UNIVERSITÉ BLAISE PASCAL
Année

UNIVERSITÉ D'AUVERGNE
N° d'ordre

ECOLE DOCTORALE
DES SCIENCES POUR L'INGENIEUR

THÈSE

Présentée à l'Université d'Auvergne
Pour obtenir le diplôme de **Docteur**

Specialité
COMPUTER VISION

Soutenue par

Ajad CHHATKULI

le 2 décembre 2016

Local Analytic and Global Convex Methods for the 3D Reconstruction of Isometric Deformable Surfaces

Jury

Président et Examinateur	David FOFI
Rapporteurs	Lourdes AGAPITO
	Mathieu SALZMANN
Directeur de thèse	Adrien BARTOLI
Co-encadrant	Daniel PIZARRO

Laboratoire ISIT-ALCoV
Faculté de Médécine, Université de Auvergne
28 place Henri Dunant, Clermont-Ferrand
Tel: +33 4 73 17 81 23

À mes parents et ma sœur.

Acknowledgements

I would like to say my sincere thanks to my supervisors Adrien Bartoli and Daniel Pizarro for helping and inspiring me through my PhD. Not only have I learned a great deal but it has also been an amazing experience working with them in our small and closely knit group. I am very thankful to the inspiration Adrien has provided me for research. His guidance and support have been instrumental to the success of my PhD. I have also valued highly his friendship, and the conversations in and out of the lab regarding work or other aspects. I am very grateful to Dani for his support and encouragement during my PhD. I have always admired Dani for his kindness, humour and his ingenuity in research. He has been a friend as well as a supervisor to me. I would also like to mention my thanks to Toby whose knowledge and understanding have been so helpful to many of us in the lab. One important thing I have been fortunate with during my stay is the incredible friendship and kindness I received from my lab mates, and also my friends I met in Clermont-Ferrand and during my Master's studies.

My PhD was funded by the EU's FP7 through the ERC research grant 307483 FLEXABLE. The funding provided me most importantly a comfortable life in France. It also gave me the opportunity to visit and learn from several conferences, to visit the University of Alcalá, many internal seminars and retreats, summer school, etc. I am very grateful to Natalia and Julia for helping me through the countless number of administrative papers required for my stay and day-to-day life in France.

In addition, I would like to express my gratitude to the thesis jury reporters, Lourdes Agapito and Mathieu Salzmann 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 David Fofi for his role of the President of the jury and his valuable comments.

Lastly, I would like to thank my family whose love and encouragement led me to my PhD and its successful completion.

Abstract

This thesis contributes to the problem of 3D reconstruction for deformable surfaces using a single camera. In order to model surface deformation, we use the isometric prior because many real object deformations are near-isometric. Isometry implies that the surface cannot stretch or compress. We tackle two different problems.

The first is called Shape-from-Template where the object's deformed shape is computed from a single image and a texture-mapped 3D template of the object surface. Previous methods propose a differential model of the problem and compute the local analytic solutions. In the methods the solution related to the depth-gradient is discarded and only the depth solution is used. We demonstrate that the depth solution lacks stability as the projection geometry tends to affine. We provide alternative methods based on the local analytic solutions of first-order quantities, such as the depth-gradient or surface normals. Our methods are stable in all projection geometries.

The second type of problem, called Non-Rigid Shape-from-Motion is the more general template-free reconstruction scenario. In this case one obtains the object's shapes from a set of images where it appears deformed. We contribute to this problem for both local and global solutions using the perspective camera. In the local or point-wise method, we solve for the surface normal at each point assuming infinitesimal planarity of the surface. We then compute the surface by integration. In the global method we find a convex relaxation of the problem. This is based on relaxing isometry to inextensibility and maximizing the surface's average depth. This solution combines all constraints into a single convex optimization program to compute depth and works for a sparse point representation of the surface.

We detail the extensive experiments that were used to demonstrate the effectiveness of each of the proposed methods. The experiments show that our local template-free solution performs better than most of the previous methods. Our local template-based method and our global template-free method performs better than the state-of-the-art methods with robustness to correspondence noise. In particular, we are able to reconstruct difficult, non-smooth and articulating deformations with the latter; while with the former we can accurately reconstruct large deformations with images taken at very long focal lengths.

Résumé

Cette thèse contribue au problème de la reconstruction 3D pour les surfaces déformables avec une seule caméra. Afin de modéliser la déformation de la surface, nous considérons l'isométrie puisque de nombreuses déformations d'objets réels sont quasi-isométriques. L'isométrie implique que, lors de sa déformation, la surface ne peut pas être étirée ou compressée. Nous étudions deux problèmes.

Le premier est le problème basé sur une modèle 3D de référence et une seule image. L'état de l'art propose une méthode locale et analytique de calcul direct de profondeur sous l'hypothèse d'isométrie. Dans cette méthode, la solution pour le gradient de la profondeur n'est pas utilisée. Nous prouvons que cette méthode s'avère instable lorsque la géométrie de la caméra tend à être affine. Nous fournissons des méthodes alternatives basées sur les solutions analytiques locales des quantités de premier ordre, telles que les gradients de profondeur ou les normales de la surface. Nos méthodes sont stables dans toutes les géométries de projection.

Dans le deuxième type de problème de reconstruction sans modèle 3D de référence, on obtient les formes de l'objet à partir d'un ensemble d'images où il apparaît déformé. Nous fournissons des solutions locales et globales basées sur le modéle de la caméra perspective. Dans la méthode locale ou par point, nous résolvons pour la normale de la surface en chaque point en supposant que la surface est infinitésimalement plane. Nous calculons ensuite la surface par intégration. Dans la méthode globale, nous trouvons une relaxation convexe du problème. Celle-ci est basée sur la relaxation de l'isométrie en constrainte d'inextensibilité et sur la maximisation de la profondeur en chaque point de la surface. Cette solution combine toutes les contraintes en un seul programme d'optimisation convexe qui calcule la profondeur et utilise une représentation éparse de la surface.

Nous détaillons les expériences approfondies qui ont été réalisées pour démontrer l'efficacité de chacune des méthodes. Les expériences montrent que notre solution libre de modèle de réference local fonctionne mieux que la plupart des méthodes précédentes. Notre méthode local avec un modèle 3D de réference et notre méthode globale sans modèle 3D apportent de meilleurs résultats que les méthodes de l'état de l'art en etant robuste au bruit de la correspondance. En particulier, nous sommes en mesure de reconstruire des déformations complexes, non-lisses et d'articulations avec la seconde méthode; alors qu'avec la première, nous pouvons reconstruire avec précision de déformations larges à partir d'images prises avec des très longues focales.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Computer Vision is a multidisciplinary branch of computer science, mathematics and others, whose ultimate goal is to make computers see and understand the world around us. One of the fundamental steps for this ultimate goal is making computers see in 3D. Although we humans effortlessly see the world in 3D, this is far from trivial for a computer. The level of details, speed and robustness our eyes and brain show in attaining this task, looks mind-bogglingly futuristic for computer vision. This thesis concerns with a small but an important aspect of the aforementioned fundamental step: finding a method that can make a computer see non-rigid objects in 3D. In the jargon of computer vision, making a computer see in 3D is referred to as 3D reconstruction. We present our contributions towards reconstructing 3D of deformable or non-rigid surfaces using a single camera. To put the task in perspective, we first discuss the established and widely used methods in 3D reconstruction and the challenges of non-rigid 3D reconstruction. We shall use the term *non-rigid* interchangeably with *deformable* to describe objects or methods of reconstruction of deformable surfaces.

Structure-from-Motion and stereo. The 3D reconstruction of rigid objects from their multiple images is one of the cornerstones of computer vision. The solution to this problem consists of a sequence of steps known as Structure-from-Motion (SfM). An SfM pipeline is shown in figure 1.1. The inputs are images and the intrinsics of the camera. With these inputs SfM simultaneously gives the reconstructed 3D of the object and the camera poses relative to a fixed coordinate frame. SfM consists of two major steps: *feature point matching* and *reconstruction*. Feature point matching gives the point matches between the images via methods such as Scale Invariant Feature Transform (SIFT) matching followed by a geometric verification. There is still a lot of room for improvement in feature point matching but for most cases, current state-of-the-art methods have been proven to be effective. The reconstruction step of SfM (Hartley and Zisserman, 2004) can be considered to be largely solved. A significant number of commercial software and hardware products such as *Photoscan* have been developed that use SfM to compute the 3D of a scene.

The reason SfM works so well is because it considers the scene to be rigid and the relative motion

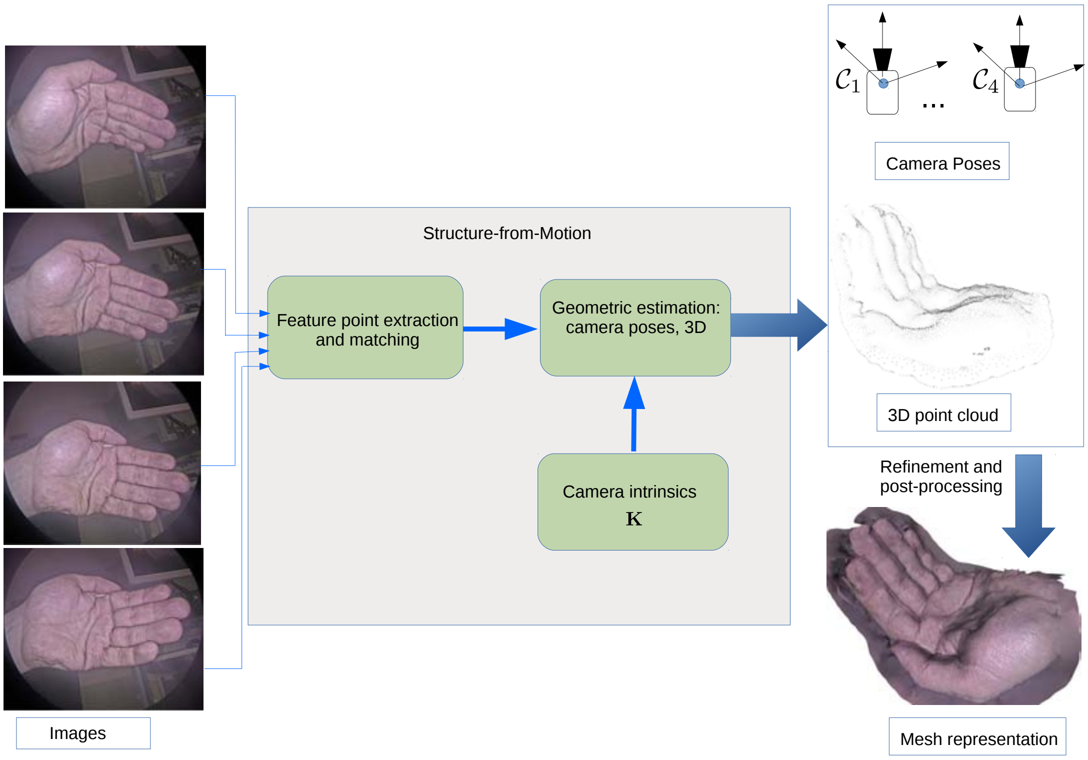

Images
Structure-from-Motion
Feature point extraction and matching
Geometric estimation: camera poses, 3D
Camera intrinsics
K
C_1
...
C_4
Camera Poses
3D point cloud
Refinement and post-processing
Mesh representation

Figure 1.1: Rigid 3D reconstruction and SfM.

between the scene objects to be zero. Rigidity is a strong prior. It allows the scene on each image to be related to that of any other image in each camera's coordinate reference, by using only six parameters: three for rotation and three for translation. The six degrees of freedom are fixed irrespective of the number of scene points we want to reconstruct.

Depth sensors. There are several types of depth sensors that have been developed to capture the object's depth at each point. Stereo vision uses the same principles as SfM to triangulate depth using two or multiple cameras at the same instant. One interesting modification of stereo is the structured light sensor. A structured light sensor projects one or many patterns on the object surface and matches the original pattern with the projected one densely. It then uses these point matches to triangulate depth in the same way as the stereo. The KINECT sensor is a structured light sensor and it obtains the dense matches using a single pattern. The most recent version of KINECT, called the KINECT One uses a time-of-flight (TOF) sensor. TOF sensors obtain 3D by measuring the phase changes due to the time delay between the emitted and reflected light pulses. Other technologies for measuring 3D use lasers to obtain a very high resolution and accurate 3D of objects. Except for KINECT and stereo, almost all 3D sensing technologies work only on static scenes. The other important problem with 3D sensors is that they always tend to be bulky and expensive. Despite the rapid evolution of such sensors, it is quite possible that we may not have a depth sensor as small as miniature cameras in the near future.

Non-rigidity. When we consider deformable surfaces, the rigidity prior no longer holds true. As a consequence, the change of 3D scenes across the images cannot be described by only 6 parameters. Thus SfM no longer yields the correct 3D of the deforming scene. We can draw analogy from the rigid 3D reconstruction and describe the non-rigid 3D reconstruction problem by figure 1.2.

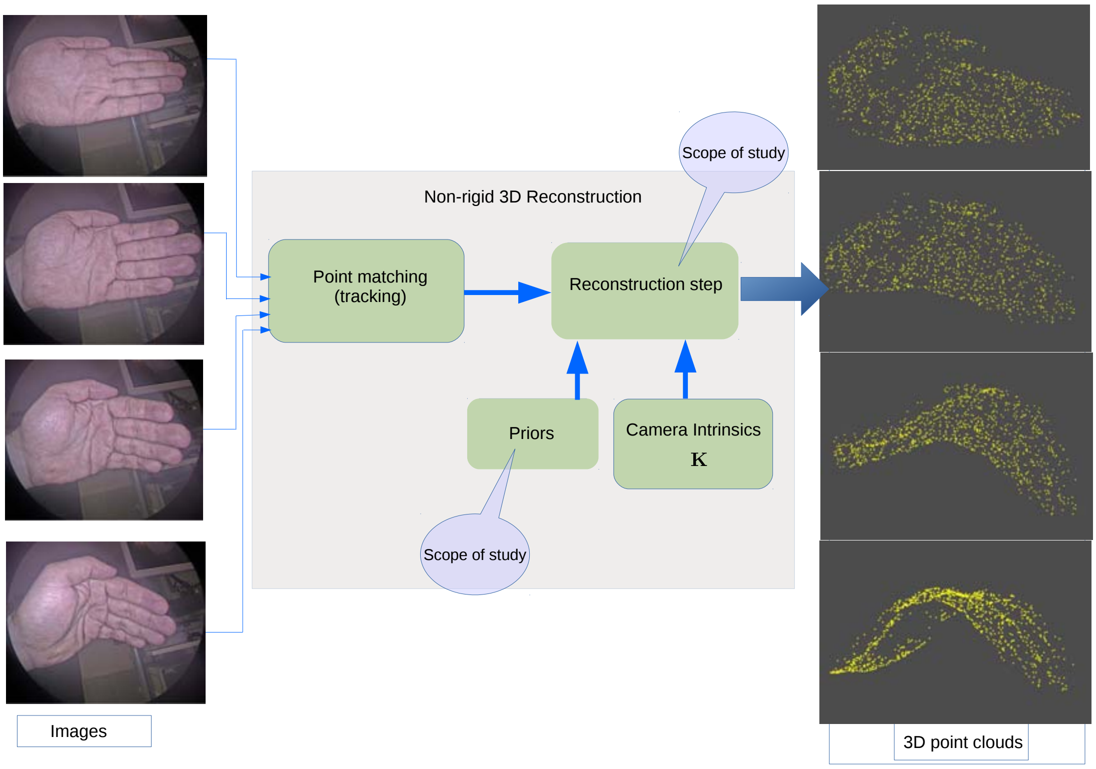


Figure 1.2: Problem formulation for 3D reconstruction of non-rigid scenes.

Fundamentally, the major bottleneck here is not in establishing point correspondences between images. Instead it is in the theoretical framework to compute the 3D surfaces from matched image points. For this reason, 3D reconstruction of deforming objects with a single camera is still a very open problem and a subject of great interest. Addressing non-rigidity in the problem of 3D reconstruction is important because much of the scenes that are of practical importance such as the human body, organs, cloths are non-rigid. Being able to reconstruct such deforming scenes would open a lot of applications, particularly in augmentation for entertainment or medical procedures. Some applications of monocular 3D reconstruction of deforming scenes have already been explored in medical endoscopic augmented reality (Collins and Bartoli, 2015; Maier-Hein et al., 2014). The difficulty in monocular non-rigid reconstruction stems from the fact that it is a severely under-constrained problem without additional priors on the surface deformation or shape space. This is due to the fact that a wide spectrum of deformations of an object can yield the same image projections. Furthermore, it is not yet quite clear which priors work better for typical deforming surfaces.

In this thesis, we use the isometric prior exclusively because it is a physical prior like rigidity but being much weaker it can also model many real deformations. In an isometric deformation, the surface deforms such that the geodesic distance between any pair of points remains unchanged. In other words, the surface does not stretch or compress anywhere in isometric deformation. The notion of isometry is well established in the field of differential geometry where a number of theorems and properties have been developed. In the real world, object deformations may not be exactly isometric but most of them show a near-isometric property. Examples include paper like surfaces, cloths, human pose or body parts, some human organs, etc. Even though isometry cannot relate surfaces with a few transformation parameters as in rigidity, we show in the thesis that with the right formulation of isometry and camera geometry, we can obtain accurate reconstructions of deformable surfaces. There are two important problems in deformable 3D reconstruction, one where a textured 3D template is known and a single input image of the deformed surface is given and the other when only the images are given. We explore both problems in the thesis. We introduce and describe these two problems below.

1.2 Shape-from-Template

Context. In Shape-from-Template (SfT) (Bartoli and Collins, 2013; Bartoli et al., 2015; Ngo et al., 2016; Perriollat et al., 2011; Salzmann and Fua, 2011a), the 3D shape is obtained from a single image and a template of the object. The template is a textured 3D model of the object in a known reference position. Figure 1.3 gives the inputs and outputs of the SfT problem. SfT is a considerably easier

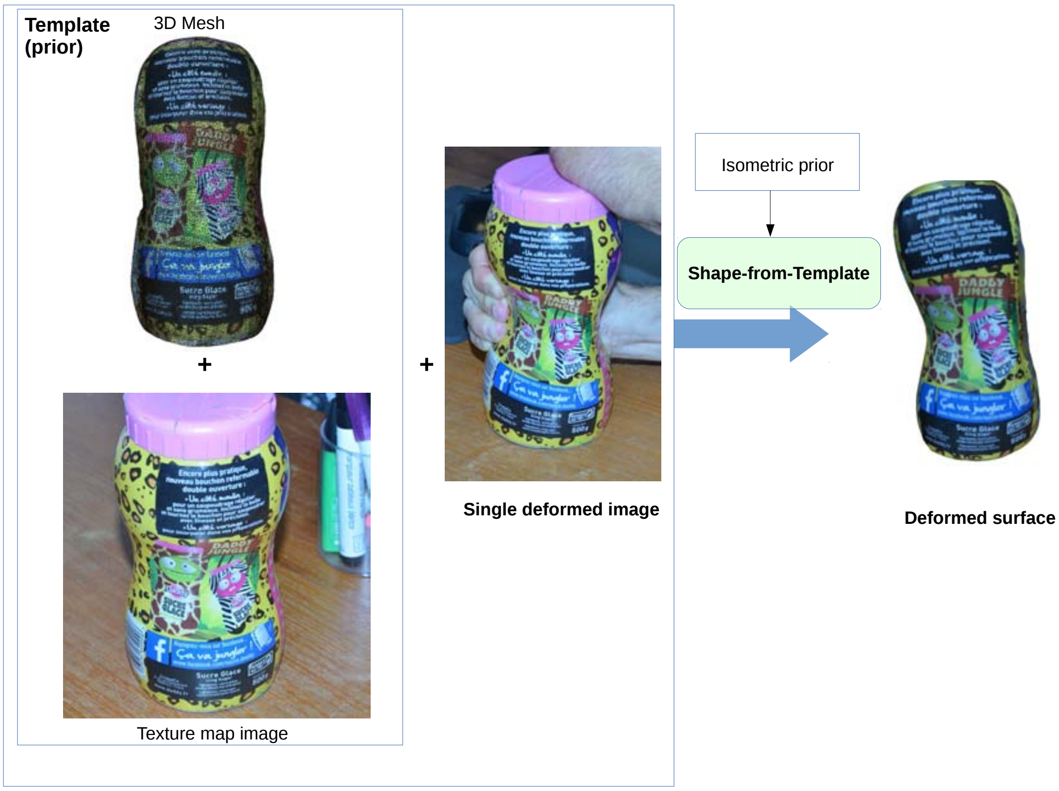


Figure 1.3: The SfT problem. In this example, we describe the template as a mesh but in general, any texture-mapped 3D model can be used.

problem among the two deformable reconstruction problems. Remarkable reconstructions using only point matches were shown in (Salzmann and Fua, 2011a) and its solvability using the first-order and zeroth-order registration terms was studied in (Bartoli et al., 2015). It was shown in the latter that the isometric model provides locally as many constraints as the rigid planar model. Here we study the shape inference step of SfT, assuming that registration between the template and image is solved using, for example, point correspondences (Collins and Bartoli, 2014b; Pilet et al., 2008; Pizarro and Bartoli, 2012). In particular we explore the shape reconstruction when the projection geometry becomes close to affine. A scene's projection geometry can range from strongly perspective to virtually affine. In the latter case, this happens when the object is very small or viewed from a large distance. We define a *stable method* as one that gives an accurate 3D reconstruction for all projection geometries. We propose two stable methods that require no initialization and are fast.

Several successful SfT methods that have been proposed are based on the isometric constraint. In essence, these methods differ from each other by the way isometry is imposed and how the final constraints are optimized. (Bartoli and Collins, 2013; Bartoli et al., 2015) describe isometric SfT as a Partial Differential Equation (PDE) system, giving local analytical solutions and a proof that isometric SfT is well-posed. (Brunet et al., 2014) finds the shape by minimizing a statistically optimal cost while imposing isometric constraints. (Collins and Bartoli, 2015) propose a real-time tracking-based approach for the perspective cameras. A different class of methods (Ngo et al., 2016; Perriollat et al., 2011; Salzmann and Fua, 2011a) relaxes isometry with inextensibility. Using the so-called *Maximum Depth Heuristic* (MDH), (Perriollat et al., 2011; Salzmann and Fua, 2011a) choose the shape that maximizes depth under the inextensibility constraint. Inextensibility means the distance between the neighboring points remain inferior to their geodesic distance in the template. Recently (Ngo et al., 2016) modified the approach of (Salzmann and Fua, 2011a) by enforcing Laplacian smoothness in the solution. The methods discussed form a solid foundation for the SfT problem. However they lack accuracy and applicability in many scenarios. We claim that the following qualities are desired of an SfT method: *a)* it should be robust to noise and outliers in correspondences, *b)* it should be accurate even with a low number of matched feature points, *c)* it should work with camera focal lengths that are very small to very large (stability) and *d)* it should be fast and analytical with no requirement for an initialization. The correspondence outliers mentioned in the property *a)* can be either tackled at the registration step or the reconstruction step. There are several successful methods that remove outliers during registration (Collins and Bartoli, 2014b; Pilet et al., 2008; Pizarro and Bartoli, 2012) while small noise in correspondences has to be dealt with in the reconstruction step. For that reason, we focus our further discussions of property *a)* in the context of noise in correspondences for the reconstruction step. The inextensibility-based methods (Ngo et al., 2016; Perriollat et al., 2011; Salzmann and Fua, 2011a) fail to capture the property in *c)* and do not entirely satisfy *d)*, while the analytical solutions in (Bartoli and Collins, 2013; Bartoli et al., 2015) fail to capture the requirements in *c).* Similarly the statistically optimal cost minimization in (Brunet et al., 2014; Collins and Bartoli, 2015) does not satisfy *d).* Thus there is a clear need for a method that satisfies all four criteria.

Local analytical methods. The local analytical solutions for SfT were first given in (Bartoli et al., 2015), where a set of non-holonomic solutions was obtained for a PDE system with a change of variable. The solutions of a PDE system are called the non-holonomic solutions when they are obtained by treating any quantities and their derivatives as separate independent unknowns. Specifically, the non-holonomic solutions give the radial component of the depth and its gradient in the spherical coordinate system. Despite the fact that the local analytical depth solution is unique, we show that *it is not well-constrained when the camera projection tends to affine*. On the other hand, we prove that the depth-gradient solution is always stable. This is a significant discovery because most methods for SfT rely on computing a solution for the depth (Bartoli and Collins, 2013; Bartoli et al., 2015; Collins et al., 2014; Ngo et al., 2016; Salzmann and Fua, 2011a). We further give an alternative approach to isometric SfT, which gives an equivalent but different PDE system, inspired from the work of (Collins and Bartoli, 2014a) on plane-based pose estimation. In (Collins and Bartoli, 2014a) the pose of a rigid plane is estimated using the non-holonomic solutions of a PDE system. While (Collins and Bartoli, 2014a) does not give solutions for a deforming surface, we adapt the PDE system for SfT for both planar and non-planar templates. The resulting PDE system is equivalent to the PDE system proposed in (Bartoli et al., 2015) but its non-holonomic solutions describe different quantities. In particular its second non-holonomic solution is used differently in the subsequent steps of reconstruction that yields slightly different results. It also gives a more intuitive description of SfT and its solutions in terms of rigid transforms of tangent planes on the surface, and surface normals.

Stable solutions and stable methods We define a *stable solution* as a non-holonomic solution of the SfT PDE system that remains well-constrained regardless of the projection geometry involved (i.e., perspective or affine). We define a *stable method* as one which solves SfT accurately for all projection geometries. We propose two stable methods that satisfy properties *a)* to *d)* described previously. We achieve this by using the stable solutions based on two first-order quantities: the depth-gradient and surface normal. In our first method, which we refer to as the *stable type-I method*, we specifically use the *radial depth-gradient solution*. We obtain this from the PDE system of (Bartoli et al., 2015). However, the radial depth gradient is only known up to sign. We resolve the sign from the depth solution, and then integrate the quantity over the surface to obtain the radial depth values. Because this is from the integration, the values are up to a global scale factor. We compute the scale factor from the average of the depth solution. Finally we obtain the depth values by a change of variable. Our second method, the *stable type-II method* is similar, however we use non-holonomic solution for the surface normal. Like the radial depth gradient, the surface normal also has a two-fold ambiguity. We resolve this again by using the depth solution. Finally we integrate the normals, then resolve the reconstruction's scale using the average of the depth solution. In practice, the results of the two stable methods differ slightly due to the influence of noise on the subsequent steps. We find the stable type-II method to be slightly superior to the stable type-I. Both of the proposed methods rival the accuracy of statistically optimal approaches (Brunet et al., 2014; Collins and Bartoli, 2015), are stable under any projection geometry and require no initialization. Our proposed stable methods were first described in our article (Chhatkuli et al., 2016b). We describe our proposed solution to SfT in chapter 4.

1.3 Non-Rigid Shape-from-Motion

Non-Rigid Shape-from-Motion (NRSfM) is the problem of finding the 3D shape of a deforming object given a set of monocular images. Unlike in SfT, we do not have a 3D template but a number of images of the deforming surface and is therefore, a much harder problem. We again refer the reader to figure 1.2, which can be considered as a general description of the NRSfM problem. This problem is naturally under-constrained because there can be many different deformations and shapes that produce the same images. Consequently, we again need priors to disambiguate the correct shapes. Depending on the type of priors used, different modelling approaches can be considered in order to solve NRSfM. Several methods have been proposed in the last decade to tackle NRSfM with a variety of deformation priors. There are two main categories of methods based on the deformation priors: statistics-based (Bregler et al., 2000; Dai et al., 2012; Garg et al., 2013a; Gotardo and Martínez, 2011; Torresani et al., 2008) and physical model-based (Agudo and Moreno-Noguer, 2015; Chhatkuli et al., 2014b; Taylor et al., 2010; Varol et al., 2009; Vicente and Agapito, 2012) methods. In the former group one assumes that the space of deformations is low-dimensional. These methods can recover deformations such as body gestures, facial expressions and simple smooth deformations. However they tend to perform poorly for objects with high-dimensional deformation spaces or atypical deformations. They can also be difficult to use when there is missing data due to *e.g.* occlusions. In the latter group one finds deformation models based on isometry (Taylor et al., 2010; Varol et al., 2009; Vicente and Agapito, 2012), elasticity (Agudo et al., 2014) or particle-interaction models (Agudo and Moreno-Noguer, 2015). As in SfT, the isometric model is especially interesting and is an accurate model for a great variety of real objects. However in NRSfM, approaches based on isometry still lack in several aspects. For example solutions tend to be complex and often require very good initialization. We propose two solutions based on isometry.

In our first work on NRSfM, we give a new formulation of isometric NRSfM as a nonlinear system of first-order Partial Differential Equations (PDE) involving the shapes' normal and depth functions. Our formulation includes an unknown template, and has SfT as a special case. Second, we show that independent solutions for depth and normal in our system of PDEs are underconstrained. This is an important result since it tells us that NRSfM cannot be solved locally using only first-order PDEs, in contrast to SfT (Bartoli et al., 2012). Finally we provide a solution to NRSfM using an assumption of infinitesimal planarity on top of isometry. Infinitesimal planarity implies the surfaces to be locally planar so that the curvature can be safely ignored. This allows us to model the relation between the projected points with a *differential homography*. By decomposing the homography, we obtain the possible solutions of the surface normals. We then provide an algorithm to disambiguate the normals using a number of views (greater than 2). Finally we obtain the shapes by integrating the normals. This work was first proposed in (Chhatkuli et al., 2014b).

Our second and final work is based on the solution of NRSfM using a convex relaxation of isometry and can be considered to be the state of the art in NRSfM. Here, we use the inextensibility constraint for approximating isometry. Inextensibility is a relaxation of isometry where one assumes that the Euclidean distances between points on the surface do not exceed their geodesic distances. Inextensibility alone is insufficient because the reconstruction can arbitrarily shrink to the camera's center. In template-based reconstruction inextensibility has been combined with the so-called Maximum-Depth Heuristic (MDH), where one maximizes the average depth of the surface subject to inextensibility constraints. This approach has been successfully applied in (Salzmann and Fua, 2011a), providing very accurate results for isometrically deforming objects. The main feature of MDH in template-based scenarios is that it can be efficiently solved with convex optimization. However, in NRSfM, the template is unknown and thus MDH cannot be used out-of-the-box. Our main contribution is to show how to solve NRSfM using MDH for isometric deformations. The problem is solved globally with convex optimization, and handles perspective projection and difficult cases such as non-smooth objects and/or deformations and large amounts of missing data (*e.g.* 50% or more due to self-occlusions). Furthermore, our solution is far easier to implement than all state-of-the-art methods and has only one hyperparameter. It can be implemented in MATLAB using only 25 lines of code. We also incorporate robustness and temporal smoothness in the original formulation to obtain better results (with robustness) and better speed (with temporal smoothness). This work is for the most part, based on our recent paper (Chhatkuli et al., 2016a). In summary, our global NRSfM solution has the following properties. *1)* a perspective camera model is used (unlike in low-rank models and few others), *2)* the isometry constraint is used, *3)* a global solution is guaranteed with a convex problem and no initialization (unlike in the recent methods which use local energy minimization) *4)* we can handle non-smooth surfaces and do not require temporal continuity *5)* we handle missing correspondences and *6)* the complete set of constraints are tied together in a single problem. We provide extensive experiments where we show that we outperform existing work by a large margin in most cases.

Summary of contributions. The thesis deals with two different problems in deformable surface reconstruction: SfT and NRSfM. We list the summary of the contributions on these problems below.

1. In the template-based reconstruction scenario of SfT, we prove that the depth solutions are unconstrained as the projection geometry tends to affine. We then give a different method for obtaining the shape based on the integration of a first-order quantity such as depth-gradient or surface normal (Chhatkuli et al., 2016b).

2. We give a local solution for surface normals using the isometric prior and thus obtain the unknown shapes in the template-less reconstruction problem of NRSfM (Chhatkuli et al., 2014b).

3. We provide the first physical prior-based convex formulation of NRSfM using the inextensibility prior (Chhatkuli et al., 2016a).

Thesis layout. We divide the thesis into 7 chapters. Chapter 2 discusses the prerequisites, such as the two-view geometry, registration, surface modelling and optimization. These concepts are used as mathematical tools in the chapters that follow and therefore we do not give their detailed explanations. We discuss the state of the art and the related concepts specific to non-rigid 3D reconstruction in chapter 3. We give our solutions and contributions to the SfT problem in chapter 4. In chapter 5 we discuss our finding of solvability of NRSfM using zeroth and first-order registration quantities and present our own point-wise (local) solution. We present our global formulation for convex NRSfM using the inextensibility prior in chapter 6. Finally we conclude and give our perspective for future work on the presented problems in chapter 7.

Chapter 2

Modelling and Mathematical Tools

The solution to the non-rigid 3D reconstruction problem begins with the basics of camera geometry and a modelling of the surface deformation. In this chapter we describe these prerequisites which will help us progress smoothly to the actual solutions of SfT and NRSfM. We begin by recalling the camera geometry and image registration. They can be considered as the basics which are ubiquitous in various problems of computer vision. Then we briefly discuss surface representation and surface deformation priors. The deformation priors mandate a brief description of the basic concepts from differential geometry. The final part of the chapter gives an overview of some optimization techniques. This is in no way a detailed description of the mathematical optimization methods but simply a brief outline of some concepts required for the understanding of the following chapters. An understanding on the subject matter discussed here is required to fully comprehend the previous work or our own proposed work.

Mathematical notations

For better readability, we define a set of rules to denote mathematical quantities throughout the thesis. In several cases, however, we slightly part from these rules. We make a note on such exceptions for each type of quantities. Additionally, despite the rules, often, some ambiguities will be made clear with the help of the context in which the notation is used. When discussing about standard mathematical tools and objects, we let the standard notations in the field supersede our own notations to avoid any misunderstanding. Such notations are made clear at the point when they appear.

Scalars. We write all scalar quantities as Latin letters in italics, e.g., s. *Exceptions:* We use λ_i to denote the ith eigenvalue of a matrix, which is a scalar. We assume the eigenvalues to be in descending order: $\lambda_i \geq \lambda_j$ with $i < j$. We therefore reserve the symbol λ exclusively for eigenvalues of a matrix. Similarly, we use σ_i for the ith singular value of a matrix.

Vectors and matrices. We use lowercase Latin letters in bold for vectors, e.g., $\mathbf{x}$. Its components which are scalar quantities, are written as the vector name in italics followed by a numeral or Latin subscript. For example, the components of a vector $\mathbf{x}$ of dimension n could be written as $x_1, x_2, \ldots, x_n$. We represent matrices as uppercase Latin letters in bold, e.g. $\mathbf{M}$. The ith eigenvalue of a matrix $\mathbf{M}$, as mentioned before, is represented by λ_i whereas, the ith eigenvector is represented by $\mathbf{v}_i(\mathbf{M})$. For a matrix $\mathbf{M}$ of size $m \times n$, we denote its sub-matrix made of the first i rows and j columns as $[\mathbf{M}]_{ij}$. We write the jth column of a matrix $\mathbf{M}$ as $[\mathbf{M}]_j$. Additionally, the identity matrix of dimension n is written as $\mathbf{I}_n$. *Exceptions:* We write the vector quantities representing a small or differential change as δ and ϵ. Because of widely accepted notations, we denote the metric tensor as g even though it is a matrix. We often denote 3D point vectors on surfaces as $\mathbf{Q}$, i.e., in upper case bold letter even if they are not matrices.

Functions. All functions, whether scalar, vector or matrix-valued are represented using Greek letters of lower or upper case, e.g., $\psi(\mathbf{x})$. For readability we will often drop the function argument and write the same function as ψ. We use the operator J_ψ to write the function giving the Jacobian matrix of ψ.

Operators. We use $\mathrm{diag}(s_1, \ldots, s_n)$ to define a diagonal matrix from the scalars $s_1, \ldots, s_n$. We use the operator J_ψ to write the function giving the Jacobian matrix of ψ. We often drop the variable of the function in such cases. We write $\phi \in C^d(\mathbb{R}^n, \mathbb{R}^m)$ to denote that the function $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is d times differentiable. We use the '$\min$' or '$\max$' operator to represent the minimum or maximum of a set respectively. We write 'minimize' or 'maximize' for minimizing or maximizing an objective function respectively. Finally '$\arg\min$' and '$\arg\max$' operators represent the optimal value of the variables of a minimization and maximization problem respectively.

Others. We write the lth norm of a vector $\mathbf{p}$ as $||\mathbf{p}||_l$, and when l is not specified as in $||\mathbf{p}||$, it represents the L2 norm of $\mathbf{p}$. Objects such as spaces, surfaces, sets and cameras are denoted in calligraphic math fonts such as $\mathcal{P}$, $\mathcal{S}$ or $\mathcal{C}$. Additionally point sets are also represented by putting the concerned variable in curly braces, e.g., $\{\mathbf{p}\}$ represents the set of all vectors $\mathbf{p}$. We denote the general linear group of dimension n as GL_n and the special group of rotation matrices of order n as SO_n. We write the set of all symmetric matrices of order n as $\mathbb{S}_n$ and the set of all positive real number as $\mathbb{R}^{++}$. In describing NRSfM, we use the subscripts and superscripts to denote the corresponding point index and the image or surface index respectively. Thus $\mathbf{q}_i^k$ may denote the ith image correspondence for the kth image.

2.1 The Camera Model

In computer vision, we reason about the world based on the images obtained from a camera. We first briefly recall how the image formation in cameras can be modelled. Although, modern cameras have a very complex lens system, for most purposes, the imaging can be very accurately modelled using a pinhole camera model. The pinhole camera, also known as the camera obscura is not just a model but an actual camera. The first published description of the camera can be found in an 1856 book by the Scottish inventor David Brewstor. The device is made with a single small hole on a hollow opaque box. Figure 2.1 illustrates a pinhole camera and its preferred mathematical representation. The hole acts as an aperture which lets the light into the camera in a controlled fashion. From each point in an object the hole allows a very narrow bundle of rays to hit the screen or image plane. The narrower the hole is, the sharper will be the image formed. When used as a model, we assume that from each point on the 3D object a single ray of light hits the screen. Consider a 3D point $\mathbf{Q} = \begin{bmatrix} Q_x & Q_y & Q_z \end{bmatrix}^\top \in \mathbb{R}^3$ is projected onto the image plane. If we assume the distance between the camera center and the screen to be 1 unit and the image coordinate axes to be centered at the physical center of the screen, the projected point is given by $\mathbf{q} = \begin{bmatrix} q_u & q_v \end{bmatrix}^\top = \frac{1}{Q_z} \begin{bmatrix} Q_x & Q_y \end{bmatrix}^\top$.

In effect, the pinhole camera projections are formed by perspective projection as shown in the bottom row of figure 2.1. In order to exactly know the projection of a 3D point on the image, we need to know beforehand certain parameters of the camera.

Camera extrinsics. In order to mathematically define the projection of a 3D point onto the image plane, first the 3D point coordinates have to be in the same coordinate system as the camera. The former or the 3D object's coordinate system is called the world coordinate system while the latter is called the camera coordinate system. When these coordinate systems do not align, we first transform the point and its reference axis via a rotation and translation so that the two coordinate systems align and then project the point. These transformation parameters required to align the axes are collectively called the extrinsics of the camera. For each camera, the camera extrinsics will consist of the 6 rigid-transform parameters. In the context of deformable 3D reconstruction with physical priors, we generally assume that the world coordinate system is the same as the camera's for each image. Therefore we only consider the camera coordinate system here and it makes the camera extrinsics irrelevant for our purpose.

Camera intrinsics. By looking at figure 2.1, we can observe that the projection of a fixed 3D point can be different depending on the relative placement of the image plane and the camera center as

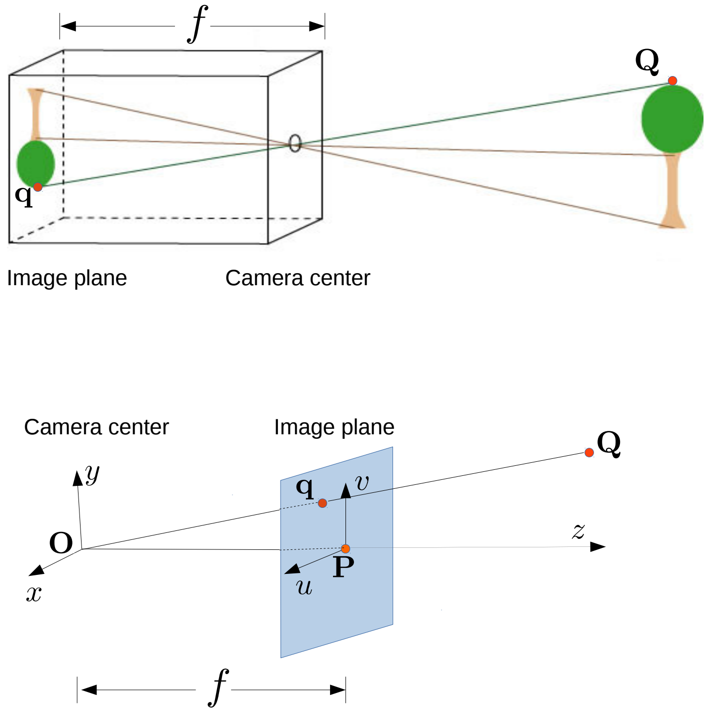


Figure 2.1: The schematic of a simple pinhole camera (top, source: wikipedia) and its mathematical representation (bottom).

well as the shape of the image plane. This relative placement defines 5 different properties that are internal to the camera. Therefore the parameters are termed as the camera intrinsics. They are: the focal length in number of x-direction pixels f_x, the focal length in terms of y-direction pixels f_y, the principal point $\mathbf{p}_c = \begin{bmatrix} c_x & c_y \end{bmatrix}^\top$. The final parameter called the skew s_K is relevant if the image axes on the image plane are not exactly perpendicular. Camera calibration is the process in which we compute these five intrinsic parameters. The camera calibration matrix is a 3×3 matrix of the intrinsic parameters as written below:

$$\mathbf{K} = \begin{bmatrix} f_x & s_K & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.1}$$

Besides these parameters built in the calibration matrix $\mathbf{K}$, there are other parameters that model the distortion of lens. This is of particular importance in wide-angle lens cameras such as the fish-eye camera.

2.2 Camera Projection

There are three important projection models defined in the literature that are relevant to our problems. The perspective projection was briefly introduced when we explained the pinhole camera. Depending on the purpose and the object-camera configuration, there are other projection models which can be useful: the orthographic projection and the weak-perspective projection. We briefly discuss all three of them below.

2.2.1 Perspective camera

The perspective camera is the most commonly used camera model as it accurately describes the imaging projections and is yet simple enough. Leaving out the relatively negligible distortions, our eyes and cameras project the world onto the retina or image sensors by perspective projection. The projection in a pinhole camera is exactly a perspective projection where rays coming from the object converge at a focal point, i.e., the camera center. Effectively, it means that parallel lines can intersect at a point after the projection. Figure 2.2 shows an example of the perspective projection of a cube.

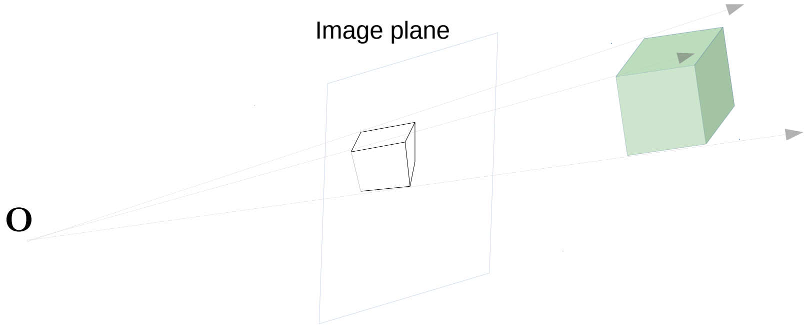


Figure 2.2: Perspective projection of a cube. Lines parallel on the 3D cube are not parallel in the projected image.

Mathematically, a perspective camera is denoted by a 3×4 matrix acting on a 3D point. The camera projects any 3D point existing in a *world coordinate frame* in position $\mathbf{Q} \in \mathbb{R}^3$ to an image point coordinate $\mathbf{q} \in \mathbb{R}^2$.

$$s \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{Q} \\ 1 \end{bmatrix}. \tag{2.2}$$

Equation (2.2) describes the perspective projection in homogeneous coordinates for the 3D and image point. The scalar $s \in \mathbb{R}\backslash\{0\}$ is introduced here because a point in the homogeneous coordinate system remains the same after multiplying with a nonzero scalar. The projection matrix encodes the camera extrinsic and intrinsic parameters. The camera orientation with respect to the world coordinate frame can be defined by its extrinsics: a rotation $\mathbf{R} \in SO_3$ and a translation $\mathbf{t} \in \mathbb{R}^3$. Similarly consider $\mathbf{K} \in \mathbb{R}^{3\times3}$ to be the matrix of the camera intrinsics. The projection matrix can be written as:

$$\mathbf{M} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}. \tag{2.3}$$

When considering non-rigid surfaces, it is often convenient to express the 3D surface points in the camera frame so that the world coordinate system is aligned to the camera frame *i.e.* $\mathbf{R} = \mathbf{I}_3$ and $\mathbf{t} = \mathbf{0}$. The perspective projection in that case can be written as:

$$s \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Q} \\ 1 \end{bmatrix}. \tag{2.4}$$

Pre-multiplying equation (2.4) by $\mathbf{K}^{-1}$ gives us the following expression for the image point:

$$s \begin{bmatrix} \mathbf{q}_n \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Q} \\ 1 \end{bmatrix}. \tag{2.5}$$

Equation (2.5) describes the relation for the *retinal or normalized coordinates* of the image points represented by the vector $\mathbf{q}_n$. It gives the projected image coordinates in the Euclidean camera frame for a given perspective camera. Effectively, the image coordinates are transformed so that the focal length is 1 and the physical center of the image is $\mathbf{p} = [0\ 0\ 1]^\top$. Equation (2.5) can be written in a different way that reveals the depth parametrization of a 3D point as below:

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} = Q_z \begin{bmatrix} \mathbf{q}_n \\ 1 \end{bmatrix}, \tag{2.6}$$

where $\mathbf{Q} = [Q_x\ Q_y\ Q_z]^\top$. Equation (2.6) essentially implies that a 3D point expressed in the camera frame can be parametrized by the depth and the retinal image coordinate. It also reveals that the 3D point $\mathbf{Q}$ lies on the sightline along the direction of the vector $\begin{bmatrix} \mathbf{q}_n \\ 1 \end{bmatrix}$. We will exploit these properties of the perspective camera extensively when we discuss the reconstruction problem.

