are for thin-shell objects only, SfT for volumetric objects has not been proposed however, we discuss some of the works that use non thin-shell models.

Most of the SfT methods exploit physics-based modelling. Most of them use isometry as a physical prior but there are some solutions that use elasticity [Haouchine et al., 2014; Malti et al., 2013], the particle model [Özgür and Bartoli, 2016] or smoothness [Bartoli and Özgür, 2016]. We organise these two categories of initialisation and refinement into methods that employ isometric and non-isometric constraints. We now discuss these two categories of methods.

2.1.1 Thin-Shell Initialisation Methods

2.1.1.1 Isometric Constraints

Most of the initialisation methods use isometry as a deformation prior. Isometry is a physical prior on deformation which can be seen as local rigidity. Isometry preserves the geodesic distances between points on a surface undergoing deformations. Therefore stretching or shrinking of the surfaces is not allowed. Inextensibility is a relaxation of isometry. It means that the Euclidean distances between the neighbouring points on the deformed surfaces are always lower than or equal to the corresponding geodesic distances on the original surface. Figure 5.1 shows two surfaces S_1 and S_2 related by a deformation. The isometric and inex-

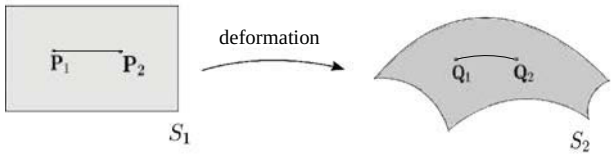


Figure 2.1: A surface S_1 transforms to surface S_2 due to a deformation. The points (P_1, P_2) on S_1 transform to (Q_1, Q_2) on S_2.

tensibility constraints on the points (P_1, P_2) on S_1 and (Q_1, Q_2) on S_2 in terms of distances between the points can be written as

$$\begin{aligned} \|Q_2 - Q_1\|_g &= \|P_2 - P_1\|_g \quad \text{isometry constraint} \\ \|Q_2 - Q_1\|_2 &\leq \|P_2 - P_1\|_g \quad \text{inextensibility constraint} \end{aligned} \tag{2.1}$$

where $\|.\|_2$ represents the Euclidean distance and $\|.\|_g$ represents the geodesic distance between two points on a surface. Therefore, we can see that intextensibility is a relaxed form of isometry. However, expressing geodesics on an arbitrary surface is not easy. Therefore, most of the methods approximate the geodesics with euclidean distances by assuming that the points are very close to each other. For example, the geodesic of (Q_1, Q_2) on S_2 can be written as the Euclidean distance between them given that Q_2 is close enough to Q_1. The sense of closeness or neighbourhood of these points are defined by the methods. Inextensibility needs to be combined with maximum depth in order to prevent the reconstructed surface from shrinking. This is called as Maximum Depth Heuristics (MDH).

We now discuss the initialisation methods that use inextensibilty and isometry constraints in two sections.

2.1.1.2 Inextensibility Constraints

[Perriollat et al., 2011] was the first method to model isometry using the inextensibility constraint (2.1). It is based on the MDH. It finds a solution to SfT by maximising depth heuristically while imposing inextensibility constraints. Figure 2.2 shows two surfaces related with an isometric mapping. Consider Q_1 at a distance μ_1 from the camera. Assuming that Q_2 is a neighbouring point of Q_1, it can be parametrised with the angle α_{12} between their

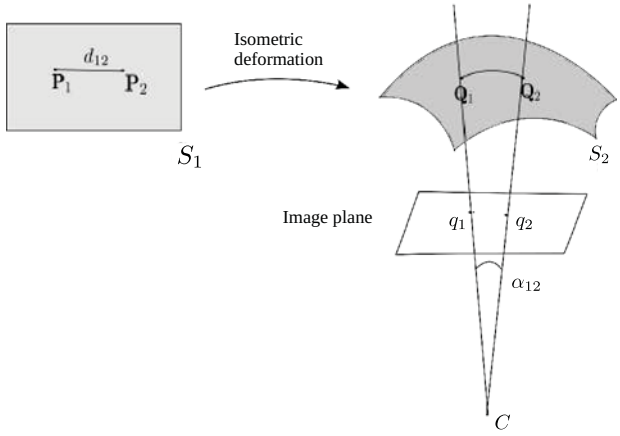


Figure 2.2: A surface S_1 transforms to surface S_2 due to an isometric deformation. The points (P_1, P_2) on S_1 transform to (Q_1, Q_2) on S_2. The sightlines of the two points (Q_1, Q_2) from the camera C pass through (q_1, q_2) on the image plane.

sightlines from the camera C. Therefore, we can write

$$Q_1 = \begin{bmatrix} \mu_1 \\ 0 \\ 0 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} \mu_2 \cos(\alpha_{12}) \\ \mu_2 \sin(\alpha_{12}) \\ 0 \end{bmatrix}. \tag{2.2}$$

Using this parametrisation of the points, the inextensibility constraint in equation (2.1) gives the upper bound of μ_1 as

$$\mu_1 \leq \frac{d_{12}}{\sin \alpha_{12}}, \tag{2.3}$$

where d_{12} is the geodesic distance between the 3D points P_1 and P_2 in the template. This is the upper bound on the depth of each point. It is chosen to be the minimum upper bound of all the neighbouring points such that the inextensibility constraint (2.1) is satisfied.

[Salzmann and Fua, 2011] made an improvement by modelling this problem as an Second-order Cone Program (SOCP) which can be globally solved using convex optimization. The method parametrises 3D points as the back-projection of 2D image points. Therefore any point Q_i can be expressed as

$$Q_i = z_i \begin{bmatrix} q_i \\ 1 \end{bmatrix}, \tag{2.4}$$

where z_i is the depth at the i^{th} point and q_i is the normalized image point. The idea is to maximise the sum of all the depths z_i such that the inextensibility constraint (2.1) is satisfied. This method uses a learned space of deformations using linear local models for small patches. This limits the applicability of this method to surfaces whose linear local models are known. However, it shows good performance when there is enough perspective in the images. [Brunet et al., 2014] proposed an initialization method based on inextensibility constraints solved using MDH. It used a parametric representation of surfaces using cubic B-splines [Dierckx,

1993] which reduces the dimensionality of the problem and provides a solution faster.

[Ngo et al., 2016] proposed a modified approach, where the method uses a laplacian smoothness prior along with inextensibility constraints. The laplacian of the template is calculated which is assumed to be preserved under the deformation. The laplacian is linearly parametrised and therefore, the problem can be solved using LLS. It is used as an initialisation method to the real-time solution to SfT proposed in [Ngo et al., 2015].

The above-mentioned methods use inextensibility constraints which is a relaxation on isometry and therefore, it is not a strict physical constraint. A more accurate representation of deformation constraints is possible by using differential modelling which we discuss next.

2.1.1.3 First-Order Differential Isometric Constraints

Recently, [Bartoli et al., 2015] proposed a local analytical solution to SfT using a warp and first-order differential isometric constraints. It shows that SfT is a well-posed problem for isometric deformations. It expresses the constraints in terms of first-order PDE and finds an algebraic solution to it. Since the method is analytical, it is very fast. However, it suffers with instabilities under near-affine conditions. [Chhatkuli et al., 2016b] proposed an improvement on these solutions to find an analytical stable solution to depth using the gradient of the depths which were otherwise discarded in [Bartoli et al., 2015].

The success of these SfT methods inspires us to extend the physics-based differential modelling of deformation to NRSfM as well. We use differential geometry to formulate the NRSfM and SfT problems in terms of PDE and we find algebraic solution to them.

2.1.1.4 Non-Isometric Constraints

[Bartoli and Özgür, 2016] proposed to solve SfT by using only smoothness as a deformation prior. The solution is unique and obtained by solving an LLS problem. It finds the solution to SfT using reprojection constraints and a smoothness constraint for a fixed scale. The problem with this method is that smoothness is a very weak constraint which can make this method unstable.

[Bartoli et al., 2015] also proposed an analytical solution to conformal SfT. This solution was also formulated using PDE and solved algebraically. Even though this method suffers form instabilities, it usually performs better or as well as isometric SfT [Bartoli et al., 2015]. Therefore, differential modelling proves to be a good tool for non-isometric deformations as well.

2.1.2 Thin-Shell Refinement Methods

These methods formulate global deformation constraints and solve them by using a non-convex optimisation. Therefore, these methods need to be initialised. The initialisation methods discussed in the previous section can be used to initialise these methods. In fact, initialisation methods should always be combined with refinement methods in order to get the best possible reconstruction. The accuracy of initialisation methods makes the refinement

significantly faster. [Chhatkuli et al., 2016b] showed results by initialising [Brunet et al., 2014] with their result. They achieved an almost real-time reconstruction. [Collins and Bartoli, 2015] made an improvement on this and achieved a real-time reconstruction. We now discuss the refinement methods that use isometric and non-isometric constraints in the next two sections.

2.1.2.1 Isometric Constraints

[Brunet et al., 2014] proposed the first solution to SfT which optimises a statistically optimal cost. The cost is composed of three errors: 3D back-projection, differential isometric constraints and smoothness. The 3D back-projection error ($E_{reprojection}$) accounts for the difference when the 3D points of the deformed shape project back to the corresponding input image points. The differential isometric constraint error ($E_{isometry}$) forces that isometry holds at infinitesimal level. It ensures that the deformed shape is isometric. The smoothness error (E_{smooth}) forces the solution to be smooth. This problem is non-convex and relies on iterative local optimization such as Levenberg-Marquardt which requires to be initialised and has a high computation time. The cost is written as

$$Cost = E_{reprojection} + l_{isometry}E_{isometry} + l_{smooth}E_{smooth}, \tag{2.5}$$

where the two parameters $l_{isometry}$ and l_{smooth} are weights to the isometric and smoothness constraints and need to be tuned. We use a similar cost in order to find a solution to volumetric SfT.

[Yu et al., 2015] introduced a temporal smoothness constraint in addition to the above mentioned constraints in order to improve the refinement.

2.1.2.2 Non-Isometric Constraints

[Malti et al., 2011; Ngo et al., 2015] model deformation as conformal and use the pixel intensity error instead of the reprojection error. [Ngo et al., 2015] is initialised with [Ngo et al., 2016] and handles occlusions and poorly textured surfaces.

[Özgür and Bartoli, 2016] proposed a solution to SfT by expressing the object as a set of particles where deformation acts as a set of forces on it. It uses deformation and reprojection constraints and finds a solution by evolving particles to achieve a global equilibrium due to the action of various forces (including gravity). It uses boundary points to fix the solution.

[Haouchine et al., 2014; Malti et al., 2013] proposed a solution to SfT for extensible surfaces by modelling deformation with elasticity. The idea is to minimise the stretching energy such that the reprojection constraint and boundary points are satisfied.

SfT methods using non-isometric constraints are mostly solved using non-convex optimisation.

2.1.3 Shape-from-Template for Volumetric Objects

Volumetric objects have non-zero thickness. SfT methods using elasticity [Haouchine et al., 2014; Malti et al., 2013] require the surface model to include thickness, which must however be 'small' so that extension along normal direction may be neglected. In continuum mechanics, this means that the thickness is at least ten times smaller than the object's largest dimension. Therefore, these methods are categorised as thin-shell methods. They require one to provide the Young modulus of the object's material and, more importantly, boundary conditions expressed as a set of known 3D point coordinates, which may restrict their applicability.

A related goal was pursued in [Vicente and Agapito., 2013] where a silhouette-based method was combined with SfT. The template is also reconstructed from a reference image using a silhouette-based method inspired from [Oswald et al., 2012]. This method reconstructs objects that have a plane of symmetry parallel to the image plane and does not infer concavities, which is also a limitation of most silhouette-based methods [Oswald et al., 2012; Prasad et al., 2006]. The template is then deformed using a data term based on silhouette, area and orthographic reprojection constraints. The deformation model extends thin-shell isometry by placing virtual nodes in the object's interior, with the objective of preserving the object's volume.

2.1.4 Relationship to our Work

In chapter 4, we propose a modelling framework for SfT using differential geometry. This framework is coherent with methods based on differential modelling [Bartoli et al., 2015] and is general, therefore, other deformation models can be used. We also propose a solution to SfT assuming that the deformation is smooth. This means that SfT can be solved analytically for any kind of deformation without explicitly modelling the deformations.

In chapter 5, we propose volumetric SfT which, in contrast to thin-shell SfT, recovers the deformation of the object's outer surface and interior. It formulates the deformation globally in terms of a cost function and minimises it using non-convex optimisation.

2.2 Non-Rigid Structure-from-Motion

The first solution to NRSfM for thin-shell objects was proposed in [Bregler et al., 2000] which modelled deformation using a low-rank shape-basis. It assumes that the shape of an object can be represented as a linear combination of a low-dimensional shape-basis. We discuss the two categories of NRSfM methods based on the modelling framework: statistics-based modelling and physics-based modelling.

2.2.1 Methods with Statistics-based Models

Starting from the work of [Bregler et al., 2000], the low-rank shape-basis has been the most commonly used shape prior in NRSfM. It is a statistical prior on a set of 3D point correspondences expressed in terms of point correspondences in images that forces the matrix containing these correspondences to have a fixed low-rank. This matrix can be further decomposed into a shape-basis and their weights. Inspired from this method, [Akhter et al., 2009] proposed NRSfM which modelled deformation as a set of trajectory basis. We discuss statistics-based methods under these two categories.

2.2.1.1 Low-Rank Shape-Basis

For N images, the image observation matrix consisting of the P matched point correspondences across the images is written as

$$W = \begin{bmatrix} u_1^1 & \dots & u_P^1 \\ v_1^1 & \dots & v_P^1 \\ \vdots & \ddots & \vdots \\ u_1^N & \dots & u_P^N \\ v_1^N & \dots & v_P^N \end{bmatrix}, \tag{2.6}$$

where (u_i^j, v_i^j) represents the image coordinates of the i^{th} point on the j^{th} image. Any shape can be written as a linear combinations of the K shape-basis B_i. Therefore the shape S^i of the i^{th} image can be written as

$$S^i = \sum_{t=1}^{K} l_t^i B_t, \tag{2.7}$$

where each shape-basis B_t is $3 \times P$ matrix and l_t^i is the set of weights that decide the scale. Projecting these shape-basis on images under a scaled orthographic projection, we can write the observation matrix as

$$W = \begin{bmatrix} u_1^1 & \dots & u_P^1 \\ v_1^1 & \dots & v_P^1 \\ \vdots & \ddots & \vdots \\ u_1^N & \dots & u_P^N \\ v_1^N & \dots & v_P^N \end{bmatrix} = \begin{bmatrix} l_1^1 R^1 & \dots & l_K^1 R^1 \\ \vdots & \ddots & \vdots \\ l_1^N R^N & \dots & l_K^N R^N \end{bmatrix} \begin{bmatrix} B_1 \\ \vdots \\ B_k \end{bmatrix} = QB, \tag{2.8}$$

where R^i consists of the first two rows of the camera projection matrix. The goal is to decompose W into two matrices Q and B which contain the information of the scale and the set of shape-basis respectively. Once Q and B are found any shape can be written using equation (2.7). Q can be further decomposed in order to find the pose.

[Bregler et al., 2000] used Singular Value Decomposition (SVD) to decompose W into Q and B by fixing K as a low positive integer. Q and B are called the coefficient matrix and shape-basis matrix respectively. This problem is non-convex and suffers from ambiguities in the solution to the shape-basis. [Del Bue et al., 2004] proposed a non-linear refinement to improve the solution. However in order to deal with these ambiguities, different kinds of priors were proposed by various methods:

1) [Del Bue, 2008] proposed to use shape-basis priors to constrain the B matrix. The idea is to estimate the shape-basis for some known 3D shapes and use them along with the unknown shape basis in order to estimate the shapes for all the images. This means that some of the basis in B are already calculated using few known 3D shapes and W is decomposed in a way that these known shape bases do not change. Given that the first b elements of B are known, the shape prior L according to [Del Bue, 2008] is given by

$$L = N \begin{bmatrix} B_1 \\ \vdots \\ B_b \end{bmatrix} = NB. \tag{2.9}$$

The joint decomposition of W and L into Q, N and B improves the conditioning of the shape. [Tao and Matuszewski, 2013] extended this idea to allow shapes to have non-linear deformations by allowing L to be non-linear. B is found from a manifold whose embeddings are learnt from a representative training dataset of the deformable object under consideration. This is not a traditional NRSfM method (as it heavily relies on training to find B), however it does highlight the success of manifold learning.

2) [Torresani et al., 2001] solves this problem by adding a shape regulariser term that imposed spatial smoothness in the observed shape using an iterative optimisation.

3) [Olsen and Bartoli, 2008] solves this problem by optimising the shape-basis using spatial and temporal smoothness priors.

4) [Bartoli et al., 2008] uses deformation modes and closeness of points in the mean shape as priors. It solves the ambiguity by using a low-rank coarse-to-fine shape model which prioritises the deformation modes that give the coarsest deformations.

5) [Fayad et al., 2010] proposed to model the deformations (patch-wise) with quadratic models where the linear modes allow shearing and stretching, quadratic modes allow bending and the mixed modes allow the twisting of the surfaces. These modes are optimised for each (overlapping) patch using the temporal smoothness priors. Then they are stitched together to obtain a global surface.

5) [Akhter et al., 2009] shows that an ambiguity in the SVD does not necessarily lead to an ambiguous reconstruction. They introduced a correction matrix that can be used to obtain

a unique solution. They showed that the camera orthonormality conditions are sufficient to find the correction matrix. [Dai et al., 2012] proposed a more efficient solution to find this matrix by minimising the trace of the shape-basis. Trace-minimisation is a tighter constraint than rank-minimisation (used in [Akhter et al., 2009]). [Garg et al., 2013a] also used trace minimisation of the correction matrix in order to find the low-rank shapes. They formulated the problem as a global variational energy minimisation problem. The goal is to minimise the trace of the shape-basis, reprojection error and the total spatial variation. This method performs dense reconstruction and yields good results for faces. However, this algorithm is computationally very expensive.

Most of these methods express shapes as a linear combinations of shape-basis. This forces the deformations to be linear. Therefore these methods are applicable to simple deformations such as a talking face. They do not cope up with the larger deformations such as a walking man or a tree moving due to the wind unless more constraints are provided.

2.2.1.2 Low-Rank Trajectory-Basis

[Akhter et al., 2009] proposed to replace the shape-basis by a trajectory-basis in the formulation suggested by [Bregler et al., 2000]. This method can easily reconstruct larger deformations than the above-mentioned methods. The fundamental idea behind this method is to express the trajectory of a point on an image as a linear combination of the trajectory-basis. The trajectory-basis are obtained using a Discrete Cosine Transformation (DCT) basis. [Akhter et al., 2009] proposed to decompose the image observation matrix W in equation (2.6) into R and T using SVD, where R contains the R^i (the first two rows of camera projection matrix for the i^{th} image) and T is the trajectory-basis matrix.

[Gotardo and Martinez, 2011] proposed a solution that uses additional higher frequencies of the DCT basis in order to estimate large deformations better than [Akhter et al., 2009]. This method first estimates the 3D using trajectory-basis. It then estimates the shape-basis by applying a kernel transformation which generalises the inner product with a radial basis function.

[Torresani et al., 2008] also uses a parametric representation of shape and trajectory-basis and finds 3D by estimating these parameters using Probabilistic Principal Component Analysis.

These methods can handle large deformations better than the low-rank shape model methods but they still need video-sequences or short-baseline data to achieve good results.

Table 2.1 summarises the statistics-based NRSfM methods. All these methods use the orthographic projection model. All these methods solve NRSfM with a non-convex formulation except [Dai et al., 2012] which uses a convex relaxation.

Most of the statistics-based methods use orthographic camera projection which may lead to flip ambiguities in the reconstruction. Also, these methods are designed for short-baseline images and therefore, they cannot handle large deformations. For good results, these methods need a large number of views.

Method	Type of basis	Additional priors	Complexity of basis
[Bregler et al., 2000]	Shape	-	Linear
[Del Bue, 2008]	Shape	Shape	Linear
[Tao and Matuszewski, 2013]	Shape	Shape	Non-linear
[Dai et al., 2012]	Shape	-	Linear
[Torresani et al., 2001]	Shape	Spatial smoothness	Linear
[Olsen and Bartoli, 2008]	Shape	Spatio-temporal smoothness	Linear
[Bartoli et al., 2008]	Shape	Deformation modes, point closeness	Linear
[Akhter et al., 2009]	Trajectory	-	Linear
[Torresani et al., 2008]	Shape, trajectory	Spatio-temporal smoothness	Linear
[Gotardo and Martinez, 2011]	Shape, trajectory	-	Non-Linear

Table 2.1: Summary of statistics-based NRSfM methods

2.2.2 Methods with Physics-based Models

For a long time, the focus of the research community has been on the statistics-based modelling of deformation. Physics-based modelling for NRSfM is rather recent. In general, these methods use the physical properties of thin-shell objects to model deformation as in SfT. They can handle more complex deformations and work with fewer images than methods with statistics-based modelling. Most of these methods, for example [Chhatkuli et al., 2014, 2016a; Collins and Bartoli, 2010; Russell et al., 2014; Taylor et al., 2010; Varol et al., 2009; Vicente and Agapito, 2012] use isometry as a deformation model except [Agudo et al., 2016] which models deformations using elasticity. [Agudo et al., 2016] first reconstructs the surface by assuming that there are no deformation acting on it (basically SfM) and then uses this solution the predict the deforming shapes. Therefore, this method resembles SfT even though it is presented as a NRSfM method by the authors.

[Collins and Bartoli, 2010; Taylor et al., 2010] approximate isometry with a rigid rotation and translation at a local (or piecewise) level. For example, [Collins and Bartoli, 2010; Taylor et al., 2010] solved NRSfM by expressing isometry as local rigidity with an orthographic camera projection. [Taylor et al., 2010] finds sets of 3 rigid points reconstructed using SfM whereas [Collins and Bartoli, 2010] performs automatic clustering of point sets. These methods rely on finding the 3 points in a close neighbourhood in order to make sure that the assumption of rigid transformation holds. Another approximation was made by [Russell et al., 2014; Varol et al., 2009], they exploited isometry as piecewise-rigidity. [Russell et al., 2014] computes fundamental matrices [Hartley and Zisserman, 2000] to obtain the solution to surface normals. However, fundamental matrices may be unstable in case of small patches.

An improvement on [Varol et al., 2009] was proposed by [Chhatkuli et al., 2014] which defines isometric constraints between points that are infinitesimally close to each other while [Varol et al., 2009] defines these constraints on a small patch (assuming piecewise-rigidity). Both of them assume perspective projection. [Chhatkuli et al., 2014] assumes the surface to be a set of infinitesimal planes while [Varol et al., 2009] assumes the surfaces to be represented as a set of planar patches. They then obtain a homography between the corresponding normalised image points of the planes and use homography decomposition [Malis and Vargas, 2007] to obtain the surface normals. The surface normals thus obtained have

a two-fold ambiguity which is resolved by using spatial smoothness in [Varol et al., 2009] while [Chhatkuli et al., 2014] uses additional views to disambiguate the normals. The surface normals thus obtained are integrated to obtain an up-to-scale representation of the surface.

[Vicente and Agapito, 2012] expressed the deformation using isometric constraint (see equation (2.1)) and proposed a global solution to NRSfM by performing a discrete non-convex optimisation based on energy minimization of isometry constraints of all the points considered. They provided solutions for both orthographic and perspective projection. The method does not yield a globally optimal solution. The solution assuming orthographic projection suffers from flip ambiguities.

Recently, [Chhatkuli et al., 2016a] proposed a solution to NRSfM by modelling deformation with the inextensibility constraint (2.1) using the perspective camera model. Relaxing isometry to inextensibility makes the problem convex and a globally optimal solution is obtained using second-order cone programming. The problem is formulated using the MDH [Perriollat et al., 2011] where the goal is to maximise the point depth for each image correspondence in the retinal frame under the inextensibility constraints. The point depths are bounded by the sum of the unknown template distances in order to make sure that a global minimum is reached.

[Agudo et al., 2014] solves NRSfM using a mix of physics-based deformation prior and statistics-based priors. [Agudo et al., 2016] solves NRSfM for potentially extensible surfaces. It requires the surface to be represented as a thin-plate (a surface with considerable thickness), however it does not reconstruct the volumes. It represents the object's mechanics in terms of in-plane stress and out of plane bending. It models the deformation using Navier's equation which are solved by using Finite Element Method. However, this method works only for video sequences. [Agudo et al., 2014, 2016] require an initialisation which is obtained by using SfM on the first few frames. Therefore these methods resemble SfT in their approach rather than NRSfM.

Table 2.2 summarises the NRSfM methods using physical priors. The use of physical priors in NRSfM is limited to isometry. Other deformation priors are unexplored. Most of these methods are applicable to both short-baseline and wide-baseline data except [Chhatkuli et al., 2014, 2016a] which break on short-baseline data.

NRSfM methods based on physics-based modelling of deformations (for example, isometry can be modelled with a rotation and translation at a local level, which incorporates camera's rotation and translation as well) do not decouple camera motion and deformation unlike the NRSfM methods based on statistics-based modelling. The motion of the object or the camera are both treated as a deformation and hence, camera-motion is not recoverable. A deformation is therefore regarded as the change in the object from one view to another.

2.2.3 Relationship to our Work

Most of the current NRSfM methods suffer from ambiguities and poor performance in reconstruction. Deformations other than isometry are very rarely explored. The success of

Table 2.2: Summary of physics-based NRSfM methods

Method	Physical prior	Constraint	Camera model
[Chhatkuli et al., 2014]	Local rigidity	Local	Perpective
[Varol et al., 2009]	Local rigidity	Piecewise	Perspective
[Russell et al., 2014]	Local rigidity	Piecewise	Orthographic
[Taylor et al., 2010]	Local rigidity	Local	Orthographic
[Collins and Bartoli, 2010]	Local rigidity	Local	Orthographic
[Vicente and Agapito, 2012]	Isometry	Global	Orthographic
[Chhatkuli et al., 2016a]	Inextensibility	Global	Perspective
[Agudo et al., 2016]	Elasticity	Global	Perspective

physics-based methods in SfT inspires us to explore NRSfM with the same modelling. We use differential geometry and Cartan's theory of connections to model this problem and propose a modelling framework that extends NRSfM to deformations other than isometry.

Chapter 3

Mathematical Formulation

3.1 Notation

We define a set of rules that we follow in the next chapters in order to make them easier to understand. We describe here the general rules and the exceptions we make.

1) We use small-case Latin letters to denote scalars. *Exception:* In chapter 6, we use ρ and α to denote scalars.

2) Bold Latin letters denote 2D and 3D vectors.

3) We use Greek letters to denote functions. We express the inverse-depth function in chapters 4, 5 as β (in the case of planar surfaces) and α (in the case of non-planar surfaces). *Exception:* In chapters 4 and 5, we use $\mathbf{\Gamma}$ to express CS and components of a connection respectively. In chapter 5, $\mathbf{f}$ is a multi-valued smooth function defined in $\mathbb{R}^n$ with the coordinates f^i and their respective basis as e_i where $i \in [1, ..., n]$.

4) We use a subscript to index the images and a superscript to index the coordinates of a point.

5) We use calligraphic letters for objects and images. In chapter 6, we use calligraphic letters for sets, and $|\mathcal{A}|$ for the size of set $\mathcal{A}$.

6) We use the operator $\mathbf{J}_\varphi$ for the Jacobian of a function φ.

7) In chapter 4, we use $\mathbf{g}$ to denote the metric tensor and $\mathbf{\Gamma}$ to denote the CS matrix.

8) In chapter 5, differential 1-forms are represented as w. d is the operator for exterior differentiation. The origin is denoted as $\mathbf{O}$.

9) In chapters 4 and 5, we give our modelling for a pair of views. It straightforwardly generalises to any number of views. We consider two surfaces $\mathcal{M}_i$ and $\mathcal{M}_j$, which are represented by images $\mathcal{I}_i$ and $\mathcal{I}_j$. A point in $\mathcal{I}_i$ is denoted by $\mathbf{x}$ and the corresponding one in $\mathcal{I}_j$ by $\mathbf{y}$. We name the points this way to avoid the subscripts in the equations. Similarly, a point on the surface $\mathcal{M}_i$ is denoted by $\mathbf{z}$ and the corresponding point on $\mathcal{M}_j$ by $\mathbf{w}$.

10) In chapter 6, the 3D points on the template and object are given by $\mathbf{P}$ and $\mathbf{Q}$. Their corresponding points on the images are given by $\mathbf{p}$ and $\mathbf{q}$ respectively. A tetrahedron attached to a 3D point, for example, $\mathbf{P}$ is given as a set of 3D points $(\mathbf{P}_{n1}, \mathbf{P}_{n2}, \mathbf{P}_{n3}, \mathbf{P}_{n4})$.

11) In chapters 4 and 5 we omit $\mathbf{x}$ and just write the functions described at $\mathbf{x}$ as α_i, β_i and ϕ_i (instead of $\alpha_i(\mathbf{x})$, $\beta_i(\mathbf{x})$ and $\phi_i(\mathbf{x})$) in order to make the equations compact.

12) In chapters 4 and 5, we also write $\frac{\partial \alpha_i}{\partial x^t}$ as $\alpha_{i,t}$, $\frac{\partial \beta_i}{\partial x^t}$ as $\beta_{i,t}$, $\frac{\partial \phi_i}{\partial x^t}$ as $\phi_{i,t}$, $\frac{\partial^2 \alpha_i}{\partial x^t \partial x^s}$ as $\alpha_{i,ts}$, $\frac{\partial^2 \beta_i}{\partial x^t \partial x^s}$ as $\beta_{i,ts}$ and $\frac{\partial^2 \phi_i}{\partial x^t \partial x^s}$ as $\phi_{i,ts}$.

13) In chapters 4 and 5, we write $(k_1, k_2, k_3, k_4, k_5)$ as the expressions that represent the ratio of first and second-order derivatives of the inverse-depth of the surface $\mathcal{M}_i$ to the inverse-depth, (p_1, p_2, p_3) which are known quantities on $\mathcal{M}_i$, written in terms of the second-order derivatives and $(c_1, c_2, c_3, c_4, c_5, c_6)$ which represent the CS at $\mathcal{M}_i$. On $\mathcal{M}_j$, we write these expressions with a bar as $(\bar{k}_1, \bar{k}_2, \bar{k}_3, \bar{k}_4, \bar{k}_5)$, $(\bar{p}_1, \bar{p}_2, \bar{p}_3)$ and $(\bar{c}_1, \bar{c}_2, \bar{c}_3, \bar{c}_4, \bar{c}_5, \bar{c}_6)$.
We remind some of these notations in the chapters.

3.2 Manifolds and Surfaces

In general, a manifold is a topological space that resembles the Euclidean space $\mathbb{R}^n$ locally. Therefore, at each point of the manifold one can find a neighbourhood that is homeomorphic to the Euclidean space of dimension n. 2D manifolds represent surfaces. If embedded in 3D, they represent 3D surfaces.

3.2.1 Infinitesimal Planarity

IP refers to the assumption that a surface at each point is approximately planar in its infinitesimal neighbourhood. This is fundamentally different from piece-wise planarity: in IP, the surface is globally curved, but in an infinitesimal neighbourhood, it may be represented by a plane. In other words, each infinitesimal model agrees with the global surface at the point where IP is assumed, but this agreement holds only at the zeroth order. We define the IP approximation of any surface as the IP surface where at each point the infinitesimal plane is the tangent plane of the original surface. Note that while the surface can be globally curved, the IP approximation is point-wise planar. It is a very interesting concept as it makes it simpler to derive properties of surfaces.

3.2.2 Infinitesimal Linearity

IL refers to the assumption that a smooth mapping between two surfaces can be represented by a set of linear mappings which map the infinitesimal neighbourhoods of the corresponding points on the surfaces. Figure 3.1 shows two curves related by a smooth mapping ψ. According to the formulation of IL in synthetic differential geometry [Kock, 2010] (it uses IL to formalise theory of connections), given that ψ maps $\mathbf{P}$ to $\mathbf{Q}$, there exists at least one linear function ψ_L that maps the infinitesimal neighbourhood of $\mathbf{P}$ to $\mathbf{Q}$. Therefore, ψ is represented with an infinite set of linear mappings ψ_L that map infinitesimal neighbourhoods of the curves. ψ_L has the same first-order differentials as ψ. It only assumes that the second or higher-order differentials are zero as it is a linearization of ψ.

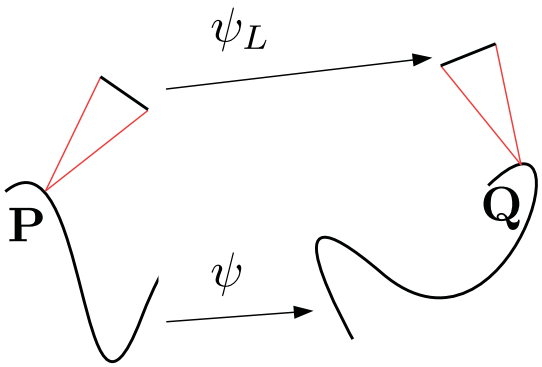


Figure 3.1: Illustration of IL. Two smooth curves are related by a mapping ψ. According to IL, there exists a linear map ψ_L that relates **P** and **Q** and agrees with ψ at zeroth and first-order.

3.3 Projection

A surface is mathematically related to an image with an image projection function. Figure 3.2 shows a surface $\mathcal{M} \in \mathbb{R}^3$ being projected into the image $\mathcal{I} \in \mathbb{R}^2$ with the function $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$. We model projection with the perspective camera, where Π takes as input the point $\mathbf{z} = \begin{pmatrix} z^1 & z^2 & z^3 \end{pmatrix}^\top$ on the surface $\mathcal{M}$ and outputs its retinal coordinates $\mathbf{x} = \begin{pmatrix} x^1 & x^2 \end{pmatrix}^\top$ in the image:

$$\mathbf{x} = \begin{pmatrix} x^1 & x^2 \end{pmatrix}^\top = \Pi(\mathbf{z}) = \begin{pmatrix} \frac{z^1}{z^3} & \frac{z^2}{z^3} \end{pmatrix}^\top. \tag{3.1}$$

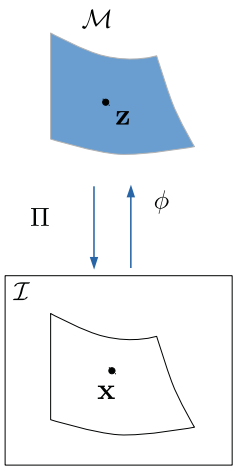


Figure 3.2: An image embedding ϕ that relates the 3D surface $\mathcal{M}$ with its image $\mathcal{I}$.

3.4 Image Embedding

The image embedding, denoted as $\phi : \mathcal{I} \to \mathcal{M}$ (see figure 3.2), represents the inverse of Π restricted to the surface $\mathcal{M} \in \mathbb{R}^3$, as it maps retinal coordinates to the 3D surface. It must satisfy the following identity:

$$\mathbf{x} = (\Pi \circ \phi)(\mathbf{x}). \tag{3.2}$$

Smooth functions that comply with equation (3.2) can be expressed with a depth function $\rho \in C^\infty(\mathcal{I}, \mathbb{R})$, where:

$$\phi(\mathbf{x}) = \rho(\mathbf{x}) \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top. \tag{3.3}$$

Alternatively, let $\alpha = \rho^{-1}$ be the inverse-depth function. This allows us to re-define the image embedding in equation (3.3) as:

$$\phi(\mathbf{x}) = \frac{1}{\alpha(\mathbf{x})} \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top. \tag{3.4}$$

where α is a function that represents the inverse of the depth of the surface at a point $\mathbf{x} = (x^1, x^2)$ in $\mathcal{I}$. A point on the surface $\mathcal{M}$ is given by

$$\mathbf{z} = \phi = \alpha^{-1} \begin{pmatrix} x^1 & x^2 & 1 \end{pmatrix}^\top. \tag{3.5}$$

For general surfaces, α is a non-linear function but for planar surfaces it is linear. Due to the assumption of IP, the restriction of α to a point becomes linear.

In chapter 4, we will show that working with the inverse-depth for defining the image embedding has an important role while defining the differential properties of surfaces.

Chapter 4

Non-Rigid Structure-from-Motion with Riemannian Geometry

Summary

In this chapter, we propose Isometric Non-Rigid Structure-from-Motion (Iso-NRSfM) using a theoretical framework based on the Riemmanian manifold to represent the unknown 3D surfaces as embeddings of the camera's retinal plane. This allows us to use the manifold's metric tensor and CS fields. These are expressed in terms of the inverse-depth of the 3D surfaces and its first and second-order derivatives. These are the unknowns for Iso-NRSfM. We prove that the metric tensor and the CS are related across images by simple rules depending only on the warps. We show that Iso-NRSfM is solvable from local image warps. It proves that NRSfM can be solved locally from a PDE formulation. We propose two solutions to Iso-NRSfM: with and without the assumption of IP. This chapter is based on our published work [Parashar et al., 2016].

4.1 Introduction

Research in NRSfM is still at an early stage. As we discussed in chapter 2, statistics-based modelling in NRSfM limits the applicability to short-baseline images only. Also, the use of orthographic camera projection leads to flip ambiguities. Physics-based modelling in NRSfM is very recent. It is widely used in SfT [Salzmann and Fua, 2011] and recent works [Bartoli et al., 2015; Chhatkuli et al., 2016b] show that the differential modelling of surfaces results in local analytical solution to SfT and makes is a well-posed problem for isometric deformations. Therefore, local SfT methods are powerful and computationally cheap.

The success of differential modelling in SfT is our motivation for this work. Our goal is to extend this modelling to NRSfM. We use Riemannian geometry for differential modelling of NRSfM. The differential quantities metric tensors and CS, defined in this geometry, represent the properties of surfaces such as length, angles, areas and curvature. For isometric deformations, we found that these properties are preserved across surfaces. This leads to a solution for NRSfM using isometry as a deformation prior which we present in this chapter.

This solution performs significantly better than the compared state-of-the-art methods. The method handles missing data and occlusions, needs very few images to perform reconstruction, handles large images conveniently without affecting the computation time and works for both short and wide-baseline images.

Chapter outline. We present the mathematical background of our solution in section 4.2. It describes the modelling of NRSfM, the concepts of metric tensor and CS, the effect of the IP assumption on these quantities, the preservation of these quantities under isometry. Section 4.3 shows how to use these concepts to write the reconstruction equations for NRSfM under isometric deformations with and without the assumption of IP. Section 4.4 explains the algorithms and analyses their computational complexity. Section 4.5 discusses the experiments and section 4.6 concludes.

