Resolving Ambiguities in Monocular 3D Reconstruction of Deformable Surfaces

THÈSE N^{O} 5458 (2012)

PRÉSENTÉE LE 28 AOÛT 2012
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE VISION PAR ORDINATEUR
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Aydin VAROL

acceptée sur proposition du jury:

Prof. R. Hersch, président du jury
Prof. P. Fua, directeur de thèse
Prof. A. Bartoli, rapporteur
Prof. M. Pauly, rapporteur
Dr M. Salzmann, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

Abstract

In this thesis, we focus on the problem of recovering 3D shapes of deformable surfaces from a single camera. This problem is known to be ill-posed as for a given 2D input image there exist many 3D shapes that give visually identical projections. We present three methods which make headway towards resolving these ambiguities. We believe that our work represents a significant step towards making surface reconstruction methods of practical use.

First, we propose a surface reconstruction method that overcomes the limitations of the state-of-the-art template-based and non-rigid structure from motion methods. We neither track points over many frames, nor require a sophisticated deformation model, or depend on a reference image. In our method, we establish correspondences between pairs of frames in which the shape is different and unknown. We then estimate homographies between corresponding local planar patches in both images. These yield approximate 3D reconstructions of points within each patch up to a scale factor. Since we consider overlapping patches, we can enforce them to be consistent over the whole surface. Finally, a local deformation model is used to fit a triangulated mesh to the 3D point cloud, which makes the reconstruction robust to both noise and outliers in the image data.

Second, we propose a novel approach to recovering the 3D shape of a deformable surface from a monocular input by taking advantage of shading information in more generic contexts than conventional Shape-from-Shading (SfS) methods. This includes surfaces that may be fully or partially textured and lit by arbitrarily many light sources. To this end, given a lighting model, we learn the relationship between a shading pattern and the corresponding local surface shape. At run time, we first use this knowledge to recover the shape of surface patches and then enforce spatial consistency between the patches to produce a global 3D shape. Instead of treating texture as noise as in many SfS approaches, we exploit it as an additional source of information. We validate our approach quantitatively and qualitatively using both synthetic and real data.

Third, we introduce a constrained latent variable model that inherently accounts for geometric constraints such as inextensibility defined on the mesh model. To this end, we learn a non-linear mapping from the latent space to the output space, which corresponds to vertex positions of a mesh model, such that the generated outputs comply with equality and inequality constraints expressed in terms of the problem variables. Since its output is encouraged to satisfy such constraints inherently, using our model removes the need for computationally expensive methods that enforce these constraints at run time. In addition, our approach is completely generic and could be used in many other different contexts as well, such as image classification to impose separation of the classes, and articulated tracking to constrain the space of possible poses.

Keywords: Computer Vision, deformable surfaces, reconstruction, shading, texture, constrained latent variable model.

Résumé

Dans cette thése, nous nous concentrons sur le probléme de la reconstruction de formes en 3D de surfaces dèformables à partir d'images acquises avec une seule camèra. Ce probléme a plusieurs interpretations puisque pour une image 2D, il existe de nombreuses formes en 3D qui donnent des projections visuellement identiques. Nous prèsentons trois mèthodes qui reprèsentent un avancement vers la rèsolution de ces ambigutès. Nous croyons que notre travail reprèsente une ètape significative pour l'utilisation pratique de la reconstruction de surfaces.

Premiérement, nous proposons une mèthode de reconstruction de surface qui dèpasse les limites des mèthodes existantes basèes sur des modéle ou des mèthodes de mouvement de surfaces non-rigides. Notre mèthode na pas besoin de suivre des points sur de nombreuses frames, ou besoin d'un modéle de dèformation sophistiquè et ne dèpend pas d'une image de rèfèrence. Notre mèthode ètablit des correspondances entre des paires dimages dans lesquelles la forme est diffèrente et inconnue. Nous estimons les homographies entre des correspondances de patches extraits de deux images. Ceci donne des reconstitutions approximatives en 3D de points au sein de chaque patche avec un facteur d'èchelle. Puisque nous considèrons des patches qui se chevauchent, notre mèthode est cohèrent sur toute la surface. Enfin, un modéle de dèformation locale est utilisè pour ajuster un maillage triangulaire au nuage de points 3D, ce qui rend la reconstruction robuste à la fois au bruit et les valeurs aberrantes dans les donnèes.

Deuxiémement, nous proposons une nouvelle approche pour la reconstruction de la forme 3D d'une surface dèformable à partir d'une entrèe monoculaire en tirant parti des informations d'ombrage dans des contextes plus gènèriques. Cela comprend des surfaces qui peuvent tre totalement ou partiellement texturèes et èclairèes par de nombreuses sources de lumiére arbitraire. Etant donnè un modéle d'èclairage, nous apprenons la relation entre un modéle d'ombrage et la forme de surface locale correspondante. Au moment de l'exècution, nous avons d'abord utilisè cette connaissance pour rècupèrer la forme des patches de surface, puis appliquè une cohèrence spatiale entre les patches pour produire une forme globale 3D. En d'autres termes, au lieu de traiter la texture comme du bruit, comme dans un grand nombre dapproches nomèes shape-from-shading (SFS), nous l'utilisons comme une information supplèmentaire. Nous validons notre approche quantitativement et qualitativement à l'aide donnèes à la fois synthètiques et rèelles.

Troisiémement, nous introduisons un modéle latent de contraintes de variables qui reprèsente en soi des contraintes gèomètriques tels que dèfinis sur le modéle de maille. A cette fin, nous apprenons une cartographie non linèaire de l'espace latent à l'espace de sortie, qui contient les positions des sommets d'un modéle de maille. Les sorties gènèrèes sont conforme avec des contraintes d'ègalitè et d'inègalitè exprimè en termes de variables du probléme. Puisque la sortie est encouragèe à rèpondre à ces contraintes de faon inhèrente, utiliser notre modéle èlimine la nècessitè pour les mèthodes de calcul coûteux qui appliquent ces contraintes lors de l'exècution. En outre, notre approche est complétement gènèrique et pourrait tre utilisèe dans de nombreux autres contextes diffèrents, telles que la classification d'image pour imposer la sèparation des classes, et le suivi articulè pour contraindre l'espace de poses èventuelles.

Mots-clès: Vision par ordinateur, surfaces dèformables, reconstruction, ombrage,Texture, modéle contraint avec des variables latentes.

Acknowledgements

There are a number of people without whom this thesis could not have been written and to whom I am greatly indebted. I would like to begin by thanking Pascal Fua for making me a part of his exceptional group. I considerably benefited from his guidance and vision throughout my studies. It is with immense gratitude that I acknowledge the help of Mathieu Salzmann, an inspirational, dedicated and generous advisor. Mathieu has taught me a great deal and working with him has truly strengthened my passion for science. The quality of this work owes much to his creativity and insight.

I thank the members of my thesis committee Adrien Bartoli, Mark Pauly and Roger D. Hersch for accepting to evaluate this work and for their valuable feedback. Many thanks to Raquel Urtasun who invited me to TTI in Chicago and worked closely with me during my stay there. I give my thanks to Appu Shaji for our many discussions and for his willingness to listen and his always thoughtful advise. To Mario Christoudias, for his encouragement and for helping me during the last year of my studies. To Engin Türetken and Karim Ali for providing me both moral support and valuable feedback during thesis writing.

My special thanks go Engin Tola and Mustafa Ozuysal for their friendship and valuable advises during the first three years of my studies. Without them, I would not integrate to life in Lausanne easily. I had the privilege of sharing my office with great people during my stay in the lab: Mathieu, Albrecht, Appu and Horesh. I am grateful that we managed to create a stimulating and convivial environment conducive to good work. Thanks to Jonas and Dat for making our work on deformable surfaces obsolete. Thanks to the Turkish community in Lausanne for all the great time and support. Particularly to Zafer, Ozge, Mustafa, Engin Tola, Cumhur, Basak, Engin Turetken, Yuksel, Cagla, Mehmet and Baris for their close friendship.

There are too numerous people in the lab to list here who have enriched my life and work, made the difficulties easier to carry, and caused me to smile every now and then in the face of small jokes and gestures. I would like to sincerely thank everyone in CVLab, past and present, for their kindness and friendship. It has been great to be one of you and among you. My very special thanks go to our beloved secretary Josiane Gisclon, for her kindness and great help whenever I need it.

Last but not least, I would like to thank my family for providing me with the support needed in order to continually push myself to succeed. Without their love and support, I wouldn't be here today!

CONTENTS

List of Figures **v**

List of Tables **xv**

1 Introduction **1**

1.1 Problem Definition 1

1.2 Potential Applications 3

1.2.1 Sailing 3

1.2.2 Medical imaging 4

1.2.3 Aerospace industry 4

1.2.4 Entertainment and marketing industry 6

1.3 Ambiguities 6

1.3.1 Ambiguities of Shape-from-Texture 6

1.3.2 Ambiguities of Shape-from-Shading 10

1.4 Contributions 12

1.4.1 Reconstructing Locally Planar Deformable Surfaces 13

1.4.2 Reconstructing Locally Textured Deformable Surfaces 14

1.4.3 Surface Reconstruction using Constrained Latent Variable Model . . . 16

1.5 Outline 17

2 Literature Overview **19**

2.1 Surface Models 19

2.1.1 Physics-Based Models 20

2.1.1.1 The Finite Element Method 20

2.1.1.2 Physics-Based Methods for Computer Vision 21
2.1.2 Statistical Learning-Based Models 22
2.2 Surface Reconstruction Methods 25
2.2.1 Shape-from-Texture 25
2.2.1.1 Template-Based Methods 25
2.2.1.2 Non-rigid Structure-from-Motion Methods 28
2.2.2 Shape-from-Shading 30
2.3 Relations to our Work 31

3 Template-Free Reconstruction of Deformable Surfaces 33
3.1 Introduction 33
3.2 Two-Frame Reconstruction 35
3.2.1 Homography Decomposition 36
3.2.2 Reconstruction of a Single Patch 38
3.2.3 Reconstruction of Multiple Patches 39
3.2.4 From Point Clouds to Surfaces 40
3.2.4.1 Mesh Fitting for a Single Frame 40
3.2.4.2 Enforcing Consistency over Multiple Frames 43
3.3 Results 43
3.4 Conclusion 46

4 Reconstruction of Locally Textured Surfaces 47
4.1 Introduction 47
4.2 Method Overview 49
4.2.1 Estimating the Shape of Local Patches 50
4.2.2 Enforcing Overall Geometric Consistency 51
4.3 Estimating Local Shape 51
4.3.1 Generating Training Data 51
4.3.2 From Intensities to Deformations 53
4.3.2.1 Gaussian Processes 53
4.3.2.2 Partitioning the Training Data 55
4.3.3 Local Reconstructions from an Input Image 59
4.4 Enforcing Global Consistency 59
4.4.1 Selecting one Shape Interpretation per Patch 60

4.4.2 Aligning the Local Patches 61
4.4.3 Post Processing 61
4.5 Results 62
4.5.1 Synthetic Images 62
4.5.1.1 Robustness to Lighting Environment 67
4.5.2 Real Images 68
4.5.3 Validation 70
4.5.3.1 Preservation of Geodesic Distances 70
4.5.3.2 Comparison against Structured Light Scans 71
4.5.4 Limitations 71
4.6 Conclusion 72

5 Constrained Latent Variable Models for Surface Reconstruction 75
5.1 Introduction 75
5.2 Learning a Constrained LVM 77
5.2.1 Primal Optimization Problem 77
5.2.2 Kernel-based Mappings 79
5.3 Experimental Evaluation 83
5.3.1 Learning Setup 83
5.3.2 Synthetic Data 89
5.3.3 Real Images with Ground-truth 89
5.3.4 Real Images without Ground-truth 92
5.4 Conclusion 92

6 Conluding Remarks 97
6.1 Future Work 98

A Appendix 101

References 107

LIST OF FIGURES

1.1 **Examples of the surfaces to which we applied our reconstruction methods**. **(a)** A well-textured sheet of paper undergoing a simple deformation. **(b)** Another well-textured surface undergoing a more complex deformation. A partially textured t-shirt **(c)** and sheet of paper **(d)** 2

1.2 **Computer Vision in sailing.** We applied our reconstruction algorithms to measure the deformation of sails from images **(a)** French boat Hydroptère **(b)** Swiss boat Alinghi. 3

1.3 **Computer Vision in surgery.** **(a)** Our methods can be applied to assist surgeons by providing them real-time measurements of organ deformations during a surgery. **(b)** They can also help autonomous surgery robots in high-level scene understanding and motion planning. Images are courtesy of www.wikipedia.org. 4

1.4 **Computer Vision in image-based measurements.** Our methods can be applied to help aerospace engineers in measuring the deformations of a plane's **(a)** wings and **(b)** jet-turbine plates during a flight. Images are courtesy of www.wikipedia.org. 5

1.5 **Computer vision in entertainment and marketing industries.** **(a)** Cloth simulation results of [34]. As an alternative to cloth simulation methods in Computer Graphics applications, one could use our reconstruction methods to obtain realistic cloth deformations with only very limited manual intervention. **(b)** They can also be used to draw graphics for displaying virtual commercials on images of real surfaces such as the ones soccer players wear during a game. 5

1.6 **Obtaining 3D-2D correspondences.** Given a feature point detected in the reference image, camera parameters, and the reference model, we can retrieve the position of the feature point on the mesh by casting a ray from the camera center passing through the image coordinates of the feature point in the reference image. The point of intersection between the ray and the reference model can be identified in terms of a facet index to which the feature point belongs and barycentric coordinates with respect to facet vertices. This yields the 3D correspondence for the feature point expressed by the mesh coordinates. The image coordinates of the matching feature point on the input image provides us the corresponding 2D location to where the feature point should project after it is deformed. 7

1.7 **Effective rank of matrix $\mathbf{M}$.** **(a)** All singular values of $\mathbf{M}$ constructed for an 11×8 rectangular triangulation with 5 correspondences sampled on every facet. Note that there is a drop down after $2N_v = 176^{th}$ singular value, reducing the effective rank of $\mathbf{M}$ in Eq. 1.2. **(b)** Last $N_v = 88$ singular values plotted separately from the rest for visual purposes. Note that only the last one is actually numerically zero but not the rest. However, they are still negligible compared to the first $2N_v$ singular values of the linear system. 8

1.8 **Ambiguous solutions in 3D whose projections are almost identical in 2D.** **(a)** the solution obtained by singular value decomposition **(b)** the solution obtained when a deformation model is used such as those in [78] **(c)** the solution when distance constraints on the mesh edges are enforced **(d)** a mesh after applying a global scale to the one in **(c)**. **Top row:** The same input image to be used by different methods **(a)** to **(d)**. **Middle row:** Their projections on the input image which are indistinguishable. **Bottom row:** Triangulations in 3D as seen from a side view point. 9