2.2.2 Orthographic camera

The orthographic camera is an approximation of the perspective camera for the conditions when projection sightlines are close-to-parallel. This happens when the depth is very large (in which case, a large focal length is used). In such cases the projection rays can be approximated to be parallel to the depth (z) axis resulting in a very simple set of projection equations. Figure 2.3 illustrates how the projection rays form an image in orthographic projection. Analytically, the u and v coordinates of the projected point q are given as:

$$\begin{bmatrix} q_u \\ q_v \end{bmatrix} = \begin{bmatrix} Q_x \\ Q_y \end{bmatrix}. \tag{2.7}$$

Although the orthographic camera does not accurately describe a real image formation, it renders certain problems much simpler. One example commonly seen in the literature of non-rigid reconstruction is the projective factorization, as described in chapter 2.6.3. A more general camera model

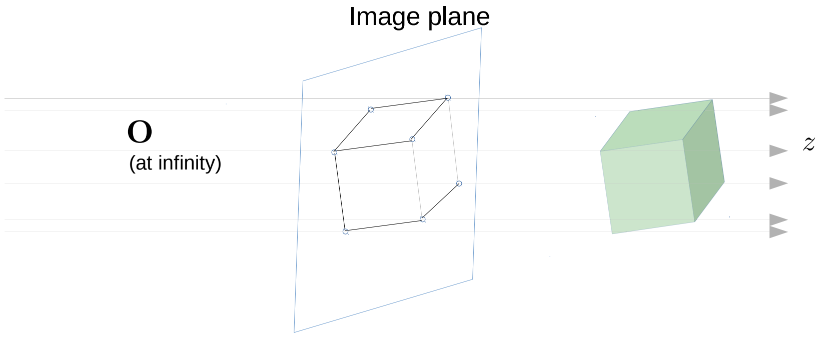


Figure 2.3: Image formation in an orthographic projection. The projection rays are parallel to the depth or z axis.

is the scaled orthographic camera model where an additional scale factor is considered as below:

$$\begin{bmatrix} q_u \\ q_v \end{bmatrix} = s_a \begin{bmatrix} Q_x \\ Q_y \end{bmatrix}. \tag{2.8}$$

2.2.3 Weak-perspective camera

The weak-perspective projection is an example of a hybrid projection. A 3D point is first projected orthographically to a fixed plane that is parallel to the image plane. The points obtained on the plane are then projected to the actual perspective camera from *the average depth* of the object. This is illustrated in figure 2.4. The weak-perspective camera is often used to approximate the perspective projection without completely resorting to an orthographic projection. In particular, the depth scaling is fixed because all the points before perspective projection are at a fixed depth (i.e., the average depth). The projection equation is given by:

$$\begin{bmatrix} q_u \\ q_v \end{bmatrix} = \begin{bmatrix} \frac{Q_x}{Q_z^{av}} & \frac{Q_y}{Q_z^{av}} \end{bmatrix}^\top. \tag{2.9}$$

Here, Q_z^{av} represents the average depth of the object. The weak-perspective camera is essentially a scaled orthographic camera where the scaling quantity s_a in equation (2.8) is fixed to the average depth Q_z^{av}. In most cases, conclusions drawn on the orthographic camera often hold for the weak-perspective camera. Another hybrid camera model related to the weak-perspective camera is the paraperspective camera. Here, an affine projection is used in place of the orthographic projection so that projection rays are parallel to each other but not necessarily to the depth axis.

2.3 Two-view Algebraic Relationship

Multiple images or views are the inputs of NRSfM. We briefly outline three important 2-view epipolar entities for calibrated and uncalibrated cameras. A more thorough discussion of them can be found in any literature related to the multiview geometry such as (Hartley and Zisserman, 2004). We assume

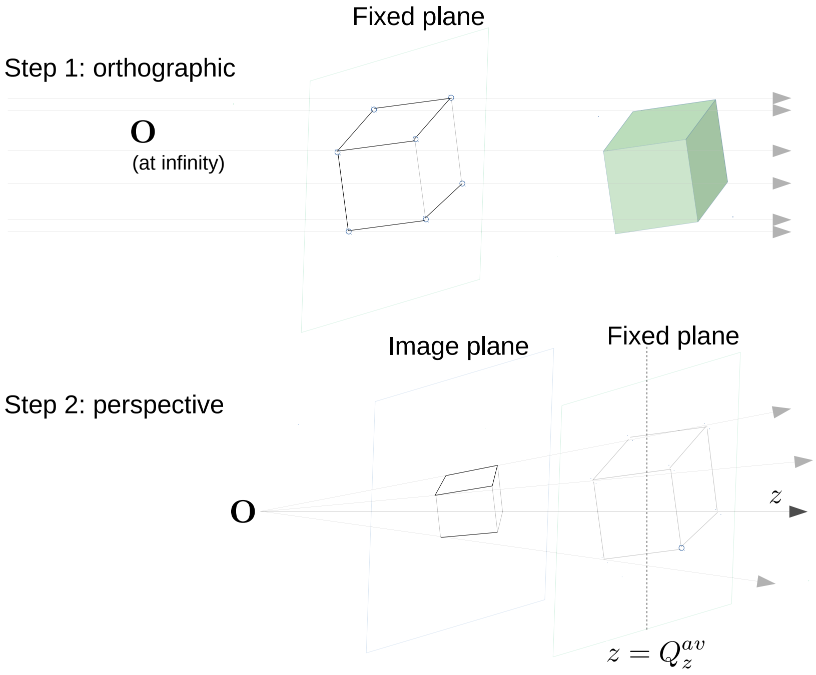


Figure 2.4: Weak-perspective projection of a cube. It involves two steps of projection: first orthographic and second perspective.

the object viewed in images to be rigid in this discussion. A description of how some of these concepts have been used in non-rigid reconstruction methods is given in chapter 3.

2.3.1 Fundamental matrix

Description. For any pair of uncalibrated cameras viewing the same rigid scene, the corresponding points in images constrain each other due to the camera geometry. The resulting constraints does not provide a point-to-point but rather a point-to-line relation. The transformation matrix that describes these constraints is the fundamental matrix. In other words, the fundamental matrix constrains the corresponding point to lie on a specific line called the epipolar line. The fundamental matrix is used for the projective reconstruction of scenes by triangulation. The result can be upgraded to a metric reconstruction by using the intrinsic calibration matrix. Consider a point $\mathbf{q}_1$ on an image of camera $\mathcal{C}_1$ and its corresponding point $\mathbf{q}_2$ on the image of camera $\mathcal{C}_2$. This is illustrated in figure 2.5.

The fundamental matrix $\mathbf{F}$ is a 3×3 rank-2 matrix and it constrains any two corresponding points such that:

$$\begin{bmatrix} \mathbf{q}_1^\top & 1 \end{bmatrix} \mathbf{F} \begin{bmatrix} \mathbf{q}_2 \\ 1 \end{bmatrix} = 0. \tag{2.10}$$

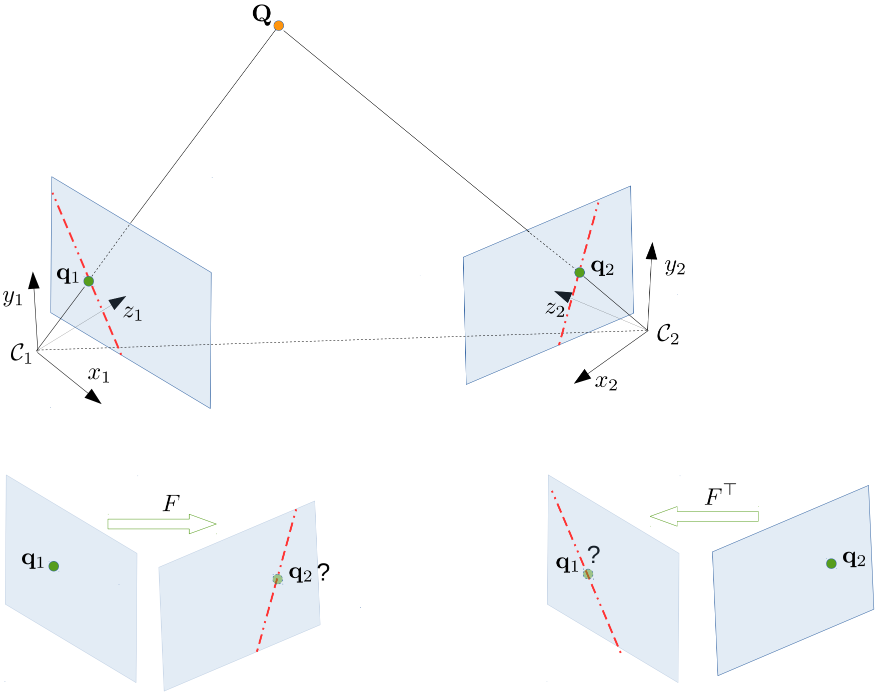


Figure 2.5: Illustration of how a fundamental matrix acts on points. The first row shows the projection of a 3D point onto two images along with the induced epipolar lines. The second row illustrates the effect of the fundamental matrix on the corresponding point.

Computation. The fundamental matrix has 7 degrees of freedom in 9 parameters. The scale ambiguity in equation (2.10) means that one of the 9 parameters can be fixed arbitrarily while the rank-2 constraint removes yet another degree of freedom. The most common way to compute a fundamental matrix between two images (cameras) using rigid scenes is to linearize the matrix and use the so-called 8 point method (Hartley, 1997). The 8-point method uses at least 8 pairs of corresponding points to compute $\mathbf{F}$ by linear least-squares (LLS). Since, all 8 points must come from the same rigid scene, using it for non-rigid reconstruction has some difficulties.

2.3.2 Essential matrix

Description. The essential matrix describes the same relation as the fundamental matrix but for corresponding point pairs in the retinal coordinates. Recall that the retinal image points can be obtained by normalization with the camera calibration matrix $\mathbf{K}$ as:

$$s \begin{bmatrix} \mathbf{q}_n \\ 1 \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix}. \tag{2.11}$$

Computation and decomposition. Unlike the fundamental matrix, the essential matrix has only 5 degrees of freedom. The first two singular values of the essential matrix are equal while the third is 0. In a highly influential paper, (Nistér, 2004) presented a method for the computation of the essential matrix using only 5 points. It can also be expressed in terms of the rotation and translation of the rigid scenes as:

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times. \tag{2.12}$$

Here $[\mathbf{t}]_\times$ denotes the skew symmetric matrix used in cross products, defined in terms of the components of the translation vector $[t_1\ t_2\ t_3]^\top$ as:

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}.$$

Due to this relation, computing the essential matrix allows the estimation of the relative camera positions. However, there still does not exist a non-rigid reconstruction method that exploits the Essential matrix structure.

2.3.3 Homography

The homography is a general linear transform of dimension 3×3. In particular, any 3D plane induces a homography between its projections on images. Such a homography is called a plane induced homography. All the projected points from the plane therefore share the same homography. Unlike the fundamental or the essential matrix, the homography is a point to point transform. In general, it is full rank and has 8 degrees of freedom after removing one for the scale. Given a homography $\mathbf{H}$ between the corresponding pair of points $\mathbf{q}_1$ and $\mathbf{q}_2$, we write:

$$\mathbf{H}\begin{bmatrix}\mathbf{q}_1^\top & 1\end{bmatrix} = s_h\begin{bmatrix}\mathbf{q}_2 \\ 1\end{bmatrix}. \tag{2.13}$$

A plane induced homography $\mathbf{H}$ can be expressed as the following:

$$\mathbf{H} = s_h\mathbf{K}(\mathbf{R} + \mathbf{t}\mathbf{n}^\top)\mathbf{K}^{-1}, \tag{2.14}$$

where, $\mathbf{K}$ is the matrix of camera intrinsics, $\mathbf{R}$ the relative rotation between the corresponding 3D points (planes), $\mathbf{t}$ the relative translation and $\mathbf{n}$ the surface normal on the plane in the first camera reference axis. s_h is the scale factor due to the scale ambiguity of the metric space.

Computation and decomposition. Equation (2.13) gives two constraints for each point pair. Consequently the computation of a homography requires at least 4 corresponding point pairs projected from the same 3D plane. The homography is also directly related to the surface normal of the 3D plane. It can be decomposed into the surface normal and a rigid transform describing the relative transformation of the plane with respect to the camera coordinate frame. When the corresponding points in the images are expressed in the retinal coordinates, the homography between corresponding points on a plane can be written as:

$$\hat{\mathbf{H}} = \mathbf{R} + \mathbf{t}\mathbf{n}^\top. \tag{2.15}$$

Homography decomposition refers to the computation of the corresponding right hand side quantities in equation (2.15) of a given homography. There are methods for the decomposition of $\hat{\mathbf{H}}$ using the Singular Value Decomposition (SVD) (Faugeras and Lustman, 1988; Zhang and Hanson, 1996) or with closed-form analytical expression (Malis and Vargas, 2007). However, the decomposition results in the two-fold ambiguity of the quantities, even after removing physically incoherent solutions. We introduce our local method for NRSfM in chapter 5 by computing the homography point-wise and giving a method to disambiguate the normals.

2.4 Registration

One of the fundamental problems of computer vision along with 3D reconstruction is solving the registration between images. For example, the computation of the 2-view geometric entities and consequently the 3D, relies on having corresponding image points. Registration is about establishing mapping between points in two images. We use the term registration loosely to mean any kind of pairing of points between images, whether they are dense or sparse. Thus registration between images could also be represented by a lookup table with entries of correspondences. In other cases, a function could represent the transformation of points from one image to another. The most appropriate representation of registration depends on the kind of reconstruction method that is being used.

2.4.1 Point matching

Wide-baseline point matching implies establishing correspondences between two images using the invariant properties of objects in images. Scale Invariant Feature Transform (SIFT) (Lowe, 2004) is probably the most used method in that respect. SIFT uses orientation of gradients as features to match certain points (keypoints) between images and is a sparse point matching method. However, sparse point matches occur irregularly and can prove insufficient for reconstruction problems, particularly in non-rigid scenarios. There are matching methods that provide dense matches (Liu et al., 2011) or semi-dense matches as in (Weinzaepfel et al., 2013).

2.4.2 Optical flow

Optical flow is the estimation of apparent motion of scene points in consecutive images in a sequence. These methods usually compute a dense flow field over each image, thus giving a dense set of point correspondences. However, they can only use very short base-line images, such as those from a video sequence. Optical flow computation is a difficult non-convex optimization problem. Nonetheless, efficient methods have been developed that use convex relaxations (Brox et al., 2004; Garg et al., 2013b; Sundaram et al., 2010).

2.4.3 Functional modelling of registration

Often it is necessary or advantageous to model the image correspondence or registration with a mathematical function instead of a simple lookup table. The functional representations can be computed directly from the image-intensities, from the matched points or by combining both (Pizarro and Bartoli, 2012). Computation based on the matched points only is generally preferred for the low cost of computation and decent accuracy. There are some important reasons to compute a functional representation for registration:

1. It can give a dense or regular grid of correspondences. This in turn can mean some methods that use point correspondences behave better than with irregular or sparse set of point correspondences.

2. Moreover, the functional represention can be differentiated to compute the first-order or second-order derivatives of the registration. We use these quantities, for example, in the methods we propose in chapters 4 and 5.

3. These representations can be made robust to outliers so that the generated correspondences have very small amount of noise.

There are various methods for spline-based registration. Among them the Thin Plate Spline (TPS) and Bicubic B-Spline (BBS) are the most common. BBS has compact support and is considered to give better registration derivatives (Pizarro et al., 2016). We use BBS to compute a functional representation of the registration whenever necessary (in chapters 4 and 5). We give the basics of the BBS registration below.

Consider a BBS registration function $\omega : \mathbb{R}^2 \to \mathbb{R}^2$ going from the first image to the second image. ω at a given point $\mathbf{p} = [u\ v]^\top \in \mathbb{R}^2$ in the first image is then expressed as a linear combination of the basis functions $\mathbf{l}_\mathbf{p} \in \mathbb{R}^{b\times 1}$ at the point $\mathbf{p}$ and the control points $\mathbf{C} \in \mathbb{R}^{b\times 2}$. The evalution of the registration function ω will give the corresponding point in the second image $\mathbf{q}$ as:

$$\mathbf{q}^\top = \omega(\mathbf{p}, \mathbf{C}) = \mathbf{l}_\mathbf{p}^\top \mathbf{C}. \tag{2.16}$$

Let us modify equation (2.16) so that we evaluate ω on all points in the first image at a time. We represent the n point correspondences on the first image as $\{\mathbf{p}_i\},\ i = 1 \dots n$ and on the second image as $\{\mathbf{q}_i\},\ i = 1 \dots n$. For convenience we write these correspondences in matrix form as:

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^\top \\ \vdots \\ \mathbf{p}_n^\top \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^\top \\ \vdots \\ \mathbf{q}_n^\top \end{bmatrix}. \tag{2.17}$$

The evaluation of ω on all points gives the matrix $\mathbf{Q}$ as:

$$\omega(\mathbf{P}, \mathbf{C}) = \mathbf{L}^\top \mathbf{C} = \mathbf{Q}, \tag{2.18}$$

where $\mathbf{L} = \begin{bmatrix} \mathbf{l}_{\mathbf{p}_1} & \dots & \mathbf{l}_{\mathbf{p}_n} \end{bmatrix} \in \mathbb{R}^{b\times n}$. In a given warp, the vector of basis functions can be computed directly from the points to be evaluated for a uniform cubic B-spline warp. Consequently, only the control points are actual unknowns. Therefore, we pose the computation of the warp function

$\omega_a(\mathbf{C}, \mathbf{L})$ as the estimation of the control points matrix $\mathbf{C}$.

$$\mathbf{C} = \arg\min_{\mathbf{C}} \|\mathbf{L}^\top \mathbf{C} - \mathbf{Q}\|_2. \tag{2.19}$$

Given the matrix $\mathbf{L}$, equation (2.19) is solved by LLS as:

$$\mathbf{C} = \left(\mathbf{L}^\top \mathbf{L}\right)^{-1} \mathbf{L}^\top \mathbf{Q}. \tag{2.20}$$

In practice, we construct the warp by imposing smoothness constraints that regularizes the solution. We ensure smoothness by minimizing its second derivatives. This modifies problem (2.20) into the following one:

$$\mathbf{C} = \arg\min_{\mathbf{C}} \|\mathbf{L}^\top \mathbf{C} - \mathbf{Q}\|_2 + \|\mathbf{B}\mathbf{C}\|_2, \tag{2.21}$$

where $\mathbf{B}$ is a matrix that gives the second derivatives of the BBS warp when multiplied by the control points. $\mathbf{B}$ can be estimated analytically. It is commonly referred to as the bending energy of the BBS warp. Finally, the control points can again be computed with LLS as:

$$\mathbf{C} = \left(\mathbf{L}^\top \mathbf{L} + \mathbf{B}\right)^{-1} \mathbf{L}^\top \mathbf{Q}. \tag{2.22}$$

We refer the reader to (Brunet, 2010; Dierckx, 1993) for the exact expression of the $\mathbf{L}$ and $\mathbf{B}$ matrices. Such a BBS registration function is not robust to mismatches between points. A more robust formulation can be constructed using an L1 M estimator as described in Appendix C or using other M-estimators.

2.5 Surface Representation

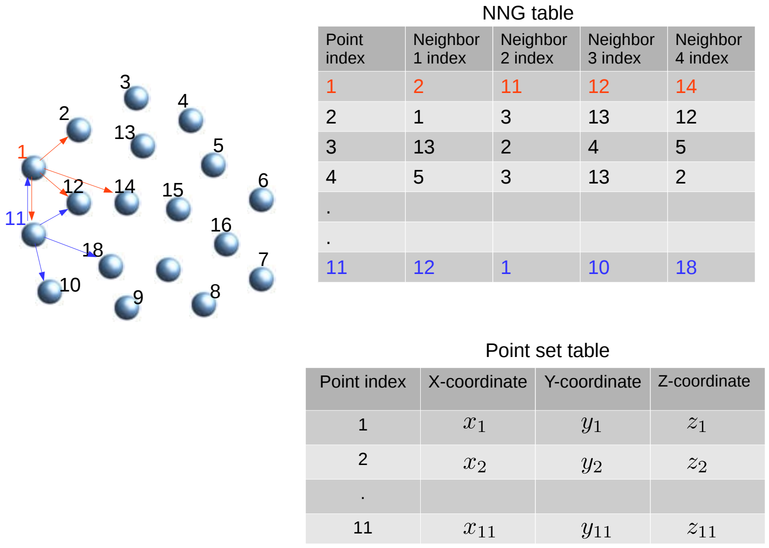
Objects in 3D can be represented in several ways. The type and complexity of the representation partly depends on the task at hand. In nonrigid 3D reconstruction, we will use surfaces as the basic objects in 3D. We here focus on the discussion of surface representation for the purpose of non-rigid 3D reconstruction. Below we list and describe the types of surface representation.

2.5.1 Point set

The point set representation provides the most basic way of describing surfaces. Point sets are synonymous to the more commonly used term point clouds. For deformable objects, the 3D points on the surface at different time are assumed to belong to different surfaces. Consider a surface $\mathcal{S}^k$ at a time instance k. If $\mathbf{Q}_i^k$ is a point on the surface, we write: $\mathcal{S}^k = \{\mathbf{Q}_i^k\}$, $i = 1, \ldots n$, where n is the number of points used to represent the surface. Most of the state-of-the-art methods in NRSfM use this representation.

2.5.2 Point set with nearest neighborhood graph (NNG)

There are many methods in surface 3D reconstruction which represent surfaces with not only points but also their neighborhood relation. By neighborhood relation, we mean the closest neighbors information for each point in terms of the L2 metric distance. Many surface priors depend on having such relations. This representation is closely related to a mesh representation. Mesh consists of a set of vertex points along with edges and faces. The point set with neighborhood on the other hand only defines directional edges, but not faces. We term the neighborhood relationship as Nearest Neighborhood Graph (NNG). The graph is truncated so that each point only has a small finite number of neighbors. However, given a mesh, it is trivial to construct the point set with NNG representation. Figure 2.6 illustrates such a representation with a simple example.



NNG table

Point index	Neighbor 1 index	Neighbor 2 index	Neighbor 3 index	Neighbor 4 index
1	2	11	12	14
2	1	3	13	12
3	13	2	4	5
4	5	3	13	2
.				
.				
11	12	1	10	18

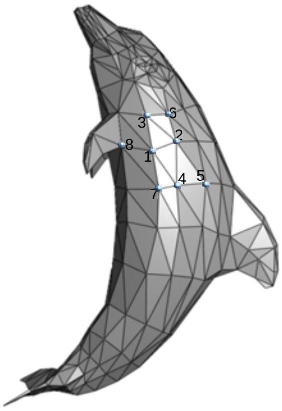
Point set table

Point index	X-coordinate	Y-coordinate	Z-coordinate
1	x_1	y_1	z_1
2	x_2	y_2	z_2
.			
11	x_{11}	y_{11}	z_{11}

Figure 2.6: Surface parametrization with points and their neighborhood. The neighborhood table is truncated to have 4 neighbors per point.

2.5.3 Surface mesh

A mesh represents a surface with vertices, edges and faces. Unlike the edges in the point set with NNG representation, the edges in a mesh are non-directional. Several edges constitute a face which can be a triangle or a quadrilateral. For example, a quadrilateral mesh is composed of vertices, edges and quadrilateral faces (polygons) formed by the edges. Meshes allow easy texturing and rendering of objects and is a highly versatile way to represent surfaces or even volumetric objects. Many numerical methods have been developed to define mathematical operators such as the differentials and geodesic lengths in a mesh. In fact, most of the methods in computer graphics are built around the mesh representation of objects. This is due to the fact that mesh representations are very easy to store and process in a computer. Figure 2.7 shows a simple mesh representation of a dolphin.



Face table

Face index	Vertex 1	Vertex 2	Vertex 3
1	1	2	3
2	4	2	1
3	5	2	4
4	7	4	1
5	1	8	7
6	2	6	3
.	.	.	.

Vertex (Point set) table

Point index	X-coordinate	Y-coordinate	Z-coordinate
1	x_1	y_1	z_1
2	x_2	y_2	z_2
.	.	.	.

Figure 2.7: Surface parametrization with a mesh of a dolphin (Source: Wikipedia).

2.5.4 Differential modelling of surfaces

Although the point set with neighborhood representation is a powerful representation, in many situations a more complete description of a surface is required. Many additional surface properties exist, which require the differential modelling. As the name implies, the differential modelling describes surfaces at the differential or infinitesimal level at any point. This requires some basic understanding of the differential geometry and manifolds. We briefly mention the following three quantities that come from the differential geometry of surfaces.

Global and local embedding. In differential geometry, the physical surface that we see in the world is an intrinsically 2-D object embedded in $\mathbb{R}^3$ and is therefore a 2-D manifold. This implies that the surface can be parametrized on a 2-D flat space as shown in figure 2.8. In the figure, the globe (excluding its poles), which is an object in $\mathbb{R}^3$ is parametrized by the world map in $\mathbb{R}^2$. There exists a one-to-one mapping between the globe (excluding the poles) and the world map due to the parametrization. In the language of manifolds, the function going from the world map to the globe (excluding the poles) is a global embedding. In simple terms, the embedding is a function going from one object $\mathcal{X}$ to another object $\mathcal{Y}$ in different spaces, while preserving some structure. Since a single function is enough to go from the world map to the globe, it is a global embedding. The local embedding on the other hand is a one-to-one mapping valid only for a point and its local or infinitesimal neighborhood. We will make use of both global and local embeddings extensively in chapter 4. In practice we establish the smooth manifold representation using a BBS representation of the embedding function φ from the point set representation similar to image registration described in chapter 2.4. A closer look at equation (2.18) will reveal that it is trivial to extend it for a 2D-to-3D mapping from 2D-to-2D registration.

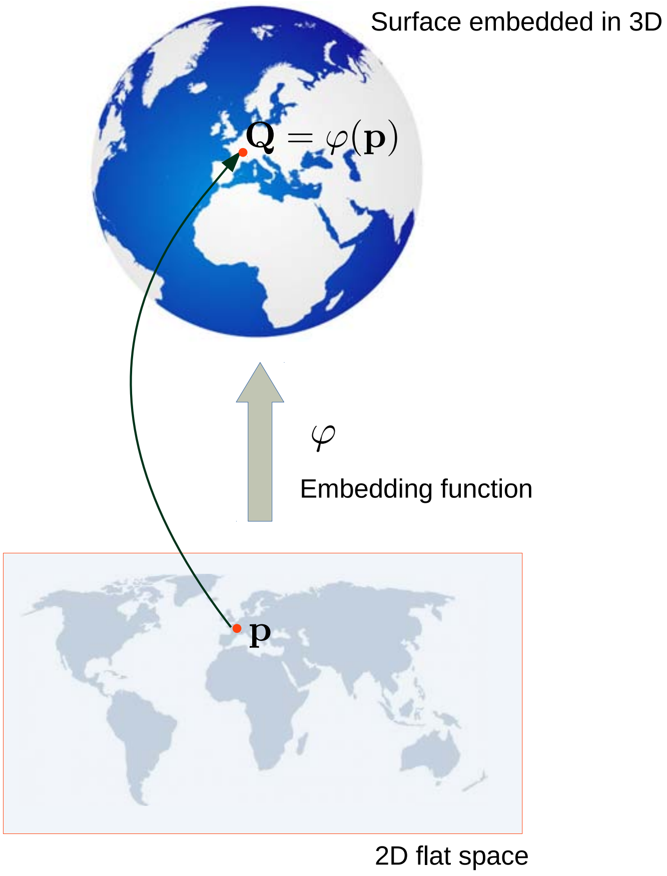


Figure 2.8: Surface parametrization with a flat space. The top represents the globe (excluding poles) which is a spherical surface *embedded in* $\mathbb{R}^3$. It is parametrized with a 2D flat space shown on the bottom.

The tangent space. For any embedded manifold, such as the surface, we can define tangent vectors at a point on the surface that pass through the surface by touching only the given point locally. At each point on the embedding, the equivalence class of these vectors forms a subspace on the embedded space $\mathbb{R}^3$, which is called the tangent space. This is an affine subspace as the affine combination of any two vectors lies on the tangent space as well. For the embedded manifold, this subspace is a plane. The tangent space is always of the same dimension as the intrinsic dimension of the manifold. Figure 2.9 depicts a surface $\mathcal{S}$ and its tangent space at a point $\mathbf{p}$. The tangent space is denoted as $T_{\mathbf{p}}\mathcal{S}$. The tangent vector can also be defined in a different way using a curve on the surface. In that case, the tangent vector at a point on the curve is the rate of change of a curve at the given point.

The first fundamental form. Roughly speaking, the first fundamental form describes the 'rule' to measure inner product on the tangent space of a surface. Consequently, the first fundmental form allows us to measure metric properties such as length or angle on the particular embedding (a surface is a 2-D manifold in $\mathbb{R}^3$) from a parametrization space $\mathcal{P} \in \mathbb{R}^2$. Given two vectors $\mathbf{a}, \mathbf{b} \in \mathcal{P}$, the dot

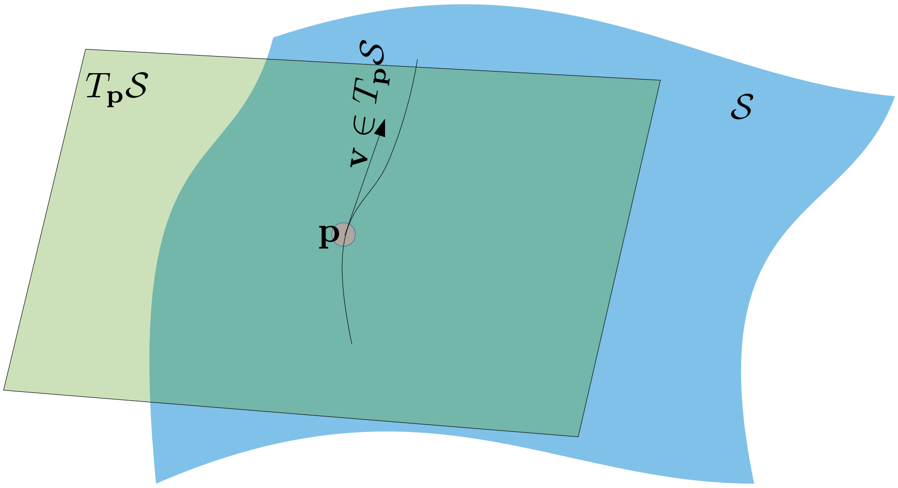

$T_{\mathbf{p}}\mathcal{S}$
$\mathbf{v} \in T_{\mathbf{p}}\mathcal{S}$
$\mathcal{S}$
$\mathbf{p}$

Figure 2.9: A surface $\mathcal{S}$ and its tangent space $T_{\mathbf{p}}\mathcal{S}$ at a point $\mathbf{p}$.

product between them can be measured from $\mathcal{P}$ as:

$$I\left(\mathbf{a},\mathbf{b}\right) = \langle \mathbf{a},\mathbf{b} \rangle = \mathbf{a}^{\top} \begin{bmatrix} E & F \\ F & G \end{bmatrix} \mathbf{b}. \tag{2.23}$$

Here E, F, G and H are scalars. The first fundamental form is closely related to the metric tensor on the parametrization space $\mathcal{P}$. The metric tensor is given by:

$$g = [g_{ij}] = \begin{bmatrix} E & F \\ F & G \end{bmatrix}. \tag{2.24}$$

In order to measure the length of a curve on the surface, consider a parametric curve $u = u(t)$ and $v = v(t)$ on the parameter t where u and v are the coordinates on the surface. We first define an elemental length ds using the first fundamental form as:

$$ds^2 = E\left(u(t), v(t)\right)\, du^2 + 2F\left(u(t), v(t)\right)\, du\, dv + G\left(u(t), v(t)\right)\, dv^2 \tag{2.25}$$

Thus the squared length is given by an integration between the ends of the curve as:

$$s^2 = \int E\left(u(t), v(t)\right)\left(\frac{du}{dt}\right)^2 + 2F\left(u(t), v(t)\right)\frac{du}{dt}\frac{dv}{dt} + G\left(u(t), v(t)\right)\left(\frac{dv}{dt}\right)^2 dt. \tag{2.26}$$

Surface deformation map. We define a surface deformation map or function as a bijective function going from a surface and its deformed version. If the map is differentiable *i.e.* a diffeomorphism, we can describe how infinitesimal lengths on the first surface change in the second. Let us consider a metric tensor $g_{\mathcal{S}_1}$ defined on a surface $\mathcal{S}_1 \in \mathbb{R}^3$. Suppose $\mathcal{S}_1$ is deformed to another surface $\mathcal{S}_2 \in \mathbb{R}^3$ by a deformation map $\psi \in C^2(\mathcal{S}_1, \mathcal{S}_2)$ as shown in figure 2.10.

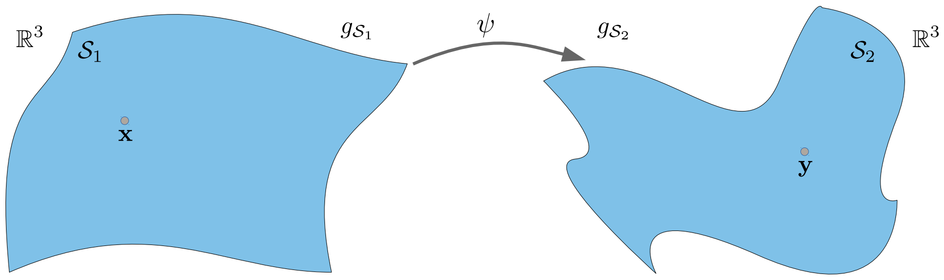


Figure 2.10: Surface deformation and the induced change in metric tensor.

Consider $\mathbf{x}$ to be a point on surface $\mathcal{S}_1$ and $\mathbf{y} = \psi(\mathbf{x})$ the corresponding point on surface $\mathcal{S}_2$. As we are considering ψ to be a diffeomorphism, it induces isomorphism between the two tangent spaces: $T_{\mathbf{x}}\mathcal{S}_1$ and $T_{\mathbf{x}}\mathcal{S}_2$. This simply means that the two tangent spaces are equal up to a transformation. This transformation in differential geometry is referred to as the pushforward $\psi_* g_{\mathcal{S}_1}$. When the surfaces are embedded in fixed coordinates as in our example, we can define the pushforward with the Jacobian of the deformation map ψ as below:

$$\psi_* g_{\mathcal{S}_1} = \mathsf{J}_\psi(\mathbf{x}). \tag{2.27}$$

The pushforward induces a new metric on the tangent space of $\mathcal{S}_1$ that allows us to measure metric quantities on $\mathcal{S}_2$.

$$g_{\mathcal{S}_2} = \mathsf{J}_\psi(\mathbf{x})^\top g_{\mathcal{S}_1} \mathsf{J}_\psi(\mathbf{x}). \tag{2.28}$$

Note that the tensor $g_{\mathcal{S}_2}$ defined as such takes vectors in $T_{\mathbf{x}}\mathcal{S}_1$ to give the inner product of the corresponding vectors in $\mathcal{S}_2$.

2.6 Surface Deformation Priors

Reconstructing a non-rigid object from its image is considered an ill-posed problem because several deformations can result in the same projection even when a template or multiple images are known. To tackle these ambiguities, either the complete space of the observed shapes or the deformations between them have to be limited by additional deformation priors. Such priors are analogous to the rigidity prior for rigid SfM. Methods in deformable 3D reconstruction all solve the problem of the aforementioned ambiguities using the surface deformation priors. We briefly outline a few of the important priors below.

2.6.1 Isometry

Isometry was introduced when we discussed about the first fundamental form in the last section. Essentially, isometry is a physical or geometric prior. In simple terms, it implies that the surface deforms in a way so that all the geodesic distance between points on the surface remain the same. Thus, in isometry, the surface deforms without any stretching or compression in any region. Many natural surfaces deform isometrically and being able to exploit such a prior could potentially solve non-rigid reconstruction for many practical scenarios. Isometry is also a strong geometrical prior, albeit far weaker than rigidity. Note that all rigid objects are also isometric. In fact isometry can be thought of as infinitesimal rigidity. This becomes clearer in chapter 4. We now discuss various ways isometry has been modelled in the literature.

Zeroth-order approximation. It is difficult to accurately represent isometry with zeroth-order representations. The reason is that representing the geodesic distance on an arbitrary unknown surface poses a great challenge. To solve the problem, many methods first establish a neighborhood of points. Then they assume that the euclidean distance between the neighboring points are equal for all deformations of the surface. For example, if $\mathbf{Q}_i^k$ and $\mathbf{Q}_j^k$ are neighboring points on surface k and $\mathbf{Q}_i^l$ and $\mathbf{Q}_j^l$ on surface l, we write the isometric constraint as:

$$\|\mathbf{Q}_i^k - \mathbf{Q}_j^k\|_2 = \|\mathbf{Q}_i^l - \mathbf{Q}_j^l\|_2. \tag{2.29}$$

Equation (2.29) is only Euclidean approximation of the exact geodesic distance and is also a non-convex constraint.

Differential representation. Deformation maps and the metric tensor were introduced in section 2.5. In differential geometry, isometry is defined as a function that preserves the first fundamental form. Although there are various ways to define isometry including the change of metric due to a pushforward in equation (2.28), it is much simpler to define isometry by introducing a parametrization space of the surfaces.

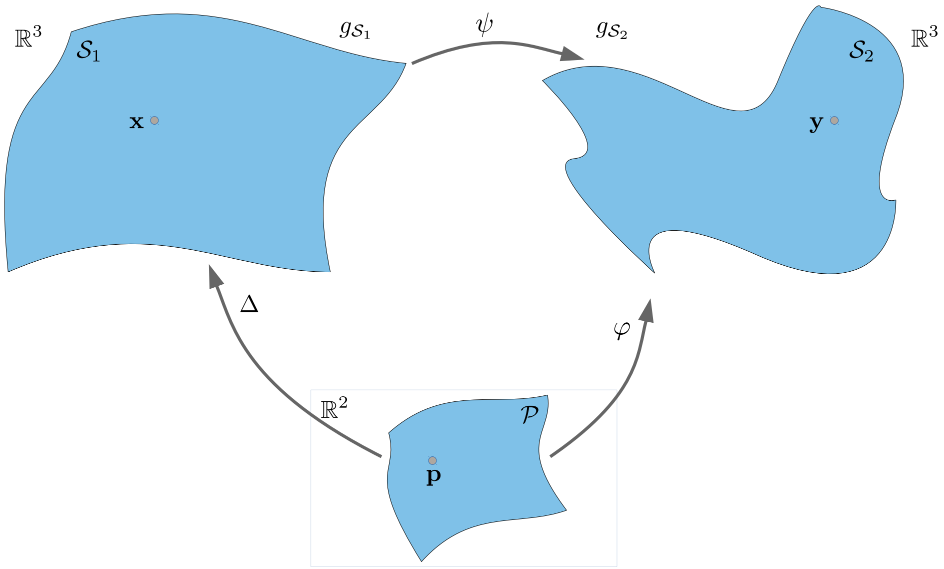


Figure 2.11: Isometry and surface parametrization.

Consider the example shown in figure 2.11, where the surface is parametrized on a set $\mathcal{P} \in \mathbb{R}^2$. $\mathit{\Delta}$ is the embedding function that maps the surface $\mathcal{S}_1$ from the parametrization space $\mathcal{P}$ and φ is the embedding function that maps the surface $\mathcal{S}_2$ from the same parametrization space $\mathcal{P}$. In that case, the first fundamental forms due to the two parametrization functions can be equated in an isometry ψ. This gives us the following equation:

$$\mathsf{J}_\varphi(\mathbf{p})^\top \mathsf{J}_\varphi(\mathbf{p}) = \mathsf{J}_\Delta(\mathbf{p})^\top \mathsf{J}_\Delta(\mathbf{p}), \quad \mathbf{p} \in \mathcal{P}. \tag{2.30}$$

Intuitively, this means that for the surfaces embedded in an Euclidean space, the change in the metric tensor as we move from $\mathcal{P}$ to $\mathcal{S}_1$ should be equal to the change when we move from $\mathcal{P}$ to $\mathcal{S}_2$. This is due to the fact that ψ introduces no change in the metric tensor because it is an isometry between Euclidean spaces.

2.6.2 Inextensibility

Inextensibility is another geometric constraint on surfaces that is closely linked to isometry. It simply means that the Euclidean distances between the neighboring points on the deformed surfaces are always less than or equal to the corresponding geodesic distances on the original surface. It requires a notion of template (original surface) where distances between neighboring points are given by a variable. Let's suppose d_{ij} represents the geodesic distance between point index i and j on the template. $\mathbf{Q}_i^k$ and $\mathbf{Q}_j^k$ are neighboring points on a surface k. We write inextensibility as:

$$||\mathbf{Q}_i^k - \mathbf{Q}_j^k||_2 \leq d_{ij}. \tag{2.31}$$

Equation (2.31) is a relaxation of isometry. If the surfaces are isometric, it holds true for any pair of points, close neighbors or not. Furthermore it is also a convex constraint, more specifically a cone constraint.

2.6.3 Low-rank model

The low rank model is a statistical prior on the set of point matches in several images in the NRSfM setting (Bregler et al., 2000). In other words, it puts a small fixed rank on the matrix of correspondences so that it can be factorized into a matrix of shape bases and coefficients. Consider a set of image point correspondences $\{\mathbf{q}_i^k\}$. The sub-index i denotes the point index and the super-index k denotes the image index, i.e., $\mathbf{q}_i^1, \mathbf{q}_i^2, \ldots, \mathbf{q}_i^m$ are the matched points for the point i. The observation matrix for m images and n points can be written as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{q}_1^1 & \cdots & \mathbf{q}_n^1 \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^m & \cdots & \mathbf{q}_n^m \end{bmatrix} = \begin{bmatrix} \mathbf{R}^1\mathbf{S}^1 \\ \vdots \\ \mathbf{R}^m\mathbf{S}^m \end{bmatrix}. \tag{2.32}$$

$\mathbf{S}^k \in \mathbb{R}^{3\times n}$ is the set of corresponding 3D points for $\left[\mathbf{q}_1^k \ldots \mathbf{q}_n^k\right]$ which forms the surface k. $\mathbf{R}^k$ is the camera projection that projects each 3D point of $\mathbf{S}^k$. Two important assumptions are made to recover $\mathbf{R}$ and $\mathbf{S}$ matrices. The first is the orthographic camera assumption. Orthography means that a matrix

$\mathbf{R}^k$ is orthonormal. The second assumption is written as:

$$\mathbf{S}^k = l_1\mathbf{B}_1 + \cdots + l_l\mathbf{B}_l. \tag{2.33}$$

where $l_1, \ldots, l_l$ are scalar coefficients and $\mathbf{B}_1, \ldots, \mathbf{B}_l \in \mathbb{R}^{3\times n}$ are the shape bases. $l \in \mathbb{R}$ is the number of shape bases. We fix l to an integer ($l \ll m$) and this is the low rank prior. With these assumptions, the observation matrix is represented as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{q}_1^1 & \cdots & \mathbf{q}_n^1 \\ \vdots & \ddots & \vdots \\ \mathbf{q}_1^m & \cdots & \mathbf{q}_n^m \end{bmatrix} = \begin{bmatrix} l_1^1 & \cdots & l_l^1 \\ \vdots & \ddots & \vdots \\ l_1^m & \cdots & l_l^m \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_l \end{bmatrix} = \mathbf{LV}. \tag{2.34}$$

Consequently the NRSfM problem transforms into finding a unique factorization of the known matrix $\mathbf{W}$ into matrices $\mathbf{L}$ and $\mathbf{V}$.

2.6.4 Others

Priors other than the ones mentioned have been used to model deformable surfaces. Two important examples are conformality and linear elasticity. Conformality is closely related to isometry. If two surfaces can be mapped to each other so that the angles between intersecting curves are preserved, we define these surfaces as conformal. It is a theoretical model more general than isometry. Conformality is easily expressed by using the metric tensor as in the case of isometry. Given the example shown in figure 2.11, where ψ is now a conformal deformation rather than isometry, we write:

$$\mathsf{J}_\varphi(\mathbf{p})^\top \mathsf{J}_\varphi(\mathbf{p}) = s_c\, \mathsf{J}_\Delta(\mathbf{p})^\top \mathsf{J}_\Delta(\mathbf{p}), \quad \mathbf{p} \in \mathcal{P}. \tag{2.35}$$

Here s_c is a positive scalar that defines how the lengths are scaled. If there is stretching involved $s_c > 1$ and if there is contraction, $s_c < 1$.

Linear elasticity (Malti et al., 2013) is another physical prior but it is based on how solid objects deform based on forces acting on it. It models the deformation based on the forces acting on it and the object's own tendency to retain its original shape. Such a tendency is called elasticity. Thus linear elasticity implies that the forces or stress leading to deformation or strain have a linear relationship.

2.7 Local Optimization

Most of the solutions discussed in the thesis rely on basics of linear algebra. These mainly involve computation of eigenvalues, linear least-squares and solving a set of linear equations. In some situations however, the solution is obtained from a mathematical optimization. A problem can be posed as a mathematical optimization when analytical solutions are either too difficult to obtain or when they do not exist. This happens to be the case in non-rigid reconstruction as most physical priors result in a highly nonlinear and difficult optimization problem. Nonetheless, the works done in the thesis can be understood by treating optimizations as black-box processes. Understanding the basics of the type of optimization used should be enough to comprehend the consequences on the solution.

An optimization problem consists of minimizing an objective function based on certain constraints. Perhaps the epitome of such problems in computer vision is the Bundle Adjustment (BA). Bundle Adjustment is a highly nonlinear problem where the 3D points and camera extrinsics are refined at the same time from an initial solution. We briefly describe some basic principles and methods used for a local optimization problem and in the next section we introduce a special class of problems called convex optimization. In this context we use the phrase *locally optimal* to mean that the obtained solution is optimal around obtained point but may not be the global minimum. Consequently a local optimization method is one which is used to obtain a locally optimal solution. Note that we are not usually interested in the minimum value of the function, but the value or state of the variable at which we can obtain such a minimum.

Given an optimization variable in a vector space, $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top \in \mathbb{R}^n$, we express an optimization problem as follows:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ \phi(\mathbf{x}) \\ &\text{subject to,} \\ &\theta_i(\mathbf{x}) \leq \mathbf{b}_i, \quad i = 1 \ldots n_c, \end{aligned} \tag{2.36}$$

where, $\phi(\mathbf{x})$ is the objective function to be minimized and $\theta_i(\mathbf{x})$ is any function of $\mathbf{x}$. $\mathbf{b}_i$ represents a constant vector for each constraint and n_c is the number of constraints.

No single method exists that can solve problem (2.36). By solving, we mean finding the global optimum of the given problem.