4.2 Mathematical Background

4.2.1 General Model

Our model of NRSfM is shown in figure 4.1. We have N input images $\mathcal{I}_1, \ldots, \mathcal{I}_N$ that show the projection of different isometric deformations of the same surface. The registration warps (η_{ij} and η_{ji}) between the pair of images ($\mathcal{I}_i$, $\mathcal{I}_j$) are known. In this framework, we compute these warps using [Pizarro et al., 2016]. This choice is explained and justified by theorem 4. Abusing notation, we also use $\mathcal{I}_i$ to denote an image's retinal plane, with $\mathcal{I}_i \subset \mathbb{R}^2$. Surfaces in 3D are modeled as Riemannian manifolds. This allows us to define lengths, angles and tangent planes on the surface [Lee, 1997]. We denote $\mathcal{M}_i \subset \mathbb{R}^3$ as the i^{th} manifold, which can be seen as a two-dimensional subset embedded in 3D. We use the extrinsic definition of $\mathcal{M}_i$, where a function embeds a subset of the plane $\mathbb{R}^2$ into $\mathbb{R}^3$. With embedding functions, one can easily compute manifold characteristics [Lee, 2003] such as the metric tensor and the CS.

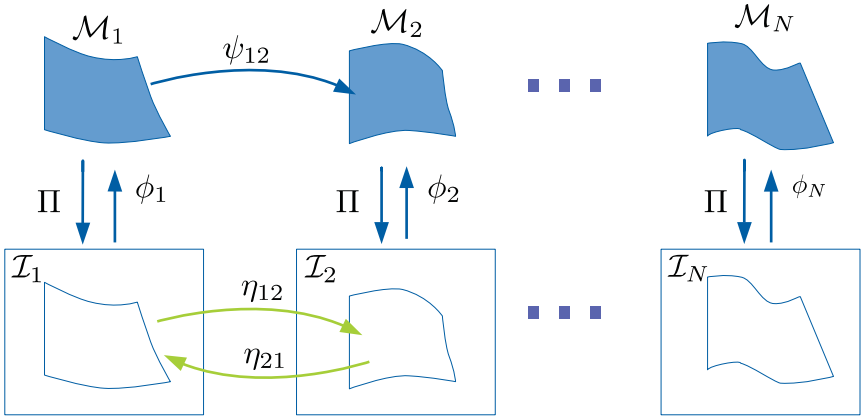


Figure 4.1: The proposed model of NRSfM, where each surface $\mathcal{M}_i$ is a Riemannian manifold defined by embedding the corresponding retinal plane $\mathcal{I}_i$.

However, these characteristics change according to the coordinate frame. We use the retinal plane $\mathcal{I}_i$ as coordinate frame for $\mathcal{M}_i$ and define as $\phi_i \in C^\infty(\mathcal{I}_i, \mathbb{R}^3)$ the image embedding for $\mathcal{M}_i$. We define as ψ_{ij} the isometric mapping between manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$.

4.2.2 The Metric Tensor

The metric tensor (see appendix A for more details) is a differential quantity used to define lengths, angles and areas on the surface [Lee, 1997]. The metric tensor of ϕ_i (in figure 4.2) is denoted as $\mathbf{g}_{mn}[\phi_i]$. We use the standard Einstein tensor notation and thus $\mathbf{g}_{mn}[\phi_i]$ is a combined reference to all elements of the metric tensor, a 2×2 matrix in this case. According to the Einstein summation convention, the summation is done over the indices appearing twice in the expression. Also, the free indices in an expression (the ones that do not appear twice in the expression) can be seen as both the indexed element or the whole arrangement. The indices m and n refer to the components of the coordinate frame of ϕ_i. In figure 4.2, we have

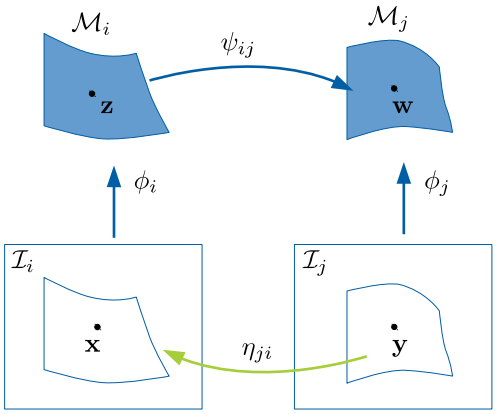


Figure 4.2: Simplified notation for two images.

$\mathbf{z} = \phi_i(\mathbf{x})$ and:

$$\mathbf{J}_{\phi_i} = \begin{pmatrix} \frac{\partial z^1}{\partial x^1} & \frac{\partial z^2}{\partial x^1} & \frac{\partial z^3}{\partial x^1} \\ \\ \frac{\partial z^1}{\partial x^2} & \frac{\partial z^2}{\partial x^2} & \frac{\partial z^3}{\partial x^2} \end{pmatrix}^\top . \tag{4.1}$$

The metric tensor of ϕ_i is then:

$$\mathbf{g}_{mn}[\phi_i] = \mathbf{J}_{\phi_i}^\top \mathbf{J}_{\phi_i} = \frac{\partial z^s}{\partial x^m}\frac{\partial z^k}{\partial x^n}\delta_{sk}, \tag{4.2}$$

with δ_{sk} the Kronecker delta function. We recall that according to the Einstein summation convention, the summation in equation (4.2) is done over indices s and k. The inverse of the metric tensor is expressed with raised indices $\mathbf{g}^{mn}[\phi_i]$. Given the change of coordinates:

$$\mathbf{x} = \eta(\mathbf{y}) \quad \text{with} \quad \mathbf{y} = \begin{pmatrix} y^1 & y^2 \end{pmatrix}^\top , \tag{4.3}$$

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

$$\mathbf{g}_{st}[\phi_i \circ \eta] = \mathbf{J}_\eta^\top \mathbf{J}_{\phi_i}^\top \mathbf{J}_{\phi_i} \mathbf{J}_\eta = \frac{\partial x^m}{\partial y^s}\frac{\partial x^n}{\partial y^t}\mathbf{g}_{mn}[\phi_i]. \tag{4.4}$$

4.2.3 Christoffel Symbols

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

$$\mathbf{\Gamma}^p_{mn}[\phi_i] = \frac{1}{2}\mathbf{g}^{pl}[\phi_i]\left(\mathbf{g}_{lm,n}[\phi_i] + \mathbf{g}_{ln,m}[\phi_i] - \mathbf{g}_{mn,l}[\phi_i]\right), \tag{4.5}$$

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

$$\mathbf{\Gamma}^q_{st}[\phi_i \circ \eta] = \frac{\partial x^m}{\partial y^s}\frac{\partial x^n}{\partial y^t}\mathbf{\Gamma}^p_{mn}[\phi_i]\frac{\partial y^q}{\partial x^p} + \frac{\partial y^q}{\partial x^l}\frac{\partial^2 x^l}{\partial y^s \partial y^t}. \tag{4.6}$$

Note that even though CS are expressed with tensor notation, they are not tensors and thus equation (4.6) does not correspond to the way tensors change coordinates. The CS of the image embedding, defined via the inverse-depth in equation (3.4), has a special structure given in Theorem 1.

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

$$\begin{aligned}\mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] &= -\frac{1}{\alpha_i}\begin{pmatrix}2\alpha_{i,1} & \alpha_{i,2}\\ \alpha_{i,2} & 0\end{pmatrix} + \frac{(\alpha_i)^2 A_i}{D_i}\begin{pmatrix}\alpha_{i,11} & \alpha_{i,12}\\ \alpha_{i,12} & \alpha_{i,22}\end{pmatrix}\\ \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] &= -\frac{1}{\alpha_i}\begin{pmatrix}0 & \alpha_{i,1}\\ \alpha_{i,1} & 2\alpha_{i,2}\end{pmatrix} + \frac{(\alpha_i)^2 B_i}{D_i}\begin{pmatrix}\alpha_{i,11} & \alpha_{i,12}\\ \alpha_{i,12} & \alpha_{i,22}\end{pmatrix},\end{aligned} \tag{4.7}$$

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

$$\begin{aligned}A_i &= -x^1\alpha_i + \left(1 + \left(x^1\right)^2\right)\alpha_{i,1} + x^1x^2\alpha_{i,2}\\ B_i &= -x^2\alpha_i + \left(1 + \left(x^2\right)^2\right)\alpha_{i,2} + x^1x^2\alpha_{i,1}\\ D_i &= \left(\alpha_i - x^1\alpha_{i,1} - x^2\alpha_{i,2}\right)^2 + (\alpha_{i,1})^2 + (\alpha_{i,2})^2\,.\end{aligned} \tag{4.8}$$

Proof. From the definition of $\phi_i(\mathbf{x})$ in equation (3.4), we can write the Jacobian matrix of $\phi_i(\mathbf{x})$ as:

$$\mathbf{J}_{\phi_i(\mathbf{x})} = \frac{1}{\alpha_i^2}\begin{pmatrix}\alpha_i - x^1\alpha_{i,1} & -x^1\alpha_{i,2}\\ -x^2\alpha_{i,1} & \alpha_i - x^2\alpha_{i,2}\\ -\alpha_{i,1} & -\alpha_{i,2}\end{pmatrix}. \tag{4.9}$$

Next we compute the metric tensor by substituting the Jacobian matrix from equation (4.9) in equation (4.2). The metric tensor is given by

$$\begin{aligned}\mathbf{g}_{11}[\phi_i(\mathbf{x})] &= \frac{1}{\alpha_i^4}\left(\epsilon^2(\alpha_{i,1})^2 + (\alpha_i)^2 - 2x^1\alpha_i\alpha_{i,1}\right)\\ \mathbf{g}_{12}[\phi_i(\mathbf{x})] &= \frac{1}{\alpha_i^4}\left(\epsilon^2\alpha_{i,1}\alpha_{i,2} - x^1\alpha_i\alpha_{i,2} - x^2\alpha_i\alpha_{i,1}\right)\\ \mathbf{g}_{22}[\phi_i(\mathbf{x})] &= \frac{1}{\alpha_i^4}\left(\epsilon^2(\alpha_{i,2})^2 + (\alpha_i)^2 - 2x^2\alpha_i\alpha_{i,2}\right).\end{aligned} \tag{4.10}$$

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

$$\begin{aligned}\mathbf{g}^{11}[\phi_i(\mathbf{x})] &= \frac{\mathbf{g}_{22}[\phi_i(\mathbf{x})]}{\det(\mathbf{g}[\phi_i(\mathbf{x})])} = \frac{(\alpha_i)^8\,\mathbf{g}_{22}[\phi_i(\mathbf{x})]}{D_i}\\ \mathbf{g}^{12}[\phi_i(\mathbf{x})] &= -\frac{\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{\det(\mathbf{g}[\phi_i(\mathbf{x})])} = -\frac{(\alpha_i)^8\,\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{D_i}\\ \mathbf{g}^{22}[\phi_i(\mathbf{x})] &= \frac{\mathbf{g}_{11}[\phi_i(\mathbf{x})]}{\det(\mathbf{g}[\phi_i(\mathbf{x})])} = \frac{(\alpha_i)^8\,\mathbf{g}_{11}[\phi_i(\mathbf{x})]}{D_i}.\end{aligned} \tag{4.11}$$

The derivatives of the metric tensor are given by

$$\begin{aligned}\mathbf{g}_{11,1}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})] + \frac{2E_i\alpha_{i,11}}{(\alpha_i)^4}\\ \mathbf{g}_{12,1}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{12}[\phi_i(\mathbf{x})] - \frac{H_i}{(\alpha_i)^4} + \frac{E_i\alpha_{i,12} + F_i\alpha_{i,11}}{(\alpha_i)^4}\end{aligned}$$

$$
\begin{aligned}
\mathbf{g}_{22,1}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{22}[\phi_i(\mathbf{x})] + \frac{2L_i}{(\alpha_i)^4} + \frac{2F_i\alpha_{i,12}}{(\alpha_i)^4} \\
\mathbf{g}_{11,2}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,2}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})] + \frac{2H_i}{(\alpha_i)^4} + \frac{2E_i\alpha_{i,12}}{(\alpha_i)^4} \\
\mathbf{g}_{12,2}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,2}}{\alpha_i}\mathbf{g}_{12}[\phi_i(\mathbf{x})] - \frac{L_i}{(\alpha_i)^4} + \frac{E_i\alpha_{i,22} + F_i\alpha_{i,12}}{(\alpha_i)^4} \\
\mathbf{g}_{22,2}[\phi_i(\mathbf{x})] &= -\frac{4\alpha_{i,2}}{\alpha_i}\mathbf{g}_{22}[\phi_i(\mathbf{x})] + \frac{2F_i\alpha_{i,22}}{(\alpha_i)^4}, \\
\text{where } E_i &= \left(1 + \left(x^1\right)^2 + \left(x^2\right)^2\right)\alpha_{i,1} - x^1\alpha_i,\ F_i = \left(1 + \left(x^1\right)^2 + \left(x^2\right)^2\right)\alpha_{i,2} - x^2\alpha_i, \\
H_i &= x^2\left(\alpha_{i,1}\right)^2 + \alpha_i\alpha_{i,2} - x^1\alpha_{i,1}\alpha_{i,2},\ \text{and } L_i = x^1\left(\alpha_{i,2}\right)^2\alpha_{i,1}\alpha_i + -x^2\alpha_{i,1}\alpha_{i,2}.
\end{aligned}
\tag{4.12}
$$

According to equation (4.5), the CS are given by

$$
\begin{aligned}
\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})] = &\frac{1}{2}\mathbf{g}^{p1}[\phi_i(\mathbf{x})]\left(\mathbf{g}_{1m,n}[\phi_i(\mathbf{x})] + \mathbf{g}_{1n,m}[\phi_i(\mathbf{x})] - \mathbf{g}_{mn,1}[\phi_i(\mathbf{x})]\right) \\
&+ \frac{1}{2}\mathbf{g}^{p2}[\phi_i(\mathbf{x})]\left(\mathbf{g}_{2m,n}[\phi_i(\mathbf{x})] + \mathbf{g}_{2n,m}[\phi_i(\mathbf{x})] - \mathbf{g}_{mn,2}[\phi_i(\mathbf{x})]\right).
\end{aligned}
\tag{4.13}
$$

Using the metric tensor and its derivatives from equations (4.10) and (4.12) in the expression for the CS given in equation (4.13), gives the result in equation (4.7). For example, $\mathbf{\Gamma}^1_{11}[\phi_i(\mathbf{x})]$ is given by

$$
\begin{aligned}
\mathbf{\Gamma}^1_{11}[\phi_i(\mathbf{x})] =& \frac{1}{2}\mathbf{g}^{11}[\phi_i(\mathbf{x})]\left(\mathbf{g}_{11,1}[\phi_i(\mathbf{x})]\right) + \frac{1}{2}\mathbf{g}^{12}[\phi_i(\mathbf{x})]\left(2\mathbf{g}_{21,1}[\phi_i(\mathbf{x})] - \mathbf{g}_{11,2}[\phi_i(\mathbf{x})]\right) \\
=& \frac{(\alpha_i)^8\,\mathbf{g}_{22}[\phi_i(\mathbf{x})]}{D_i}\left(-\frac{2\alpha_{i,1}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})] + \frac{E_i\alpha_{i,11}}{(\alpha_i)^4}\right) \\
&- \frac{(\alpha_i)^8\,\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{D_i}\left(-\frac{4\alpha_{i,1}}{\alpha_i}\mathbf{g}_{12}[\phi_i(\mathbf{x})] - \frac{2H_i}{(\alpha_i)^4}\right) \\
&- \frac{(\alpha_i)^8\,\mathbf{g}_{12}[\phi_i(\mathbf{x})]}{D_i}\left(\frac{F_i\alpha_{i,11}}{(\alpha_i)^4} + \frac{2\alpha_{i,2}}{\alpha_i}\mathbf{g}_{11}[\phi_i(\mathbf{x})]\right) = \frac{-2\alpha_{i,1}}{\alpha_i} + \frac{(\alpha_i)^2 A_i}{D_i}.
\end{aligned}
\tag{4.14}
$$

□

4.2.4 Commutativity under Isometry

Images and surfaces in Iso-NRSfM follow the commutative diagram shown in figure 4.2. Therefore,

$$
\begin{aligned}
\phi_j &= \psi_{ij} \circ \phi_i \circ \eta_{ji} \\
\mathbf{J}_{\phi_j} &= \mathbf{J}_{\psi_{ij}}\mathbf{J}_{\phi_i}\mathbf{J}_{\eta_{ji}}.
\end{aligned}
\tag{4.15}
$$

The metric tensor of ϕ_j can be written according to equation (4.2). It is given by

$$
\mathbf{J}^\top_{\phi_j}\mathbf{J}_{\phi_j} = \mathbf{J}^\top_{\eta_{ji}}\mathbf{J}^\top_{\phi_i}\mathbf{J}^\top_{\psi_{ij}}\mathbf{J}_{\psi_{ij}}\mathbf{J}_{\phi_i}\mathbf{J}_{\eta_{ji}} = \mathbf{J}^\top_{\eta_{ji}}\mathbf{J}^\top_{\phi_i}\mathbf{J}_{\phi_i}\mathbf{J}_{\eta_{ji}}. \tag{4.16}
$$

The fact that mappings between manifolds are isometric ($\mathbf{J}^\top_{\psi_{ij}}\mathbf{J}_{\psi_{ij}} = \mathbf{I}_{3\times3}$) [Bartoli et al.,

2015] allows us to derive the following fundamental result: *in Iso-NRSfM, both metric tensors and CS commute between surfaces with a change of variable given by the image warps.* This result is formalised with Theorem 2 and Corollary 1.

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

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

$$\phi_j = \psi_{ij} \circ \phi_i \circ \eta_{ji}. \tag{4.17}$$

From equations (4.4) and (4.17) we have:

$$\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[(\psi_{ij} \circ \phi_i) \circ \eta_{ji}] = \frac{\partial x^s}{\partial y^m}\frac{\partial x^t}{\partial y^n}\mathbf{g}_{st}[\psi_{ij} \circ \phi_i]. \tag{4.18}$$

By definition, isometric mappings do not change the local metric and so $\mathbf{g}[\psi_{ij} \circ \phi_i] = \mathbf{g}[\phi_i]$, which applied to equation (4.18) gives:

$$\mathbf{g}_{mn}[\phi_j] = \frac{\partial x^s}{\partial y^m}\frac{\partial x^t}{\partial y^n}\mathbf{g}_{st}[\phi_i]. \tag{4.19}$$

Identifying equation (4.4) with equation (4.19) gives the equality $\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[\phi_i \circ \eta_{ji}]$. □

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

Proof. As described in equation (4.5), $\mathbf{\Gamma}^p_{mn}[\phi_j]$ is a function of $\mathbf{g}_{mn}[\phi_j]$ and its derivatives. From Theorem 2 we have that $\mathbf{g}_{mn}[\phi_j] = \mathbf{g}_{mn}[\phi_i \circ \eta_{ji}]$. By multiplying this expression in both sides by $\mathbf{g}^{mn}[\phi_j]$, we get:

$$\mathbf{g}^{mn}[\phi_j]\mathbf{g}_{mn}[\phi_j] = \mathbf{g}^{mn}[\phi_j]\mathbf{g}_{mn}[\phi_i \circ \eta_{ji}] = \delta_{mn}, \tag{4.20}$$

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

$$\partial_l \mathbf{g}_{mn}[\phi_j] = \partial_l \mathbf{g}_{mn}[\phi_i \circ \eta_{ji}], \tag{4.21}$$

giving $\mathbf{g}_{mn,l}[\phi_j] = \mathbf{g}_{mn,l}[\phi_i \circ \eta_{ji}]$. By substitution of these identities in equation (4.19) we obtain:

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

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

These results show that, given the metric tensor and the CS for one image embedding, they can be transferred to the rest of the embeddings using the warps, which are known entities

in Iso-NRSfM. Note that this result cannot be generalised to non-isometric mappings. This establishes the ground rules for developing a local solution to Iso-NRSfM where the number of unknowns does not grow with the number of images. The main idea is to define the unknowns in a reference image and to use the warps to transfer all constraints into it.

4.2.5 Infinitesimal Planarity

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

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

Proof. Suppose $\mathcal{M}$ is a plane described by the equation $\mathbf{n}^\top \mathbf{z} + d = 0$, where $\mathbf{z} = \begin{pmatrix} z^1 & z^2 & z^3 \end{pmatrix}^\top$ and $\mathbf{n}$ is the plane's normal. From equation (3.3), the embedding is expressed with a depth function $\phi(\mathbf{x}) = \rho(\mathbf{x}) \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top$. By combining the depth parametrisation with the plane equation, we have:

$$\mathbf{n}^\top \rho(\mathbf{x}) \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top + d = 0, \tag{4.22}$$

from which we compute ρ as:

$$\rho(\mathbf{x}) = -\frac{d}{\mathbf{n}^\top \begin{pmatrix} \mathbf{x} & 1 \end{pmatrix}^\top}. \tag{4.23}$$

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

$$\phi(\mathbf{x}) = \beta(\mathbf{x})^{-1}(\mathbf{x} \quad 1)^\top, \tag{4.24}$$

where $\beta(\mathbf{x})$ is linear in $\mathbf{x}$. □

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

$$\mathbf{\Gamma}^1_{mn}[\phi(\mathbf{x})] = \frac{1}{\beta(\mathbf{x})} \begin{pmatrix} -2\beta_1(\mathbf{x}) & -\beta_2(\mathbf{x}) \\ -\beta_2(\mathbf{x}) & 0 \end{pmatrix} \qquad \mathbf{\Gamma}^2_{mn}[\phi(\mathbf{x})] = \frac{1}{\beta(\mathbf{x})} \begin{pmatrix} 0 & -\beta_1(\mathbf{x}) \\ -\beta_1(\mathbf{x}) & -2\beta_2(\mathbf{x}) \end{pmatrix}, \tag{4.25}$$

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

Proof. From the definition of $\phi(\mathbf{x})$ in equation (4.24), we can write the Jacobian matrix of $\phi(\mathbf{x})$ as:

$$\mathbf{J}_\phi(\mathbf{x}) = \frac{1}{\beta(\mathbf{x})^2} \begin{pmatrix} \beta(\mathbf{x}) - x^1\beta_1(\mathbf{x}) & -x^1\beta_2(\mathbf{x}) \\ -x^2\beta_1(\mathbf{x}) & \beta(\mathbf{x}) - x^2\beta_2(\mathbf{x}) \\ -\beta_1(\mathbf{x}) & -\beta_2(\mathbf{x}) \end{pmatrix}. \tag{4.26}$$

Using equation (4.2), the metric tensor at $\phi(\mathbf{x})$ can be written as $\mathbf{J}^{\top}_{\phi(\mathbf{x})}\mathbf{J}_{\phi(\mathbf{x})}$. The expression is given by

$$\begin{aligned}
\mathbf{g}_{11}[\phi(\mathbf{x})] &= \frac{1}{\beta^4}\left(\epsilon^2\left(\beta_1\right)^2 + \beta^2 - 2x^1\beta\beta_1\right) \\
\mathbf{g}_{12}[\phi(\mathbf{x})] &= \frac{1}{\beta^4}\left(\epsilon^2\beta_1\beta_2 - x^1\beta\beta_2 - x^2\beta\beta_1\right) \\
\mathbf{g}_{22}[\phi(\mathbf{x})] &= \frac{1}{\beta^4}\left(\epsilon^2\left(\beta_2\right)^2 + \beta^2 - 2x^2\beta\beta_2\right).
\end{aligned} \tag{4.27}$$

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

$$\begin{aligned}
\mathbf{g}_{11,1}[\phi(\mathbf{x})] &= -\frac{4\beta_1}{\beta}\mathbf{g}_{11}[\phi(\mathbf{x})] \\
\mathbf{g}_{12,1}[\phi(\mathbf{x})] &= -\frac{4\beta_1}{\beta}\mathbf{g}_{12}[\phi(\mathbf{x})] - \frac{\left(x^2\left(\beta_1\right)^2 + \beta\beta_2 - x^1\beta_1\beta_2\right)}{\left(\beta\right)^4} \\
\mathbf{g}_{22,1}[\phi(\mathbf{x})] &= -\frac{4\beta_1}{\beta}\mathbf{g}_{22}[\phi(\mathbf{x})] + \frac{2\left(x^1\left(\beta_2\right)^2\beta_1\beta + -x^2\beta_1\beta_2\right)}{\left(\beta\right)^4} \\
\mathbf{g}_{11,2}[\phi(\mathbf{x})] &= -\frac{4\beta_2}{\beta}\mathbf{g}_{11}[\phi(\mathbf{x})] + \frac{2\left(x^2\left(\beta_1\right)^2 + \beta\beta_2 - x^1\beta_1\beta_2\right)}{\left(\beta\right)^4} \\
\mathbf{g}_{12,2}[\phi(\mathbf{x})] &= -\frac{4\beta_2}{\beta}\mathbf{g}_{12}[\phi(\mathbf{x})] - \frac{\left(x^1\left(\beta_2\right)^2\beta_1\beta + -x^2\beta_1\beta_2\right)}{\left(\beta\right)^4} \\
\mathbf{g}_{22,2}[\phi(\mathbf{x})] &= -\frac{4\beta_2}{\beta}\mathbf{g}_{22}[\phi(\mathbf{x})].
\end{aligned} \tag{4.28}$$

Note that there are no second-order derivatives in the above expression because they vanish in the case of planes. This leads to the CS in equation (4.25). □

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

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

$$\mathbf{\Gamma}^1_{22}[\phi(\mathbf{x})] = \mathbf{\Gamma}^2_{11}[\phi(\mathbf{x})] = 0 \qquad 2\mathbf{\Gamma}^2_{12}[\phi(\mathbf{x})] = \mathbf{\Gamma}^2_{22}[\phi(\mathbf{x})] \qquad \mathbf{\Gamma}^1_{11}[\phi(\mathbf{x})] = 2\mathbf{\Gamma}^2_{12}[\phi(\mathbf{x})]. \tag{4.29}$$

We derive Theorem 4 from equation (4.29). It shows that the warps must comply with the 2D Schwarzian derivatives [Sasaki and Yoshida, 2002], which are second-order bilinear PDE that arise in the field of projective differential geometry.

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

2D Schwarzian equations.

Proof. The elements of the CS for $\mathcal{M}_i$ with $i = [1, \dots, N]$ have the form of (4.25), and thus must comply with the following algebraic constraints:

$$\boldsymbol{\Gamma}^1_{22}[\phi_i] = \boldsymbol{\Gamma}^2_{11}[\phi_i] = 0 \qquad 2\boldsymbol{\Gamma}^2_{12}[\phi_i] = \boldsymbol{\Gamma}^2_{22}[\phi_i] \qquad \boldsymbol{\Gamma}^1_{11}[\phi_i] = 2\boldsymbol{\Gamma}^2_{12}[\phi_i]. \tag{4.30}$$

From Corollary 1 we have $\boldsymbol{\Gamma}^p_{mn}[\phi_j] = \boldsymbol{\Gamma}^p_{mn}[\phi_i \circ \eta_{ji}]$. Now we use equation (4.6) to compute $\boldsymbol{\Gamma}^p_{nm}[\phi_i \circ \eta_{ji}]$ from $\boldsymbol{\Gamma}^p_{nm}[\phi_i]$ given in equation (4.25). Given that $\mathbf{x} = \eta_{ji}(\mathbf{y})$, we write

$$\begin{aligned}\boldsymbol{\Gamma}^p_{nm}[\phi_i \circ \eta_{ji}] = \partial_1 y^p \left(-2\frac{\beta_{i,1}}{\beta_i}\partial_m x^1 \partial_n x^1 - \frac{\beta_{i,2}}{\beta_i}\left(\partial_m x^1 \partial_n x^2 + \partial_m x^2 \partial_n x^1\right)\right) + \\ \partial_2 y^p \left(-2\frac{\beta_{i,2}}{\beta_i}\partial_m x^2 \partial_n x^2 - \frac{\beta_{i,1}}{\beta_i}\left(\partial_m x^1 \partial_n x^2 + \partial_m x^2 \partial_n x^1\right)\right) + \partial_1 y^p \partial^2_{mn} x^1 + \partial_2 y^p \partial^2_{mn} x^2.\end{aligned} \tag{4.31}$$

By forcing conditions in equation (4.30) in $\boldsymbol{\Gamma}[\phi_i \circ \eta_{ji}]$ we obtain the following four second-order PDE only in η_{ji}

$$\begin{aligned}\left(\partial^2_{11} x^1\right)\left(\partial_1 x^2\right) - \left(\partial^2_{11} x^2\right)\left(\partial_1 x^1\right) = 0 \\ \left(\partial^2_{22} x^1\right)\left(\partial_2 x^2\right) - \left(\partial^2_{22} x^2\right)\left(\partial_2 x^1\right) = 0 \\ \left(\partial_{11} x^1\right)\left(\partial_2 x^2\right) - \left(\partial_{11} x^2\right)\left(\partial_2 x^2\right) + 2\left(\left(\partial_{12} x^1\right)\left(\partial_1 x^2\right) - \left(\partial_{12} x^2\right)\left(\partial_1 x^1\right)\right) = 0 \\ \left(\partial_{22} x^1\right)\left(\partial_1 x^2\right) - \left(\partial_{22} x^2\right)\left(\partial_1 x^1\right) + 2\left(\left(\partial_{12} x^1\right)\left(\partial_2 x^2\right) - \left(\partial_{12} x^2\right)\left(\partial_2 x^1\right)\right) = 0.\end{aligned} \tag{4.32}$$

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

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

4.3 Reconstruction Equations

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

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

4.3.1 Relating the Metric Tensor and the Christoffel Symbols

For a non-planar surface, the CS at $\mathbf{z} \in \mathcal{M}_i$ are given by equation (4.7). We define them as:

$$\mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] = \begin{pmatrix} c_1 & c_3 \\ c_3 & c_5 \end{pmatrix} \quad \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] = \begin{pmatrix} c_2 & c_4 \\ c_4 & c_6 \end{pmatrix}, \tag{4.33}$$

where c_1, c_2, c_3, c_4, c_5 and c_6 are expressed in terms of the first and second-order derivatives of $\alpha_i(\mathbf{x})$ defined in equation (3.4). The expressions in equation (4.7) are:

$$\begin{aligned} c_1 &= -2k_1 + k_3 A_i \qquad c_2 = k_3 B_i \qquad c_3 = -k_2 + k_4 A_i \\ c_4 &= -k_1 + k_4 B_i \qquad c_5 = k_5 A_i \qquad c_6 = -2k_2 + k_5 B_i, \end{aligned} \tag{4.34}$$

with:

$$\begin{aligned} A_i &= -x^1 + \left(1 + \left(x^1\right)^2\right) k_1 + x^1 x^2 k_2 \\ B_i &= -x^2 + \left(1 + \left(x^2\right)^2\right) k_2 + x^1 x^2 k_1 \\ D_i &= \left(1 - x^1 k_1 - x^2 k_2\right)^2 + \left(k_1\right)^2 + \left(k_2\right)^2, \end{aligned} \tag{4.35}$$

where $k_1 = \dfrac{\alpha_{i,1}}{\alpha_i}$, $k_2 = \dfrac{\alpha_{i,2}}{\alpha_i}$, $k_3 = \dfrac{\alpha_{i,11}}{\alpha_i D_i}$, $k_4 = \dfrac{\alpha_{i,12}}{\alpha_i D_i}$ and $k_5 = \dfrac{\alpha_{i,22}}{\alpha_i D_i}$. The jacobian and hence the metric tensor at $\mathbf{z}$ can be written in terms of (k_1, k_2). Our goal is to find a relationship between the metric tensor parametrised with (k_1, k_2) and the CS $(c_1, c_2, c_3, c_4, c_5, c_6)$. Having such a relationship, we can formulate a system of equations exploiting the transfer of variables in the CS and metric tensor from one surface to another. From c_1 and c_2 in equation (4.34), we can write:

$$\frac{c_1 + 2k_1}{c_2} = \frac{A_i}{B_i}. \tag{4.36}$$

Similarly, from c_5 and c_6 in equation (4.34), we can write:

$$\frac{c_5}{c_6 + 2k_2} = \frac{A_i}{B_i}. \tag{4.37}$$

From c_3 and c_4 in equation (4.34), we find $\frac{A_i}{B_i} = \frac{c_3 + k_2}{c_4 + k_1}$. We substitute $\frac{A_i}{B_i}$ in equations (4.36) and (4.37), we obtain:

$$(c_1 + 2k_1)(c_4 + k_1) = c_2(c_3 + k_2) \qquad (c_6 + 2k_2)(c_3 + k_2) = c_5(c_4 + k_1). \tag{4.38}$$

From the first expression in equation (4.38), we find $k_2 = \frac{(c_1 + 2k_1)(c_4 + k_1)}{c_2} - c_3$ and substitute it in the second expression. We obtain the following quartic in k_1:

$$(c_4 + k_1)\left(8k_1^3 + 8\left(c_1 + c_4\right)k_1^2 + 2\left(c_1\left(c_1 + 4c_4\right) + c_2\left(c_6 - 2c_3\right)\right)k_1 + 2c_1^2c_4 + c_1c_2\left(c_6 - 2c_3\right) - c_2^2c_5\right) = 0. \tag{4.39}$$

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

Combining equation (4.38) with equation (4.34), we can express (k_1, k_2) in terms of the CS $(c_1, c_2, c_3, c_4, c_5, c_6)$ as a rational expression of degree two. This gives a system of two polynomials of degree 8 in 6 variables for each pair of views. Existing solvers such as [Henrion and Lasserre, 2003] cannot solve such high degree polynomial systems. We conclude that the first and second-order derivatives of $\alpha_i(\mathbf{x})$ cannot be solved jointly via estimating the CS. However, we see that the expressions of the CS in equation (4.34) are linear in terms of (k_1, k_2) and (k_3, k_4, k_5). By assuming (k_3, k_4, k_5) to be known, we can find a unique relationship between the metric tensor and the CS and vice versa. Therefore, splitting the problem in two steps of solving for the first and the second-order derivatives of $\alpha_i(\mathbf{x})$ separately leads to a solution to Iso-NRSfM.

4.3.2 Solving for the First-Order Derivatives

We assume that the second-order derivatives of $\alpha_i(\mathbf{x})$ are known. They can be assumed to be zero (as in the case of the infinitesimal planarity assumption) or they can be obtained by the method we describe next. We show how to solve for the first-order derivatives of $\alpha_i(\mathbf{x})$. We also show that this solution has a similar structure as the solution to Iso-NRSfM under IP assumption. We first select a pair of surfaces $(\mathcal{M}_i, \mathcal{M}_j)$ (see figure 4.2) and a point $\mathbf{x} = (x^1, x^2)^\top \in \mathcal{I}_i$. We evaluate $\mathbf{\Gamma}^p_{mn}[\phi_i]$ at $\mathbf{x}$, namely $\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]$. According to equation (4.7), it is given by:

$$\mathbf{\Gamma}^1_{mn}[\phi_i(\mathbf{x})] = \begin{pmatrix} -2k_1 + A_ip_1 & -k_2 + A_ip_2 \\ -k_2 + A_ip_2 & A_ip_3 \end{pmatrix} \qquad \mathbf{\Gamma}^2_{mn}[\phi_i(\mathbf{x})] = \begin{pmatrix} B_ip_1 & -k_1 + B_ip_2 \\ -k_1 + B_ip_2 & -2k_2 + B_ip_3 \end{pmatrix}, \tag{4.40}$$

where $k_1 = \frac{\beta_1(\mathbf{x})}{\beta(\mathbf{x})}$ and $k_2 = \frac{\beta_2(\mathbf{x})}{\beta(\mathbf{x})}$. The expressions (p_1, p_2, p_3) are functions of second-order derivatives of $\alpha_i(\mathbf{x})$ and therefore, they are known. A_i and B_i are linear expressions in (k_1, k_2)

according to equation (4.8). Next we compute $\mathbf{J}_{\phi_i}$ in terms of (k_1, k_2):

$$\mathbf{J}_{\phi_i}(\mathbf{x}) = \frac{1}{\beta(\mathbf{x})} \begin{pmatrix} 1 - k_1 x^1 & -k_2 x^1 \\ -k_1 x^2 & 1 - k_2 x^2 \\ -k_1 & -k_2 \end{pmatrix}. \tag{4.41}$$

By substitution of equation (4.41) in equation (4.2) we have:

$$\begin{aligned} \mathbf{g}_{11}[\phi_i(\mathbf{x})] =& \frac{1}{\beta(\mathbf{x})^2} \left({k_1}^2 + \left(k_1 x^1 - 1\right)^2 + \left(k_1 x^2\right)^2 \right) \\ \mathbf{g}_{12}[\phi_i(\mathbf{x})] =& \frac{1}{\beta(\mathbf{x})^2} \left(k_1 k_2 \left(1 + \left(x^1\right)^2 + \left(x^2\right)^2 \right) - k_2 x^1 - k_1 x^2 \right) \\ \mathbf{g}_{22}[\phi_i(\mathbf{x})] =& \frac{1}{\beta(\mathbf{x})^2} \left({k_2}^2 + \left(k_2 x^1\right)^2 + \left(k_2 x^2 - 1\right)^2 \right). \end{aligned} \tag{4.42}$$

We define $\mathbf{G}_{mn}[\phi_i(\mathbf{x})] = \beta(\mathbf{x})^2 \mathbf{g}_{mn}[\phi_i(\mathbf{x})]$, which only depends on (k_1, k_2). Let $\mathbf{x} = \eta_{ji}(\mathbf{y})$. We use equation (4.6) and Corollary 1 to compute $\mathbf{\Gamma}^k_{mn}[\phi_j(\mathbf{y})] = \mathbf{\Gamma}^k_{mn}[(\phi_i \circ \eta_{ji})(\mathbf{y})]$ as:

$$\begin{aligned} \mathbf{\Gamma}^1_{mn}[(\phi_i \circ \eta_{ji})(\mathbf{y})] &= \begin{pmatrix} -2\bar{k}_1 + A_j \bar{p}_1 & -\bar{k}_2 + A_j \bar{p}_2 \\ -\bar{k}_2 + A_j \bar{p}_2 & A_j \bar{p}_3 \end{pmatrix} \\ \mathbf{\Gamma}^2_{mn}[(\phi_i \circ \eta_{ji})(\mathbf{y})] &= \begin{pmatrix} B_j \bar{p}_1 & -\bar{k}_1 + B_j \bar{p}_2 \\ -\bar{k}_1 + B_j \bar{p}_2 & -2\bar{k}_2 + B_j \bar{p}_3 \end{pmatrix}, \end{aligned} \tag{4.43}$$

where according to equation (4.6), $(\bar{k}_1, \bar{k}_2)$ are linear combinations of (k_1, k_2) and $(\bar{p}_1, \bar{p}_2, \bar{p}_3)$. $(\bar{p}_1, \bar{p}_2, \bar{p}_3)$ are known. A_j and B_j are linear expressions in $(\bar{k}_1, \bar{k}_2)$ according to equation (4.8). From equation (4.42) one can find $\mathbf{g}_{st}[\phi_j(\mathbf{y})]$ in function of $(\bar{k}_1, \bar{k}_2)$, and thus in function of (k_1, k_2).