1.9 **The crater illusion [67].** Pictures of craters can look like convex bumps instead of concave depressions depending on the light direction. **(a)** If we imagine the light source to be at the top, like the sun is, we observe a concave crater. **(b)** Using the same reasoning for the flipped image, we see a convex bump instead of a crater. 11

1.10 **Ambiguities for flat surfaces. First row:** Three different 3D surfaces. **Second row:** Corresponding intensity patches. Even though the 3D planar shapes have different orientations, their image appearances are identical. 11

1.11 **Ambiguities for deformed surfaces. First row:** Three different 3D surfaces. **Second row:** Corresponding intensity patches. Even though the 3D shapes are different, their image appearances are almost identical. 12

1.12 **Algorithm work flow for reconstructing locally planar deformable surfaces. (a)** Image patches are reconstructed individually up to a scale ambiguity which causes their reconstructions not to be aligned. **(b)** Using shared correspondences between the patches (blue points), we recover consistent scales for all patches and reconstruct the whole surface up to a single global scale. **(c)** Finally, a triangulated mesh is fitted to the resulting 3D point cloud to account for textureless parts of the surface and outliers in the correspondences. It can be used to provide a common surface representation across the frames and to enforce temporal consistency. 14

1.13 **Algorithmic flow for reconstructing locally textured deformable surfaces.** We partition the image into patches, some of which are labeled as textured and others as featureless. We compute the 3D shape of textured patches such as the blue one by establishing point correspondences with a reference image in which the shape is known. We use Gaussian Processes trained on synthetic data to predict plausible 3D shapes for featureless patches such as the red ones. Finally, neighborhood alignment of the patches is done using a Markov Random Field to choose among all possible local interpretations those that are globally consistent. 15

1.14 **Reconstruction results obtained using our constrained latent variable model Top row:** Images on which we applied our reconstruction method with the reprojections of the recovered meshes. **Bottom row:** Reconstructed meshes seen from a different viewpoint. 17

3.1 3D reconstruction of textured deformable surfaces from single video sequences without using a reference image. 34

3.2 **Algorithm work flow. (a)** Image patches are reconstructed individually up to a scale ambiguity which causes their reconstructions not to be aligned. **(b)** Using shared correspondences between the patches (blue points), we recover consistent scales for all patches and reconstruct the whole surface up to a single global scale. **(c)** Finally, a triangulated mesh is fitted to the resulting 3D point cloud to account for textureless parts of the surface and outliers in the correspondences. It can be used to provide a common surface representation across the frames and to enforce temporal consistency. 35

3.3 **Splitting the input image into overlapping patches.** Numbered circles represent the correspondences found between the input and support frames and colored squares are the patches. Note that some correspondences are shared by 2 or 4 patches. These shared correspondences are used later to estimate the relative scale of these patches with respect to each other in order to have a consistent shape. 36

3.4 **Equivalence between a deforming surface and moving virtual cameras (a)** A deformable surface in two different frames observed with a fixed monocular camera setup. **(b)** Equivalent representation where the surface is now fixed, but each patch is seen from two cameras: the original one, $\mathbf{P}_0$, and a virtual one, $\mathbf{P}_i$, which can be found by decomposing the homography relating the patch at time t_0 and time t_1. 37

3.5 **Changing the support frame. (a-d)** 4 of the synthetic meshes we used for our experiments. **(e)**. The three meshes recovered using (a) as the input frame and (b,c,d) in turn as the support frame. Note how similar they are. 44

3.6 **Robustness to noise. (a)** Mean distance of the vertices to the ground truth using the first frame as the support frame and the others as the input frame. The curves correspond to gaussian noise of variance 0.5, 1.0, and 2.0 added to the correspondences. **(b)** Results using the same correspondences and the method of [84], which, unlike ours, requires *a priori* knowledge of the shape in the support frame. 44

3.7 **Deforming piece of paper. Top row** Reconstructed 3D meshes reprojected into successive images. **Bottom row** The same meshes seen from a different viewpoint. 44

3.8 **Deforming Tshirt. Top row** Reconstructed 3D meshes reprojected into successive images. **Bottom row** The same meshes seen from a different viewpoint. 45

4.1 3D reconstruction of two poorly-textured deformable surfaces from single images. 48

4.2 **Algorithmic flow.** We partition the image into patches, some of which are labeled as textured and others as featureless. We compute the 3D shape of textured patches such as the blue one by establishing point correspondences with a reference image in which the shape is known. We use Gaussian Processes trained on synthetic data to predict plausible 3D shapes for featureless patches such as the red ones. Finally, neighborhood alignment of the patches is done using a Markov Random Field to choose among all possible local interpretations those that are globally consistent. 50

4.3 Panoramic image of the environment in which we performed our experiments . 52

4.4 **Mapping from intensity to surface deformation.** Projecting an intensity patch to the set of orthogonal intensity modes produces a set of intensity modal weights $\mathbf{x}$ that describe its intensity profile. Given a mapping $\mathcal{M}$ from these weights to the deformation modal weights $\mathbf{c}$, we reconstruct the shape of the patch in 3D. 54

4.5 **Ambiguities for flat (a) and deformed (b) surfaces. First rows** Three different 3D surfaces. **Second rows** Corresponding intensity patches. Even though the 3D shapes are different, their image appearances are almost identical. . . . 56

4.6 **Single vs Multiple GPs.** **(a)** Given a uniform intensity patch, there are infinitely many 3D planar patches that could have generated it. In our scheme, they are parametrized by the c_1 and c_2 weights assigned to the first two deformation modes, which encode out-of-plane rotations. The ovals represent iso-intensity values of these patches as a function of c_1 and c_2. **(b)** If we train a GP using *all* the training samples simultaneously, it will predict the same erroneous surface orientation depicted by the black dot for any uniform intensity patch. If we first partition the training samples according to angular slices shown in green and white in (a) and train a GP for each, we can predict the patch orientation shown as blue dots, which are much closer to the true orientations shown in red. **(c)** Mean and variance of the vertex-to-vertex distance between the predicted patch deformations and the ground-truth shapes for each local GP. **(d)** Accuracy of a local GP as a function of the number of training samples. GPs are accurate even when using as few as 1000 samples. In our experiments, for each local GP, we use 1400 samples on average from the training set. 57

4.7 **Intensity and Deformation Modes. Top Block.** A subset of the low-frequency intensity modes. **Bottom Block.** A subset of the low-frequency deformation modes. The top-left and middle ones encode out-of-plane rotations and the following ones are bending modes. 58

4.8 **Enforcing Shape Consistency** **(a)** Two different instances of the evaluation of the geometric consistency of patches i and j, shown in blue and green respectively. In both cases, the predicted normals of points along the same lines of sight, drawn in yellow, are compared. Since these points have the same projections, their normals should agree. Thus, the patches on the left are found to be more consistent than those on the right. **(b)** Moving patches along their respective lines of sight. The patches i and j are moved to distances d_i and d_j from the optical center so as to minimize the distance between them in their regions of overlap. 60

4.9 **Comparisons against a texture-based method [80] on a synthetic sequence First row** Input images. **Second row** Local Patches. The blue patches indicate the regions where feature correspondences are given, and the red patches are non-textured areas, selected by our patch selection algorithm. **Third row** Reconstructed point cloud (green dots) and ground-truth mesh vertices (black dots) seen from another view point. **Forth row** Estimated triangulation (green) and ground-truth triangulation (black) seen from another view point. **Fifth row** Reconstruction results using the method in [80] (green mesh) and ground-truth triangulation (black). 63

4.10 **Comparisons against SfS methods on a synthetic sequence First row:** Input images. **Second row:** Ground truth depth maps **Third row:** Depth maps computed from our reconstructions. **Forth to fifth rows:** Depth maps computed by the methods in [104], [29]. 64

4.11 Reconstruction error of both methods for all the frames in the sequence. Note that the proposed method provides much better reconstructions, except for 6 frames in the sequence. 65

4.12 **Reconstruction from noisy images. Top Row:** Input images with Gaussian noise of zero mean and 3, 8, and 26 intensity variances, respectively. **Second Row:** Comparison of our reconstructions, in blue, against the ground-truth mesh in red. **Third Row:** Intensity profiles of the same patch extracted from the noisy images.**Fourth Row:** Intensity profiles of the patch after mode projection. 66

4.13 **Reconstruction with multiple 3D hypotheses for the textured patch. (a)** Input images. **(b)** 3D hypotheses for the textured patches. **(c)** 3D reconstructions of the surfaces. 67

4.14 **Robustness to Lighting Environment** The surface is lit by three different lighting schemes. **Top row:** Intensity variation in the surface. **Bottom row:** Reconstructed surfaces. 68

4.15 **Paper Sequence. First row** Input images. **Second row** Local Patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. **Third row** Reconstructed point cloud seen from another viewpoint. **Fourth row** Geodesic distances between prominent landmarks as identified on the left. Point d in frame 7636 was outside our reconstruction, which explains the missing value in the table. 69

4.16 **T-Shirt Sequence. First row** Input images. **Second row** Local Patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. **Third row** Reconstructed point cloud seen from another viewpoint. 70

4.17 **Failure Modes. (a)** Non-Lambertian surface. **(b)** Folds that create self-shadows. **(c)** Background albedo very similar to the surface. 71

4.18 **Accuracy estimation using structured light scans. Top Row.** Two different surfaces and their corresponding structured light scans. **Middle Block.** From top to bottom, point clouds from the scans (red) and reconstructions by either our algorithm or that of [80] (green), reconstructed 3D surface rendered using a color going from blue to red as the vertex-to-surface distance to the ground-truth increases, and corresponding histogram of vertex-to-surface distances. **Bottom Block.** Depth maps obtained from our reconstructions and from the methods in [29, 104]. 73

5.1 **Generating 3D points on a hemisphere.** (Left) Predictions from random samples in the latent space using an unconstrained latent variable model. (Right) Predictions from the same samples using our constrained latent variable model. Both models were learned using only the black dots as labeled training samples. 76

5.2 **Training times.** (Left) Training time as a function of the number of validation samples when using all constraints. (Right) Training times for different constraint selection strategies for a fixed number of validation samples $V = 140$. 86

5.3 **Reconstructing a piece of cardboard from synthetic data.** (a,b) Reconstruction error and constraint violation as a function of image noise for $N = 50$ and $V = 100$. (c,d) Similar errors for $N = 50$ and $V = 150$. Our model was trained using equality constraints. 87

5.4 **Reconstructing a piece of cloth from synthetic data.** (a,b) Reconstruction error and constraint violation as a function of image noise for $N = 50$ and $V = 100$. (c,d) Similar errors for $N = 50$ and $V = 150$. Our model was trained using inequality constraints. 88

5.5 **Reconstructing a piece of cardboard using different constraint selection strategies.** Reconstruction error and constraint violation as a function of noise for $N = 100$ and $V = 100$. The curves for the single and most-violated strategies overlap. 90

5.6 **Real images with ground-truth.** (Top two rows) Images of a deforming piece of paper with reconstructed meshes seen from a different viewpoint. (Bottom two rows) Similar images for a deforming t-shirt. In both cases, we show our reconstructions reprojected on the images. 91

5.7 **Reconstructing a deforming bed sheet.** Comparison of our reconstructions with those of two baselines for two frames of the sequence. In each frame, we show the reconstructed mesh reprojected on the original image, as well as a side view of the mesh. 93

5.8 **Reconstructing a deforming cushion cover.** Reconstructions obtained with our model for 3 frames reprojected on the original image, and seen from a different viewpoint. 94

5.9 **Constraint violation.** Comparison of our model with two baselines for (a) the bed sheet and (b) the cushion sequences. 94

LIST OF TABLES

5.1 **Predicting shapes from known latent variables.** Reconstruction error as a function of the number of unlabeled examples V for a fixed $N = 50$. Note that the constraint violation measure is different for inequalities and for equalities. . 84

5.2 **Predicting shapes from known latent variables.** Constraint violation as a function of the number of unlabeled examples V for a fixed $N = 50$. Note that the constraint violation measure is different for inequalities and for equalities. . 85

5.3 **Reconstructing a piece of paper.** Reconstruction error was computed with respect to the Kinect ground-truth. 90

5.4 **Reconstructing a deforming t-shirt.** Reconstruction error was computed with respect to the Kinect ground-truth. 92

Dedicated to my mother

Yıldız Ziynet Varol

for her love and kindness.

CHAPTER

ONE

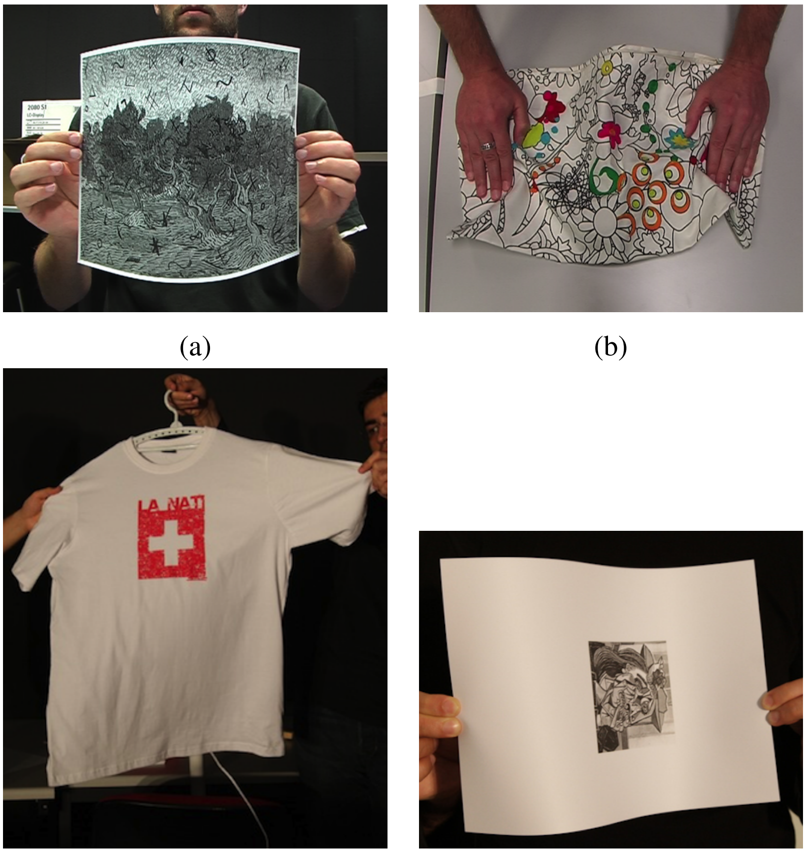
INTRODUCTION

Reconstructing 3D deformable surfaces from a single viewpoint is an active research area in Computer Vision with applications in many domains such as sports, entertainment and medical imaging. The main challenge comes from the fact that for a given 2D input image there exist many 3D shapes that give visually identical projections. In addition to the inherent ambiguities associated with solving an inverse problem, image noise makes the task even more challenging. In this thesis, we are exploring several means of resolving ambiguities in the surface reconstruction process by simultaneously exploiting various kinds of image information and using priors and constraints.