In solving a general optimization problem, it is a standard practice to start with an initial solution and employ one of many methods that gives a locally optimal minimum. The accuracy or relevance of the obtained solution obtained by many things, one of which is how close the initial solution was to the actual global minimum. We discuss one important case of problem (2.36) when the objective function is a square of a vector-valued function. Such an objective function is often referred to as an energy function. We also assume the problem to have no constraints. Such a problem is simply expressed as:

$$\underset{\mathbf{x}}{\text{minimize}}\ ||\phi(\mathbf{x})||^2. \tag{2.37}$$

Problem (2.37) is an unconstrained energy minimization. Such problems occurs frequently in computer vision because many of the functions to be minimized are L2 norms of various error functions. In essence, this is simply a least-squares minimization but with a nonlinear objective. We assume $\phi \in C^2(\mathbb{R}^n, \mathbb{R}^m)$, i.e., ϕ is a vector valued function and at least once differentiable. We mention two closely related methods that can be used to solve such problems.

2.7.1 Gauss-Newton algorithm

The Gauss-Newton algorithm is an efficient method to compute a local minimum of a non-linear least squares problem such as problem (2.37). The reason it is preferred over other methods such as Newton's method is because it does not require the computation of second derivatives. Gauss-

Newton algorithm, in each step of its iterations, finds the required change in variables for minimizing the first-order approximation of ϕ around the current point $\mathbf{x}_0$. This is written as:

$$\mathbf{s}_{\mathbf{x}} = \arg\min_{\mathbf{s}_{\mathbf{x}}} \|\phi\left(\mathbf{x}_0\right) + \mathsf{J}_\phi \mathbf{s}_{\mathbf{x}}\|^2. \tag{2.38}$$

The Jacobian J_ϕ here is evaluated at the current point $\mathbf{x}_0$. Equation (2.38) can be solved by LLS, for each iteration as follows:

$$\mathbf{s}_{\mathbf{x}} = -\left(\mathsf{J}_\phi^\top \mathsf{J}_\phi\right)^{-1} \mathsf{J}_\phi^\top. \tag{2.39}$$

It can be shown that the direction of each step given by equation (2.39) is towards the descent direction. Such an iterative scheme, however, may or may not converge to a local optimum depending on the provided initial solution and the nature of the function. Under some conditions, Gauss-Newton shows quadratic convergence and in other cases linear. The convergence is not guaranteed specifically when $\mathsf{J}_\phi^\top \mathsf{J}_\phi$ is ill-conditioned.

2.7.2 The Levenberg-Marquardt algorithm

Levenberg-Marquardt algorithm is a modification of Gauss-Newton algorithm that adds a regularization on equation (2.39) so that the new equation is always well-conditioned even when $\mathsf{J}_\phi^\top \mathsf{J}_\phi$ approaches singularity. Thus the new step is obtained as follows:

$$\mathbf{s}_{\mathbf{x}} = -\left(\mathsf{J}_\phi^\top \mathsf{J}_\phi + s \operatorname{diag}\left(\mathsf{J}_\phi^\top \mathsf{J}_\phi\right)\right)^{-1} \mathsf{J}_\phi^\top. \tag{2.40}$$

Equation (2.40) has a single scalar s that controls the amount of regularization. In general, we start with a high regularization when the initial solution can be far from the optimal solution. In that case, the iterations behave as gradient descent. In every iteration we reduce s by a constant factor if we obtain an improvement on the cost function. As s becomes smaller, the iterations behave as in Gauss-Newton algorithm. Because, Gauss-Newton shows better convergence when the current state of the variable is near the optimum, this gives a very balanced strategy.

2.8 Convex Optimization

An overwhelming number of optimization problems in computer vision are highly nonlinear and have several minima or maxima. In general, when there is a nonlinear optimization problem, we resort to one of many local optimization strategies discussed in the previous section, such as Levenberg-Marquardt to compute a locally optimal solution. However, for a certain class of problems known as the convex problems there exists more efficient optimization methods that can guarantee a globally optimal solution. Given the optimization vector $\mathbf{x}$, a convex optimization problem can be expressed as:

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}}\ \phi(\mathbf{x})\\
&\text{subject to,}\\
&\eta_i(\mathbf{x}) \geq 0, \quad i = 1, \ldots, m\\
&\mathbf{A}\mathbf{x} + \mathbf{b} = 0.
\end{aligned} \tag{2.41}$$

The primary condition for problem (2.41) to be convex is that the function $\phi(\mathbf{x})$ has to be convex and $\eta_i(\mathbf{x}) \geq 0$ must define a convex set of the variable $\mathbf{x}$. The affine equality constraints are always convex and in fact, they are the only convex equality constraints. Here, $\phi(\mathbf{x}) \in C^2(\mathbb{R}^n, \mathbb{R})$ and $\eta_i(\mathbf{x}) \in C^2(\mathbb{R}^n, \mathbb{R})$. The twice differentiability of the functions are assumed for computational purpose as most algorithms that are efficient in convex optimization require this condition. Some important convex optimization problems are Linear Programming (LP), Second-Order Cone Programming (SOCP) and Semi-Definite Programming (SDP) in increasing order of complexity and generality. These three problems are important because they can be solved in polynomial time using methods of convex optimization. Later in chapter 6, we make use of convex optimization by formulating the NRSfM problem as an SOCP. In this section, we first give the basic definitions of the three important classes of convex optimization problems. We then provide some descriptions of the interior point method that is often the method of choice for solving convex optimization problems.

Linear Programming. An LP is one of the simplest problems that is solved using techniques in convex optimization. Many other problems such as LLS or even simple linear systems are naturally convex but closed form solutions for such problem are usually the preferred approach. We define an LP as the following convex problem.

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}}\ \mathbf{c}^\top\mathbf{x} + \mathbf{d}\\
&\text{subject to,}\\
&\mathbf{G}\mathbf{x} \leq \mathbf{h},\\
&\mathbf{A}\mathbf{x} = \mathbf{b}.
\end{aligned} \tag{2.42}$$

We can summarize a Linear Program (LP) or problem (2.42) as follows. An LP is a minimization of an affine function subject to affine equality and inequality constraints. Historically, LP refers to a more specific version of problem (2.42), where a linear function is minimized under affine inequality and non-negative variable constraint. However problem (2.42) is the most general form of an LP.

Second-Order Cone Programming. An SOCP is an extension of LP where there are, in addition to affine constraints, cone inequality constraints. Mathematically we express an SOCP problem as:

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}}\ \mathbf{c}^\top\mathbf{x}\\
&\text{subject to,}\\
&\|\mathbf{A}_i\mathbf{x}+\mathbf{b}_i\| \leq \mathbf{c}_i^\top\mathbf{x}+\mathbf{d}_i, \quad i=1,\dots,m\\
&\mathbf{F}\mathbf{x}=\mathbf{g}.
\end{aligned} \tag{2.43}$$

Essentially, it involves minimizing a linear function subject to conic and affine inequality and equality constraints. Cone inequality constraints are defined as the L2-norm of an affine function of the optimization variable being less than or equal to an affine function of the same optimization variable. They are so named because the feasible region of the constraint form a cone in $\mathbb{R}^n$, where n is the dimension of $\mathbf{x}$. There are various methods that can solve SOCP almost as fast as LP (Boyd and Vandenberghe, 2004).

Semi-Definite Programming. Semi-Definite Programs (SDPs) form a more general class of convex problems that can be expressed as the following:

$$\begin{aligned}
&\underset{\mathbf{X}}{\text{minimize}}\ \langle\mathbf{C},\mathbf{X}\rangle, \quad \mathbf{X},\mathbf{C}\in\mathbb{S}^n\\
&\text{subject to,}\\
&\langle\mathbf{A}_i,\mathbf{X}\rangle=\mathbf{b}_i, \quad \mathbf{A}_i\in\mathbb{S}^n,\ i=1,\dots,m\\
&\mathbf{X}\succeq 0.
\end{aligned} \tag{2.44}$$

Here the inner product between any two matrices $\mathbf{A}\in\mathbb{S}^n$ and $\mathbf{B}\in\mathbb{S}^n$ is defined as:

$$\langle\mathbf{A},\mathbf{B}\rangle = \text{trace}(\mathbf{A}^\top\mathbf{B}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}\mathbf{B}_{ij}. \tag{2.45}$$

Problem (2.44) is also referred to as Linear Matrix Inequalities (LMI) as the optimization variable is now a matrix rather than a vector. The nonlinearity is encoded in the fact that the matrix $\mathbf{X}$ has to be positive semi-definite, hence the name SDP. Both LP and SOCP can be written as an SDP, which is a generalization of many convex programs. One very well known problem that comes up in computer vision as an SDP is the max-cut problem in graph theory.

2.8.1 Interior point methods

Interior point methods are a class of iterative methods that can be used to solve constrained linear or nonlinear convex problems. Although unconstrained convex optimization problems can be solved by any local optimization methods such as the Levenberg-Marquardt algorithm, the constrained problems require a different strategy. Interior point methods fall into a class of methods called the barrier methods. Perhaps the most remarkable aspect of these methods is their polynomial time complexity. This is in contrast to older methods, for example, the simplex method for solving LPs. We explain briefly a widely used interior point method known as the primal dual interior point method. Most convex optimization tools use a variation of this method for solving convex problems such as LPs, SOCPs or SDPs. Other variations of interior point methods include the barrier method and the central path method.

Primal-dual interior point method. Several versions of description exist for the primal-dual interior point method and interior point methods in general. We follow the most widely used description of the method, as illustrated in (Boyd and Vandenberghe, 2004) and others. Consider the following constrained convex optimization problem with only inequality constraints.

$$\begin{aligned}&\text{minimize } \phi(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \phi \in C^2(\mathbb{R}^n, \mathbb{R})\\ &\text{subject to,}\\ &\eta_i(\mathbf{x}) \geq 0, \quad \eta_i \in C^2(\mathbb{R}^n, \mathbb{R}),\ i = 1, \dots, m.\end{aligned} \tag{2.46}$$

Interior point methods use a logarithmic barrier function defined as:

$$B(\mathbf{x}, u) = \phi(\mathbf{x}) - u \sum_{i=1}^{m} \log\left(\eta_i(\mathbf{x})\right), \quad u \in \mathbb{R}^{++}. \tag{2.47}$$

Recall that $\mathbb{R}^{++}$ represents the set of all nonzero and positive real numbers. The small value of u ensures that minimizing $B(\mathbf{x}, u)$ is almost equivalent to minimizing $\phi(\mathbf{x})$. Therefore, as $u \to 0$, the minimum of $B(\mathbf{x}, u)$ gives the minimum of $\phi(\mathbf{x})$. We now describe how we can minimize equation (2.47) under the inequality constraints of problem (2.46).

Using the chain rule for differentiation, the Jacobian of the barrier function is obtained as:

$$\mathsf{J}_B = \mathsf{J}_\phi - u \sum_{i=1}^{m} \frac{1}{\eta_i(\mathbf{x})} \mathsf{J}_{\eta_i}. \tag{2.48}$$

Recall that J_ϕ and J_{η_i} denote the Jacobian matrices (in this case, they are row vectors) of $\phi(\mathbf{x})$ and $\eta_i(\mathbf{x})$. We add one more constraint on u by introducing a vector $\mathbf{l} \in \mathbb{R}^m, \mathbf{l} = [l_1, \dots, l_m]^\top$ such that,

$$\eta_i(\mathbf{x}) l_i = u. \tag{2.49}$$

Here l_i is a Lagrange multiplier for the constraint function η_i. We now substitute equation (2.49) into equation (2.48). This lets us establish condition similar to the Karush-Kuhn-Tucker (KKT) conditions for the minimum of a constrained optimization problem as follows.

$$\mathsf{J}_\phi(\mathbf{x}) = \mathsf{J}_\eta^\top(\mathbf{x})\mathbf{l}. \tag{2.50}$$

For convenience we assume $\eta = [\eta_1 \dots \eta_m]^\top$ so that we can define the Jacobian of η as J_η. The iterative algorithm now proceeds by applying Newton's method to equations (2.49) and (2.50). Suppose,

$\delta_{\mathbf{x}}$ and $\delta_{\mathbf{l}}$ are the required small steps on $\mathbf{x}$ and $\mathbf{l}$ respectively, we obtain the following:

$$\begin{bmatrix} \mathbf{H} & -\mathsf{J}_\eta \\ \mathrm{diag}(\mathbf{l})\mathsf{J}_\eta & \mathrm{diag}(\eta) \end{bmatrix} \begin{bmatrix} \delta_{\mathbf{x}} \\ \delta_{\mathbf{l}} \end{bmatrix} = \begin{bmatrix} -\mathsf{J}_\phi + \mathsf{J}_\eta^\top \mathbf{l} \\ u\mathbf{1}_m - \mathrm{diag}(\eta) \end{bmatrix} \tag{2.51}$$

where $\mathbf{H}$ is the Hessian matrix of the barrier function $B(\mathbf{x}, u)$. The steps are then found by solving the linear system of equations (2.51). An important practical consideration overlooked here is the choice of u and the Lagrange multipliers $\mathbf{l}$. Choosing a high value of u requires a large number of outer iterations and a low value implies a large number of Newton iterations. We refer the reader to (Boyd and Vandenberghe, 2004) for more details.

Chapter 3

Previous Work

In this chapter we give a detailed discussion on the previous work in SfT and NRSfM separately. We organize related work in SfT and NRSfM into separate sections. In the first section we describe the state-of-the-art in SfT based on physcial priors and in the second section we give a brief overview of the different NRSfM methods. We emphasize and elaborate the following two important points in the discussion. First, the state-of-the-art methods in SfT show good performance in reconstructing isometric or near-isometric surfaces, except when the projection rays forming the image are close to affine. Second, very few NRSfM methods work with the perspective camera model, and there is a lack of physical-prior based methods. A thorough discussion on previous work in SfT and NRSfM can also be found in (Salzmann and Fua, 2011b) and (Tao, 2014).

3.1 Shape-from-Template

In the timeline of non-rigid 3D reconstruction, SfT methods were proposed much later than the NRSfM methods. For example, (Bregler et al., 2000) is widely credited for one of the first methods proposed for NRSfM. However, the SfT methods have evolved rapidly and matured to realtime applicable methods (Collins and Bartoli, 2015; Ngo et al., 2016). One simple reason for such progress on SfT is the fact that it is far more well-constrained than the NRSfM problem. Nonetheless, it is useful in several scenarios. The methods are mostly based on physical models of surfaces, mainly the isometric model. We here discuss the major work on SfT, needless to say that this does not include every proposed method of SfT. All of the SfT methods discussed here are based on the physical prior of isometry. In order to organize the discussion on previous works in isometric SfT, we classify the methods based on their surface priors and the way they are optimized: *i)* zeroth-order methods based on inextensibility (Ngo et al., 2016; Perriollat et al., 2011; Salzmann and Fua, 2011a), *ii)* statistically optimal cost refinement (Brunet et al., 2014; Collins and Bartoli, 2015) and *iii)* analytical solutions from quadratic PDEs (Bartoli and Collins, 2013; Bartoli et al., 2015). Table 3.1 summarizes important characteristics of these methods.

3.1.1 Zeroth-order methods based on inextensibility

Zeroth-order implies no differential quantities are used for the shape estimation. These methods either represent the surface as a point set with NNG or a mesh. The MDH-based methods in *i)* (Brunet et al., 2014; Perriollat et al., 2011; Salzmann and Fua, 2011a) solve the SfT problem by maximizing depth while putting an upper bound on the distance between neighboring points.

(Perriollat et al., 2011) proposed the Maximum Depth Heuristic (MDH) with the inextensibility as described in equation (2.31). The inextensibility is used as a relaxation of the isometric constraint for computational purposes. In order to apply inextensibility, it considers a point $\mathbf{Q}_i$ at a distance u_i from the camera. Any neighboring point at a distance of u_j from the camera is then parametrized with an angle α_{ij} between the sightlines for $\mathbf{Q}_i$ and $\mathbf{Q}_j$. This gives the following parametrization.

$$\mathbf{Q}_i = \begin{bmatrix} u_i \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{Q}_j = \begin{bmatrix} u_j \cos(\alpha_{ij}) \\ u_j \sin(\alpha_{ij}) \\ 0 \end{bmatrix}. \tag{3.1}$$

Applying equation (3.1) on the inextensibility constraint directly gives the upper bound for u_i as:

$$u_i \leq \frac{d_{ij}}{\sin \alpha_{ij}}, \tag{3.2}$$

where d_{ij} is the geodesic distance between the 3D points $\mathbf{Q}_i$ and $\mathbf{Q}_j$. It should be noted here that taking the upper bound for u_i is equivalent to taking the upper bound of depth for each point. The estimated value for u_i is chosen using all the neighbors as the minimum value of the set of all the upper bounds:

$$\hat{u}_i = \min_{j=1,\dots,n_i,\ j\neq i} \left(\frac{d_{ij}}{\sin \alpha_{ij}} \right). \tag{3.3}$$

However, the computed values of $\hat{u}_i$ require a few iterations of refinement to make all the bounds coherent with the inextensibility constraint of equation (2.31).

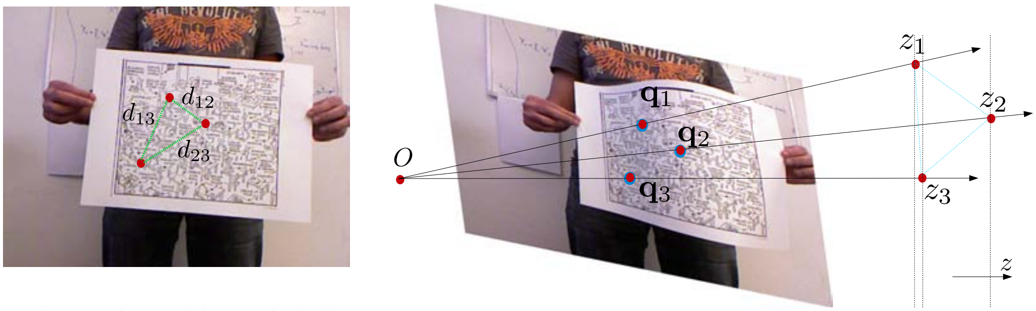
(Salzmann and Fua, 2011a) reformulated the MDH as a convex problem and applied Second-Order Cone Programming (SOCP) to compute depth. The method uses a more natural parametrization of the 3D points using the perspective back-projection of a 2D image point as in equation (2.6). We rewrite the equation here in a more relevant notation as:

$$\mathbf{Q}_i = z_i \begin{bmatrix} \mathbf{q}_i \\ 1 \end{bmatrix}, \tag{3.4}$$

where z_i is the depth at the ith point and $\mathbf{q}_i$ is the normalized image correspondence at the point. The basics of the method can be explained with the following optimization problem.

$$\begin{aligned} & \underset{\{z_i\}}{\text{maximize}} \sum_{i=1}^{n} z_i, \\ & \text{subject to,} \quad \forall i \in \{1 \dots n\},\ j \in \mathcal{N}(i) \\ & z_i \geq 0 \\ & \left\| z_i \begin{bmatrix} \mathbf{q}_i \\ 1 \end{bmatrix} - z_j \begin{bmatrix} \mathbf{q}_j \\ 1 \end{bmatrix} \right\|_2 \leq d_{ij}. \end{aligned} \tag{3.5}$$

Here $\mathcal{N}(i)$ denotes the set of indices j for which $\mathbf{Q}_j$ is in the neighborhood of a point $\mathbf{Q}_i$. In problem (3.5), the sum of all depths are maximized subject to the inextensibility constraint of equation (2.31). Figure 3.1 illustrates the MDH with a single deformed image and a known template. It is clear from the figure that MDH methods work due to the perspective effect, i.e., the diverging sightlines as depths are increased also increase the distances between points on the sightlines. In practice (Salzmann and Fua, 2011a) imposes robustness in problem (3.5) and performs the optimization also introducing a reprojection error. It also makes use of a learned space of deformations using a linear local model for each small patch on the mesh. Although such an approach limits the method to the use of surfaces



Flat Template and the geodesic distances

Input Image

Bounds on z_i due to inextensibility

Figure 3.1: Illustration of MDH showing the 3D template distances (on the left) and the sightlines where depths are maximized (on the right).

where learned models for patches already exist, the global convex formulation does give it an edge over several other methods. However, the method requires enough perspective for a stable solution.

(Ngo et al., 2016) proposed a modified approach, where the method uses the inextensibility constraints but does not maximize the depth explicitly. Instead, the method solves for the deformed surface by finding a transformation of the original mesh to the deformed mesh with mesh vertices represented by $\mathbf{x} \in \mathbb{R}^{N_v \times 1}$, N_v being the total number of vertices. It uses a mesh Laplacian to parametrize the surface up to a rigid transform. This is given by:

$$\|\mathbf{A}\mathbf{x}\|^2 = 0. \tag{3.6}$$

The method gives a unique solution to the Laplacian $\mathbf{A}$ that can be found using the reference or template mesh. For that purpose, the vertices on the mesh are linearly parametrized with *control mesh vertices*. Let us assume $\mathbf{c}$ is the vector of N_c control mesh vertices ($N_c \ll N_v$) written as:

$$\mathbf{c} = \begin{bmatrix} \mathbf{v}_{i_1} \\ \vdots \\ \mathbf{v}_{i_{N_c}} \end{bmatrix}. \tag{3.7}$$

Then the linear parametrization of $\mathbf{x}$ is written with a matrix $\mathbf{P} \in \mathbb{R}^{3N_v \times 3N_c}$:

$$\mathbf{x} = \mathbf{P}\mathbf{c}. \tag{3.8}$$

This effectively reduces the size of the actual problem while at the same time introduces smoothness to the structure of $\mathbf{x}$. The Laplacian parametrization removes the learning mechanism of (Salzmann and Fua, 2011a) and puts a smoothness prior on the space of deformations. The parametrization described in equations (3.6) and (3.8) is inspired from methods proposed in computer graphics (Sorkine et al., 2004; Sumner and Popović, 2004). Such priors are well suited for most applications because they do not make statistical assumptions about the material properties of surfaces. However, the method requires solving a non-convex problem in the end. The solution requires an iterative approach and the computation time does not scale up linearly with increase in the mesh size. The zeroth-order methods are non-analytical and consequently some of their theoretical aspects are still not well understood. On the one hand, it is clear that inextensibility does not constrain depth strongly in near-affine conditions as well as in perspective conditions. On the other hand, zeroth-order inextensibility-based methods can be formulated and solved in the point set with neighborhood representation, i.e., no differential quantities are required. We will describe inextensibility and MDH again in more details, as we revisit them in our solution for NRSfM in chapter 6.

3.1.2 Statistically optimal cost refinement

A complete statistically optimal cost for isometric SfT was proposed in (Brunet et al., 2014). The method optimizes a statistically optimal cost that includes the 3D back-projection constraint, differential isometric constraint and smoothness. The problem is written as:

$$\underset{\phi(\mathbf{p})}{\text{minimize}}\ E_{data} + l_{iso}E_{iso} + l_{smooth}E_{smooth}. \tag{3.9}$$

Problem (3.9) minimizes three error terms to compute the embedding of the deformed surface $\phi(\mathbf{p}) \in C^2(\mathbb{R}^2, \mathbb{R}^3)$. The embedding is represented using a BBS function of a point on the flat template $\mathbf{p} \in \mathbb{R}^2$. E_{data} denotes back-projection error. E_{iso} is the error measured using the differential isometric constraint (2.30) at each point. The smoothness E_{smooth} is measured using the bending energy of the BBS representing ϕ. These error terms are weighted by using the scalars l_{iso} and l_{smooth}. On the one hand problem (3.9) is non-convex and relies on iterative local optimization such as Levenberg-Marquardt. Such a refinement involves the use of an initialization and requires a much higher computation time. Apart from that, the optimization requires relative weighting of the three constraints with two parameters l_{iso} and l_{smooth} that need to be precisely tuned to get optimal results. On the other hand, it carries the advantage of having a holonomic solution of depth, where the relationship between depth and its Jacobian is taken into account. This also means that it does not suffer from the depth instability in near-affine conditions as is the case for other methods. The method of (Collins and Bartoli, 2015) is more general than (Brunet et al., 2014) and works in real-time. It handles arbitrary surface meshes and solves the registration and shape inference problem together using dense point matches. However, it also requires an initial solution, which is obtained by tracking the object frame-to-frame.

There is also a class of methods which do tracking and reconstruction together (Malti et al., 2011; Ngo et al., 2015; Yu et al., 2015) similar to (Collins and Bartoli, 2015). (Malti et al., 2011) introduced the pixel intensity error instead of the feature-based reprojection error in equation (3.9) to obtain simulaneous tracking and reconstruction. A conformal deformation prior was chosen here which is more general than the isometric prior. (Ngo et al., 2015) does the same with a better method for initialization proposed in (Ngo et al., 2016) and M-estimators to handle occlusions and poorly textured surfaces. (Yu et al., 2015) uses an energy for temporal smoothness apart from the remaining costs to get a dense reconstruction and tracking in RGB videos.

3.1.3 Analytical solutions from quadratic PDEs

The analytical methods (Bartoli and Collins, 2013; Bartoli et al., 2015) use a flat template-to-image registration warp and the surface parametrization function derivatives to directly compute the surface's depth analytically. At each surface point, the depth is obtained as the non-holonomic solutions to a PDE system. It uses the zeroth and first-order information like the methods in section 3.1.2 but the solutions are obtained analytically. This requires computing the gradient of the local image warp. Methods of this type discard the depth-gradient solution and keep only the depth solution. The local analytical nature of the solutions means that the methods are very fast and can be parallelized efficiently as the solution for each point is found independently. As an advantage of being analytical, these methods also form a powerful tool to analyze the effect of different projection geometries on the recovered shape. We will elaborate the analytical solutions of (Bartoli et al., 2015) in details when

Table 3.1: SfT methods and their characteristics.

Methods	Surface Representation	Surface Prior	Constraint type	Primary computation	Affine Stability
(Perriollat et al., 2011)	Point set with NNG	Inextensibility	Zeroth-order	Non-convex optimization	Not stable
(Salzmann and Fua, 2011a)	Point set with NNG	Inextensibility	Zeroth-order	Convex (SOCP)	Not stable
(Ngo et al., 2016)	Mesh	Inextensibility	Zeroth-order	Non-convex optimization	Not stable
(Brunet et al., 2014)	2D Riemannian Manifold	Isometry	First-order	Non-convex optimization	Stable
(Bartoli et al., 2015)	2D Riemannian Manifold	Isometry	First-order	Analytic	Not stable
Proposed Methods	2D Riemannian Manifold	Isometry	First-order	LLS	Stable

we discuss our method for SfT in chapter 4.

Our proposed methods for SfT are analytical for the most part and yet do not suffer from depth instability at near affine conditions. Table 3.1 lists some basic characteristics of the state-of-the-art methods in SfT and compares them to the proposed methods in these characteristics.

3.2 Non-Rigid Shape-from-Motion

In contrast to SfT methods, NRSfM has seen very few methods based on physical models such as isometry. The factorization-based approaches using the low-rank deformation model have been the focus of research in NRSfM for a long time. NRSfM methods can be divided in many ways. For clarity we classify them as methods based on statistical priors (low-rank methods) and methods based on physical priors.

3.2.1 Methods based on statistical priors

Starting from the work of Bregler et al. (Bregler et al., 2000), the low rank model has been the most commonly used shape prior in NRSfM. The low-rank shape model was described in chapter 2.6.3, which is a statistical prior on a set of surfaces. As shown before, the low rank prior is used to obtain surface 3D reconstructions by factorizing the matrix of point correspondences or observation matrix into a coefficient matrix and shape basis matrix. However, such a factorization under the low rank constraint is non-convex and suffers from ambiguities. Many works have been proposed to include additional priors in resolving the ambiguities of factorization-based NRSfM. Additional priors are important here even after applying the low-rank constraint because some shape ambiguities remain in affine projections (Collins and Bartoli, 2010; Pizarro et al., 2013). We discuss the priors and methods introduced to solve the inherent problems of low-rank NRSfM below.

***i)* Shape basis priors.** Shape basis priors were introduced in (Del Bue, 2008) to constrain better the shape basis matrix $\mathbf{V}$ in equation (2.34). The basic assumption is that a collection of known 3D shapes are generated from a subset of the same shape basis as the unknown shapes to be reconstructed. This puts limits on the shape basis of the unknown shapes, consequently constraining the unknown shapes. It proposes to modify equation (2.34) so that the unknown shapes are obtained in part from a known shape basis. The known part, say b number of basis shapes are precomputed from a known sequence of 3D shapes. This gives us the following equation.

$$\mathbf{W} = \begin{bmatrix}\mathbf{L}_b & \mathbf{L}_{l-b}\end{bmatrix}\begin{bmatrix}\mathbf{B}_b \\ \mathbf{B}_{l-b}\end{bmatrix} \tag{3.10}$$

A linear basis of the known 3D shapes is given by:

$$\mathbf{P} = \mathbf{N}\mathbf{B}_b \tag{3.11}$$

where the matrix $\mathbf{P} \in \mathbb{R}^{b \times n}$ is the shape prior obtained from the linear basis $\mathbf{B}_b$ of the known 3D shapes. Extending the idea described in (Del Bue, 2008), (Tao and Matuszewski, 2013) proposed representing a non-linear shape prior with few parameters in a low-dimensional manifold. This allows the shapes to have non-linear complex deformations.

***ii)* Spatio-temporal smoothness prior.** Smoothness, whether spatial/surface smoothness or temporal smoothness, is a natural property of many real objects and can be modelled in the reconstruction process in various ways. Spatial smoothness was first used in NRSfM in (Torresani et al., 2001) using a regularizer for neighboring points. (Olsen and Bartoli, 2008) proposed to use spatial as well as temporal terms. Spatio-temporal smoothness was also exploited by (Torresani et al., 2008) using a Probabilistic Principal Component Analysis (PCA) model of shape. A Gaussian prior is then used on the weights (matrix $\mathbf{L}$ of the shape basis). An alternative method to impose temporal smoothness was proposed by (Akhter et al., 2008). It models the shape coefficients with Discrete Cosine Transform (DCT) as $\mathbf{W} = \mathbf{LV} = \mathbf{D}(\mathbf{C})\mathbf{V}$. $\mathbf{D}$ consists of rotations while $\mathcal{C}$ contains basis of the time-trajectory of 3D points. However this limits the number of DCT coefficients by the selected rank l of the matrix $\mathbf{W}$. (Gotardo and Martínez, 2011) proposed an alternative formulation by which it models W with high and low frequency DCT coefficients and the basis matrix without increasing the rank parameter l.

***iii)* Linear and nonlinear combination of shape basis.** Most low-rank methods express the unknown shapes as a linear combination of the basis shapes. In other words, the unknown shapes lie on the linear subspace of the basis shapes. This demands the deformations to be limited to small linear ones. To model larger complicated deformations (Gotardo and Martínez, 2011) proposed to apply a kernel transformation. The kernel transformation generalizes the inner product with a kernel such as the radial basis function. As such, this itself is not a prior and rather expands the solution space making the problem more difficult to solve. In order to find the solution in such cases, (Gotardo and Martinez, 2011) proposed the shape trajectory analysis. Linear shape basis problem on the other hand, have the advantage that they can be solved more efficiently. (Dai et al., 2012) showed that a convex relaxation of the linear shape basis problem performs better than many other low-rank methods with additional constraints.

***iv)* Choice of rank.** Another problem in low-rank NRSfM is the choice of the rank parameter l. In general, the choice of the rank l affects the solution due to the way the problem is solved. (Garg et al., 2013a) proposed a joint optimization that computes the rank parameter as well as the unknown shapes. However, there is still no guarantee that a given set of surfaces can be accurately represented by a matrix with the chosen rank.

3.2.2 Methods based on physical priors

Physical model-based approaches have been introduced to avoid the difficulties and problems with statistical priors. As in the case of SfT, efforts have been made on using isometry to constrain the problem in NRSfM (Chhatkuli et al., 2014b; Taylor et al., 2010; Varol et al., 2009; Vicente and Agapito, 2012). Physical model-based methods, in general, can handle larger complex deformations and at the same time work with very small number of images than methods that use statistical priors. The isometric prior can be used in the NRSfM problem locally (point-wise) or semi-locally (patch-wise) or even globally by considering the whole set of surfaces and image points together. We describe the physical model-based methods below in two categories, local or semi-local and global:

***i)* Local or semi-local approaches.** Local approaches do not combine constraints at different points or set of points. They evaluate the constraints independently to obtain point-wise or point set-wise reconstruction. A semi-local method using a perspective camera and homographies was proposed in (Varol et al., 2009). It can reconstruct surfaces that are composed of large planar patches. As described in chapter 2.3.3, a homography obtained from normalized image points of a plane can be decomposed to obtain the surface normal of the plane. (Varol et al., 2009) first divides the surface into planar patches and estimates the homography for each planar patch. It then decomposes the homography obtained at each plane to obtain the solutions for surface normal. However, the obtained solutions have a two-fold ambiguity and consequently it resorts to spatial smoothness of surfaces for disambiguating the surface normals. The final shapes are then obtained by integrating surface normals. The deformation is modelled using only a rotation and translation, i.e. a rigid transform for each region of the surface. This is can be regarded as a special case of isometry.

A more general isometric model with local rigidity is exploited by (Collins and Bartoli, 2010; Taylor et al., 2010) using the affine camera model. (Taylor et al., 2010) proposed a 3-point rigid SfM solution with a convex relaxation and gave an NRSfM method by assuming each 3-point set is rigid, i.e. the surface is locally rigid. This is more general than the assumption of piece-wise rigidity of (Varol et al., 2009) but requires a minimum of 4 images. (Collins and Bartoli, 2010) at the same time proposed a similar solution with the assumption of local rigidity but without using a convex relaxation. It used automatically clustered point sets and solved the general case of three or more images.

An interesting semi-local solution was proposed in (Russell et al., 2014) based on local fundamental matrices computed from small point sets. However it also uses piecewise rigidity assumption over the more general local rigidity prior. Thus, it is more suitable for articulated objects than for smoothly deforming surfaces. A recent method for local solution of NRSfM was proposed in (Parashar et al., 2016). It shares the local (infinitesimal) planarity assumption made in the proposed local solution of the thesis but goes further by using the manifold representation of surfaces. The local solutions to the surface normals are computed by exploiting the properties of the metric tensor. The metric tensors are transferred across different images which involves second-order quantities called the Christoffel symbols.

***ii)* Global approaches.** Global approaches combine all the constraints in all the points together to form the problem formulation. The use of the combined constraints means that the methods can possibly handle difficult conditions if an optimal solution can be found. However, to optimize such a large system of constraints most methods have to employ an energy minimization scheme. (Vicente and Agapito, 2012) proposed one such global approach. It uses the isometric constraints under the assumption of an orthographic camera. The method also provides a way to include the perspective camera. However, the solutions are obtained with discrete non-convex optimization on an initial solution and are not globally optimal. Furthermore, it is a complex method to implement and test. Some global approach also mix the physical models with the statistical low-rank model. For example, (Agudo and Moreno-Noguer, 2015) uses a shape basis as well as an isometry-like prior but the method requires an initialization, obtained from rigid factorization on the first set of frames. In that regard, it could be argued that the core of the method is rather like a template-based approach.

Table 3.2 lists some important methods and their characteristics in comparison to the proposed methods. We propose two different solutions. Our local solution gives pointwise solution to surface normals and disambiguates them using additional views rather than smoothness. The method is based on the perspective camera model. Compared to all previous methods, our global method is the first to formulate a convex problem by relaxing isometry to inextensibility in NRSfM, from which we obtain a globally optimal solution using SOCP. The method is fast, accurate, simple to understand and uses a perspective camera model.

Table 3.2: NRSfM methods and their characteristics.

Methods	Surface Representation	Surface Prior	Camera Model	Constraint type	Primary computation
(Gotardo and Martínez, 2011)	Point sets	Low-rank and temporal smoothness	Orthographic	Global	Non convex
(Dai et al., 2012)	Point sets	Low-rank	Orthographic	Global	Convex with non-convex refinement
(Taylor et al., 2010)	Mesh	Isometry	Orthographic	Local	Small systems
(Vicente and Agapito, 2012)	Point sets with NNG	Isometry	Orthographic and perspective	Global	Non-convex
(Parashar et al., 2016)	2D Riemannian Manifold	Isometry	Perspective	Local	Small quartic systems
Proposed local method	2D Riemannian Manifold (implicit)	Isometry	Perspective	Local	Small systems
Proposed global method	Point sets with NNG	Inextensibility	Perspective	Global	Convex

Chapter 4

Shape-from-Template

We give our problem modelling for SfT and describe two important related methods for the template-based reconstruction from a single image. While doing that, we also establish why previous methods are not suited as local projection geometry tends to affine. The methods we provide here are initialization-free and for the most part analytic. We detail our experiments and results on developable and non-developable surfaces undergoing smooth deformations, which show that the proposed methods perform better than the state-of-the-art in perspective as well as near-affine conditions. This chapter is based on our published work (Chhatkuli et al., 2016b).

4.1 Differential Geometric and PDE-based Modelling

We follow the problem modelling of (Bartoli et al., 2015) as depicted in figure 4.1. We start with the 3D template $\mathcal{T} \subset \mathbb{R}^3$. The flat template is a 2D domain $\Omega \subset \mathbb{R}^2$ obtained from $\mathcal{T}$, which is parametrized with $\Delta \in C^1(\Omega, \mathbb{R}^3)$. Consequently the flattening function is Δ^{-1}. In practice it is obtained from a conformal flattening of a texture-mapped mesh or often simply by taking an image of the 3D object. The 3D template $\mathcal{T}$ is transformed by an isometric deformation $\psi \in C^1(\mathcal{T}, \mathbb{R}^3)$. The deformed surface $\mathcal{S} \subset \mathbb{R}^3$ is projected by a known camera projection function Π onto an image $\mathcal{I} \subset \mathbb{R}^2$. We define $\mathcal{S}$ in the camera's coordinate frame. We denote the registration between Ω and $\mathcal{I}$ as $\eta \in C^1(\Omega, \mathbb{R}^2)$. We parametrize the deformed surface $\mathcal{S}$ by an unknown embedding function $\varphi \in C^1(\Omega, \mathbb{R}^3)$.

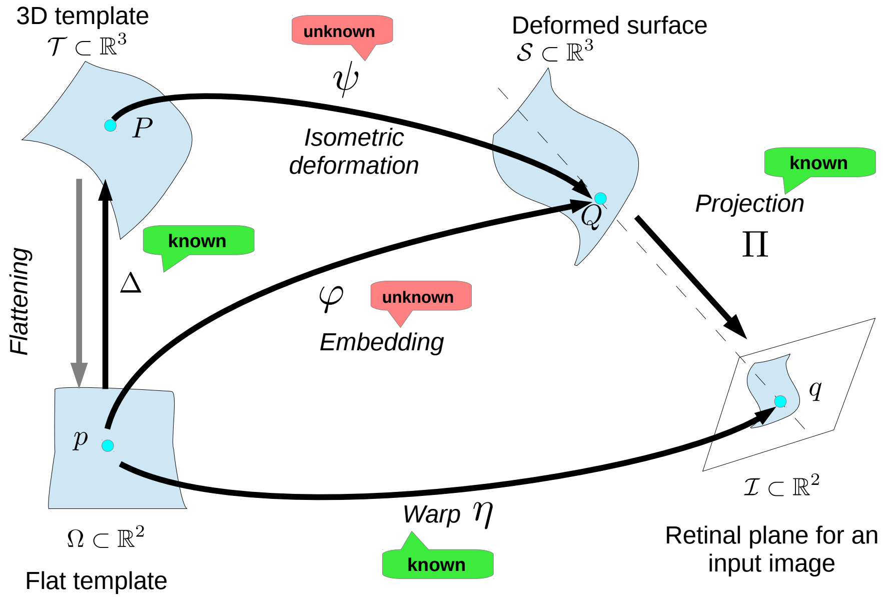


Figure 4.1: Differential geometric modelling of Shape-from-Template.

Our goal is to solve the SfT problem, represented by ψ. In practice we work with the embedding φ. This is equivalent since $\varphi = \psi \circ \Delta$. We obtain φ from the known functions Δ, η and Π, and the fact that the surface deforms isometrically. Below we describe the differential constraints and then define the SfT problem with a set of PDEs.

4.1.1 Differential constraints

We divide the constraints on φ into the deformation constraint and the reprojection constraint. The deformation constraint imposes the isometric prior while the reprojection constraint is analogous to a data term, which ensures that the reprojection of the surface matches the input image. This information is related to the camera geometry and imaging via the known projection function Π. We give details for both constraints as presented in (Bartoli and Collins, 2013; Bartoli et al., 2015) and additionally generalize them for different camera models with which we obtain a generalized equation in section 4.2.

4.1.1.1 Deformation constraint

We start with the equation for the embedding $\varphi = \psi \circ \mathit{\Delta}$; its differentiation leads to:

$$\mathsf{J}_{\varphi} = (\mathsf{J}_{\psi} \circ \mathit{\Delta})\, \mathsf{J}_{\mathit{\Delta}}. \tag{4.1}$$

Pre-multiplying equation (4.1) by its transpose gives us:

$$\mathsf{J}_{\varphi}^{\top} \mathsf{J}_{\varphi} = \mathsf{J}_{\mathit{\Delta}}^{\top} (\mathsf{J}_{\psi} \circ \mathit{\Delta})^{\top} (\mathsf{J}_{\psi} \circ \mathit{\Delta})\, \mathsf{J}_{\mathit{\Delta}}. \tag{4.2}$$

Isometric deformations preserve geodesic distances and for such deformations we have:

$$(\mathsf{J}_{\psi} \circ \mathit{\Delta})^{\top} (\mathsf{J}_{\psi} \circ \mathit{\Delta}) = \mathbf{I}_3, \tag{4.3}$$

which simply states that the metric tensor on the surface remains unchanged with an isometry described by ψ. Substituting equation (4.3) in equation (4.2) gives:

$$\mathsf{J}_{\varphi}^{\top} \mathsf{J}_{\varphi} = \mathsf{J}_{\mathit{\Delta}}^{\top} \mathsf{J}_{\mathit{\Delta}}. \tag{4.4}$$

Equation (4.4) states that the first fundamental form is preserved in an isometry. Thus it remains the same for the surface embedding φ and the template parametrization $\mathit{\Delta}$.

4.1.1.2 Reprojection constraint

The reprojection constraint is obtained with the reprojection equation:

$$\eta = \mathit{\Pi} \circ \varphi. \tag{4.5}$$

Equation (4.5) enforces consistency between the warp η and the projection of the embedding in the image. Without loss of generality we assume that the world coordinate frame is the camera's and we denote as $f > 0$ the camera's focal length. We then use the reprojection constraint to express the embedding $\varphi = [\varphi_x\ \varphi_y\ \varphi_z]^{\top}$ with the depth function $\varphi_z \in C^1(\mathit{\Omega}, \mathbb{R})$. We consider two possible camera models: the perspective camera and the infinitesimal weak-perspective camera.

The perspective camera. With perspective projection $\mathit{\Pi}^{\mathrm{P}}$ we have:

$$\eta = \mathit{\Pi}^{\mathrm{P}} \circ \varphi = \left[f \frac{\varphi_x}{\varphi_z} \quad f \frac{\varphi_y}{\varphi_z} \right]^{\top}. \tag{4.6}$$

Using equation (4.6) the embedding φ may be parametrized by the depth function φ_z and the template-to-image warp function in homogeneous coordinates $\tilde{\eta}^\top = [\eta^\top \; 1]$ as:

$$\varphi = \Phi^{\mathrm{P}}\tilde{\eta} \quad \text{with} \quad \Phi^{\mathrm{P}} = \operatorname{diag}\left(\frac{\varphi_z}{f}, \frac{\varphi_z}{f}, \varphi_z\right). \tag{4.7}$$

The infinitesimal weak-perspective camera. This camera model was proposed to simplify the PDEs by approximating the gradient (Bartoli et al., 2013). It is based on the weak-perspective model, which approximates the perspective camera (Hartley and Zisserman, 2004). It first projects the scene orthographically onto a fronto-parallel plane placed at the scene's average depth and then scales it. The infinitesimal weak-perspective camera instantiates a weak-perspective camera at each point. This gives the same projection as the perspective camera but simplifies the expression for the depth-gradient. It is non-analytic. The infinitesimal weak-perspective projection Π^{WP} yields:

$$\eta = \Pi^{\mathrm{WP}} \circ \varphi = \left[f\tfrac{\varphi_x}{\zeta} \quad f\tfrac{\varphi_y}{\zeta}\right]^\top. \tag{4.8}$$

In this model ζ represents the depth. It is different at each point and given by φ_z, while preserving the property that $\mathsf{J}_\zeta = \mathbf{0}_{1\times 2}$. The back-projection equation with the infinitesimal weak-perspective model is:

$$\varphi = \Phi^{\mathrm{WP}}\tilde{\eta} \quad \text{with} \quad \Phi^{\mathrm{WP}} = \operatorname{diag}\left(\frac{\zeta}{f}, \frac{\zeta}{f}, \varphi_z\right). \tag{4.9}$$

Unified camera model. We give a unified model for a general camera by rewriting the reprojection constraint with the back-projection matrix as:

$$\varphi = \Phi\tilde{\eta} \quad \text{with} \quad \Phi \in \{\Phi^{\mathrm{P}}, \Phi^{\mathrm{WP}}\}. \tag{4.10}$$

The partial derivatives of the back-projection matrix Φ is:

$$\mathbf{M} = \frac{\partial \Phi}{\partial \varphi_z} = \begin{cases} \mathbf{M}^{\mathrm{P}} = \operatorname{diag}(\frac{1}{f}, \frac{1}{f}, 1) & \text{perspective} \\ \\ \mathbf{M}^{\mathrm{WP}} = \operatorname{diag}(0, 0, 1) & \text{infinitesimal weak-perspective.} \end{cases} \tag{4.11}$$

4.1.2 General PDE

The constraints described in section 4.1.1 were first used in (Bartoli et al., 2015) to analytically solve isometric SfT and to show that it is a well-posed problem. Here, we give the following generalized PDE system to describe SfT:

$$\text{Find } \varphi \text{ s.t.} \begin{cases} \mathsf{J}_\varphi^\top \mathsf{J}_\varphi = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta \\ \Pi \circ \varphi = \eta. \end{cases} \tag{4.12}$$

In the following sections we reformulate system (4.12) and show that it intrinsically has three non-holonomic unknowns and three equations. The holonomic solutions for system (4.12), which enforces the differential dependency between the solutions may not exist in practice in the presence of noise. We study the space of solutions of system (4.12) using two different PDE formulations, which result in type-I solutions and type-II solutions. The type-I solutions provide the radial component of depth and its gradient. The type-II solutions give the depth and the surface normal. The two formulations provide the same direct depth solution, however as we will show, the second non-holonomic solutions for type-I and type-II solutions are used differently in the subsequent steps that produce accurate but slightly different reconstructions.