Alternatively, from equation (4.4) and using the definition of $\mathbf{G}_{mn}[\phi_i(\mathbf{x})]$ and $\mathbf{G}_{mn}[\phi_j(\mathbf{y})]$ we have the following identity:

$$\frac{1}{\beta(\mathbf{x})^2} \mathbf{G}_{st}[\phi_j(\mathbf{y})] = \frac{1}{\beta(\mathbf{y})^2} \frac{\partial x^m}{\partial y^s} \frac{\partial x^n}{\partial y^t} \mathbf{G}_{mn}[\phi_i(\mathbf{x})]. \tag{4.44}$$

We cancel $\beta(\mathbf{x})$ and $\beta(\mathbf{y})$ by converting system (4.44) into the following two equations:

$$\begin{aligned} \mathbf{G}_{11}[\phi_j(\mathbf{y})] \left(\frac{\partial x^m}{\partial y^1} \frac{\partial x^n}{\partial y^2} \mathbf{G}_{mn}[\phi_i(\mathbf{x})] \right) - \mathbf{G}_{12}[\phi_j(\mathbf{y})] \left(\frac{\partial x^m}{\partial y^1} \frac{\partial x^n}{\partial y^1} \mathbf{G}_{mn}[\phi_i(\mathbf{x})] \right) = 0 \\ \mathbf{G}_{22}[\phi_j(\mathbf{y})] \left(\frac{\partial x^m}{\partial y^1} \frac{\partial x^n}{\partial y^2} \mathbf{G}_{mn}[\phi_i(\mathbf{x})] \right) - \mathbf{G}_{12}[\phi_j(\mathbf{y})] \left(\frac{\partial x^m}{\partial y^2} \frac{\partial x^n}{\partial y^2} \mathbf{G}_{mn}[\phi_i(\mathbf{x})] \right) = 0. \end{aligned} \tag{4.45}$$

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

Equation (4.45) is a system of two cubics in variables (k_1, k_2) modeling Iso-NRSfM for manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$ at point $\mathbf{x} \in \mathcal{I}_i$. We denote the two equations as $\mathcal{Q}_{ij}(\mathbf{x}, \eta_{1j}(\mathbf{x}), k_1, k_2)$. By keeping the first index as the reference manifold, for instance $i = 1$, and obtaining the polynomials for the rest of the views we have $2N - 2$ polynomial equations in two vari-

ables $\mathcal{Q}_1(k_1,k_2) = \{\mathcal{Q}_{1j}(\mathbf{x},\eta_{1j}(\mathbf{x}),k_1,k_2)\}_{j=2}^{N}$. The solution (k_1,k_2) to the polynomial system $\mathcal{Q}_1(k_1,k_2)$ at the point $\mathbf{x}=\mathbf{x}_1$ allows us to reconstruct the metric tensor, the CS and the tangent plane for point $\mathbf{x}_1$ in view $\mathcal{I}_1$. Using equation (4.41) we can reconstruct $\mathbf{J}_{\phi_1}(\mathbf{x}_1)$ up to an unknown scale $\beta(\mathbf{x}_1)^{-1}$. It is not necessary to recover this scale to estimate a unit normal, which is computed by taking the cross product of the two columns of $\mathbf{J}_{\phi_1}(\mathbf{x}_1)$ and normalising.

We solve system $\mathcal{Q}_1(k_1,k_2)$ by finding the values of (k_1,k_2) that minimise the sum-of-squares of all polynomials in the system. This optimisation is solved globally using moment based convex optimisation [Henrion and Lasserre, 2003]. Given (k_1,k_2), we calculate $(\bar{k}_1,\bar{k}_2)$ by using equation (4.6) at each point.

Notice that the structure of the CS given in equation (4.25) for planes is very similar to equation (4.40) with (p_1,p_2,p_3) as zeros. This shows that the solution to Iso-NRSfM with the IP assumption is a special case of this solution. We express the system with zero second-order derivatives as $\mathcal{P}_1(k_1,k_2)$, which is solved in a similar way as $\mathcal{Q}_1(k_1,k_2)$.

4.3.3 Solving for the Second-Order Derivatives

We now show how to solve for the second-order derivatives of $\alpha_i(\mathbf{x})$, assuming that the first-order derivatives of $\alpha_i(\mathbf{x})$ are known from the previous step (we start by solving for the first-order derivatives assuming that second-order derivatives are null). The expressions for the CS in equation (4.40) become linear in the second-order derivatives of $\alpha_i(\mathbf{x})$. This means that $\mathbf{\Gamma}^p_{mn}[\phi_i(\mathbf{x})]$ is a linear function of (k_3,k_4,k_5). Given that $\mathbf{x}=\eta_{ji}(\mathbf{y})$ and equation (4.6), $\mathbf{\Gamma}^p_{mn}[(\phi_i\circ\eta_{ji})(\mathbf{y})]$ is given by:

$$\mathbf{\Gamma}^1_{mn}[(\phi_i\circ\eta_{ji})(\mathbf{y})] = \begin{pmatrix} \bar{c}_1 & \bar{c}_3 \\ \bar{c}_3 & \bar{c}_5 \end{pmatrix} \qquad \mathbf{\Gamma}^2_{mn}[(\phi_i\circ\eta_{ji}(\mathbf{y}))] = \begin{pmatrix} \bar{c}_2 & \bar{c}_4 \\ \bar{c}_4 & \bar{c}_6 \end{pmatrix}, \tag{4.46}$$

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

$$\bar{k}_3 = \frac{\left(\bar{c}_1+2\bar{k}_1\right)A_j+\bar{c}_2B_j}{A_j^2+B_j^2} \quad \bar{k}_4 = \frac{\left(\bar{c}_3+\bar{k}_2\right)A_j+\left(\bar{c}_4+\bar{k}_1\right)B_j}{A_j^2+B_j^2} \quad \bar{k}_5 = \frac{\bar{c}_5A_j+\left(\bar{c}_6+2\bar{k}_2\right)B_j}{A_j^2+B_j^2}. \tag{4.47}$$

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

In order to solve for the second-order derivatives of $\alpha_i(\mathbf{x})$, we differentiate the first-order

reconstruction equations (4.45). The expressions are given in equation (4.48).

$$
\begin{aligned}
&\frac{\partial \mathbf{G}_{11}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)\\
+&\mathbf{G}_{11}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0\\
&\frac{\partial \mathbf{G}_{11}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)\\
+&\mathbf{G}_{11}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^1}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0\\
&\frac{\partial \mathbf{G}_{22}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^1}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)\\
+&\mathbf{G}_{22}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^1}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0\\
&\frac{\partial \mathbf{G}_{22}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \frac{\partial \mathbf{G}_{12}[\phi_j(\mathbf{y})]}{\partial x^2}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right)\\
+&\mathbf{G}_{22}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^1}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) - \mathbf{G}_{12}[\phi_j(\mathbf{y})]\frac{\partial}{\partial x^2}\left(\frac{\partial x^m}{\partial y^2}\frac{\partial x^n}{\partial y^2}\mathbf{G}_{mn}[\phi_i(\mathbf{x})]\right) = 0.
\end{aligned}
\tag{4.48}
$$

The derivatives of $\mathbf{G}_{mn}[\phi_i(\mathbf{x})]$ in equation (4.45) are given in equation (4.49).

$$
\begin{aligned}
\frac{\partial \mathbf{G}_{11}[\phi_i(\mathbf{x})]}{\partial x^1} &= 2k_1\left(x^1k_1 - 1\right) + 2\left(\epsilon k_1 - x^1\right)\left(k_3D - {k_1}^2\right)\\
\frac{\partial \mathbf{G}_{12}[\phi_i(\mathbf{x})]}{\partial x^1} &= k_2\left(2x^1k_1 - 1\right) + \left(\epsilon k_2 - x^2\right)\left(k_3D - {k_1}^2\right) + \left(\epsilon k_1 - x^1\right)\left(k_4D - k_1k_2\right)\\
\frac{\partial \mathbf{G}_{22}[\phi_i(\mathbf{x})]}{\partial x^1} &= 2x^1{k_2}^2 + 2\left(\epsilon k_2 - x^2\right)\left(k_4D - k_1k_2\right)\\
\frac{\partial \mathbf{G}_{11}[\phi_i(\mathbf{x})]}{\partial x^2} &= 2x^2{k_1}^2 + 2\left(\epsilon k_1 - x^1\right)\left(k_4D - k_1k_2\right)\\
\frac{\partial \mathbf{G}_{12}[\phi_i(\mathbf{x})]}{\partial x^2} &= k_1\left(2x^2k_2 - 1\right) + \left(\epsilon k_2 - x^2\right)\left(k_4D - k_1k_2\right) + \left(\epsilon k_1 - x^1\right)\left(k_5D - {k_2}^2\right)\\
\frac{\partial \mathbf{G}_{22}[\phi_i(\mathbf{x})]}{\partial x^2} &= 2k_2\left(x^2k_2 - 1\right) + 2\left(\epsilon k_2 - x^2\right)\left(k_5D - {k_2}^2\right),\\
\text{with } D &= \left(1 - x^1k_1 - x^2k_2\right)^2 + {k_1}^2 + {k_2}^2 \text{ and } \epsilon = 1 + \left(x^1\right)^2 + \left(x^2\right)^2.
\end{aligned}
\tag{4.49}
$$

Equation (4.49) shows that the derivatives are linear functions of (k_3, k_4, k_5). Using equation (4.47), the reconstruction equations (4.48) form a linear system in three variables only, which can be solved using LLS. Therefore, for each pair of manifolds $(\mathcal{M}_i, \mathcal{M}_j)$ at point $\mathbf{x} \in \mathcal{I}_i$, equation (4.48) is a system of four linear equations in variables (k_3, k_4, k_5). We denote the four equations as $\mathcal{S}_{ij}(\mathbf{x}, \mathbf{y}, k_3, k_4, k_5)$. Fixing the i^{th} manifold as a reference, for instance $i = 1$, we obtain $4N - 4$ linear equations in three variables which are written as $\mathcal{S}_1(k_3, k_4, k_5) = \{\mathcal{S}_{1j}(\mathbf{x}, \eta_{1j}(\mathbf{x}), k_3, k_4, k_5)\}_{j=2}^{N}$. The solution (k_3, k_4, k_5) to the linear system $\mathcal{S}_1(k_3, k_4, k_5)$ for the point $\mathbf{x} = \mathbf{x}_1$ gives the second derivatives of $\alpha_i(\mathbf{x})$. We use them to compute a better estimate of the CS in equation (4.7). Using equation (4.47), we can obtain $(\bar{k}_3, \bar{k}_4, \bar{k}_5)$ using (k_3, k_4, k_5).

4.4 Algorithms

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

4.4.1 Solution under Infinitesimal Planarity

With the assumption of IP, we can write the metric tensor of equation (4.40) and the CS of equation (4.42) on a manifold $\mathcal{M}_i$ in terms of two variables which correspond to the ratio of first-order derivatives of the inverse-depth β of the image embedding ϕ_i with the inverse-depth function. We can also write the metric tensor and the CS on the rest of the images in terms of the variables in the first image (equations (4.4) and (4.6) respectively), which leads to two variables for N images. We solve the system of two cubics in two variables of equation (4.45) for all images by minimising the sum-of-squares of the polynomials. The algorithm takes the following steps:

Inputs: Warps η_{j1}, $j \in [2, N]$.

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

2) Find (k_1, k_2). Evaluate the polynomial $\mathcal{P}_1(k_1, k_2)$ and solve by minimising the sum of squares using [Henrion and Lasserre, 2003]. This gives (k_1, k_2). Find $(\bar{k}_1, \bar{k}_2)$ in terms of (k_1, k_2) and the η_{j1} warps, $j \in [2, N]$, using equation (4.6).

3) Find normals and 3D points. Find the jacobian in terms of (k_1, k_2) using equation (4.41). Compute the normals by taking the cross-product of the jacobian's columns and normalising it. Use the method in [Chhatkuli et al., 2014] to recover the 3D surfaces by integrating the normal fields.

Outputs: Points and normals on 3D surfaces.

4.4.2 General Solution

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

Inputs: Warps η_{j1}, $j \in [2, N]$.

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

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

3) Find the second-order derivatives (k_3,k_4,k_5). Evaluate the linear system $\mathcal{S}_1(k_3, k_4, k_5)$ and solve using LLS. This gives (k_3, k_4, k_5). Find $(\bar{k}_3, \bar{k}_4, \bar{k}_5)$ using equation (4.47).

4) Find (k_1, k_2). Evaluate the sum-of-squares polynomials of the system $\mathcal{Q}_1(k_1, k_2)$ and find (k_1, k_2) by minimising it using [Henrion and Lasserre, 2003]. Find $(\bar{k}_1, \bar{k}_2)$ in terms of (k_1, k_2) and the warps η_{j1}, $j \in [2, N]$, using equation (4.6).

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

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

Outputs: Points and normals on 3D surfaces.

4.4.3 Complexity Analysis

We discuss the complexity of the algorithm under the IP assumption and in the general solution. For both solutions, we assume that the warps are provided. We calculate the warps using schwarps [Pizarro et al., 2016], that impose the Schwarzian equations (4.29) as a soft constraint. However we would like to point out that we do not require warps between all possible pairs of images in the sequence. We use a reference view and thus require the computation of $N - 1$ warps only, not N^2.

Solution with Infinitesimal Planarity

The solution to Iso-NRSfM under IP solves only one sextic polynomial for N images. This polynomial is formed by computing the sum-of-squares of $2(N-1)$ cubic polynomials (4.45). Forming this sextic polynomial has a linear complexity but solving it is independent of N.

General Solution

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

4.5 Experimental Results

We tested Iso-NRSfM on synthetic and real datasets. Figure 4.3 shows some images from real datasets on which the methods were evaluated. For quantitative comparison, we measured the normal error (mean difference between computed and ground truth normals in degrees)

Figure 4.3: Some images of the rug, table mat, kinect paper and tshirt datasets. The five rightmost images of the table mat dataset are zoomed in to improve visibility.

and the depth error (mean difference between computed and ground truth 3D coordinates in mm). We denote Iso-NRSfM with IP as **infP** and as **iso** otherwise. We compared our method with six other NRSfM methods: **diffH** [Chhatkuli et al., 2014], **mdhI** [Chhatkuli et al., 2016a], **kerF** [Gotardo and Martinez, 2011], **plaH** [Varol et al., 2009], **pieceI** [Taylor et al., 2010], **inextI** [Vicente and Agapito, 2012]. All the codes for these methods were obtained from the authors' websites except **plaH** which we re-implemented.

4.5.1 Synthetic Datasets

We simulated random views of a cylindrical surface deforming isometrically. The image size is $640p \times 480p$ and the focal length is $400p$. We tracked 400 points. We compared all methods by varying the number of views and noise in the images. **kerF** needs a temporal sequence, 10 views are not enough for reconstruction especially for short-baseline viewpoints. It therefore did not do well. Also, **mdhI** needs the views to be very different and therefore, it also gave very bad results on this sequence. [Chhatkuli et al., 2016a] mentioned that their method fails on such sequences. The results are shown in figure 4.4. The results are obtained after averaging the errors over 50 trials (the default is $1p$ noise and 10 views).

4.5.1.1 Varying the Number of Views

infP gives a very good reconstruction for three views which improves when more images are added. **iso** performs much better than **infP** as it does not assume IP; this helps in reconstructing the high curvature deformations more accurately. The errors of **iso** are almost half of **infP**. **plaH** and **diffH** give higher errors than **infP** and **iso** with **plaH** being better than **diffH**. However, **diffH** improves faster with the number of views and gives better results

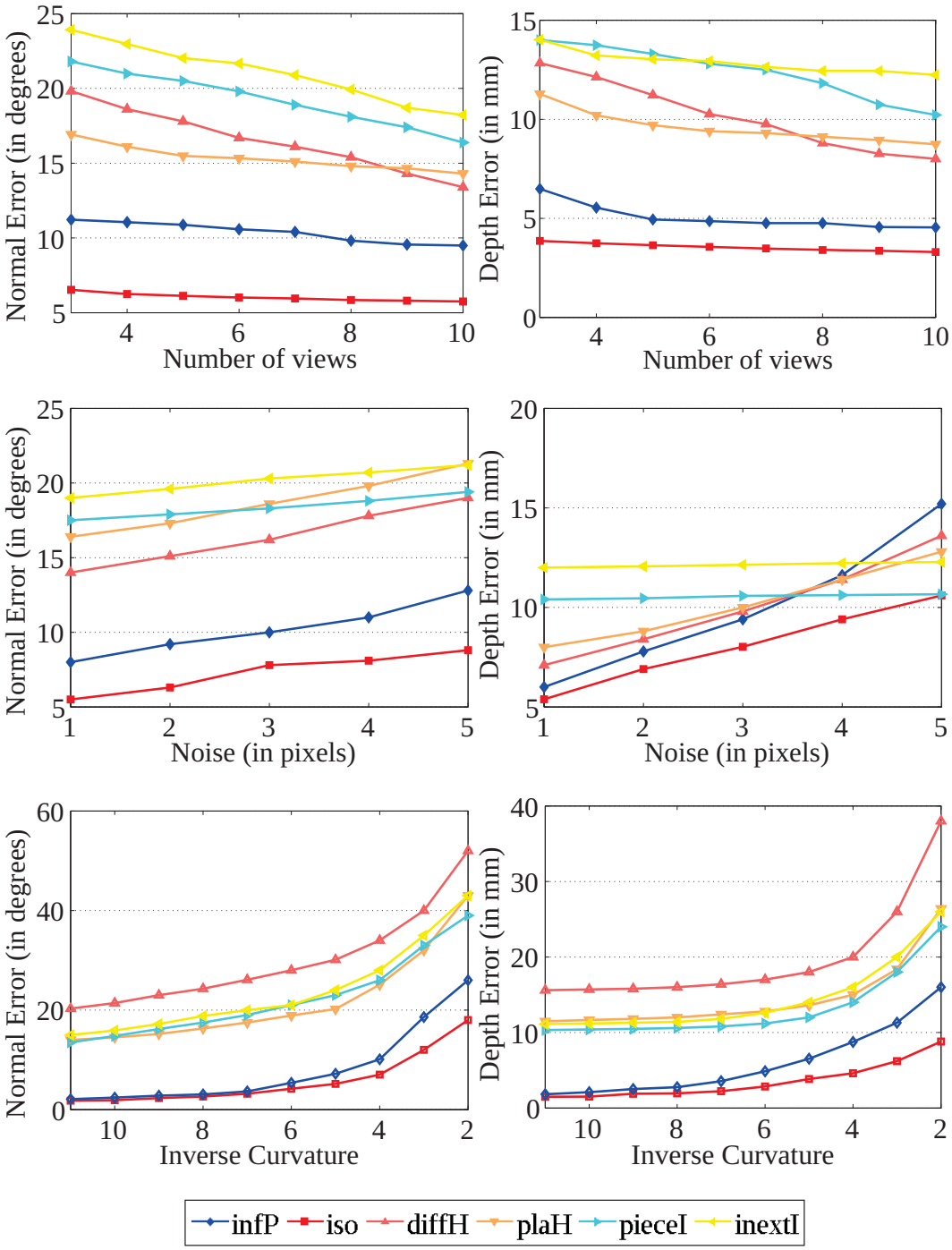


Figure 4.4: Synthetic data experiments. Average normal and depth errors with respect to number of views, noise and curvature. Best viewed in colour.

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

Table 4.1: Performance of warps in noisy conditions. The average pixel error due to added noise w.r.t. the B-spline control points are shown. The warps reduce the noise only when fewer control points are used, but then significantly degrade their derivatives.

than **plaH** for 8-10 views. The performance of these methods is off by almost 5 degrees compared with **infP**. **pieceI** and **inextI** give decent results only with 8-10 views but their performance is worst amongst the compared methods. Overall, **infP** and **iso** consistently show lower errors than all other methods.

4.5.1.2 Varying Noise

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

We also made an experiment to study the influence of noise on the warps. This is because we use warps to represent the image transformation and that estimating these could be reducing some of the noise applied to the correspondences. We simulated two synthetic images (I1 and I2) and added a 1-5 pixels noise to point correspondences in I2. We computed the schwarps [Pizarro et al., 2016] (using B-splines) between I1 and I2 using our default setup (30 control points and a fixed value for the hyperparameter controlling the Schwarps smoothing). We then computed the standard deviation of the pixel noise after computing the warps. The result is given in the last row of table 4.1, where we can see that the amount of noise is almost unaffected. We did the same experiment with fewer control points, which resulted in improved noise values (see the table 4.1). However, we found that using fewer points has an impact on the final accuracy as they degrade the derivatives, which directly influence the output of our method. We conclude that the warps do not remove the noise from the point correspondences.

4.5.1.3 Varying Curvature

We simulated a cylindrical surface with a varying radius. The curvature is the inverse of the radius. We simulated 10 surfaces with the radius varying from 2 to 11 and used 10 different views of each surface for the experiment. We see that the performance of **iso** and **infP** is best amongst the compared methods. Their performance is very similar when the surfaces are almost flat or less bent. As the curvature of the surfaces increase, we can see that **iso** performs much better than **infP**. **iso** handles high deformations better than **infP** because it estimates the second-order derivatives of the surface while **infP** assumes them to be zero. **diffH** and **plaH** need wide baseline views with different deformations, therefore, their performance is worse or comparable with **inextI** and **pieceI**.

4.5.2 Real Datasets

We conducted experiments with the four datasets shown in figure 4.3 and performed two types of experiments. The tshirt dataset is a wide baseline dataset and the rest of them are short baseline datasets. Our observations are summarised below.

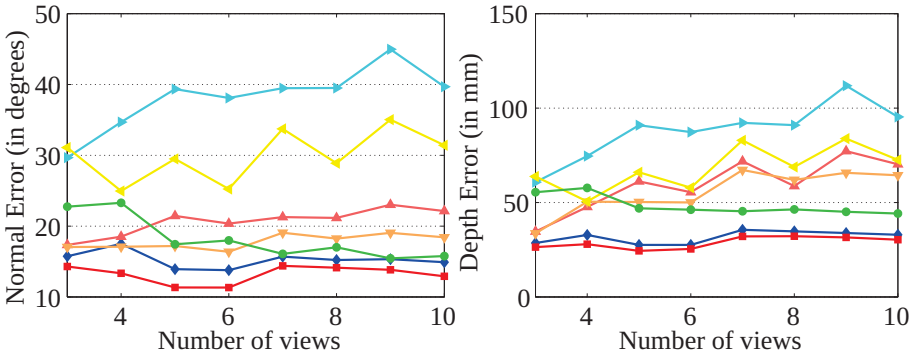
4.5.2.1 Short Sequences

The rug, table mat and kinect paper datasets are long sequences (159, 60 and 191 images with 350, 300 and 1500 point correspondences) and the t-shirt is a short dataset (10 images, 85 point correspondences). The long sequences are uniformly reduced by picking 10 images at regular intervals in the sequence. The results are shown in figure 4.5. The results for the tshirt dataset are averaged over 20 trials of randomly sampled images. The figure clearly shows that **iso** works best among the compared methods while the performance of **mdhI** and **infP** is quite good as well.

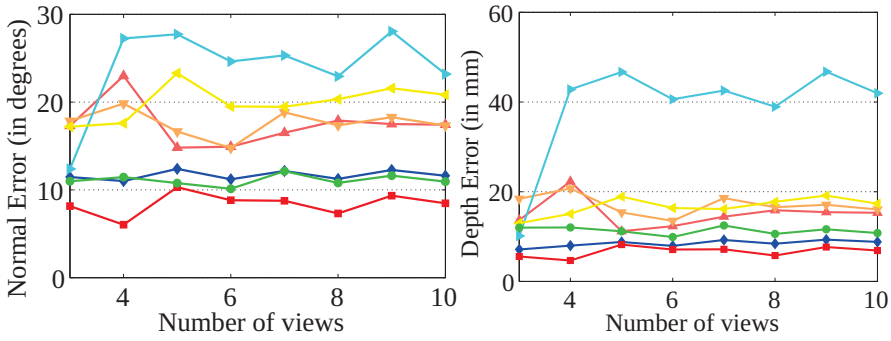
Rug dataset. **iso** and **infP** have the best performance on this dataset; **iso** being better than **infP**. **mdhI** improves with the number of views and gives comparable results to **infP** for 7-10 views. **diffH** and **plaH** show a worse performance than **mdhI**, with **plaH** being better.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

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

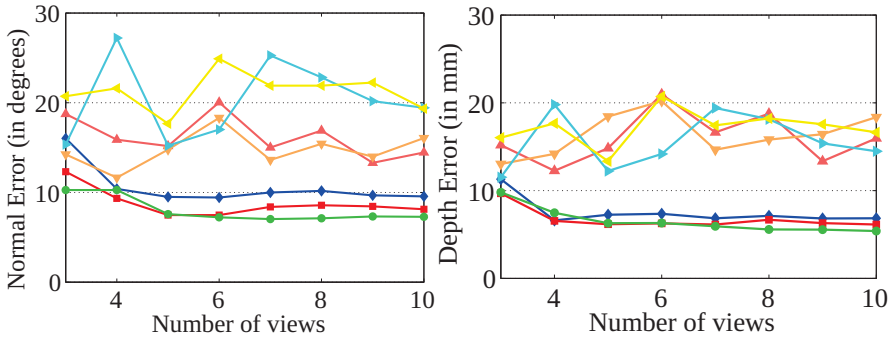
Kinect paper dataset. **mdhI** has the best performance on this dataset. **infP** and **iso** are very close to **mdhI**. **inextI** and **pieceI** have the worst normal error as compared to other



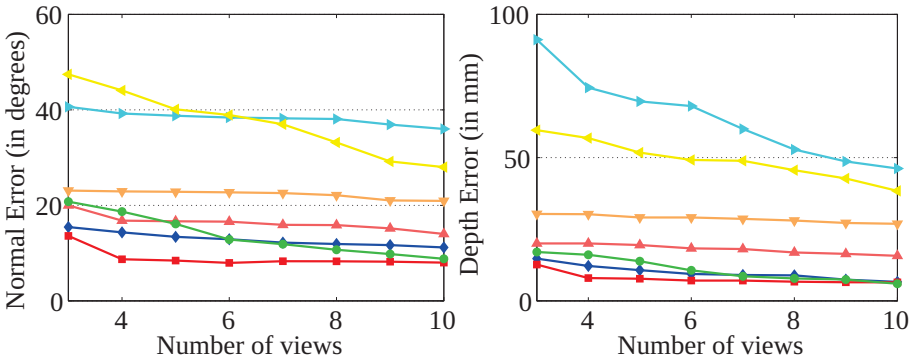
(a) Rug dataset



(b) Table mat dataset



(c) Kinect paper dataset



(d) Tshirt dataset

Figure 4.5: Experiments on short sequences. The average normal and depth error for each experiment with number of views varying between 3-10 is shown. The views of the rug, table mat and kinect paper datasets are selected by uniform sampling the long sequences. The views of the tshirt dataset are selected by randomly sampling the dataset and the results are averaged over 20 trials. Best viewed in colour.

methods. Their errors are almost twice as the best performing methods **mdhI**, **infP** and **iso**. **diffH** and **plaH** show better performance in terms of normals as compared to **inextI** and **pieceI**. The performance of **diffH**, **plaH**, **inextI** and **pieceI** in terms of depth error is similar.

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

Summary of experiments on short sequences **iso** and **infP** give the best performance on the rug and table mat datasets while **mdhI** gives best results on the kinect paper dataset. It is important to note that **iso** and **infP** converge quickly as compared to the rest of the methods. They show very good results for as few as three views. **iso** converges much quicker than **infP**; the errors seem to have been stabilised with four views only. Figure 4.6 shows 3D reconstruction error maps for the rug, table mat, kinect paper and tshirt datasets. We showed results for **iso**, **infP** and **mdhI** only because they are the most competitive methods amongst the compared methods. In case of occlusion, **iso** and **infP** can reconstruct the entire surface as long as the surface is visible in at least three images, therefore, even if the kinect paper dataset is occluded by a hand, it can be reconstructed by them. However, **mdhI** reconstructs only the visible part of each surface. Therefore we observe the occlusion in the reconstruction and rendering as well.

4.5.2.2 Long Sequences

The rug, table mat and kinect paper datasets are long sequences with 60, 159 and 191 images. Since our method can easily handle a large number of images, it is important to show results on large sequences by considering all images in the dataset. A limitation of current NRSfM methods is that they cannot handle a large number of views. Also, several NRSfM methods such as **diffH** and **plaH** [Chhatkuli et al., 2014; Varol et al., 2009] reconstruct the reference image only and are computationally expensive to recover the other shapes. **kerF** reconstructs the entire image set in one execution and therefore, we compare our method with **kerF** on long sequences. The cat dataset is a relatively short sequences (60 images) therefore, we added the results of **inextI** for this dataset on the entire sequence. Figure 4.7 shows the comparison of our method with others. It is very clearly visible that our method performs much better than the compared methods. **mdhI** and **infP** show good results as well. Table 4.2 summarises the results of these methods.

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

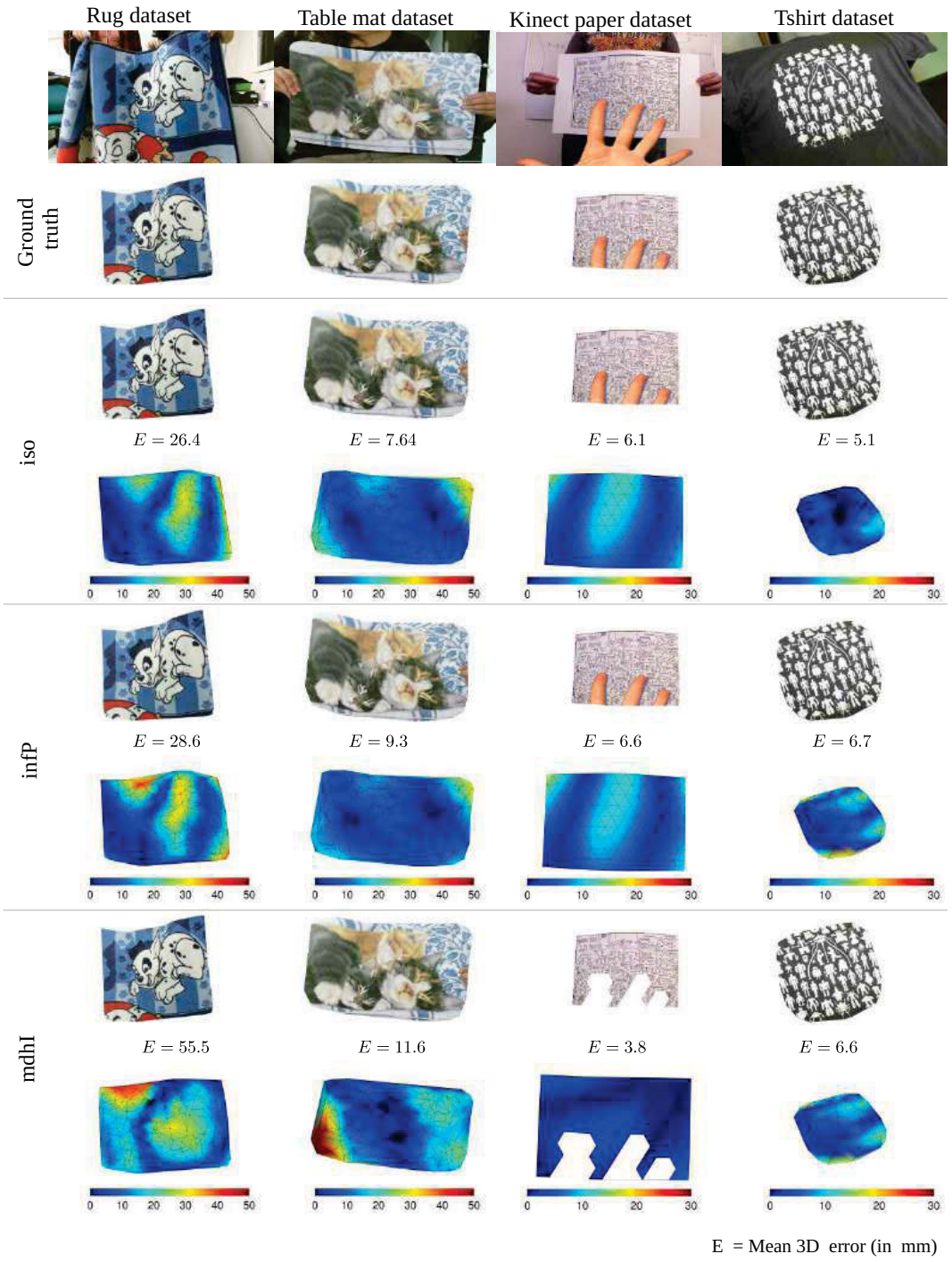
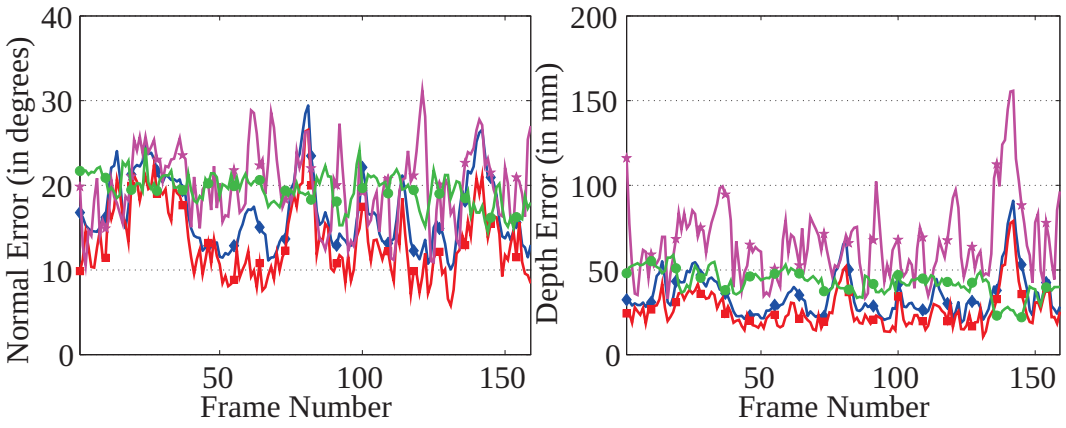
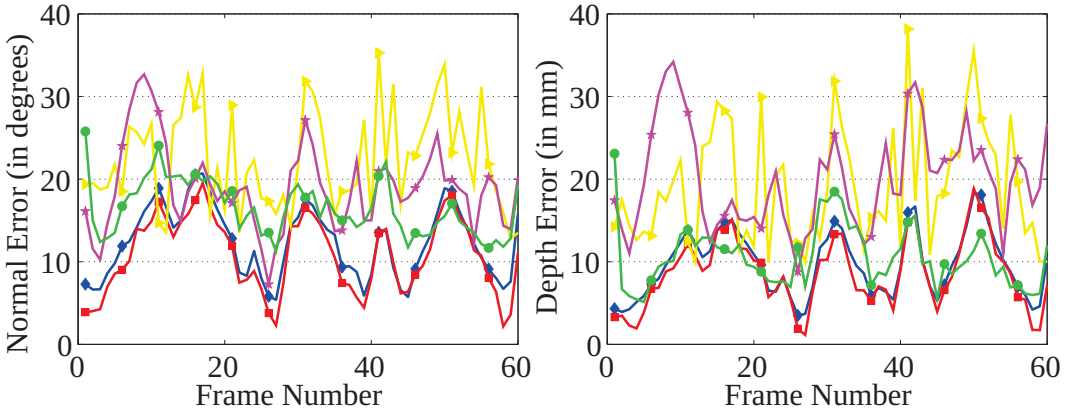


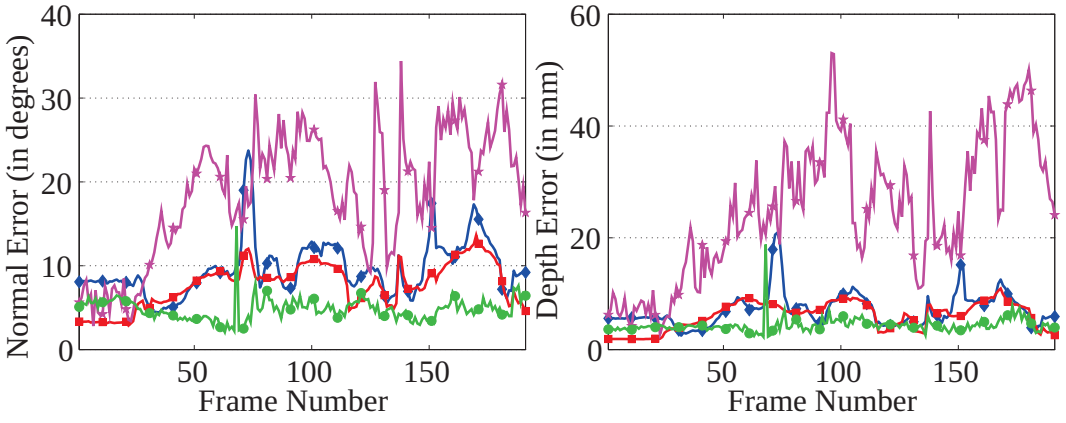
Figure 4.6: Reconstruction error maps and renderings for the rug, table mat, kinect paper and tshirt datasets. We remind that **mdhI** reconstructs only the visible part of the surface. Therefore, the rendering and error map for the kinect paper dataset is broken for this method. Best viewed in colour.



(a) Rug dataset



(b) Table mat dataset



(c) Kinect paper dataset

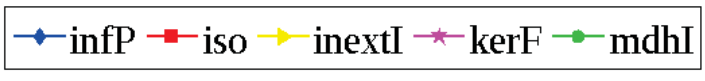


Figure 4.7: Experiments on long sequences. The average normal and depth error for each frame is shown. The experiment with **inextI** is only performed for the table mat dataset. Best viewed in colour.

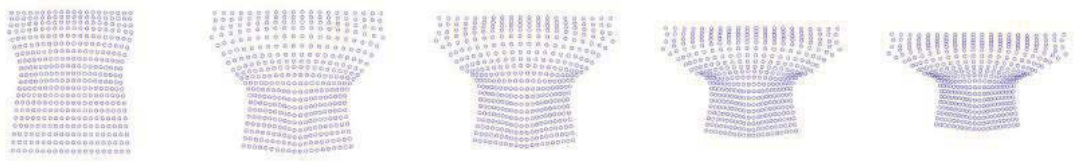


Figure 4.8: Images $(10, 20, 30, 40, 50)$ of a partially stretched rubber like surface. The first image has the least deformation and the last one has the most.