In the remainder of the chapter, we first describe the problem we address in this thesis and discuss a few practical applications. Next, we present some of the ambiguities resulting from the ill-posed nature of the surface reconstruction problem and motivate our work. Finally, we conclude the chapter by listing our main contributions to the field.

1.1 Problem Definition

Our goal in this work is to recover the 3D shape of a deformable surface given either a single image, or an image sequence of the surface acquired by a single camera. Some of the images on which we applied our reconstruction algorithms are shown in Figure 1.1. For all cases, we assume that the camera is calibrated, and that its internal parameters do not change during the capture of the sequence. We use a rectangular triangulated mesh to represent the surface of interest in 3D. We take advantage of a reference image of the surface of interest for which we



(a) (b)

(c) (d)

Figure 1.1: Examples of the surfaces to which we applied our reconstruction methods. **(a)** A well-textured sheet of paper undergoing a simple deformation. **(b)** Another well-textured surface undergoing a more complex deformation. A partially textured t-shirt **(c)** and sheet of paper **(d)** .

know the 3D shape (i.e template model) if it is available. This image is used to extract the texture information of the target surface. If such an image is not provided, we rely on two images of the same surface under different surface configurations while recovering their shape. For the cases where we exploit shading information, we assume that we are provided with the calibration of the lighting environment in which the deforming surface is captured as well as the surface albedo which can be measured from the reference image.

1.2 Potential Applications

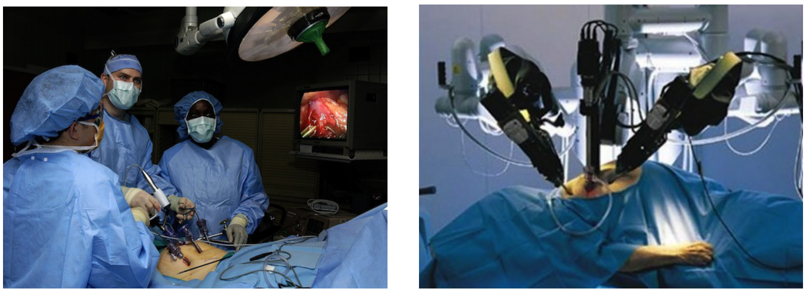
While being a very challenging task in Computer Vision, our problem has potential applications in various fields ranging from medical imaging to sports industry. We discuss a few of them below.

1.2.1 Sailing

Sailors, particularly the ones on boards of racing sail boats such as the French boat Hydroptère and the Swiss boat Alinghi as shown in Figure 1.2(a) and (b) respectively, are interested in measuring shape changes of their sails while traveling. These measurements help them make adjustments, and better understand the behavior of the boat and improve their design for higher speeds. We have successfully developed and installed real-time sail surface deformation measurement systems that use images of the sails captured by an on-boat camera. Some of these methods were inspired from the ones presented in this thesis.

(a) (b)

Figure 1.2: Computer Vision in sailing. We applied our reconstruction algorithms to measure the deformation of sails from images **(a)** French boat Hydroptère **(b)** Swiss boat Alinghi.



(a) (b)

Figure 1.3: Computer Vision in surgery. **(a)** Our methods can be applied to assist surgeons by providing them real-time measurements of organ deformations during a surgery. **(b)** They can also help autonomous surgery robots in high-level scene understanding and motion planning. Images are courtesy of www.wikipedia.org.

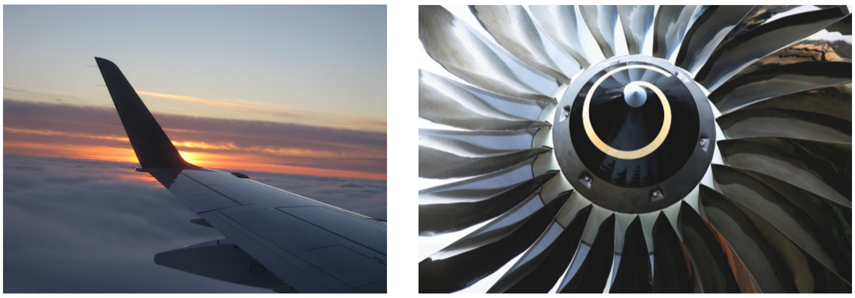
1.2.2 Medical imaging

One of the other most promising applications of our work is in medical imaging. As the techniques for surgery get more and more advanced, less invasive methods, such as laparoscopy, are becoming more popular. Laparoscopic surgery is a recent surgical method in which operations on the body are executed through a small incision as opposed to the traditional technique which requires larger incisions. A picture of a typical operating room for such a surgery is shown in Figure 1.3(a). A relatively low resolution camera captures the interior of the body and the organs as a surgeon performs the operation.

The reconstruction methods that are discussed in this work, for instance, could be useful for surgeons in two ways. First, they could be used to obtain a 3D model of the organ of interest and provide real-time feedback on its shape by augmenting the low-resolution camera view. Second, the deformations of the organs could be analyzed offline to train other surgeons. In a very similar spirit, our methods could also be utilized by autonomous surgery robots as shown in Figure 1.3(b), which is potentially one of the next important advances in surgical methods. Such measurements would be useful for an automated surgery robot in obtaining a high-level scene understanding of the operation field and planning collision-free motion paths for its arms.

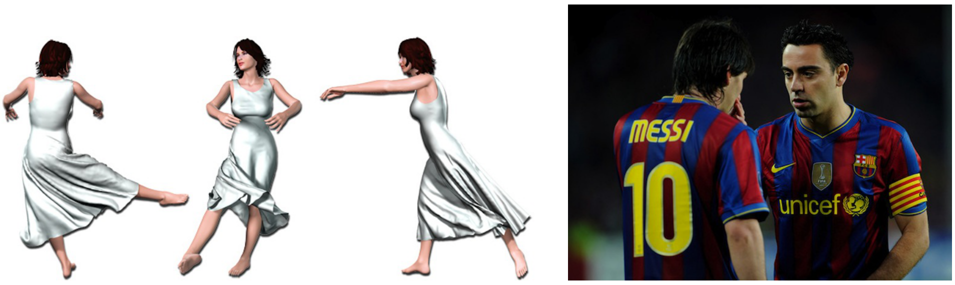
1.2.3 Aerospace industry

Another potential application of our methods lies in image-based measurements for industrial design and experimentation. Specifically, plane wings, such as the one shown in Figure 1.4(a),



(a) (b)

Figure 1.4: Computer Vision in image-based measurements. Our methods can be applied to help aerospace engineers in measuring the deformations of a plane's **(a)** wings and **(b)** jet-turbine plates during a flight. Images are courtesy of www.wikipedia.org.



(a) (b)

Figure 1.5: Computer vision in entertainment and marketing industries. (a) Cloth simulation results of [34]. As an alternative to cloth simulation methods in Computer Graphics applications, one could use our reconstruction methods to obtain realistic cloth deformations with only very limited manual intervention. **(b)** They can also be used to draw graphics for displaying virtual commercials on images of real surfaces such as the ones soccer players wear during a game.

deform significantly during a flight. Their deformations can be measured by an on-board video camera while operating under real working conditions. Such measurements can replace the results obtained by physically-based simulations which are costly in terms of labor force and time and require knowledge of physical properties of potentially complicated bodies. Our reconstructions would provide realistic feedbacks to the plane manufacturers and could help them in tailoring their designs. Similarly, to analyze their dynamical behavior while operating, our methods could be applied to jet-engine plates such as those in Figure 1.4(b).

1.2.4 Entertainment and marketing industry

There is a considerable effort in the Computer Graphics community for simulating realistic surface deformations. These efforts involve simulating cloth like surface deformations, such as the one depicted in Figure 1.5(a) [34]. As an alternative, we could use our methods to capture 3D deformations of real clothes on artists from video and use the reconstructed surfaces in animation movies, video games or other applications involving such special effects.

Similarly, as augmented reality applications are getting more popular, we expect virtual advertisements to begin appearing in a broader range of applications, such as on the shirts of the soccer players shown on Figure 1.5(b). For example, one could use our methods to estimate the deformation of the soccer shirts and draw virtual graphics that replace the real commercials appearing on them. These graphics could potentially be tailored for the different countries where the soccer game is broadcasted.

1.3 Ambiguities

Reconstructing deformable surfaces from single views is an ill-posed problem. Therefore putting the mentioned applications into practice requires the use of prior information and constraints due to the ambiguities. In what follows in this chapter, we first describe and demonstrate the nature of these ambiguities and propose solutions for resolving them. We illustrate the fact that there are different 3D shapes that correspond to the same image observations by examples. Particularly, we introduce some of the ambiguities faced while performing surface reconstructions by exploiting texture and shading cues present in the image.

1.3.1 Ambiguities of Shape-from-Texture

Shape-from-Texture (SfT) methods have been widely used to recover deformable surfaces from images in a monocular setting. Here we give an overview of the analysis studied in [78] which shows that reconstructing surface geometry from point correspondences is an ambiguous problem even when a reference image, for which the 3D shape is known, is provided.

Here we start by assuming that we are given a reference image of a textured surface whose 3D shape is represented by a known triangulated mesh, and an input image in which the surface undergoes an unknown deformation with respect to the reference. Furthermore, we assume that

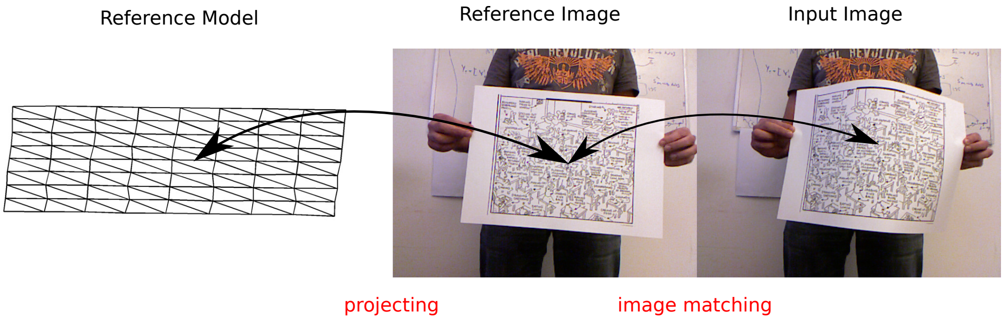


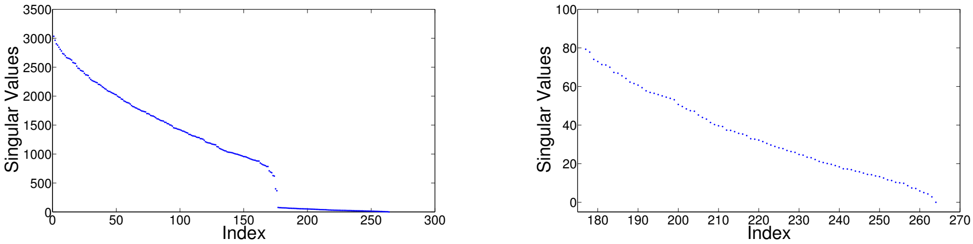
Figure 1.6: Obtaining 3D-2D correspondences. Given a feature point detected in the reference image, camera parameters, and the reference model, we can retrieve the position of the feature point on the mesh by casting a ray from the camera center passing through the image coordinates of the feature point in the reference image. The point of intersection between the ray and the reference model can be identified in terms of a facet index to which the feature point belongs and barycentric coordinates with respect to facet vertices. This yields the 3D correspondence for the feature point expressed by the mesh coordinates. The image coordinates of the matching feature point on the input image provides us the corresponding 2D location to where the feature point should project after it is deformed.

we are provided with 2D point correspondences between the two images which can be established by any feature point matching algorithm. Given those, we can obtain 3D-to-2D correspondences from the mesh to the input image by associating 2D points in the reference image to the corresponding 3D mesh points, which we represent in terms of barycentric coordinates with respect to the facets they belong to, as demonstrated by Figure 1.6. The reconstruction problem then becomes one of finding the 3D locations of the mesh vertices representing the deformed surface such that they reproject correctly on the input image.

More formally, given a 3D point defined by its barycentric coordinates $\boldsymbol{\beta}$, we can express the fact that this point should reproject at the image location $\mathbf{u}$ as

$$k \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} = \mathbf{A} \sum_{j=1}^{3} \boldsymbol{\beta}_j \mathbf{Y}_{f,j} \ , \tag{1.1}$$

where f is the index of the facet to which the 3D point belongs, $\mathbf{Y}_{f,j}$ is its j^{th} vertex, and k is a scalar accounting for depth and $\mathbf{A}$ is the known matrix of camera internal parameters. Without loss of generality, we assume that the world and camera frames are aligned. From the last row of the system of Eq. 1.1, k can be expressed as a linear function of the unknown vertex coordinates. Replacing k in the remaining rows yields two linear equations in terms of



(a) (b)

Figure 1.7: Effective rank of matrix $\mathbf{M}$. **(a)** All singular values of $\mathbf{M}$ constructed for an 11×8 rectangular triangulation with 5 correspondences sampled on every facet. Note that there is a drop down after $2N_v = 176^{th}$ singular value, reducing the effective rank of $\mathbf{M}$ in Eq. 1.2. **(b)** Last $N_v = 88$ singular values plotted separately from the rest for visual purposes. Note that only the last one is actually numerically zero but not the rest. However, they are still negligible compared to the first $2N_v$ singular values of the linear system.