4.2 Type-I Solutions Stability and Type-I Stable Method

The well-posedness of isometric SfT defined by the PDE system (4.12) was proved in (Bartoli et al., 2015) where the local non-holonomic solutions for the PDE system were derived. We first describe the analytic method to obtain the non-holonomic solutions as given by (Bartoli and Collins, 2013; Bartoli et al., 2015). In order to achieve that, they reformulate system (4.12) so that it consists of three non-holonomic unknowns: the depth and its gradient with respect to the flat template. Our contribution here is that we generalize the equations for two different camera models and derive solutions for both models.

4.2.1 Type-I solutions

We consider the embedding parametrized by the depth function φ_z as shown in equation (4.7) or (4.9). The problem of SfT can be viewed as that of finding the depth function φ_z so that the deformation constraints (4.4) are met. We derive a non-linear PDE system that holds for both perspective and infinitesimal weak-perspective projection (4.10) and deformation constraints (4.4). We first differentiate equations (4.7) and (4.9) and use the expressions defined in equation (4.11) to get J_φ:

$$\mathsf{J}_\varphi = \mathbf{M}\tilde{\eta}\mathsf{J}_{\varphi_z} + \mathit{\Phi}\mathsf{J}_{\tilde{\eta}}, \tag{4.13}$$

To verify equation (4.13) one can expand equation (4.7) or (4.9) before differentiating them. Similarly, following through the matrix multiplications in equation (4.13) we reach the same result.

We introduce equation (4.13) in the deformation constraint (4.4) to obtain the following non-linear PDE system:

$$\mathsf{J}_{\varphi_z}^\top \tilde{\eta}^\top \mathbf{M}^2 \tilde{\eta} \mathsf{J}_{\varphi_z} + \mathsf{J}_{\varphi_z}^\top \tilde{\eta}^\top \mathbf{M}\mathit{\Phi}\mathsf{J}_{\tilde{\eta}} + \mathsf{J}_{\tilde{\eta}}^\top \mathit{\Phi}\mathbf{M}\tilde{\eta}\mathsf{J}_{\varphi_z} + \mathsf{J}_{\tilde{\eta}}^\top \mathit{\Phi}^2 \mathsf{J}_{\tilde{\eta}} \quad = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta. \tag{4.14}$$

System (4.14) models SfT in terms of φ_z and J_{φ_z} for perspective and infinitesimal weak-perspective projections. Assuming J_{φ_z} and φ_z are independent variables, we obtain the non-holonomic solutions of system (4.14) analytically. We denote them as $\bar{\varphi}_z \in C^1(\mathit{\Omega}, \mathbb{R})$ and $\bar{\kappa} \in C^0(\mathit{\Omega}, \mathbb{R}^2)$. Despite the fact that system (4.14) admits exact solutions for both $\bar{\varphi}_z$ and $\bar{\kappa}$, they are not generally consistent since $\mathsf{J}_{\bar{\varphi}_z} \neq \bar{\kappa}$. With errors in η, system (4.14) is in fact an overdetermined PDE system with no general (*i.e.* holonomic) solutions.

4.2.1.1 Perspective camera

The PDE system (4.14) is specialized to perspective projection by choosing $\varPhi^{\mathrm{P}}$ from equation (4.7) and $\mathbf{M}^{\mathrm{P}}$ from equation (4.11):

$$\left(1+\frac{\eta^\top\eta}{f^2}\right)\mathsf{J}_{\varphi_z}^\top\mathsf{J}_{\varphi_z}+\frac{\varphi_z}{f^2}(\mathsf{J}_{\varphi_z}^\top\eta^\top\mathsf{J}_\eta+\mathsf{J}_\eta^\top\eta\mathsf{J}_{\varphi_z})+\frac{\varphi_z^2}{f^2}\mathsf{J}_\eta^\top\mathsf{J}_\eta=\mathsf{J}_\Delta^\top\mathsf{J}_\Delta.$$

We simplify this system by changing variables with:

$$\alpha=\varphi_z\nu \quad \text{and} \quad \nu=\sqrt{1+\frac{\eta^\top\eta}{f^2}}, \tag{4.15}$$

giving $\mathsf{J}_\alpha=\nu\mathsf{J}_{\varphi_z}+\frac{\varphi_z}{\nu f^2}\eta^\top\mathsf{J}_\eta$. This leads to an equivalent but simpler PDE system in α and J_α:

$$\mathsf{J}_\alpha^\top\mathsf{J}_\alpha+\alpha^2\gamma=\mathsf{J}_\Delta^\top\mathsf{J}_\Delta, \tag{4.16}$$

where:

$$\gamma=\frac{1}{\nu^2f^2}\left(\mathsf{J}_\eta^\top\mathsf{J}_\eta-\frac{1}{\nu^2f^4}\mathsf{J}_\eta^\top\eta\eta^\top\mathsf{J}_\eta\right). \tag{4.17}$$

In order to obtain the non-holonomic solutions we subsitute J_α by an independent vector function $\beta\in C^0(\varOmega,\mathbb{R}^2)$ in equation (4.16). This gives us:

$$\beta^\top\beta+\alpha^2\gamma=\mathsf{J}_\Delta^\top\mathsf{J}_\Delta. \tag{4.18}$$

Following (Bartoli and Collins, 2013; Bartoli et al., 2015) we can always find a single algebraic solution of system (4.18). We denote the non-holonomic solutions of α and β as $\bar{\alpha}$ and $\bar{\beta}$. To obtain them we first modify equation (4.18) as:

$$\beta^\top\beta\gamma^{-1}=\mathsf{J}_\Delta^\top\mathsf{J}_\Delta\gamma^{-1}-\alpha^2\mathbf{I}_2. \tag{4.19}$$

The left-hand side of equation (4.19) clearly has rank 1 and therefore the second eigenvalue of the right hand side is 0 as well. With that we obtain:

$$\bar{\alpha}=\sqrt{\lambda_2\left(\mathsf{J}_\Delta^\top\mathsf{J}_\Delta\gamma^{-1}\right)}. \tag{4.20}$$

Substituting the solution for α in equation (4.18) the non-holonomic solution $\bar{\beta}$ is obtained as:

$$\bar{\beta}=\pm\sqrt{\lambda_1(\varUpsilon)}\mathbf{v}_1(\varUpsilon), \tag{4.21}$$

where:

$$\varUpsilon=\mathsf{J}_\Delta^\top\mathsf{J}_\Delta-\lambda_2\left(\mathsf{J}_\Delta^\top\mathsf{J}_\Delta\gamma^{-1}\right)\gamma. \tag{4.22}$$

We may recover $\bar{\varphi}_z$ from equation (4.20) followed by the change of variable (4.15). Instead of $\bar{\kappa}$, we recover $\bar{\beta}$ from equation (4.21). The solutions $\bar{\alpha}$ and $\bar{\beta}$ are the non-holonomic solutions of the perspective type-I PDE system (4.18) obtained algebraically and thus they exist in all practical circumstances. In our stable type-I method we give an alternative for obtaining the depth at each point using $\bar{\beta}$.

4.2.1.2 Infinitesimal weak-perspective camera

The PDE for the infinitesimal weak-perspective camera is found by choosing Φ^{WP} from equation (4.9) and $\mathbf{M}^{\text{WP}}$ from equation (4.11):

$$\mathsf{J}_{\varphi_z}^\top \mathsf{J}_{\varphi_z} + \frac{\zeta^2}{f^2}\mathsf{J}_\eta^\top \mathsf{J}_\eta = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta, \tag{4.23}$$

where we set $\zeta = \varphi_z$ in the infinitesimal weak-perspective model as ζ gives the 'average' depth at a differential level.

System (4.23) has exactly the same structure as system (4.16), the simplified PDE system for perspective projection. We directly obtain the non-holonomic solutions of the system $\bar{\varphi}_z$ and $\bar{\kappa}$, without any change of variable. To obtain $\bar{\varphi}_z$ and $\bar{\kappa}$ in the infinitesimal weak-perspective model we first assign $\gamma^{\text{WP}} = f^{-2}\mathsf{J}_\eta^\top \mathsf{J}_\eta$. This transforms equation (4.23) into the following:

$$\kappa^\top \kappa + \varphi_z^2 \gamma^{\text{WP}} = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta. \tag{4.24}$$

Noting the similarity of equation (4.24) with equation (4.18), we give the non-holonomic solutions for equation (4.24) as:

$$\bar{\varphi}_z = \sqrt{\lambda_2\left(\mathsf{J}_\Delta^\top \mathsf{J}_\Delta \left(\gamma^{\text{WP}}\right)^{-1}\right)} \quad \text{and} \quad \bar{\kappa} = \pm\sqrt{\lambda_1(\Upsilon^{\text{WP}})}\mathbf{v}_1(\Upsilon^{\text{WP}}), \tag{4.25}$$

where:

$$\Upsilon^{\text{WP}} = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta - \lambda_2\left(\mathsf{J}_\Delta^\top \mathsf{J}_\Delta \left(\gamma^{\text{WP}}\right)^{-1}\right)\gamma^{\text{WP}}.$$

4.2.1.3 Obtaining the embedding

(Bartoli and Collins, 2013; Bartoli et al., 2015) use $\bar{\varphi}_z$ directly to get the embedding φ through equation (4.7) or (4.9), neglecting the information contained in $\bar{\beta}$ and consequently $\bar{\kappa}$. At first glance this direct-depth method seems to be sensible as $\bar{\beta}$ is known only up to sign and requires integration to recover depth. We show in section 4.2.2 however, that the depth solution $\bar{\varphi}_z$ is not well-constrained, unlike the depth-gradient. We give a method of obtaining a better 3D reconstruction using $\bar{\beta}$ in section 4.2.3.

4.2.2 Stability

We prove two important results regarding the stability of the type-I non-holonomic solutions of PDEs (4.16) and (4.23). We list them as propositions below.

Proposition 1. *The non-holonomic solution for depth $\bar{\varphi}_z$ is weakly constrained when the projection geometry tends to affine.*

Proposition 2. *The non-holonomic solution for the depth-gradient $\bar{\kappa}$ is well-constrained in all projection geometries.*

Figure 4.3 gives a general diagram showing the effect of different projection geometries on SfT. We prove these results for the perspective and infinitesimal weak-perspective cameras.

We define a projection function $\mathit{\Pi}_s$ on a 3D point $\mathbf{Q} = [Q_x\ Q_y\ Q_z]^\top$, depending on a parameter s that allows us to continuously select the amount of perspective:

$$\mathit{\Pi}_s(\mathbf{Q}) = \frac{(s+1)\,f}{Q_z + sf} \begin{bmatrix} Q_x & Q_y \end{bmatrix}^\top . \tag{4.26}$$

We obtain a perspective projection with the focal length f when $s = 0$, and an orthographic projection[1] when $s \to \infty$:

$$\lim_{s\to\infty} \mathit{\Pi}_s(\mathbf{Q}) = \begin{bmatrix} Q_x & Q_y \end{bmatrix}^\top . \tag{4.27}$$

The infinitesimal weak-perspective approximation of $\mathit{\Pi}_s$ is:

$$\mathit{\Pi}_s^{\mathrm{WP}}(\mathbf{Q}) = \frac{(s+1)\,f}{\zeta + sf} \begin{bmatrix} Q_x & Q_y \end{bmatrix}^\top . \tag{4.28}$$

Proof of propositions 1 and 2 for the perspective camera. We first substitute the projection model $\mathit{\Pi}_s$ into the PDE system (4.14) by simply redefining the back-projection matrix $\mathit{\Phi}$ for perspective projection as:

$$\mathit{\Phi}_s = \mathrm{diag}\left(\frac{\varphi_z + sf}{(s+1)f}, \frac{\varphi_z + sf}{(s+1)f}, \varphi_z + sf \right) . \tag{4.29}$$

Introducing $\mathit{\Phi}_s$ in the type-I PDE system (4.14) we obtain:

$$\begin{aligned} &\left(1 + \frac{\eta^\top \eta}{((s+1)f)^2}\right) \mathsf{J}_{\varphi_z}^\top \mathsf{J}_{\varphi_z} + \frac{\varphi_z + sf}{((s+1)f)^2} (\mathsf{J}_{\varphi_z}^\top \eta^\top \mathsf{J}_\eta) + \frac{\varphi_z + sf}{((s+1)f)^2} (\mathsf{J}_\eta^\top \eta \mathsf{J}_{\varphi_z}) + \\ &\frac{(\varphi_z + sf)^2}{((s+1)f)^2} \mathsf{J}_\eta^\top \mathsf{J}_\eta = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta . \end{aligned} \tag{4.30}$$

We first prove proposition 1. By taking the limit $s \to \infty$ in equation (4.30) we find the following system:

$$\mathsf{J}_{\varphi_z}^\top \mathsf{J}_{\varphi_z} + \mathsf{J}_\eta^\top \mathsf{J}_\eta = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta , \tag{4.31}$$

which represents the general PDE system for affine projection (Pizarro et al., 2013). In equation (4.31) the depth variable φ_z vanishes, which means that in affine projection depth is not any more constrained. This can be also proved in the space of solutions after the change of variable with equation (4.15). $\bar{\alpha}$ depends on the eigenvalues of matrix $\mathsf{J}_\Delta^\top \mathsf{J}_\Delta \gamma^{-1}$. When s is a large number, the

[1] The orthographic projection and affine projection are equivalent up to an affine image transform. Thus the discussion is still valid for any affine projection although described for an orthographic projection.

solution of $\bar{\alpha}$ in equation (4.20) is ill-conditioned. We write γ from equation (4.17) as a function of s:

$$\gamma_s = \frac{1}{\nu_s^2\left((s+1)f\right)^2}\left(\mathsf{J}_\eta^\top \mathsf{J}_\eta - \frac{1}{\nu_s^2\left((s+1)f\right)^4}\mathsf{J}_\eta^\top \eta\eta^\top \mathsf{J}_\eta\right), \tag{4.32}$$

with:

$$\nu_s = \sqrt{1 + \frac{\eta^\top \eta}{((s+1)f)^2}}. \tag{4.33}$$

Taking the limit of equation (4.32) we find $\lim_{s\to\infty}\gamma_s = \mathbf{0}_{2\times 2}$. $\bar{\alpha}$ is then computed from a matrix whose elements tend to infinity.

As for proposition 2, although φ_z has already vanished from equation (4.31), the solution to its gradient can still be recovered by applying the rank-1 constraint to equation (4.31). Thus $\bar{\kappa}$ is simply given by:

$$\bar{\kappa} = \pm\lambda_1\left(\mathsf{J}_\Delta^\top \mathsf{J}_\Delta - \mathsf{J}_\eta^\top \mathsf{J}_\eta\right)\mathbf{v}_1\left(\mathsf{J}_\Delta^\top \mathsf{J}_\Delta - \mathsf{J}_\eta^\top \mathsf{J}_\eta\right). \tag{4.34}$$

which means that depth-gradient is equally well constrained with affine projection. □

Proof of propositions 1 and 2 for the infinitesimal weak-perspective camera. By taking the infinitesimal weak-perspective approximation of the projection model $\mathit{\Pi}_s^{\mathrm{WP}}$ and plugging it into equation (4.23) after setting $\zeta = \varphi_z$ we reach the following system:

$$\mathsf{J}_{\varphi_z}^\top \mathsf{J}_{\varphi_z} + \left(\frac{\varphi_z + sf}{(s+1)f}\right)^2 \mathsf{J}_\eta^\top \mathsf{J}_\eta = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta. \tag{4.35}$$

Again by taking the limit $s \to \infty$ on both sides of equation (4.35), we reach system (4.31) of affine projection. If there is no perspective effect and the camera is affine we cannot compute the average depth of the scene as it vanishes from the equations. The proof of proposition 1 follows in the same way as in the perspective camera by using $\gamma_s = \frac{1}{((s+1)f)^2}\mathsf{J}_\eta^\top \mathsf{J}_\eta$. For proposition 2 the solution of $\bar{\kappa}$ in equation (4.35) is identical to that for the perspective model when $s \to \infty$. □

4.2.3 Stable type-I methods

In sections 4.2.1 and 4.2.2 we revisited the non-holonomic type-I solutions and proved that the depth solution was unstable. However obtaining the embedding using the stable solution is not straightforward mainly due to the two-fold ambiguity of the solution. Furthermore, an integration step is also required before we can obtain depth from the disambiguated solution. We give the details of the proposed stable type-I method as follows. We propose to use $\bar{\beta}$ which is stable in perspective and affine conditions to solve SfT. In order to obtain depth $\hat{\varphi}_z$ from $\bar{\beta}$ we need to go through the following four steps: *i)* sign disambiguation for $\bar{\beta}$, *ii)* numerical integration of $\bar{\beta}$, *iii)* arbitrary integration constant computation and *iv)* variable change, for the perspective camera. For the infinitesimal weak-perspective camera the last step is not required and we use $\bar{\phi}_z, \bar{\kappa}$ instead of $\bar{\alpha}, \bar{\beta}$.

Algorithm 1: Stable type-I method for the perspective camera.

```
Input: warp $\eta$, template embedding $\Delta$, domain $\Omega$
Output: deformed embedding $\hat{\varphi}$
• PDE solution
    Compute the Jacobians $\mathsf{J}_\eta$ and $\mathsf{J}_\Delta$
    Solve the PDE system (4.16) to obtain $\bar{\alpha}$ and $\pm\bar{\beta}$
• Sign disambiguation
    Compute the Jacobian $\mathsf{J}_{\bar{\alpha}}$ for the solution $\bar{\alpha}$
    Select the sign for $\bar{\beta}$ so that the largest component of $\bar{\beta}$ has the same sign as that
    component of $\mathsf{J}_{\bar{\alpha}}$
    Compute the two absolute angles between the vectors $\mathsf{J}_{\bar{\alpha}}$ and $\bar{\beta}$
    Discard $\bar{\beta}$ in points where the best angle is greater than a threshold (sum of the mean
    angle for all points and two times the standard deviation)
    Interpolate to compute $\bar{\beta}$ for the discarded values
• Numerical integration
    Integrate the disambiguated value of $\bar{\beta}$ to obtain $\hat{\alpha} + k_\alpha$
• Integration constant
    Compute the integration constant:

    $k_\alpha = \underset{\Omega}{\text{median}}\left((\hat{\alpha} + k_\alpha) - \bar{\alpha}\right)$
• Change of variable
    Apply the change of variable $\hat{\varphi}_z = \frac{\hat{\alpha}}{\nu}$ and compute $\hat{\varphi}$ using the perspective camera
    model
```

4.2.3.1 Sign disambiguation

According to equation (4.21), the non-holonomic solutions $\bar{\beta}$ and $\bar{\kappa}$ are known up to a local sign change. (Bartoli et al., 2015; Pizarro et al., 2013) mention a few ways to disambiguate the sign for different but related problems, based on external cues, such as shading, temporal smoothing, or surface smoothing. We show below that we can do without these additional cues, which may be unavailable or even unstable in practice.

If there is some perspective, even very loose, we know that a non-holonomic solution for φ_z exists. We thus propose to disambiguate the sign of $\bar{\beta}$ or $\bar{\kappa}$ by using the non-holonomic solution to depth $\bar{\varphi}_z$. In the perspective camera the process has four steps: **1**) We first differentiate $\bar{\alpha}$ to obtain $\mathsf{J}_{\bar{\alpha}}$. **2**) We select the sign of $\bar{\beta}$ so that the resulting vector is closest to $\mathsf{J}_{\bar{\alpha}}$. **3**) We discard the computed $\bar{\beta}$ at regions of the template where $\mathsf{J}_{\bar{\alpha}}$ differs substantially from $\bar{\beta}$. This can occur due to the instability of the depth solution. We use the angle between the two vectors as a metric:

$$\measuredangle(\mathbf{p}) = \left|\text{acos}\left(\frac{\mathsf{J}_{\bar{\alpha}}\bar{\beta}}{\|\mathsf{J}_{\bar{\alpha}}\|\|\bar{\beta}\|}\right)\right|. \tag{4.36}$$

The above computed angle is simply an angle between vectors and does not have a physical significance. It is only used as a metric to choose among the solutions $\pm\bar{\beta}$. We use the sum of the mean angle for all points and twice the standard deviation as the threshold. **4**) We use smoothing to compute

Algorithm 2: Stable type-I method for the infinitesimal weak-perspective camera.

```
Input: warp $\eta$, template embedding $\mathit{\Delta}$, domain $\mathit{\Omega}$
Output: deformed embedding $\hat{\varphi}$
• PDE solution
    Compute the Jacobians $\mathsf{J}_\eta$ and $\mathsf{J}_{\mathit{\Delta}}$
    Solve the PDE system (4.23) to obtain $\bar{\varphi}_z$ and $\pm\bar{\kappa}$
• Sign disambiguation
    Compute the Jacobian $\mathsf{J}_{\bar{\varphi}_z}$ for the solution $\bar{\varphi}_z$
    Select the sign for $\bar{\kappa}$ so that the largest component of $\bar{\kappa}$ has the same sign as that
        component of $\mathsf{J}_{\bar{\varphi}_z}$
    Compute the two absolute angles between the vectors $\mathsf{J}_{\bar{\varphi}_z}$ and $\pm\bar{\kappa}$
    Discard $\bar{\kappa}$ in points where the best angle is greater than a threshold (sum of the mean
        angle for all points and two times the standard deviation)
    Interpolate to compute $\bar{\kappa}$ for the discarded values
• Numerical integration
    Integrate the disambiguated value of $\bar{\kappa}$ to obtain $\hat{\varphi}_z + k_z$
• Integration constant
    Compute the integration constant:
        $k_z = \underset{\mathit{\Omega}}{\operatorname{median}}\left((\hat{\varphi}_z + k_z) - \bar{\varphi}_z\right)$
    Find the embedding $\hat{\varphi}$ using the perspective camera model
```

values for $\bar{\beta}$ for the regions where they were discarded.

4.2.3.2 Numerical integration

The non-holonomic solution $\bar{\beta}$ is not guaranteed to be integrable. We thus need a numerical integration method to estimate $\hat{\varphi}_z$. We propose to use a parametric function represented by a Bicubic B-Spline (BBS). With a BBS, or any other linear basis expansion model, we can integrate $\bar{\beta}$ by means of sparse linear least-squares. The solution is defined up to an additive integration constant. We define the LLS integration as:

$$\hat{\alpha} + k_\alpha = \underset{\alpha_s}{\arg\min} \int_{\mathit{\Omega}} \|\mathsf{J}_{\alpha_s} - \bar{\beta}\|^2 \, d\mathbf{p}. \tag{4.37}$$

We evaluate the integral by using a summation over a dense grid of points $\mathbf{p}$ on the flat template space $\mathit{\Omega}$.

4.2.3.3 Integration constant

After integration we obtain $\hat{\alpha} + k_\alpha$, where k_α is an arbitrary integration constant. We propose to use $\bar{\alpha}$ to estimate k_α. First we take samples from $\hat{\alpha} + k_\alpha$ and $\bar{\alpha}$. We then obtain k_α by using the median of the differences between the samples. This is expressed as:

$$k_\alpha = \underset{\mathit{\Omega}}{\operatorname{median}}\left((\hat{\alpha} + k_\alpha) - \bar{\alpha}\right). \tag{4.38}$$

4.2.3.4 Change of variable

We apply the change of variable given by equation (4.15) on the estimate of $\hat{\alpha}$ obtained above. This gives us the depth estimate at each point and thus the final surface embedding is obtained by estimating the surface point coordinates using the perspective camera model. Algorithm 1 summarizes the steps for the stable type-I method for the perspective camera.

For the infinitesimal weak-perspective camera, all the steps are very similar except that we directly obtain $\bar{\varphi}_z$ and $\bar{\kappa}$ as the non-holonomic solutions instead of $\bar{\alpha}$ and $\bar{\beta}$ respectively. This means that while the rest of the steps remain the same, the final depth and shape are obtained without the change of variable. For clarity, we describe these steps separately for the infinitesimal weak-perspective camera in algorithm 2.

4.3 Type-II Solutions, Stability and Type-II Stable Method

The type-I solutions discussed in section 4.2 involve a change of variable of the depth and its derivatives where the non-holonomic solutions are measured in a modified space for the perspective camera. We present a new interpretation of SfT using the type-II solutions and type-II stable methods for two important reasons. First type-II PDE for SfT has not been studied before; it describes explicitly how tangent planes on the embedding are related by rigid transforms from the template to the deformed surface. Additionally, as we show in the experimental results, it can lead to slightly different results from the stable type-I method. This is primarily due to the use of different spaces for numerical integration, which is further explained in chapter 4.5 and Appendix B. In this section we propose non-holonomic solutions of the general PDE (4.12) involving depth and the surface normal, which provide a more intuitive geometrical interpretation despite being equivalent to type-I solutions up to a change of variable.

4.3.1 Type-II solutions

To use the PDE system proposed in (Collins and Bartoli, 2014a) for SfT, we need to ensure that the template shape is locally planar with normals pointing towards the positive depth axis. Such case is easily realized for a flat template. For a general 3D template, we make use of a locally isometric flattening operator, which provides a new flat parametrization space that is related to the 3D template by a locally isometric map. We then exploit the fact that isometric deformations induce a rigid transformation between the tangent plane in the template and the one in the surface embedding. Thus, from the deformation constraint (4.4), we have that the embedding's Jacobian is a Stiefel matrix. We use this property to find the non-holonomic solutions. For general 3D templates, the flattening is not necessarily an isometric map. Therefore we first show that one can always change the parametrization space so that locally the template parametrization (flattening) is isometric. We call the new parametrization space as the locally isometric flattening. This change involves computing the Cholesky factorization of $\mathsf{J}_{\Delta}^{\top}\mathsf{J}_{\Delta}$ at every point. We first present the locally isometric flattening and then describe the new set of non-holonomic solutions for the depth and the surface normal.

4.3.1.1 Locally isometric flattening

For a generic flattening with parametrization $\mathit{\Delta}$, the columns of the embedding's Jacobian J_φ are in general not orthonormal. However we prove here that we can still get a local embedding using a different parametrization space such that the embedding's Jacobian will have orthonormal columns. We consider the required local embedding as $\phi_{\mathbf{p}'} \in C^1(\mathit{\Omega}', \mathbb{R}^3)$ parametrized with a new flat template space $\mathit{\Omega}'$ such that the new flat template is locally isometric to the 3D template $\mathcal{T}$. This leads us to the following equation:

$$\mathsf{J}_{\phi_{\mathbf{p}'}}^\top \mathsf{J}_{\phi_{\mathbf{p}'}} = \mathsf{J}_{\mathit{\Delta}'_{\mathbf{p}'}}^\top \mathsf{J}_{\mathit{\Delta}'_{\mathbf{p}'}} = \mathbf{I}_2 \tag{4.39}$$

where $\mathit{\Delta}'_{\mathbf{p}'} \in C^1(\mathit{\Omega}', \mathbb{R}^3)$ is a locally isometric parametrization that uses the new flat template and maps each point on $\mathit{\Omega}'$ isometrically to the 3D template $\mathcal{T}$. Here the domain $\mathit{\Omega}'$ of $\mathit{\Delta}'_{\mathbf{p}'}$ in general is not a connected space. We also map each point on the original flat template space $\mathit{\Omega}$ to the new flat template space $\mathit{\Omega}'$ using a local function $\rho_{\mathbf{p}} \in C^1(\mathit{\Omega}, \mathbb{R}^2)$. As it is a one to one mapping between subsets of $\mathbb{R}^2$, its inverse $\rho_{\mathbf{p}'}^{-1} \in C^1(\mathit{\Omega}', \mathbb{R}^2)$ is well defined. We show the complete parametrization in figure 4.2. In the following discussion, we show that such a template can be constructed for any surface embedded in $\mathbb{R}^3$ in a disc topology. To simplify the notations, we drop the subscripts $\mathbf{p}$ and $\mathbf{p}'$ from the local functions.

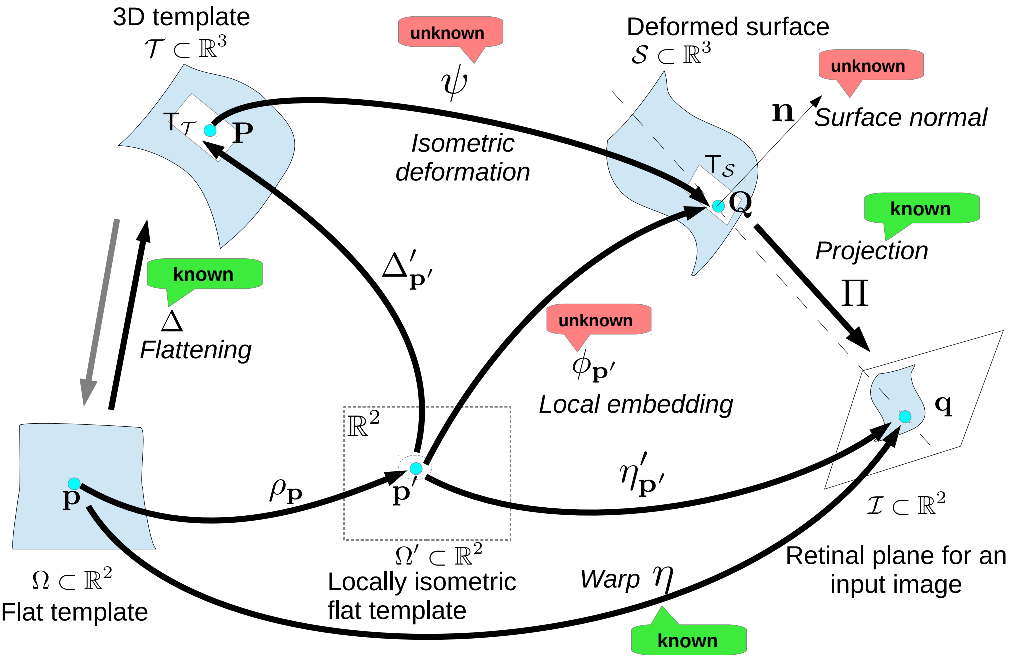


Figure 4.2: Differential geometric modelling of Shape-from-Template with the locally isometric flattening. The new space depicted in the middle is locally isometric to the 3D template and the deformed surface. This property is required to construct the type-II SfT PDE.

The only new restriction that needs to be imposed for the new flattening is equation (4.39). $\mathit{\Delta}'$ can be expressed in terms of $\mathit{\Delta}$ and ρ^{-1} as $\mathit{\Delta}' = \mathit{\Delta} \circ \rho^{-1}$. This gives us the first-order relation:

$$\mathsf{J}_{\mathit{\Delta}'} = (\mathsf{J}_\mathit{\Delta} \circ \rho^{-1}) \mathsf{J}_{\rho^{-1}}, \tag{4.40}$$

where $\mathsf{J}_\Delta \circ \rho^{-1}$ is the Jacobian of the known parametrization function of the 3D template evaluated in Ω. Combining equations (4.39) and (4.40) we obtain:

$$\mathsf{J}_{\rho^{-1}}^{-\top}\mathsf{J}_{\rho^{-1}}^{-1} = (\mathsf{J}_\Delta \circ \rho^{-1})^\top (\mathsf{J}_\Delta \circ \rho^{-1}). \tag{4.41}$$

Our goal here is to find the Jacobian of the new template-to-image warp $\mathsf{J}_{\eta'}$ and for that purpose it suffices for us to compute the Jacobian $\mathsf{J}_{\rho^{-1}}$. Any ρ^{-1} that satisfies equation (4.41) will lead to a $\mathsf{J}_{\Delta'}$ which will in turn satisfy equation (4.39). As equation (4.39) is the only requirement for the locally isometric template, finding $\mathsf{J}_{\rho^{-1}}$ is equivalent to obtaining the new parametrization. To obtain a value for $\mathsf{J}_{\rho^{-1}}$ that is consistent with equation (4.41) we perform the Cholesky decomposition of the right-hand side of equation (4.41) which is a symmetric positive definite matrix. This gives us:

$$\text{Chol}\left(\left(\mathsf{J}_\Delta \circ \rho^{-1}\right)^\top \left(\mathsf{J}_\Delta \circ \rho^{-1}\right)\right) = \chi\chi^\top$$

where $\chi \in C^0(\Omega', GL_2)$ is a lower triangular matrix-valued function. A value for $\mathsf{J}_{\rho^{-1}}$ that satisfies equation (4.41) is:

$$\mathsf{J}_{\rho^{-1}} = \chi^{-\top}. \tag{4.42}$$

There is in fact a class of matrices that differ by a single rotation, which satisfy equation (4.41). For all purposes, the value of the Jacobian $\mathsf{J}_{\rho^{-1}}$ given by equation (4.42) corresponds to a valid ρ^{-1}. We proceed further by considering the new flat template-to-image warp as $\eta' : \Omega' \to \mathbb{R}^2$. η' is related to the known flat template-to-image warp η as:

$$\eta' = \eta \circ \rho^{-1}. \tag{4.43}$$

Thus the Jacobian of the new template-to-image warp η' is obtained as:

$$\mathsf{J}_{\eta'} = \left(\mathsf{J}_\eta \circ \rho^{-1}\right) \mathsf{J}_{\rho^{-1}}. \tag{4.44}$$

4.3.1.2 SfT PDE with locally isometric flattening

After changing the parametrization space to a locally isometric flattening, we are now ready to give the analytic solutions for the depth ϕ_z and the surface normal $\mathbf{n} \in C^0(\Omega', \mathbb{R}^3)$ for each point on $\mathcal{S}$. Note that the new flat template space implies no changes in the function values of the embedding or the template-to-image warp but their derivatives are however different from those of the original functions. Thus we have $\eta(\mathbf{p}) = \eta'(\mathbf{p}')$ and $\varphi(\mathbf{p}) = \phi(\mathbf{p}')$ but $\mathsf{J}_{\eta'} \neq \mathsf{J}_\eta$ and $\mathsf{J}_\phi \neq \mathsf{J}_\varphi$ in general. The new deformation constraint is given as:

$$\mathsf{J}_\phi^\top \mathsf{J}_\phi = \mathbf{I}_2. \tag{4.45}$$

Considering the change in the parametrization space, the new reprojection constraint in the PDE system (4.12) becomes:

$$\Pi \circ \phi = \eta'. \tag{4.46}$$

Differentiating equation (4.46) we obtain:

$$(\mathsf{J}_{\Pi} \circ \phi)\, \mathsf{J}_{\phi} = \mathsf{J}_{\eta'}. \tag{4.47}$$

We write the generalized type-II PDE system for SfT by combining equations (4.45) and (4.47) as:

$$\text{Find } \phi_z,\ \mathsf{J}_{\phi} \text{ s.t.} \quad \begin{cases} (\mathsf{J}_{\Pi} \circ \phi)\, \mathsf{J}_{\phi} = \mathsf{J}_{\eta'} \\ \mathsf{J}_{\phi}^{\top} \mathsf{J}_{\phi} = \mathbf{I}_2. \end{cases} \tag{4.48}$$

We select perspective or infinitesimal weak-perspective systems by substituting the value of $\mathsf{J}_{\Pi} \circ \phi$ accordingly in the PDE system (4.48).

4.3.1.3 Perspective camera

We parametrize the surface 3D points on $\mathcal{S}$ using the depth function ϕ_z as $\phi = \phi_z[\frac{\eta'^{\top}}{f}\ 1]^{\top}$. This allows us to expand the Jacobian of the projection matrix evaluated on the surface as:

$$\mathsf{J}_{\Pi} \circ \phi = \frac{f}{\phi_z} \begin{bmatrix} \mathbf{I}_2 & -\frac{\eta'}{f} \end{bmatrix}. \tag{4.49}$$

Combining equations (4.47) and (4.49) we obtain the following reprojection constraint:

$$\frac{f}{\phi_z} \begin{bmatrix} \mathbf{I}_2 & -\frac{\eta'}{f} \end{bmatrix} \mathsf{J}_{\phi} = \mathsf{J}_{\eta'}. \tag{4.50}$$

With equation (4.50) we can rewrite our new PDE system for SfT as:

$$\text{Find } \phi_z,\ \mathsf{J}_{\phi} \text{ s.t.} \quad \begin{cases} \frac{f}{\phi_z} \begin{bmatrix} \mathbf{I}_2 & -\frac{\eta'}{f} \end{bmatrix} \mathsf{J}_{\phi} = \mathsf{J}_{\eta'} \\ \mathsf{J}_{\phi}^{\top} \mathsf{J}_{\phi} = \mathbf{I}_2. \end{cases} \tag{4.51}$$

We denote the embedding's Jacobian J_{ϕ} as τ_{32} where $\tau \in C^0(\Omega', SO_3)$ is a function giving a rotation matrix. The third column of τ gives the solution for the surface normal $\bar{\mathbf{n}}$ and the first two columns form the embedding's Jacobian J_{ϕ}. If we obtain J_{ϕ} and thus the first two columns of τ, the surface normal $\bar{\mathbf{n}}$ can be retrieved from the cross-product of the two columns of τ_{32}. We give the steps for computing the non-holonomic solutions of system (4.51) below.

We first compute a rotation function $\theta \in C^0(\Omega', SO_3)$ such that:

$$\begin{bmatrix} \mathbf{I}_2 & -\frac{\eta'}{f} \end{bmatrix} \theta = \begin{bmatrix} \omega & \mathbf{0} \end{bmatrix} \tag{4.52}$$

where $\omega \in C^0(\Omega', \mathbb{R}^{2\times 2})$ is a matrix-valued function. θ and ω can be computed directly from equation (4.52). The actual steps are provided in Appendix A. Next we multiply the left-hand side of equation (4.50) by $\theta\theta^{\top}$ to obtain:

$$\frac{f}{\phi_z} \begin{bmatrix} \omega & \mathbf{0} \end{bmatrix} \theta^{\top} \tau_{32} = \mathsf{J}_{\eta'}. \tag{4.53}$$

We simplify equation (4.53) by introducing another rotation function $\xi \in C^0(\Omega', SO_3)$ such that $\xi_{32} = \theta^\top \tau_{32}$. This gives us the following new equation with the unknown depth ϕ_z and the unknown sub-Stiefel matrix ξ_{22}:

$$\frac{f}{\phi_z}\xi_{22} = \omega^{-1}\mathsf{J}_{\eta'}. \tag{4.54}$$

To solve equation (4.54) we use the fact that $\xi_{22} \in SS_{2\times 2}$ is a sub-Stiefel matrix and thus its largest singular value is 1. Equating the largest singular values for both sides of equation (4.54), we obtain:

$$\phi_z = \frac{1}{f}\sigma_1^{-1}\left(\omega^{-1}\mathsf{J}_{\eta'}\right) \tag{4.55}$$

where $\sigma_1(\mathbf{M})$ is an operator giving the largest singular value of the matrix $\mathbf{M}$. Similarly ξ_{22} can be recovered from the relation $\xi_{22} = \frac{1}{f}\phi_z \omega^{-1}\mathsf{J}_{\eta'}$.

We parametrize the Stiefel matrix ξ_{32} by using the computed sub-Stiefel part ξ_{22} and an unknown vector $\mathbf{r}$ so that $\xi_{32}^\top = [\xi_{22}^\top\ \mathbf{r}]$. Using the orthogonality of the Stiefel matrix we have:

$$\mathbf{I}_2 - \xi_{22}^\top \xi_{22} = \mathbf{r}\mathbf{r}^\top. \tag{4.56}$$

Since $\mathbf{r}$ is a vector, both sides of equation (4.56) have rank 1. Consequently, $\mathbf{r}$ can be obtained by taking the SVD of the left-hand side of equation (4.56) and choosing the right singular vector corresponding to its non-zero singular value as below:

$$\mathbf{r} = \pm \mathbf{v}_1(\mathbf{I}_2 - \xi_{22}^\top \xi_{22}). \tag{4.57}$$

However, this results in two solutions for ξ_{32}; we write them as: ξ_{32}^a and ξ_{32}^b. The two solutions for the embedding's Jacobian are now simply given by:

$$\mathsf{J}_\phi = \tau_{32} = \begin{cases} \theta\xi_{32}^a \\ \theta\xi_{32}^b. \end{cases} \tag{4.58}$$

Finally we can also obtain the two solutions for the surface normal $\bar{\mathbf{n}}_1$ and $\bar{\mathbf{n}}_2$ from the cross product of the columns of the two solutions for τ_{32}.

4.3.1.4 Infinitesimal weak-perspective camera

The infinitesimal weak-perspective camera model can be used in place of the perspective model by parametrizing the 3D image points with the infinitesimal weak-perspective depth function ζ and the template-to-image warp in homogeneous coordinates $\tilde{\eta}' = [\eta'^\top\ 1]^\top$:

$$\phi = \frac{\zeta}{f}\tilde{\eta}'. \tag{4.59}$$

Here ζ varies over the template while its gradient $\mathsf{J}_\zeta = \mathbf{0}$. This gives us the Jacobian of the projection matrix evaluated on the surface as:

$$\mathsf{J}_\Pi \circ \phi = \frac{f}{\zeta}\begin{bmatrix}\mathbf{I}_2 & \mathbf{0}\end{bmatrix}. \tag{4.60}$$

Allowing $\zeta = \phi_z$, *i.e.* to be different for each point while keeping the Jacobian of the projection matrix as in equation (4.60) gives us the infinitesimal weak-perspective system. Thus combining equations (4.47) and (4.60) we obtain the following reprojection constraint for the infinitesimal weak-perspective model:

$$\frac{f}{\phi_z}\begin{bmatrix}\mathbf{I}_2 & \mathbf{0}\end{bmatrix}\mathsf{J}_\phi = \mathsf{J}_{\eta'}. \tag{4.61}$$

As in the perspective solutions we form the PDE system by considering J_ϕ as a Stiefel matrix. Thus the problem becomes:

$$\text{Find } \phi_z,\ \mathsf{J}_\phi \text{ s.t. } \frac{f}{\phi_z}\begin{bmatrix}\mathbf{I}_2 & \mathbf{0}\end{bmatrix}\tau_{32} = \mathsf{J}_{\eta'}, \tag{4.62}$$

where $\tau \in C^1(\Omega', SO_3)$ is a rotation matrix such that $\mathsf{J}_\phi = \tau_{32}$. Simplifying problem (4.62) leads to the following:

$$\frac{f}{\phi_z}\tau_{22} = \mathsf{J}_{\eta'}. \tag{4.63}$$

Equation (4.63) is identical to the perspective system described by equation (4.54) except that we do not have a rotation θ involved in the infinitesimal weak-perspective system and thus the equation is much simpler. As we are allowing ϕ_z to change at each point and solving for the pose and scale simultaneously the solutions are infinitesimal weak-perspective solutions. We obtain ϕ_z as:

$$\phi_z = \frac{1}{f}\sigma_1^{-1}\left(\mathsf{J}_{\eta'}\right). \tag{4.64}$$

Similarly τ_{22} can be obtained as:

$$\tau_{22} = \frac{1}{f}\phi_z \mathsf{J}_{\eta'}. \tag{4.65}$$

We recover the two solutions for τ_{32} in the same way as we did for ξ_{32} for the perspective camera and thus obtain the non-holonomic solutions for the surface normal using the cross product.

4.3.1.5 Obtaining the embedding

Solving the generalized type-II SfT PDE system (4.48) gives us the set of non-holonomic solutions: $\bar{\phi}_z$ and $\bar{\mathbf{n}}_{1,2}$. One obvious way to obtain the embedding ϕ is to use the direct depth solution $\bar{\phi}_z$ and the normalized points obtained from η as: $\bar{\phi} = \bar{\phi}_z[\frac{\eta'^\top}{f}\ 1]^\top$. This solution is identical to the direct-depth method proposed in (Bartoli et al., 2015) as both of them are the non-holonomic solutions for the depth. However in doing so, we essentially discard the solution for the Jacobian or the normal and make use of only one non-holonomic solution *i.e.* the depth ϕ_z. In section 4.3.2 we describe how the solution for the Jacobian $\bar{\tau}_{32}$ and thus the normal $\bar{\mathbf{n}}$ is better constrained than the depth solution $\bar{\phi}_z$. We provide the actual algorithm for obtaining a stable 3D reconstruction with the stable type-II method in section 4.3.3.

4.3.2 Stability

We show here that the amount of perspective affects how depth is constrained in the type-II solutions in the same way as in the type-I solutions. We prove two important results for the type-II solutions.