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

Table 4.2: Summary of experiments on long sequences. The average normal (E_n) and depth error (E_d), are measured in degrees and mm respectively and shown over the entire sequences.

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

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

Summary of experiments on long sequences. Table 4.2 summarises the results of the compared methods on the rug, table mat and kinect paper datasets. **iso** and **infP** give the best performance on rug and the table mat datasets while **mdhI** gives the best results on the kinect paper dataset. **kerF** gives decent results on the rug and table mat datasets but does not do well on the kinect paper dataset. However, its performance is always worse than **iso**, **infP** and **mdhI**. **inextI** was only compared on the table mat dataset; it gives the worst results compared to the other methods.

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

Table 4.3: Mean shape error (in degrees) for the experiment with partially stretched surface. The error increases with the number of images as the deformation increases. **iso** performs better than **infP**.

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

4.5.3 Elastic Objects

Our methods model deformations with isometry. In case of non-isometric deformations, theorem 2 and corollary 1 do not hold and therefore, there is no meaningful theoretical solution guaranteed. However, we solve Iso-NRSfM by finding a set of CS that minimise the sum of squares of polynomials in equation (4.45). Therefore, we made an experiment to test Iso-NRSfM with objects deforming non-isometrically in order to test the limitation for our methods **infP** and **iso**. We used the partially stretched surface dataset introduced in [Özgür and Bartoli, 2016]. It consists of 50 shapes of an elastic surface partially stretched from its longest side in a sequential order. The images are shown in figure 4.8. We made five experiments on this dataset. These experiments include 10, 20, 30, 40 and 50 images respectively. The experiment with 10 images has the least elastic deformation (this can be seen in figure 4.8) and the one with 50 images has the most elastic deformation. For each experiment, we calculate the shape error for each image. The mean shape error calculated over the entire image set in each experiment increases about linearly with the degree of extension. Therefore, we see that the method does not collapse completely for non-isometric deformations but gracefully degrades. Table 4.3 summarises the results of this experiment.

4.5.4 Computation Time Comparison

We compared the performance of our methods with the others in terms of computation time on a standard computer with 16 GB RAM. Ours and the rest of the methods are implemented

Figure 4.9: Experiment with an almost stationary object. The first five images of the table mat sequence are used. The reconstruction of **iso** is shown in red. The ground truth is indicated with black. E_s represents the mean shape error (in degrees). E_d represents the mean depth error (in mm). The performance of **iso** is almost the same as **infP** on these five images. Best viewed in colour.

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

Table 4.4: Comparison of computation time (in seconds) for 10, 30 and 60 views. The best performing method is highlighted in bold. **infP** and **iso** show a much lower increase in computation time while **plaH**, **diffH**, **pieceI** and **inextI** show a drastic increase. They could not be evaluated for 60 views.

in MATLAB. **infP** takes $\approx$ 10 seconds for any number of images. **iso** is initialised with **infP**, and the solution to the first and the second-order derivatives of $\alpha_i(\mathbf{x})$ is found iteratively upto 5 iterations. Solving for the first-order derivatives takes a similar duration as **infP** while solving for second-order derivatives and integrating normals have a linear complexity but they are very fast. We made an experiment with 10, 30 and 60 views. We observed that the computation time of **iso** ($\approx$ 60 seconds) and **infP** ($\approx$10 seconds) is almost the same (very small increase) in the three experiments. **mdhI** and **kerF** also are very fast but the computation time increases significantly as the number of images increases. **plaH**, **diffH**, **pieceI** and **inextI** show a drastic increase in computation time on changing the number of images from 10 to 30. We did not compute these timings for 60 images. Table 4.4 summarises the results.

4.5.5 Nearly-Stationary Objects

We made an experiment with a nearly-stationary object. In the table mat dataset, we picked the first 5 frames which are shown in figure 4.9. The mat is nearly-stationary. We observed that our methods **iso** and **infP** did get a decent reconstruction for these datasets. The errors are higher as compared to the experiments with the full sequence, as expected. Figure 4.9 shows the images and the reconstruction. The results are shown for **iso**. The performance of **infP** was almost the same as **iso**. This shows that the solution to Iso-NRSfM is well-posed. There is always a solution for $N \geq 3$, as long as the images are not exactly the same.

4.6 Conclusions

We proposed a theoretical framework for modelling and solving NRSfM locally for surfaces deforming isometrically. It uses Riemannian manifolds and applies to the minimal and redundant cases of $N \geq 3$ views. Unlike existing methods, the proposed method has only five variables to solve for N views. Therefore, it can handle a large number of views without a significant increase in the computation time. The complexity is linear, which is a substantial improvement from the current state-of-the-art. Since the method is local, it handles missing data and occlusions. However, it does not handle self-occlusions but they could be inferred a posteriori. We proposed two methods that solve NRSfM with and without the assumption of IP. The performance of these methods is quite similar (except in computation time). This shows that IP is a good assumption to make. We tested our methods on datasets with wide-baseline and short-baseline viewpoints, large and small deformations. Our results show that the proposed methods consistently give significantly better results than the state-of-the-art methods even for as few as three views.

Chapter 5

A Modelling Framework for Deformable 3D Reconstruction

Summary

In this chapter, we propose a generic modelling framework for deformable 3D reconstruction using differential geometry and Cartan's theory of connections. We express the properties of surfaces in terms of differential quantities such as moving frames and connections and show how to express the deformation constraints in terms of these quantities. With the assumption of IP, we derive PDE for the reconstruction of surfaces with isometric, conformal, skewless, equiareal deformations and solve them algebraically. We show that NRSfM is not solvable for equiareal deformations. We also obtain a solution for SfT without modelling deformation explicitly. This solution is derived under the assumption of IL.

5.1 Introduction

In the previous chapter, we obtained a solution to NRSfM (Iso-NRSfM) for deformable thin-shell objects undergoing isometric deformations. It uses the differential quantities of metric tensor and CS to model NRSfM. It is a significant improvement on the state-of-the-art. It is a local solution which has limited variables for any number of images, which makes it computationally very cheap. However, it gives very good results with very few images. It handles missing data and occlusions implicitly and works with both short and wide-baseline images. It shows that physics-based modelling in NRSfM leads to better solutions than existing ones.

In this chapter, we extend NRSfM to deformations other than isometry. Our goal is to propose local solutions which maintain the strengths of Iso-NRSfM. We generalise the concepts of metric tensor and CS from Riemannian geometry (defined for isometry and conformity only) to moving frames and connections in differential geometry. Differential geometry is a very good tool for physics-based modelling as in pure mathematics, it is well-established that it is the comprehensive study of surfaces. Cartan's theory of connections not only generalises the concept of CS on surfaces but also gives the laws that combine moving frames and connections in order to completely describe the properties of surfaces. It then leads to interesting results. With differential geometry and Cartan's connections, we propose a unified framework for modelling NRSfM and SfT. This framework is local and coherent with other solutions to NRSfM (Iso-NRSfM) and SfT [Bartoli et al., 2015] with differential modelling.

This framework allows us to go one step further in NRSfM and SfT and to model these problems for various types of deformations in terms of PDE. With the assumption of IP, we show that these PDE can be solved algebraically. We show that nonetheless equiareal NRSfM cannot be solved locally.

An interesting result in this chapter is the solvability of SfT using geometric constraints instead of deformation constraints. We show that the solution is unique and well-posed.

This chapter contributes a modelling framework for deformable 3D reconstruction. It is general, practical and easy to use.

Chapter outline. We present the mathematical background of this framework in section 5.2. It describes the connections, moving frames and shows how to express deformation constraints using these quantities. Section 5.3 proposes a framework for model-based reconstruction of deformable objects (SfT) and the reconstruction algorithm. Section 5.4 proposes a framework for model-free reconstruction of deformable objects (NRSfM) and the reconstruction algorithm. Section 5.5 discusses the experiments and section 5.6 concludes.

5.2 Mathematical Background

We describe surfaces and deformations using concepts from differential geometry and the theory of manifolds. We discuss next some of the deformations relevant to this framework.

5.2.1 Affine Connections

In differential geometry, a connection is a geometric object attached to a point on a smooth manifold. It transports some of the geometric properties (such as lengths, angles and areas) of the surface at this point to its neighbourhood.

Definition 1. *Cartan's affine connection is a set of geometric relations (expressed in terms of the affine moving frame defined locally at a point in space) that relate a point to its infinitesimal neighbourhood.*

In order to define an affine connection in an n-dimensional space, we fix an origin $\mathbf{O}$ and n-linearly independent basis vectors $(e_1, e_2, \ldots, e_n)$ originating from $\mathbf{O}$ which describe a frame at $\mathbf{f}$ as

$$\mathbf{f} = \mathbf{O} + f^1 e_1 + f^2 e_2 + ... + f^n e_n, \tag{5.1}$$

where f^t are the coordinates of the point $\mathbf{f}$ on the manifold. An affine connection is described using the following first order differential system which represents the vectorial variations of $\mathbf{f}$ and the coordinate basis:

$$\begin{aligned} \mathrm{d}\mathbf{f} &= w^1 e_1 + w^2 e_2 + \ldots + w^n e_n \\ \mathrm{d}e_t &= w_t^1 e_1 + w_t^2 e_2 + \ldots + w_t^n e_n, \end{aligned} \tag{5.2}$$

where the 1-forms (see appendix C for details) w^t and w_t^s are known as *dual* and *connection* forms respectively. Note that this formulation of connections is applicable to any kind of frame. In case of moving frames, a connection 1-form describes the change in the moving frame as one moves to the infinitesimal neighbourhood of a point on the manifold. An affine connection defines the local geometric and physical properties (such as unit lengths and areas) of the affine space around a point in the manifold. Cartan's vision behind connections [Cartan, 1923, 1924, 1926] was to define geometric properties on an object without defining the object itself. He derived these laws using the theory of moving frames [Cartan, 1937] which we describe next.

5.2.2 Moving Frames on Surfaces

The moving frame in a 3D surface $\mathcal{M}$ is a set of 3 linearly independent vectors. It can be defined in several ways. The connection forms defined in equation (5.2) are dependent on the particular choice of the moving frame. In general we assume that any point on the surface (manifold) admits a local parametrisation described by the function $f(x^1, x^2)$ (see figure 5.1). A natural choice is to use the tangent vectors of the surface to define the moving frame:

$$e_1 = \frac{\partial f}{\partial x^1} \qquad e_2 = \frac{\partial f}{\partial x^2} \qquad e_3 = \frac{\partial f}{\partial x^1} \times \frac{\partial f}{\partial x^2}, \tag{5.3}$$

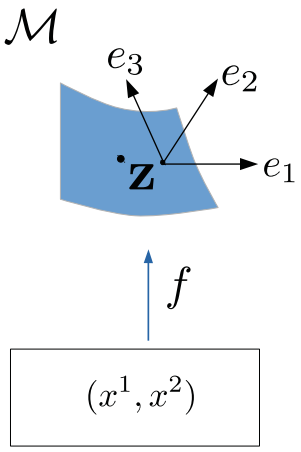


Figure 5.1: A moving frame on a surface $\mathcal{M}$ defined using a local parametrisation (x^1, x^2).

with $e_1^\top e_3 = 0$ and $e_2^\top e_3 = 0$. The expression of the total derivative of $\mathbf{f}$ in terms of the moving frame is given by

$$\mathrm{d}\mathbf{f} = \frac{\partial f}{\partial x^1}\mathrm{d}x^1 + \frac{\partial f}{\partial x^2}\mathrm{d}x^2, \tag{5.4}$$

By identifying the terms in equation (5.4) with equation (5.2), we obtain the dual forms $w^1 = \mathrm{d}x^1$, $w^2 = \mathrm{d}x^2$ and $w^3 = 0$. The connection forms can be found by taking the total derivative of the basis vectors

$$\mathrm{d}e_t = \frac{\partial e_t}{\partial x^1}\mathrm{d}x^1 + \frac{\partial e_t}{\partial x^2}\mathrm{d}x^2, \tag{5.5}$$

and finding the representation of equation (5.5) in the basis formed by the moving frame (e_1, e_2, e_3):

$$\mathrm{d}e_t = w_t^1 e_1 + w_t^2 e_2 + w_t^3 e_3, \tag{5.6}$$

where $w_t^s = \Gamma_{t1}^s \mathrm{d}x^1 + \Gamma_{t2}^s \mathrm{d}x^2$ and Γ_{tk}^s are scalar functions. In case of the frame used in equation (5.3), Γ_{tk}^s for $s, t, k \in \{1, 2\}$ are CS of the surface and Γ_{tk}^s with $s = 3$ and $t, k \in \{1, 2\}$ contain the coefficients of the second fundamental form of the surface.

Example 1. *Given the following surface of a plane:*

$$f(x^1, x^2) = \mathbf{u}x^1 + \mathbf{v}x^2 + \mathbf{o}, \tag{5.7}$$

where $\mathbf{u}$ *and* $\mathbf{v}$ *are three-dimensional unit vectors that define the plane and* $\mathbf{o}$ *is a three-dimensional displacement vector, we define the moving frame of equation* (5.3)*:*

$$e_1 = \mathbf{u} \quad e_2 = \mathbf{v} \quad e_3 = \mathbf{u} \times \mathbf{v}. \tag{5.8}$$

The total derivative is given by

$$\mathrm{d}\mathbf{f} = \begin{pmatrix} \mathrm{d}x^1 & \mathrm{d}x^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix}^\top = \begin{pmatrix} \omega^1 & \omega^2 \end{pmatrix} \begin{pmatrix} e_1 & e_2 \end{pmatrix}^\top. \tag{5.9}$$

As e_1 and e_2 are constant vectors then $\mathrm{d}e_1 = 0$ and $\mathrm{d}e_2 = 0$. This makes all connection forms $w_t^s = 0$ for $t \in \{1, 2\}$ and $s \in \{1, 2, 3\}$. We remind that this result is correct for the particular parametrization of the plane given in equation (5.7) and the moving frame in equation (5.8).

Example 2. *If the plane of equation (5.7) is projected into the image with coordinates $(\bar{x}_1, \bar{x}_2)$, it can be alternatively parametrised using the image embedding ϕ from equation (3.3) as*

$$\phi = \frac{1}{\beta(\bar{x}^1, \bar{x}^2)} \begin{pmatrix} \bar{x}^1 \\ \bar{x}^2 \\ 1 \end{pmatrix}, \tag{5.10}$$

with

$$\beta(\bar{x}^1, \bar{x}^2) = -\frac{\mathbf{n}^\top \begin{pmatrix} \bar{x}^1 \\ \bar{x}^2 \\ 1 \end{pmatrix}}{\mathbf{n}^\top \mathbf{o}} \qquad \mathbf{n} = \mathbf{u} \times \mathbf{v}. \tag{5.11}$$

The moving frame on the surface $\mathcal{M}$ described by the image embedding (5.11) is given by

$$\begin{aligned} e_1 = \phi_1 &= \frac{1}{\beta^2} \left(\beta - \bar{x}^1\beta_1, -\bar{x}^2\beta_1, -\beta_1\right)^\top \\ e_2 = \phi_2 &= \frac{1}{\beta^2} \left(-\bar{x}^1\beta_2, \beta - \bar{x}^2\beta_2, -\beta_2\right)^\top \\ e_3 = \phi_1 \times \phi_2 &= \frac{1}{\beta^3} \left(\beta_1, \beta_2, \beta - \bar{x}^1\beta_1 - \bar{x}^2\beta_2\right)^\top. \end{aligned} \tag{5.12}$$

Using the frame and its first-order derivatives, we obtain the connection forms w_t^s related to ϕ. We write $w_t^s = \Gamma_{tk}^s d\bar{x}^k$ and obtain

$$\begin{aligned} \begin{pmatrix} \Gamma_{11}^1 & \Gamma_{11}^2 & \Gamma_{11}^3 \\ \Gamma_{21}^1 & \Gamma_{21}^2 & \Gamma_{21}^3 \\ \Gamma_{31}^1 & \Gamma_{31}^2 & \Gamma_{31}^3 \end{pmatrix} &= -\frac{1}{\beta} \begin{pmatrix} 2\beta_1 & 0 & 0 \\ \beta_2 & \beta_1 & 0 \\ 0 & 0 & 3\beta_1 \end{pmatrix} \\ \begin{pmatrix} \Gamma_{12}^1 & \Gamma_{12}^2 & \Gamma_{12}^3 \\ \Gamma_{22}^1 & \Gamma_{22}^2 & \Gamma_{22}^3 \\ \Gamma_{32}^1 & \Gamma_{32}^2 & \Gamma_{32}^3 \end{pmatrix} &= -\frac{1}{\beta} \begin{pmatrix} \beta_2 & \beta_1 & 0 \\ 0 & 2\beta_2 & 0 \\ 0 & 0 & 3\beta_2 \end{pmatrix}. \end{aligned} \tag{5.13}$$

As expected, the connections in equation (5.13) are different from those of Example 1. Equation (5.13) plays an important role in this paper as it shows the general structure of the connection's coefficients for planar surfaces and by extension the IP approximation of surfaces described by their image embeddings.

We now show how to write moving frames and connections on planar surfaces.

5.2.3 Moving Frames and Parametrisations

Figure 5.2 shows a surface $\mathcal{M}$ defined using two different parametrizations $f(\mathbf{x})$ and $g(\bar{\mathbf{x}})$. The domains of f and g are mutually homeomorphic and η is the diffeomorphic mapping that expresses the change of coordinates:

$$\mathbf{x} = \eta(\bar{\mathbf{x}}) = \begin{pmatrix} \eta^1(\bar{\mathbf{x}}) \\ \eta^2(\bar{\mathbf{x}}) \end{pmatrix}. \tag{5.14}$$

By differentiating, we obtain the relation between the dual forms of the two parametrisations as

$$\begin{aligned} \mathbf{d}x^1 &= \frac{\partial x^1}{\partial \bar{x}^1}\mathbf{d}\bar{x}^1 + \frac{\partial x^1}{\partial \bar{x}^2}\mathbf{d}\bar{x}^2 \\ \mathbf{d}x^2 &= \frac{\partial x^2}{\partial \bar{x}^1}\mathbf{d}\bar{x}^1 + \frac{\partial x^2}{\partial \bar{x}^2}\mathbf{d}\bar{x}^2. \end{aligned} \tag{5.15}$$

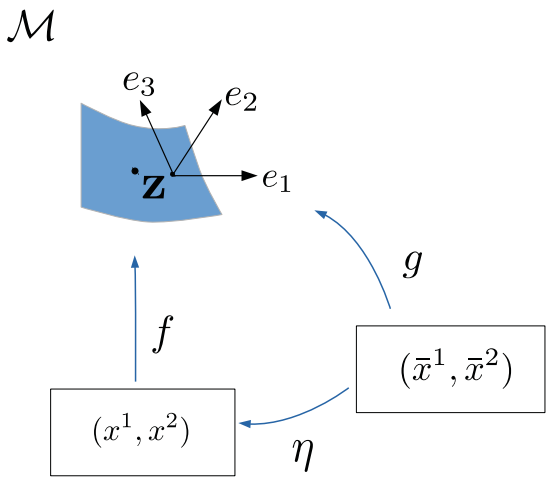


Figure 5.2: A moving frame on a surface $\mathcal{M}_i$ defined using local parametrisations (x^1, x^2) and $(\bar{x}^1, \bar{x}^2)$ related by η.

Given that $g = f \circ \eta$ in figure 5.2, we obtain $\mathbf{J}_g = \mathbf{J}_f \mathbf{J}_\eta$. Therefore, the moving frames vectors $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ and (e_1, e_2, e_3) defined using $(\bar{x}^1, \bar{x}^2)$ and (x^1, x^2) respectively are related by the following relationship

$$\begin{aligned} \begin{pmatrix} \bar{e}_1 & \bar{e}_2 \end{pmatrix} &= \begin{pmatrix} e_1 & e_2 \end{pmatrix} \mathbf{J}_\eta \\ \bar{e}_3 = \bar{e}_1 \times \bar{e}_2 &= |\mathbf{J}_\eta|(e_1 \times e_2) = |\mathbf{J}_\eta| e_3 \\ \begin{pmatrix} \bar{e}_1 & \bar{e}_2 & \bar{e}_3 \end{pmatrix} &= \begin{pmatrix} \begin{pmatrix} e_1 & e_2 \end{pmatrix} \mathbf{J}_\eta & |\mathbf{J}_\eta| e_3 \end{pmatrix}. \end{aligned} \tag{5.16}$$

Differentiating the above relation and expressing the derivatives of frame bases according to

equation (5.2), we obtain

$$\begin{aligned}\begin{pmatrix}\bar{e}_1 & \bar{e}_2 & \bar{e}_3\end{pmatrix}\begin{pmatrix}\bar{w}_1^1 & \bar{w}_2^1\\ \bar{w}_1^2 & \bar{w}_2^2\\ \bar{w}_1^3 & \bar{w}_2^3\end{pmatrix} &= \begin{pmatrix}e_1 & e_2 & e_3\end{pmatrix}\begin{pmatrix}w_1^1 & w_2^1\\ w_1^2 & w_2^2\\ w_1^3 & w_2^3\end{pmatrix}\mathbf{J}_\eta + \begin{pmatrix}e_1 & e_2\end{pmatrix}\mathbf{d}\mathbf{J}_\eta\\ \begin{pmatrix}\bar{e}_1 & \bar{e}_2 & \bar{e}_3\end{pmatrix}\begin{pmatrix}\bar{w}_3^1\\ \bar{w}_3^2\\ \bar{w}_3^3\end{pmatrix} &= \begin{pmatrix}e_1 & e_2 & e_3\end{pmatrix}\begin{pmatrix}w_3^1\\ w_3^2\\ w_3^3\end{pmatrix}|\mathbf{J}_\eta| + e_3\mathbf{d}|\mathbf{J}_\eta|.\end{aligned} \tag{5.17}$$

In these equations, we express $\begin{pmatrix}e_1 & e_2 & e_3\end{pmatrix}$ as $\left(\begin{pmatrix}e_1 & e_2\end{pmatrix}\mathbf{J}_\eta \quad |\mathbf{J}_\eta|e_3\right)\begin{pmatrix}\mathbf{J}_\eta^{-1}\\ |\mathbf{J}_\eta|^{-1}\end{pmatrix}$ and using equation (5.16), we write the relation between the connection forms as

$$\begin{aligned}\begin{pmatrix}\bar{w}_1^1 & \bar{w}_2^1\\ \bar{w}_1^2 & \bar{w}_2^2\end{pmatrix} &= \mathbf{J}_\eta^{-1}\begin{pmatrix}w_1^1 & w_2^1\\ w_1^2 & w_2^2\end{pmatrix}\mathbf{J}_\eta + \mathbf{J}_\eta^{-1}\mathbf{d}\mathbf{J}_\eta\\ \begin{pmatrix}\bar{w}_1^3 & \bar{w}_2^3\end{pmatrix} &= |\mathbf{J}_\eta|^{-1}\begin{pmatrix}w_1^3 & w_2^3\end{pmatrix}\mathbf{J}_\eta\\ \begin{pmatrix}\bar{w}_3^1\\ \bar{w}_3^2\end{pmatrix} &= \mathbf{J}_\eta^{-1}\begin{pmatrix}w_3^1 & w_3^2\end{pmatrix}|\mathbf{J}_\eta|\\ \bar{w}_3^3 &= w_3^3 + \mathbf{d}\left(|\mathbf{J}_\eta|\right)|\mathbf{J}_\eta|^{-1}.\end{aligned} \tag{5.18}$$

Equations (5.16) and (5.18) show that the moving frame and the connections of the 3D surface $\mathcal{M}$ (in figure 5.2) derived using the functions f and g are linearly related in terms of the first and second-order derivatives of η. We refer to these equations as the change of variable equations of moving frames and connections. In the next section, we show how to draw relations between different surfaces.

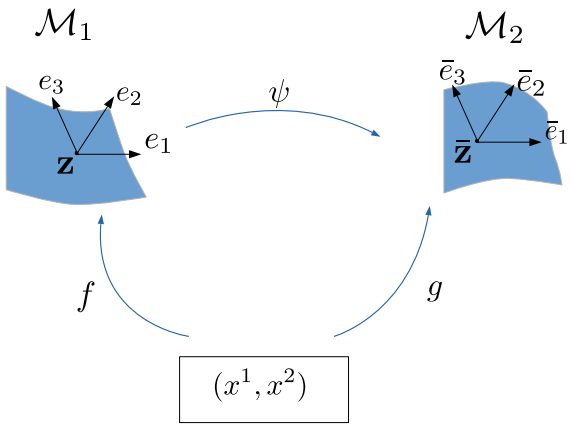


Figure 5.3: Two surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$ related by ψ are parametrised using (x^1, x^2).

5.2.4 Smooth Mappings between Surfaces

Mappings are defined by functions that connect points between two surfaces. Figure 5.3 shows a mapping $\psi : \mathcal{M}_1 \to \mathcal{M}_2$ between surface $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively parametrized by embeddings $f(\mathbf{x})$ and $g(\mathbf{x})$. The mapping ψ takes a point $\mathbf{z}$ on $\mathcal{M}_1$ and transports it to $\bar{\mathbf{z}}$ on $\mathcal{M}_2$. Therefore, $\bar{\mathbf{z}} = \psi(\mathbf{z})$. We assume that mappings between surfaces are diffeomorphic which implies that they are smooth, bijective and with a smooth inverse.

Among this general class of mappings we study the following types: *1)* isometric (distances-preserving), *2)* conformal (angle-preserving), *3)* equiareal (area-preserving) and *4)* skewless (orthogonal frame basis' angle-preserving). The set of isometric mappings is a subset of 2), 3) and 4) and is also given by the intersection of 2) and 3). The set of skewless mappings includes 1), 2) and a subset of 3). These properties are described in figure 5.4. All these kinds of mappings are identified by how they affect the metric tensor (first fundamental form) between the two surfaces. They are thus defined with first order differential constraints.

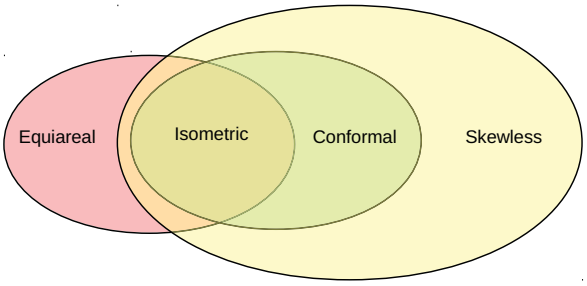


Figure 5.4: Classification of various types of smooth mappings.

Given that $\mathbf{g}$ and $\bar{\mathbf{g}}$ are the metric tensors of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively, where:

$$\mathbf{g} = \mathbf{J}_f^\top \mathbf{J}_f \qquad \bar{\mathbf{g}} = \mathbf{J}_g^\top \mathbf{J}_g \tag{5.19}$$

the four categories of mappings are described by the following invariants:

Type of mapping	*Invariant*
Isometric	$\mathbf{g} = \bar{\mathbf{g}}$
Conformal	$\mathbf{g} \propto \bar{\mathbf{g}}$
Equiareal	$\lvert\mathbf{g}\rvert = \lvert\bar{\mathbf{g}}\rvert$
Skewless	$\gamma(\mathbf{g}) = \gamma(\bar{\mathbf{g}})$

(5.20)

where

$$\gamma(\mathbf{g}) = \frac{\mathbf{g}_{12}^2}{\mathbf{g}_{11}\mathbf{g}_{22}} \tag{5.21}$$

Given the moving frames (e_1, e_2, e_3) at $\mathcal{M}_1$ and $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ at $\mathcal{M}_2$ from equation (5.3), the coefficients of the metric tensor can be described as follows from equation (5.19):

$$\mathbf{g} = \begin{pmatrix} e_1^\top e_1 & e_1^\top e_2 \\ e_2^\top e_1 & e_2^\top e_2 \end{pmatrix} \quad \bar{\mathbf{g}} = \begin{pmatrix} \bar{e}_1^\top \bar{e}_1 & \bar{e}_1^\top \bar{e}_2 \\ \bar{e}_2^\top \bar{e}_1 & \bar{e}_2^\top \bar{e}_2 \end{pmatrix} \tag{5.22}$$

From equations (5.22) and (5.20), the four categories of mappings can be transferred to constraints on the frames.

Isometric Mappings

Isometric mappings preserve the lengths and angles of the moving frames on the corresponding points on the two surfaces. The constraints are

$$\bar{e}_1^\top \bar{e}_1 = e_1^\top e_1 \qquad \bar{e}_2^\top \bar{e}_2 = e_2^\top e_2 \qquad \bar{e}_1^\top \bar{e}_2 = e_1^\top e_2. \tag{5.23}$$

Conformal Mappings

Conformal mappings preserve the angles of the moving frames on the corresponding points on the two surfaces but the lengths of the frame vectors are isotropically scaled. The constraints are

$$\begin{pmatrix} \bar{e}_1^\top \bar{e}_1 & \bar{e}_2^\top \bar{e}_2 & \bar{e}_1^\top \bar{e}_2 \end{pmatrix} \propto \begin{pmatrix} e_1^\top e_1 & e_2^\top e_2 & e_1^\top e_2 \end{pmatrix}. \tag{5.24}$$

Equiareal Mappings

Equiareal mappings preserve the area (expressed as the squared norm of the cross product of tangent vectors) of the tangent plane defined by the moving frames on the corresponding points on the two surfaces. The constraint is

$$\begin{aligned} |\bar{e}_1 \times \bar{e}_2|^2 &= \left(\bar{e}_1^\top \bar{e}_1\right)\left(\bar{e}_2^\top \bar{e}_2\right) - \left(\bar{e}_1^\top \bar{e}_2\right)^2 \\ &= \left(e_1^\top e_1\right)\left(e_2^\top e_2\right) - \left(e_1^\top e_2\right)^2 = |e_1 \times e_2|^2 . \end{aligned} \tag{5.25}$$

Skewless Mappings

Skewless mappings preserve the angles along the orthogonal frame basis on the surface. It is composed of two local anisotropic scaling along the orthogonal frame basis followed by a conformal mapping (see figure 5.5 for more details). Next we prove a theorem to formalise the construction of these mappings.

Theorem 5 (Skewless Mappings)**.** *A mapping is skewless iff it can be only decomposed into a conformal mapping and two anisotropic scaling along the orthogonal frame basis.*

Proof. First we prove the reverse implication of the theorem, i.e., two anisotropic scaling and a conformal mapping lead to a skewless mapping. In figure 5.5, we have

$$\psi_{xyc} = \psi_c \circ \psi_y \circ \psi_x. \tag{5.26}$$

Since ψ_x and ψ_y are anisotropic local scaling along the orthogonal frame basis, they preserve the angles along the orthonormal frame basis. Therefore, ψ_x, ψ_y and their composition is a

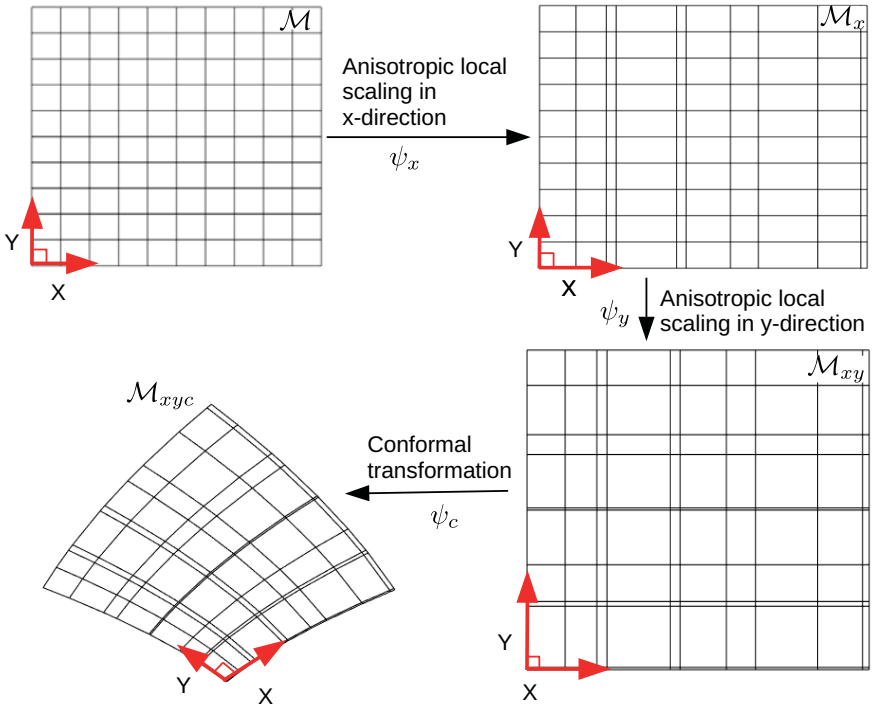


Figure 5.5: An example of skewless deformation. A surface grid undergoes anisotropic scaling in two orthogonal directions and then undergoes a conformal transformation. Therefore, only the angles between the basis are preserved.

skewless mapping. ψ_c is a conformal mapping and therefore, it is skewless too, which makes ψ_{xyc} a skewless mapping.

Note that $\psi_c \circ \psi_y \circ \psi_x$ and $\psi_c \circ \psi_x \circ \psi_y$ are both skewless mappings as the anisotropic scalings are commutative. However, $\psi_x \circ \psi_c \circ \psi_y$ is not a skewless mapping anymore due to the non-commutativity of the conformal mapping.

In order to prove the forward implication, we need to show that a skewless mapping ψ_{xyc} can always be decomposed into a conformal mapping ψ_c and two anisotropic local scalings along the orthogonal frame basis.

We can always express ψ_{xyc} as

$$\psi_{xyc} = \psi_c \circ \psi_y \circ \psi_x \circ \psi_u, \tag{5.27}$$

where ψ_u is an unknown mapping. On decomposing ψ_{xyc} as in equation (5.27), we have that $\psi_y \circ \psi_x \circ \psi_u$ must be a skewless mapping. ψ_u cannot be a conformal mapping as it is non-commutative with anisotropic scalings. ψ_u can be expressed as

$$\psi_u = \psi_{x'} \circ \psi_{y'} \circ \psi_{x'y'}, \tag{5.28}$$

where $\psi_{x'}$ and $\psi_{y'}$ are transformations along the orthogonal frame basis. $\psi_{x'y'}$ represents a transformation that is not along the orthogonal frame basis and therefore it does not preserve

the angles along the orthogonal frame basis.

Therefore $\psi_y \circ \psi_x \circ \psi_u$ can only be a skewless mapping if $\psi_{x'y'}$ is identity. Otherwise, it causes a scaling which is not along the orthogonal frame basis which makes $\psi_y \circ \psi_x \circ \psi_u$ a non-skewless mapping.

Hence a skewless mapping can only be defined as a combination of anisotropic scaling followed by a conformal mapping.

□

The constraint for skewless mappings is

$$\frac{\left(\bar{e}_1^\top \bar{e}_2\right)^2}{\left(\bar{e}_1^\top \bar{e}_1\right)\left(\bar{e}_2^\top \bar{e}_2\right)} = \frac{\left(e_1^\top e_2\right)^2}{\left(e_1^\top e_1\right)\left(e_2^\top e_2\right)}. \tag{5.29}$$

It is possible to propose more kinds of mappings and express the properties they preserve in terms of the moving frames. In the next section we will show how to exploit these properties (expressed in terms of moving frames) for the reconstruction of deformable objects.

5.2.5 Infinitesimally Linear Mappings between Surfaces

In this section we show that linear mappings preserve the connection forms across surfaces. This is a very important property that we use in this chapter to propose a reconstruction algorithm (without having to categorise the mapping according to equation (5.20)) by extending these properties to the so-called infinitesimally linear maps.

Theorem 6. *(Linear Mappings) A linear mapping between two planes preserves the connection forms* $(w_1^1, w_2^1, w_1^2, w_2^2)$.

Proof. Given that $g = \psi \circ f$ in figure 5.3, we have $\mathbf{J}_g = 0$. The moving frames (e_1, e_2, e_3) and $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ on the planes $\mathcal{M}_1$ and $\mathcal{M}_2$ are related by

$$\begin{pmatrix} \bar{e}_1 & \bar{e}_2 \end{pmatrix} = \mathbf{J}_{\psi\circ f} \begin{pmatrix} e_1 & e_2 \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{\psi\circ f} e_1 & \mathbf{J}_{\psi\circ f} e_2 \end{pmatrix}. \tag{5.30}$$

Equation (5.30) shows the relation between the moving frames (e_1, e_2, e_3) at $\mathcal{M}_1$ and $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ at $\mathcal{M}_2$. Therefore, on differentiating this relation we obtain

$$\begin{pmatrix} \mathrm{d}\bar{e}_1 & \mathrm{d}\bar{e}_2 \end{pmatrix} = \mathbf{J}_{\psi\circ f} \begin{pmatrix} \mathrm{d}e_1 & \mathrm{d}e_2 \end{pmatrix} + \mathrm{d}\mathbf{J}_{\psi\circ f} \begin{pmatrix} e_1 & e_2 \end{pmatrix}. \tag{5.31}$$

Given that ψ is a linear mapping, $\mathrm{d}\mathbf{J}_{\psi\circ f} = 0$ in the previous equation. We multiply the two expressions obtained in the above equation with $\bar{e}_1^\top$ and expand the expression using the connection relations in equation (5.2) and obtain

$$\begin{aligned} \bar{w}_1^1 \bar{e}_1 \bar{e}_1^\top + \bar{w}_1^2 \bar{e}_2 \bar{e}_1^\top &= \mathbf{J}_{\psi\circ f} \left(w_1^1 e_1 + w_1^2 e_2\right) \bar{e}_1^\top \\ \bar{w}_2^1 \bar{e}_1 \bar{e}_1^\top + \bar{w}_2^2 \bar{e}_2 \bar{e}_1^\top &= \mathbf{J}_{\psi\circ f} \left(w_2^1 e_1 + w_2^2 e_2\right) \bar{e}_1^\top. \end{aligned} \tag{5.32}$$

Since w_s^t are scalar functions (they are differential forms), we employ equation (5.30) and rewrite the above expression as

$$\begin{aligned}\bar{w}_1^1\bar{e}_1\bar{e}_1^\top + \bar{w}_1^2\bar{e}_2\bar{e}_1^\top &= w_1^1\bar{e}_1\bar{e}_1^\top + w_1^2\bar{e}_2\bar{e}_1^\top \\ \bar{w}_2^1\bar{e}_1\bar{e}_1^\top + \bar{w}_2^2\bar{e}_2\bar{e}_1^\top &= w_2^1\bar{e}_1\bar{e}_1^\top + w_2^2\bar{e}_2\bar{e}_1^\top .\end{aligned} \tag{5.33}$$

Since $(\bar{e}_1, \bar{e}_2)$ are linearly independent, on expressing w_t^s in terms of $\Gamma_{tk}^s \mathrm{d}x^k$ in the above equation we obtain $\bar{\Gamma}_{tk}^s = \Gamma_{tk}^s$ and therefore, $\begin{pmatrix}\bar{w}_1^1 & \bar{w}_2^1 & \bar{w}_1^2 & \bar{w}_2^2\end{pmatrix} = \begin{pmatrix}w_1^1 & w_2^1 & w_1^2 & w_2^2\end{pmatrix}$. □

We use the concept of IL mappings which allows us to extend the result of Theorem 6 for mappings between two generic surfaces. To summarise, it is possible to find constraints on smooth mappings (with an assumption of IL) without adding a deformation prior (like isometry or conformity).