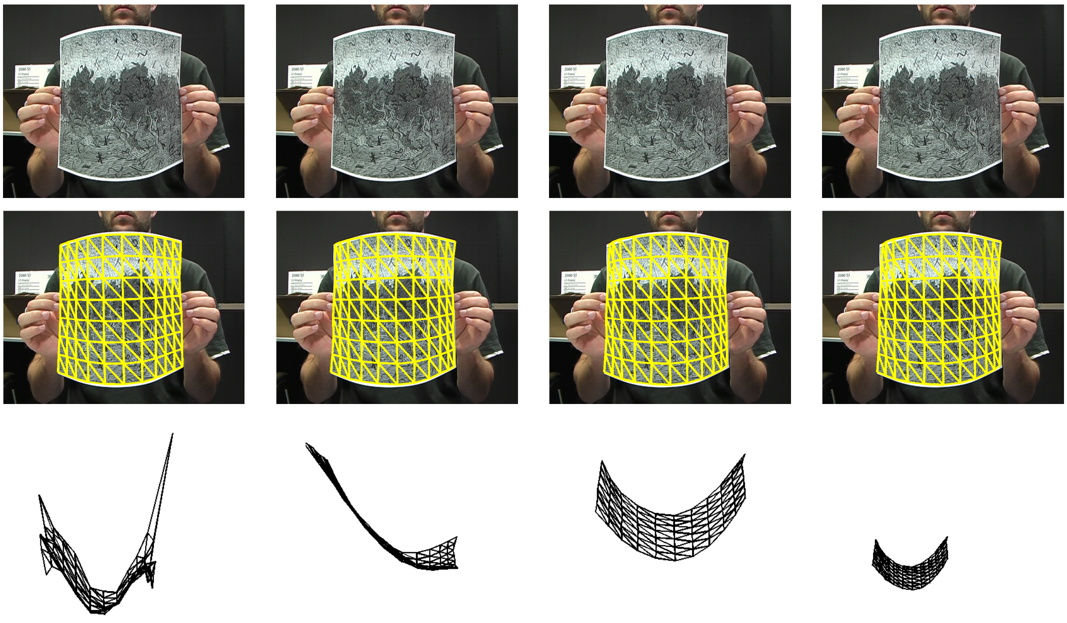
the vertex coordinates. Similar equations can be obtained for all point correspondences and grouped in a linear system of the form

$$\mathbf{M}\mathbf{Y} = \mathbf{0} \tag{1.2}$$

where $\mathbf{Y}$ is the vector of all the vertex coordinates. $\mathbf{M}$ is a $2N_c \times 3N_v$ matrix encoding similar projection equations for all correspondences where N_c is the number of correspondences and N_v is the number of vertices in the mesh.

Solving this system yields a surface which reprojects correctly on the input image. However, in general this linear system is ill-conditioned as many of its singular values are small compared to the rest. To illustrate this, we randomly sampled 5 barycentric coordinates in every facet of in an 11×8 rectangular triangulation and projected them to image coordinates by a known camera model. Note that, for well textured surfaces, $\mathbf{M}$ is a thin matrix with more rows than columns, formally $2N_c > 3N_v$.

Figure 1.7 shows singuar values of $\mathbf{M}$ in Eq. 1.2 constructed from a set of synthetic 3D-to-2D correspondences. Even though, in theory only one of them is numerically zero [78], we observe a drop down after $2N_v = 176$ singular values. The zero singular value suggests that the solution we compute by singular values decomposition of $\mathbf{M}$ will be valid up to a single global scale. The remaining small singular values that appear before the last one reveal that the matrix is not effectively full rank minus one but even lower. Therefore, there are N_v vectors spanning



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

Figure 1.8: Ambiguous solutions in 3D whose projections are almost identical in 2D. **(a)** the solution obtained by singular value decomposition **(b)** the solution obtained when a deformation model is used such as those in [78] **(c)** the solution when distance constraints on the mesh edges are enforced **(d)** a mesh after applying a global scale to the one in **(c)**. **Top row:** The same input image to be used by different methods **(a)** to **(d)**. **Middle row:** Their projections on the input image which are indistinguishable. **Bottom row:** Triangulations in 3D as seen from a side view point.

the effective null space of $\mathbf{M}$ whose any linear combination is another approximate solution to the linear system. Considering also that the 3D-to-2D correspondences are typically noisy for real images, in a practical setting, any of these approximate solutions could be the desired one and hence, there is a need for regularization through additional priors and constraints on the surface deformations.

To illustrate this on a real image, we formed the linear system in Eq. 1.2 for an input image shown at the top row of Figure 1.8. We computed its solution with and without additional constraints. As depicted in the middle and bottom rows, all the solutions project reasonably well on the image while being remarkably different in 3D. Applying singular value decomposition (SVD) on the matrix $\mathbf{M}$ and then scaling the singular vector that corresponds to the smallest singular value such that the mean edge length of the reference model is preserved results in the solution depicted in Figure 1.8(a). Using one of the deformation models, as described in [78], to regularize the surface shape results in a reconstruction shown in Figure 1.8(b). Enforcing distance constraints, that will be discussed in detail in Chapter 5, gives a visually plausible solution as shown in Figure 1.8(c), whereas any globally scaled version of this solution gives another mesh whose projection is identical to the original one as shown in Figure 1.8(d).

We observe that relying on reprojection constraints of the feature points results in a family of solutions all of which describe the input image visually indifferently. Note that for the cases when a reference image is not available, there exists less information about the surface of interest. Therefore, the problem becomes even more computationally challenging and ill-posed due to the increase in the number of unknowns to be estimated. We continue our analysis by discussing the ambiguities associated with using shading cues in surface reconstruction.

1.3.2 Ambiguities of Shape-from-Shading

Shading is one other important source of information that can be used to reconstruct surfaces from images and it is particularly useful for the surfaces which are only partially textured such as the ones depicted by Figure 1.1(c) and (d).

The Shape-from-Shading (SfS) problem is to recover the 3D shape of a surface from a single image. Developing robust methods for SfS has been one of the goals of Computer Vision since the seminal work of Horn in the 70's [41]. In all the previous attempts, intensities observed on an image are assumed to be related to angles between surface normals and the dominant light source direction. For cases where lighting environment calibration is provided, the goal is to recover the surface geometry from the intensity values. Otherwise, it is to recover both the lighting environment and the geometry simultaneously.

Similarly to the SfT problem, which was discussed in the previous section, the SfS problem is also highly ambiguous. This is the case even for the human perceptual system. For example, consider the crater images given in Figure 1.9. The shadow information in these images is not strong enough to estimate the dominant light direction. Therefore, we have little cues on where the light is coming from, as the surface might be lit from above or from below. Due to evolutionary reasons, human beings have a strong tendency to assume that the light is coming from above, as in the case of the sun or moon. Hence, we interpret the image given in Figure 1.9(a) as a concave shaped crater. However, when we look at the flipped image shown in Figure 1.9(b), we observe a convex bump instead. This example demonstrates that the light

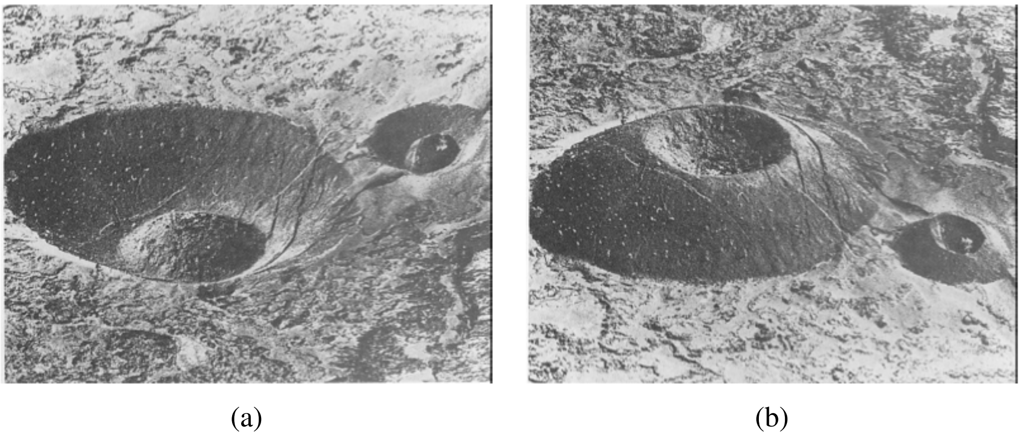

(a) (b)

Figure 1.9: The crater illusion [67]. Pictures of craters can look like convex bumps instead of concave depressions depending on the light direction. **(a)** If we imagine the light source to be at the top, like the sun is, we observe a concave crater. **(b)** Using the same reasoning for the flipped image, we see a convex bump instead of a crater.

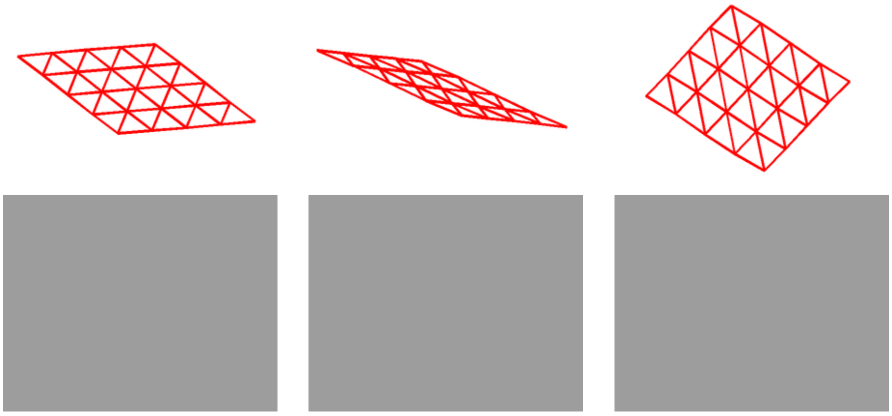


Figure 1.10: Ambiguities for flat surfaces. First row: Three different 3D surfaces. **Second row:** Corresponding intensity patches. Even though the 3D planar shapes have different orientations, their image appearances are identical.

direction and surface normals are inextricably linked. Similar sources of the ambiguities in SfS are formalized and discussed in detail in [73].

Even when the lighting environment is calibrated and available, SfS can be potentially ambiguous as different shapes in 3D can produce identical, or nearly identical, intensity profiles. This is particularly true for local reconstruction methods where image patches are analyzed

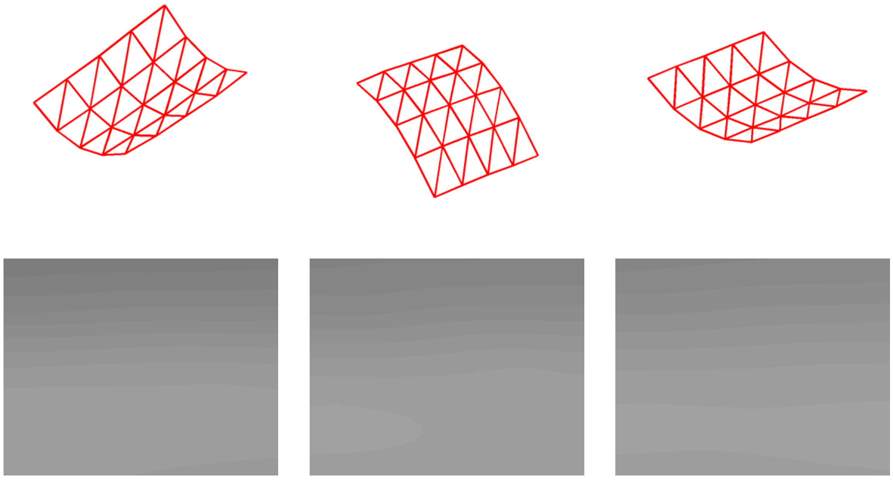


Figure 1.11: Ambiguities for deformed surfaces. First row: Three different 3D surfaces. **Second row:** Corresponding intensity patches. Even though the 3D shapes are different, their image appearances are almost identical.

individually. To illustrate this, we synthetically rendered gray-scale images of both flat and deformed 3D shapes using a known lighting environment and surface albedo as shown in Figure 1.10 and Figure 1.11, respectively. Note that not only for the flat surfaces but also for the set of deformed ones the image appearances corresponding to the different shapes in 3D are almost the same as depicted in the bottom rows of the same figures. This suggests that the inverse problem of reconstructing local surface geometry using only image appearance information is potentially ambiguous since multiple 3D surfaces could describe the same image patch equally well.

1.4 Contributions

Our goal is to make headway towards resolving the above mentioned ambiguities that exist in deformable surface reconstruction from a single view. We believe that our work represents a significant step towards making surface reconstruction methods of practical use. Throughout the thesis, we seek priors and constraints that are generic enough to be applied to different cases while still being effective for practical purposes. Using them, we overcome some of the limitations of the state of the art methods that will be discussed in the next chapter. Below we list our main contributions in this thesis:

- We begin by introducing local planarity constraints that allow us perform the surface reconstruction *without* a reference image by modeling the global surface as a set of connected planar patches. We show that our approach is capable of recovering 3D geometries of well-textured smoothly deforming surfaces in the absence of such an image, which was not possible with the earlier template-based methods.
- We then introduce a hybrid method that can reconstruct surfaces by simultaneously exploiting *both* shading and texture cues in the input image in a principled way. This approach is very similar to the previous one in that it models the whole surface as a connected set of local surface patches. However, this method is designed to handle both textureless and textured surface patches while modeling them as deformable parts instead of the rigid ones as in the earlier case. Thus our method allows us to handle more complicated deformations while being able to reconstruct poorly textured surfaces.
- Finally, we demonstrate that by using a surface parameterization that inherently satisfies the commonly employed *edge-length constraints* in the literature, we are able to improve our reconstruction accuracy. To this end we present a learning mechanism that can be used to train a constrained latent variable model for deformable surfaces.

In what follows we cover our main contributions in greater detail.

1.4.1 Reconstructing Locally Planar Deformable Surfaces

It has been previously shown that well-textured deformable 3D surfaces can be reconstructed from single video streams. However, the methods that exploit texture have some limitations. Particularly, template-based methods require a reference view in which the shape of the surface is known *a priori*. In most of the practical cases, such an image might not be readily available. Conventional Non-rigid Structure from Motion methods, on the other hand, require tracking points over long sequences, which is hard to perform.