Proposition 3. *The non-holonomic solution for depth $\bar{\phi}_z$ is weakly constrained when the projection geometry tends to affine.*

Proposition 4. *The non-holonomic solution for the embedding's Jacobian and thus the surface normal $\bar{\mathbf{n}}$ is well-constrained in all projection geometries.*

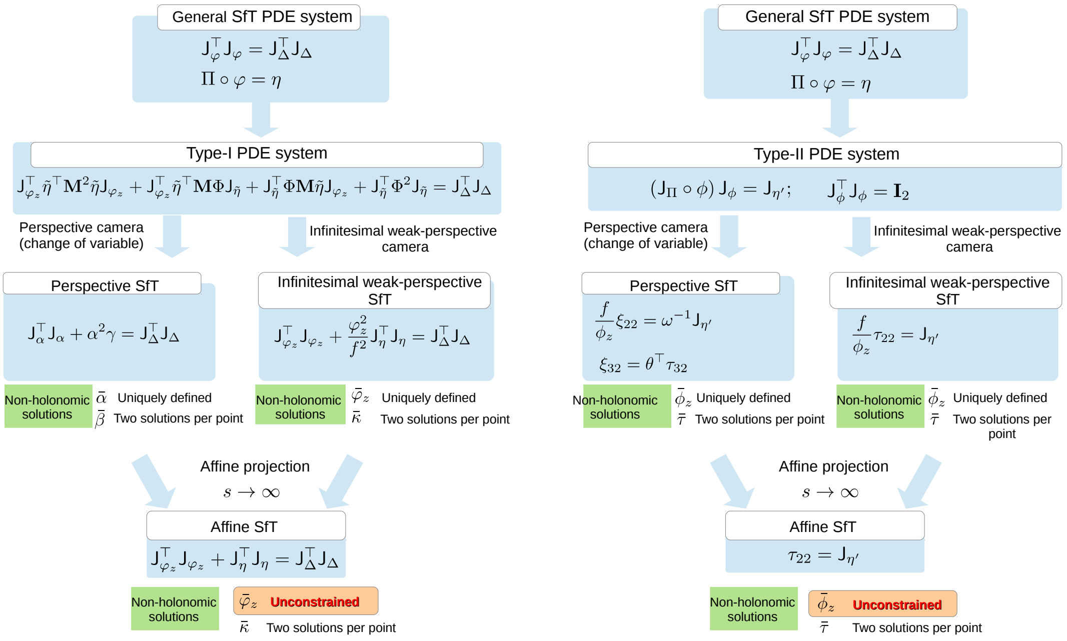

General SfT PDE system
$\mathsf{J}_\varphi^\top \mathsf{J}_\varphi = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta$
$\Pi \circ \varphi = \eta$
Type-I PDE system
$\mathsf{J}_{\varphi_z}^\top \tilde{\eta}^\top \mathbf{M}^2 \tilde{\eta} \mathsf{J}_{\varphi_z} + \mathsf{J}_{\varphi_z}^\top \tilde{\eta}^\top \mathbf{M} \Phi \mathsf{J}_{\tilde{\eta}} + \mathsf{J}_{\tilde{\eta}}^\top \Phi \mathbf{M} \tilde{\eta} \mathsf{J}_{\varphi_z} + \mathsf{J}_{\tilde{\eta}}^\top \Phi^2 \mathsf{J}_{\tilde{\eta}} = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta$
Perspective camera (change of variable)
Infinitesimal weak-perspective camera
Perspective SfT
$\mathsf{J}_\alpha^\top \mathsf{J}_\alpha + \alpha^2 \gamma = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta$
Non-holonomic solutions
$\bar{\alpha}$ Uniquely defined
$\bar{\beta}$ Two solutions per point
Infinitesimal weak-perspective SfT
$\mathsf{J}_{\varphi_z}^\top \mathsf{J}_{\varphi_z} + \frac{\varphi_z^2}{f^2} \mathsf{J}_\eta^\top \mathsf{J}_\eta = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta$
Non-holonomic solutions
$\bar{\varphi}_z$ Uniquely defined
$\bar{\kappa}$ Two solutions per point
Affine projection
$s \to \infty$
Affine SfT
$\mathsf{J}_{\varphi_z}^\top \mathsf{J}_{\varphi_z} + \mathsf{J}_\eta^\top \mathsf{J}_\eta = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta$
Non-holonomic solutions
$\bar{\varphi}_z$ Unconstrained
$\bar{\kappa}$ Two solutions per point
General SfT PDE system
$\mathsf{J}_\varphi^\top \mathsf{J}_\varphi = \mathsf{J}_\Delta^\top \mathsf{J}_\Delta$
$\Pi \circ \varphi = \eta$
Type-II PDE system
$(\mathsf{J}_\Pi \circ \phi)\, \mathsf{J}_\phi = \mathsf{J}_{\eta'}; \quad \mathsf{J}_\phi^\top \mathsf{J}_\phi = \mathbf{I}_2$
Perspective camera (change of variable)
Infinitesimal weak-perspective camera
Perspective SfT
$\frac{f}{\phi_z} \xi_{22} = \omega^{-1} \mathsf{J}_{\eta'}$
$\xi_{32} = \theta^\top \tau_{32}$
Non-holonomic solutions
$\bar{\phi}_z$ Uniquely defined
$\bar{\tau}$ Two solutions per point
Infinitesimal weak-perspective SfT
$\frac{f}{\phi_z} \tau_{22} = \mathsf{J}_{\eta'}$
Non-holonomic solutions
$\bar{\phi}_z$ Uniquely defined
$\bar{\tau}$ Two solutions per point
Affine projection
$s \to \infty$
Affine SfT
$\tau_{22} = \mathsf{J}_{\eta'}$
Non-holonomic solutions
$\bar{\phi}_z$ Unconstrained
$\bar{\tau}$ Two solutions per point

Figure 4.3: SfT type-I solutions (left) and type-II solutions (right) for different projection models and amount of perspective.

Figure 4.3 illustrates the effect of projection geometries on the *type-I* and *type-II solutions* of SfT. As in section 4.2.2 we make use of the projection function $\mathit{\Pi}_s$ that depends on a parameter s and the initial focal length f to continuously select the amount of perspective. The only difference being that we now use the symbol ϕ instead of φ for the embedding.

Proof of propositions 3 and 4 for the perspective camera. We take the first-order derivatives of equation (4.26) on the surface, which gives us:

$$\mathsf{J}_{\mathit{\Pi}_s}(\mathbf{Q}) = f \frac{s+1}{Q_z + sf} \begin{bmatrix} \mathbf{I}_2 & \frac{-\eta'}{f(s+1)} \end{bmatrix}. \tag{4.66}$$

Combining equation (4.47) with equation (4.66), we obtain:

$$f \frac{s+1}{Q_z + sf} \begin{bmatrix} \mathbf{I}_2 & \frac{-\eta'}{f(s+1)} \end{bmatrix} \tau_{32} = \mathsf{J}_{\eta'}. \tag{4.67}$$

We first prove proposition 3. Evaluating the limit $s \to \infty$ for equation (4.67) gives us:

$$\begin{bmatrix} \mathbf{I}_2 & 0 \end{bmatrix} \tau_{32} = \mathsf{J}_{\eta'} \quad \Leftrightarrow \quad \tau_{22} = \mathsf{J}_{\eta'}. \tag{4.68}$$

The depth Q_z is now no longer constrained in equation (4.68) at the limit. This proves our first result that depth is not constrained for affine projection in SfT.

The proof of proposition 4 follows directly from the fact that τ_{22} is well-constrained in equation (4.68) despite the projection geometry. Following the steps in (Collins and Bartoli, 2014a) or in section 4.3.1, one can easily derive the two solutions for J_ϕ and thus the two solutions for the normal from equation (4.68). □

Proof of propositions 3 and 4 for the infinitesimal weak-perspective camera. Here we start with the infinitesimal weak-perspective approximation for the projection function given by equation (4.28). Taking the limit $s \to \infty$ in equation (4.28) leads to an expression identical to an expression identical to equation (4.68). The proof for propositions 3 and 4 then follows with the same arguments as in the perspective camera. □

4.3.3 Stable type-II methods

We presented the non-holonomic type-II solutions for the general SfT PDE system (4.12) and proved that the depth solution was unstable. We give the method for obtaining the final embedding using the stable solution as follows. The stable type-II methods use the solution to the surface normal directly to obtain the final embedding. We first obtain the surface normal by taking the cross product of the Jacobian columns and then to use Shape-from-Normals to obtain depth and the final embedding. The two steps are almost identical but having an analytic surface normal gives a better geometrical appeal to the problem. Furthermore, one can imagine several scenarios where the required end result from SfT would be a surface normal rather than the embedding itself. However, the non-holonomic solution to the embedding's Jacobian or the normal have a two-fold ambiguity. Therefore we solve the following problems in our type-II methods to obtain the 3D embeddding: *i)* normal disambiguation, *ii)* Shape-from-Normals and *iii)* scale computation. All of these steps are identical for the perspective and infinitesimal weak-perspective cameras. Figure 4.4 illustrates the specifics and similarities of stable type-I and stable type-II methods together.

4.3.3.1 Normal disambiguation

In type-II solutions we obtain two different non-holonomic solutions for the embedding's Jacobian $\varnothing_{32}$ and thus two different values for the normal vector: $\bar{\mathbf{n}}_1$ and $\bar{\mathbf{n}}_2$. To disambiguate the surface normal, we go through the following list of steps. **1**) We first compute a surface embedding $\bar{\phi}$ from the direct depth solution and a normal field $\mathbf{n}_d$ using the surface $\bar{\phi}$. **2**) For each point on the surface, we compute the following dot product with the normal vector $\mathbf{n}_d$: $|\mathbf{n}_d^\top \bar{\mathbf{n}}_1|$ and $|\mathbf{n}_d^\top \bar{\mathbf{n}}_2|$. **3**) We select the highest dot product among the two and choose the corresponding normal vector $\bar{\mathbf{n}}_1$ or $\bar{\mathbf{n}}_2$. This process can be expressed as: $\bar{\mathbf{n}} = \bar{\mathbf{n}}_k$, such that, $k = \arg\max_{k\in\{1,2\}} |\mathbf{n}_d^\top \bar{\mathbf{n}}_k|$. Unlike in the type-I methods we do not remove regions or points here, as it gives little or no improvement.

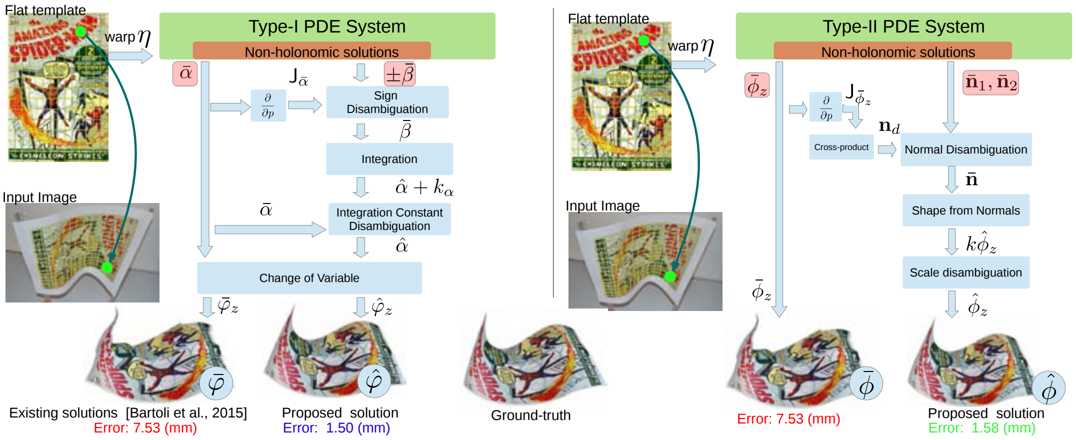


Figure 4.4: Direct-depth method and stable type-I and type-II methods for SfT. The existing solution represents the results from the direct-depth method and the proposed solutions represent the results from the stable methods.

4.3.3.2 Shape-from-Normals

Once we disambiguate the normals at each point, the next step is to obtain an embedding by integrating the normals. This can be done very efficiently in a least-squares manner using spline functions such as the B-splines. With the integration we obtain depth at each evaluated point. The integration of normals is defined as:

$$\hat{\phi}_k = k_z\hat{\phi} = \arg\min_{\phi_s} \int_{\Omega'} \left(\left(\bar{\mathbf{n}}^\top [\mathsf{J}_{\phi_s}]_1\right)^2 + \left(\bar{\mathbf{n}}^\top [\mathsf{J}_{\phi_s}]_2\right)^2 \right) d\mathbf{p}' \tag{4.69}$$

where k_z represents the unknown scale of reconstruction due to the integration of unit normals and $\hat{\phi}_k \in C^1(\Omega', \mathbb{R}^3)$ is the shape obtained after integration and $\mathbf{p}'$ represents a point on Ω'. We represent the jth column of the Jacobian matrix J_{ϕ_s} as $[\mathsf{J}_{\phi_s}]_j$.

4.3.3.3 Scale computation

The surface embedding obtained by integrating the unit surface normals are not in the correct scale. To fix the scale, we first compute an approximate embedding $\bar{\phi}$ from the direct-depth solution. We parametrize the obtained shape as $\hat{\phi}_k = [Q_x \; Q_y \; Q_z]^\top$. We then use $\bar{\phi}$ and the shape $\hat{\phi}_k$ obtained from Shape-from-Normals to fix the scale as follows:

$$k_z = \left(\frac{\int_{\Omega'} \left(Q_x\bar{\phi}_x + Q_y\bar{\phi}_y + Q_z\bar{\phi}_z\right) \, d\mathbf{p}'}{\int_{\Omega'} \left(Q_x^2 + Q_y^2 + Q_z^2\right) \, d\mathbf{p}'} \right)^{-1}. \tag{4.70}$$

Thus after the scale correction we obtain a stable reconstruction from the stable type-II method. These steps are summarized in algorithm 3.

Algorithm 3: Stable type-II methods for the perspective and infinitesimal weak-perspective camera.

```
Input: warp $\eta$, template embedding $\varDelta$, domain $\varOmega$
Output: deformed embedding $\hat{\phi}$
• PDE Solution
    Compute the Jacobians $\mathtt{J}_\eta$ and $\mathtt{J}_\varDelta$
    Change the parametrization space and compute $\mathtt{J}_{\eta'}$
    Solve the PDE system (4.51) or (4.62) to obtain $\bar{\phi}_z$ and $\bar{\mathbf{n}}_1, \bar{\mathbf{n}}_2$
• Normal disambiguation
    Compute a surface embedding $\bar{\phi}$ from the direct depth solution $\bar{\phi}_z$
    Find a normal field $\mathbf{n}_d$ on the surface using the embedding $\bar{\phi}$
    Select the analytic normal that is closest in angle to $\mathbf{n}_d$ corresponding to the highest
      dot-product, i.e. $\bar{\mathbf{n}} = \bar{\mathbf{n}}_k$, such that $k = \arg\max_{k\in\{1,2\}} |\mathbf{n}_d^\top \bar{\mathbf{n}}_k|$.
• Shape-from-Normals
    Integrate the disambiguated surface normal field $\bar{\mathbf{n}}$ by solving the minimization
      problem (4.69)
• Scale computation
    Correct the scale of the surface embedding to obtain $\hat{\phi}$
```

4.4 Experimental Results

4.4.1 Compared methods and error measurements

We performed the experiments using MATLAB. The results of the experiments were used to obtain plots for the depth error (3D error) and the normal error in different conditions. We computed the depth error by taking the root mean square error of the reconstructed 3D point coordinates. We measured the normal error by taking the root mean square deviation in angle of the reconstructed surface normals from the ground-truth surface normals. Our set of proposed methods consists of the stable type-I method (**typeI**-P, **typeI**-WP) and the stable type-II method (**typeII**-P, **typeII**-WP). The suffixes 'P' and 'WP' stand for the perspective and infinitesimal weak-perspective camera models respectively. We ran the methods based on the infinitesimal weak-perspective camera model only for the scenes with changing focal length as they are expected to work only for large focal lengths. We use **direct**-P for the analytic direct-depth method (Bartoli et al., 2015). We also compared against the zeroth-order methods based on inextensibility, denoted here as **inext-mdh**-P (Salzmann and Fua, 2011a) and **inext-lap**-P (Ngo et al., 2016). Finally we tested the statistically optimal cost optimization (Brunet et al., 2014) with the direct depth solution as input. We denote the refined solution as **refined**-P. We tuned each method with the best set of parameters for each dataset. We describe the complete algorithm for SfT, experimental setup and the results on each dataset separately below.

4.4.2 Complete algorithm

The SfT framework used for the analytic methods requires a 3D template and template-to-image registration warps. Here we briefly list the steps for inferring the deformed shape starting from the 3D template and an input image for real datasets where outliers appear naturally: *1)* We compute the Δ parametrization from the template flattening to the 3D template using Bicubic B-Splines (BBS) with a smoothness prior (Dierckx, 1993). This is an LLS problem. When the 3D template is not flat, we make use of a template image as the flattening. *2)* We obtain point correspondences between the template image and the input image using SIFT (Lowe, 2004) or KAZE (Alcantarilla et al., 2012) or a combination of them. An alternative approach is to match using a denser graph matching method such as (Collins et al., 2014). For most real datasets, these correspondences will in general contain some outliers. In either case, outliers are removed using (Pizarro and Bartoli, 2012). *3)* The correspondences thus established will have none or very few outliers even in difficult conditions. In more difficult situations such as the example shown in Appendix C, we opt for a convex L1-minimization in place of the LLS problem in (Dierckx, 1993) to reduce the effect of outliers while estimating the template-to-image warp. If the registrations are still not good enough, a robust M-estimator may be used but we found an L1 estimation to be sufficient for the examples. Optionally methods such as the pixel intensity based registration refinement also given in (Pizarro and Bartoli, 2012) or the affine transform based outlier rejection (Puerto and Mariottini, 2013) can be used to improve the registration if necessary. *4)* We use the registration obtained in the above step to generate 2D correspondences and the registration derivatives. *5)* Finally we obtain the reconstructed points from SfT.

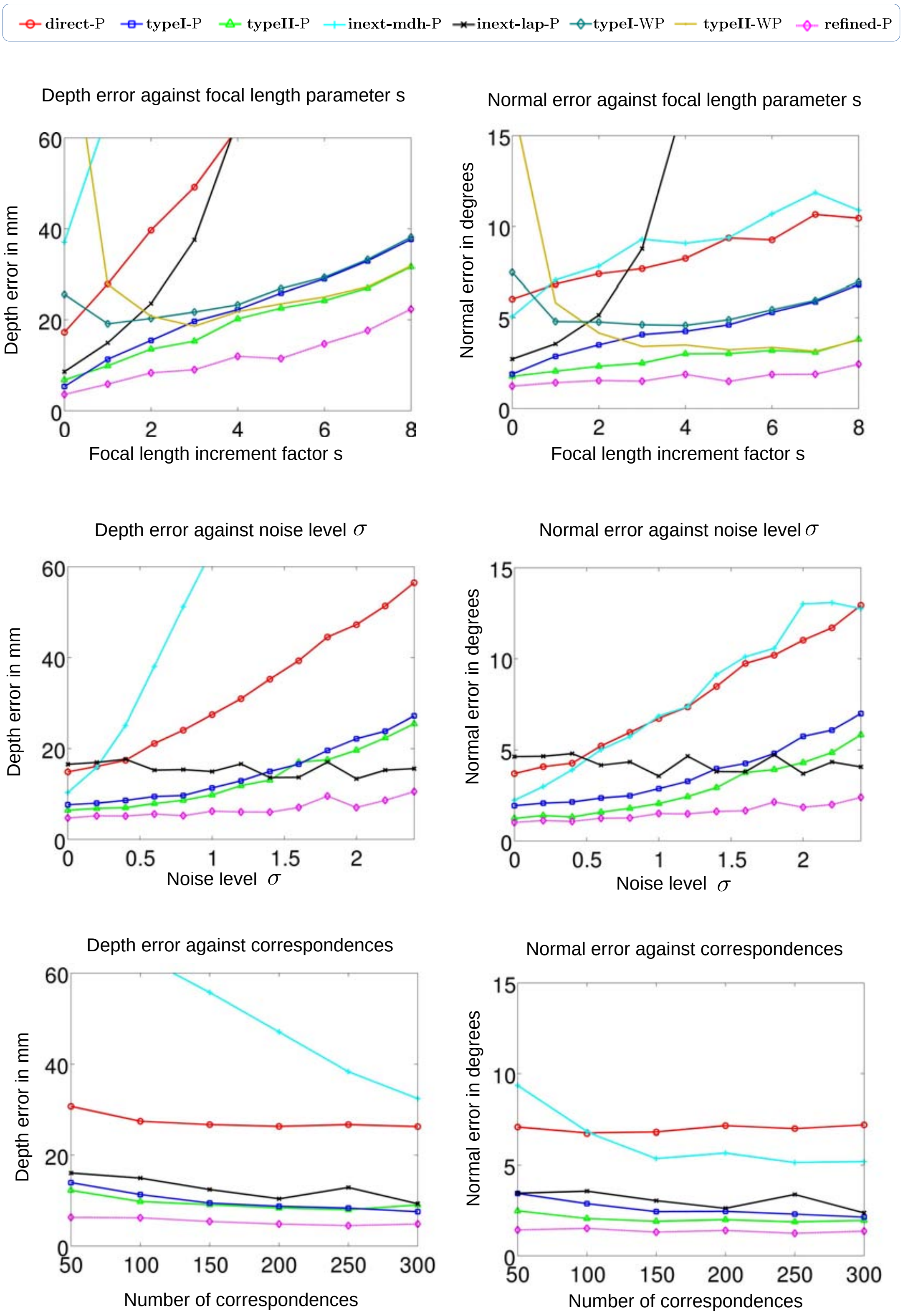


Figure 4.5: Plots for the synthetic dataset. We show the depth errors in the first column and the normal errors in the second column.

4.4.3 Synthetic data

We simulated 10 different surfaces generated by isometric deformation of a flat template surface (Perriollat and Bartoli, 2013). The template size used was 640 px $\times$ 480 px. The images for each deformation were taken using a virtual pin-hole camera of varying focal length. We fixed the focal length using a single parameter as: $f = (s + 1)500$ px. While changing the focal length, we also translated the object so that the size of its projection remained fixed in the image. In the experiments we varied s from 0 to 8. The number of correspondences N used to estimate the warp η was varied from 50 to 300. We added Gaussian noise in the images with a standard deviation σ varying between 0 and 2.4 px. In each experiment we changed only one parameter, fixing the others. The default values of the parameters were $s = 1$, $N = 100$ and $\sigma = 1.0$ px. The resulting plots from the experiments are shown in figure 4.5.

The plots show that the stable methods **typeI**-P and **typeII**-P have the best performance among all methods, after **refined**-P. The stable method **typeII**-P has a slight edge over **typeI**-P in most conditions. This is because compared to **typeI**-P, **typeII**-P uses integration in the space of normals rather than a space of quantity containing the registration η term. Even so, both of these methods show better performance than the original analytic solution **direct**-P and also compared to the zeroth-order methods **inext-mdh**-P and **inext-lap**-P. Both the direct-depth method and zeroth-order methods perform poorly against the increasing focal length. On the other hand our proposed methods **typeI**-P and **typeII**-P show remarkable stability and have performance similar to **refined**-P. Another interesting observation is the convergence of the infinitesimal weak-perspective camera model to the perspective accuracy with the increase in the focal length. These observations validate the theoretical results we obtained in sections 4.2.2 and 4.3.2. The zeroth-order method **inext-lap**-P performs well against the increasing number of correspondences and the image noise. The analytic solution **direct**-P shows poor performance against increasing noise while, on the other hand, both **typeI**-P and **typeII**-P are robust against noise in correspondences. Similar observations also hold for the normal error.

4.4.4 Real data

We tested all methods with four different real datasets. We used one more dataset to make an experiment where we show how a robust registration can help the reconstruction in appendix C. The first two were built with developable surfaces with a flat 3D template. We constructed the others out of non-developable surfaces. We describe each of these datasets and observations independently below. Table 4.1 summarizes the results on all real datasets.

The KINECT Paper dataset. The KINECT Paper dataset (Varol et al., 2012a) consists of 191 frames taken with about the same angle and focal length of a sheet of paper being deformed. The number of matched features in each frame of the sequence is around $N = 1300$ but varies from frame to frame. The image size is 640 px $\times$ 480 px with focal length of approximately 526 px. The performance of different methods for each frame is plotted in figure 4.6.

As the dataset is highly perspective and has a large number of feature correspondences distributed more or less uniformly over the scenes, all of the methods perform well. The mean depth errors for our stable methods, however, are significantly lower than those for the others, including **refined**-P, as shown in table 4.1.

The Zooming dataset. In order to test the performance of methods with varying focal length, we used a real dataset with a single deformation (Bartoli et al., 2013) and known template. The dataset shows a folded sheet of paper with different focal lengths and views. The focal length varies from 2696 px to 7875 px with an image size of 1728 px × 1552 px. Each zoom level has 7 to 10 images with different viewing angles. We computed the ground-truth from each view in camera coordinates using stereo triangulation and feature correspondences from SIFT. Figure 4.6 shows the computed plots for the shape and the depth errors for different methods. We also show the qualitative results with error coded texture maps for three images of different focal lengths in figure 4.7.

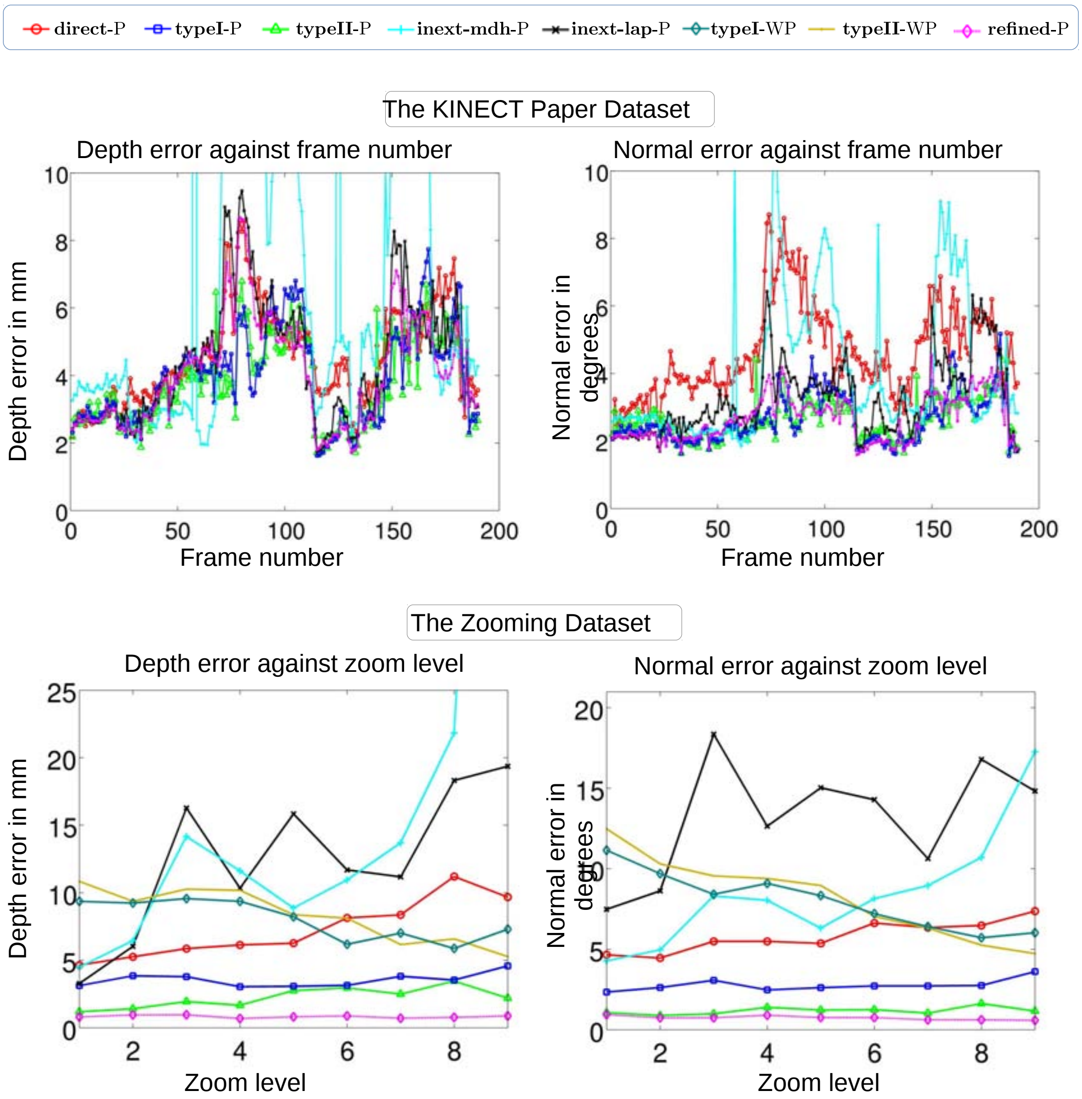
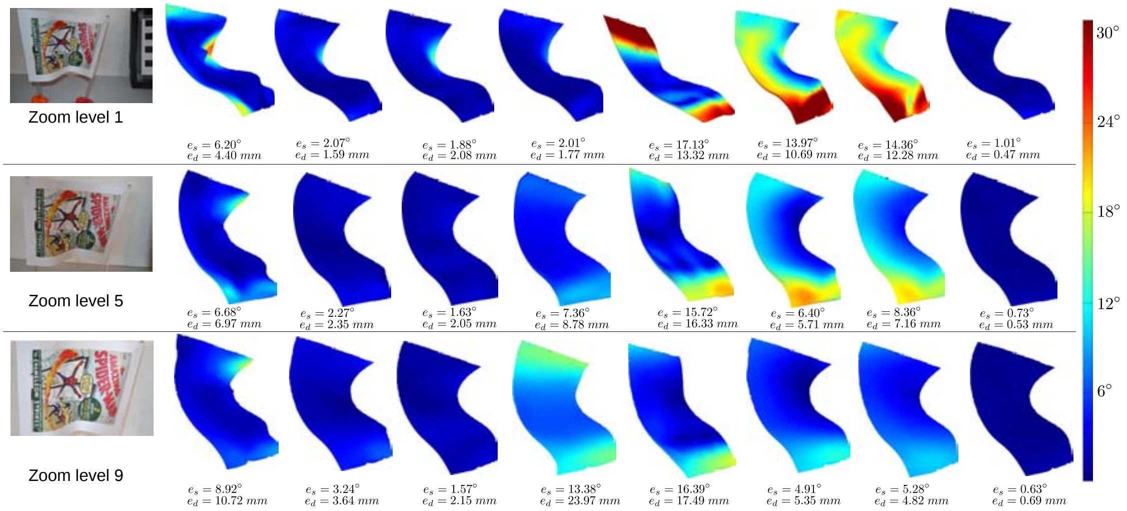


Figure 4.6: Plots for the KINECT Paper dataset and the Zooming dataset.


Zoom level 1
$e_s = 6.20°$ $e_d = 4.40\ mm$
$e_s = 2.07°$ $e_d = 1.59\ mm$
$e_s = 1.88°$ $e_d = 2.08\ mm$
$e_s = 2.01°$ $e_d = 1.77\ mm$
$e_s = 17.13°$ $e_d = 13.32\ mm$
$e_s = 13.97°$ $e_d = 10.69\ mm$
$e_s = 14.36°$ $e_d = 12.28\ mm$
$e_s = 1.01°$ $e_d = 0.47\ mm$
Zoom level 5
$e_s = 6.68°$ $e_d = 6.97\ mm$
$e_s = 2.27°$ $e_d = 2.35\ mm$
$e_s = 1.63°$ $e_d = 2.05\ mm$
$e_s = 7.36°$ $e_d = 8.78\ mm$
$e_s = 15.72°$ $e_d = 16.33\ mm$
$e_s = 6.40°$ $e_d = 5.71\ mm$
$e_s = 8.36°$ $e_d = 7.16\ mm$
$e_s = 0.73°$ $e_d = 0.53\ mm$
Zoom level 9
$e_s = 8.92°$ $e_d = 10.72\ mm$
$e_s = 3.24°$ $e_d = 3.64\ mm$
$e_s = 1.57°$ $e_d = 2.15\ mm$
$e_s = 13.38°$ $e_d = 23.97\ mm$
$e_s = 16.39°$ $e_d = 17.49\ mm$
$e_s = 4.91°$ $e_d = 5.35\ mm$
$e_s = 5.28°$ $e_d = 4.82\ mm$
$e_s = 0.63°$ $e_d = 0.69\ mm$
30°
24°
18°
12°
6°

(a) **direct**-P (b) **typeI**-P (c) **typeII**-P (d) **inext-mdh**-P (e) **inext-lap**-P (f) **typeI**-WP (g) **typeII**-WP (h) **refined**-P

Figure 4.7: Rendered 3D results with error coded texture maps for the Zooming dataset. We use the normal error to generate the texture maps.

The error plots show that both **typeI**-P and **typeII**-P perform the best in the dataset while **direct**-P, **inext-mdh**-P and **inext-lap**-P do not perform very well. In particular, there could be two important reasons behind the low accuracy of **inext-lap**-P for this dataset. First the images do not have very short focal lengths and since it requires some perspective, the results can be expected to be less accurate in such conditions. Second, the optimal parameters for the method change from image to image. We used a single set of parameters for all images in the dataset. We impose smoothness for all reconstructions using a BBS warp. The results further confirm that **direct**-P is more susceptible to noise in image correspondences.

The Cushion dataset. The datasets tested so far are all developable surfaces. Thus a flat template could be used so that $\varDelta$ was always an identity. A slightly different situation occurs when the template is not flat and its given flattening is non-isometric. In this dataset we allowed a cushion to deform into several shapes. The deformations were largely isometric since there was very little stretching or expansion of the textured cloth. We made 5 different deformations of the cushion and we used one of the deformation as the template. Making an isometric flattening was not possible in this case and therefore we used the template image as the flattening. This also entailed the use of a locally isometric flattening for **typeII**-P and a nontrivial $\varDelta$. The size of the images used is 3456 px $\times$ 2304 px. The focal length of the camera is about 2700 px, thus the images contain a moderately large amount of perspective.

The feature correspondences here were computed by combining SIFT and KAZE (Alcantarilla et al., 2012) features. Due to the lack of point correspondences in some regions, the computed warps have less accuracy than in all other datasets. We present the results for three of the deformations in figure 4.8 using error coded texture maps along with the shape and depth error for each reconstruction. We observe again that the stable methods **typeI**-P and **typeII**-P have the best performance. The

Table 4.1: Mean depth errors in real datasets.

	Depth error measurements for different methods in mm					
Datasets	**direct**-P	**typeI**-P	**typeII**-P	**inext-mdh**-P	**inext-lap**-P	**refined**-P
KINECT Paper	4.57	3.98	**3.82**	7.78	4.43	4.02
Zooming	7.28	3.54	2.22	20.16	12.47	**0.82**
Cushion	14.37	5.02	**3.48**	7.71	7.39	5.99
Can	3.03	1.38	**1.07**	4.07	1.91	1.31

Table 4.2: Compared methods and their characteristics.

Methods	Constraints	Primary computation	Time (sec)	Stability in affine condition
direct-P	Differential first-order	Small systems	0.13	Not stable
typeI-P	Differential first-order	LLS Integration	0.53	Stable
typeII-P	Differential first-order	LLS Integration	1.74	Stable
inext-mdh-P	Zeroth-order MDH	Convex optimization SOCP	2.96	Not stable
inext-lap-P	Zeroth-order inextensibility	LLS, non-convex optimization	7.18	Not stable
refined-P	Differential first-order	Non-convex optimization	26.37	Stable

zeroth-order methods **inext-mdh**-P and **inext-lap**-P also show good performance as the images have high perspective. However we failed to obtain good results with **direct**-P. This further confirms the greater sensitivity of **direct**-P to noise in the feature correspondences due to the instability of the depth solution.

The Can dataset. We prepared a dataset by deforming a can made of a cardboard material. The dataset consists of 3 different deformations of different degrees and a template surface made with the original surface. As it was not possible to flatten the surface physically, we again used the template image as the flattening. The size of the images used is 4800 px × 3200 px with a focal length of 11000 px. We computed the flat template-to-image warps again by combining SIFT and KAZE feature correspondences. We present the qualitative results with error coded texture maps and error measurements in figure 4.8. The results show that our proposed methods are again the best performing. Similarly **direct**-P shows a medium accuracy while **inext-mdh**-P performs poorly in 2 out of 3 scenes.

4.5 Discussions

Isometric SfT methods are close to achieving reconstruction accuracies that give them applicability in real scenarios. However, as we showed here, several aspects of current state-of-the-art methods, specifically their poor performance in low perspective and their sensitivity to noise in image correspondences pose major problems to achieving useful SfT results. We proposed methods that push the boundary of the state of the art further in terms of reconstruction accuracy while extending applicability in different projection geometries. We found that obtaining depth directly to get the 3D shape

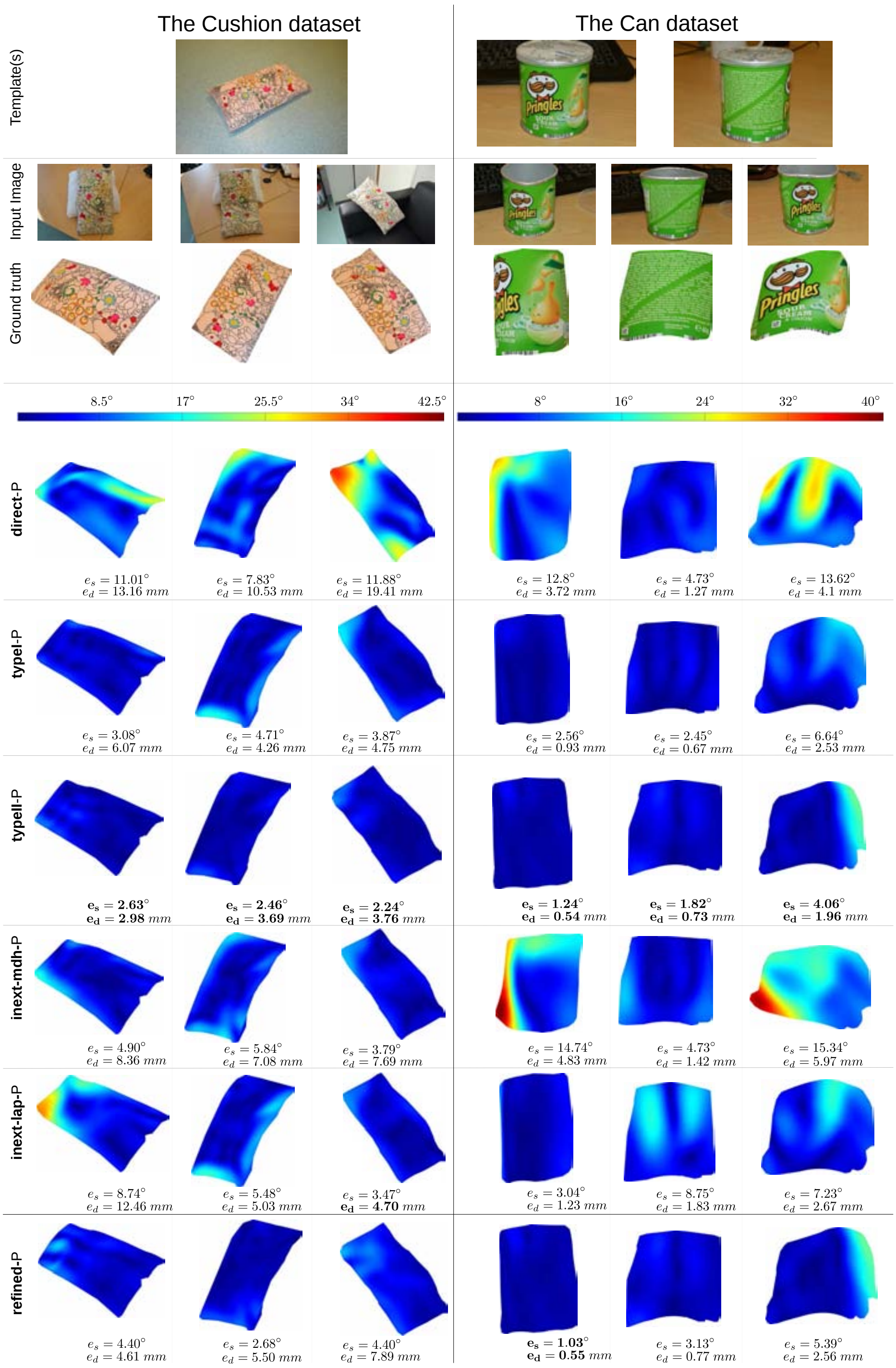

The Cushion dataset
The Can dataset
Template(s)
Input Image
Ground truth
8.5° 17° 25.5° 34° 42.5°
8° 16° 24° 32° 40°
direct-P
$e_s = 11.01°$ $e_d = 13.16\ mm$
$e_s = 7.83°$ $e_d = 10.53\ mm$
$e_s = 11.88°$ $e_d = 19.41\ mm$
$e_s = 12.8°$ $e_d = 3.72\ mm$
$e_s = 4.73°$ $e_d = 1.27\ mm$
$e_s = 13.62°$ $e_d = 4.1\ mm$
typeI-P
$e_s = 3.08°$ $e_d = 6.07\ mm$
$e_s = 4.71°$ $e_d = 4.26\ mm$
$e_s = 3.87°$ $e_d = 4.75\ mm$
$e_s = 2.56°$ $e_d = 0.93\ mm$
$e_s = 2.45°$ $e_d = 0.67\ mm$
$e_s = 6.64°$ $e_d = 2.53\ mm$
typeII-P
$\mathbf{e_s = 2.63°}$ $\mathbf{e_d = 2.98}\ mm$
$\mathbf{e_s = 2.46°}$ $\mathbf{e_d = 3.69}\ mm$
$\mathbf{e_s = 2.24°}$ $\mathbf{e_d = 3.76}\ mm$
$\mathbf{e_s = 1.24°}$ $\mathbf{e_d = 0.54}\ mm$
$\mathbf{e_s = 1.82°}$ $\mathbf{e_d = 0.73}\ mm$
$\mathbf{e_s = 4.06°}$ $\mathbf{e_d = 1.96}\ mm$
inext-mdh-P
$e_s = 4.90°$ $e_d = 8.36\ mm$
$e_s = 5.84°$ $e_d = 7.08\ mm$
$e_s = 3.79°$ $e_d = 7.69\ mm$
$e_s = 14.74°$ $e_d = 4.83\ mm$
$e_s = 4.73°$ $e_d = 1.42\ mm$
$e_s = 15.34°$ $e_d = 5.97\ mm$
inext-lap-P
$e_s = 8.74°$ $e_d = 12.46\ mm$
$e_s = 5.48°$ $e_d = 5.03\ mm$
$e_s = 3.47°$ $\mathbf{e_d = 4.70}\ mm$
$e_s = 3.04°$ $e_d = 1.23\ mm$
$e_s = 8.75°$ $e_d = 1.83\ mm$
$e_s = 7.23°$ $e_d = 2.67\ mm$
refined-P
$e_s = 4.40°$ $e_d = 4.61\ mm$
$e_s = 2.68°$ $e_d = 5.50\ mm$
$e_s = 4.40°$ $e_d = 7.89\ mm$
$\mathbf{e_s = 1.03°}$ $\mathbf{e_d = 0.55}\ mm$
$e_s = 3.13°$ $e_d = 0.77\ mm$
$e_s = 5.39°$ $e_d = 2.56\ mm$

Figure 4.8: Rendered 3D results with error coded texture maps for the Cushion and the Can datasets.

is not the best approach despite its appeal. In contrast, we proposed the use of stable solutions (based on depth-gradient or surface normal) that proved to be accurate in perspective as well as affine conditions. We provided theoretical proofs to their stability. Though being very similar, they differ in two important aspects: disambiguation and integration. In particular we use numerical integration with bending energy of the BBS, which imposes smoothness in the given space. Compared to the stable type-I method, the stable type-II method uses the space of surface normals for integration that gives a slightly better performance seen in the experiments. Both of these two new methods we presented used analytic solutions and obtained near or sometimes better than the statistically optimal results. In short, we found the first-order solution of the SfT PDE to be more stable than the zeroth-order solution. However, reconstruction by integration for the proposed methods implies that we can only use them for smooth surfaces.

Our results also show that the inextensibility-based methods are not stable in near-affine projection geometries. Although we do not theoretically prove this point, an intuitive understanding can be obtained from the fact that inextensibility prior is strong when the sightlines passing through the point correspondences are diverging and not close to parallel. When such is not the case as in near affine cases, a large range of change in depth only implies a small change in the Euclidean distances between points. This affects the conditioning of the problem.

For all the experiments with the analytic methods, we computed the template-to-image warp required for the analytic methods globally. In future works, the warp could be computed locally. By nature, a local warp would possibly capture the local changes better than a global one, thus giving better reconstruction in cases of large local deformations. Table 4.2 shows a summary of the main characteristics for different methods. It is true that the zeroth-order methods are affected by inaccuracies of the computed warp depending on the presence of outliers but such conditions are rare in our experiments. The time noted is the average time taken to reconstruct a single scene for the real datasets using a standard desktop PC. The parameters used in our methods for the global warps are very easy to tune and the methods themselves give local solutions meaning that they can be parallelized if needed for higher speed.

4.6 Conclusion

In this chapter we discussed the SfT problem and its local analytic solutions proposed in (Bartoli et al., 2015). In that context we presented two important results regarding the local analytic solutions. First the depth solution is not well-constrained when the projection geometry tends to affine whereas the first-order solution related to the surface normal or depth-gradient remains stable in near-affine conditions. We thus presented our methods based on the stable non-holonomic solutions of the SfT PDE proposed in (Bartoli et al., 2015). We found from several experiments that our proposed methods are able to give better reconstructions than the state-of-the-art methods for smooth objects. In the next chapter we will reformulate the non-rigid 3D reconstruction problem of SfT into the template-less case of NRSfM.

Chapter 5

Local Solution for Non-Rigid Shape-from-Motion

In this chapter we first describe the local isometric constraints for NRSfM similar to that of SfT. We here prove that unlike in SfT, the first-order local NRSfM constraints are under-constrained. We then give a local method of solving NRSfM using the second-derivatives of the registration warp. We achieve this by equating the registration warp and its derivatives to that of a *differential homography* at each point. We introduce differential homography as an alternative way to compute homography other than using four or more point-correspondences. This is possible due to the assumption of infinitesimal planarity and isometry. Instead we use the differential information of the image registration warps to compute the homography at each point. We then use the computed homography for obtaining the solution to NRSfM. For the sake of completeness, we start by describing the SfT problem formulation. This chapter is based on our published work (Chhatkuli et al., 2014b).

5.1 A General Framework for Isometric Surfaces

We first review isometric SfT as in (Bartoli and Collins, 2013; Bartoli et al., 2012). We extend SfT to NRSfM by adding more views and keeping the template as an additional unknown. This allows us to analyze the existence of local solutions of the NRSfM system.

5.1.1 Shape-from-Template

Figure 5.1.a shows a general diagram for SfT whose solution is based on the reprojection and the deformation constraints (Bartoli et al., 2012). The known template is represented by a 2D domain $\mathcal{T}$ corresponding to the 3D template's conformal flattening. The deformed shape $\mathcal{S}$ is modeled by an unknown embedding $\varphi \in C^2(\mathcal{T};\mathbb{R}^3)$, and $\mathcal{I}$ is an image of $\mathcal{S}$. We use Π to denote perspective projection to coordinates normalized with respect to the camera intrinsics. The registration between $\mathcal{T}$ and $\mathcal{I}$ is known and modelled by an image warp $\eta \in C^2(\mathcal{T};\mathbb{R}^2)$. The *reprojection constraint* is then $\eta = \Pi \circ \varphi$. Let $\varphi = (\varphi_x \; \varphi_y \; \varphi_z)^\top$ where $\varphi_x, \varphi_y, \varphi_z \in C^2(\mathcal{T};\mathbb{R})$ model each dimension of φ.