In the next section, we show how to use theorem 6, the moving frame constraints derived for smooth mappings in equations (5.23)-(5.25) and the change of variable for moving frames in equation (5.16) and connections in equation (5.18) for 3D reconstruction of deformable objects.

5.3 Model-Based 3D Reconstruction

We now propose a general framework based on the theory of connections presented in the previous section to model and to solve SfT for thin-shell objects.

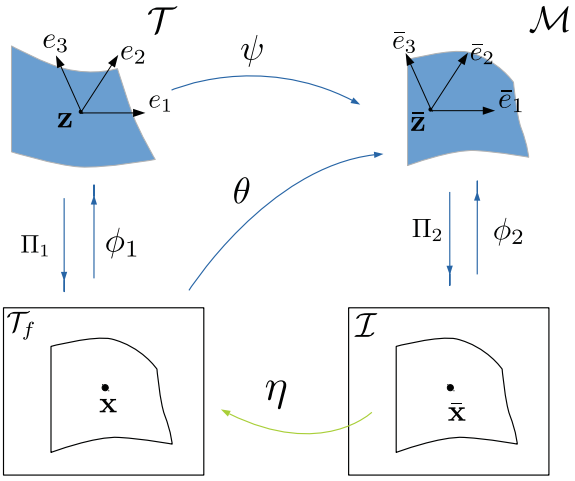


Figure 5.6: Modelling of model-based 3D reconstruction of deformable objects from a single view.

Figure 5.6 shows the general modelling of SfT. Given a 3D model $\mathcal{T}$ (template) of the object, our goal is to find the surface $\mathcal{M}$ as observed in the image $\mathcal{I}$. η is the image warp from $\mathcal{I}$ to the flattened 3D model $\mathcal{T}_f$. η is known and can be estimated in practice with dense image registration methods. ψ is the deformation function between $\mathcal{T}$ and $\mathcal{M}$. For rigid objects ψ is a Euclidean transformation. We model the surface $\mathcal{M}$ and the model $\mathcal{T}$ using image embeddings ϕ_1 and ϕ_2:

$$\phi_1(x^1, x^2) = \frac{1}{\beta_1(x^1, x^2)} \begin{pmatrix} x^1 \\ x^2 \\ 1 \end{pmatrix} \qquad \phi_2(\bar{x}^1, \bar{x}^2) = \frac{1}{\bar{\beta}_2(\bar{x}^1, \bar{x}^2)} \begin{pmatrix} \bar{x}^1 \\ \bar{x}^2 \\ 1 \end{pmatrix} \tag{5.34}$$

Following equation (5.3) the moving frames (e_1^t, e_2^t, e_3^t) and $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ on $\mathcal{T}$ and $\mathcal{M}$ respectively are given as

$$\begin{aligned} e_1^t = \phi_{11} &= \frac{1}{\beta_1^2} \left(\beta_1 - x^1\beta_{11}, -x^2\beta_{11}, -\beta_{11}\right)^\top \\ e_2^t = \phi_{12} &= \frac{1}{\beta_1^2} \left(-x^1\beta_{12}, \beta_1 - x^2\beta_{12}, -\beta_{12}\right)^\top \\ e_3^t = \phi_{11} \times \phi_{12} &= \frac{1}{\beta_1^3} \left(\beta_{11}, \beta_{12}, \beta_1 - x^1\beta_{11} - x^2\beta_{12}\right)^\top, \end{aligned} \tag{5.35}$$

$$\begin{aligned} \bar{e}_1 = \bar{\phi}_{21} &= \frac{1}{\bar{\beta}_2^2} \left(\bar{\beta}_2 - \bar{x}^1\bar{\beta}_{21}, -\bar{x}^2\bar{\beta}_{21}, -\bar{\beta}_{21}\right)^\top \\ \bar{e}_2 = \bar{\phi}_{22} &= \frac{1}{\bar{\beta}_2^2} \left(-\bar{x}^1\bar{\beta}_{22}, \bar{\beta}_2 - \bar{x}^2\bar{\beta}_{22}, -\bar{\beta}_{22}\right)^\top \\ \bar{e}_3 = \bar{\phi}_{21} \times \phi_{22} &= \frac{1}{\bar{\beta}_2^3} \left(\bar{\beta}_{21}, \bar{\beta}_{22}, \bar{\beta}_2 - \bar{x}^1\bar{\beta}_{21} - \bar{x}^2\bar{\beta}_{22}\right)^\top. \end{aligned} \tag{5.36}$$

Using equation (5.22) we compute the elements of the metric tensor in $\mathcal{T}$ as:

$$\begin{aligned} \left(e_1^t\right)^\top e_1^t &= \frac{1}{k^2}\left(\epsilon^2 k_1^2 + 1 - 2x^1 k_1\right) = \frac{E_t}{k^2} \\ \left(e_2^t\right)^\top e_2^t &= \frac{1}{k^2}\left(\epsilon^2 k_2^2 + 1 - 2x^2 k_2\right) = \frac{G_t}{k^2} \\ \left(e_1^t\right)^\top e_2^t &= \frac{1}{k^2}\left(\epsilon^2 k_1 k_2 - x^1 k_2 - x^2 k_1\right) = \frac{F_t}{k^2}, \end{aligned} \tag{5.37}$$

where $\epsilon^2 = \left(1 + \left(x^1\right)^2 + \left(x^2\right)^2\right)$, $k = \beta_1$, $k_1 = \dfrac{\beta_{11}}{\beta_1}$ and $k_2 = \dfrac{\beta_{12}}{\beta_1}$. Under the assumption of infinitesimal planarity, the first-order derivatives of these expressions are given by

$$\begin{aligned} \frac{\partial E_t}{\partial x^1} &= -2k_1 E_t \quad \frac{\partial F_t}{\partial x^1} = -k_2 E_t - k_1 F_t \quad \frac{\partial G_t}{\partial x^1} = -2k_2 F_t \\ \frac{\partial E_t}{\partial x^2} &= -2k_1 F_t \quad \frac{\partial F_t}{\partial x^2} = -k_1 G_t - k_2 F_t \quad \frac{\partial G_t}{\partial x^2} = -2k_2 G_t. \end{aligned} \tag{5.38}$$

Similarly, we can write $(\bar{E}, \bar{F}, \bar{G})$ and their first-order derivatives for $(\bar{e_1}, \bar{e}_2, \bar{e}_3)$ at $\mathcal{M}$ in terms of $(\bar{k}_1, \bar{k}_2, \bar{k})$.

In order to compare the moving frames at the two surfaces $\mathcal{T}$ and $\mathcal{M}$, we need to define them in the same parametrisation space. Therefore, we derive (E, F, G) at $\mathcal{M}_1$ in terms of $(\bar{x}^1, \bar{x}^2)$ by using the change of variable as suggested in equation (5.16)

$$
\begin{aligned}
E &= \left(\frac{\partial x^1}{\partial \bar{x}^1}\right)^2 E_t + 2\frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial x^2}{\partial \bar{x}^1}F_t + \left(\frac{\partial x^2}{\partial \bar{x}^1}\right)^2 G_t \\
F &= \frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial x^1}{\partial \bar{x}^2}E_t + \left(\frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial x^2}{\partial \bar{x}^2} + \frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial x^1}{\partial \bar{x}^2}\right)F_t + \left(\frac{\partial x^2}{\partial \bar{x}^1}\right)^2 G_t \\
G &= \left(\frac{\partial x^1}{\partial \bar{x}^2}\right)^2 E_t + 2\frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial x^2}{\partial \bar{x}^2}F_t + \left(\frac{\partial x^2}{\partial \bar{x}^2}\right)^2 G_t.
\end{aligned} \tag{5.39}
$$

The derivatives of (E, F, G) according to equation (5.38) are given by

$$
\begin{aligned}
\frac{\partial E}{\partial \bar{x}^1} &= -2\left(\frac{\partial x^1}{\partial \bar{x}^1}k_1 + \frac{\partial x^2}{\partial \bar{x}^1}k_2\right)E + 2\frac{\partial^2 x^1}{\partial\left(\bar{x}^1\right)^2}A1 + 2\frac{\partial x^2}{\partial\left(\bar{x}^1\right)^2}A2 \\
\frac{\partial E}{\partial \bar{x}^2} &= -2\left(\frac{\partial x^1}{\partial \bar{x}^1}k_1 + \frac{\partial x^2}{\partial \bar{x}^1}k_2\right)F + 2\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}A1 + 2\frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}A2 \\
\frac{\partial G}{\partial \bar{x}^1} &= -2\left(\frac{\partial x^1}{\partial \bar{x}^2}k_1 + \frac{\partial x^2}{\partial \bar{x}^2}k_2\right)F + 2\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}C1 + 2\frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}C2 \\
\frac{\partial G}{\partial \bar{x}^2} &= -2\left(\frac{\partial x^1}{\partial \bar{x}^2}k_1 + \frac{\partial x^2}{\partial \bar{x}^2}k_2\right)G + 2\frac{\partial^2 x^1}{\partial\left(\bar{x}^2\right)^2}C1 + 2\frac{\partial x^2}{\partial\left(\bar{x}^2\right)^2}C2 \\
\frac{\partial F}{\partial \bar{x}^1} &= -\left(\frac{\partial x^1}{\partial \bar{x}^1}k_1 + \frac{\partial x^2}{\partial \bar{x}^1}k_2\right)F - \left(\frac{\partial x^1}{\partial \bar{x}^2}k_1 + \frac{\partial x^2}{\partial \bar{x}^2}k_2\right)E \\
&\quad + \frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}A1 + \frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}A2 + \frac{\partial^2 x^1}{\partial\left(\bar{x}^1\right)^2}C1 + \frac{\partial x^2}{\partial\left(\bar{x}^1\right)^2}C2 \\
\frac{\partial F}{\partial \bar{x}^1} &= -\left(\frac{\partial x^1}{\partial \bar{x}^1}k_1 + \frac{\partial x^2}{\partial \bar{x}^1}k_2\right)G - \left(\frac{\partial x^1}{\partial \bar{x}^2}k_1 + \frac{\partial x^2}{\partial \bar{x}^2}k_2\right)F \\
&\quad + \frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}C1 + \frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}C2 + \frac{\partial^2 x^1}{\partial\left(\bar{x}^2\right)^2}A1 + \frac{\partial x^2}{\partial\left(\bar{x}^2\right)^2}a2
\end{aligned} \tag{5.40}
$$

where

$$
\begin{aligned}
A1 &= \frac{\partial x^1}{\partial \bar{x}^1}E_t + \frac{\partial x^2}{\partial \bar{x}^1}F_t, \;\; A2 = \frac{\partial x^1}{\partial \bar{x}^1}F_t + \frac{\partial x^2}{\partial \bar{x}^1}G_t \\
C1 &= \frac{\partial x^1}{\partial \bar{x}^2}E_t + \frac{\partial x^2}{\partial \bar{x}^2}F_t, \;\; C2 = \frac{\partial x^1}{\partial \bar{x}^2}F_t + \frac{\partial x^2}{\partial \bar{x}^2}G_t.
\end{aligned}
$$

Now that we have both (E, F, G) and $(\bar{E}, \bar{F}, \bar{G})$ with respect to $(\bar{x}_1, \bar{x}_2)$, we write the constraints for various kinds of deformations expressed in equations (5.23)-(5.25). Since (k_1, k_2, k) are known, our goal is to find $(\bar{k}_1, \bar{k}_2, \bar{k})$ in order to obtain the normal and depth at $\mathcal{M}$.

5.3.1 Reconstruction Equations for Smooth Mappings

We now derive the reconstruction equations for the various types of deformations we discussed in section 5.2.

Isometric Mappings

Given that ψ is an isometric mapping, the relationship between the moving frames at $\mathcal{M}$ and $\mathcal{T}$ are described in equation (5.23). This gives the following three constraints:

$$\frac{E}{k^2} = \frac{\bar{E}}{\bar{k}^2} \quad \frac{F}{k^2} = \frac{\bar{F}}{\bar{k}^2} \quad \frac{G}{k^2} = \frac{\bar{G}}{\bar{k}^2}. \tag{5.41}$$

We differentiate these constraints under the assumption of IP (using the expressions in equations (5.38) and (5.40)) and obtain the following equations

$$\begin{aligned}
\bar{k}_1 &= \frac{\partial x^1}{\partial \bar{x}^1} k_1 + \frac{\partial x^2}{\partial \bar{x}^1} k_2 - \frac{1}{2E}\left(\frac{\partial^2 x^1}{\partial \left(\bar{x}^1\right)^2} A1 + \frac{\partial^2 x^2}{\partial \left(\bar{x}^1\right)^2} A2\right) \\
\bar{k}_2 &= \frac{\partial x^1}{\partial \bar{x}^1} k_1 + \frac{\partial x^2}{\partial \bar{x}^1} k_2 - \frac{1}{2G}\left(\frac{\partial^2 x^1}{\partial \left(\bar{x}^2\right)^2} C1 + \frac{\partial^2 x^2}{\partial \left(\bar{x}^2\right)^2} C2\right),
\end{aligned} \tag{5.42}$$

where

$$\begin{aligned}
A1 &= \frac{\partial x^1}{\partial \bar{x}^1} E + \frac{\partial x^2}{\partial \bar{x}^1} F, \;\; A2 = \frac{\partial x^1}{\partial \bar{x}^1} F + \frac{\partial x^2}{\partial \bar{x}^1} G, \\
C1 &= \frac{\partial x^1}{\partial \bar{x}^2} E + \frac{\partial x^2}{\partial \bar{x}^2} F, \;\; C2 = \frac{\partial x^1}{\partial \bar{x}^2} F + \frac{\partial x^2}{\partial \bar{x}^2} G.
\end{aligned}$$

These expressions are linear in $(\bar{k}_1, \bar{k}_2)$, independent of $(k, \bar{k})$ and exploit the first and the second-order derivatives of the warp η. $\bar{k}$ can be recovered from equation (5.41) using $(\bar{k}_1, \bar{k}_2)$ obtained from the solution of equation (5.42).

Conformal Mappings

Given that ψ is a conformal mapping, the relationship between the moving frames at $\mathcal{M}$ and $\mathcal{T}$ are described in equation (5.24). This gives the following three constraints:

$$\begin{aligned}
\frac{1}{k^2}\begin{pmatrix} E & G & F \end{pmatrix} &\propto \frac{1}{\bar{k}^2}\begin{pmatrix} \bar{E} & \bar{G} & \bar{F} \end{pmatrix} \\
\frac{1}{k^2}\begin{pmatrix} E & G & F \end{pmatrix} &= \alpha^2 \frac{1}{\bar{k}^2}\begin{pmatrix} \bar{E} & \bar{G} & \bar{F} \end{pmatrix},
\end{aligned} \tag{5.43}$$

where α^2 is the scale of conformity. On differentiating these expressions, we obtain equation (5.42) with $(\bar{k}_1, \bar{k}_2)$ replaced with $\alpha^2(\bar{k}_1, \bar{k}_2)$. We get rid off the $(k, \bar{k})$ and $\propto$ in equation (5.43) by taking ratios of (E, F, G) with $(\bar{E}, \bar{F}, \bar{G})$. This leads to

$$E\bar{G} = \bar{E}G \quad F\bar{G} = \bar{F}G. \tag{5.44}$$

Using $(\bar{k}_1, \bar{k}_2)$ obtained from the solution of equations (5.42) in this equation, we obtain a quadratic equation in α^2 which leads to two possible solutions (upto ambiguity) for conformal mappings. $\bar{k}$ can be recovered from equation (5.43) using $(\bar{k}_1, \bar{k}_2)$ and α^2.

[Bartoli et al., 2015] proposed an analytical solution for isometric and conformal mappings by solving the quadratic expressions in (5.41).

Equiareal Mappings

Given that ψ is an equiareal mapping, the relationship between the moving frames at $\mathcal{M}$ and $\mathcal{T}$ is described in equation (5.25). This is the only constraint due to the equiareal mappings. By differentiating this constraint, we obtain two first-order constraints.

$$\begin{aligned}
&\frac{EG}{k^4} - \frac{F}{k^4}^2 = \frac{\bar{E}\bar{G}}{\bar{k}^4} - \frac{\bar{F}^2}{\bar{k}^4} \\
&\frac{\partial}{\partial \bar{x}^i}\left(\frac{\bar{E}}{k^2}\right)\frac{\bar{G}}{\bar{k}^2} + \frac{\bar{E}}{\bar{k}^2}\frac{\partial}{\partial \bar{x}^i}\left(\frac{\bar{G}}{k^2}\right) - 2\frac{\bar{F}}{\bar{k}^2}\frac{\partial}{\partial \bar{x}^i}\left(\frac{\bar{F}}{k^2}\right) = \\
&\left(\frac{\partial}{\partial x^1}\left(\frac{E}{k^2}\right)\frac{\partial x^1}{\partial \bar{x}^i} + \frac{\partial}{\partial x^2}\left(\frac{E}{k^2}\right)\frac{\partial x^2}{\partial \bar{x}^i}\right)\frac{G}{k^2} \\
&+ \left(\frac{\partial}{\partial x^1}\left(\frac{G}{k^2}\right)\frac{\partial x^1}{\partial \bar{x}^i} + \frac{\partial}{\partial x^2}\left(\frac{G}{k^2}\right)\frac{\partial x^2}{\partial \bar{x}^i}\right)\frac{E}{k^2} \\
&- 2\frac{F}{k^2}\left(\frac{\partial}{\partial x^1}\left(\frac{F}{k^2}\right)\frac{\partial x^1}{\partial \bar{x}^i} + \frac{\partial}{\partial x^2}\left(\frac{F}{k^2}\right)\frac{\partial x^2}{\partial \bar{x}^i}\right) \quad \forall i \in (1,2)
\end{aligned} \tag{5.45}$$

Using the expressions of differentials from equations (5.38) and (5.40), the above-mentioned constraints are written as

$$\begin{aligned}
&\frac{EG}{k^4} - \frac{F}{k^4}^2 = \det(\mathbf{J}_\eta)^2\left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right) = \frac{\bar{E}\bar{G}}{\bar{k}^4} - \frac{\bar{F}^2}{\bar{k}^4} \\
&\bar{k}_1\left(\frac{\bar{E}\bar{G}}{\bar{k}^4} - \frac{\bar{F}^2}{\bar{k}^4}\right) = \left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right)\det(\mathbf{J}_\eta)^2\left(k_1\frac{\partial x^1}{\partial \bar{x}^1} + k_2\frac{\partial x^2}{\partial \bar{x}^1}\right) \\
&- \left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right)\det\mathbf{J}_\eta\left(\frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial^2 x^2}{\partial \bar{x}^1\partial \bar{x}^2} + \frac{\partial x^2}{\partial \bar{x}^2}\frac{\partial^2 x^1}{(\partial \bar{x}^1)^2}\right) \\
&+ \left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right)\det\mathbf{J}_\eta\left(\frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2} + \frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial^2 x^2}{(\partial \bar{x}^1)^2}\right) \\
&\bar{k}_2\left(\frac{\bar{E}\bar{G}}{\bar{k}^4} - \frac{\bar{F}^2}{\bar{k}^4}\right) = \left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right)\det(\mathbf{J}_\eta)^2\left(k_1\frac{\partial x^1}{\partial \bar{x}^2} + k_2\frac{\partial x^2}{\partial \bar{x}^2}\right) \\
&- \left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right)\det\mathbf{J}_\eta\left(\frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial^2 x^2}{(\partial \bar{x}^2)^2} + \frac{\partial x^2}{\partial \bar{x}^2}\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}\right) \\
&+ \left(\frac{E_tG_t}{k^4} - \frac{F_t}{k^4}^2\right)\det\mathbf{J}_\eta\left(\frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial^2 x^1}{(\partial \bar{x}^2)^2} + \frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial^2 x^2}{\partial \bar{x}^1\partial \bar{x}^2}\right).
\end{aligned} \tag{5.46}$$

Using the first constraint in the rest, we end up with the following expressions

$$\begin{aligned}
&\bar{k}_1\det\mathbf{J}_\eta = \det\mathbf{J}_\eta\left(k_1\frac{\partial x^1}{\partial \bar{x}^1} + k_2\frac{\partial x^2}{\partial \bar{x}^1}\right) - \\
&\left(\frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial^2 x^2}{\partial \bar{x}^1\partial \bar{x}^2} + \frac{\partial x^2}{\partial \bar{x}^2}\frac{\partial^2 x^1}{(\partial \bar{x}^1)^2}\right) + \left(\frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2} + \frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial^2 x^2}{(\partial \bar{x}^1)^2}\right) \\
&\bar{k}_2\det\mathbf{J}_\eta = \det\mathbf{J}_\eta\left(k_1\frac{\partial x^1}{\partial \bar{x}^2} + k_2\frac{\partial x^2}{\partial \bar{x}^2}\right) -
\end{aligned}$$

$$\left(\frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial^2 x^2}{(\partial \bar{x}^2)^2}+\frac{\partial x^2}{\partial \bar{x}^2}\frac{\partial^2 x^1}{\partial \bar{x}^1 \partial \bar{x}^2}\right)+\left(\frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial^2 x^1}{(\partial \bar{x}^2)^2}+\frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial^2 x^2}{\partial \bar{x}^1 \partial \bar{x}^2}\right). \tag{5.47}$$

Given that (k_1, k_2) and the first and second-order derivatives of η are known, these expressions are linear in $(\bar{k}_1, \bar{k}_2)$. $\bar{k}$ can be found by using $(\bar{k}_1, \bar{k}_2)$ obtained from equation (5.47) in equation (5.45).

Skewless Mappings

Given that ψ is a skewless mapping, the relationship between the moving frames at $\mathcal{M}$ and $\mathcal{T}$ is described in equation (5.29). We write the constraint as

$$F^2 \bar{E}\bar{G} = \bar{F}^2 EG. \tag{5.48}$$

This expression is independent of $(k, \bar{k})$. This constraint can be rewritten as

$$\begin{aligned}\left(\bar{E}\bar{G}-\bar{F}^2\right)F^2 &= \left(\det \mathbf{J}_\eta\right)^2\left(EG-F^2\right)\bar{F}^2\\ \left(\bar{E}\bar{G}-\bar{F}^2\right)EG &= \left(\det \mathbf{J}_\eta\right)^2\left(EG-F^2\right)\bar{E}\bar{G}.\end{aligned} \tag{5.49}$$

By differentiating the skewless constraint in equation (5.48), we get

$$\begin{aligned}&2F\left(\frac{\partial F}{\partial x^1}\frac{\partial x^1}{\partial \bar{x}^i}+\frac{\partial F}{\partial x^2}\frac{\partial x^2}{\partial \bar{x}^i}\right)\bar{E}\bar{G}+F^2\left(\frac{\partial \bar{E}}{\partial \bar{x}^i}\bar{G}+\bar{E}\frac{\partial \bar{G}}{\partial \bar{x}^i}\right)=\\ &2\bar{F}\frac{\partial \bar{F}}{\partial \bar{x}^i}EG+\bar{F}^2\left(\frac{\partial E}{\partial x^1}\frac{\partial x^1}{\partial \bar{x}^i}+\frac{\partial E}{\partial x^2}\frac{\partial x^2}{\partial \bar{x}^i}\right)G\\ &+\bar{F}^2E\left(\frac{\partial G}{\partial x^1}\frac{\partial x^1}{\partial \bar{x}^i}+\frac{\partial G}{\partial x^2}\frac{\partial x^2}{\partial \bar{x}^i}\right)\qquad \forall i\in(1,2)\end{aligned} \tag{5.50}$$

We expand these expressions using equations (5.38) and (5.40) and obtain

$$\begin{aligned}&EF\left(\bar{E}\bar{G}-\bar{F}^2\right)\left(\frac{\partial x^1}{\partial \bar{x}^2}k_1+\frac{\partial x^2}{\partial \bar{x}^2}\right)-\bar{F}\bar{E}(EG-F^2)\bar{k}_2=\\ &F\bar{E}\bar{G}\left(\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}A1+\frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}A2+\frac{\partial^2 x^1}{\partial\left(\bar{x}^1\right)^2}C1+\frac{\partial x^2}{\partial\left(\bar{x}^1\right)^2}C2\right)\\ &\bar{F}^2\left(E\left(\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}C1+\frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}C2\right)+G\left(\frac{\partial^2 x^1}{\partial\left(\bar{x}^1\right)^2}A1+\frac{\partial x^2}{\partial\left(\bar{x}^1\right)^2}A2\right)\right)\\ &GF\left(\bar{E}\bar{G}-\bar{F}^2\right)\left(\frac{\partial x^1}{\partial \bar{x}^1}k_1+\frac{\partial x^2}{\partial \bar{x}^1}\right)-\bar{F}\bar{G}(EG-F^2)\bar{k}_1=\\ &F\bar{E}\bar{G}\left(\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}C1+\frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}C2+\frac{\partial^2 x^1}{\partial\left(\bar{x}^2\right)^2}A1+\frac{\partial x^2}{\partial\left(\bar{x}^2\right)^2}A2\right)\\ &\bar{F}^2\left(G\left(\frac{\partial^2 x^1}{\partial \bar{x}^1\partial \bar{x}^2}A1+\frac{\partial x^2}{\partial \bar{x}^1\partial \bar{x}^2}A2\right)+E\left(\frac{\partial^2 x^1}{\partial\left(\bar{x}^2\right)^2}C1+\frac{\partial x^2}{\partial\left(\bar{x}^2\right)^2}C2\right)\right)\end{aligned}$$

where

$$A1=\frac{\partial x^1}{\partial \bar{x}^1}E+\frac{\partial x^2}{\partial \bar{x}^1}F,\ \ A2=\frac{\partial x^1}{\partial \bar{x}^1}F+\frac{\partial x^2}{\partial \bar{x}^1}G,$$

$$C1 = \frac{\partial x^1}{\partial \bar{x}^2}E + \frac{\partial x^2}{\partial \bar{x}^2}F,\ C2 = \frac{\partial x^1}{\partial \bar{x}^2}F + \frac{\partial x^2}{\partial \bar{x}^2}G.$$

Using equations (5.48) and (5.49) in the above expressions, we obtain the following two constraints for skewless deformation

$$EG\bar{F}\bar{k}_2 - EF\bar{G}\left(k_1\frac{\partial x^1}{\partial \bar{x}^2} + k_2\frac{\partial x^2}{\partial \bar{x}^2}\right) + B_1\bar{G} + \det \mathbf{J}_\eta^{-1} D_1 EG\bar{G} = 0$$
$$EG\bar{F}\bar{k}_1 - GF\bar{E}\left(k_1\frac{\partial x^1}{\partial \bar{x}^1} + k_2\frac{\partial x^2}{\partial \bar{x}^1}\right) + B_2\bar{E} + \det \mathbf{J}_\eta^{-1} D_2 EG\bar{E} = 0.$$

where

$$\begin{aligned}
A_1 &= \frac{\partial x^1}{\partial \bar{x}^1}E^t + \frac{\partial x^2}{\partial \bar{x}^1}F^t, \quad A_2 = \frac{\partial x^1}{\partial \bar{x}^1}F^t + \frac{\partial x^2}{\partial \bar{x}^1}G^t,\\
C_1 &= \frac{\partial x^1}{\partial \bar{x}^2}E^t + \frac{\partial x^2}{\partial \bar{x}^2}F^t, \quad C_2 = \frac{\partial x^1}{\partial \bar{x}^2}F^t + \frac{\partial x^2}{\partial \bar{x}^2}G^t,\\
B_1 &= G\left(A_1\frac{\partial^2 x^1}{(\partial x^1)^2} + A_2\frac{\partial^2 x^2}{(\partial x^1)^2}\right) + E\left(C_1\frac{\partial^2 x^1}{\partial x^1\partial x^2} + C_2\frac{\partial^2 x^2}{\partial x^1\partial x^2}\right),\\
B_2 &= F\left(A_1\frac{\partial^2 x^1}{\partial x^1\partial x^2} + A_2\frac{\partial^2 x^2}{\partial x^1\partial x^2}\right) + E\left(C_1\frac{\partial^2 x^1}{(\partial x^2)^2} + C_2\frac{\partial^2 x^2}{(\partial x^2)^2}\right),\\
D_1 &= -\frac{\partial x^2}{\partial \bar{x}^2}\frac{\partial^2 x^1}{(\partial x^1)^2} + \frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial^2 x^2}{(\partial x^1)^2} + \frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial^2 x^1}{\partial x^1\partial x^2} - \frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial^2 x^2}{\partial x^1\partial x^2},\\
D_2 &= -\frac{\partial x^2}{\partial \bar{x}^2}\frac{\partial^2 x^1}{\partial x^1\partial x^2} + \frac{\partial x^1}{\partial \bar{x}^2}\frac{\partial^2 x^2}{\partial x^1\partial x^2} + \frac{\partial x^2}{\partial \bar{x}^1}\frac{\partial^2 x^1}{(\partial x^2)^2} - \frac{\partial x^1}{\partial \bar{x}^1}\frac{\partial^2 x^2}{(\partial x^2)^2}.
\end{aligned} \tag{5.51}$$

Given that (k_1, k_2) and the first and second-order derivatives of η are known and $(\bar{E}, \bar{F}, \bar{G})$ are quadratic in $(\bar{k}_1, \bar{k}_2)$ (written according to equation (5.37)), these expressions are cubic in $(\bar{k}_1, \bar{k}_2)$. These expressions are independent of $(k, \bar{k})$ and therefore, $\bar{k}$ cannot be found for skewless mappings.

5.3.2 Reconstruction Equations for Infinitesimally Linear Mappings

Given that ψ an infinitesimally linear mapping, according to theorem 6, the connection forms $\left(\left(w^f\right)_1^1, \left(w^f\right)_2^1, \left(w^f\right)_1^2, \left(w^f\right)_2^2\right)$ defined using ϕ_1 at $\mathcal{T}$ are the same as the respective connection forms $(\bar{w}_1^1, \bar{w}_2^1, \bar{w}_1^2, \bar{w}_2^2)$ defined using θ at $\mathcal{M}$. Using equation (5.18), the connection forms at $\mathcal{M}$ defined using θ are related by a change of variable with the CS at the same surface defined using ϕ_2.

Therefore, the connections at $\mathcal{T}$ (parametrised with x^1 and x^2) are related to the connections at $\mathcal{M}$ (parametrised with $\bar{x}^1$ and $\bar{x}^2$) by a change of variable suggested in equation (5.18) for any smooth and IL mapping ψ between them. The components of these connection forms

under a change of variable are given by

$$\begin{aligned}\begin{pmatrix}\bar{\Gamma}^k_{11} & \bar{\Gamma}^k_{12}\\ \bar{\Gamma}^k_{21} & \bar{\Gamma}^k_{22}\end{pmatrix} &= \frac{\partial \bar{x}^k}{\partial x^1}\mathbf{J}_\eta^{-1}\begin{pmatrix}\Gamma^1_{11} & \Gamma^1_{12}\\ \Gamma^1_{21} & \Gamma^1_{22}\end{pmatrix}\mathbf{J}_\eta \\ &+ \frac{\partial \bar{x}^k}{\partial x^2}\mathbf{J}_\eta^{-1}\begin{pmatrix}\Gamma^2_{11} & \Gamma^2_{12}\\ \Gamma^2_{21} & \Gamma^2_{22}\end{pmatrix}\mathbf{J}_\eta + \mathbf{J}_\eta^{-1}\frac{\partial \mathbf{J}_\eta}{\partial x^k},\end{aligned} \tag{5.52}$$

where $k = (1, 2)$ and $\mathbf{J}_\eta^{-1} = \begin{pmatrix}\frac{\partial \bar{x}^1}{\partial x^1} & \frac{\partial \bar{x}^1}{\partial x^2}\\ \frac{\partial \bar{x}^2}{\partial x^1} & \frac{\partial \bar{x}^2}{\partial x^2}\end{pmatrix}$. These components of the connection forms for the image embedding described in equation (3.4) are evaluated in equation (5.12). For $\mathcal{T}$ and $\mathcal{M}$, these expressions are written only in terms of (k_1, k_2) and $(\bar{k}_1, \bar{k}_2)$ respectively. This leads to the following two constraints

$$\begin{aligned}\bar{k}_1 &= \frac{\partial x^1}{\partial \bar{x}^1}k_1 + \frac{\partial x^2}{\partial \bar{x}^1}k_2 - \frac{\partial \bar{x}^2}{\partial x^1}\frac{\partial^2 x^1}{\partial \bar{x}^1 \bar{x}^2} - \frac{\partial \bar{x}^2}{\partial x^2}\frac{\partial^2 x^2}{\partial \bar{x}^1 \bar{x}^2}\\ \bar{k}_2 &= \frac{\partial x^1}{\partial \bar{x}^2}k_1 + \frac{\partial x^2}{\partial \bar{x}^2}k_2 - \frac{\partial \bar{x}^1}{\partial x^1}\frac{\partial^2 x^1}{\partial \bar{x}^1 \bar{x}^2} - \frac{\partial \bar{x}^1}{\partial x^2}\frac{\partial^2 x^2}{\partial \bar{x}^1 \bar{x}^2}.\end{aligned} \tag{5.53}$$

Since (k_1, k_2) are known, these equations are linear in terms of $(\bar{k}_1, \bar{k}_2)$. These expressions are independent of $(k, \bar{k})$ and therefore, $\bar{k}$ cannot be found for IL mappings.

We rewrite this solution as

$$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\bar{k}_1\\ \bar{k}_2\end{pmatrix} = \mathbf{J}_\eta^\top\begin{pmatrix}k_1\\ k_2\end{pmatrix} - \begin{pmatrix}\frac{\partial \bar{x}^2}{\partial x^1}\frac{\partial^2 x^1}{\partial \bar{x}^1 \bar{x}^2} + \frac{\partial \bar{x}^2}{\partial x^2}\frac{\partial^2 x^2}{\partial \bar{x}^1 \bar{x}^2}\\ \frac{\partial \bar{x}^1}{\partial x^1}\frac{\partial^2 x^1}{\partial \bar{x}^1 \bar{x}^2} + \frac{\partial \bar{x}^1}{\partial x^2}\frac{\partial^2 x^2}{\partial \bar{x}^1 \bar{x}^2}\end{pmatrix}. \tag{5.54}$$

The right hand side of the expression is known and therefore we see that there is a unique solution for IL mappings. This is coherent with the solution in [Bartoli and Özgür, 2016] (which also shows that non-isometric SfT has a unique solution). [Salzmann et al., 2007] argued that it is not possible to solve SfT at zeroth-order without using a deformation prior. Warps provide additional priors in terms of the first and the second-order derivatives which leads to additional constraints and makes our solution well-posed.

5.3.3 Reconstruction Algorithm

We wrote reconstruction equations for various kinds of smooth deformations in equations (5.42), (5.47), (5.51) and (5.53). All of these equations are expressed in terms of the unknowns $(\bar{k}_1, \bar{k}_2)$. We present the following algorithm to solve these equations.

Inputs: The warp η and (k, k_1, k_2) on the corresponding points of $\mathcal{T}$ and $\mathcal{M}$.

1) Find $(\bar{k}_1, \bar{k}_2)$. For isometric and conformal deformations, the solution to equation (5.42) gives $(\bar{k}_1, \bar{k}_2)$. These equations are linear in $(\bar{k}_1, \bar{k}_2)$. For skewless deformations, equations (5.51) are cubic in $(\bar{k}_1, \bar{k}_2)$. We solve them by minimising the sum-of-squares using [Henrion and Lasserre, 2003]. For equiareal and infinitesimally linear deformations, equa-

Deformation	Assumption	Degree of constraints	Scale	Normals
Isometric	IP	1	Yes	Yes
Conformal	IP	1	Yes	Yes
Skewless	IP	3	**No**	Yes
Equiareal	IP	1	Yes	Yes
Smooth	IL	1	**No**	Yes
Isometric [Bartoli et al., 2015]	-	2	Yes	Yes
Conformal [Bartoli et al., 2015]	-	2	**No**	Yes

Table 5.1: Summary of model-based 3D reconstruction of deformable thin-shell objects. For each deformation we obtain two constraints in $(\bar{k}_1, \bar{k}_2)$. The degree of these constraints is shown in the table and discuss whether depths and normals for deformation are recoverable.

tions (5.47) and (5.53) are also linear in $(\bar{k}_1, \bar{k}_2)$ and therefore easily solvable.

3) Find normals at $\mathcal{M}$. Compute unit normal at each point on $\mathcal{M}$ according to equation (5.12) in terms of $(\bar{k}_1, \bar{k}_2)$.

4) Find depth at $\mathcal{M}$. For isometric, conformal and equiareal deformations, the scale can be evaluated by using $(\bar{k}_1, \bar{k}_2)$ in equations (5.41), (5.43) and (5.45) respectively. For skewless deformations, the scale cannot be recovered.

Outputs: Points (for isometric and equiareal deformations) and normals on 3D surfaces.

Model-based 3D reconstruction of deformable thin-shell objects is summarised in table 5.1.