As our first contribution, in Chapter 3, we introduce an approach to recovering the shape of a 3D deformable surface from image pairs in short video sequences that does not suffer from any of the above limitations. We neither track points over many frames, nor require a sophisticated deformation model, or depend on a reference image. Furthermore, all key algorithmic steps, which are depicted by Figure 1.12,only involve either solving linear or convex optimization problems, which can be done reliably. In short, our technique overcomes the limitations

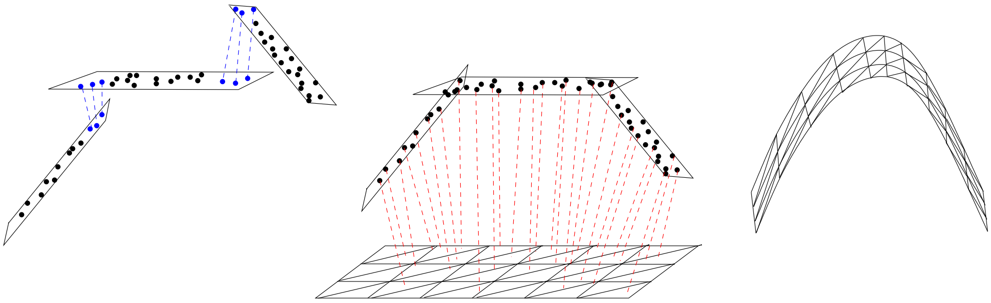

(a)
(b)
(c)

Figure 1.12: **Algorithm work flow for reconstructing locally planar deformable surfaces. (a)** Image patches are reconstructed individually up to a scale ambiguity which causes their reconstructions not to be aligned. **(b)** Using shared correspondences between the patches (blue points), we recover consistent scales for all patches and reconstruct the whole surface up to a single global scale. **(c)** Finally, a triangulated mesh is fitted to the resulting 3D point cloud to account for textureless parts of the surface and outliers in the correspondences. It can be used to provide a common surface representation across the frames and to enforce temporal consistency.

of the previous approaches by requiring only two images of a surface in unknown and different configurations. This corresponds to a more realistic scenario where a reference image is not readily available. We discuss this method in detail and present the results we obtained using it.

1.4.2 Reconstructing Locally Textured Deformable Surfaces

While texture-based methods, such as the one proposed in Chapter 3, have been proved to be effective in surface reconstruction tasks, they are ill-equipped to handle partially-textured surfaces. As our second contribution, in Chapter 4, we propose a novel approach to recovering the 3D shape of a deformable surface from a monocular input by taking advantage of shading information in more generic contexts than the conventional Shape-from-Shading (SfS) methods. This includes surfaces that may be fully or partially textured and lit by arbitrarily many light sources. To this end, given a lighting model, we learn the relationship between a shading pattern and the corresponding local surface shape. At run time, we first use this knowledge to recover the shape of surface patches and then enforce spatial consistency between the patches to produce a global 3D shape. Instead of treating texture as noise as in many SfS approaches, we exploit it as an additional source of information.

More specifically, we represent surface patches as triangulated meshes whose deformations are parametrized as weighted sums of deformation modes. We use spherical harmonics to

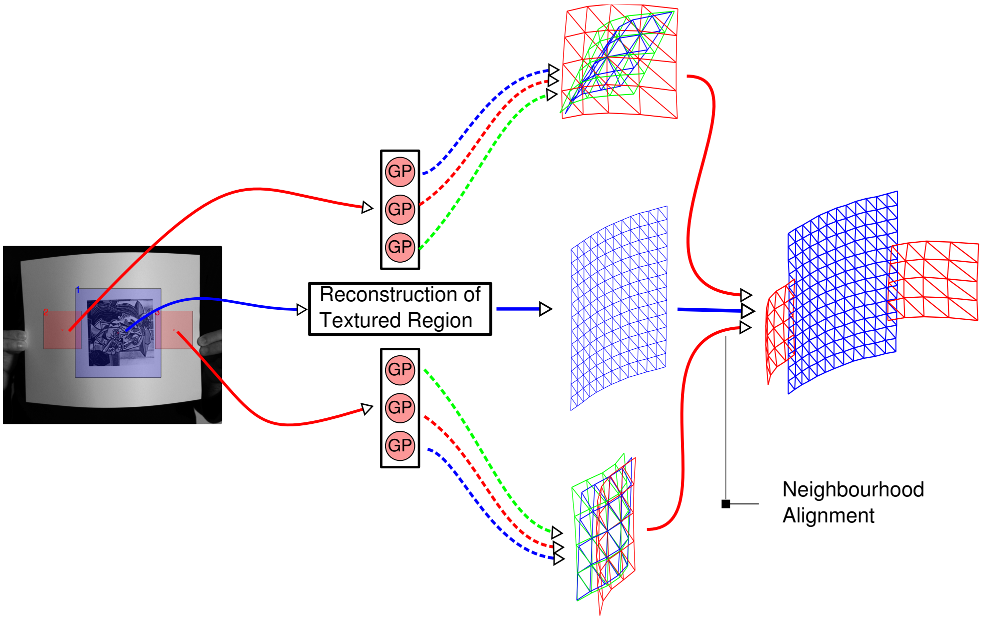


Figure 1.13: Algorithmic flow for reconstructing locally textured deformable surfaces. We partition the image into patches, some of which are labeled as textured and others as featureless. We compute the 3D shape of textured patches such as the blue one by establishing point correspondences with a reference image in which the shape is known. We use Gaussian Processes trained on synthetic data to predict plausible 3D shapes for featureless patches such as the red ones. Finally, neighborhood alignment of the patches is done using a Markov Random Field to choose among all possible local interpretations those that are globally consistent.

model the lighting environment, and calibrate this model using a light probe. This lets us shade and render realistically deforming surface patches that we use to create a database of pairs of intensity patterns and 3D local shapes. We exploit this data set to train Gaussian Process (GP) mappings from intensity patterns to deformation modes.

At run time, given an input image, we find featureless surface patches and use the GPs to predict their potential shapes, which usually yields several plausible interpretations per patch. We find the correct candidates by linking each individual patch with its neighbors in a Markov Random Field (MFR). This procedure is demonstrated in Figure 1.13. We exploit texture information to constrain the global 3D reconstruction and add robustness. To this end, we estimate the 3D shape of textured patches using a correspondence-based technique [80] and add these estimates into the Markov Random Field.

In short, our contribution is an approach to SfS that can operate in a much broader context than earlier ones: We can equally handle weak or full perspective cameras; the surfaces can be partially or fully textured; we can handle any lighting environment that can be approximated by spherical harmonics [76]; there is no need to pre-segment the surface and we return a solution in the right scale as opposed to one up to a scale factor. While some SfS earlier methods address subsets of these problems, we are not aware of any that tackles them all. We demonstrate the effectiveness of our approach on synthetic and real images, and show that it outperforms state-of-the-art texture-based shape recovery and SfS techniques.

1.4.3 Surface Reconstruction using Constrained Latent Variable Model

Recent surface reconstruction methods employ latent variable models for reducing the dimensionality of the problem independent of the image information they exploit. These models, such as those obtained by Principal Component Analysis (PCA), provide valuable compact representations for surface reconstruction from images. However, most existing models cannot directly encode prior knowledge about the specific problem at hand.

In Chapter 5, we introduce a constrained latent variable model whose generated output inherently accounts for geometric constraints such as inextensibility defined on the mesh model. To this end, we learn a non-linear mapping from the latent space to the output space, which corresponds to vertex positions of a mesh model, such that the generated outputs comply with equality and inequality constraints expressed in terms of the problem variables. Using our constrained model removes the need for computationally expensive methods that enforce these constraints at run time since its output is encouraged to satisfy such constraints inherently. In addition, our approach is completely generic and could be used in many other different contexts as well, such as image classification to impose separation of the classes, and articulated tracking to constrain the space of possible poses.

Our experimental evaluation shows that our constrained latent variable model produces more accurate reconstructions than the standard linear subspace models and the increasingly popular Gaussian Process Latent Variable Model (GPLVM) [49], which corresponds to the unconstrained version of our model. Figure 1.14 depicts some of our reconstructions for the images where the ground-truth measurements are available.

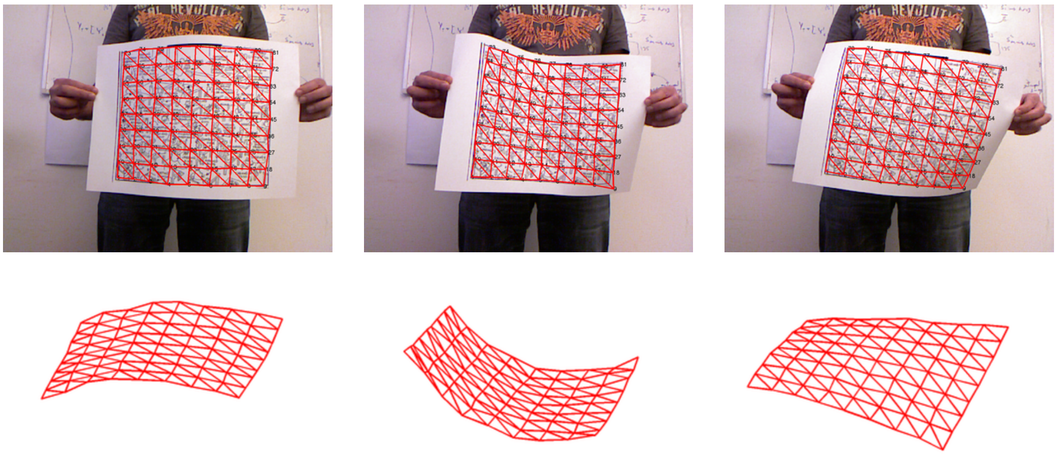


Figure 1.14: Reconstruction results obtained using our constrained latent variable model **Top row:** Images on which we applied our reconstruction method with the reprojections of the recovered meshes. **Bottom row:** Reconstructed meshes seen from a different viewpoint.

1.5 Outline

We begin in Chapter 2 by describing the previous approaches found in 3D reconstruction of surfaces literature and relate these methods to our work in this thesis. Chapter 3 introduces our approach that models a deformable surface as a connected set of planar patches. Our method does not require a template image unlike the template-based state-of-the art surface reconstruction methods. We validate our method on both synthetic and real images. Chapter 4 introduces our hybrid approach to reconstruct surfaces by using shading and texture cues simultaneously. Our framework is validated on both synthetic and real data where ground-truth is available. We show that our reconstructions are superior to those methods which use solely texture or shading information. Chapter 5 introduces our surface reconstruction approach which uses a constrained latent variable. We show that using a constrained latent variable model yields in superior accuracy in reconstruction compared to the unconstrained ones. Finally, we conclude in Chapter 6 with a retrospective and a brief discussion on future works.

Our work in this thesis partially appears in a number of peer-reviewed international conferences and journals [107, 108, 109].

CHAPTER

TWO

LITERATURE OVERVIEW

Reconstructing deformable surfaces from monocular images has been a popular research area in Computer Vision in the last decades. Although being a reasonably easy problem for human beings, it is a very challenging one for computer algorithms because of the ambiguities inherent to it. To overcome them, a number of surface models and reconstruction methods have been proposed in the literature. Here, we briefly discuss the existing approaches in the community that are related to our work in this thesis. We point out their similarities and differences with our methods to help the reader place our contributions in their right places in the existing literature.

We start by introducing some of the models that have been used to parametrize and regularize surfaces both for simulation and reconstruction purposes. Then, we present an overview of surface reconstruction methods most of which rely on these models. We classify these methods with respect to the main source of image information they exploit. Finally, we relate these methods to those presented in this thesis.

2.1 Surface Models

The existing approaches to deformable surface reconstruction rely on various techniques to represent surface deformations and constrain them in a way to make the task easier. These methods can be roughly classified into two main groups: Physics-Based and Statistical Learning-Based models. We review some of them here.

2.1.1 Physics-Based Models

Physics-based deformation models have been widely used in recovering deformable surfaces as well as simulating them with known external forces. These approaches are inspired by earlier techniques developed for Mechanical Engineering applications. The original 2D models have been used not only for shape recovery [44] but also for 2D surface registration [5]. They have also been extended to 3D modeling [97, 98]. In this formulation, surface deformations are recovered so that the minimum of the sum of internal and external energies is satisfied. Internal energy simulates the physically-based stiffness equations of a solid body and brings shape regularization in terms of its smoothness. External energy encodes the image data and attracts the shape to bend towards the image features so that its deformation fits to the image observations.

These methods, especially the Finite Element Method (FEM) [8, 123], are inspired by physical equations that govern the deformations of bodies under internal and external forces. In the remainder of this section, we describe FEM and its use in Computer Vision as well as its limitations.

2.1.1.1 The Finite Element Method

The Finite Element Method, or also known as Finite Element Analysis, is a numerical method for finding approximate solutions to engineering problems involving partial different equations. It has been applied to accurately compute deformations of structures such as beams, plates and 3D bodies under external forces [8, 123]. In this method, a discrete connected set of elements, such as cubes or tetrahedron, are used to model the structure of interest. These elements serve as mass nodes linked together by springs and dampers, resulting in a mechanical system whose behavior can be modeled by a differential equation in terms of node displacements and their first and second order derivatives. Other quantities in this system are functions of node connectivity and material properties, such as Young's modulus, Poisson's ratio or shear modulus of the material, which might be linear or nonlinear functions of displacements.

With such a system, small displacements around the initial shape can be accurately modeled as it is valid to assume that the material properties and the topology of the connected nodes remain unchanged around the initial shape. However, this is no longer true in cases where the object of interest exhibits large deformations because of both geometric and material non-linearities. Under these conditions, solving the differential equations becomes computationally expensive and even unstable. Nevertheless, a number of algorithms has been developed to overcome the nonlinearities. For example Updated Lagrangian approach, where several solutions starting from the rest configuration is iteratively computed in small steps while updating it with the current solutions, has been introduced instead of the Total Lagrangian Approach in which the matrices are computed once and remain constant. Alternatively, the co-rotational approach, where rigid rotations are treated separately from the rest of deformations, has been presented that bring numerical stability.

2.1.1.2 Physics-Based Methods for Computer Vision

In the original FEM formulation the goal is to compute displacements of a deformable body under external forces. This makes FEM suitable for certain Computer Graphics applications where the goal is to simulate the dynamics of a deforming cloth on virtual characters as they move [22, 111]. However *inverse* FEM suits better for Computer Vision applications where the goal is to recover 3D shape given noisy observations rather than simulating deformations of an object under known external forces.

While yielding physically accurate deformations of a structure, FEM has three main disadvantages that make it impractical: high computational complexity which prevents it to be used for real-time applications, requirement of exact physical properties of the object of interest which might not be available for general cases, and high dimensionality of unknowns, which makes it prune to over-fitting to potentially noisy input data. To overcome these weaknesses, modal analysis, which is another concept borrowed from Mechanical Engineering domain, has been proposed. It has been applied to techniques not only for medical imaging [66, 68] but also for other tasks such as image segmentation [61, 62].