If $\mathcal{S}$ results of an isometric deformation of the 3D template, and since $\mathcal{T}$ was obtained by conformal flattening, the *deformation constraint* is that the first fundamental form of φ is a scaled identity matrix (Bartoli et al., 2015):

$$\mathsf{J}_\varphi^\top \mathsf{J}_\varphi = \lambda^2 \mathbf{I}_2, \tag{5.1}$$

where J is again the first-order partial derivatives operator and $\lambda \in C^2(\mathcal{T};\mathbb{R}^+)$ is the flattening scale. As the two columns of J_φ are orthogonal we may rewrite equation (5.1) as:

$$(\mathsf{J}_\varphi \; \lambda\xi)(\mathsf{J}_\varphi \; \lambda\xi)^\top = \lambda^2 \mathbf{I}_{3\times 3}. \tag{5.2}$$

where $\xi \in C^2(\mathcal{T};\mathbb{R}^3)$ models the surface normal field. Note that ξ depends on φ, as it is a unitary vector orthogonal to the two columns of J_φ. To summarize, SfT consists of finding the embedding φ and normal field ξ given the warp η, the flattening scale λ and the projection Π, by solving a nonlinear PDE system:

$$\text{Find } \varphi \in C^2(\mathcal{T};\mathbb{R}^3) \text{ st} \quad \begin{cases} (\mathsf{J}_\varphi \; \lambda\xi)(\mathsf{J}_\varphi \; \lambda\xi)^\top = \lambda^2 \mathbf{I}_3 & \text{Deformation} \\ \eta = \Pi \circ \varphi & \text{Reprojection.} \end{cases} \tag{5.3}$$

equation (5.3) involves first-order derivatives of the unknown function φ. Following (Bartoli and Collins, 2013), differentiating the reprojection constraint and substituting it into the deformation constraint yields:

$$\mathsf{J}_\eta \mathsf{J}_\eta^\top + \lambda^2 (\mathsf{J}_\Pi \circ \varphi)\xi\xi^\top (\mathsf{J}_\Pi \circ \varphi)^\top = \lambda^2 (\mathsf{J}_\Pi \circ \varphi)(\mathsf{J}_\Pi \circ \varphi)^\top, \tag{5.4}$$

where $\mathsf{J}_\Pi \circ \varphi$ is a 2×3 matrix that only depends on the surface depth φ_z. Equation (5.4) is a PDE system of 3 independent equations in φ_z and ξ. Very recently, (Bartoli et al., 2012) obtained the pointwise solutions of equation (5.4), by ignoring the differential relationship between φ_z and ξ. Those solutions are called *non-holonomic* and (Bartoli et al., 2012) showed that they can be obtained

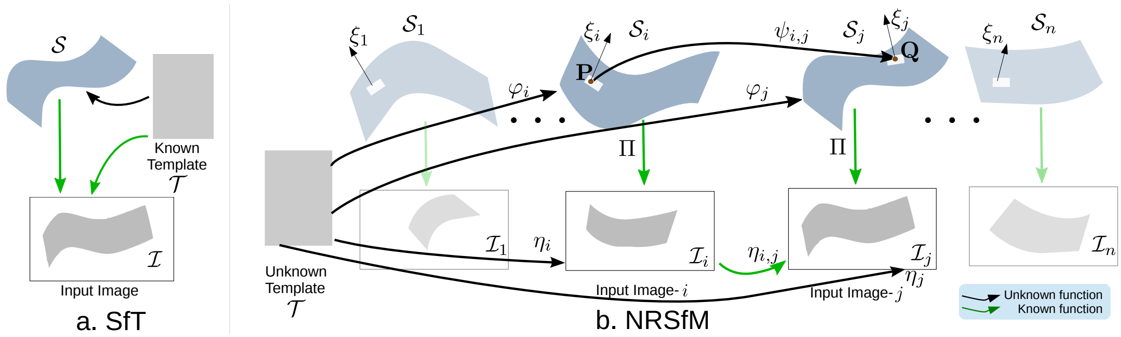


Figure 5.1: Geometric modelling of SfT and NRSfM.

analytically.

5.1.2 From SfT to NRSfM: no known template, but more images

To extend the first-order differential modelling of SfT to NRSfM we introduce n images showing a different deformation and keep the template $\mathcal{T}$ as an unknown. We use the index $i = 1, \ldots, n$ to define the i-th shape $\mathcal{S}_i$, image $\mathcal{I}_i$, warp η_i and embedding φ_i. The inter-image registration warp $\eta_{i,j}$ is known and related to the unknown warps η_i and η_j as:

$$\eta_{i,j} = \eta_j \circ \eta_i^{-1}. \tag{5.5}$$

We denote the unknown isometric deformation between $\mathcal{S}_i$ and $\mathcal{S}_j$ as $\psi_{i,j} \in C^2(\mathcal{S}_i; \mathcal{S}_j)$, where $\varphi_j = \psi_{i,j} \circ \varphi_i$. From equation (5.1), it is clear that $\psi_{i,j}$ preserves the first fundamental form between φ_i and φ_j:

$$\mathsf{J}_{\varphi_i}^\top \mathsf{J}_{\varphi_i} = \mathsf{J}_{\varphi_j}^\top \mathsf{J}_{\varphi_j} \tag{5.6}$$

In NRSfM the objective is to find the embeddings $\varphi_i, \quad i = 1, \ldots, n$ and the unknown template $\mathcal{T}$ given the pairwise image warps:

$$\text{Find} \left| \begin{array}{l} \mathcal{T} \subset \mathbb{R}^2 \\ \varphi_i \in C^2(\mathcal{T}; \mathbb{R}^3) \\ i = 1, \ldots, n \end{array} \right. \quad \text{st} \quad \begin{cases} \eta_{i,j} = \eta_j \circ \eta_i^{-1} \quad j = 1, \ldots, n \quad j \neq i & \text{Consistency} \\ \eta_i = \mathit{\Pi} \circ \varphi_i & \text{Reprojection} \\ (\mathsf{J}_{\varphi_i}\, \lambda \xi_i)(\mathsf{J}_{\varphi_i}\, \lambda \xi_i)^\top = \lambda^2 \mathbf{I}_3 & \text{Deformation.} \end{cases} \tag{5.7}$$

5.1.3 Isometric NRSfM is not locally solvable at first-order

We show that system (5.7) can be expressed as a nonlinear PDE system in terms of the surfaces' depth and normal, and the unknown template. Our main result is that the resulting system is not locally solvable, which means that its non-holonomic solutions are underconstrained. We first derive the NRSfM system for two views i and j. We start from equation (5.4), which combines the reprojection and deformation constraints of equation (5.7). We parametrize $\mathsf{J}_{\eta_i} \in C^2(\mathcal{T}; \mathbb{R}^{2\times 2})$ in the following general form:

$$\begin{aligned} \mathsf{J}_{\eta_i} &= \sigma \mathbf{M} \mathbf{R}_\theta \quad \text{with} \quad \mathbf{R}_\theta \mathbf{R}_\theta^\top = \mathbf{I}_2, \\ \mathbf{M} &= \begin{bmatrix} 1 & \beta \\ 0 & \alpha \end{bmatrix} \end{aligned} \tag{5.8}$$

where $\mathbf{R}_\theta$ is a 2D rotation of angle θ. Invoking Cholesky decomposition, the 4 dimensions of $\mathbf{R}^{2\times 2}$ are equivalent to $(\sigma, \theta, \beta, \alpha)$. Multiplying J_{η_i} by its transpose we obtain:

$$\mathsf{J}_{\eta_i} \mathsf{J}_{\eta_i}^\top = \sigma^2 \mathbf{M}\mathbf{M}^\top. \tag{5.9}$$

Equation (5.9) reveals that equation (5.4) is invariant to the 2D rotation $\mathbf{R}_\theta$. To use equation (5.4) with φ_j while following the consistency relation in equation (5.7), we differentiate (5.5) as:

$$\left(\mathsf{J}_{\eta_j} \circ \eta_i^{-1}\right) \mathsf{J}_{\eta_i}^{-1} = \mathsf{J}_{\eta_{i,j}}. \tag{5.10}$$

We then combine equations (5.8) and (5.10) to obtain

$$\mathsf{J}_{\eta_j} \circ \eta_i^{-1} = \sigma \mathsf{J}_{\eta_{i,j}} \mathbf{M} \mathbf{R}_\theta. \tag{5.11}$$

By multiplying each side of equation (5.11) to the right by its transpose, the rotation vanishes:

$$\left(\mathsf{J}_{\eta_j} \circ \eta_i^{-1}\right) \left(\mathsf{J}_{\eta_j} \circ \eta_i^{-1}\right)^\top = \sigma^2 \mathsf{J}_{\eta_{i,j}} \mathbf{M}\mathbf{M}^\top \mathsf{J}_{\eta_{i,j}}^\top. \tag{5.12}$$

As $\mathcal{T}$ is just required to be a conformal flattening of the 3D template we may choose the scale factor $\sigma = \lambda$. Introducing equation (5.8) and (5.12) in equation (5.4) we obtain the isometric NRSfM system of PDEs for two unknown surfaces φ_i and φ_j:

$$\begin{cases} \mathbf{M}\mathbf{M}^\top + (\mathsf{J}_\Pi \circ \varphi_1)\xi_i \xi_i^\top (\mathsf{J}_\Pi \circ \varphi_i)^\top = (\mathsf{J}_\Pi \circ \varphi_i)(\mathsf{J}_\Pi \circ \varphi_i)^\top \\ \mathsf{J}_{\eta_{i,j}} \mathbf{M}\mathbf{M}^\top \mathsf{J}_{\eta_{i,j}}^\top + (\mathsf{J}_\Pi \circ \varphi_j)\xi_j \xi_j^\top (\mathsf{J}_\Pi \circ \varphi_j)^\top = (\mathsf{J}_\Pi \circ \varphi_j)(\mathsf{J}_\Pi \circ \varphi_j)^\top. \end{cases} \tag{5.13}$$

equation (5.13) is an algebraic system of 6 equations and 8 unknowns $(\varphi_{i,z}, \xi_i, \varphi_{j,z}, \xi_j, \alpha, \beta)$ at every point. The non-holonomic solutions of system (5.13) are thus underconstrained for two views. In the general case of n views the system has $3n + 2$ unknowns and $3n$ independent equations. Its non-holonomic solutions are thus underconstrained for $n > 2$ views as well. Our main result is that without further assumptions, one cannot solve isometric NRSfM by relaxing the relationship between depth and normal, as was done in SfT (Bartoli and Collins, 2013; Bartoli et al., 2012).

5.2 Infinitesimally Planar Isometric NRSfM

We show isometric NRSfM can be solved locally (and analytically) if we assume that the surfaces are infinitesimally planar. This approximation is equivalent to representing the surfaces with triangular meshes, where the size of the triangles is infinitesimally small. The result is that higher order surface derivatives are *locally* zero.

5.2.1 Infinitesimal projective structure

We define $\hat{\varphi}_i$ as the locally planar approximation of the embedding φ_i:

$$\hat{\varphi}_i = \varphi_i + \mathsf{J}_{\varphi_i}\delta, \tag{5.14}$$

where $\delta \in \mathbb{R}^2$ are local 2D coordinates around each point in $\mathcal{T}$. Equation (5.14) parametrizes the tangent planes of $\mathcal{S}_i$. We show next that two corresponding tangent planes on $\mathcal{S}_i$ and $\mathcal{S}_j$ are related by a rigid transform when $\psi_{i,j}$ is an isometry.

Differentiating $\varphi_j = \psi_{i,j} \circ \varphi_i$ gives:

$$\mathsf{J}_{\varphi_j} = (\mathsf{J}_{\psi_{i,j}} \circ \varphi_i)\mathsf{J}_{\varphi_i}. \tag{5.15}$$

Pre-multiplying equation (5.15) by its transpose and using equation (5.6), we show that the 3×3 matrix $\left(\mathsf{J}_{\psi_{i,j}} \circ \varphi_i\right)$ is indeed orthonormal:

$$\begin{aligned} & \mathsf{J}_{\varphi_j}^\top \mathsf{J}_{\varphi_j} = \mathsf{J}_{\varphi_i}^\top (\mathsf{J}_{\psi_{i,j}} \circ \varphi_i)^\top (\mathsf{J}_{\psi_{i,j}} \circ \varphi_i)\mathsf{J}_{\varphi_i} = \mathsf{J}_{\varphi_i}^\top \mathsf{J}_{\varphi_i} \\ \implies \quad & (\mathsf{J}_{\psi_{i,j}} \circ \varphi_i)^\top (\mathsf{J}_{\psi_{i,j}} \circ \varphi_i) = \mathbf{I}_3. \end{aligned} \tag{5.16}$$

Using equation (5.16) we represent $\hat{\varphi}_j$ as a rigid transformation of $\hat{\varphi}_i$:

$$\hat{\varphi}_j = \varphi_j + \mathsf{J}_{\varphi_j}\delta = \psi_{i,j} \circ \varphi_i + (\mathsf{J}_{\psi_{i,j}} \circ \varphi_i)\mathsf{J}_{\varphi_i}\delta = \mathbf{t}_{ij} + \mathbf{R}_{ij}\varphi_i. \tag{5.17}$$

where $\mathbf{R}_{ij} = \mathsf{J}_{\psi_{i,j}} \circ \varphi_i$ is a 3D rotation from equation (5.16) and $\mathbf{t}_{ij} = \psi_{i,j} \circ \varphi_i - \mathbf{R}_{ij}\varphi_i = \varphi_j - \mathbf{R}_{ij}\varphi_i$ represents a translation. Equation (5.17) means that two corresponding tangent planes in $\mathcal{S}_i$ and $\mathcal{S}_j$ are related by a rigid transform.

We modify the reprojection constraint in equation (5.7) for an infinitesimally planar surface as $\hat{\eta}_i = \Pi \circ \hat{\varphi}_i$. $\hat{\eta}_i$ as a function of δ is the warp between the template and the projection of the tangent plane. Using homogeneous coordinates we have $\begin{bmatrix} \hat{\eta}_i \\ 1 \end{bmatrix} \propto \hat{\varphi}_i$ and from equation (5.14) $\begin{bmatrix} \hat{\eta}_i \\ 1 \end{bmatrix} \propto \mathbf{H}_i \begin{bmatrix} \delta \\ 1 \end{bmatrix}$, with $\mathbf{H}_i = [\mathsf{J}_{\varphi_i} \ \varphi_i]$. The warp $\hat{\eta}_i$ is therefore a homography induced by the tangent plane. The image warp $\hat{\eta}_{i,j} = \hat{\eta}_j \circ \hat{\eta}_i^{-1}$ too is thus a homography given by $\mathbf{H}_{i,j} = \mathbf{H}_j \mathbf{H}_i^{-1}$.

The structure of $\mathbf{H}_{i,j}$ is well-known (Malis and Vargas, 2007): it represents the transformation between two images showing the projection of two planes related by a rigid transform (the above-derived $\mathbf{R}_{ij}$ and $\mathbf{t}_{ij}$). From (Malis and Vargas, 2007), $\mathbf{H}_{i,j}$ can be decomposed as:

$$\mathbf{H}_{i,j} = \mathbf{R}_{ij} + \xi_i \mathbf{t}_{ij}^\top, \tag{5.18}$$

where

$$\mathbf{R}_{ij} = \mathsf{J}_{\psi} \circ \varphi_i \quad \text{and} \quad \mathbf{t}_{ij} = \varphi_j - \mathbf{R}_{ij}\varphi_i. \tag{5.19}$$

Given $\mathbf{H}_{i,j}$, we can thus extract the normal field of the surface and φ_i, However, (Malis and Vargas, 2007) shows that there is always a two-fold solution for ξ_i, $\mathbf{R}_{ij}$ and $\mathbf{t}_{ij}$. With two views it is thus not possible to disambiguate reconstruction. Extra cues must be introduced. (Varol et al., 2009) proposes to use smoothness but it is not guaranteed to give the correct solution. If we use three or more views we get a collection of normals for each point (*i.e.* two for each pair of views). We can thus disambiguate the normals using more than 2 views and clustering the normals to find an agreement with the dot-product measure. A more detailed explanation is given in chapter 5.2.3.

5.2.2 Differential homography computation

We propose a method to obtain $\mathbf{H}_{i,j}$ from the registration warp $\eta_{i,j}$. This has also been used in (Bartoli and Özgür, 2016). Given a point $\mathbf{p} = [u\ v]^\top \in \mathcal{I}_i$, we assume that $\eta_{i,j}(\mathbf{p}+\epsilon) = \hat{\eta}_{i,j}(\epsilon)$ for a small $\epsilon = [\epsilon_u\ \epsilon_v]^\top$. We also consider $\hat{\mathbf{H}}_{i,j}(\epsilon) = \mathbf{H}_{i,j}(\mathbf{p}+\epsilon)$, which gives:

$$\begin{bmatrix} \rho(\epsilon)\eta_{i,j}(\mathbf{p}+\epsilon) \\ \rho(\epsilon) \end{bmatrix} = \hat{\mathbf{H}}_{i,j} \begin{bmatrix} \epsilon \\ 1 \end{bmatrix} \quad \text{where} \quad \hat{\mathbf{H}}_{i,j} = \begin{bmatrix} a & b & c \\ g & h & k \\ d & e & 1 \end{bmatrix}, \tag{5.20}$$

and $\rho(\epsilon) = d\epsilon_x + e\epsilon_y + 1$ is an unknown linear function. When $\epsilon = \mathbf{0}$ then $\rho = 1$ and $\eta_{i,j}(\mathbf{p}) = (c\ \ k)^\top$, from which we obtain c and k. By taking first and second derivatives with respect to ϵ on both sides of equation (5.20) and evaluating them at $\epsilon = \mathbf{0}$ we obtain the following system of equations in the elements of $\hat{\mathbf{H}}_{i,j}$:

$$\begin{gathered} \eta_{i,j} = \begin{bmatrix} c \\ k \end{bmatrix} \quad \mathsf{J}_{\eta_{i,j}} = \begin{bmatrix} a-cd & b-ce \\ g-kd & h-ke \end{bmatrix} \quad \frac{\partial^2 \eta_{i,j}}{\partial u^2} = \begin{bmatrix} -2d(a-cd) \\ -2d(g-kd) \end{bmatrix} \\ \frac{\partial^2 \eta_{i,j}}{\partial v^2} = \begin{bmatrix} -2e(b-ce) \\ -2e(h-ke) \end{bmatrix} \quad \frac{\partial^2 \eta_{i,j}}{\partial u \partial v} = \begin{bmatrix} -d(b-ce)-e(a-cd) \\ -d(h-ke)-e(g-kd) \end{bmatrix}. \end{gathered} \tag{5.21}$$

Once the value of c and k are obtained from the warp $\eta_{i,j}$, we formulate a system of 10 equations in 6 from system (5.21). We vectorize the 6 unknowns and solve for them using linear least-squares.

The homography $\mathbf{H}_{i,j}$ can be found from $\hat{\mathbf{H}}_{i,j}$ by a coordinate transfer. This changes the third column of the matrix $\mathbf{H}_{i,j}$. We write $\mathbf{H}_{i,j}$ as:

$$\mathbf{H}_{i,j} = \begin{bmatrix} a & b & c' \\ g & h & k' \\ d & e & f' \end{bmatrix}. \tag{5.22}$$

Thus at $\epsilon = \mathbf{0}$, both $\mathbf{H}_{i,j}$ and $\hat{\mathbf{H}}_{i,j}$ should evaluate to the same point, i.e.,

$$\mathbf{H}_{i,j} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \hat{\mathbf{H}}_{i,j} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{5.23}$$

Expanding equation (5.23) gives us the following relation for the third column of $\mathbf{H}_{i,j}$ as:

$$\begin{bmatrix} c' \\ k' \\ f' \end{bmatrix} = \begin{bmatrix} c - au - bv \\ k - gu - hv \\ 1 - du - ev \end{bmatrix}. \tag{5.24}$$

5.2.3 Algorithm

Our algorithm involves the following steps given n views of the surface: *1)* Select one view as the reference and compute the registration warp with respect to all other views using e.g. (Dierckx, 1993). *2)* For every point in the reference view and every possible pair ($n-1$ pairs) we obtain a homography using first and second order derivatives of the registration warp (equation (5.21)). *3)* Decompose the $n-1$ homographies that we obtain for each point between the reference image to the others. Thus we have two normals from each homography at this step. *4)* Remove normals that are not front facing. *5)* Cluster the normals and obtain two normals corresponding to the two largest clusters: if the two largest clusters are similarly supported, then disambiguate using agreement with neighbours (*e.g.* smoothness). Otherwise keep the normal of the largest cluster. *6)* Integrate the normal field to obtain the reference surface embedding up to an unknown scale. *7)* To reconstruct the $(n-1)$ remaining surfaces we can either change the reference surface or use SfT given that the surface computed for the reference image is now the known template.

Even though 3 views is the minimal case, in practice we use more views to avoid ambiguities due to the presence of noise and deformations. Note that we can use any image as the reference image for each point.

5.3 Experimental Evaluation

We tested our method with synthetic data along with two real datasets of a deforming piece of paper and cloth. The different views show large deformations and wide baseline viewpoints. We computed image warps using SIFT keypoints (Lowe, 2004) followed by robust registration (Pizarro and Bartoli, 2012). We modelled the inter-image warps with Bicubic B-Splines (BBS) with 20 × 20 control points (Brunet et al., 2011). We compared our method (**DiffH**) with four other: **DiscH** is our pipeline with discrete homography computation from 4 point correspondences hallucinated using η at a distance r from the central point, **p-phom** (Varol et al., 2009), **o-lrigid** (Taylor et al., 2010) and **o-sinext** (Vicente and Agapito, 2012). The comparison was done for various numbers of views and noise levels for the synthetic data and for different numbers of views for the real datasets. Quantitative evaluations were obtained by measuring the shape error (mean error of the computed normals in degrees) and the depth error (mean error in the reconstructed 3D coordinates).

5.3.1 Synthetic data

We simulated 10 different scenes of an isometrically deformed sheet of paper (Perriollat and Bartoli, 2013). The images were taken at a focal length of 200 px and their dimensions are 640 px × 480 px.

We randomly selected 400 correspondences computed with a Gaussian noise of standard deviation σ in px. We varied the number of views from $n = 4$ to $n = 10$ and the noise standard deviation from $\sigma = 0$ to $\sigma = 4\ px$. We fixed $n = 10$ for the evaluation in varying noise and $\sigma = 1.2\ px$ for the evaluation in varying number of views. The results are shown in figure 5.2.

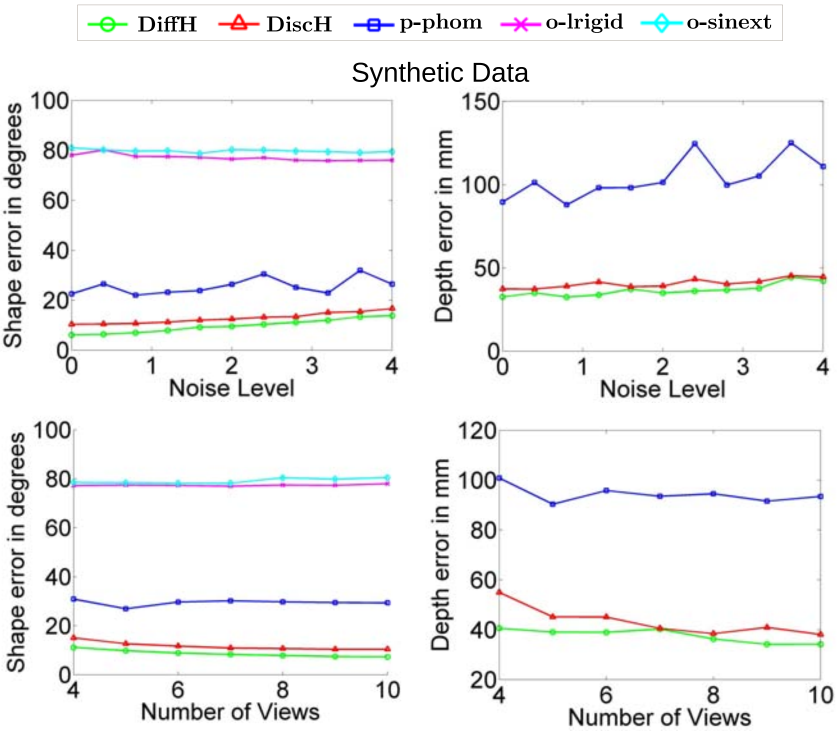


Figure 5.2: Plots for the synthetic data.

The results show that **o-lrigid** and **o-sinext** do not produce correct reconstructions with the shape error around 80 degrees. The reason for this is for the most part, the perspective nature of the images. Both **o-lrigid** and **o-sinext** methods use the orthographic camera. The depth errors for these two methods are not shown as they go beyond the scale used in the graphs. Using 10 views and no added noise, we observed a depth error of $194.14\ mm$ for **o-lrigid** and $209.2\ mm$ for **o-sinext**. **p-phom** also failed to produce good results as its approach to normal disambiguation using solely smoothness is too weak. The shape error for **DiffH** on the other hand, remains lower than 10 degrees for $n > 4$. **DiscH** follows behind with shape error about 4 degrees larger than that for **DiffH**. Clearly **DiscH** is able to reconstruct the surfaces in most circumstances but we observe better reconstructions with **DiffH** owing to the more stable local homographies. As **DiscH** estimates the homographies using a radius parameter that determines how many points or the area of the object are considered for a single homography, the accuracy of the result for a particular value of the radius parameter depends heavily on the deformation. We use the optimal value for r in **DiscH**, while we also observed that at the limit with very small values of r the reconstructions were worse.

5.3.2 Real data

We have constructed two different real datasets. The first shows a sheet of paper (the Hulk dataset) and the second shows a T-shirt (the T-shirt dataset). The Hulk dataset consists of a set of 10 images taken at different unrelated smooth deformations. We use the cover of a comics as texture. The image size is $4928\ px \times 3264\ px$ with a focal length of $3800\ px$. The T-shirt dataset also consists of a set of 10 images taken at different deformations. The image size is $4800\ px \times 3200\ px$ with a focal length of $3800\ px$. We used SfM using several images to compute the ground truth 3D shape for both of these datasets.

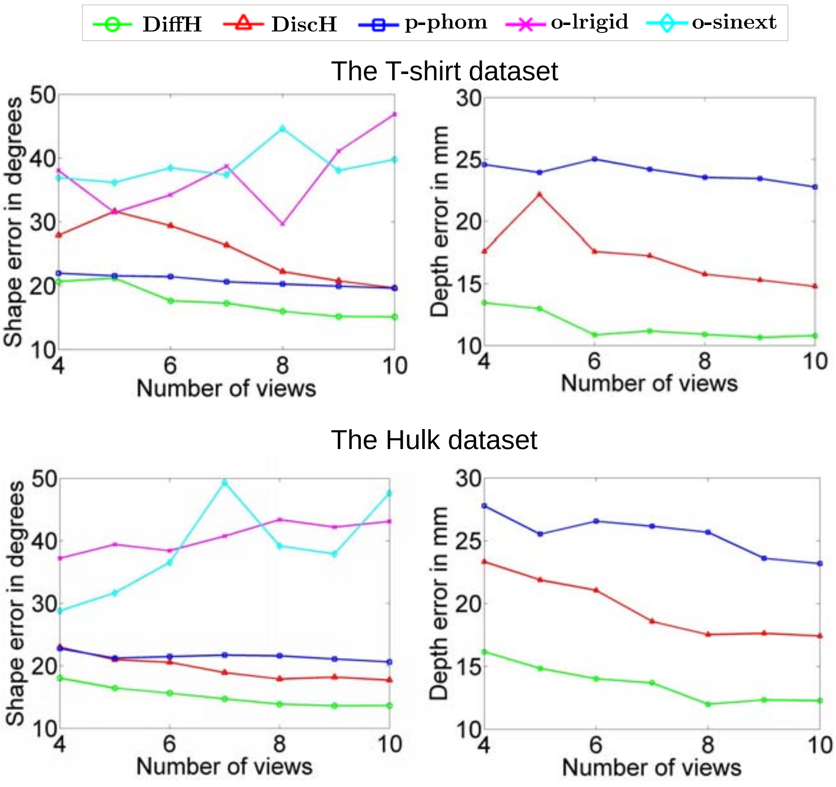


Figure 5.3: Plots for the real data.

We evaluated the five different methods with varying number of views. The results are shown in figure 5.3 and confirm our observations for the synthetic data. Again the depth errors are shown only for the three methods: **DiffH**, **DiscH** and **p-phom**. With 10 views, the mean depth error for **o-lrigid** is $48.4\ mm$ and for **o-sinext** it is $86.5\ mm$ in the Hulk dataset, and in the T-shirt dataset they are $47.3\ mm$ and $76.9\ mm$ respectively.

We also show the texture mapped reconstructions obtained using 10 views for all the compared methods on three simple examples for each real dataset in figure 5.4. The results show that **o-lrigid** and **o-sinext** miscalculated the flips and thus reconstructed the wrong shape because they use the orthographic camera. **p-phom** does not disambiguate the normals properly in most cases, thus producing good shape only for some parts of the object or for some specific deformations. These observations can also be confirmed by the shape and depth error measurements given below each reconstruction.

5.4 Discussions

We proposed a point-wise or local NRSfM method for smooth objects. As in the case of the proposed SfT method, the smoothness assumption is required to recover surface from its normals at each point. We represented both the surface embedding and the registration functions using BBS. Even though using splines means non-smoothness cannot be modelled in the obtained embedding function, additional refinement could be introduced to recover the exact surface. As an example a recent work (Gallardo et al., 2016) proposes an SfT method based on nonlinear optimization to recover non-smooth creases on surfaces. The optimization is performed starting with a smooth initial solution and combining isometry with shading cues. However, a recent local method for NRSfM (Parashar et al., 2016) has also been proposed that provides a better solution using a more complete differential model. It also uses the infinitesimal planarity on the embedding to simplify the differential model. One drawback in our method is that the computation of homography uses linear equations which are not always well-constrained, specifically when the perspective is very low. Thus it requires a very strong perspective.

5.5 Conclusion

We have presented in this chapter the first differential modelling and study of isometric NRSfM. Our model unifies SfT and NRSfM and shows that non-holonomic solutions in the first-order NRSfM are under-constrained: the relationship between depth and normal cannot be directly relaxed as in SfT. This was an important result because it gave clear indications that isometric NRSfM can be solved with two possible approaches *(i)* using second-order quantities related to registration and constraining NRSfM locally or *(ii)* constraining NRSfM globally but with only zeroth-order registration quantities. In this chapter we followed the first approach. We showed that the local isometric NRSfM problem has a solution for $n > 2$ views if we consider the surface embeddings to be infinitesimally planar. Our solution involves only linear least-squares and analytical homography decompositions. We have given a method to deal with ambiguities in the general case of n views. We showed that our method outperforms several state-of-the-art methods and produces very accurate results in sparse unordered datasets that show wide-baseline viewpoints and large deformations. In the next chapter we will consider the second approach and indeed show that much better results can be obtained by solving a global formulation of isometric NRSfM.

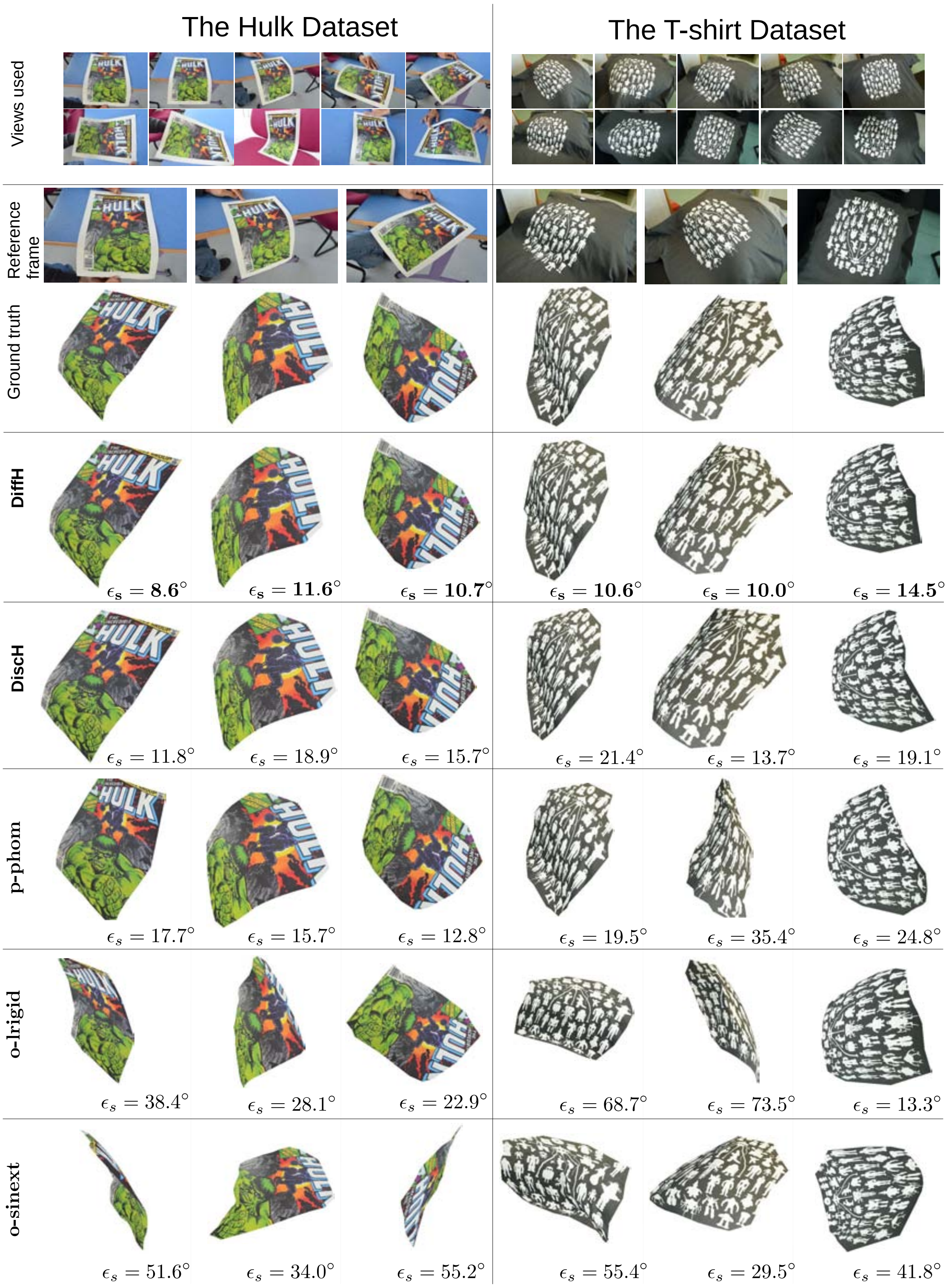


Figure 5.4: Qualitative results for three examples in the Hulk dataset and the T-shirt dataset: e_s is the shape error in degrees and e_d is the depth error in mm.

Chapter 6

Non-Rigid Shape-from-Motion with Inextensibility

In this chapter, we describe our global approach to the isometric NRSfM problem. Rather than using local or point-wise constraints independently, we combine all the constraints into a single optimization. In order to make such an optimization tractable, we relax isometry to inextensibility so that the whole problem becomes that of a convex optimization. We maximize the point depth for each image correspondence in the retinal frame under the inextensibility constraints. We bound the sum of the unknown template distances, thus in effect bounding the point depths. This guarantees that we obtain a global minimum. The global formulation means that even if constraints at a point might not be strong enough, combining them all together gives a tighter bound to the solution. Apart from that we show that the formulation is flexible enough to have robustness and temporal smoothness. This chapter is partly based on our published work (Chhatkuli et al., 2016a).

6.1 Modelling

Unlike in the previous chapters, we model the NRSfM problem using only the zeroth-order quantities of registration and the embedding. As before we express all surface 3D points in their respective camera coordinate frame. Figure 6.1 illustrates the modelling and the associated geometric terms that are described further in this section.

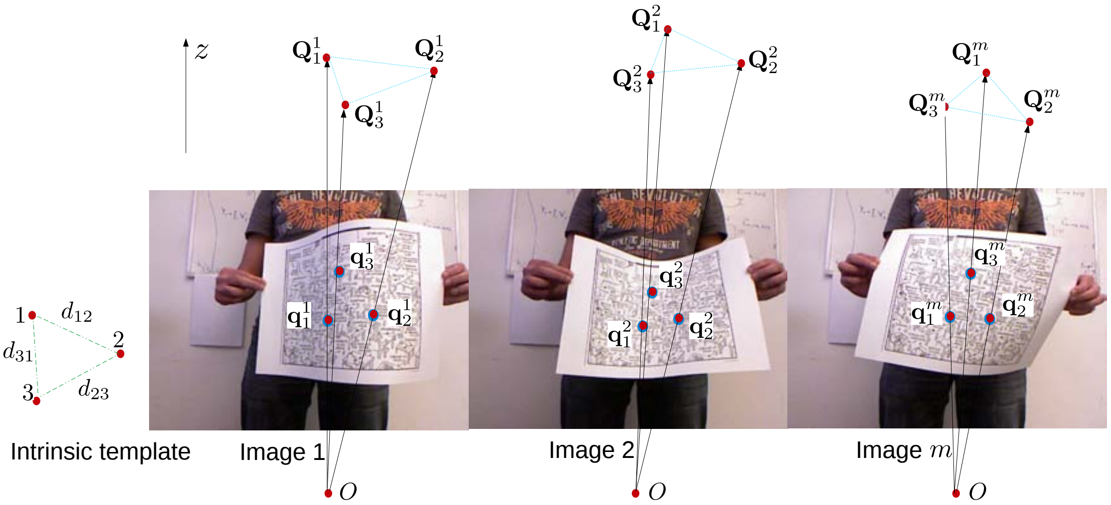


Figure 6.1: The NRSfM problem and its associated geometric terms. We use $\mathbf{O}$ to represent the camera center from which we draw the sight lines. We show only three points for clarity. In practice there can be virtually any number of points and each point can have many neighbours.

6.1.1 Point-based reconstruction

We define image measurements as a set of n normalized point correspondences in m images denoted by $\mathcal{C} \triangleq \{\mathbf{q}_i^k\}$. The 2D vector $\mathbf{q}_i^k \triangleq \begin{pmatrix} u_i^k & v_i^k \end{pmatrix}^\top$ denotes the ith point seen in the kth image. We define the unknown set of 3D points by $\mathcal{R} \triangleq \{\mathbf{Q}_i^k\}$, where $\mathbf{Q}_i^k \triangleq \begin{pmatrix} x_i^k & y_i^k & z_i^k \end{pmatrix}^\top$ denotes the unknown 3D position of $\mathbf{q}_i^k$ in camera coordinates. Because we are using the perspective camera, $\mathbf{Q}_i^k$ and $\mathbf{q}_i^k$ are related by

$$\mathbf{Q}_i^k = z_i^k \begin{pmatrix} \mathbf{q}_i^{k\top} & 1 \end{pmatrix}^\top + \epsilon_i^k \tag{6.1}$$

where ϵ_i^k is measurement noise. The NRSfM problem is solved by determining the unknown set $\mathcal{Z} \triangleq \{z_i^k\}$.

6.1.2 The intrinsic template

In chapter 3 we briefly explained the Maximum Depth Heuristic (MDH) and described how previous method (Salzmann and Fua, 2011a) solved the SfT problem using inextensibility prior and maximizing point depths. We start with the MDH-based SfT problem and migrate to the NRSfM problem. We formalize the 3D template with what we call the *intrinsic template*. This is used to solve the set of point depths $\mathcal{Z}$. We use the term intrinsic because it models properties of the surface that are invariant to isometric deformations. The intrinsic template is an undirected graph that links the n scene points through its edges. This is defined by a nearest-neighbourhood graph (NNG) whose edges store the geodesic distances between pairs of points. The NNG is denoted as $\mathcal{N}$ with n points (or *nodes*) and K edges per node. We denote $\mathcal{N}(i)$ as the set of K-neighbours of the ith point. Each edge $e_{ij} \triangleq (i, [\mathcal{N}(i)]_j)$ of the graph has an associated geodesic distance d_{ij}. Because we assume the surface deforms isometrically, we can assume d_{ij} is constant for any deformation. We denote the intrinsic template as the pair $\mathcal{T} \triangleq \{\mathcal{N}, \mathcal{D}\}$, with $\mathcal{D} \triangleq \{d_{ij}\}$.

6.1.3 Template-based reconstruction

MDH for reconstructing a deformable surface was first proposed in the template-based scenario. We therefore first describe the template-based reconstruction with MDH and move to the generic NRSfM problem. In template-based reconstruction (*i.e.* Shape-from-Template), $\mathcal{T}$ is known from the object's reference shape, which is usually built from a geometric mesh. We now describe the MDH for reconstructing an object from a single image. Without loss of generality we assume this is image 1, so the goal is to solve for $\{z_i^1\}$. A solution was first proposed in (Perriollat et al., 2008), then solved with convex optimization in (Salzmann and Fua, 2009). In MDH the deformation model is based on surface inextensibility, which says that the Euclidean distance between any two points $\mathbf{Q}_i^k$ and $\mathbf{Q}_j^k$ is upper bounded by the geodesic distance d_{ij}. The geodesic distance d_{ij} is known because the template shape is known. For simplicity we neglect the effect of the measurement noise ϵ_i^k as in (Salzmann and Fua, 2009). The problem formulation is as follows:

$$\begin{aligned}
&\underset{\{z_i^1\}}{\text{maximize}} \sum_{i=1}^{n} z_i^1 \\
&\text{subject to,} \\
&z_i^1 \geq 0 \\
&\left\| z_i^1 \begin{bmatrix} \mathbf{q}_i^1 \\ 1 \end{bmatrix} - z_j^1 \begin{bmatrix} \mathbf{q}_j^1 \\ 1 \end{bmatrix} \right\|_2 \leq d_{ij} \\
&\forall i \in \{1 \dots n\},\ j \in \mathcal{N}(i).
\end{aligned} \tag{6.2}$$

The main properties of problem (6.2) are the following. *1)* It is a Second Order Cone Program (SOCP) that can be solved efficiently and globally with modern optimization tools such as MOSEK and SeDuMi. *2)* The neighbour order K in the intrinsic template can be any. A larger K introduces more cone constraints, however it also significantly increases the computational time. Keeping a lower K is thus important for efficiency purposes.

6.2 MDH-based NRSfM

6.2.1 Initial formulation

The MDH for NRSfM can be expressed as the maximization of the sum of all depths $\{z_i^k\}$ under the inextensibility constraint and the condition that all depths and the unknown geodesic distances of the intrinsic template are positive. Unlike in template-based reconstruction, we require multiple images and in general point correspondences will not be found in all images due to occlusions, missed tracks in optical flow, etc. We therefore introduce the visibility set $\mathcal{V} \triangleq \{v_i^k\}$, where $v_i^k = 1$ if the ith point is visible in the kth image and $v_i^k = 0$ otherwise. We assume the visibility set to be known, meaning that we know which points are missing in each image. We formulate the problem as follows:

$$\begin{aligned}
&\underset{\{z_i^k\},\{d_{ij}\}}{\text{maximize}} \sum_{k=1}^{m}\sum_{i=1}^{n} v_i^k z_i^k \\
&\text{subject to,} \\
&z_i^k \geq 0, \quad d_{ij} \geq 0 \\
&v_i^k v_j^k \left\| z_i^k \begin{bmatrix} \mathbf{q}_i^k \\ 1 \end{bmatrix} - z_j^k \begin{bmatrix} \mathbf{q}_j^k \\ 1 \end{bmatrix} \right\|_2 \leq v_i^k v_j^k d_{ij} \\
&\forall k \in \{1 \ldots m\},\ i \in \{1 \ldots n\},\ j \in \mathcal{N}(i).
\end{aligned} \tag{6.3}$$

To handle missing correspondences, we fix $z_i^k = 0$ if $v_i^k = 0$ and therefore we do not reconstruct the points that are not visible. The known visibility set is used in problem (6.2) to disconnect the inextensibility conditions when any of the points involved is not visible. In contrast to the template-based problem (6.2), in the template-less problem (6.3) we do not know the intrinsic template $\mathcal{T}$. It is clear that solving problem (6.3) directly is not possible for two reasons: *1)* the optimization is not well posed because d_{ij} is unbounded (one can keep increasing d_{ij} and the constraints will still be satisfied), *2)* the NNG is an unknown. We now give the solutions to both issues.

6.2.2 Bounding the distances

In order to bound the problem, our idea is to fix the scale of the intrinsic template, by fixing the sum of the geodesic distances to a positive scalar (1 in our case). Formally we include in problem (6.3) the following linear constraint:

$$\sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} d_{ij} = 1. \tag{6.4}$$

By including equation (6.4), $\{z_i^k\}$ cannot increase indefinitely without violating equation (6.4), yet the problem is still an SOCP. We illustrate this in figure 6.2. The effect of equation (6.4) is to fix the scale of the reconstruction. In NRSfM we are free to fix the scale of the reconstruction arbitrarily, because just like in rigid SfM, it is never recoverable. Having fixed the scale, the reconstructed depths cannot increase arbitrarily, because with a perspective camera as the depths increase so do Euclidean distances between pairs of points. At some point, the Euclidean distances will exceed the geodesic distances and the inextensibility constraints (last line of problem (6.3)) will be violated.

6.2.3 The nearest-neighbour graph

The function of the NNG is to constrain the depths between pairs of points on the object's surface (problem (6.3), last line). These pairs can be any pairs of points, however they give the strongest

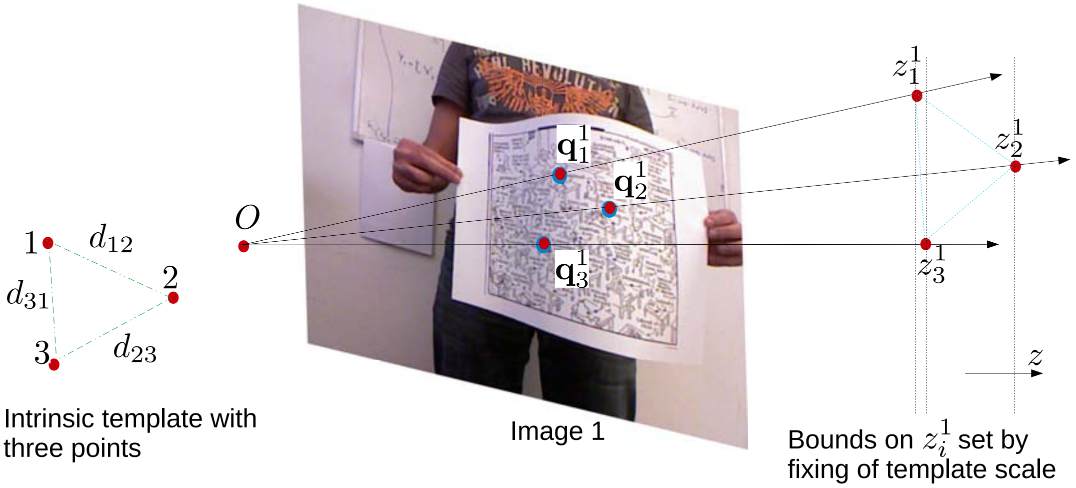


Figure 6.2: Illustration of the bounds set by equation (6.4) for NRSfM using three points and one image. This is a modification of figure 3.1 for NRSfM.

constraints when the points are close together on the surface. This is because for closer points the inextensibility inequalities become tighter. Of course, we do not know exactly which points are close together *a priori*. A good estimate can be made from the distance of the correspondences in the images, because nearby points on the surface tend to be close in the images. We denote the Euclidean distance between two points $\mathbf{q}_i^k$ and $\mathbf{q}_j^k$ in image k by δ_{ij}^k, and we use these to build the NNG. The specific algorithm we propose is as follows:

1. Compute distances $\{\delta_{ij}^k\} \quad \forall i \in \{1 \dots n\}$,
   $j \in \{1 \dots n\},\ k \in \{1 \dots m\}$, and $i \neq j$.