5.4 Model-Free 3D Reconstruction

We propose local solutions to 3D reconstruction of a deformable thin-shell object from multiple views. Figure 5.7 shows the modelling. We have N input images $\mathcal{I}_1, \ldots, \mathcal{I}_N$ representing

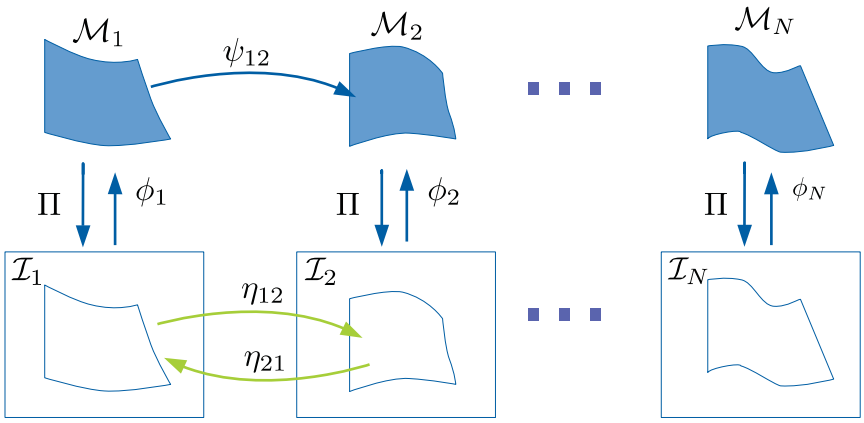


Figure 5.7: Modelling N views of a deforming 3D surface.

different deformations of the same object $\mathcal{M}_1, \ldots, \mathcal{M}_N$ viewed in a perspective camera. Our goal is to reconstruct the surfaces viewed in the N images. η_{ij} represents the image warp

between the pair of images ($\mathcal{I}_i$, $\mathcal{I}_j$). ψ_{ij} is the deformation function between $\mathcal{M}_i$ and $\mathcal{M}_j$ modelled using the image embedding described in equation (3.4). We write the constraints for a pair of views i and j. This modelling can be extended to any number of image pairs. The moving frames (e_1^i, e_2^i, e_3^i) and $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$ on $\mathcal{M}_i$ and $\mathcal{M}_j$ are written according to equation (5.12). The expressions (E_i, F_i, G_i), $(\bar{E}, \bar{F}, \bar{G})$ and their first-order derivatives are expressed according to equations (5.37) and (5.38). Using equation (5.39), we can write (e_1, e_2) at $\mathcal{M}_i$ which allows us to write (E, G, F) and its derivatives with respect to $(\bar{x}_1, \bar{x}_2)$. In this case, we can write the reconstruction equations in a similar fashion to the previous section but the unknowns are both (k_1, k_2, k) and $(\bar{k}_1, \bar{k}_2, \bar{k})$. Under the assumption of IL, we can express $(\bar{k}_1, \bar{k}_2, \bar{k})$ in terms of (k_1, k_2, k) using equation (5.53) which allows us to restrict the unknowns to (k_1, k_2, k) only.

5.4.1 Reconstruction Equations

Now we derive the reconstruction equations for the various types of smooth deformations we discussed in this chapter.

Isometric/Conformal Mappings

Given that ψ is an isometric/conformal mapping (with the isometric and conformal mapping constraints in equations (5.41) and (5.43) respectively), the reconstruction equations for a pair of views $\mathcal{I}_i$ and $\mathcal{I}_j$ are given in equation (5.44) as a set of two cubic expressions in terms of (k_1, k_2) and $(\bar{k}_1, \bar{k}_2)$. Using equation (5.53) in equation (5.44), we obtain two cubic equations in terms of (k_1, k_2) only. We proposed a solution to these equations for isometric deformations in chapter 4. It is important to note that the solution in previous chapter uses metric tensors and CS. Using the proposed framework it becomes clear that isometry and conformity share the same constraints in the context of NRSfM. This result was straightforward from the theory developed in chapter 4.

Equiareal Mappings

Given that ψ is an equiareal mapping, the reconstruction equations are given in equation (5.47) in terms of (k_1, k_2) and $(\bar{k}_1, \bar{k}_2)$. Using equation (5.53) in equation (5.47) leads to a system of equations independent of $(\bar{k}_1, \bar{k}_2)$ and (k_1, k_2). Therefore, these expressions cannot be used to solve for (k_1, k_2). Next we prove that equiareal NRSfM cannot be solved algebraically.

Theorem 7 (Non-solvability of Equiareal NRSfM)**.** *Equiareal NRSfM is not locally solvable.*

Proof. The constraint in equiareal mappings is given by equation (5.45). Without the assumption of IP, these expressions are in terms of β, $\bar{\beta}$ and their first and second-order derivatives. Using equation (5.53), this results in 7 variables (β, $\bar{\beta}$ and the first and second-order derivatives of β) in 3 equations. This system is not solvable. Even if we differentiate equation (5.45) further, it always results in higher number of variables than equations.

Deformation	Assumption	Degree of constraints	Scale	Normals
Isometric	IL	3	No	Yes
Conformal	IL	3	No	Yes
Skew-less	IL	7	No	Yes
Equiareal	IL	3	No	No
Smooth	IL	1	No	No
Isometric	-	3	No	Yes
Conformal	-	3	No	Yes
Skew-less	-	7	-	-
Equiareal	-	3	No	No
Smooth	-	1	No	No

Table 5.2: Summary of model-free 3D reconstruction of deformable thin-shell objects. For each deformation we obtain two constraints in (k_1, k_2). The degree of these constraints is shown in the table and discuss whether depths and normals for deformation are recoverable.

Under the assumption of IP, the simplified constraint is given in equation (5.47). Using equation (5.53) in this constraint leads to expressions independent of $(\bar{k}_1, \bar{k}_2)$ and (k_1, k_2). Therefore, it is not solvable. Differentiating equation (5.47) to find more constraints under the assumption of IP does not make sense as it contradicts the assumption of IP. Hence, we show that the PDE (5.45) for equiareal NRSfM does not possess a local solution.

□

Skewless Mappings

Given that ψ is a skewless mapping, the reconstruction equations are given in equation (5.51) in terms of (k_1, k_2) and $(\bar{k}_1, \bar{k}_2)$. Using equation (5.53) in equation (5.51), we obtain two septic equations in terms of (k_1, k_2) only. These equations can be solved by minimising the sum-of-squares.

Smooth Linear Mappings

Given that ψ is a smooth infinitesimal mapping, there can only be two constraints obtained from equation (5.53) in terms of (k_1, k_2) and $(\bar{k}_1, \bar{k}_2)$ respectively. Therefore, a solution for (k_1, k_2) cannot be obtained. This is an important result as it becomes clear that NRSfM cannot be solved based only on deformation smoothness.

5.4.2 Reconstruction Algorithm

We wrote reconstruction equations for various kinds of smooth deformations in equations (5.44), (5.47), (5.51) and (5.53). All of these equations are expressed in terms of the unknowns (k_1, k_2). We present the following algorithm to solve these equations.

Inputs: Warps η_{j1}, $j \in [2, N]$. The index 1 corresponds to the first image in the sequence. It can be chosen randomly.

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

2) Find (k_1, k_2). For isometric, conformal and skewless deformations, the solution to equations (5.44) and (5.51) gives (k_1, k_2). We solve these equations by minimising the sum-of-squares using [Henrion and Lasserre, 2003].

3) Find $(\bar{k}_1, \bar{k}_2)$. $(\bar{k}_1, \bar{k}_2)$ can be written in terms of (k_1, k_2) and the first and second order derivatives of η_{ji} using equation (5.53).

4) Find normals at $\mathcal{M}$. Compute unit normal at each point on $\mathcal{M}$ according to equation (5.12) in terms of $(\bar{k}_1, \bar{k}_2)$ and (k_1, k_2).

5) Find depth at $\mathcal{M}$. The depth can be evaluated by using the method described in [Chhatkuli et al., 2014].

Outputs: Points (for isometric deformations only) and normals on 3D surfaces.

Model-free 3D reconstruction of deformable thin-shell objects is summarised in table 5.2.

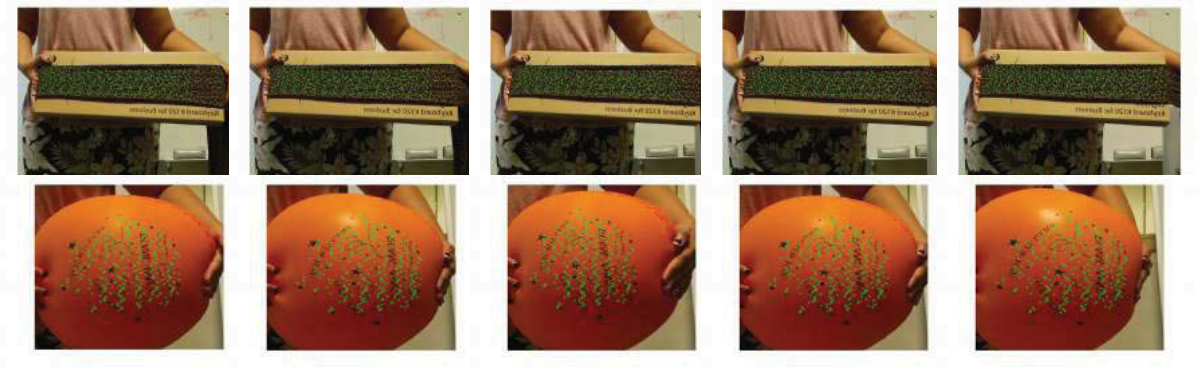


Figure 5.8: Images from the sock and balloon datasets. Some of the tracked points are shown. Best viewed in colour.

5.5 Experiments and Discussion

We tested our proposed methods for model-based and model-free reconstruction of thin-shell deformable objects on two synthetic and four real datasets. These datasets show objects undergoing different types of deformations according to the properties of their material. The *Cylinder* dataset is a synthetic dataset which consists of isometric deformations of a cylindrical surface viewed in images of the size $640p \times 480p$ with a focal length of $400p$. 400 points are tracked across the isometrically deformed surfaces. The *Paper* dataset (introduced in [Varol et al., 2012a]) consists of 190 images (with 1500 tracked points) of a paper deforming isometrically. The synthetic *Rubber* dataset (introduced in [Özgür and Bartoli, 2016]) consists of 50 partially-stretched surfaces of a rubber with 400 tracked points. The *Tissue* dataset is a piece of elastic tissue with 100 points matched to a undeformed model of the tissue. It has only one image. This dataset was introduced in [Bartoli and Özgür, 2016]. The *Balloon* dataset consists of 20 surfaces (3000 point tracks) of a balloon deformed conformally. This dataset

was recorded using Kinect2.0 and the point tracks were obtained using [Sundaram et al., 2010]. The *Sock* dataset consists of 20 surfaces (3500 point tracks) of a sock that undergoes elastic deformations on one direction only. This dataset was recorded using Kinect2.0 and the point tracks were obtained using [Sundaram et al., 2010]. A few images of the *Balloon* and *Sock* datasets are shown in figure 5.8.

For quantitative comparison, we measured the normal error (mean difference between computed and ground-truth normals in degrees) and the depth error (mean difference between computed and ground-truth 3D coordinates in mm).

In the model-free scenario (NRSfM), we proposed solutions using isometric, conformal and skewless deformation constraints. Our solution for conformal and isometric deformations is obtained by solving same equations. Therefore, our method **IsoConN** represents a solution to NRSfM assuming that the object undergoes isometric or conformal deformation and **SkewN** represents the solution to NRSfM assuming skewless deformations. We compare our results with **MDHN** (maximum depth heuristics based NRSfM method proposed in [Chhatkuli et al., 2016a]) and **KerN** (low-dimensional shape-basis based NRSfM method proposed in [Gotardo and Martinez, 2011]).

In the model-based scenario (SfT), we proposed solutions using isometric, (**IsoS**), conformal (**ConS**), skewless (**SkewS**) and equiareal (**EqArS**) deformation constraints. We also proposed a solution (**NoDefS**) which does not need deformation to be modelled explicitly. It assumes IL. We compare our results with isometric (**IsoFS**) and conformal (**ConFS**) SfT proposed in [Bartoli et al., 2015] and **LinModS** (also an isometry based SfT proposed in [Salzmann and Fua, 2011]). We also compare our results with **NoIsoS** (a smoothness based SfT proposed in [Özgür and Bartoli, 2016]). [Özgür and Bartoli, 2016] proposed five solutions, we report only the best working solution for this method as **NoIsoS**. We now discuss the results of these methods on each dataset separately.

5.5.1 Synthetic Datasets

We evaluated the performance of the compared methods on two synthetic datasets: Cylinder and Rubber.

5.5.1.1 Cylinder Dataset

The cylinder dataset consists of images of a cylindrical surface deforming isometrically. We added a random noise with a Gaussian distribution of 1 pixel of standard deviation to the images of the dataset. The mean normal and depth error of each compared method is evaluated by averaging the result of the experiment over 20 trials. For evaluating SfT methods, the results are shown in figure 5.9. The first four methods (**IsoConN**, **SkewN**, **MDHN** and **KerN**) are NRSfM methods. **IsoConN** shows the best performance amongst the NRSfM methods in this dataset. The images of this dataset do not come from a video sequence, therefore **KerN** did not do well for this dataset. In this dataset, the object and the camera do not move while deforming. This makes the problem ill-posed. [Chhatkuli et al., 2016a]

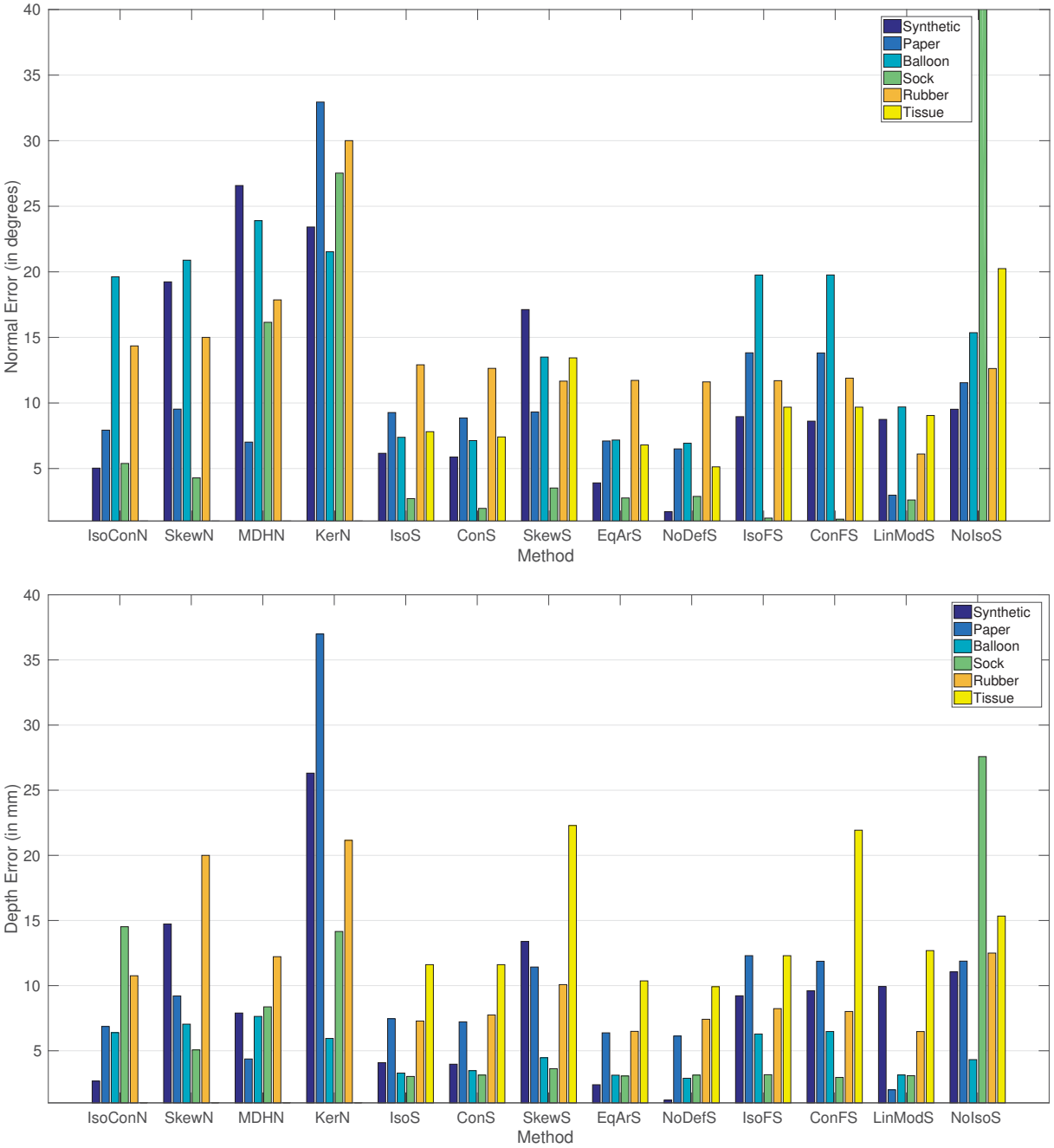


Figure 5.9: Normal and depth errors on all datasets. The errors shown are evaluated by averaging the errors on the entire imageset. The first four methods shown are NRSFM methods and the rest are SfT methods. In general, the performance of SfT methods is better than NRSfM methods on these datasets. The methods ending with N and S are NRSfM and SfT methods respectively. **NoIsoS** represents the best performing SfT method in [Bartoli and Özgür, 2016]. Best viewed in colour.

reported this as a failure scenario for **MDHN** and therefore, it did not perform well for this dataset. The model-based (SfT) methods, however, showed a much better performance as compared to NRSfM methods on this dataset. **IsoS**, **ConS**, **EqArS**, **NoDefS**, **IsoFS** and **ConFS** are all analytical solutions to SfT. Our methods (**IsoS**, **ConS**, **EqArS**, **NoDefS**) are linear solutions to SfT (exploiting the first and second-order derivatives of the warp η) while the solutions for **IsoFS** and **ConFS** are obtained by solving quadratic equations (see equation (5.20)) consisting of only the first-order derivatives of the warp and the embedding. The constraints in our methods are obtained by using equation (5.20) and its derivatives which forces the solution to be locally smooth implicitly. Therefore, **IsoS**, **ConS**, **EqArS** and **NoDefS** show better performance than **IsoFS** and **ConFS**. **SkewS** (assumes the deformation to be skewless) does not show good results on this dataset. This is probably because it is less-constrained than other deformation models. **IsoFS**, **ConFS** and **LinModS** show a similar performance on this dataset. The performance of these methods is very good. **NoIsoS** is the best performing method out of the five methods proposed in [Bartoli and Özgür, 2016]. It needs high perspective in images and therefore it performed very well in this dataset. The performance of this method is quite close to other SfT methods (**IsoS**, **ConS**, **IsoFS**, **ConFS** and **LinModS**). In general, this method is unstable in the sense that the performance of the proposed methods in [Bartoli and Özgür, 2016] is usually very different from each other. We picked the best performing method for each image in the dataset by comparing the reconstruction with the ground truth.

Performance of methods under noisy conditions. We evaluated the performance of the methods under noisy conditions. We added 1-5 pixel noise to the images of the dataset and obtain the results for each method. We report the normal and depth errors for each experiment averaged over 20 trials. Figure 5.10 shows our results. Our SfT (**IsoS**, **ConS**, **SkewS**, **EqArS**, **NoDefS**) and NRSfM (**IsoConN**, **SkewN**) methods show a very stable performance in noisy conditions. The performance of other methods is very stable as well. Our methods (both SfT and NRSfM) exploit the first and second-order derivatives of the warps. These derivatives (especially the second-order ones) may be largely affected by the noise. Therefore, the methods **EqArS**, **NoDefS**, **IsoConN** (which solve SfT/NRSfM in terms of the first and second-order derivatives of the warp only) show a larger increase in the errors compared to other methods (such as **IsoFS** and **ConFS**) that use the first-order derivatives of the warps only. **LinModS**, **NoIsoS**, **ConS** and **IsoS** are very stable in the presence of noise. The skewless methods **SkewN** and **SkewS** also show a good tolerance to noise even though their performance on this dataset is not so good. **MDHN** and **KerN** do not perform well on this dataset and therefore it is largely affected by noise as well.

Curvature test of our methods. The best performing methods on this dataset are **EqArS** and **NoDefS** (with **NoDefS** being better). These solutions (like the rest of our methods) are dependent on the first and second-order derivatives of the warps. The second-order derivatives may be unstable however they can be corrected by warp refinement methods such as [Pizarro et al., 2016]. However it may be difficult to find them in some cases, therefore it is interesting to see the performance of these methods without second-order derivatives. We

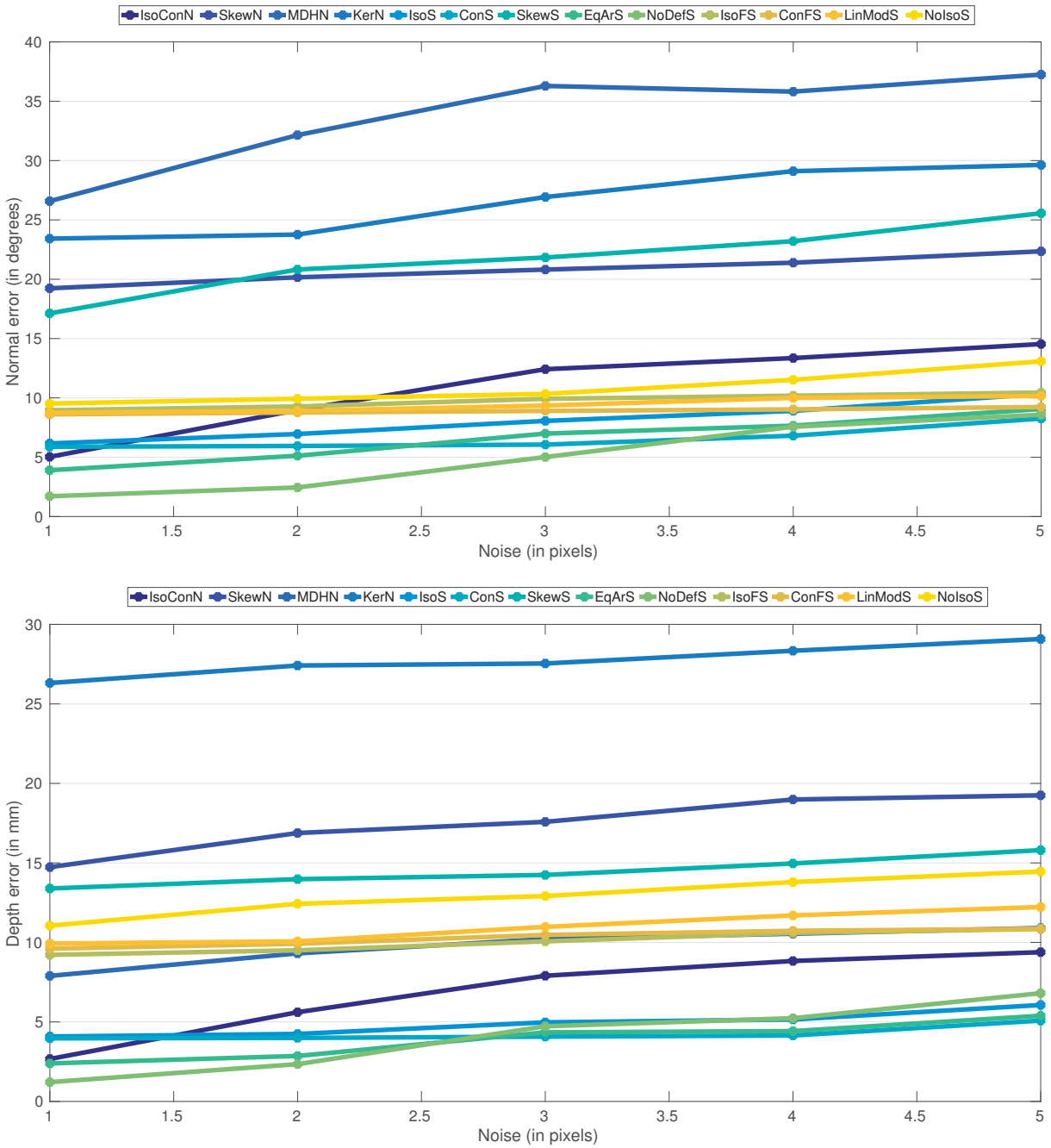


Figure 5.10: Normal and depth errors on the cylinder dataset by varying the noise from 1 to 5 pixels. The mean of errors on the entire imageset of each dataset is reported. Best viewed in colour.

make an approximation to the solution of **NoDefS** (SfT under IL assumption) by disregarding the second-order derivatives. The new solution is called **NoDefS0**. Figure 5.11 shows the results of all methods with respect to the varying curvature of the surface. The curvature is the inverse of the radius. In this experiment we show reconstruction for 10 surfaces of the cylinder with the radius varying from 2 to 10. In general the reconstruction errors increase with bending. However, **IsoConN**, **SkewN** and **NoDefS** do not show a sharp increase in errors due to bending. **NoDefS0** is affected by bending (especially with strong bending), however it still shows decent reconstruction for highly bent surfaces.

The interesting thing about **NoDefS** is the simplicity of the solution. The reconstruction algorithm is only two lines of code given that (k_1, k_2) in equation (5.53) are obtained from the template. This means that the computational cost of this method is very low as it involves simple addition and multiplications only. Even by ignoring second-order derivatives, **NoDefS0** gives decent results.

5.5.1.2 Rubber Dataset

We evaluated all compared SfT and NRSfM methods on the first 20 images of this dataset. The rubber is partially stretched from its longest side in a sequential order. Figure 5.9 summarises the performance of different methods on this dataset. The reconstruction of surfaces 10 and 20 by the compared methods are shown in figure 5.12. Amongst NRSfM methods, **IsoConN** shows the best performance. The rest of the NRSfM methods do not perform well. Amongst SfT methods, **LinModS** shows the best performance with **EqArS** and **NoDefS** being close (in terms of depth). Figure 5.12 shows that the curvature of the surface 10 is best captured by **EqArS** with **NoDefS** whereas surface 20 is decently reconstructed by **LinModS** only. For low deformations of this dataset, **EqArS** with **NoDefS** show better results than **LinModS** but they cannot cope up with the higher deformations of this dataset.

5.5.2 Real Datasets

Now we discuss the results of the compared methods on real datasets.

5.5.2.1 Paper Dataset

This dataset consists of 191 images. We picked 20 images by uniformly sampling the dataset. This makes the dataset more closer to wide-baseline and therefore, **KerN** does not have good results. Figure 5.9 summarises the performance of the compared methods on this dataset. Amongst NRSfM methods, **MDHN** shows the best performance, with **IsoConN** being very close to it. **SkewN** also shows good results on this dataset. Our SfT methods show a very good performance on this dataset. **NoIsoS** also leads to decent reconstruction in this dataset. However, **LinModS** shows the best performance. The reconstructions of two surfaces of this dataset are shown in figure 5.13. **IsoFS** and **ConFS** do not show a good performance on the selected images of this dataset. We recall that in order to compare them with our methods,

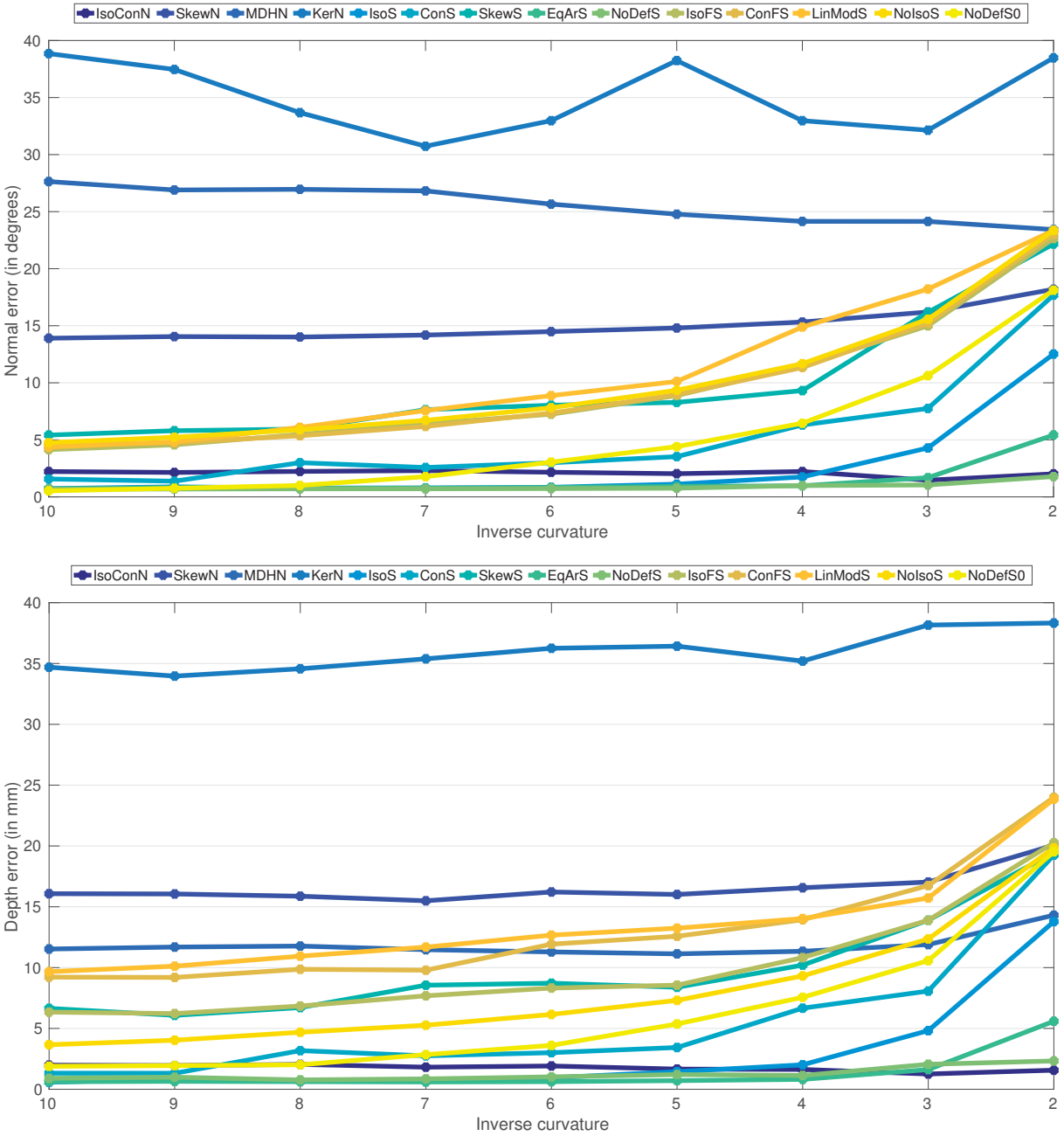


Figure 5.11: Normal and depth errors on the cylinder dataset by varying the radius from 2 to 10. The surface with radius 2 is the most curved. The mean of errors on the entire imageset of each dataset is reported. Best viewed in colour.

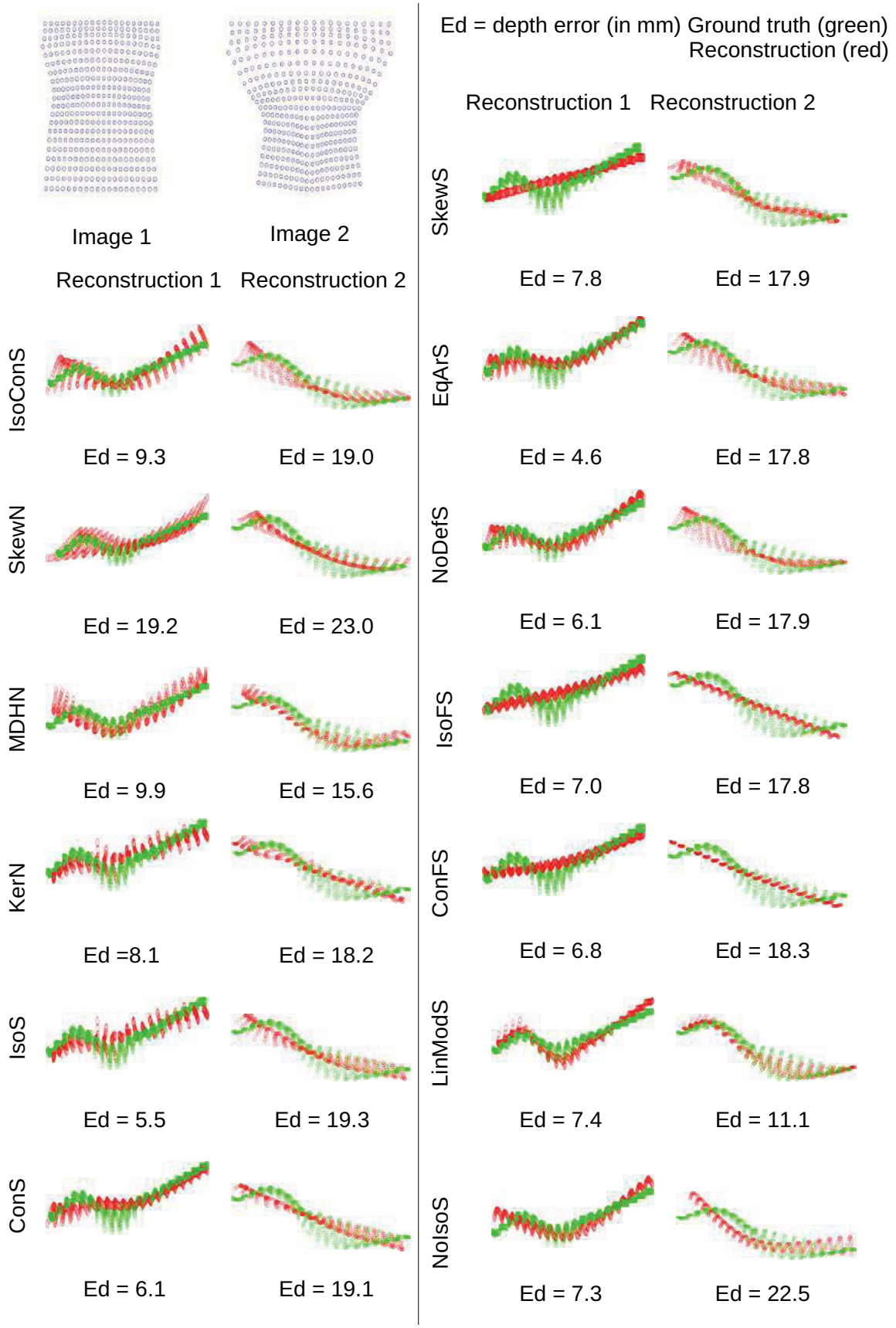


Figure 5.12: Error maps for surfaces 10 and 20 from the rubber dataset. The second image shows the maximum stretch of the rubber. The depth error maps show the difference in the reconstruction and ground truth. Best viewed in colour.

we do not perform the refinement step proposed for these methods. This is because our goal is to compare the analytical solutions of these methods to ours.

5.5.2.2 Balloon Dataset

This dataset consists of 20 images uniformly sampled from a video sequence of 80 images. We observe that the mean normal error for all NRSfM methods on this dataset is around 20 degrees which is very high as compared to the performance of these methods on other datasets. However, the mean depth error for these methods is quite comparable to the rest of the datasets. This indicates to the flattening of the reconstructed surface. However, **IsoConN** shows the best performance amongst compared NRSfM methods. Our SfT methods except **SkewS** show a very good performance on this dataset and their results are very similar. The reconstruction from **SkewS** is quite flat. **IsoFS**, **ConFS** and **NoIsoS**, all of these methods also lead to flat reconstructions. **LinModS** performed quite well on this dataset. Its performance is quite close to **EqArS** and **NoDefS** which are the best performing methods on this dataset. This dataset is near-affine and therefore, most of the methods did not do well. The reconstructions of two surfaces of this dataset using the compared methods are shown in figure 5.14.

5.5.2.3 Sock Dataset

This dataset consists of 20 images from a video sequence of a sock being stretched in one direction only. Analytical methods show a very good performance on this dataset. Amongst NRSfM methods, **SkewN** showed the best performance as the sock is undergoing almost skewless deformation. The reconstruction of normals by **IsoConN** is quite good but it cannot cope with the stretching of the surface and therefore, it leads to a higher depth error. **MDHN** and **KerN** did not do well on this dataset. SfT methods except **NoIsoS** (it completely broke on this dataset) showed a very good reconstruction even though the object was stretched. **NoDefS** shows the best performance. However, the rest of the methods are quite competitive. An important thing to note is that even though **IsoFS** and **ConFS** show the best performance in computing normals, their computation of depth is not as good as **IsoS** and **ConS**. This is because **IsoS** and **ConS** use the constraints of **IsoFS** and **ConFS** (given in equation (5.20)) along with additional constraints obtained by differentiating equation (5.20) which gives **IsoS** and **ConS** the liberty to reconstruct stretched surfaces better. The reconstructions of two surfaces of this dataset using the compared methods are shown in figure 5.15.

5.5.2.4 Tissue Dataset

This dataset contains a single image and therefore only SfT methods could be compared for this dataset. **NoDefS** shows the best performance on this dataset with the rest of the methods (except **SkewS**, **ConS** and **NoIsoS**) being close enough to the best solution. The reconstructions of the tissue using the compared methods are shown in figure 5.16.

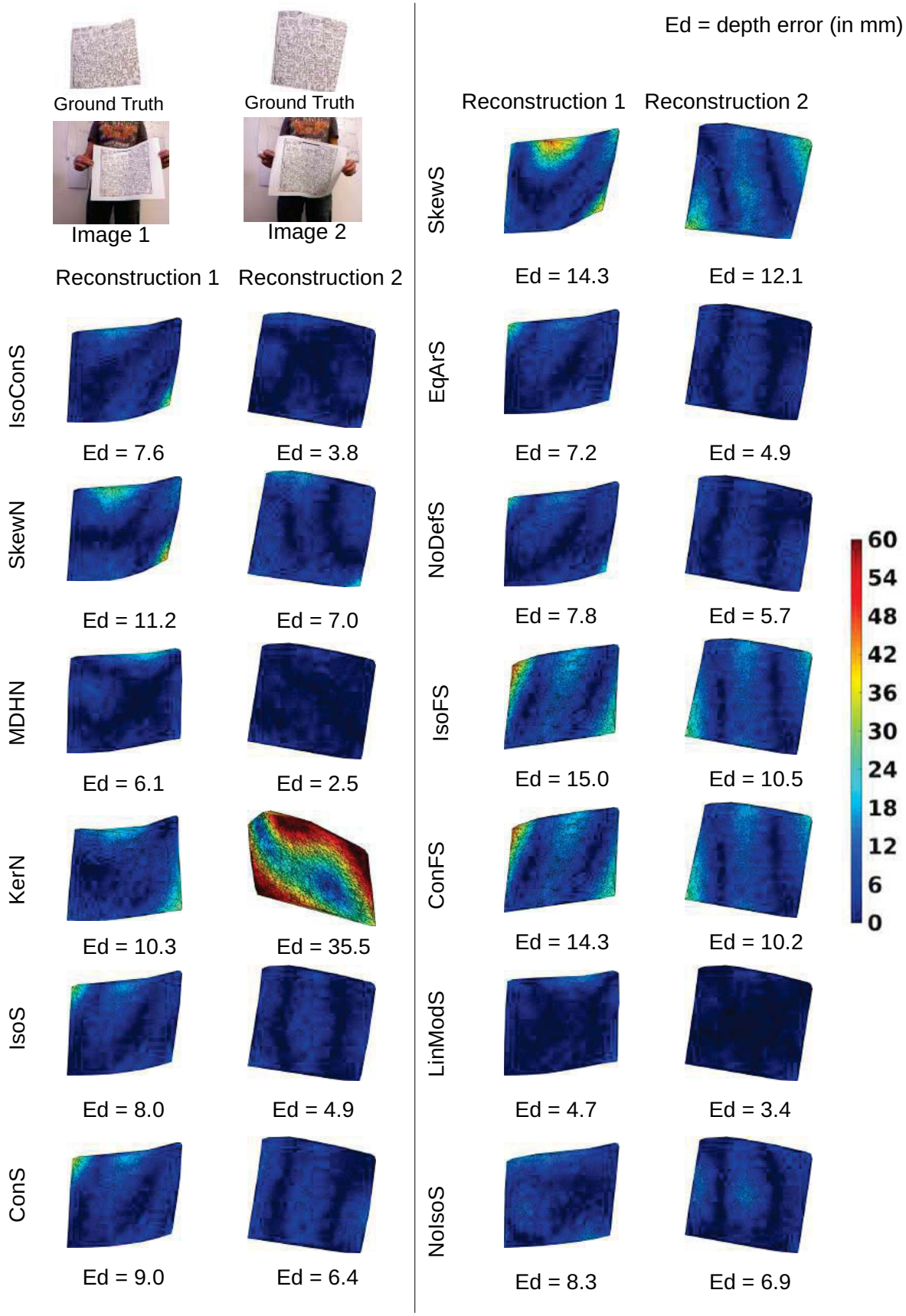


Figure 5.13: Error maps for two surfaces the paper dataset. The depth error maps show the difference in the reconstruction and ground truth. Best viewed in colour.