In these methods, a linear combination of a set *vibration modes* that are obtained by solving the generalized eigenvalue problem

$$\hat{\mathbf{K}}\phi = \omega^2 \hat{\mathbf{M}}\phi \tag{2.1}$$

where $\hat{\mathbf{K}}$ and $\hat{\mathbf{M}}$ are the stiffness and mass matrices in the FEM formulation and ω's and ϕ's are the vibration frequencies and the modes, respectively. Given these, shape $\mathbf{Y}$ and displacements $\mathbf{dY}$ can be expressed as

$$\begin{aligned} \mathbf{dY} &= \sum_{i=1}^{N_s} \omega_i \phi_i & (2.2) \\ \mathbf{Y} &= \mathbf{Y}_0 + \mathbf{dY} & (2.3) \end{aligned}$$

where N_s is the number of vibration modes and $\mathbf{Y}$ and $\mathbf{Y}_0$ represent the deformed and undeformed rest shapes, respectively. The fact that lower frequency modes dominate the global shape compared to the higher ones justifies using N_s parameters, where $N_s < 3 \times number\ of\ nodes$, and thus allows for a lower dimensional surface representation.

These Physics-based models have been highly popular in Computer Vision applications since the original Snakes [44]. Even though they approximated the external forces in a similar formulation by a quadratic function that measures the sum of square of the curvatures along the surface, which makes another approximation, its quadratic form allows use of efficient solvers. Its popularity in the field made other researchers follow the same formalism in their research. A very similar approach was also used for 2D shape estimation [72] as well as 3D shape recovery from stereo [33]. Other physical-based approaches have been applied in order to reconstruct surfaces from medical images. Among all of them, balloon forces [24], superquadrics [96], and thin plates [54, 55] are proven to be useful for surface reconstruction performed on medical images. We suggest readers to review the survey papers discussing different formulations proposed for medical images for a more complete overview of the existing literature on this domain [56, 58].

These models have proved very successful when the deformation is linear, however they rely on material properties of the surfaces, which may not be always available. Furthermore, they are restricted to small deformation where linear deformation models are valid. The complexity of the nonlinear deformation models restrict their use in case of large deformations.

2.1.2 Statistical Learning-Based Models

Due to the complexity of the Physics-Based methods in practice, Statistical Learning-Based deformation models that take advantage of training data have been popular in the literature. Instead of guessing usually unknown physical properties of the objects, they extract the statistics on the shape deformations from a training set. As it is the case with the Physically-Based deformation models, the main benefit achieved by using Statistical Learning-Based models is dimensionality reduction compared to the original problem involving large number of degrees of freedom. As those degrees of freedom are not decoupled, the right parametrization should involve less degrees of freedom which lie on a lower-dimensional manifold. We categorize these methods, that are used to discover this manifold and the low-dimensional surface parametrization from training samples, into two main classes: linear and non-linear methods.

Linear dimensionality reduction methods assume that there is a linear relation between an example $\mathbf{Y}$ and its latent, typically low-dimensional, representation $\mathbf{c}$. Formally,

$$\mathbf{Y} = \mathbf{Y}_0 + \mathbf{S}\mathbf{c} + \epsilon \tag{2.4}$$

where $\mathbf{Y}_0$ is mean of the training samples, ϵ models Gaussian noise, and $\mathbf{S}$ is the matrix containing the basis vectors that span the new lower-dimensional parameters space.

The most popular method to obtain $\mathbf{S}$ is Principal Component Analysis (PCA) [43]. It is a well-known mathematical procedure that converts a set of observed samples, which are probably correlated, into a set of uncorrelated values which are computed by projecting the original data to orthogonal bases vectors. For non-rigid surfaces, in the case of the generalized eigenvalue problem of Eq. 2.1, this procedure results in a set of orthonormal deformation bases that correspond to deformation modes sorted from low to high frequencies. The matrix $\mathbf{S}$ contains N_s number of such deformation modes.

As a linear model would not be sufficiently powerful to capture potentially non-linear manifolds several non-linear dimensionality reduction methods have been introduced in the literature such as Kernel PCA [88], Locally Linear Embeddings [77], Isomap [95], Laplacian Eigenmaps [10], and Maximum Variance Unfolding [114]. However, since these methods are not designed to provide the inverse mappings, that is from low-dimensional representations to original dimensions, they are not very well-suited to surface reconstruction problem. More recently, the Gaussian Process Latent Variable Model (GPLVM) [48] has been proposed that provides such a mapping given a kernel function that measures the similarities between training data. In this model, a high-dimensional prediction for a test latent point can be computed as well as its confidence. This allows for incorporating priors for the shape. In addition, its latent representation makes the method efficient to use for surface reconstruction tasks while still requiring large number of training samples. The GPLVM formulation was extended to account for motion dynamics in the Gaussian Process Dynamical Model (GPDM) [113]. Such models have been popular in many Computer Vision tasks such as our problem of non-rigid surface recovery.

Both global and local statistical learning-based linear deformation methods have been proposed for template-based surface reconstruction methods such as the ones in [78]. Global deformation models were used to parametrize a surface with all its nodes as a weighted sum of first few columns of $\mathbf{S}$ in Eq. 2.4, that we call global deformation modes. These are obtained by applying PCA on a large set of deformation examples of a single surface model. Motion capture systems, such as Vicon®, have been used to gather these training shapes for a particular surface of interest. As an alternative, a numerical method for synthetically generating them to build a database of shapes has been proposed in [83] that replaces the time-consuming motion capture process. It relies on the fact that an inextensible mesh, where the edge-lengths of the mesh remain constant independent of the global shape, can be parametrized with a few angles between its neighboring facets.

As global deformation models are beneficial in reducing the dimensionality of the reconstruction problems, they have some drawbacks. First, gathering enough number of training data for an arbitrary surface model is not a straightforward process by the existing methods. Second, a global deformation model is only valid for a particular surface model and cannot be reused for another model having different topology. To overcome these limitations, an efficient extension of the GPLVM was used to learn a prior over local surface patch deformations [85]. To do so, a smaller set of training samples is required as local deformations are more constrained than those of a global surface. In addition, the same model can be used for surfaces of arbitrary shapes as local surface patches can be assembled together to form any global shape.

Non-rigid Structure-from-Motion methods have also been proposed in the literature for non-rigid shape recovery. They do not rely on a reference image in contrast to the template-based surface reconstruction approaches but instead track points over image sequences. Similar to template-based methods, they also make use of linear deformation models and deformation basis. Such basis can either be known a priori [1] or learned from the video sequence online [18, 100, 101, 102, 103]. The number of basis that are used to represent the deformations in the video accurately can also be estimated online [4].

In the remainder of the chapter we discuss the use of these deformation models in surface reconstruction from images.

2.2 Surface Reconstruction Methods

In this section we review the existing methods for non-rigid surface recovery from single images. We classify them according to the image information they exploit, particularly texture and shading.

2.2.1 Shape-from-Texture

Recent advances in non-rigid surface reconstruction from monocular images have mostly focused on exploiting textural information. These techniques can be roughly classified into template-based approaches and Non-rigid Structure-from-Motion (NRSFM) methods. Although the former require more information such as the camera calibration and a template model of the surface they are more robust to noise in image measurements and can be formalized with less number of degrees of freedoms making it more generally applicable compared to the latter. On the other hand, as NRSFM methods do not requite a template image or a surface model, they can be employed for the cases where template-based approaches are not applicable. In the following, we discuss the constraints and priors these methods suggest to disambiguate the surface recovery.

2.2.1.1 Template-Based Methods

Template-based methods start from a reference image in which the 3D surface shape is known. They require point correspondences established between the reference image and an input image from which the unknown 3D shape is recovered. Given such correspondences one can formulate a linear system of equations

$$\mathbf{M}\mathbf{Y} = \mathbf{0} \tag{2.5}$$

where $\mathbf{M}$ is a matrix encoding the reporjection error for the correspondences and $\mathbf{Y}$ is the vector of all the vertex coordinates. Solving this system in the least-square sense yields a surface for which the mean reprojection error of all the correspondences is small. However, in general this linear system is ill-conditioned as many of its singular values are small compared to the rest [78]. Therefore, with noisy correspondences, any linear combination of the least-square solution to it and null vectors of $\mathbf{M}$ would be the correct shape and it is not possible to identify it from point correspondences without additional priors or constraints on the surface deformations.

Imposing temporal consistency is one of the approaches suggested to overcome these ambiguities and improve the reconstruction accuracy in a tracking framework given a video sequence. The main assumption is that surfaces deform smoothly from frame to frame so its consecutive deformations are highly correlated. One way to implement such a prior is to consider multiple frames simultaneously and stack the unknown vectors of mesh coordinates $\mathbf{Y}$'s as well as the individual $\mathbf{M}$'s for all frames. Without any additional equations correlating different frames, this larger system is still under-constrained. However, by adding additional equations which penalize displacements from frame to frame for every pair of two consecutive frames, it can be made well posed as shown in [82]. Alternatively, temporal consistency can be imposed by preventing the orientation of mesh edges from changing excessively from one frame to the next. This prior can be expressed as Second Order Cone Programming (SOCP) [17] constraints besetting an upper bound that encodes the maximum orientation change allowed. Such temporal constraints are more suitable for the surface reconstruction and can handle high surface deformations without introducing superfluous smoothness as shown in [81].

While being very applicable to surfaces having different material properties, the use of these constraints is limited to tracking scenarios where both an accurate initialization and full video sequence are available. Therefore, alternative constraints such as geometric ones have been proposed for the cases where an initialization is not provided or only a single input image exists. Here we briefly overview some of the most popular geometry-based constraints existing in the literature.

- **Developable Surface Constraints:** A developable surface is a surface with zero Gaussian curvature which is defined as the product of the two principle curvatures at any point. In other words, it is a surface that can be flattened onto a plane without distortion. There have been surface reconstruction methods which are specifically developed to handle them. For example, in [35], it has been shown that in a calibrated setup solving a number of Ordinary Differential Equations yields surface reconstructions given the surface boundaries on the input image. More recently, a parametrization for developable surfaces in terms of guiding rules and bending angles has been proposed for surface reconstruction [69]. Also, it has been shown that in cases where the rulings are parallel these type of constraints turn the initially ill-posed problem to a well-posed one [92]. Unfortunately, these constraints are only valid for a specific class of surfaces and do not generalize to the others. Therefore, more generic constraints have also been introduced such as smoothness and distance constraints.

- **Smoothness Constraints:** As discussed in Section 2.1 a popular approach to impose shape smoothness is to parametrize the surface in terms of a few most-dominant global deformation modes and to regularize it such that lower-frequency components are encouraged [85]. These deformation modes could be obtained by solving the generalized eigenvalue problem involving a stiffness matrix [63, 68] as in Eq. 2.1 or applying PCA on a set of deformation examples [83]. In fact, it was reported that these modes computed by PCA produced more accurate results compared to the ones obtained by a stiffness matrix which is constructed by not exact but guessed physical parameters of the surface [78]. Although they are proven to be useful in recovering smoothly deforming surfaces, they do not perform as well when dealing with sharp deformations [80]. Using more deformation modes would be one solution in theory in the expense of introducing more unknowns which are the weights corresponding to these modes. Local deformation modes are used instead of global ones in [80, 84, 89] to be able to handle sharp folds appearing on the surface.

- **Distance Constraints:** Unfortunately, imposing only smoothness constraints is not sufficient to make the reconstruction problem a well-posed one [80] so additional constraints on the surface geometry are required. Employing distance constraints across the surface has been one of the most effective ways of disambiguating the process. For example, edge-length equality constraints or Euclidian distance constraints which enforce edge-lengths of the mesh to remain constant are employed by Salzmann et al. [80] in conjunction with global deformation models . Extended linearization [25] has been used to linearize the quadratic length-equality constraints in terms of the unknowns which are the weights for global deformation modes. A similar linearization has been performed in [59]. More recently, using a local linear deformation model with the same constraints has been suggested in [80] yielding very similar results as in [85]. Alternatively, methods that enforce such in extensibility constraints in between feature points instead of across the mesh edges have been proposed for both orthographic [28] and full projective cases [70]. One advantage of these methods over the one in [85] is that they do not, at least initially, assume surface smoothness. More recently, geodesic length constraints to model an inexentensible surface have been suggested instead of the Euclidian ones. The observation is that geodesic distances in between any pair of surface points are actually preserved but not Euclidian ones [80]. This is especially true for sharply-folding surfaces where the edge-lengths are not preserved but are upper-bounded with the distances computed for the reference configuration of the mesh. A comparison showing mean vertex-to-vertex distances between reconstructions and ground-truth meshes when using different distance constraints reveals that inequality constraints used with a local deformation model tend to outperform the global smoothness methods of [84] as well as the nonlinear local models of [85]. A recent work of Bartoli et al. [7] showed that template-based surface reconstruction from a single view with these constraints generally has a single solution.

2.2.1.2 Non-rigid Structure-from-Motion Methods

Template-based methods which are discussed in the previous chapter have proven effective in reconstructing surfaces from monocular images provided a template image where the 3D shape is known a priori. However, in many practical cases, this reference image might not be available and methods that can operate without it are required. Here, we briefly overview their common formulation and discuss the existing ambiguities as well as the previous attempts to resolve them. For a more involved discussion on NRSFM methods, we refer the reader to a recent survey paper [79].

Non-rigid Struture-from-Motion methods do not rely on a reference image and work on multiple images of the same surface with different deformations in a video sequence. The aim is to recover both 3D positions of image features throughout the sequence and camera parameters for each frame in the sequence provided frame-to-frame correspondences. The formulation that is employed by most of the recent methods was initially introduced by Bregler et al. [21]. It was an extension of the original Structure-from-Motion (SfM) method for rigid objects by Tomasi and Kanade [99]. The earlier NRSFM formulation [21], which was suggested for a weak projective camera, was later extended to handle full perspective cases by several authors [6, 51, 110, 112, 118]. Provided a number of outlier-free frame-to-frame correspondences between feature points for a video sequence, once can build a linear system that encodes the reprojection equations for 3D positions of the feature points described as a linear combination of basis vectors. As there is no mesh representation available, these basis vectors depend on the specific distribution of the feature points and cannot be pre-computed as it was the case with the template-based methods.

In the case of a full perspective camera, a linear system that groups reprojection equations of N_c frame-to-frame correspondences throughout a sequence of N_f frames can be expressed as

$$\mathbf{W} = \mathbf{C}\mathbf{B} \tag{2.6}$$

where $\mathbf{W}$ is the measurement matrix which involves 2D correspondences and perspective depth scalars for all the features points for all frames, $\mathbf{C}$ is a matrix containing camera parameters and linear deformation weights to be estimated for each frame and $\mathbf{B}$ contains the shape basis. In the weak perspective case where $\mathbf{W}$ depends on only the observed parameters and singular value decomposition yields estimates $\hat{\mathbf{C}}$ and $\hat{\mathbf{B}}$ for both $\mathbf{C}$ and $\mathbf{B}$, respectively. Unfortunately, this is no longer true for the full perspective camera where $\mathbf{W}$ cannot be constructed solely from the video sequence as it includes the perspective depth values which are unknown. Several methods have been proposed to overcome this problem. For example, in [118] an iterative method to compute the shape assuming the depths are fixed and updating them with fixed a fixed shape is proposed. Alternatively a closed-form solution [38] that involves the tensor estimation and factorization method of [37] is proposed. Unfortunately, as discussed by the authors this method is not robust to image noise.