2. If the ith or jth point is not visible in image k, set $\delta_{ij}^k = -\infty$.

3. Take the maximum distance over the images.
   $\hat{\delta}_{ij} = \max_k\{\delta_{ij}^k\} \quad \forall i \in \{1 \dots n\},\ j \in \{1 \dots n\}$.

4. For each point i put into $\mathcal{N}(i)$ the points j with the K smallest values of $\hat{\delta}_{ij}$ $(j \neq i)$.

5. We find the connected components using each point index i and its neighborhood $\mathcal{N}(i)$. We reconstruct each component separately.

The above algorithm keeps only those points in a neighborhood that are close to each other in all the images. This implies that if a material is torn apart or an object splits, we treat them as separate objects. In that case, they could be reconstructed separately and the scale could be fixed after the reconstruction to merge them in images when they are a single object. The only parameter that needs to be selected here is the neighbourhood size K. Our method is not very sensitive to this parameter but a reasonable value (*e.g.*, 20) should be chosen depending on the density of the correspondences and required speed of optimization.

6.2.4 NRSfM with temporal smoothness

One potential application of NRSfM is to reconstruct surfaces from a video sequence of a deforming object. In such a setup, the surface points can be assumed to move smoothly over time. This can be expressed by replacing the maximization term in problem (6.3) with the following:

$$\underset{\{z_i^k\},\{d_{ij}\}}{\text{maximize}} \sum_{k=1}^{m}\sum_{i=1}^{n} v_i^k z_i^k - s_t \sum_{k=1}^{m-1}\sum_{i=1}^{n} \|v_i^{k+1} v_i^k (z_i^{k+1} - z_i^k)\|_1 \tag{6.5}$$

subject to the same constraints as in problem (6.3). The added term in problem (6.5) causes the depth values to change slowly between consecutive views, albeit with an added computational complexity. Many methods including (Salzmann and Fua, 2011a; Vicente and Agapito, 2012) use such first-order approach to impose temporal smoothness. However, using a large number of views (say, greater than 100) can increase the size of problem (6.3) making it very time consuming to solve. Using the formulation of problem (6.5) can make it possibly intractable in such situation. We introduce a different approach to impose temporal smoothness that attempts on reduction of the size of problem (6.3). We define temporal smoothness as the smooth evolution of depth over time and use uniform cubic B-splines to represent depth as a function of time. Thus for each 3D point over the time sequence, the unknown variables are the set of control points representing the evolution of depth in the sequence.

1-D uniform cubic B-splines. B splines can be used to parametrize an N-D function using weighting parameters known as the control points. We use a 1-D spline to parametrize the depth function $z_i(k) \in \mathbb{R}^+$. Note that it is a function of a single variable, i.e., the image number k. The spline is evaluated as a linear function of control points at each image, given by:

$$z_i^k = z_i(k) = \eta_k^\top \mathbf{w}_i, \quad i = 1 \dots n,\ k = 1 \dots m \tag{6.6}$$

where η_k is a function of time (image number) k and $\mathbf{w}_i$ is the vector of control points for the point i. Given that we use $m_c < m$ number of control points to represent each point depth on the surface, the set of control points is $\mathbf{w}_i = [w_1\ w_2 \dots w_{m_c}]^\top \in \mathbb{R}^{m_c}$. The lifting function η_k can be precomputed and is the same for all points on a surface. A good description of the lifting function and its computation can be found in (Brunet, 2010). For our purpose, it is a known sparse vector with at most 4 non-zero values and of the same size as the vector of control points. Using equation (6.6), we can rewrite the NRSfM problem in terms of the new unknowns as below:

$$\begin{aligned}
&\underset{\{\mathbf{w}_i\},\{d_{ij}\}}{\text{maximize}} \sum_{k=1}^{m}\sum_{i=1}^{n} v_i^k \eta_k^\top \mathbf{w}_i \\
&\text{s.t.} \\
&\eta_k^\top \mathbf{w}_i \geq 0 \\
&d_{ij} \geq 0 \\
&\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} d_{ij} = 1 \\
&v_i^k v_j^k \left\| \eta_k^\top \mathbf{w}_i \begin{bmatrix} \mathbf{q}_i^k \\ 1 \end{bmatrix} - \eta_k^\top \mathbf{w}_j \begin{bmatrix} \mathbf{q}_j^k \\ 1 \end{bmatrix} \right\|_2 \leq v_i^k v_j^k d_{ij} \\
&i = 1\ldots n,\ k = 1\ldots m,\ j \in \mathcal{N}(i).
\end{aligned} \tag{6.7}$$

We solve for the set of unknown control points $\{\mathbf{w}_i\}$ and the set of geodesic distances $\{d_{ij}\}$. The final depth values are obtained from equation (6.6) after the control points are obtained by solving problem (6.7). The total number of unknowns in problem (6.7) is thus $Kn+nm_c$ instead of $Kn+nm$. Usually we set $m_c < 0.3m$ and thus for a large problem this can result in a significant reduction of computation time with a negligible drop in accuracy.

6.3 MDH-based Robust NRSfM

The basic problem formulation presented in chapter 6.2 gives very good reconstructions when the input correspondences have no outliers. However in presence of a few outlier correspondences, they break down easily. One reason for it is that the method works globally, in the sense that all the constraints are used together to solve for all the depths in a single optimization. Thus the constraints at a point, for instance on the outlier point, can affect the solution of all other points. This is in contrast to local methods (Chhatkuli et al., 2014b) that solve the NRSfM problem one point at a time independently. Several strategies exist on dealing with outlier correspondences. Recovering inlier correspondences is most efficient with a dedicated outlier removal method such as (Pilet et al., 2008; Pizarro and Bartoli, 2012). However these methods often miss a few outlier points. Consequently, an outlier rejection strategy is necessary but not sufficient for the MDH-based NRSfM, as even a very few missed outliers can result in a completely wrong solution. We thus require a method that gives good reconstructions even in the presence of a small percentage of outlier image correspondences. In the SfT method (Ngo et al., 2016), the authors introduce an outlier removal strategy using a Laplacian framework. They then solve the final step of reconstruction using an iterative non linear refinement with slack variables to handle outliers. We here show that robustness with slack variables can be added into problem (6.3) without losing its convexity so that a global solution is obtained. We achieve robustness by introducing slack variables in the inextensibility constraint that can ‘capture’ outliers.

We introduce sets of scalar variables $\{a_i^k\}$ and $\{b_i^k\}$ for each point in each view so that the back projection function is:

$$\mathbf{Q}_i^k = \begin{bmatrix} a_i^k \\ b_i^k \\ 0 \end{bmatrix} + z_i^k \begin{bmatrix} \mathbf{q}_i^k \\ 1 \end{bmatrix}. \tag{6.8}$$

Equation (6.8) allows the optical rays from the corresponding point on image $\mathbf{q}_i^k$ to move in order to 'correct' for the outlier correspondences. We further assume that the first image correspondences are correct and thus no such correction is required on the first image. This is due to the nature of optical flow or point matching methods we use for experiments. Thus, we set $a_i^1 = 0$ and $b_i^1 = 0$. Given that only few of the points are actually outliers, a correct NRSfM solution should result in sparse sets of $\{a_i^k\}$ and $\{b_i^k\}$. We modify problem (6.3) to include equation (6.8) and perform an L1-minimization of the slack variables as below:

$$\begin{aligned}
&\underset{\{z_i^k\},\{d_{ij}\},\{a_i^k\},\{b_i^k\}}{\text{maximize}} \sum_{k=1}^{M}\sum_{i=1}^{N} z_i^k - s_r \sum_{k=1}^{M}\sum_{i=1}^{N} \left|a_i^k\right| + \left|b_i^k\right| + \left|x_i^k b_i^k - y_i^k a_i^k\right| \\
&\text{s.t.} \\
&z_i^k \geq 0 \\
&d_{ij} \geq 0 \\
&a_i^1 = 0, \quad b_i^1 = 0 \\
&\sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} d_{ij} = 1 \\
&\left\| z_i^k \begin{bmatrix} \mathbf{q}_i^k \\ 1 \end{bmatrix} + \begin{bmatrix} a_i^k \\ b_i^k \\ 0 \end{bmatrix} - z_j^k \begin{bmatrix} \mathbf{q}_j^k \\ 1 \end{bmatrix} - \begin{bmatrix} a_j^k \\ b_j^k \\ 0 \end{bmatrix} \right\|_2 \leq d_{ij} \\
&i = 1 \ldots N,\ k = 1 \ldots M,\ j \in \mathcal{N}(i).
\end{aligned} \tag{6.9}$$

The L1-minimization of the slack variables in the cost function in fact favors solutions where small corrections in the sightline are made for only some of the points, i.e., it favors sparse and small valued solutions for the set of slack variables $\{a_i^k\}$ and $\{b_i^k\}$. We now require a single hyperparameter s_r to balance the depth maximization w.r.t. the correction for outliers. The term containing the slack variables a_i^k and b_i^k in the maximization is chosen so as to minimize the correction of sightline. We found such an error measurement to give more favorable results compared to the reprojection error. Problem (6.9) is much better constrained than problem (6.3) when the image point correspondences have noise or outliers.

6.4 Experimental Results

6.4.1 Implementation details

We have implemented all of our methods[1] in MATLAB which uses the MOSEK (ApS, 2015) SOCP solver. MOSEK is faster than many other SOCP solvers, especially for large scale problems. All of the methods can be implemented in very few lines of code (25 to 35) with the YALMIP interface (Löfberg, 2004) for MATLAB. However we use our optimized interface to solve NRSfM for the proposed methods in favor of their speed. For example, we can solve with 60 images, 300 points and $K = 20$ in about 4 minutes in a standard 2012 desktop PC. This computation time is the fastest among the compared methods for the number of images and points considered. The robust version of the method takes about 13 minutes for the same problem. On the other hand, the method imposing temporal smoothness based on splines as in problem (6.7) takes only 130 seconds for the same task.

6.4.2 Method comparison and error metrics

We compare our results against five other methods whose source code is provided by the authors. We name our first NRSfM formulation that implements problem (6.3) and equation (6.4) as **tlmdh** and its robust version of problem (6.9) as **r-tlmdh**. We name the implementation of our NRSfM with temporal smoothness described by equation (6.5) as **t-tlmdh** and our NRSfM with temporal smoothness based on 1D splines as **s-tlmdh**. We name the non-convex soft inextensibility based method for orthographic camera (Vicente and Agapito, 2012) as **o-sinext** and the local homography method for perspective camera (Chhatkuli et al., 2014b) as **p-isolh**. We write the local method of (Parashar et al., 2016) based on the metric tensor as **p-isomet**. We name the prior free factorization method of (Dai et al., 2012) as **o-spfac** and the kernel based factorization method (Gotardo and Martinez, 2011) as **o-kfac**. We name the locally rigid method based on 3-point SfM (Taylor et al., 2010) as **o-lrigid**. Each method requires one or more parameters to be tuned. We fix these parameters to optimal values for each dataset and keep them constant for all experiments. For our methods we fix a single hyperparameter for all datasets. We set $s_t = 0.2$ for **t-tlmdh** and $s_r = 25$ for **r-tlmdh**. Similarly, we set the number of control points for depth in **s-tlmdh** to 20% of the total number of images.

We measure a method's accuracy with two metrics: 3D Root Mean Square Error (RMSE) and the %3D error often used in most NRSfM literature (Agudo and Moreno-Noguer, 2015). The 3D RMSE is computed from the ground truth 3D point positions. Because NRSfM has a scale ambiguity no method can reconstruct the absolute scale of the object. For methods which use perspective camera (**tlmdh** and **p-isolh**) we scale their reconstructions to best align them with the ground truth. For the methods which use affine cameras (**o-sinext**, **o-lrigid** and **o-spfac**), we transform their reconstructions with a similarity transform to best align them with the ground truth. The $\%$ 3D error is defined as follows:

$$\%\text{ 3D error} = \frac{\|\mathbf{P}_{GT} - \mathbf{P}_{REC}\|_{\text{fro}}}{\|\mathbf{P}_{GT}\|_{\text{fro}}} \tag{6.10}$$

[1] The optimized codes are available at http://isit.u-clermont1.fr/~ab/Research/

where $\mathbf{P}_{GT}$ represents the ground truth 3D shape ($3 \times n$ matrix) and $\mathbf{P}_{REC}$ represents the reconstructed 3D shape.

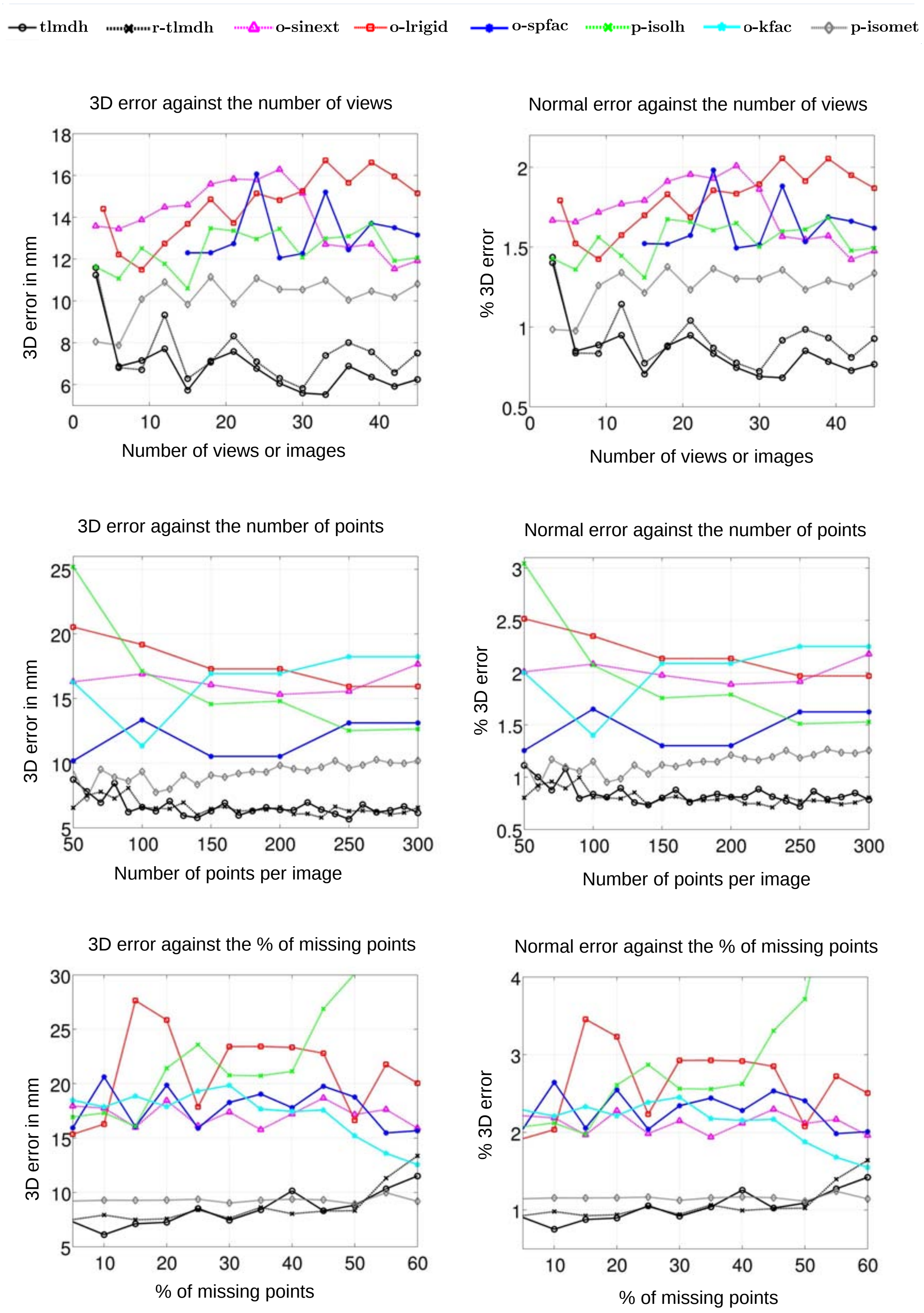


Figure 6.3: Plots for synthetic Flag dataset. The 3D errors are shown in the left column and the % 3D errors in the right column. Legend is shown on the top.

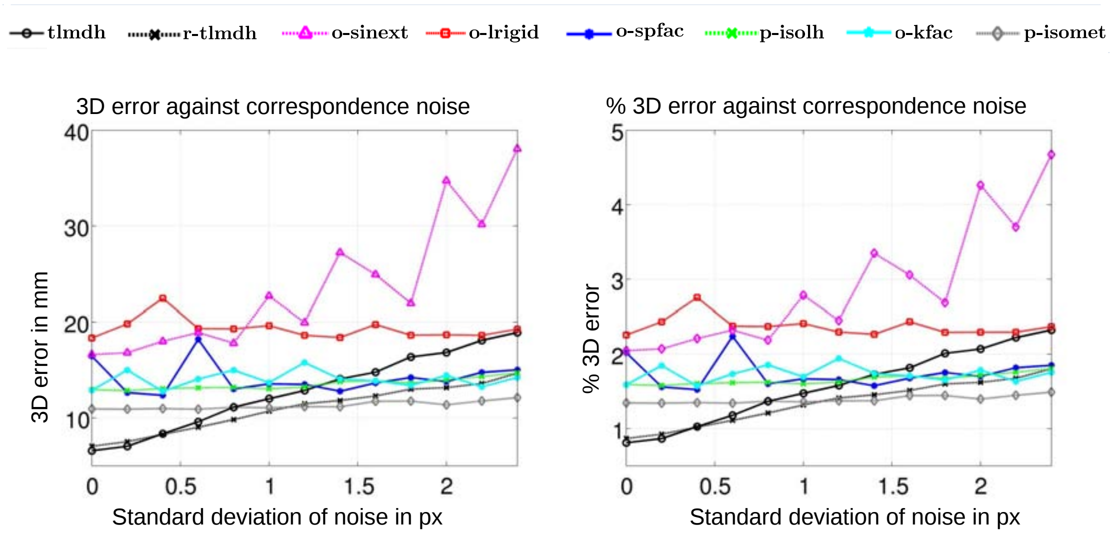


Figure 6.4: Plot of 3D error against noise in pixels. The 3D errors shown in the left column and the % 3D errors in the right column. Legend is shown on the top.

6.4.3 Developable Surfaces

Most non-rigid reconstruction methods focus on developable surfaces for experiments. A developable surface, such as a piece of paper or cloth, can be flattened into a planar surface without tearing or stretching. Obtaining continuous tracks of correspondences without partial images is relatively easy for such surfaces. While the surfaces often appear simple, they sometimes have high frequency and non-linear deformations. We experiment with 7 different datasets representing such surfaces.

The Flag dataset. We use the cloth capture data (mocap) (White et al., 2007) to generate semi-synthetic data. Even though the object is real, the input data for all the methods are generated from a virtual camera with perspective projection. The data shows a flag waving with wind with some changes in the camera viewpoint, making it perhaps the simplest of all datasets. The images are generated with dimensions 640 px × 480 px using a camera focal length of 640 px. The data has altogether 450 frames. We use this data to test the performance of our method and the competitive methods in several practical scenarios: with changing number of images, changing number of corresponding points and missing correspondences. For changing the number of images, we randomly draw a subset of m images from the 450 images with m varying from 5 to 60. For varying the number of points, we randomly select a subset of n points varying from 50 to 300. Finally, for varying the amount of missing correspondences for each image we randomly remove a percentage of correspondences ranging from 5 to 60. For the default conditions, we use 40 images, 300 points and no missing data. In order to fill the missing correspondences required by some methods we follow (Hu et al., 2013) for matrix completion. Note that our method **tlmdh** works with incomplete data and therefore we do not complete missing correspondences for our method. **p-isolh** computes registration functions with B-splines and so we use them to fill in the missing correspondences for that method. Figure 6.4 shows the plots for the dataset.

The results show that our method **tlmdh** performs very well with just 5 images and considerably better than all other methods. However, in high noise, **p-isomet** shows the best performance. Its use of the registration warps makes it robust to random noise to some extent. The same is true for high percentage of missing data. **p-isomet** also uses registration warps and performs very well in high noise. The factorization-based method **o-spfac** and the local homography based method **p-isolh** also does better compared to the remaining methods. We obtain an RMSE 3D error of 6.3 mm using 40 images. Similarly, it can be seen that our method is able to reconstruct the surface with as many as 60% random missing data. We also consider the effect of noise in correspondences and use our **r-tlmdh** method to show how it performs under correspondence noise.

The KINECT Paper dataset. We use the KINECT Paper dataset (Varol et al., 2012b) as one of our real datasets for evaluation, originally used for template-based reconstruction (Ngo et al., 2016). The dataset shows a VGA resolution sequence of a large piece of textured paper undergoing smooth deformations. Some example images were shown in figure 6.1 and 6.2. We generate correspondences by tracking points in the sequence using an optical flow-based method (Garg et al., 2013a) designed for non-rigid surfaces. The tracks are outlier free and semi-dense. Due to the large number of frames we again subsample them for all methods except **o-kfac**, which requires temporal continuity. Figure 6.5 shows the plots of 3D error and % 3D error for all the images in the dataset. We obtain very accurate reconstructions that in fact compares with template-based reconstructions (Chhatkuli et al., 2014a; Ngo et al., 2016). The best performing methods are **r-tlmdh**, **tlmdh** and **s-tlmdh** with mean 3D errors of 4.62 mm, 5.41 mm and 7.15 mm respectively. The local isometric method based on the metric tensor **p-isomet** is the best performing state-of-the-art method with 7.63 mm 3D error. The factorization-based methods: **o-kfac** and **o-spfac** have 3D errors of 13.93 mm and 14.66 mm respectively while **p-isolh** shows an error of 13.64 mm. The mean 3D and %3D errors for all methods in the dataset are given in table 6.1 and 6.2 respectively.

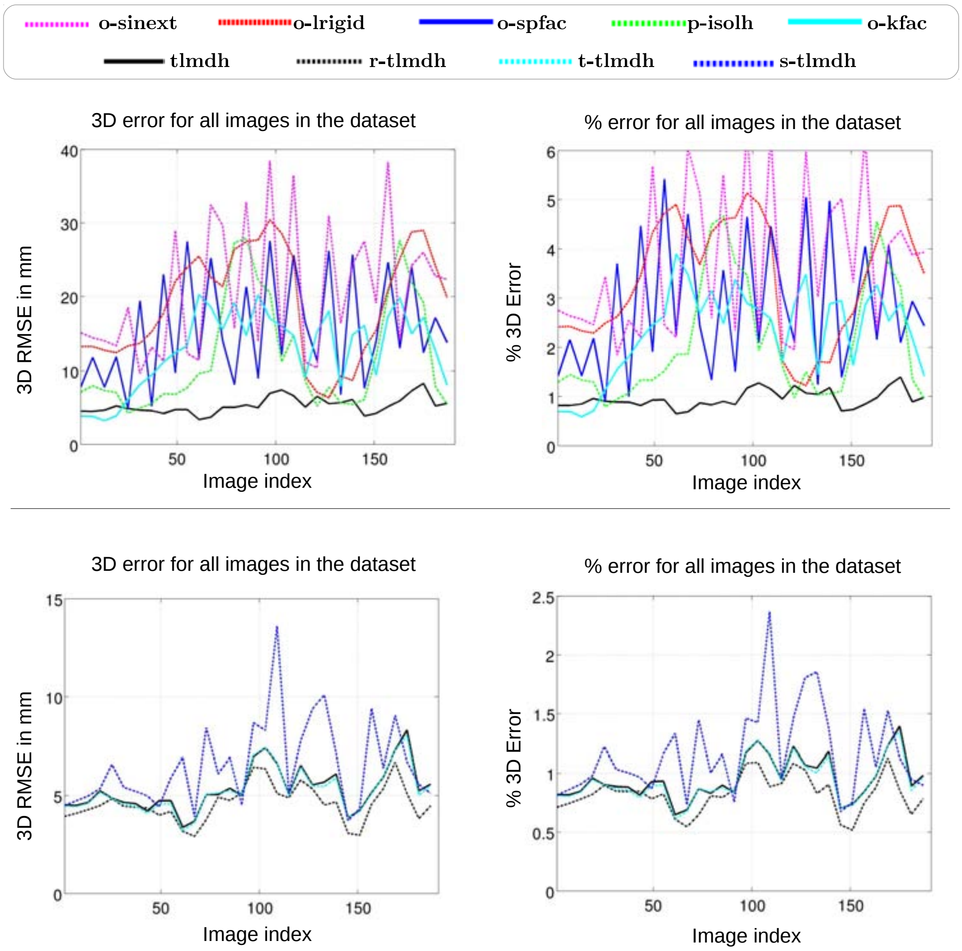


Figure 6.5: Mean 3D errors and percentage errors for all images in the KINECT Paper dataset. The top row shows errors for **tlmdh** against the compared methods and the bottom row shows **tlmdh** against the other proposed methods.

The Hulk and the T-Shirt dataset. The Hulk dataset (Chhatkuli et al., 2014b) consists of a comic cover printed on a piece of paper in 21 different deformations. Similarly, the t-shirt dataset (Chhatkuli et al., 2014b) consists of a textured t-shirt with 10 different deformations. We show a few example images of the dataset in figure 6.6. These datasets provide images with wide-baseline matches. We do not test the factorization-based methods on these datasets as they have very few images and also do not form a temporal sequence. Large number of images ($m > 3/2L$), where L is the number of shape basis here, are required by **o-spfac** and a continuous video sequence is required by **o-kfac**. We give the mean error results in table 6.1 and 6.2. The best performing methods are **tlmdh** and **r-tlmdh** with mean 3D errors of 3.51 mm and 3.45 mm for the hulk dataset; 5.41 mm and 5.39 mm for the t-shirt dataset respectively. Among the state-of-the-art methods, **p-isomet** shows the best performance with 10.76 mm and 10.60 mm error for the hulk and t-shirt datasets respectively. The next best performing method is **p-isolh** that gives a mean depth error of 14.53 mm and 8.94 mm for the Hulk and t-shirt datasets respectively.

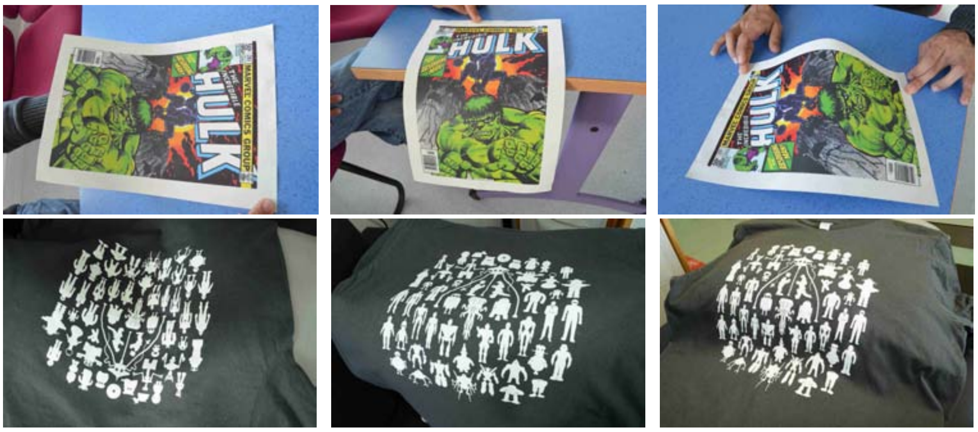


Figure 6.6: Example of images present in the Hulk dataset (top row) and the T-Shirt dataset (bottom row).

The Cardboard dataset. We construct a dataset using non-smooth deformations of a cardboard object. The dataset consists of 8 different deformations and images where the groundtruth 3D for each was obtained with stereo. The object used consists of repeating texture and large amount of texture-less regions. The images are taken with a focal length of about 3800 px and have a resolution of 4800×3200 px. We give some example images from the dataset in figure 6.7 below. We use a dense wide-baseline matching (Weinzaepfel et al., 2013) to compute correspondences between the images. The resulting correspondences are noisy and contains several outliers, more specifically in the texture-less regions. Among our methods we test only **tlmdh** and **r-tlmdh** as we do not have a temporal continuity in the dataset images. The performance of **r-tlmdh** is particularly noteworthy with 8.35 mm RMS error in contrast to 14.86 mm for **tlmdh**. The next best performing method is **p-isolh** with an RMS error of 10.02 mm. It handles the effect of outliers to some extent by the use of BBS spline-based registration. The local isometric method based on the metric tensor **p-isomet** failed to give any results for the dataset, possibly due to non-smooth surfaces and registration warps. Detailed results are provided in table 6.1 and 6.2. We also show a comparison plot using different number of images in figure 6.8.

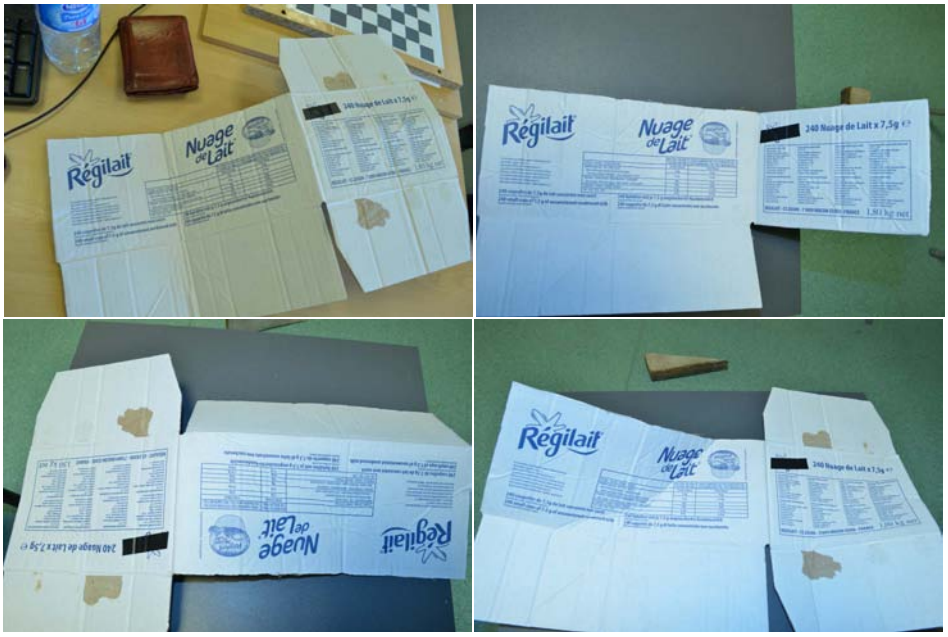


Figure 6.7: Example images from the Cardboard dataset.

The Rug and the Table-mat datasets. We make use of existing datasets used in (Parashar et al., 2016). The datasets are recorded with Kinect for X-box One and its images have a resolution of 1920×1080 px. They are taken with a focal length of 1054 px. Some example images for both the datasets are shown in figure 6.9. The Rug dataset shows a rug being deformed smoothly in 159 images, while the Table-mat dataset shows a table-mat being deformed smoothly in 60 images. The correspondences are provided with the ground truth and there are no missing correspondences. However, due to the low frame-rate of the recorded sequences, the correspondences provided are not very accurate and contain outliers. We show the comparison of the proposed methods with the state-of-the-art methods for all the frames in figure 6.11 for the rug dataset and figure 6.10 for the table mat dataset. We show the mean accuracy measures in table 6.1 and 6.2. We obtain the best results from **r-tlmdh** and **tlmdh** with 3D errors of 25.72 mm and 26.60 mm for the rug dataset; while for the table-mat dataset the compared method **p-isomet** shows the best performance with 9.6 mm compared to 14.80 mm and 16.91 mm for **r-tlmdh** and **tlmdh** respectively. We also obtain good results from **s-tlmdh** with a mean 3D error of 27.54 mm for the Rug dataset and 16.74 mm for the Table-mat dataset. The compared methods **o-spfac** and **o-kfac** have a mean 3D error of 31.01 mm and 34.62 mm for the Rug dataset; 17.51 mm and 16.25 mm for the Table-mat dataset. Note that the datasets are constructed with optical flow tracking on a very low frame rate sequence and thus all methods have a relatively high absolute mean error. Perhaps for the same reason, we failed to reconstruct the surfaces with **o-lrigid** using all the views. The proposed methods do not show the same level of accuracy as in the other datasets. This is also due to the relatively smaller viewpoint change and deformations present in these datasets.

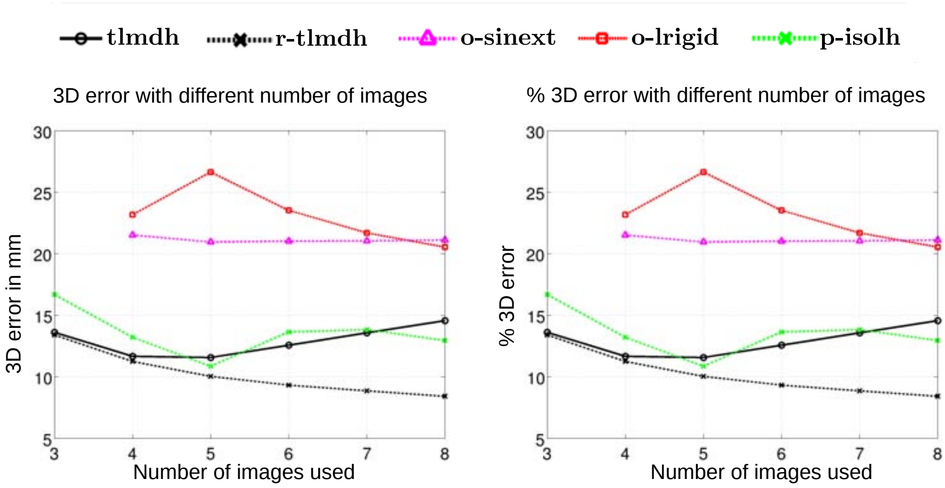


Figure 6.8: Mean 3D errors and percentage errors for different number of images in the Cardboard dataset.

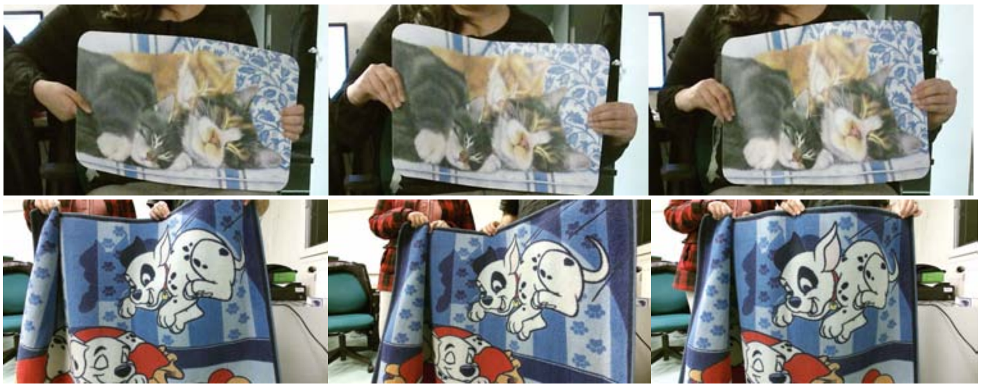
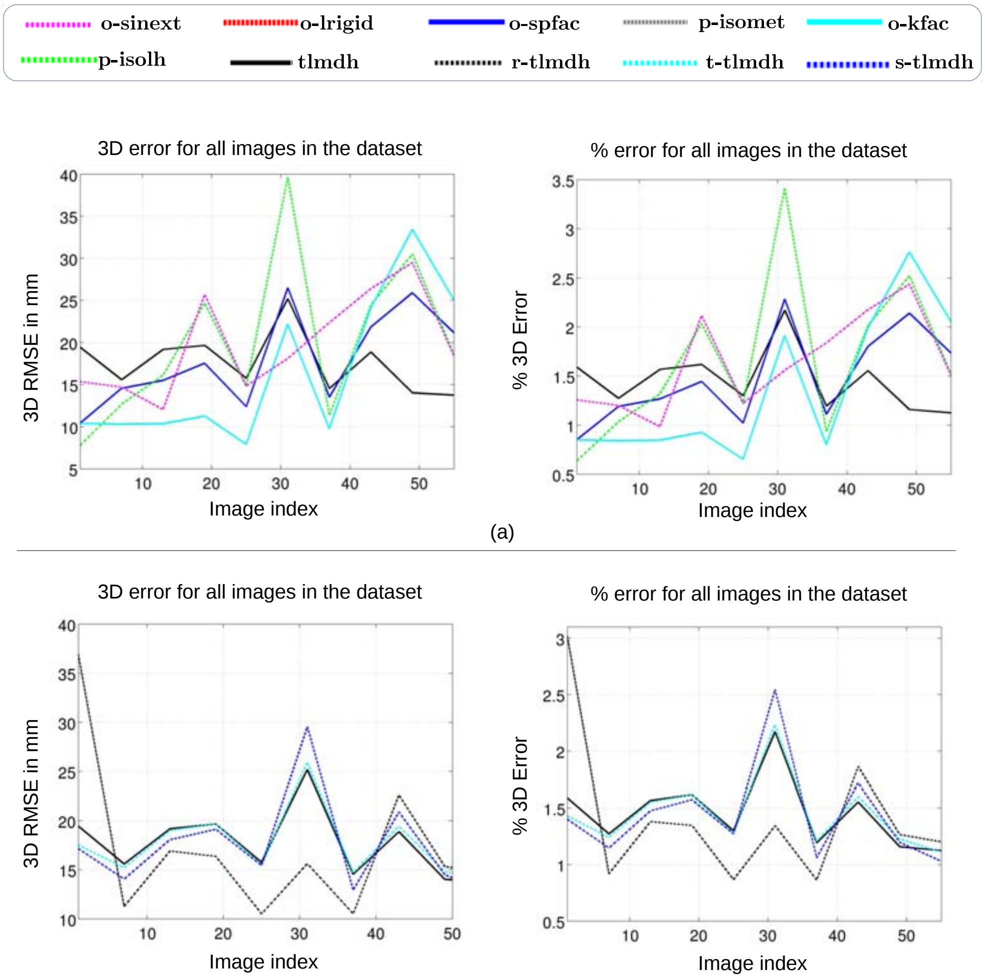


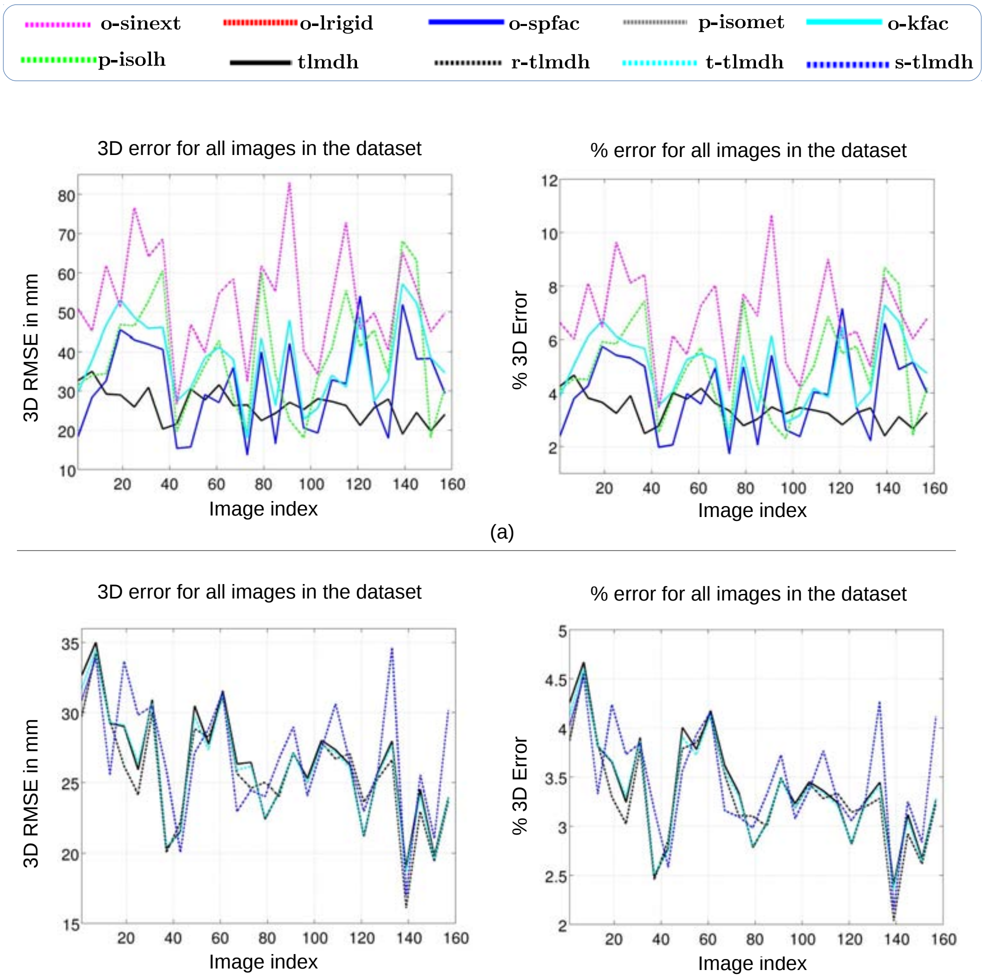
Figure 6.9: Example images for the Table-mat (top, cropped to the size of 592×349 px) and the Rug (bottom, original images) datasets.



(a)

(b)

Figure 6.10: Mean 3D errors and percentage errors for all the images in the Table-mat dataset. The top row shows errors for **tlmdh** against the compared methods and the bottom row shows **tlmdh** against our other methods.



(a)

(b)

Figure 6.11: Mean 3D errors and percentage errors for all the images in the Rug dataset. The top row shows errors for **tlmdh** against the compared methods and the bottom row shows **tlmdh** against our other methods.

Newspaper sequence. We construct a video sequence of a tearing piece of newspaper that consists of deformation as well as articulated movement. We record the sequence using KINECT for Xbox One at full frame rate using the libfreenect2 library (Xiang et al., 2016). The sequence has 460 images of resolution 1920 × 1080 px, taken at a focal length of about 1054 px. Some example images are shown in figure 6.12. We track points on the sequence again using dense point tracking (Sundaram et al., 2010). We randomly select 900 points that are tracked in all frames. Figure 6.13 shows the error plots of different methods for each image in the sequence. Table 6.1 gives the mean accuracy measure for different methods in the sequence. The results clearly show high accuracy of the proposed methods. The mean 3D errors for **tlmdh**, **r-tlmdh** and **s-tlmdh** are 11.63 mm, 11.62 mm and 13.35 mm respectively. The closest compared method **p-isomet** has a mean 3D error of 18.40 mm. **o-spfac** shows a 3D error of 24.94 mm. There are two important reasons the proposed methods work well in this dataset: first is that the point tracking gives very good set of correspondences here due to the higher frame rate of the dataset. More importantly, the tearing of the piece of newspaper and the articulated movement tend to produce a good amount of viewpoint change. These conditions, at the same time are difficult for the compared methods to handle.

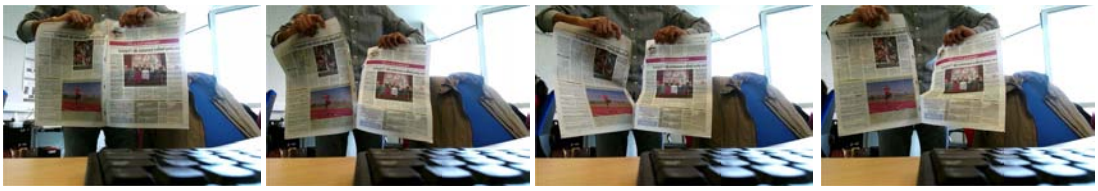


Figure 6.12: Example images from the Newspaper sequence.

Failure cases. Failure cases occur in NRSfM due to the problem being ill-posed due to lack of motion and deformation. Naturally any method would fail when the problem is ill-posed. However, a method can also fail to give good results with a well-posed problem. We found one such example for our method from (Salzmann et al., 2007). The dataset is a bending piece of paper imaged from a fixed camera viewpoint with a relatively longer focal length, and it contains no ground truth. We use optical flow (Sundaram et al., 2010) to obtain correspondences. The qualitative reconstructions for three frames are shown in figure 6.14. The general shape of the paper looks reasonable but in

Table 6.1: Mean 3D errors in real datasets.

	3D error measurements for different methods in mm							
Datasets	**tlmdh**	**r-tlmdh**	**p-isomet**	**p-isolh**	**o-spfac**	**o-kfac**	**o-sinext**	**o-lrigid**
KINECT Paper	5.41	**4.62**	7.63	13.64	14.66	13.93	21.45	18.65
Hulk	3.51	**3.45**	10.76	14.54	22.98	-	26.37	24.20
T-Shirt	5.41	**5.39**	10.60	8.94	-	-	18.23	-
Cardboard	14.56	**8.43**	-	12.95	-	-	35.34	20.54
Rug	26.60	**25.72**	26.15	38.26	31.01	34.62	49.14	-
Table mat	16.91	14.80	**14.21**	20.71	17.51	16.24	19.15	-
Newspaper	11.63	**11.62**	18.40	37.21	24.94	30.74	31.01	30.74

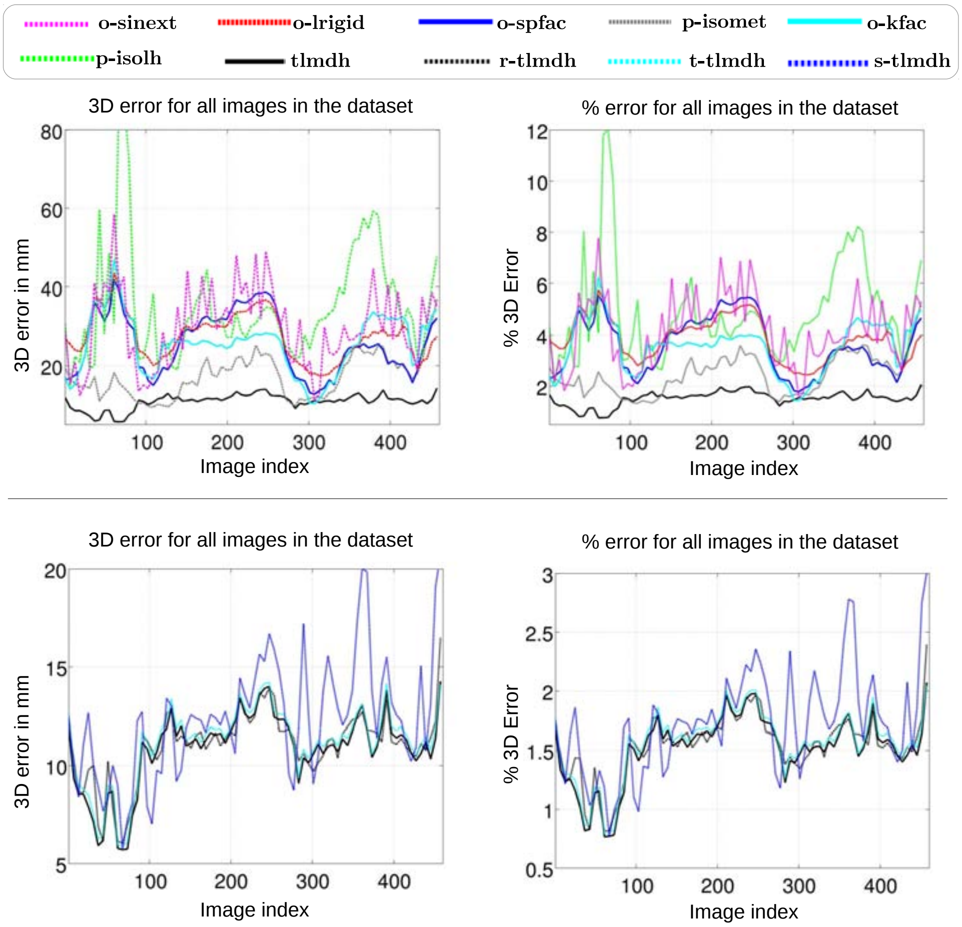


Figure 6.13: Mean 3D errors and percentage errors for all the images in the Newspaper sequence. The top row shows errors for **tlmdh** against the compared methods and the bottom row shows **tlmdh** against the other proposed methods.

the first image it is bent when it should be flat and the degree of bending is not properly captured in the second image. We know that better reconstructions are possible on this dataset (Vicente and Agapito, 2012), so the problem is not itself ill-posed. The imperfect reconstruction from our method is probably caused by the lack of change in camera viewpoint.