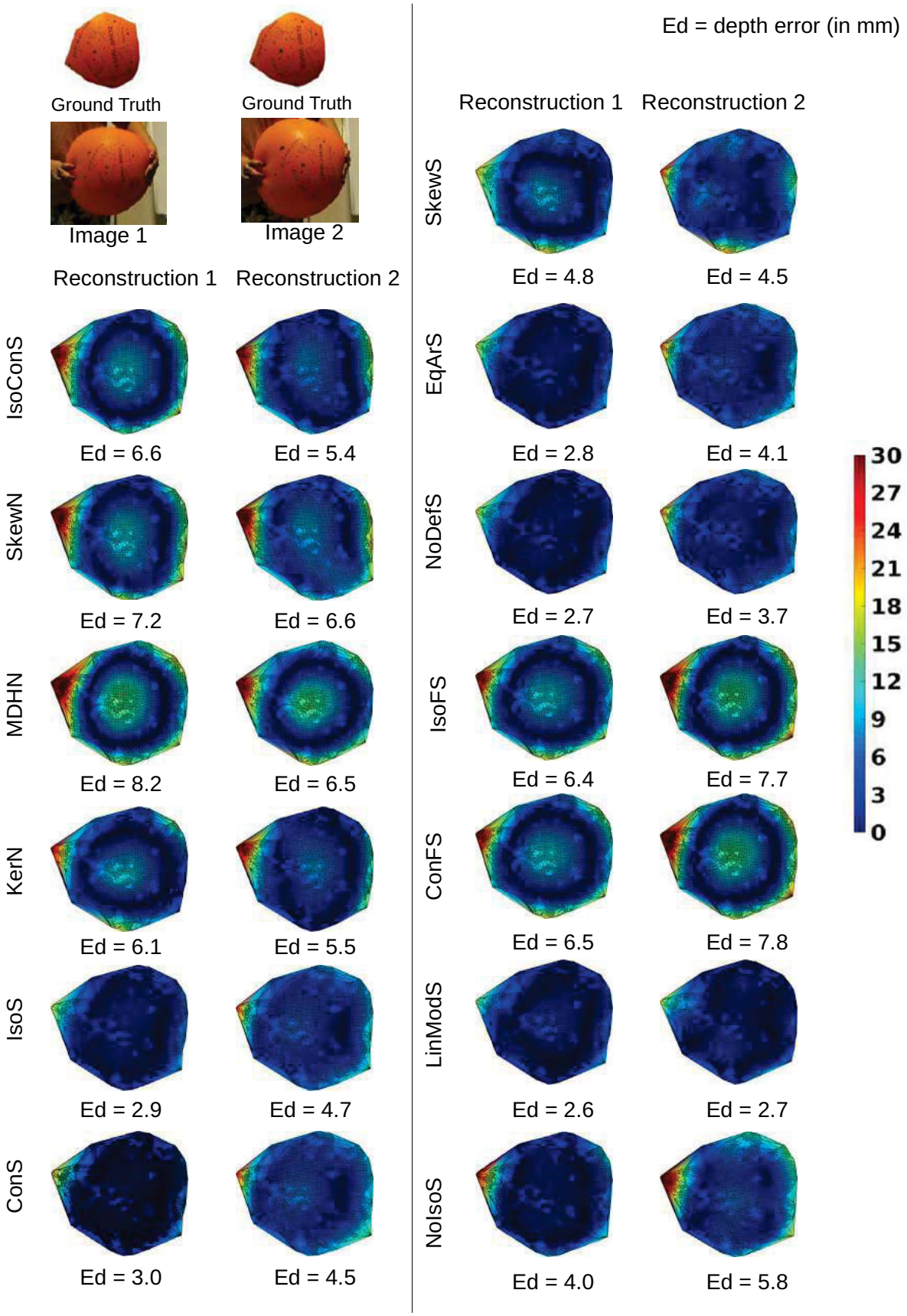


Figure 5.14: Error maps for two surfaces the balloon dataset. The depth error maps show the difference in the reconstruction and ground truth. Best viewed in colour.

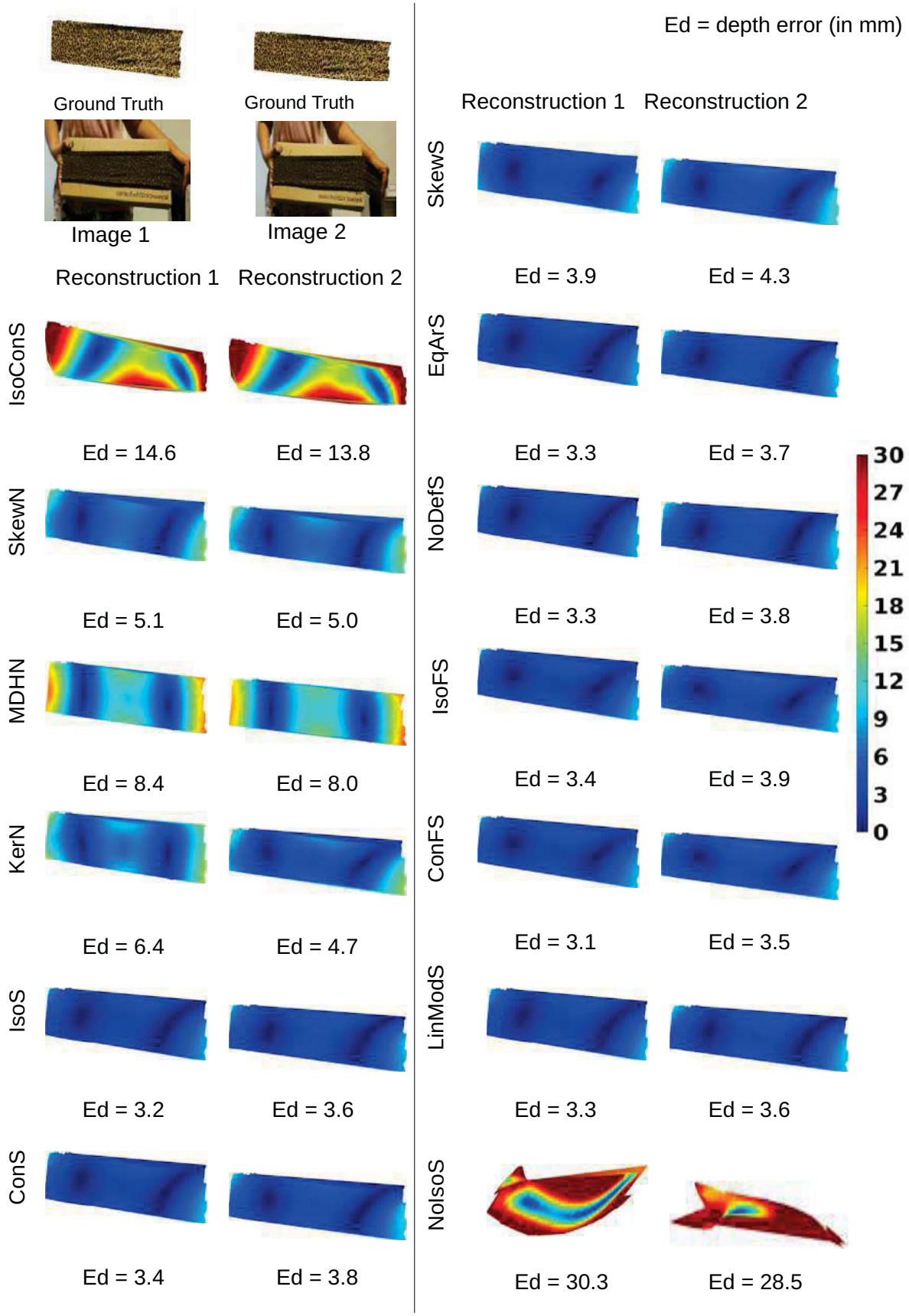


Figure 5.15: Error maps for two surfaces the sock dataset. The depth error maps show the difference in the reconstruction and ground truth. Best viewed in colour.

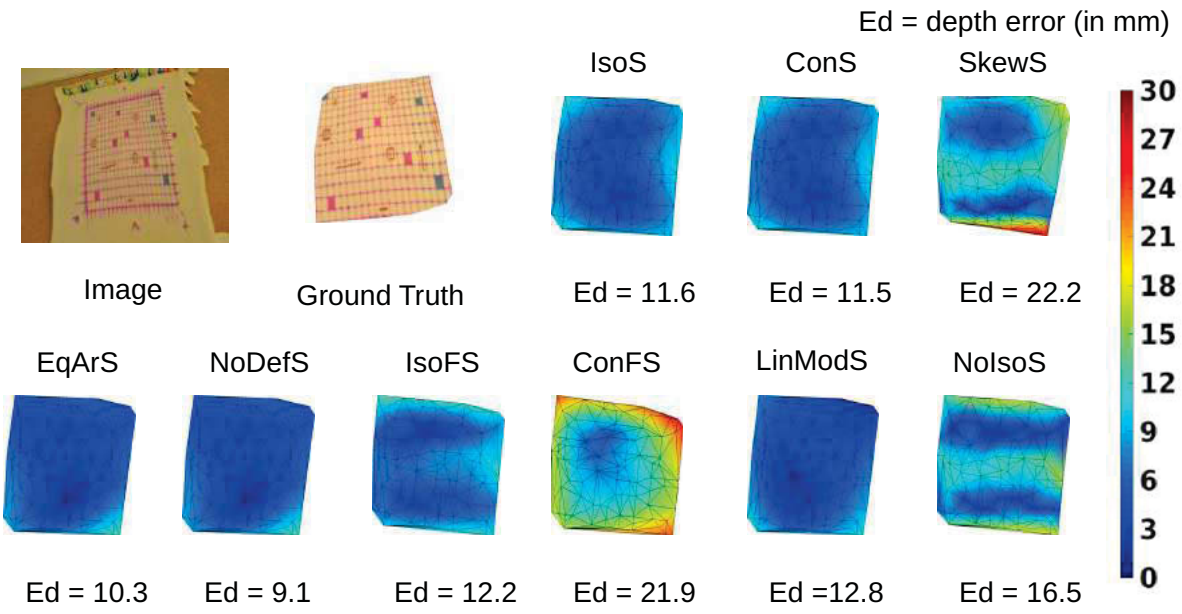


Figure 5.16: Error maps of reconstruction from all SfT methods of the tissue dataset. Best viewed in colour.

Discussion on smoothness. An important point to note here is that **NoIsoS** and **NoDefS** both reconstruct the surface without modelling deformation explicitly. However, we found **NoIsoS** to be unstable. [Bartoli and Özgür, 2016] proposed five solutions to cope with the stability issue. Also, it needs the input image to have a high amount of perspective in order to give decent results. The main idea of this method is to use smoothness (expressed in terms of the second-order differentials of the surface) in order to reconstruct surfaces. Our solution to **NoDefS** (along with other solutions) also relies on the second-order derivatives of the surfaces under the assumption of IL. Our experiments show that our solution to **NoDefS** is not unstable. We discuss the difference between the two solutions now. [Bartoli and Özgür, 2016] exploit smoothness by minimising the second order derivatives of the surface. For a function f described in equation (5.1), it tries to minimise $\mathrm{d}^2 f$. Using equations (5.3),(5.4) and (5.5), we write

$$\mathrm{d}^2 f = \mathrm{d}\left(e_1 \mathrm{d}x^1 + e_2 \mathrm{d}x^2\right) = \mathrm{d}e_1 \mathrm{d}x^1 + \mathrm{d}e_2 \mathrm{d}x^2. \tag{5.55}$$

Therefore, this solution tries to minimise $(\mathrm{d}e_1, \mathrm{d}e_2)$ whereas our solution (**NoDefS**) is obtained by imposing a structure to the components of $(\mathrm{d}e_1, \mathrm{d}e_2)$ (connection forms in equation (5.6)). This is the underlying difference between the two methods. This structure of $(\mathrm{d}e_1, \mathrm{d}e_2)$ cannot be studied without differential geometry which makes this framework a comprehensive study of deformations.

5.5.3 Summary of Experiments

The results of all the compared methods on each dataset are summarised in figure 5.9. On these datasets, the best performing NRSfM methods are **IsoConN** and **MDHN**. For SfT methods, our methods (except **SkewS**) showed a very good performance on all the datasets. **NoDefS** showed the best performance on most of the datasets. **LinModS** showed a very good performance on these datasets, it showed the best performance in the rubber and paper datasets. However, **NoDefS** and **EqArS** have been very competitive with **LinModS**. **SkewN** gives very good results on the sock dataset. This is because the sock is undergoing a skewless deformation. An important point to note about skewless deformation is that it should be along the two directions of the orthogonal frame basis of the surface. Therefore, **SkewN** and **SkewT** can perform better if the basis is orthogonal and is aligned to the direction of deformation. This means that the frame basis on a surface should be aligned to the direction of deformation which needs to be known apriori. Our SfT methods (except **SkewS**) are linear and therefore, computationally very cheap. In general, for 50 images and 400 points, it takes 2-5 seconds for these methods to evaluate the results. The computation time for our NRSfM methods is about 15-20 seconds for around 100 points tracked over 30 images. These computation times are reported on a standard PC.

5.6 Conclusions

We presented a theoretical modelling framework for deformable 3D reconstruction using differential geometry and Cartan's theory of connections. This includes the model-based and model-free cases. We showed how to obtain solutions to isometric, conformal (same as the isometric solution) and skewless NRSfM under the assumption of IP using this framework. We showed that equiareal NRSfM cannot be solved locally. We used this framework to find solutions to isometric, conformal, skewless and equiareal SfT under the assumption of IP. These solutions are computationally very cheap. Using this framework, we also proposed a solution to SfT (under the assumption of IL) without modelling the deformation explicitly. Our methods rely on the first and second-order derivatives of the warp function which are known to be unstable when obtained using sparse registration algorithms. We used a warp refinement method presented in [Pizarro et al., 2016]. As discussed in the previous chapter, it is a theoretical requirement for our methods to comply with the Schwarzian equations presented in [Pizarro et al., 2016]. Therefore, it makes sense to use this method for warp correction. Our NRSfM methods solve the minimum data case, have linear complexity in the number of points, handle missing data, work for both short and wide-baseline datasets, work with both large and small number of images. Our experiments showed that the proposed NRSfM and SfT methods outperform most of the compared methods in terms of accuracy and computation time.

Chapter 6

Volumetric Shape-from-Template

Summary

In this chapter we propose a solution to SfT for deformable volumetric objects. The objective is to infer an object's shape from a single image and a 3D object template. It uses the object's full volume to express the deformation constraints and reconstructs the object's surface and interior deformation. This is a challenging problem because for opaque objects, only a part of the outer surface is visible in the image. Inspired by mesh-editing techniques, we use an ARAP deformation model that softly imposes local rigidity. We formalise ARAP isometric SfT as a constrained variational optimisation problem which we solve using iterative optimisation. This chapter is based on our published work [Parashar et al., 2015].

6.1 Introduction

The objective in SfT is to obtain the object's deformed shape in the camera coordinate frame using a deformation constraint formulated from the object's physical material. Existing SfT methods use deformation constraints on the object's outer surface, whose thickness is considered infinitesimal. We thus call them thin-shell SfT methods. Thin-shell SfT is very well adapted to thin objects, such as a piece of paper or a balloon, whose outer surfaces may be well approximated by an open or a closed surface. Even though thin-shell SfT methods can handle thicker objects such as a foam ball, they do not fully exploit the strong constraints induced by the object's non-empty interior.

Volumetric SfT reconstructs the object's interior deformation (which is not reconstructed by thin-shell SfT) and reconstructs the object's outer surface more accurately than thin-shell SfT due to the stronger deformation constraint it uses. The biggest challenge for volumetric SfT is to reconstruct the object's back surface and interior.

We solve volumetric SfT using the ARAP deformation model. It maximises local rigidity while penalising stretching, sheering and compression. It helps in preserving the object's interior structure and enables to reconstruct the object's full outer surface and interior.

We present volumetric SfT as an unconstrained non-linear least-squares optimisation problem which can be solved using standard numerical solvers such as Levenberg-Marquardt. We propose two heuristic initialisation methods. Experimental results on synthetic and real data show that volumetric SfT improves accuracy to a large extent compared to state-of-the-art thin-shell SfT methods.

Chapter outline. We present the mathematical modelling of volumetric SfT in section 6.2. It describes the modelling of volumetric SfT model and the deformation. Section 6.3 defines volumetric SfT and proposes a solution. Section 6.4 discusses the experiments and section 6.5 concludes.

6.2 Mathematical Modelling

6.2.1 Geometric Model

Figure 6.1 shows a general diagram of volumetric SfT extending a thin-shell framework described in the previous chapter. We denote the 3D template as the volume $\mathcal{V}_T \subset \mathbb{R}^3$, the unknown deformed volume as $\mathcal{V}_S \subset \mathbb{R}^3$, and their respective outer surfaces as $\partial\mathcal{V}_T$ and $\partial\mathcal{V}_S$. We denote as $\mathcal{S} \subset \partial\mathcal{V}_S$ the deformed object's visible surface part, *i.e.*, the part which is directly observed in the input image $\mathcal{I} \subset \mathbb{R}^2$, and $\mathcal{T} \subset \partial\mathcal{V}_T$ the corresponding part in the template surface. We use a 2D surface parameterisation space $\mathcal{F}$, called the flattened template. This allows us to represent the template's outer surface $\partial\mathcal{V}_T$ by a known invertible embedding $\Delta \in C^2(\mathcal{F}, \mathbb{R}^3)$. In practice, $\mathcal{F}$ and Δ may be computed from $\partial\mathcal{V}_T$ by any flattening method; we use conformal flattening [Sheffer et al., 2005]. Using $\mathcal{F}$, the unknown

deformed surface $\mathcal{S}$ may be represented by an embedding $\varphi \in C^2(\mathcal{F}, \mathbb{R}^3)$. The deformation between $\mathcal{V}_T$ and $\mathcal{V}_S$ is the unknown mapping $\psi \in C^2(\mathcal{V}_T, \mathbb{R}^3)$.

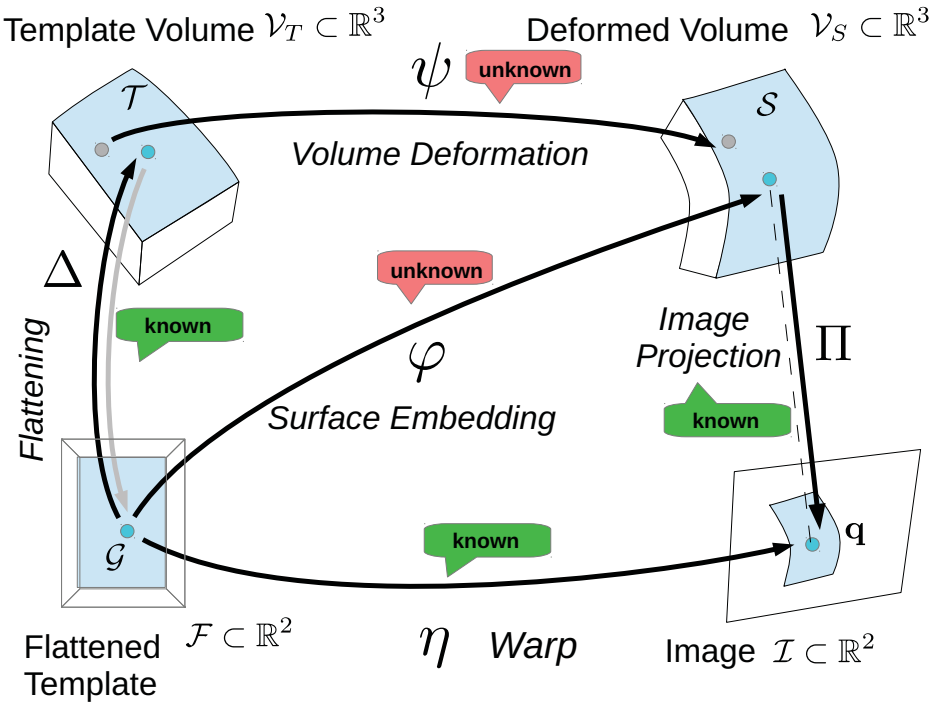


Figure 6.1: General diagram of volumetric SfT. The visible surface part is shown in blue.

The task in volumetric SfT is not only to compute the volume $\mathcal{V}_S$ of the deformed object, but to find a full volume deformation function $\psi \in C^2(\mathcal{V}_T, \mathbb{R}^3)$, matching points between the object's template and deformed states. This is a challenging task, as most of $\mathcal{V}_S$ is not directly observed in the image: assuming the object is opaque, the only visual information comes from the outer surface's visible part. The surface embedding φ may of course be directly recovered from the volume deformation ψ computed by volumetric SfT as $\varphi = \psi \circ \Delta$. Depending on the formulation, thin-shell SfT computes either the surface embedding φ [Bartoli et al., 2012] or a 3D surface deformation, which is a restriction of ψ to $\partial\mathcal{V}_T$ [Östlund et al., 2013]. The full volume deformation ψ cannot be directly recovered in either case. Our initialisation strategy for volumetric SfT involves inferring ψ from φ through two new solutions which we name volume interpolation.

A point in the 3D template is given by $\mathbf{P}$ and the corresponding point on the flat template is $\mathbf{p}$. In a similar way, a point on the deformed object is given by $\mathbf{Q}$. The corresponding point in the image is given by $\mathbf{q}$. Finally, we define $\eta \in C^2(\mathcal{F}, \mathbb{R}^2)$ as the registration warp between $\mathcal{F}$ and the image. Estimating the warp directly gives two pieces of information. First, it identifies the subset $\mathcal{G} \subset \mathcal{F}$ corresponding to the surface's visible part in the flattened template. Second, it establishes the reprojection constraint on φ and ψ as:

$$\eta = \Pi \circ \varphi = \Pi \circ \psi \circ \Delta, \tag{6.1}$$

where Π denotes perspective projection in coordinates normalised with respect to the camera's intrinsics $\Pi(\mathbf{Q}) = \frac{1}{Q^3}(Q^1\ Q^2)^\top$ with $\mathbf{Q} = (Q^1\, Q^2\, Q^3)^\top$.

6.2.2 Deformation Model

Thin-shell isometry allows SfT to resolve the visible surface part uniquely [Bartoli et al., 2012] and to extrapolate the non-visible surface part [Östlund et al., 2013]. Applied to an object's interior volume, isometry yields the following differential constraint on the mapping ψ:

$$\mathbf{J}_\psi^\top \mathbf{J}_\psi = \mathbf{I}_{3\times 3}. \tag{6.2}$$

According to the Mazur-Ulam theorem [Rassias and Šemrl., 1993], equation (6.2) constrains ψ to be a rigid transformation. So as to model deformations, one must relax equation (6.2). One possibility is the so-called ARAP heuristic [Sorkine and Alexa., 2007], which means searching for ψ such that $\left\|\mathbf{J}_\psi^\top \mathbf{J}_\psi - \mathbf{I}_{3\times 3}\right\|_p^2$ (p denotes the type of norm) is minimised over $\mathcal{V}_T$. We propose to combine ARAP with the reprojection constraint to preserve the object's local structure while driving its deformation to comply with the image constraints.

6.3 Volumetric Shape-from-Template

We present ARAP volumetric SfT, which finds the deformation ψ that transforms the volume $\mathcal{V}_T$ into the unknown volume $\mathcal{V}_S$ whose surface is partially observed in $\mathcal{I}$.

6.3.1 Formulation and Non-Convex Solution

Problem formulation. Combining the reprojection constraint (6.1) with ARAP leads to the following variational problem:

$$\min_\psi \rho \underbrace{\int_{\mathcal{V}_T} \left\|\mathbf{J}_\psi^\top \mathbf{J}_\psi - \mathbf{I}\right\|_p^2 d\mathcal{V}_T}_{\text{ARAP penalty}} + (1-\rho) \underbrace{\int_{\mathcal{G}} \|\eta - \Pi \circ \psi \circ \Delta\|_2^2 \, d\mathcal{G}}_{\text{Reprojection}}. \tag{6.3}$$

The reprojection constraint is convex, but the ARAP penalty is not. Problem (6.3) is thus difficult to solve, as it involves integrals and equality constraints. Local analytical solutions, as the ones proposed in the previous chapter and in [Bartoli et al., 2012] for thin-shell SfT, are not applicable at non-visible points since they do not have a data term. This is because the reprojection constraint applies on the visible surface part $\mathcal{S}$ only, corresponding to the subset $\mathcal{G}$ of the flattened template.

Discretisation and optimisation. We evenly discretise the template volume $\mathcal{V}_T$ with a set of 3D points $\mathcal{P}_{\mathcal{V}_T}$. We define the deformation functional $\epsilon_d[\psi]$:

$$\epsilon_d[\psi] = \frac{1}{|\mathcal{P}_{\mathcal{V}_T}|} \sum_{\mathbf{P}\in\mathcal{P}_{\mathcal{V}_T}} \left\|\mathbf{J}_\psi^\top(\mathbf{P})\mathbf{J}_\psi(\mathbf{P}) - \mathbf{I}_{3\times 3}\right\|_p^2, \tag{6.4}$$

where $|\mathcal{P}_{\mathcal{V}_T}|$ represents the size of set $\mathcal{P}_{\mathcal{V}_T}$.

We write $\epsilon_r[\psi]$ over a regular discretisation $\mathcal{P}_\mathcal{G}$ of $\mathcal{G}$:

$$\epsilon_r[\psi] = \frac{1}{|\mathcal{P}_\mathcal{G}|} \sum_{\mathbf{p}\in\mathcal{P}_\mathcal{G}} \|\eta(\mathbf{p}) - \Pi(\psi(\Delta(\mathbf{p})))\|_2^2. \tag{6.5}$$

Finally, we optimise the following unconstrained non-linear least-squares problem:

$$\psi \;=\; \arg\min_{\psi} \rho\epsilon_d[\psi] + (1-\rho)\epsilon_r[\psi] \qquad 0 < \rho < 1, \tag{6.6}$$

where ρ is a weight that balances the ARAP penalty and the reprojection constraint.

In order to find a numerical solution to problem (6.6), we use a parametric representation of the solution $\tilde{\psi} \in C^2(\mathcal{V}_T \times \mathbb{R}^n, \mathbb{R}^3)$, where n is the dimension of the parameter space. Let $\mathbf{L} \in \mathbb{R}^n$ be the parameter vector and $\mathbf{Q} \in \mathcal{V}_T$, we have $\tilde{\psi}(\mathbf{Q}, \mathbf{L}) \in \mathcal{V}_S$. We have multiple choices for $\tilde{\psi}$ such as the popular linear basis expansion representations (the NURBS [Piegl, 1991], the Thin-Plate Splines (TPS) [Bookstein, 1989], the B-Spline [Carl and Boor, 1978], tetrahedron mesh displacements, etc.). We use the TPS representation.

Problem (6.6) is then optimised using Levenberg-Marquardt. Iterative methods can be highly accurate but because problem (6.6) is non-convex due to the ARAP penalty, the iterations may converge to a non-global minimum. Therefore, it is important to provide an initial solution close to the global minimum.

6.3.2 Convex Initialisation

Our initialisation strategy finds an approximate solution ψ_0 to problem (6.3) in two main steps.

1) **Isometric thin-shell SfT.** We first compute the embedding φ that represents the visible surface $\mathcal{S}$. We approximate the deformation from $\mathcal{T}$ to $\mathcal{S}$ by thin-shell isometry, giving the following problem reformulation:

$$\text{Find } \varphi \text{ s.t. } \begin{cases} \mathbf{J}_\varphi^\top \mathbf{J}_\varphi = \mathbf{J}_\Delta^\top \mathbf{J}_\Delta \\ \eta = \Pi \circ \psi \circ \Delta \end{cases} \quad \text{on } \mathcal{G}, \tag{6.7}$$

where $\mathbf{J}_\varphi$ is a 3×2 matrix. Problem (6.7) has an analytical solution given in [Bartoli et al., 2012].

2) **Volume interpolation.** We use φ to infer ψ representing the full volume deformation. We propose two strategies.

i) Global Smoothness (GS). Our first strategy is based on the assumption that the deformation of the volume is smooth. We can, therefore, formulate the problem as finding the smoothest volumetric deformation such that the deformation at the surface agrees with the

solution from thin-shell SfT. We write the discretised transport error:

$$\epsilon_e[\psi] = \frac{1}{|\mathcal{P}_\mathcal{G}|} \sum_{\mathbf{p}\in\mathcal{P}_\mathcal{G}} \|\varphi(\mathbf{p}) - \psi(\Delta(\mathbf{p}))\|_2^2. \tag{6.8}$$

Because φ was computed in step *1)*, this is a linear least-squares cost in ψ. We then compute ψ_0 as the solution of the following system:

$$\psi_0 = \arg\min_{\psi} \alpha\epsilon_e[\psi] + (1-\alpha)\epsilon_s[\psi] \quad 0 < \alpha < 1, \tag{6.9}$$

where $\epsilon_s = \iint \|\frac{d^2\psi}{d\mathbf{p}^2}\|_2^2 d\mathbf{p}$ is a smoothing term called the bending energy and α is a weight balancing the transport error and smoothness. As in the non-convex solution, we use a TPS representation of ψ. The bending energy is then a quadratic function of the TPS parameters [Bookstein, 1989], making problem (6.9) linear least-squares, thus convex and easily solvable. GS is a natural way of initialising ψ from φ, but as smoothness is the only constraint it uses to propagate the visible surface deformation, it may spoil the object's inner local structure by causing local shear, shrinking and extension. Our second volume interpolation method addresses this issue.

ii) Local Rigidity (LR). This method is based on the idea that from thin-shell SfT, we can compute a local rigid transform at every point on the visible surface to propagate shape through the object's volume, in an ARAP manner. The key idea is to initialise ψ on the surface's visible part from φ, and use local rigidity to iteratively 'complete' ψ. This is implemented by iteratively drawing local rigid transformations to locally extrapolate the deformation. Concretely, we first find correspondences to all points in $\mathcal{P}_{\mathcal{V}_T}$ (which may be seen as a discretisation of ψ) and then fit a continuous parametric representation of ψ. We write the corresponding point of $\mathbf{P} \in \mathcal{P}_{\mathcal{V}_T}$ as $\mathbf{Q}(\mathbf{P})$.

We first use a Delaunay triangulation of the point set $\mathcal{P}_{\mathcal{V}_T}$ to define a tetrahedral mesh. A given tetrahedron has four vertices, which we denote as $\mathbf{P}_{n_1}$, $\mathbf{P}_{n_2}$, $\mathbf{P}_{n_3}$ and $\mathbf{P}_{n_4}$. Drawing

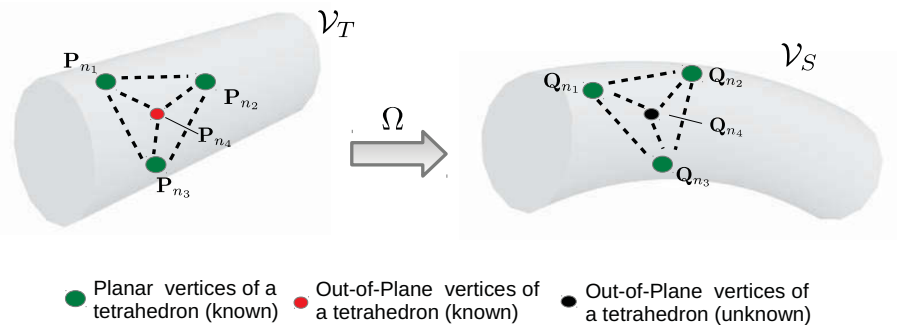


Figure 6.2: Volume interpolation using Local Rigidity.

a local rigid transformation is achieved by selecting a tetrahedron which has three vertices, say the first three ones, lying in the 'completed' domain of ψ, for which $\mathbf{Q}_{n_i} = \mathbf{Q}(\mathbf{P}_{n_i})$ exist for $i = 1, \ldots, 3$. At the early iterations, this means that these three vertices will have to be

in the surface's visible part, and that $\mathbf{Q}_{n_i} = \varphi(\Delta^{-1}(\mathbf{P}_{n_i}))$, $i = 1, \ldots, 3$. From these three correspondences $\{\mathbf{P}_{n_i} \leftrightarrow \mathbf{Q}_{n_i}\}$, $i = 1, \ldots, 3$, we fit a rigid transform Ω in the least-squares sense using [Horn et al., 1988]. Completing ψ by local rigidity is then simply done by setting $\mathbf{Q}(\mathbf{P}_{n_4}) = \Omega(\mathbf{P}_{n_4})$. At each iteration, we cycle through all tetrahedra with three vertices lying in the completed domain of ψ. This obviously causes the fourth vertex of many tetrahedra to receive multiple predictions, as several tetrahedra may share it as their single unknown vertex. In order to approximate ARAP as best as possible, we keep the prediction for which Ω was fitted with the lowest error. We stop the iterations when all points in $\mathcal{P}_{\mathcal{V}_T}$ have been given a correspondence. We finally define the discretised transport error:

$$\epsilon'_e[\psi] = \frac{1}{|\mathcal{P}_{\mathcal{V}_T}|} \sum_{\mathbf{P} \in \mathcal{P}_{\mathcal{V}_T}} \|\mathbf{Q}(\mathbf{P}) - \psi(\mathbf{P})\|_2^2, \tag{6.10}$$

and obtain ψ_0 as the solution of the following optimisation problem:

$$\psi_0 = \arg\min_{\psi} \alpha \epsilon'_e[\psi] + (1 - \alpha)\epsilon_s[\psi] \quad 0 < \alpha < 1, \tag{6.11}$$

where α is a weight balancing the transport error and smoothness. Equation (6.11) is linear least-squares, thus convex and easily solved.

6.4 Experimental Results

We report experiments with synthetic data and three sets of real data with different geometries and materials: a woggle, a sponge and an arm. The refinement solution (6.3) is tested using the L1 and L2 norms, and is then called L1-refinement and L2-refinement respectively. The two initialisation solutions are called GS-initialisation and LR-initialisation. We also compare with two isometric thin-shell SfT methods [Bartoli et al., 2012; Östlund et al., 2013], which were discussed as being representative of the state-of-the-art in the introduction. We use a constant weight $\rho = 0.005$ in the refinement problem (6.6) (for both L1-refinement and L2-refinement) and a constant weight $\alpha = 0.0001$ in both equation (6.9) for GS-initialisation and equation (6.11) for LR-initialisation. We noticed that the algorithms were not very sensitive to these values up to an order of magnitude.

6.4.1 Synthetic Datasets

We test our method for volumetric SfT in various conditions of noise, deformation and correspondences. We simulate a box of dimension $20 \times 20 \times 10 cm^3$ and deform it by bending each of its layers along a vertical rule with some varying bending angle. The higher the bending angle, the more important the box's deformation. If the bending angle is zero, the box is undeformed. We then create a virtual image of the box by projecting it using a perspective camera and add noise to the pixels. The default bending angle is 10 degrees. The results are shown in figure 6.3, and are averaged over multiple runs for each geometric configuration.

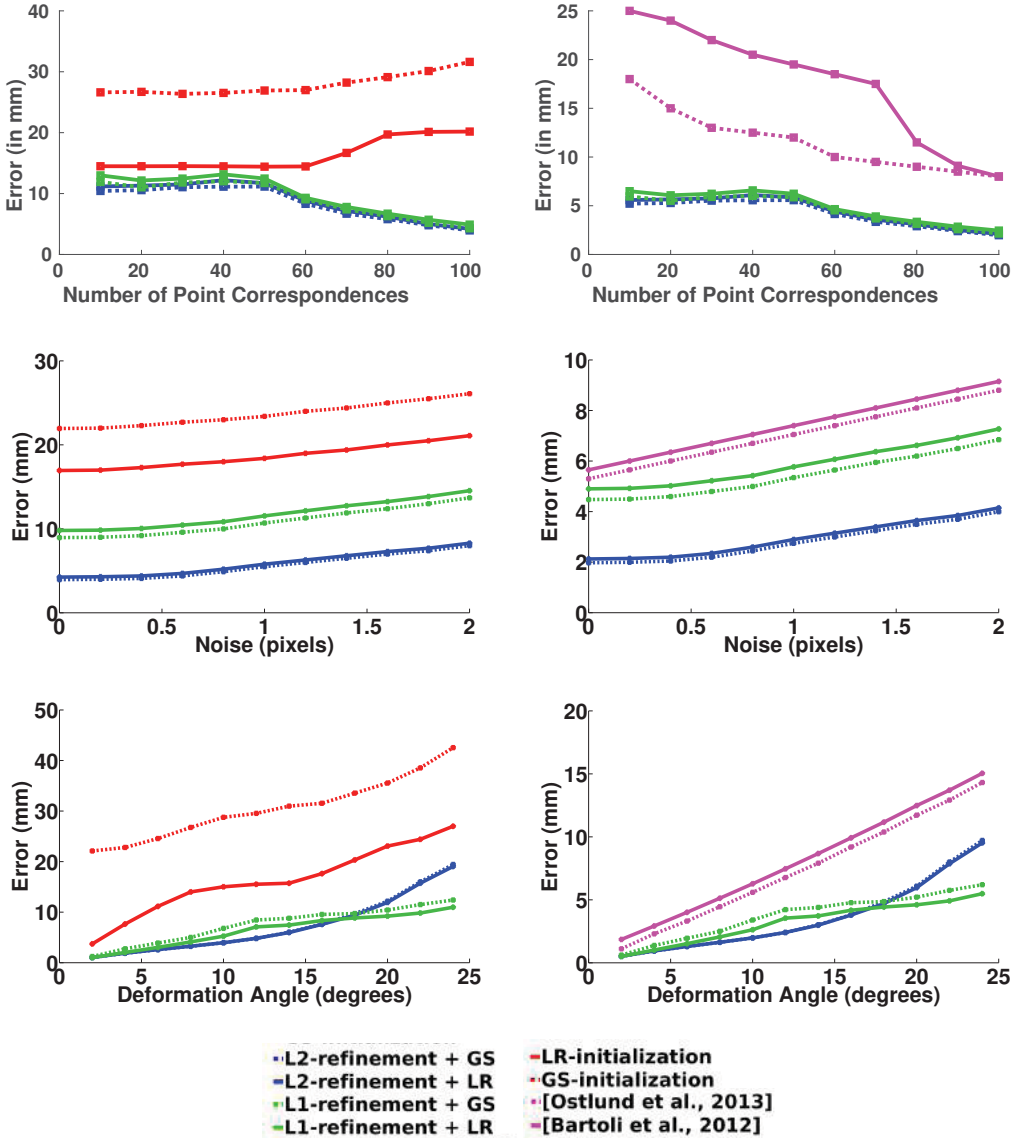


Figure 6.3: Synthetic data experiments.The graphs on the left show the 3D volume error and the ones on the right show the error on the 3D visible surface. Best viewed in colour.