NRSFM methods suffer from ambiguities as template-based ones do. For example, the decomposition of the matrix $\mathbf{W}$ into $\mathbf{C}$ and $\mathbf{B}$ can only be done up to an invertible transformation. This can be observed by writing

$$\mathbf{W} = \hat{\mathbf{C}}\mathbf{G}\mathbf{G}^{-1}\hat{\mathbf{B}} = \mathbf{C}\mathbf{B} \tag{2.7}$$

where $\mathbf{G}$ is any $3N_s \times 3N_s$ matrix and N_s is the number of basis vectors describing the shapes. The matrix $\mathbf{G}$ is called the corrective transformation and its existence has also been observed by the original rigid SfS method of Tomasi and Kanade [99]. The fact that any invertible matrix $\mathbf{G}$ would satisfy Eq. 2.7 makes the reconstruction ill-posed and yields in ambiguities. To overcome them a number of constraints have been incorporated into the factorization. For example, orthonormality constraints [3, 19, 51, 103, 117] can be used to ensure that the rotation matrices are orthonormal, that is the multiplication of the transposed rotation matrix with itself results in an identity matrix. Although they are effective in reducing the ambiguities of NRSFM, the results remain sensitive to the image noise.

Therefore additional constraints similar to those used in template-based methods have been introduced for NRSFM such as temporal consistency and geometric constraints. Since these methods deal with video sequences instead of still frames, the use of temporal consistency is better adapted to them. For example in [1, 26, 74] shape variations from frame-to-frame were penalized with a hand-tuned regularization term. In a similar spirit, a linear dynamical model was proposed by Torresani et al. [103]. There also have been studies [74, 75] where large camera motions from frame-to-frame are penalized instead of the shape displacements.

As the motion models do not fully disambiguate the reconstruction in some cases and they are only effective for orderly captured input frames, geometric constraints have been investigated in many studies. In the early ones, shapes are encouraged to remain close to a mean shape [18] or an initial shape [1] which is recovered using rigid SfM. To overcome the inherent ambiguity in the estimation of shape basis, several other studies have been presented. For example in [117, 118], an algorithm is proposed that chooses N_s most independent images in the sequence and enforces the shapes in those frames to be generated by a single basis shape. Other recent approaches that more directly encourage independent basis shapes are presented in [6, 20]. In contrast to these methods, which rely a linear subspace deformation models, Fayad et al. [30] suggested the use of a higher-order subspace model which exploits a quadratic deformation model allowing for reconstructions of more complicated deformations given accurate initializations.

As it is the case in template-based methods, local deformation models have proven effective in NRSFM approaches. Local surface deformations are modeled as planar [94, 107], quadratic [31]. In all of these methods, surface patches are reconstructed locally first and then global consistency between them is enforced while recovering the global deformation. More recently, another method by Taylor at al. [93] has been proposed in which triplets of neighboring triangles that move rigidly are identified and a soup of 3D triangles representing the surface is recovered whose distances between the vertices are preserved.

2.2.2 Shape-from-Shading

In the absence of texture, the natural technique to use for surface recovery is shape-from-shading. However, despite many generalizations of the original formulation of Horn [42] to account for increasingly sophisticated shading effects, such as interreflections [32, 64], specularities [65], shadows [47], or non-Lambertian materials [2], most state-of-the-art solutions can only handle a subset of these effects and, therefore, only remain valid in tightly controlled environments. Shape-from-shading techniques have been made more robust by exploiting deformation models [86, 87]. However, this was only demonstrated for the single light source case where modeling the lighting is easy.

A more practical solution to exploiting shading is to use it in conjunction with texture. In a very specialized application, shading models were used to synthetically flatten a book page [121]. In [116], texture information was first used to triangulate an input image which was then used to estimate the normals of the individual facets with respect to the reference model. Shading information was used to overcome the twofold ambiguity in normal direction that arises from template matching. In [59], the inextensibility constraints mentioned earlier and used in [84] were replaced with shading equations, which allowed the reconstruction of stretchable surfaces. In this method, the surface is assumed to be lit by a distant far source and shading equation relate the intensities of surface patches surrounding the feature points in the input and reference images. In [60], shading information was used to select the best candidate among a set ambiguous reconstructions generated for an input image.

2.3 Relations to our Work

The surface reconstruction methods we propose in this thesis are related to the above-mentioned methods in this chapter. Our methods have both similarities and differences when compared to the existing other approaches in the literature. Here we briefly discuss them.

In Chapter 3, we present a template-free surface reconstruction method. Our method does not require a template image in which 3D shape of the surface is available in contrast to the template-based methods of Section 2.2.1.1. Therefore, we consider it as one of the NRSFM methods. However, unlike most of the other NRSFM methods discussed in Section 2.2.1.2, ours does not require tracks of feature points over a relatively long video sequence which is hard to achieve. We show that we can reconstruct smoothly deforming textured surfaces from only two images of a surface in unknown and different configurations. In this approach, we model a global surface as a set of connected planar patches. Our approach of modeling the surface has later been extended for surface reconstruction tasks by several authors such as [31, 93, 94].

In Chapter 4, we introduce a hybrid method that can reconstruct surfaces by simultaneously exploiting both shading and texture cues in the input image in a principled way. This approach is very similar to the previous one in that it models the whole surface as a connected set of local surface patches. However, this method is designed to handle both textureless and textured surface patches while modeling them as deformable parts instead of rigid ones. Our approach can be considered as one of the template-based methods as we require a template image to measure the albedo of the surface of interest. However unlike the other template-based methods that exploit textural information, which are discussed in Section 2.2.1.1, we exploit shading information similar to the SfS methods of Section 2.2.2. On the other hand, our method differs from the existing SfS methods as it does not treat texture as noise but instead exploits it as an additional source of information.

In Chapter 5, we introduce a latent variable model to parametrize inextensible deformable surfaces, whose output accounts for the geometric constraints such as inextensibility defined on the mesh model. These constraints are commonly employed in surface reconstruction methods of Section 2.2. As discussed in this chapter, both linear and non-linear latent variable model have been used in non-rigid surface recovery. However, since existing models are unable to make use of the known physical properties of the surface, they produce shapes that violate important constraints, and therefore look unnatural. Physics-Based approaches, such as FEM as discussed in Section 2.1 have been introduced as an alternative approach is to directly encode the physical properties of the system. Even though these Physics-Based approaches have the advantage of explicitly encoding prior knowledge, they involve solving high-dimensional optimization problems. In our approach, we use a novel non-linear latent variable model which brings a low-dimensional parametrization and accounts for the inherent constraints of the problem. To this end, we learn a non-linear mapping from the latent space to the output space such that the generated outputs comply with equality and inequality constraints expressed in terms of the problem variables.

CHAPTER

THREE

TEMPLATE-FREE RECONSTRUCTION OF DEFORMABLE SURFACES

3.1 Introduction

It has been previously shown that well-textured deformable 3D surfaces, such as those in Figure 3.1, could be recovered from single video streams. However, the methods that exploit texture have some limitations. Particularly, template-based methods require a reference view in which the shape of the surface is known *a priori*, which often may not be available. On the other hand, non-rigid Structure from Motion methods require tracking points over long sequences, which is hard to do. In this chapter, we introduce an approach to recovering the shape of a 3D deformable surface from image pairs in short video sequences that does not suffer from any of the above limitations: We do not track points over many frames, require a sophisticated deformation model, or depend on a reference image. Furthermore, all key algorithmic steps depicted by Figure 3.2 only involve either solving linear or convex optimization problems, which can be done reliably. In short, our technique overcomes the limitations of the previous approaches by requiring only two images of a surface in unknown and different configurations. This corresponds to a more realistic scenario in many situations.

More specifically, given two images for which the shapes are both unknown and different, we first establish image-to-image correspondences. We then split each image into small overlapping patches, which we assume to be flat. This lets us estimate a homography between any two corresponding patches, from which we can recover the 3D positions of the feature points in

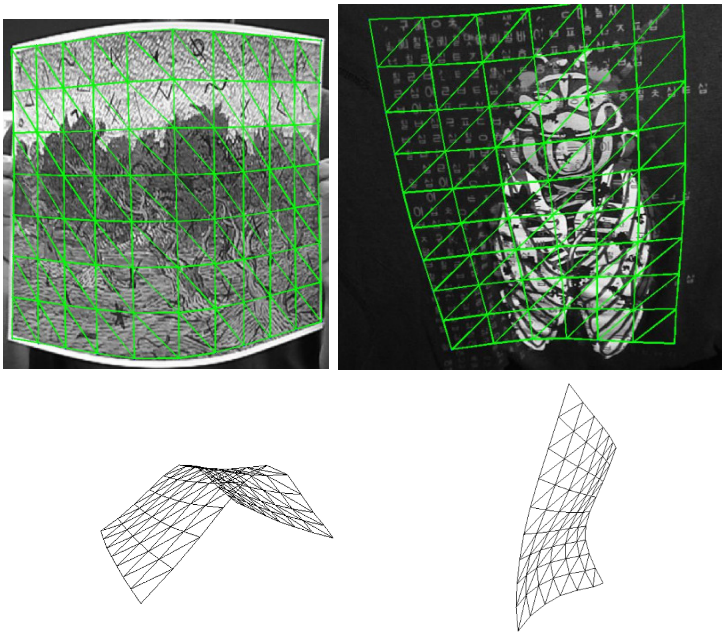


Figure 3.1: 3D reconstruction of textured deformable surfaces from single video sequences without using a reference image.

the patches up to a scale factor. Since the image patches overlap, we can enforce scale consistency among all the reconstructed 3D points, which yields a cloud of 3D points that describes the deformed surface up to a single global scale factor. Finally, to further ensure robustness to noise and outliers, and to have a common surface representation for the different frames of the sequence, we fit an inextensible triangulated mesh regularized by a local deformation model to the resulting point cloud, which can be expressed as a convex optimization problem.

In the remainder of this chapter we first describe our method of reconstructing a planar surface patch up to a scale factor provided that we can establish enough correspondences between a pair of images depicting two different configurations of the same surface. We then present how we perform reconstruction of multiple surface patches and solve for the relative scale between them in order to get a consistent point cloud. Before presenting our qualitative and quantitate results, we introduce our mesh fitting algorithm applied to the reconstructed point cloud.

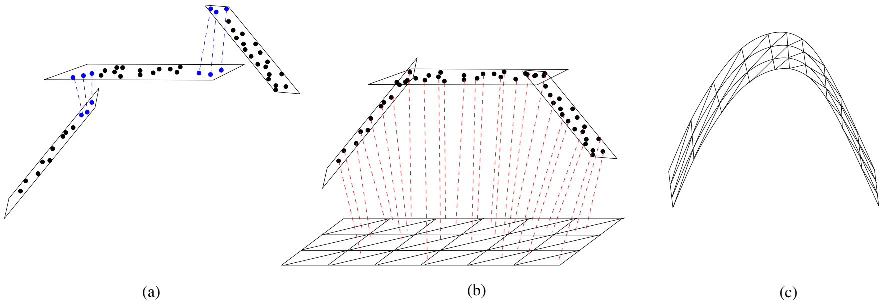


Figure 3.2: Algorithm work flow. **(a)** Image patches are reconstructed individually up to a scale ambiguity which causes their reconstructions not to be aligned. **(b)** Using shared correspondences between the patches (blue points), we recover consistent scales for all patches and reconstruct the whole surface up to a single global scale. **(c)** Finally, a triangulated mesh is fitted to the resulting 3D point cloud to account for textureless parts of the surface and outliers in the correspondences. It can be used to provide a common surface representation across the frames and to enforce temporal consistency.

3.2 Two-Frame Reconstruction

In this section, we show how we can reconstruct the shape of a 3D deforming surface from 2 frames, provided that we can establish enough correspondences and that the surface changes from one frame to the other. Note that this is very different both from conventional stereo, which relies on the shape being the same in both frames, and from recent monocular approaches to 3D shape recovery, which require knowledge of the shape in a reference image [70, 81, 84, 122].

In the following, we refer to the first image of the pair as the *input image* in which we want to recover the 3D shape and to the second as the *support image*. We assume the camera to be calibrated and the matrix $\mathbf{A}$ of intrinsic parameters given. To simplify our notations and without loss of generality, we express all 3D coordinates in the camera referential. Finally, we assume that the surface is inextensible and model it as a set of overlapping planar patches that only undergo rigid transformations between the two images. In practice, on images such as those presented in the result section, we use patches of size 100×100 pixels that overlap by 50 pixels.

Given point correspondences between the input and support images established using SIFT[52], all subsequent algorithmic steps depicted by Figure 3.2 only involve solving linear or convex optimization problems. We first split the input image into small overlapping patches and compute homographies between pairs of corresponding patches. For each patch, the corresponding

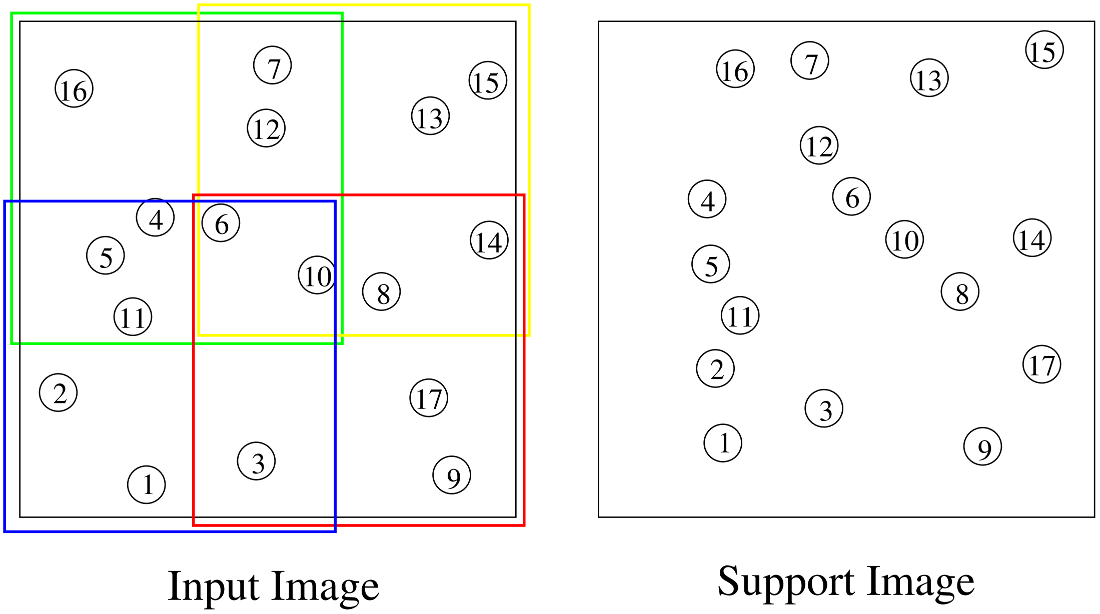


Figure 3.3: Splitting the input image into overlapping patches. Numbered circles represent the correspondences found between the input and support frames and colored squares are the patches. Note that some correspondences are shared by 2 or 4 patches. These shared correspondences are used later to estimate the relative scale of these patches with respect to each other in order to have a consistent shape.

homography can be decomposed into relative rotation and translation, which let us compute the 3D coordinates of all its feature points up to a scale factor. We can then recover a cloud of 3D points for the whole surface up to a global scale factor, by enforcing consistency between neighboring patches. Finally, to fill the gaps in the reconstructed points and to discard outliers, we fit a triangulated surface model to this cloud. In the remainder of this section, we describe these steps in more details.