Table 6.2: Mean % 3D errors in real datasets.

	% 3D error measurements for different methods							
Datasets	**tlmdh**	**r-tlmdh**	**p-isomet**	**p-isolh**	**o-spfac**	**o-kfac**	**o-sinext**	**o-lrigid**
KINECT Paper	0.97	**0.83**	1.38	2.37	2.64	2.49	3.82	3.30
Hulk	**0.62**	**0.62**	2.81	4.17	5.10	-	5.82	5.31
T-Shirt	**1.69**	**1.69**	3.32	3.11	-	-	5.45	-
Cardboard	3.49	**2.06**	-	3.22	-	-	9.11	4.94
Rug	3.41	**3.30**	3.35	4.90	3.98	4.45	6.30	-
Table mat	1.40	1.22	**1.17**	1.71	1.45	1.34	1.58	-
Newspaper	**1.63**	**1.63**	2.63	5.20	3.50	4.24	4.34	4.31

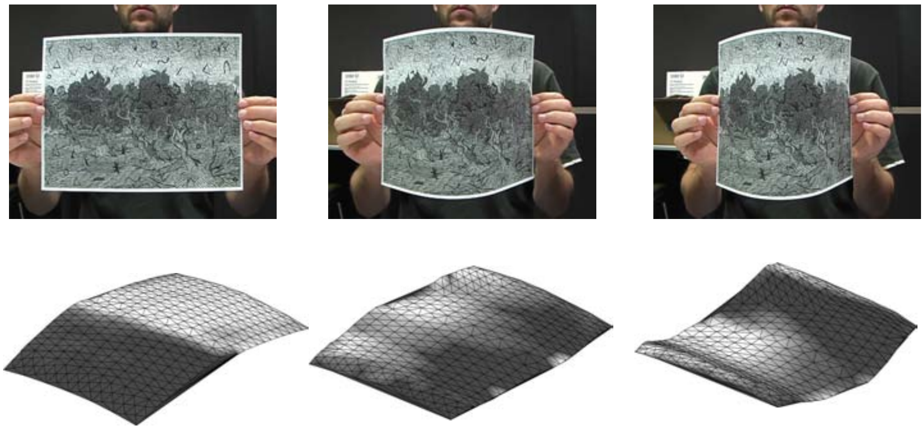


Figure 6.14: Failure cases: Images (top row) and their respective reconstructions (bottom row). The first two shapes appear largely incorrect.

6.4.4 Non-developable objects

We use two different datasets to perform NRSfM on non-developable surfaces. They are complex objects where some of the compared methods are not even applicable, for example, both **p-isolh** and **p-isomet** requires registration warps, which is non-trivial to implement in volumetric objects. We perform experiments here to show what we can obtain in highly difficult non-rigid reconstruction applications with our proposed **tlmdh** method. Below we describe the datasets and the experiments performed.

The Stepping Trousers dataset. The dataset (White et al., 2007) is constructed from motion capture ground truth data with perspective projection. The data shows a pair of trousers stepping around with considerable rapid deformations of the cloth. The images are obtained at a resolution of 640 px × 480 px with a perspective camera of focal length 320 px. The dataset is semi-synthetic but due to articulations, volume/partial views and rapid nonlinear deformations, it is arguably the most complex data used for NRSfM to date. Unlike the flag dataset, missing correspondences are significant due to self-occlusions. The missing correspondences are handled by filling in the correspondences using (Hu et al., 2013) for all methods except ours. Figure 6.15 shows three reconstructed frames. From top to bottom, it shows our best reconstruction, a reconstruction with medium accuracy and our worst reconstruction. Alongside we show the reconstructions for the compared method **o-spfac**. Note that it is non-trivial to implement the compared methods in the missing data scenario without using a low-rank prior. Thus we only test the best performing low-rank method **o-spfac**. The plots of 3D error for each image for these two methods are shown in figure 6.16. Because this is a large object, the 3D RMSE error can be large, yet the reconstructions can appear reasonable. We therefore also measure accuracy with a %3D error. We obtain a mean 3D error of 22.54 mm and % 3D error of 2.37% for our method while for **o-spfac** those are 51.5 mm and 11.56%respectively. Our results indeed show that large objects with complex deformations in small scale can be reconstructed with our method, although some difficulties can be seen primarily due to high surface curvature. The reconstructions and the plot show that our method can capture a large portion of the deformations correctly even though the parts of the object undergoing deformation are very small in the image, making the projections almost affine. In certain cases, however, it estimates the shapes incorrectly on those parts as shown in the third reconstruction of the sequence in figure 6.15.

The hand dataset. In tasks such as gesture recognition, several applications require reconstructing a moving hand. When such a task is done, usually a specialized modelling of hand motion and its articulations is used. We show that an accurate reconstruction of a deforming hand can be done solely with the inextensibility prior using our method. We test with two sequences of a deforming hand recorded by an endoscopic camera. The camera images are of dimensions 960×540 px, taken with a focal length of 462 px and capture detailed texture. We obtain ground truth reconstructions of the first and last frame using stereo and post processing. We compute correspondences by densely tracking the hand's texture using (Sundaram et al., 2010). Note that the correspondences are not perfect due to image noise and weak texture. Because most methods cannot handle a huge number of points, we uniformly subsample to 1000 points. Figure 6.17 shows reconstructions of the hand compared to ground truth for our method, **o-spfac**, **p-isolh** and **p-isomet**. The results show that our method can handle complex deformations of a hand. All three compared methods were unable to capture the second deformation where they have a 3D error of over 30 mm. On the other hand we obtain a slightly higher 3D error of 7.38 mm in the third column.

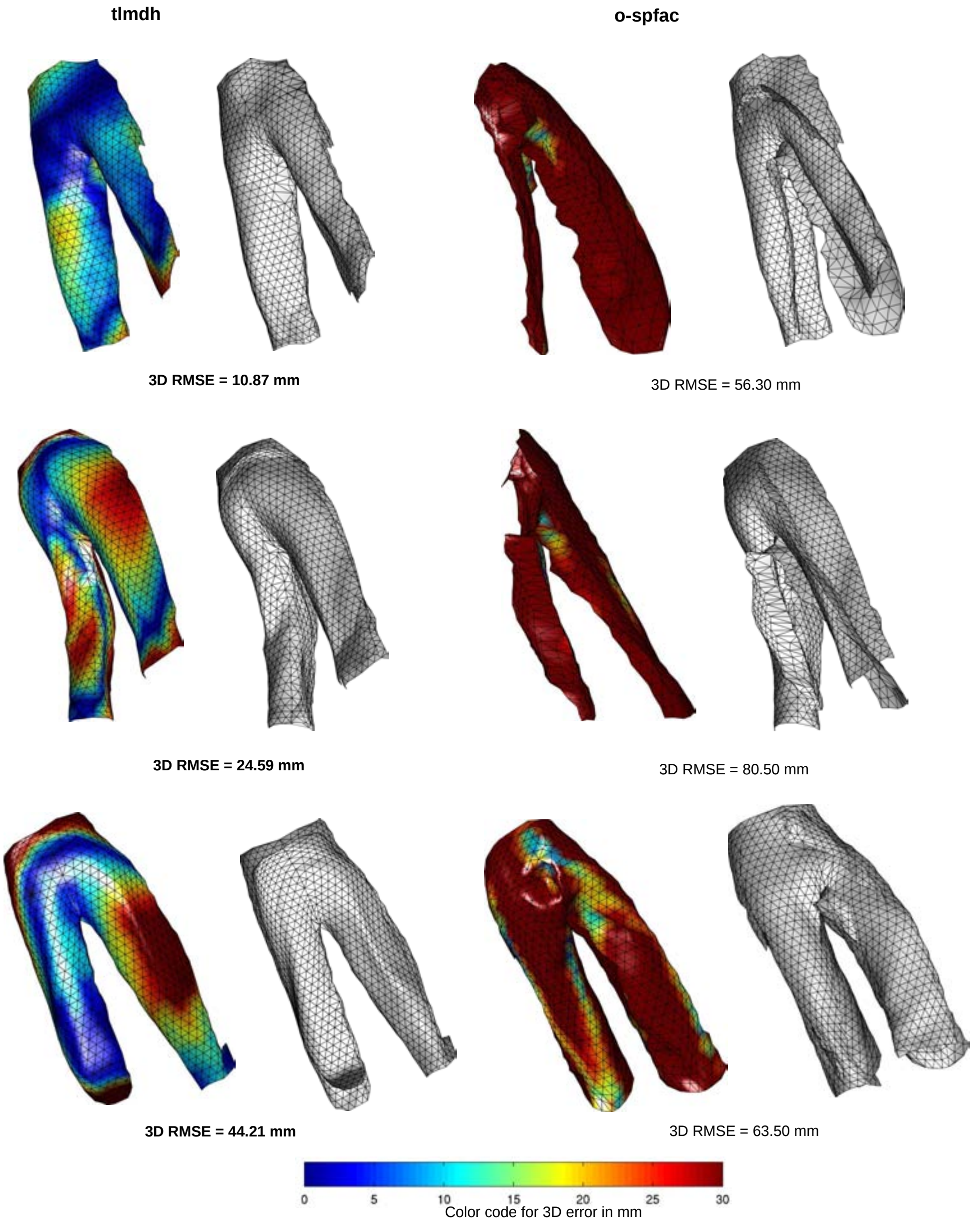


Figure 6.15: Reconstructions of the stepping trousers dataset for our method and **o-spfac**. Top row shows the reconstructed meshes overlaid on top of the ground truth. Bottom row shows the reconstructed mesh texture mapped with 3D error for each face in the color code shown. Note that we show our best result in the first column and the worst in the last column with a medium accuracy result in the middle.

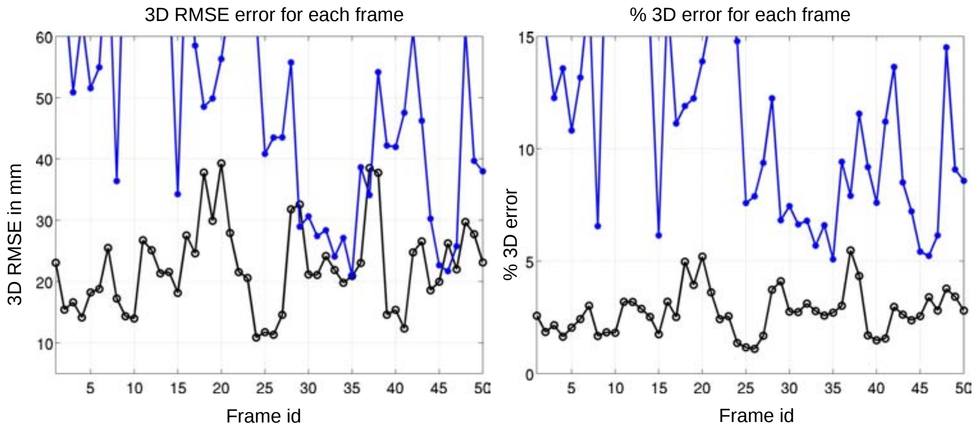


Figure 6.16: Plot of the depth error in trousers for each sampled image (**tlmdh** in black and **o-spfac** in blue).

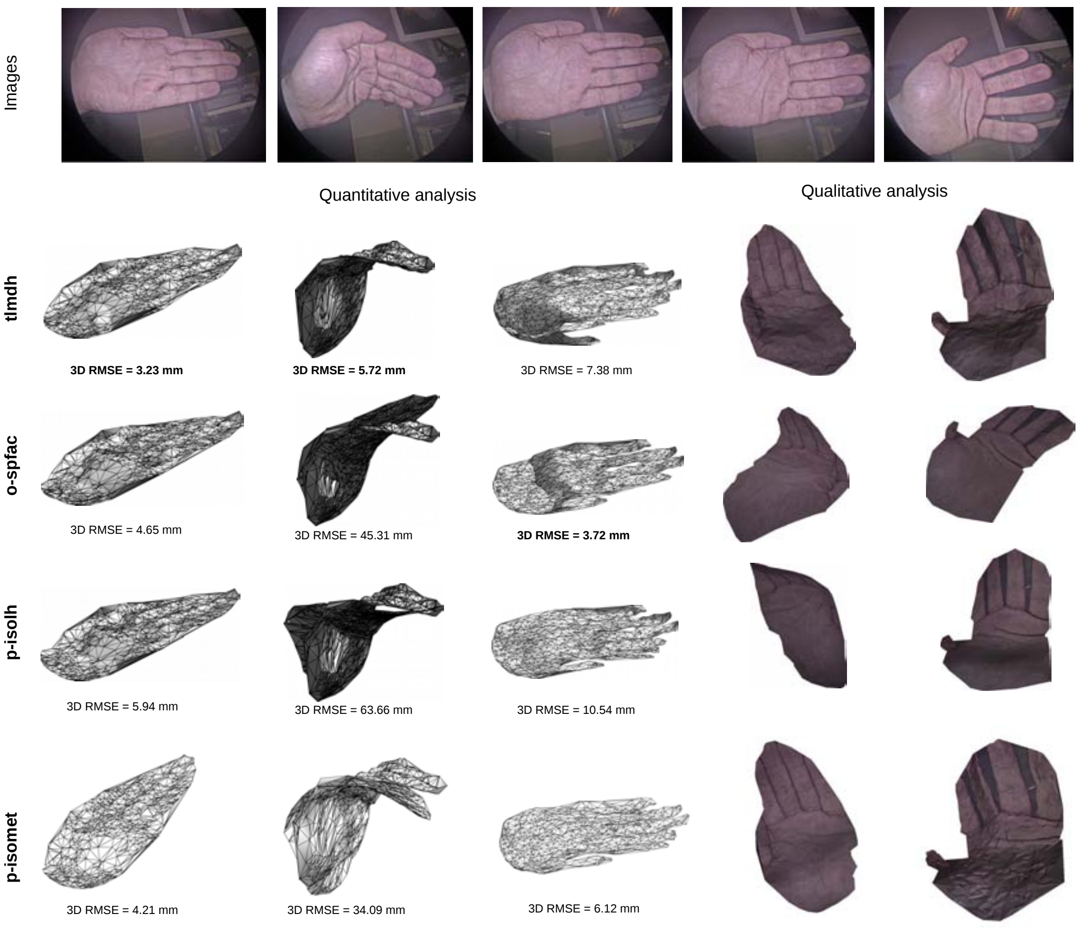


Figure 6.17: Results on the hand dataset. We use the best performing methods in other datasets for comparison: **o-spfac**, **p-isolh** and **p-isomet**. Ground truth is shown for three images, overlaid on top of the reconstructions. We texture map the meshes and show qualitative results for the two other images where ground truth 3D is not available.

6.4.5 NRSfM with rigid objects

All rigid objects are isometric, therefore our NRSfM method can be used to reconstruct rigid scenes. However isometry is weaker than rigidity, so it can be expected to perform slightly worse. Nonetheless it is interesting to study such cases for two reasons. First our method gives a convex solution to the problem with a general number of images, which has not been seen before in rigid SfM with perspective cameras. It may therefore find uses for initialising rigid bundle adjustment. The second reason is for a theoretical understanding of our method using rigid scenes, which may be simpler to analyse than for deformable scenes. For example, it may be interesting to study the critical motions associated with the inextensibility relaxation. We show some results from the public dataset (Jensen et al., 2014) on the house sequence using SIFT correspondences. We plot the average % 3D error for each of the 49 images for our method and compare this to a state-of-the-art rigid SfM method (VisualSfM (Wu, 2013)). We see that a reasonable error is obtained for the majority of the images.

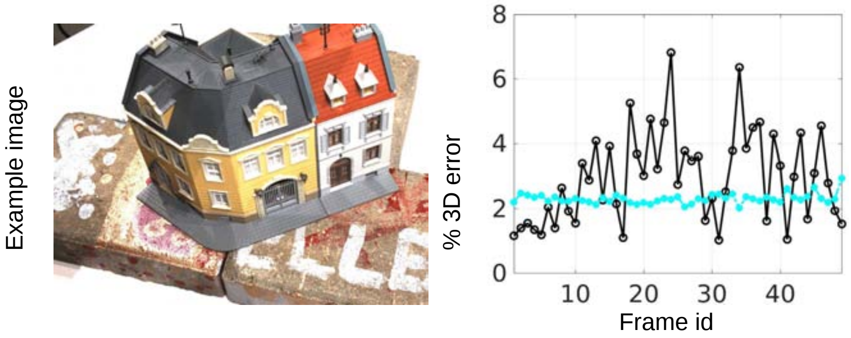


Figure 6.18: Results on rigid scenes. VisualSfM results are shown in cyan dots.

6.5 Discussions

We presented four different convex formulations for solving NRSfM. The first formulation presented in problem (6.3), named **tlmdh** should be the method of choice when the point correspondences for different images have no outliers and small noise. The robust formulation **r-tlmdh**, like **tlmdh** works with wide baseline large deformations and as few as four images, albeit with an added computational cost. Both of these methods show very good performance in the experiments. However, we found that the method **t-tlmdh** of using first-order temporal smoothness as described in problem (6.5) provides no real improvement over the original problem. The 1D spline-based method **s-tlmdh** on the other hand, gave significant reduction in the size of the problem. It is interesting to note that enforcing temporal smoothness does not usually improve the resulting reconstruction because the original problem (6.3) is already well constrained. Similarly, in case of no outliers, the solution of problem (6.9) is similar to that of problem (6.3). In regard to the computational complexity of solving these problems, the worst case scenario is $O(n^3)$ per iteration where n is the number of unknowns and we require about 30 iteration to solve any problem. However, the sparsity of the problem means the actual computational complexity is much lower than $O(n^3)$ per iteration.

6.6 Conclusion

We have brought forward the MDH-based formulation, which has enjoyed great success in inextensible template-based reconstruction, to the more general problem of templateless non-rigid reconstruction known as NRSfM. We have shown that this leads to a convex formulation, which can be solved globally and optimally as an SOCP problem. This forms the first convex, global and optimal NRSfM formulation based on physical constraints. Results on synthetic and real images have shown that the proposed methods outperform existing ones by a large margin in many cases. In future work, we plan to study alternative relaxations of isometry apart from inextensibility. It may also be possible to formulate our approach into a sequential or incremental NRSfM so that realtime performance can be achieved. In the next chapter we give a broad perspective of all the works presented in the thesis as well as directions for possible future works.

Chapter 7

Conclusion and Future Work

In this thesis we described our contributions in feature point based approaches to the 3D reconstuction of deformable surfaces. We formulated our methods based on the isometric prior because most real object deformations can be modelled as being isometric. Strictly speaking, these deformations are near-isometric rather than exactly isometric. However, we found in our work that the isometric prior when used in our formulations can handle many real world surface deformations. In rigid SfM, there now exists methods that give a dense 3D reconstruction starting from a rough camera calibration and images of the scene. Several software and hardware products perform 3D reconstruction using these methods. They first compute feature points and then use the established feature point correspondences to pose the geometric constraints. Having point correspondences makes it easier to express the reconstruction problem geometrically in both rigid and deformable cases. This is also the reason a large majority of the deformable reconstruction methods rely on feature points. Furthermore, current progress in optical flow and feature point matching gives a strong motivation to have a reconstruction step relying on feature points. However, there is still a lacking of robust and accurate methods that can solve real world problems in deformable 3D reconstruction. The methods we proposed in the thesis provide a step forward in achieving good reconstructions in well textured surfaces.

Specifically, we studied two different problems of deformable 3D reconstruction. The first problem, Shape-from-Template (SfT) is nearing real-world applicability. Nearly all of the state-of-the-art methods use one or other forms of the isometric prior. The second problem is the more general Non-Rigid Shape-from-Motion where we do not have a 3D template as in SfT. There is still a significant interest in various ways of modelling the deformation for NRSfM. Below we describe our contributions and possible directions for future work for each problem.

7.1 Shape-from-Template

We exploited the differential formulation first proposed by (Bartoli et al., 2015) and established important results concerning the stability of the proposed local solutions. In summary, we found that the local solution for depth is not stable at near affine conditions. With experiments we also showed that inextensibility-based methods do not remain stable in such conditions. In order to be robust in near-affine conditions, we proposed two similar methods based on the local solutions for the depth gradient and surface normal, which are first-order differential quantities. We discovered that the reconstruction obtained from the solutions of these quantities are better and more stable than the direct computation of depth. Our proposed methods work with smooth thin-shelled surfaces. Adapting the method to non-smooth or volumetric objects is non-trivial but can be done with nonlinear optimization. One other direct improvement that can be introduced here is by the use of better registration functions. It has been shown that Schwarps (Pizarro et al., 2016) can provide better registration derivatives. However, one important issue with our proposed method is that it cannot be easily adapted for large non-isometric deformations. It remains to be seen how first-order isometry can be imposed in a robust way so that non-isometric deformations can be better modelled. One more work that could be done in near future is to perform joint tracking and reconstruction using the local analytic solutions.

7.2 Non-Rigid Shape-from-Motion

We presented our local as well as global formulations for solving NRSfM. We studied the local solutions for NRSfM using the isometric prior. However, we found that the first-order constraints were not enough to compute the non-holonomic local solutions. Thus we opted for finding a homography at each point using the second and lower order quantities of the registration warps. For that purpose, we assumed isometry and infintesimal planarity of the surfaces. Such approach works in several cases of sparse sets of images with large deformation scenes. However newer and more complete differential modelling has been proposed recently that should be the preferred approach for obtaining local solutions of NRSfM. The method (Parashar et al., 2016) models the change in metric tensor using the first-order and second-order surface deformation properties.

We also presented our global formulation of NRSfM that is convex and performs better than the previous methods including our own local solution. We relaxed the isometric constraint to zeroth-order inextensibility and combined all camera projection and the inextensibility constraints into a single optimization. The problem formulation was inspired from a well-established approach in template-based methods of maximizing depth under the inextensibility constraints. The formulation is flexible enough to impose temporal smoothness and add robustness in the problem. The methods proposed here work with rigid as well as non-rigid scenes with the point set representation of surfaces and registration. However, similar to the inextensibility-based methods in SfT, the proposed NRSfM methods are also not well-constrained in near-affine cases. One simple way to tackle such cases is by using a nonlinear refinement similar to (Vicente and Agapito, 2012) or (Brunet et al., 2014) starting with our solution as an initial solution.

Last but not the least, other possible directions that could be explored in the 3D reconstruction of deformable surfaces are the use of learning methods. As discussed in chapter 3, the statistics-based methods already use linear bases that can be learned and some work has already been done in combining physical model and the linear bases. On the other hand, several learning methods based on the Convolutional Neural Network (CNN) are attempting the problem. One direct approach to use CNN is to use the registration warps seen in chapter 4 and 5 as input images to the CNNs. In that case the network could be trained using simulated point correspondences. It is also important to understand the difficulties and limits of the direct approaches of learning point depths directly from images. Using the geometric structure of deformable 3D reconstruction as explored in the thesis, alongside a large data learning-based method could possibly provide an accurate and robust solution to the problem.

Appendices

APPENDIX A

Computation of θ and ω in SfT

We need to find $\theta \in C^0(\Omega', SO_3)$ such that it satisfies equation (4.52):

$$\begin{bmatrix}\mathbf{I}_2 & -\frac{\eta'}{f}\end{bmatrix}\theta = \begin{bmatrix}\omega & \mathbf{0}\end{bmatrix}$$

for some $\omega \in C^0(\Omega', \mathbb{R}^{2\times 2})$. Equation (4.52) implies that the third column θ_3 of θ is orthonormal to both rows of $[\mathbf{I}_2 \; -\frac{\eta'}{f}]$. Note that we represent the ith column of the matrix function θ as θ_i. This gives us a closed form solution for θ_3 as:

$$\theta_3 = \frac{\tilde{\eta}'_f}{\|\tilde{\eta}'_f\|_2}, \tag{i}$$

where $\tilde{\eta}'_f \in C^1(\Omega', \mathbb{R}^3)$ is the local flat template-to-image warp function giving *normalized* homogeneous coordinates. Its components can be written as $\tilde{\eta}'_f = [\frac{\eta'_x}{f} \; \frac{\eta'_y}{f} \; 1]^\top$. The columns of θ will have the following orthonormality relations:

$$\begin{aligned}[\theta_3]_\times \theta_1 &= \theta_2 \\ [\theta_3]_\times^\top \theta_2 &= \theta_1.\end{aligned} \tag{ii}$$

Any combination of vectors θ_1 and θ_2 that satisfy equation (ii) will form the required rotation θ. First we can choose θ_1 orthogonal to θ_3 as:

$$\theta_1 = \frac{1}{\sqrt{\frac{\eta_x'^2}{f^2} + 1}} \begin{bmatrix} -1 \\ 0 \\ \frac{\eta'_x}{f} \end{bmatrix}. \tag{iii}$$

Using the values of θ_3 and θ_1, we obtain θ_2 using $\theta_2 = \theta_3 \times \theta_1$. We obtain ω by simply expanding the left-hand side of equation (4.52) using the computed θ.

APPENDIX B

Similarities and differences of type-I and type-II PDEs in SfT

We discussed the type-I and type-II PDEs, their solutions and the stable methods in sections IV and V. Here we first show that the type-I direct depth solution and type-II direct depth solution are exactly the same. Then we describe how the stable type-I and stable type-II methods can differ in their final reconstructions.

Equivalence of type-I and type-II solutions The first non-holonomic solution of type-I PDE α and type-II PDE φ_z are given by equation (4.20) and equation (4.55) respectively. The type-I and type-II PDEs are obtained using the same deformation and reprojection constraints. With the equivalence of type-I and type-II PDEs, the non-holonomic solutions are directly related by the change in variable rewritten below:

$$\varphi_z = \frac{\alpha}{\nu} \quad \text{and} \quad \nu = \sqrt{1 + \frac{\eta^\top \eta}{f^2}}. \tag{iv}$$

The quantity ν involved in the change of variable is a known quantity and thus it proves that the direct depth reconstructions using type-I and type-II PDEs are the same.

In order to prove the equivalence of the second non-holonomic solutions, we consider the case of flat template, where we have $\mathsf{J}_\phi = \mathsf{J}_\varphi$. The result can be extended to non-flat templates by using the locally isometric flattening. We differentiate equation (iv) on both sides that gives us:

$$\mathsf{J}_\alpha = \nu \mathsf{J}_{\varphi_z} + \varphi_z \mathsf{J}_\nu. \tag{v}$$

Furthermore we already have the relation between J_{φ_z} and J_φ from equation (13) in section IV-A, rewritten below:

$$\mathsf{J}_\varphi = \mathbf{M}\tilde{\eta}\mathsf{J}_{\varphi_z} + \varPhi \mathsf{J}_{\tilde{\eta}}. \tag{vi}$$

Thus substituting the value of J_{φ_z} from equation (v) in equation (vi), we have:

$$\mathsf{J}_\varphi = \mathbf{M}\tilde{\eta}\frac{(\mathsf{J}_\alpha - \varphi_z \mathsf{J}_\nu)}{\nu} + \varPhi \mathsf{J}_{\tilde{\eta}}. \tag{vii}$$

In other words, one can go from the type-I set of solutions for $(\alpha, \mathsf{J}_\alpha)$ to type-II set of solutions for

$(\varphi_z, \mathsf{J}_\varphi)$ using equations (iv) and (vii). Similarly to go from type-II set of solutions to type-I set we note the fact that J_{φ_z} is given by the third row of J_φ and thus obtain $(\alpha, \mathsf{J}_\alpha)$ by using equations (4.15) and (v).

Difference in stable type-I and stable type-II reconstructions As shown above, the non-holonomic solutions of the type-I and type-II PDES are equivalent up to some change in variable. Nevertheless, the second non-holonomic solutions measure different quantities. This naturally leads to differences in the subsequent steps: the resolution of the two-fold ambiguity and the numerical integration. The second non-holonomic solution of type-I PDE β or J_α has a sign ambiguity, whereas that of type-II PDE: J_ϕ or $\mathbf{n}$ has a two-fold ambiguity. The resolution of ambiguities in these two cases can produce outcomes that influence the further steps differently. More importantly, the numerical integration involved in the steps differ as the integrand in stable type-I method β is a 2-vector while the integrand in stable type-II method $\mathbf{n}$ is a vector describing the shape normal, *i.e*. a 3-vector. Noise can influence the numerical integration in the different spaces differently. The numerical integration is done by using LLS with the bending energy of BBS. It enforces smoothness on the integrand's space. Such smoothness is better suited on the space of surface normals than on the radial depth gradient where the norm of the image coordinates is involved. Thus the small but consistent difference seen in the reconstructions is originate from the influence of noise in the sequence of steps after the non-holonomic solutions are found and not from the stable PDE solutions themselves.

APPENDIX C

SfT in the presence of outliers

We use a sugar bottle with a non-developable 3D template to test how outliers affect the reconstruction and how they can be dealt with. The bottle is used to create a deformation which we use to test the effect of outliers on our best performing method, *i.e.* **typeII**-P. We take the images with the same camera focal length as the Can dataset shown in figure I. We combine SIFT and KAZE features to obtain the template image to input image warp. Outliers appear naturally in the matching due to deformation and repeated texture. We show that with outliers, SfT will fail if a standard BBS registration (Dierckx, 1993) is used, while on the other hand using an automatic outlier removal method (Pilet et al., 2008; Pizarro and Bartoli, 2012) followed by a robust registration can suffice to have a good reconstruction.

Registration under outliers

We briefly describe the registration problem when there are outliers in the corresponding points. Let $\{\mathbf{p}_i \in \mathbb{R}^2\}$ be the set of corresponding points in the template image and $\{\mathbf{q}_i \in \mathbb{R}^2\}$ be the set of corresponding points in the input image obtained from feature point matching, with $i = 1 \dots N$. We define the BBS registration function going from the template image to the input image as:

$$\Psi(\mathbf{p}_i, \mathbf{l}) = \mathbf{q}_i \tag{viii}$$

where $\mathbf{l} \in \mathbb{R}^{2n_c}$ is the parametrization vector of the warp function Ψ and n_c is the number of the control centers. The warp is estimated from the correspondence points by minimizing the following least-squares problem:

$$\bar{\mathbf{l}} = \arg\min_{\mathbf{l}} \sum_{i=1}^{N} \|\Psi(\mathbf{p}_i, \mathbf{l}) - \mathbf{q}_i\|_2. \tag{ix}$$

For BBS registration the function Ψ is a linear function given by $\Psi(\mathbf{p}_i, \mathbf{l}) = \mathbf{A}\mathbf{l}$ where $\mathbf{A}$ is a matrix of basis vectors constructed from the set of points $\{\mathbf{p}_i \in \mathbb{R}^2\}$ by imposing smoothness and is a known matrix. Thus equation (ix) is in fact a LLS problem for a standard BBS registration. However the presence of outliers in the correspondence points means one cannot correctly estimate $\mathbf{l}$ with the LLS problem of equation (ix). We can remove a large number of outliers using existing methods such as (Pilet et al., 2008; Pizarro and Bartoli, 2012). This allows us to compute $\mathbf{l}$ using the remaining set of correspondences. However because these methods are not perfect, a small number of outliers may remain. To tackle this, we first employ the outlier rejection method (Pizarro and Bartoli, 2012) and fit $\mathbf{l}$ with a robust cost function, otherwise known as an M-estimator. Specifically we use the L1 M-estimator. Thus the minimization problem of equation (ix) is modified as follows:

$$\hat{\mathbf{l}} = \arg\min_{\mathbf{l}} \sum_{i=1}^{N} \|\Psi(\mathbf{p}_i, \mathbf{l}) - \mathbf{q}_i\|_1. \tag{x}$$

We choose the L1 M-estimator because the resulting problem is convex and can be solved using an off-the-shelf solver such as the L1-magic (Candes and Tao, 2005). It is fast to optimize and contains no addtional free parameter. However, it is a non-redescending M-estimator and can only handle a small percentage of outliers. When there is a higher percentage of outliers even after outlier rejection, one can use redescending M-estimators such as Tukey's bisquare M-estimator.

In the example, 174 *out of* 1617 *points* are actual outliers. We remove a set of 172 points with the outlier rejection method (Pizarro and Bartoli, 2012) and perform the robust registration. After the outlier rejection we estimate 44 *out of* 1445 *points* to be actual outliers. For better visualization we select one out of every 40 initial matches and show them along with the detected inlier matches in figure I. We also show one out of every 5 detected outliers in the bottom row of figure I.

Reconstruction

Despite outlier removal and robust registration, some inaccuracies persist naturally in the final registration. The reconstruction step needs to be robust to noisy correspondences to be able to produce a good reconstruction in such scenario as mentioned in property *a)* in section I. We show the stable type-II texture mapped reconstructions with and without the outlier rejection (Pizarro and Bartoli, 2012) and the robust registration steps in figure II. The example proves the point that handling outliers in the registration step is enough to produce good reconstructions with the proposed method even in difficult cases.

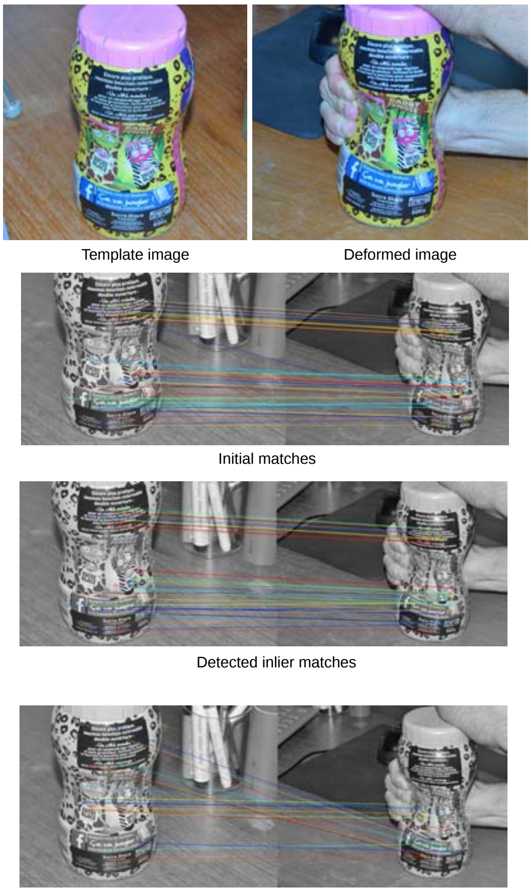


Figure I: Cropped images and point matches with outliers

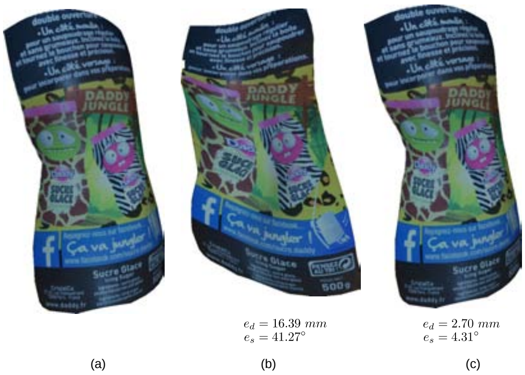


Figure II: Results: (a) ground truth (b) **typeII**-P reconstruction with standard registration (c) **typeII**-P reconstruction with robust registration.

Bibliography

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

Agudo, A. and Moreno-Noguer, F. (2015). Simultaneous Pose and Non-Rigid Shape With Particle Dynamics. In CVPR.

Akhter, I., Sheikh, Y., Khan, S. and Kanade, T. (2008). Nonrigid Structure from Motion in Trajectory Space. In NIPS.

Alcantarilla, P. F., Bartoli, A. and Davison, A. J. (2012). KAZE Features. In ECCV.

ApS, M. (2015). The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).

Bartoli, A. and Collins, T. (2013). Template-Based Isometric Deformable 3D Reconstruction with Sampling-Based Focal Length Self-Calibration. In CVPR.

Bartoli, A., Gerard, Y., Chadebecq, F. and Collins, T. (2012). On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In CVPR.

Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T. and Pizarro, D. (2015). Shape-from-Template. IEEE Trans. Pattern Anal. Mach. Intell. *37*, 2099–2118.

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

Bartoli, A., Pizarro, D. and Collins, T. (2013). A Robust Analytical Solution to Isometric Shape-from-Template with Focal Length Calibration. In ICCV.

Boyd, S. and Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.

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

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

Brunet, F. (2010). Contributions to Parametric Image Registration and 3D Surface Reconstruction. PhD thesis, Université d'Auvergne.

Brunet, F., Bartoli, A. and Hartley, R. (2014). Monocular template-based 3D surface reconstruction: Convex inextensible and nonconvex isometric methods. Computer Vision and Image Understanding *125*, 138–154.

Brunet, F., Gay-Bellile, V., Bartoli, A., Navab, N. and Malgouyres, R. (2011). Feature-Driven Direct Non-Rigid Image Registration. International Journal of Computer Vision *93*, 33–52.

Candes, E. J. and Tao, T. (2005). Decoding by Linear Programming. IEEE Transactions on Information Theory *51*, 4203–4215.

Chhatkuli, A., Pizarro, D. and Bartoli, A. (2014a). Stable Template-Based Isometric 3D Reconstruction in All Imaging Conditions by Linear Least-Squares. In CVPR.

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

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

Chhatkuli, A., Pizarro, D., Collins, T. and Bartoli, A. (2016b). A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration. IEEE Trans. Pattern Anal. Mach. Intell. *PP*.

Collins, T. and Bartoli, A. (2010). Locally affine and planar deformable surface reconstruction from video. In International Workshop on Vision, Modeling and Visualization.

Collins, T. and Bartoli, A. (2014a). Infinitesimal Plane-Based Pose Estimation. International Journal of Computer Vision *109*, 252–286.

Collins, T. and Bartoli, A. (2014b). Using Isometry to Classify Correct/Incorrect 3D-2D Correspondences. In ECCV.

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

Collins, T., Mesejo, P. and Bartoli, A. (2014). An Analysis of Errors in Graph-Based Keypoint Matching and Proposed Solutions. In ECCV.

Dai, Y., Li, H. and He, M. (2012). A simple prior-free method for non-rigid structure-from-motion factorization. In CVPR.

Del Bue, A. (2008). A factorization approach to structure from motion with shape priors. In CVPR.

Dierckx, P. (1993). Curve and Surface Fitting with Splines. Oxford University Press, Inc.

Faugeras, O. D. and Lustman, F. (1988). Motion and Structure From Motion in a Piecewise Planar Environment. International Journal of Pattern Recognition and Artificial Intelligence *2*, 485–508.

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

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

Garg, R., Roussos, A. and Agapito, L. (2013b). A variational approach to video registration with subspace constraints. International Journal of Computer Vision *104*, 286–314.

Gotardo, P. F. and Martinez, A. M. (2011). Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. IEEE Trans. on Pattern Analysis and Machine Intelligence *33*, 2051–2065.

Gotardo, P. F. U. and Martínez, A. M. (2011). Kernel non-rigid structure from motion. In ICCV.

Hartley, R. I. (1997). In Defense of the Eight-Point Algorithm. IEEE Trans. Pattern Anal. Mach. Intell. *19*, 580–593.

Hartley, R. I. and Zisserman, A. (2004). Multiple View Geometry in Computer Vision. Cambridge University Press.

Hu, Y., Zhang, D., Ye, J., Li, X. and He, X. (2013). Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization. IEEE Trans. Pattern Anal. Mach. Intell. *35*, 2117–2130.

Jensen, R. R., Dahl, A. L., Vogiatzis, G., Tola, E. and Aanæs, H. (2014). Large Scale Multi-view Stereopsis Evaluation. In CVPR.

Liu, C., Yuen, J. and Torralba, A. (2011). SIFT Flow: Dense Correspondence Across Scenes and Its Applications. IEEE Trans. Pattern Anal. Mach. Intell. *33*, 978–994.

Löfberg, J. (2004). YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In Proceedings of the CACSD Conference.

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

Maier-Hein, L., Groch, A., Bartoli, A., Bodenstedt, S., Boissonnat, G., Chang, P., Clancy, N., Elson, D. S., Haase, S., Heim, E., Hornegger, J., Jannin, P., Kenngott, H., Kilgus, T., Müller-Stich, B. P., Oladokun, D., Röhl, S., dos Santos, T. R., Schlemmer, H., Seitel, A., Speidel, S., Wagner, M. and Stoyanov, D. (2014). Comparative Validation of Single-Shot Optical Techniques for Laparoscopic 3-D Surface Reconstruction. IEEE Transactions on Medical Imaging *33*, 1913–1930.

Malis, E. and Vargas, M. (2007). Deeper understanding of the homography decomposition for vision-based control. Technical Report RR-6303 INRIA.

Malti, A., Bartoli, A. and Collins, T. (2011). A Pixel-Based Approach to Template-Based Monocular 3D Reconstruction of Deformable Surfaces. In Workshop on Dynamic Shape Capture and Analysis (4DMod-ICCV).

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

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

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

Nistér, D. (2004). An Efficient Solution to the Five-Point Relative Pose Problem. IEEE Trans. Pattern Anal. Mach. Intell. *26*, 756–777.

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

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

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

Perriollat, M., Hartley, R. and Bartoli, A. (2008). Monocular Template-based Reconstruction of Inextensible Surfaces. In BMVC.

Perriollat, M., Hartley, R. and Bartoli, A. (2011). Monocular template-based reconstruction of inextensible surfaces. International journal of computer vision *95*, 124–137.

Pilet, J., Lepetit, V. and Fua, P. (2008). Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation. International Journal of Computer Vision *76*, 109–122.

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

Pizarro, D., Bartoli, A. and Collins, T. (2013). Isowarp and Conwarp: Warps that Exactly Comply with Weak-Perspective Projection of Deforming Objects. In BMVC.

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

Puerto, G. A. and Mariottini, G. L. (2013). A Fast and Accurate Feature-Matching Algorithm for Minimally-Invasive Endoscopic Images. IEEE Trans. Med. Imaging *32*, 1201–1214.

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

Salzmann, M. and Fua, P. (2009). Reconstructing sharply folding surfaces: A convex formulation. In CVPR.

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

Salzmann, M. and Fua, P. (2011b). Deformable Surface 3D Reconstruction from Monocular Images. Synthesis lectures on computer vision, Morgan & Claypool.

Salzmann, M., Hartley, R. and Fua, P. (2007). Convex Optimization for Deformable Surface 3-D Tracking. In ICCV.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C. and Seidel, H.-P. (2004). Laplacian Surface Editing. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing.

Sumner, R. W. and Popović, J. (2004). Deformation Transfer for Triangle Meshes. ACM Transactions on Graphics *23*, 399–405.

Sundaram, N., Brox, T. and Keutzer, K. (2010). Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. In ECCV.

Tao, L. (2014). 3D Non-Rigid Reconstruction with Prior Shape Constraints. PhD thesis, University of Central Lancashire.

Tao, L. and Matuszewski, B. J. (2013). Non-rigid Structure from Motion with Diffusion Maps Prior. In CVPR.

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

Torresani, L., Hertzmann, A. and Bregler, C. (2008). Nonrigid Structure-from-Motion: Estimating Shape and Motion with Hierarchical Priors. IEEE Trans. Pattern Anal. Mach. Intell. *30*, 878–892.

Torresani, L., Yang, D. B., Alexander, E. J. and Bregler, C. (2001). Tracking and Modeling Non-Rigid Objects with Rank Constraints. In CVPR.

Varol, A., Salzmann, M., Fua, P. and Urtasun, R. (2012a). A constrained latent variable model. In CVPR.

Varol, A., Salzmann, M., Fua, P. and Urtasun, R. (2012b). A constrained latent variable model. In CVPR.

Varol, A., Salzmann, M., Tola, E. and Fua, P. (2009). Template-free monocular reconstruction of deformable surfaces. In ICCV.

Vicente, S. and Agapito, L. (2012). Soft Inextensibility Constraints for Template-Free Non-rigid Reconstruction. In ECCV.

Weinzaepfel, P., Revaud, J., Harchaoui, Z. and Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. In ICCV.

White, R., Crane, K. and Forsyth, D. (2007). Capturing and Animating Occluded Cloth. In SIGGRAPH.

Wu, C. (2013). Towards Linear-time Incremental Structure From Motion. In 3DV.

Xiang, L., Echtler, F., Kerl, C., Wiedemeyer, T., Lars, hanyazou, Gordon, R., Facioni, F., laborer2008, Wareham, R., Goldhoorn, M., alberth, gaborpapp, Fuchs, S., jmtatsch, Blake, J., Federico, Jungkurth, H., Mingze, Y., vinouz, Coleman, D., Burns, B., Rawat, R., Mokhov, S., Reynolds, P., Viau, P., Fraissinet-Tachet, M., Ludique, Billingham, J. and Alistair (2016). libfreenect2: Release 0.2.

Yu, R., Russell, C., Campbell, N. D. F. and Agapito, L. (2015). Direct, Dense, and Deformable: Template-Based Non-rigid 3D Reconstruction from RGB Video. In ICCV.

Zhang, Z. and Hanson, A. R. (1996). 3D Reconstruction Based on Homography Mapping. In ARPA Image Understanding Workshop.