The three graphs on the left column of figure 6.3 show the 3D volume error in mm, computed as the Root Mean Square Error (RMSE) over a dense grid of points sampled over the object's outer surface and interior. The results on these graphs thus only concern the proposed volumetric SfT methods. We observe that the refinement methods all do significantly better than all initialisation methods. LR-initialisation does consistently and substantially better than GS-initialisation. This is explained by the fact that LR-initialisation follows the ARAP methodology for local propagation, while GS-initialisation simply uses smoothness, which is a weaker constraint. L2-refinement does generally better than L1-refinement, except when the deformation increases beyond a certain point. All methods degrade with the amount of deformation and noise. Increasing the number of points improves the refinement methods but slightly degrades the initialisation methods. The three graphs on the right column of figure 6.3 show the visible surface error in mm, computed as the RMSE over a dense grid of points sampled over the object's visible surface.

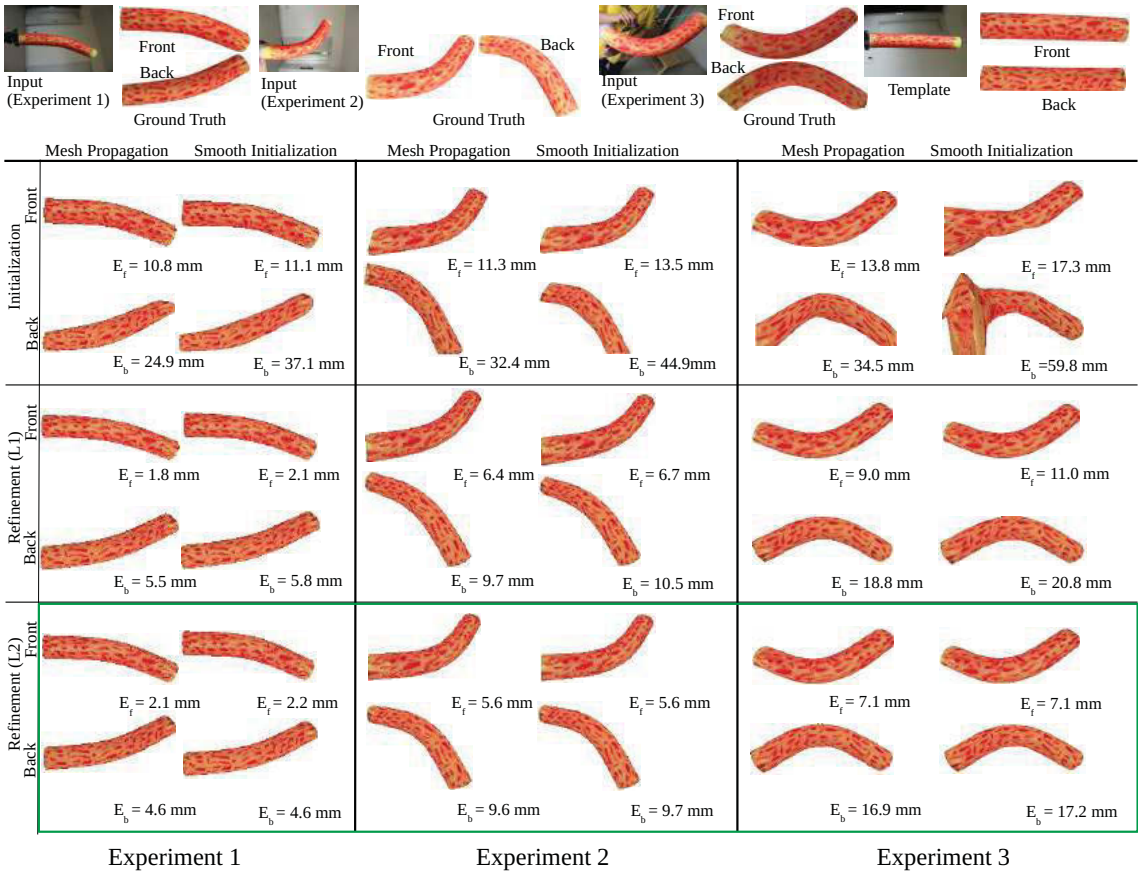


Figure 6.4: Results on the woggle. The green boxes show the best performing algorithm for each deformation level.

The same observations which we made for the refinement methods on the 3D volume error can be made, and the general trends also apply to the two tested thin-shell SfT methods. Importantly, we observe that volumetric SfT does consistently and in several case substantially better than thin-shell SfT, even if the measured error concerns only the visible surface part,

which is theoretically handled well by both types of methods. This means that the extra constraints used in volumetric SfT compared to thin-shell SfT have a very positive influence on this part of the reconstruction too.

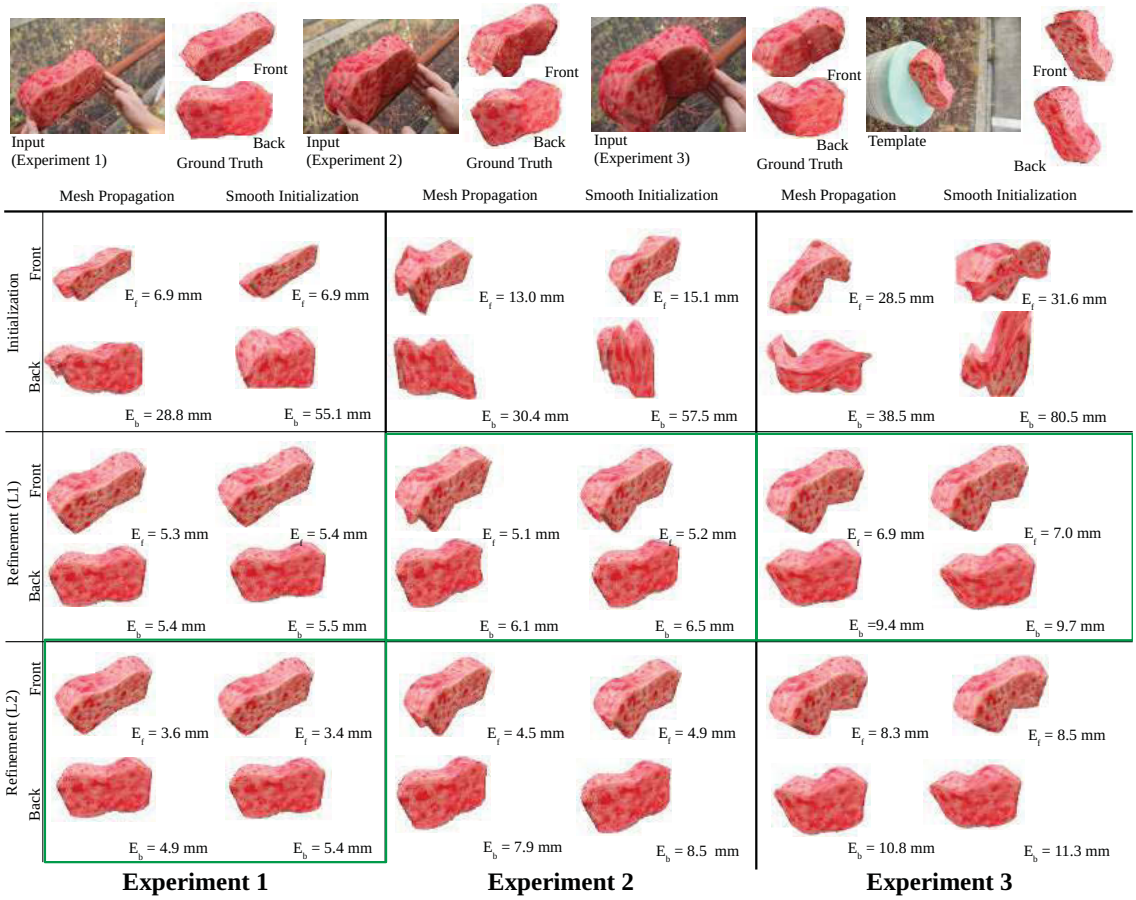


Figure 6.5: Results on the sponge. The green boxes show the best performing algorithm for each deformation level.

6.4.2 Real Datasets

We evaluate the performance of the methods with three real-world objects captured across a range of deformed states.

6.4.2.1 Test Data and Ground Truth Acquisition

The three objects are a foam tube called a woggle (figure 6.4), a sponge (figure 6.5) and a human arm (figure 6.6). We construct the 3D template of each object using Photoscan, a dense rigid SfM package [Photoscan, 2014]. To achieve this we photograph the objects in a rigid pose from a number of different viewpoints in order to capture the full 3D geometry (we use 47, 55 and 78 images for the three objects respectively). We apply a small amount of manual post-processing to fill holes and make the templates watertight. Then we physically apply forces to the real objects to obtain a set of deformed shapes, which we grouped into

Error Analysis (in mm)	Tube Data			Sponge Data			Arm Data		
	Low	Med.	High	Low	Med.	High	Low	Med.	High
Thin-shell SfT[1]	11.8	16.0	19.8	8.0	15.3	31.8	18.5	20.1	33.4
Thin-shell SfT[2]	10.4	11.4	12.7	6.0	25.5	35.4	15.5	18.9	35.7
GS-initialisation	11.1	13.5	17.3	6.9	15.1	31.6	15	18.3	31.6
LR-initialisation	10.8	11.3	13.8	6.9	13.0	28.5	13.5	14.2	25.6
L1-refinement + GS	2.1	6.7	11.0	5.4	5.2	7.0	3.4	4.1	5.2
L1-refinement + LR	1.8	6.4	9.0	5.3	5.1	6.9	2.9	4.1	5.1
L2-refinement + GS	2.2	5.6	7.1	3.4	4.9	8.5	3.0	4.5	7.3
L2-refinement + LR	2.1	5.6	7.1	3.6	4.5	8.3	2.7	4.2	6.8

Table 6.1: 3D visible surface error E_f for the datasets shown in figures 6.4, 6.5 and 6.6. Thin-shell SfT[1] [Bartoli et al., 2012] and Thin-shell SfT[2] [Östlund et al., 2013].

three levels: low, medium and high deformation. For each level we compute the ground truth shape by photographing again the deformed object from approximately 50 viewpoints and then running Photoscan. Because Photoscan provides the reconstructed object and the pose for each camera image, this provides us with the ground truth shape of the object's outer surface (including the back surface) in camera coordinates.

Similarly to the vast majority of previous SfT methods, ours takes as input point correspondences between the 3D template and the input image. These can be computed automatically using for instance SIFT combined with outlier detection [Pilet et al., 2005; Pizarro and Bartoli, 2012]. However, to keep the results independent of the matching algorithm, we define correspondences manually. For the three objects this gives between 50 to 350 correspondences per image. We click between 30 and 40 correspondences per image and create the others using TPS interpolation [Bookstein, 1989].

6.4.2.2 Performance Metrics and Method Comparison

We calculate two types of 3D errors, E_f and E_b, both expressed in mm, for the visible and non-visible surface parts respectively, as the RMSE discrepancy between the true and reconstructed 3D points at the correspondences. For each of the three datasets, and each of the three deformation levels, the top of each figure shows the template, the input image and the ground truth shape. On each figure, the deformation goes through low, medium and high level from left to right. The rows then show the results of both initialisation methods and their use to initialisation both refinement methods, giving a total of six combinations.

The 3D errors E_f, including the two thin-shell methods, are finally summarised in table 6.1.

We observe that LR-initialisation gives consistently better results than GS-initialisation, which is in accordance with our observations made on simulated data. The difference becomes very important for stronger deformations. This has a very small impact on the refinement results, for both refinement methods. We can observe small differences between the two refinement methods. However, refinement+LR converges faster than refinement+GS because

Figure 6.6: Results on the arm. The green boxes show the best performing algorithm for each deformation level.

LR-initialisation is closer to the correct solution. However, none of them is consistently better than the other, even if for the woggle L2-refinement is slightly more accurate, whereas for the sponge and arm L1-refinement is slightly more accurate. The 3D error E_f for volumetric SfT for both refinement methods is consistently smaller than for thin-shell SfT. Depending on the dataset and the deformation level, it is between two and ten times smaller. This confirms our observations made on simulated data that, even if the surface's visible part is handled naturally by both volumetric and thin-shell SfT, the stronger volume deformation constraints used by the former allows it to obtain a much more accurate result.

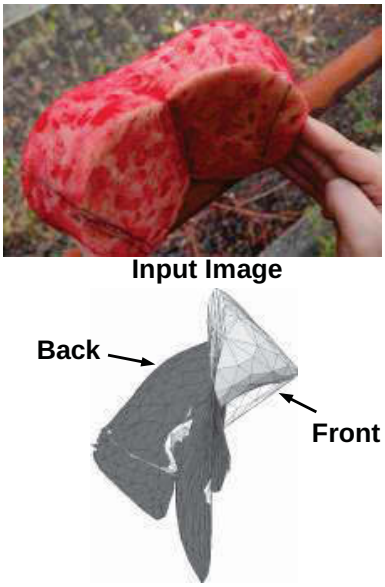


Figure 6.7: Testing with the top image led to the local minimum shown in the bottom image.

The results shown in figures 6.4, 6.5 and 6.6 show that the reconstructed object shape is visually close to the true shape. This means that volumetric SfT could allow a user to handle a physical object as a proxy interactor in applications such as virtual shape editing. Quantitatively, the woggle, sponge and arm are 37 cm, 15 cm and 20 cm long, respectively. The relative highest error over the whole reconstructed volume deformation, for the highest level of deformation, is thus smaller than 5%, 7% and 5% of the objects' size, respectively. Our unoptimised MATLAB implementation on a standard desktop with 3.1GHz processor takes between 10 - 25 seconds for the refinement to converge. The computation time for LR-initialisation is 3-5 seconds while for GS-initialisation it is 1-2 seconds.

In practice, volumetric SfT always converges with LR or GS-initialisation. But, in order to create conditions of failure, we initialised the refinement at equation (6.11) very far from the optimal solution. Figure 6.7 shows a failure case in the sponge dataset.

6.5 Conclusions

We presented volumetric SfT, which reconstructs an object from a single image and a 3D template, by using deformation constraints on the object's outer surface and interior. Pre-

vious thin-shell SfT methods use constraints on the object's outer surface only. Volumetric SfT is thus to be used with non-empty and non-flat objects. We proposed an implementation of volumetric SfT using ARAP. Our implementation uses non-convex refinement and has an initialisation procedure following an ARAP propagation of a surface deformation obtained by thin-shell SfT through the object's volume. Our method has significantly more accurate results than state-of-the-art isometric thin-shell SfT, reducing the error of an order of magnitude in some cases. Experiments with synthetic and real data show that our method has a typical maximum relative error of 5% in reconstructing the deformation of an entire object, including its back and interior for which no visual data is available. ARAP volumetric SfT opens the way to doing Human-Computer Interaction using a proxy object such as a cushion and a simple monocular webcam. Volumetric SfT may also be instantiated with other deformation models, such as biomechanical models.

Chapter 7

Conclusions

In this thesis, we presented solutions to thin-shell deformable 3D reconstruction in both model-based (SfT) and model-free (NRSfM) cases. We also proposed an SfT solution for reconstructing volumetric objects. We conclude the deformable 3D reconstruction for thin-shell and volumetric objects and also discuss the future prospects.

7.1 Thin-Shell Deformable 3D Reconstruction

We proposed a unified framework for modelling NRSfM and SfT using differential geometry and Cartan's theory of connections. This framework is general and easy to use for the physics-based modelling of various kinds of deformations. We also proposed a deformation model (skewless deformation) and showed how to perform deformable 3D reconstruction with the skewless deformation prior. We presented a solution to SfT which does not require an explicit modelling of deformation. We also showed that this framework is coherent with existing SfT solutions [Bartoli et al., 2015]. It is also coherent with our solution to NRSfM (proposed in chapter 4) for objects undergoing isometric deformations, which was developed using Riemannian geometry. This method, in fact, served as an inspiration to look beyond isometry in deformable 3D reconstruction.

Our methods are a significant improvement on the state-of-the-art in terms of accuracy, simplicity, computational complexity and applicability to wider situations. These methods are applicable to both wide and short-baseline images, handle a large number of images, yield a very accurate reconstruction with very few images and deal with missing data and occlusions implicitly.

These methods are local and can be solved analytically. Therefore, one can conclude that local and analytical methods are very quick and accurate. They are capable of playing a very important role in achieving real-time solutions to NRSfM and SfT without using high-performance computing devices.

This thesis is a new step towards local and analytical construction of NRSfM and SfT solutions. However there is still a lot to be explored about this local framework and its

applicability to real-life 3D reconstruction scenarios. Now we discuss future prospects:

1) Image embedding. We have proposed an embedding to describe the surface. This embedding is very efficient but one needs to explore more embeddings in order to find an efficient way to model the surfaces.

2) Warp issue. We use warps in all the solutions we proposed. Warps can be a source of error (especially when higher order derivatives are required). We refine the warps according to [Pizarro et al., 2016]. However, they are the bottleneck of our methods. We suspect that higher accuracy in warps can be achieved by using learning methods such as Convolutional Neural Networks.

3) Warp correction. Our methods rely on the second-order derivatives of the warps which are usually a big source of error. Like [Pizarro et al., 2016] presented a warp correction method based on Schwarzian equations, other methods are also possible in the model-based scenario.

4) Combining visual cues. So far, our framework only exploits inter-image motion. Some of the deformable reconstruction methods like [Wang et al., 2016] use other visual cues, such as shading and are capable of reconstructing challenging objects. It would be interesting to find out if other cues, like contours and shading can also be used in our framework.

5) Non-smooth objects. Since our methods are local, they need the surfaces to be locally smooth only. Our methods could be extended to non-smooth objects and a reconstruction similar to [Yu et al., 2015] be achieved. Obtaining warps for non-smooth objects should be explored.

6) Real-time NRSfM and SfT solutions. Since the local methods are computationally very efficient, making real-time applications with these methods should be explored.

7.2 Volumetric Deformable 3D Reconstruction

The challenge for volumetric reconstruction is to reconstruct the parts of the objects that are not seen in the camera: the interior and the back surface. It is, therefore, not possible to reconstruct volumes in a model-free case. We presented the first SfT solution to reconstruct volumes. It involved a non-convex optimisation which could be initialised by two of our proposed methods. We also showed that volumetric SfT yields a better reconstruction than the thin-shell SfT methods. This is because volumetric constraints are much stronger than the constraints on the surfaces.

A complete volumetric reconstruction should yield a more accurate registration of objects that lie in the interior, such as tumours inside the body organs. However most of the body organs do not deform near-isometrically, therefore it is important to model volumetric SfT with a deformation model less strict than ARAP. One such deformation model is conformity. In future, we intend to explore volumetric SfT with incompressibility (volume-preserving) deformations. This constraint is widely used in biomechanical modelling. [Delingette, 1998] discussed the aptness of this constraint for modelling soft body organs.

Appendices

Appendix A

The Metric Tensor

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

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

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

where dx, dy and dz represent the components of $\vec{v}$ in x, y and z coordinates respectively and the identity matrix in equation (A.1) is the metric tensor. The metric tensor is denoted as $\mathbf{g}$. The identity metric tensor implies that the distances remain constant as one moves along the coordinate frame. Now if we measure the length of the same infinitesimal vector $\vec{v}$ in a spherical coordinate system (r, θ, ϕ), we have:

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

Here, the metric tensor $\mathbf{g}$ is not the identity and changes at each point. This is necessary in order to measure the same distances at various locations in the coordinate frame. Fig. A.1 shows two points A and B represented on a spherical coordinate system. Moving these points to A' and B', in the direction of the $r-$coordinate, will also change the distance between them. However, this does not happen in a Cartesian coordinate system. On further elaboration, we see that the change of coordinates from Cartesian to spherical (described by

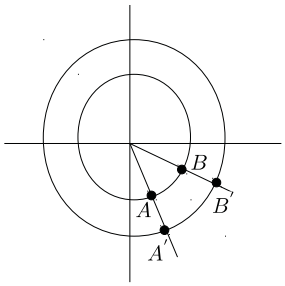


Figure A.1: Translating points A and B in spherical coordinates.

the transformation function f), results in the following expression for the metric tensor:

$$(x, y, z) = f(r, \theta, \phi) \quad \mathbf{g} = \mathbf{J}_f^T \mathbf{J}_f \tag{A.3}$$

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

Moving on to the application of this theory in our work, we have a surface $\mathcal{M}_i$ (see Fig. 4.2) undergoing an isometric deformation which leads to $\mathcal{M}_j$. A point $\mathbf{z} \in \mathcal{M}_i$ becomes $\mathbf{w} \in \mathcal{M}_j$. Since ψ_{ij} is an isometric deformation, the infinitesimal distances around $\mathbf{w}$ will be the same as that of $\mathbf{z}$, as both of them represent the same point on different isometric surfaces. This is analogous to equations (A.1) and (A.2).

Appendix B

Christoffel Symbols

In the previous section we saw that the metric tensor in a Cartesian coordinate system (equation (A.1)) is written as an identity matrix, but this may not be true for the metric tensors in other coordinate systems (equation (A.2)). In such coordinate systems, since the metric tensor keeps on changing with the coordinates, we can define its change using CS. Therefore, CS is a set of numbers which are the components of vectors that represent the change in metric tensor. They are defined as the CS of first kind ($\mathbf{\Gamma}_{cab}$) and the CS of second kind ($\mathbf{\Gamma}^{d}_{ab}$) related by the expression $\mathbf{\Gamma}_{cab} = \mathbf{g}_{cd}\mathbf{\Gamma}^{d}_{ab}$. Therefore it is very easy to recover the CS of one kind given the other ones. In our work, we found that the expressions of the CS of second kind were simpler and therefore, we use only these. In the Cartesian coordinate system, the metric tensor is the identity and therefore, all the CS of second kind are zero. However, in the spherical coordinate system, the metric tensor is variable and the CS of second kind are:

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

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

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

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

$$\partial_r \mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2r & 0 \\ 0 & 0 & 2r\sin^2\theta \end{pmatrix} \quad \partial_\theta \mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & r^2\sin 2\theta \end{pmatrix} \partial_\phi \mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{B.3}$$

Given $\mathbf{g}^{pl} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^{-2} & 0 \\ 0 & 0 & r^{-2}\sin^{-2}\theta \end{pmatrix}$ and $\mathbf{g}_{lm,n}$ (obtained in equation (B.3)), we obtain the CS of the spherical coordinate system given in equation (B.1) using equation (B.2). For example, $\mathbf{\Gamma}^{r}_{\theta\theta}$, according to equation (B.2) is given by

$$\begin{aligned} \mathbf{\Gamma}^{r}_{\theta\theta} &= \frac{1}{2}\mathbf{g}^{rr}\left(\mathbf{g}_{r\theta,\theta} + \mathbf{g}_{r\theta,\theta} - \mathbf{g}_{\theta\theta,r}\right) + \frac{1}{2}\mathbf{g}^{r\theta}\left(\mathbf{g}_{\theta\theta,\theta} + \mathbf{g}_{\theta\theta,\theta} - \mathbf{g}_{\theta\theta,\theta}\right) + \frac{1}{2}\mathbf{g}^{r\phi}\left(\mathbf{g}_{\phi\theta,\theta} + \mathbf{g}_{\phi\theta,\theta} - \mathbf{g}_{\theta\theta,\phi}\right) \\ &= \frac{1}{2}\left(0 + 0 - 2r\right) + 0 + 0 = -r. \end{aligned} \tag{B.4}$$

Appendix C

Differential K-forms

A differential K-form represents a smooth section on the infinitesimal tangent space of the manifold. For example, a 0-form describes a point on the manifold, a 1-form describes a line element, a 2-form describes an area element, a 3-form describes a volume element, and so on.

In differential geometry, differential forms [Cartan, 1970; O'Neill, 2006] are used to perform the multivariate calculus independently of the coordinates. A scalar function f, parametrised with m variables $(x^1, x^2, \dots x^m)$ such that a point $\mathbf{f} = f(x^1, x^2, \dots x^m)$ is a 0-form. In this case, the exterior derivative of f is the same as the total derivative of f.

Now, the differential 1-form expressing the exterior derivative of f is given by

$$\mathrm{d}\mathbf{f} = \sum_{t=1}^{m} \frac{\partial f}{\partial x^t} \mathrm{d}x^t, \tag{C.1}$$

where $\mathrm{d}x^t$ are the 1-forms and $\dfrac{\partial f}{\partial x^t}$ represents a linear function on the tangent space of the function f in $\mathbb{R}^n$.

Differential forms are defined locally, in terms of the local coordinates. Hence they are easily transferable from one coordinate system to another and therefore, very useful for defining local properties of the surfaces.

Bibliography

A. Agudo, L. Agapito, B. Calvo, and J. M. M. Montiel. Good Vibrations: A Modal Analysis Approach for Sequential Non-Rigid Structure from Motion. In *CVPR*, 2014.

A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. Sequential Non-Rigid Structure from Motion using Physical Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):979–994, 2016.

I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid Structure from Motion in Trajectory Space. In *NIPS*, 2009.

A. Bartoli and E. Özgür. A Perspective on Non-Isometric Shape-from-Template. In *ISMAR*, 2016.

A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine Low-rank Structure-from-Motion. In *CVPR*, 2008.

A. Bartoli, Y. Gérard, F. Chadebecq, and T. Collins. On Template-based Reconstruction from a Single View: Analytical Solutions and Proofs of Well-posedness for Developable, Isometric and Conformal surfaces. In *CVPR*, 2012.

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

F. L. Bookstein. Principal Warps: Thin-plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):567–585, 1989.

C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-rigid 3D Shape from Image Streams. In *CVPR*, 2000.

T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *ECCV*, 2004.

F. Brunet, A. Bartoli, and R. Hartley. Monocular Template-based 3D Surface Reconstruction: Convex Inextensible and Nonconvex Isometric Methods. *Computer Vision and Image Understanding*, 125:138–154, 2014.

Carl and D. Boor. *A Practical Guide to Splines*, volume 27. Springer-Verlag New York, 1978.

É. Cartan. Sur les variétés à connexion affine, et la théorie de la relativité généralisée (première partie). *Annales Scientifiques de l'École Normale Supérieure*, 40:325–412, 1923.

É. Cartan. Sur les varietes a connexion projective. *Bulletin de la Société Mathématique*, 52: 205–241, 1924.

É. Cartan. Espaces à connexion affine, projective et conforme. *Acta Mathematica*, 48:1–42, 1926.

É. Cartan. *La théorie des groupes finis et continus et la géométrie différentielle traitées par la méthode du repère mobile*. Paris: Gauthier-Villars, 1937. ISBN 978-0-12-088735-4.

É. Cartan. *Differential Forms. (Translated from French)*. Hermann, 1970. ISBN 978-0-12-088735-4.

A. Chhatkuli, D. Pizarro, and A. Bartoli. Non-Rigid Shape-from-Motion for Isometric Surfaces using Infinitesimal Planarity. In *BMVC*, 2014.

A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. Inextensible Non-Rigid Shape-from-Motion by Second-Order Cone Programming. In *CVPR*, 2016a.

A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):833–850, 2016b.

Y. Choe and R. L. Kashyap. 3-D Shape from a Shaded and Textural Surface Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):907–919, 1991.

T. Collins and A. Bartoli. Locally Affine and Planar Deformable Surface Reconstruction from Video. In *International Workshop on Vision, Modeling and Visualization*, 2010.

T. Collins and A. Bartoli. Realtime Shape-from-Template: System and Applications. In *ISMAR*, 2015.

Y. Dai, H. Li, and M. He. A Simple Prior-free Method for Non-rigid Structure-from-Motion Factorization. In *CVPR*, 2012.

A. Del Bue. A Factorization Approach to Structure from Motion with Shape Priors. In *CVPR*, 2008.

A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid Structure from Motion using Non-Parametric Tracking and Non-linear Optimization. In *CVPRW*, 2004.

H. Delingette. Toward Realistic Soft-tissue Modeling in Medical Simulation. *Proceedings of the IEEE*, 86(3):512–523, 1998.

P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, Inc., 1993. ISBN 0-19-853441-8.

R. Fabbri. *Multiview Differential Geometry in Application to Computer Vision*. Ph.D. dissertation, Division Of Engineering, Brown University, Providence, RI, 02912, July 2010.

R. Fabbri and B. B. Kimia. Multiview Differential Geometry of Curves. *International Journal of Computer Vision*, 120(3):324–346, Dec 2016. ISSN 1573-1405.

J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In *ECCV*, 2010.

M. Gallardo, T. Collins, and A. Bartoli. Using Shading and a 3D Template to Reconstruct Complex Surface Deformations. In *BMVC*, 2016.

M. Gallardo, T. Collins, and A. Bartoli. Dense Non-Rigid Structure-from-Motion and Shading with Unknown Albedos. In *ICCV*, 2017.

R. Garg, A. Roussos, and L. Agapito. Dense Variational Reconstruction of Non-rigid Surfaces from Monocular Video. In *CVPR*, 2013a.

R. Garg, A. Roussos, and L. Agapito. A Variational Approach to Video Registration with Subspace Constraints. *International Journal of Computer Vision*, 104(3):286–314, 2013b.

P. Gotardo and A. Martinez. Kernel Non-Rigid Structure from Motion. In *ICCV*, 2011.

N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, M. O. Berger, and S. Cotin. Image-guided Simulation of Heterogeneous Tissue Deformation for Augmented Reality during Hepatic Surgery. In *ISMAR*, 2013.

N. Haouchine, J. Dequidt, M. O. Berger, and S. Cotin. Single View Augmentation of 3D Elastic Objects. In *ISMAR*, 2014.

N. Haouchine, F. Roy, L. Untereiner, and S. Cotin. Using Contours as Boundary Conditions for Elastic Registration During Minimally Invasive Hepatic Surgery. In *IROS*, 2016.

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

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

B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices. *Journal of the Optical Society of America A*, 5 (7):1127–1135, July 1988.

M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-Time Volumetric Non-rigid Reconstruction. In *ECCV*, 2016.

A. Kock. *Synthetic Geometry of Manifolds.* Cambridge University Press, 2010. ISBN 978-0-521-11673-2.

B. Koo, E. Özgür, B. Le Roy, E. Buc, and A. Bartoli. Deformable Registration of a Pre-operative 3D Liver Volume to a Laparoscopy Image using Contour and Shading Cues. In *MICCAI*, 2017.

J. Lee. *Riemannian Manifolds : An Introduction to Curvature*. Springer, 1997. ISBN 0-387-98322-8.

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

C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense Correspondence Across Scenes and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5): 978–994, May 2011.

Q. Liu-Yin, R. Yu, L. Agapito, A. Fitzgibbon, and C. Russell. Better Together: Joint Reasoning for Non-rigid 3D Reconstruction with Specularities and Shading. In *BMVC*, 2016.

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

L. Maier-Hein, A. Groch, A. Bartoli, S. Bodenstedt, G. Boissonnat, P.-L. Chang, N. Clancy, D. S. Elson, S. Haase, E. Heim, et al. Comparative Validation of Single-shot Optical Techniques for Laparoscopic 3-D Surface Reconstruction. *IEEE Transactions on Medical Imaging*, 33(10):1913–1930, 2014.

E. Malis and M. Vargas. Deeper understanding of the Homography Decomposition for Vision-based Control. Technical Report RR-6303, INRIA, 2007.

A. Malti, A. Bartoli, and T. Collins. A Pixel-Based Approach to Template-Based Monocular 3D Reconstruction of Deformable Surfaces. In *ICCVW*, 2011.

A. Malti, R. Hartley, A. Bartoli, and J.-H. Kim. Monocular Template-Based 3D Reconstruction of Extensible Surfaces with Local Linear Elasticity. In *CVPR*, 2013.

F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D Stretchable Surfaces from Single Images in Closed Form. In *CVPR*, 2009.

T. D. Ngo, S. Park, A. Jorstad, A. Crivellaro, C. D. Yoo, and P. Fua. Dense Image Registration and Deformable Surface Reconstruction in Presence of Occlusions and Minimal Texture. In *ICCV*, 2015.

T. D. Ngo, J. Östlund, and P. Fua. Template-based Monocular 3D Shape Recovery using Laplacian Meshes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (1):172–187, 2016.

S. I. Olsen and A. Bartoli. Implicit Non-Rigid Structure-from-Motion with Priors. *Journal of Mathematical Imaging and Vision*, 31(2-3):233–244, 2008.

B. O'Neill. *Elementary Differential Geometry*. Elsevier, 2006. ISBN 978-0-12-088735-4.

J. Östlund, A. Varol, T. D. Ngo, and P. Fua. Laplacian Meshes for Monocular 3D Shape Recovery. In *ECCV*, 2012.

J. Östlund, T. D. Ngo, and P. Fua. Monocular 3D Shape Recovery using Laplacian Meshes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

M. Oswald, E. Töppe, and D. Cremers. Fast and Globally Optimal Single View Reconstruction of Curved Objects. In *CVPR*, 2012.

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

S. Parashar, D. Pizarro, A. Bartoli, and T. Collins. As-rigid-as-possible Volumetric Shape-from-Template. In *ICCV*, 2015.

S. Parashar, D. Pizarro, and A. Bartoli. Isometric Non-Rigid Shape-From-Motion in Linear Time. In *CVPR*, 2016.

M. Perriollat, R. Hartley, and A. Bartoli. Monocular Template-based Reconstruction of Inextensible Surfaces. *International Journal of Computer Vision*, 95(2):124–137, 2011.

Photoscan. Agisoft Photoscan 1.0.4, 2014. URL http://www.agisoft.ru/products/photoscan.

L. Piegl. On NURBS: A Survey. *IEEE Computer Graphics and Applications*, 11:55–71, 1991.

J. Pilet, V. Lepetit, and P. Fua. Real-Time Non-Rigid Surface Detection. In *CVPR*, 2005.

D. Pizarro and A. Bartoli. Feature-based Deformable Surface Detection with Self-occlusion Reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

D. Pizarro, R. Khan, and A. Bartoli. Schwarps: Locally Projective Image Warps Based on 2D Schwarzian Derivatives. *International Journal of Computer Vision*, 119(2):93–109, 2016.

M. Prasad, A. Zisserman, and A. Fitzgibbon. Single View Reconstruction of Curved Surfaces. In *CVPR*, 2006.

T. Rassias and P. Šemrl. On the Mazur-Ulam Theorem and the Aleksandrov Problem for Unit Distance Preserving Mappings. *Proceedings of the American Mathematical Society*, 118:919–925, 1993.

C. Russell, R. Yu, and L. Agapito. Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes. In *ECCV*, 2014.

M. Salzmann and P. Fua. Linear Local Models for Monocular Reconstruction of Deformable Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.

M. Salzmann, V. Lepetit, and P. Fua. Deformable Surface Tracking Ambiguities. In *CVPR*, 2007.

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

A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. ABF++: Fast and Robust Angle Based Flattening. In *SIGGRAPH*, 2005.

L. Smith, D. Nevins, N. T. Dat, and P. Fua. Measuring the Accuracy of Softball Impact Simulations. *Sports Engineering*, 19(4):265–272, 2016.

N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, 80(2):189–210, 2007.

O. Sorkine and M. Alexa. As-rigid-as-possible Surface Modeling. In *Symposium on Geometry Processing*, 2007.

N. Sundaram, T. Brox, and K. Keutzer. Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. In *ECCV*, 2010.

L. Tao and B. J. Matuszewski. Non-rigid Structure from Motion with Diffusion Maps Prior. In *CVPR*, 2013.

J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid Structure from Locally-rigid Motion. In *CVPR*, 2010.

L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. Tracking and Modeling Non-Rigid Objects with Rank Constraints. In *CVPR*, 2001.

L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid Structure-from-Motion: Estimating Shape and Motion with Hierarchical Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008.

A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free Monocular Reconstruction of Deformable Surfaces. In *ICCV*, 2009.

A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *CVPR*, 2012a.

A. Varol, A. Shaji, M. Salzmann, and P. Fua. Monocular 3D Reconstruction of Locally Textured Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (6):1118–1130, 2012b.

S. Vicente and L. Agapito. Soft Inextensibility Constraints for Template-Free Non-rigid Reconstruction. In *ECCV*, 2012.

S. Vicente and L. Agapito. Balloon Shapes: Reconstructing and Deforming Objects with Volume from Images. In *3DV*, 2013.

X. Wang, M. Salzmann, F. Wang, and J. Zhao. Template-Free 3D Reconstruction of Poorly-Textured Nonrigid Surfaces. In *ECCV*, 2016.

P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large Displacement Optical Flow with Deep Matching. In *ICCV*, 2013.

R. White and D. A. Forsyth. Combining Cues: Shape from Shading and Texture. In *CVPR*, 2006.

R. Yu, C. Russell, N. D. F. Campbell, and L. Agapito. Direct, Dense, and Deformable: Template-Based Non-rigid 3D Reconstruction from RGB Video. In *ICCV*, 2015.

S. Zhang, A. Nealen, and D. Metaxas. Skeleton Based As-Rigid-As-Possible Volume Modeling. In *Eurographics*, 2010.

K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H. Shum. Large Mesh Deformation using the Volumetric Graph Laplacian. *ACM Transactions on Graphics*, 24:496–503, 2005.

M. Zollhöfer, E. Sert, G. Greiner, and J. Süßmuth. GPU based ARAP Deformation using Volumetric Lattices. In *Eurographics*, 2012.