3.2.1 Homography Decomposition

Since we model the surface as a set of rigidly moving patches, we can define these patches over the input image by splitting it into small overlapping regions as depicted by Figure 3.3. For each such patch, we estimate the homography that links its feature points to the corresponding ones in the support image. To this end, we perform a RANSAC-based robust homography estimation [40] and label the correspondences which disagree with the estimated homography as outliers. This yields a reduced number of points on the images, which we now consider as

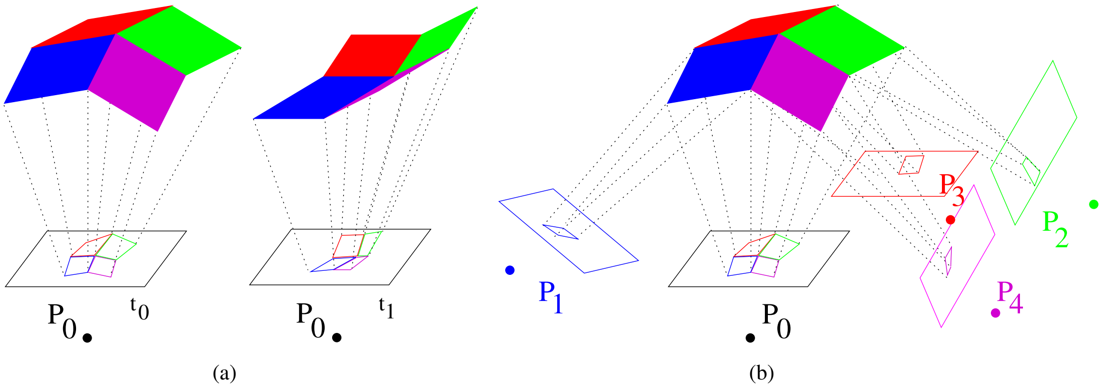


Figure 3.4: Equivalence between a deforming surface and moving virtual cameras (a) A deformable surface in two different frames observed with a fixed monocular camera setup. **(b)** Equivalent representation where the surface is now fixed, but each patch is seen from two cameras: the original one, $\mathbf{P}_0$, and a virtual one, $\mathbf{P}_i$, which can be found by decomposing the homography relating the patch at time t_0 and time t_1.

our correspondences, and which are grouped into local patches with an estimated homography for each.

Given the homography estimated for a patch, we now seek to retrieve its 3D surface normal $\mathbf{n}_i$ as well as its rigid motion between the two frames expressed as a rotation and translation. As depicted by Figure 3.4, this is equivalent to assuming that the patch is fixed and that the camera is moving, which yields one virtual camera per patch. Since we know its internal parameters, its translation $\mathbf{t}_i$, its rotation $\mathbf{R}_i$ and $\mathbf{n}_i$ can be recovered up to a scale factor by decomposing the homography [53, 120]. Let $\mathbf{P}_i = \mathbf{A}[\mathbf{R}_i|\mathbf{t}_i]$ be the projection matrix of the virtual camera for patch i. The decomposition of the corresponding homography $\mathbf{H}_i$ is expressed as

$$\mathbf{H}_i = \mathbf{R}_i - \frac{\mathbf{t}_i \mathbf{n}_i^T}{d^i} = \mathbf{R}_i - \mathbf{t'}_i \mathbf{n}_i^T \quad , \tag{3.1}$$

where d^i is the unknown distance of the patch to the camera and $\mathbf{t'}_i$ is the scaled translation. This decomposition results in two distinct solutions for the relative camera motion and the patch normals. We pick the solution with the normal whose sum of the angle differences with the neighboring patches is smallest.

3.2.2 Reconstruction of a Single Patch

Given a virtual camera $\mathbf{P}_i$, whose external parameters were estimated from the homography, and the original camera $\mathbf{P}_0 = \mathbf{A}[\mathbf{I}|\mathbf{0}]$, we seek to reconstruct the C^i 3D points $\mathbf{Q}^i_j$, $1 \leq j \leq C^i$ of patch i. To this end, we minimize the reprojection errors both in the input and support frames, which, for a single point j can be formulated as the least-squares solution to the linear system

$$\mathbf{B}^i_j \mathbf{Q}^i_j = \mathbf{b}^i_j \;, \tag{3.2}$$

where

$$\mathbf{b}^i_j = \begin{bmatrix} -p^0_{14} + r^i_{j,x} p^0_{34} \\ -p^0_{24} + r^i_{j,y} p^0_{34} \\ -p^i_{14} + s^i_{j,x} p^i_{34} \\ -p^i_{24} + s^i_{j,y} p^i_{34} \end{bmatrix}_{4\times 1} \quad , \text{ and} \tag{3.3}$$

$$\mathbf{B}^i_j = \begin{bmatrix} p^0_{11} - r^i_{j,x} p^0_{31} & p^0_{12} - r^i_{j,x} p^0_{32} & p^0_{13} - r^i_{j,x} p^0_{33} \\ p^0_{21} - r^i_{j,y} p^0_{31} & p^0_{22} - r^i_{j,y} p^0_{32} & p^0_{23} - r^i_{j,y} p^0_{33} \\ p^i_{11} - s^i_{j,x} p^i_{31} & p^i_{12} - s^i_{j,x} p^i_{32} & p^i_{13} - s^i_{j,x} p^i_{33} \\ p^i_{21} - s^i_{j,y} p^i_{31} & p^i_{22} - s^i_{j,y} p^i_{32} & p^i_{23} - s^i_{j,y} p^i_{33} \end{bmatrix}_{4\times 3} \quad , \tag{3.4}$$

and where p^k_{mn} the $(m,n)^{th}$ entry of the k^{th} projection matrix $\mathbf{P}_k$, and $\mathbf{r^i_j}$ and $\mathbf{s^i_j}$ are the 2D coordinates on the input frame and on the support frame, respectively.

Furthermore, to ensure that the patch remains flat, we constrain its points to lie on a plane whose normal is the one given by the homography decomposition of Eq. (3.1). Since the reconstruction of the points in camera coordinates can only be up to a scale factor, we can fix without loss of generality the depths of the plane to a constant value, $d^i = d_0$. For a single point j, the planarity constraint can then also be formulated as a linear equation in terms of $\mathbf{Q}^i_j$ as

$$\mathbf{n}_i^T \mathbf{Q}^i_j = -d_0 \;. \tag{3.5}$$

We combine Eqs. (3.2) and (3.5) into the linear system

$$\mathbf{G}^i_j \mathbf{Q}^i_j = \mathbf{g}^i_j \;, \tag{3.6}$$

where

$$\mathbf{G}^i_j = \begin{bmatrix} \mathbf{B}^i_j \\ \mathbf{n}_i^T \end{bmatrix}_{5\times 3} \text{ and } \mathbf{g}^i_j = \begin{bmatrix} \mathbf{b}^i_j \\ -d_0 \end{bmatrix}_{5\times 1} .$$

We can then group individual systems for each point in patch i into the system

$$\begin{bmatrix} \mathbf{G}_1^i & & & & \\ & \ddots & & & \\ & & \mathbf{G}_j^i & & \\ & & & \ddots & \\ & & & & \mathbf{G}_{C^i}^i \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^i \\ \vdots \\ \mathbf{Q}_j^i \\ \vdots \\ \mathbf{Q}_{C^i}^i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1^i \\ \vdots \\ \mathbf{g}_j^i \\ \vdots \\ \mathbf{g}_{C^i}^i \end{bmatrix} , \qquad (3.7)$$

whose solution is valid up to a scale factor in camera coordinates.

3.2.3 Reconstruction of Multiple Patches

The method described above lets us reconstruct 3D patches individually each with its own depth in camera coordinates. However, because the depths of different patches are inconsistent, this results in an unconnected set of 3D points. We therefore need to re-scale each patch with respect to the others to form a consistent point cloud for the whole surface. To this end, we use overlapping patches in the input image where each patch shares some of the correspondences with its neighbors. Let $\hat{\mathbf{Q}}$ be a single point shared by patches i and i' such that $\hat{\mathbf{Q}} = \mathbf{Q}_j^i = \mathbf{Q}_{j'}^{i'}$. The scales d^i and $d^{i'}$ for the two patches can then be computed by solving the linear system

$$\begin{bmatrix} \mathbf{B}_j^i & \mathbf{0}_{4\times 2} \\ \mathbf{n}_i^T & 1 \;\; 0 \\ \mathbf{B}_{j'}^{i'} & \mathbf{0}_{4\times 2} \\ \mathbf{n}_{i'}^T & 0 \;\; 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{Q}} \\ d^i \\ d^{i'} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_j^i \\ 0 \\ \mathbf{b}_{j'}^{i'} \\ 0 \end{bmatrix} . \qquad (3.8)$$

As before, the equations for all the shared points of all the patches can be grouped together, which yields the system

$$\mathbf{E} \begin{bmatrix} \tilde{\mathbf{Q}} \\ d^1 \\ \vdots \\ d^{N_p} \end{bmatrix} = \mathbf{e} , \qquad (3.9)$$

where $\tilde{\mathbf{Q}}$ is the vector of all shared 3D points, N_p is the number of planar patches and $\mathbf{E}$ and $\mathbf{e}$ are formed by adequately concatenating the matrices and the vectors of Eq. (3.8). Solving Eq. (3.9) gives the relative scales $\left[d^1 ... d^{N_p}\right]$ for all the patches, which lets us compute a consistent 3D point cloud for the whole surface. Note, however, that, since these scales are relative, the resulting point cloud is recovered up to a single global scale factor.

3.2.4 From Point Clouds to Surfaces

In the previous sections, we have presented an approach to reconstructing 3D points from two images depicting two different configurations of the surface. Because the recovered point clouds may still contain some outliers and because in many applications having a common surface representation for all the frames of a sequence is of interest, we fit a triangulated mesh to the reconstructed point clouds within a convex optimization framework.

3.2.4.1 Mesh Fitting for a Single Frame

Given the vector $\mathbf{Q}$ obtained by concatenating the N reconstructed 3D points, we seek to recover the deformation of a given mesh with N_v vertices and N_e edges that best fits $\mathbf{Q}$. Since $\mathbf{Q}$ has been reconstructed up to a global scale factor, we first need to resize it, so that it matches the mesh area. In camera coordinates, a rough approximation of the scale of a surface can be inferred from the mean depth of its points. Computing such values for both the mesh and the point cloud allows us to resize the latter to a scale similar to that of the mesh. Then, because the surface may have undergone a rigid transformation, we align the mesh to the point cloud by applying a standard Iterative Closest Point (ICP) algorithm [119]. In the current implementation, a coarse manual initialization is provided for ICP. This is the only non fully automated step in the whole algorithm. It is required to indicate an area of interest in the absence of a reference image.

From this first alignment, we can deform the mesh to fit the point cloud. To do so, we first estimate the location of each 3D point $\mathbf{Q}_j$ on the mesh. These locations are given in barycentric coordinates $\boldsymbol{\beta}$ with respect to the mesh facets, and can be obtained by intersecting rays between the camera center and the 3D points with the mesh. Given this representation, each 3D point can be written as $\mathbf{Q}_j = \sum_{k=1}^{3} \beta_k \mathbf{y}_{f(j)}^k$, where $f(j)$ represents the facet to which point j was attached, and $\mathbf{y}_{f(j)}^k$ is its k^{th} vertex. Fitting a mesh to the whole point cloud can then be written as the solution of the linear system

$$\mathbf{MY} = \mathbf{Q} \;, \tag{3.10}$$

where $\mathbf{M}$ is a $3N \times 3N_v$ matrix containing the barycentric coordinates of all 3D points, and $\mathbf{Y}$ is the vector of concatenated mesh vertices.

Because the scale factor obtained from the depth of the points is only a rough estimate of the true scale, we need to refine it. This can be done by introducing a variable γ accounting for the scale of the point cloud in the above-mentioned reconstruction problem, and solve

$$\mathbf{MY} = \gamma\mathbf{Q}\ . \tag{3.11}$$

However, without further constraints on the mesh, nothing prevents it from shrinking to a single point and therefore perfectly satisfy the equation. Assuming that the surface is inextensible, we can overcome this issue by maximizing γ under inequality constraints that express the fact that the edges of the mesh cannot stretch beyond their original length. The problem can then be re-formulated as the optimization problem

$$\begin{aligned} \underset{\mathbf{Y},\gamma}{\text{maximize}}\ & w_s\gamma - \|\mathbf{MY} - \gamma\mathbf{Q}\| \\ \text{subject to}\ & \|\mathbf{y}_k - \mathbf{y}_j\| \leq l_{j,k}\ ,\ \forall(j,k) \in \mathcal{E} \\ & \gamma_{low} \leq \gamma \leq \gamma_{up}\ , \end{aligned} \tag{3.12}$$

where $\mathcal{E}$ is the set of mesh edges, $l_{j,k}$ is the original length of the edge between vertices $\mathbf{v}_j$ and $\mathbf{v}_k$, and w_s is a weight that sets the relative influence between point distance minimization and scale maximization. To further constrain the scale of the point cloud, we introduced a lower and an upper bounds γ_{low} and γ_{up}. The advantage of using inequality constraints over edge length equalities is twofold. First, the inequality constraints are convex, and can therefore be optimized easily. Second, these constraints also are more general than the equality ones, since they allow to account for folds appearing between the vertices of the mesh, which is bound to happen in real scenarios.

Finally, to account for outliers in the 3D reconstructed points, we introduce a linear local deformation model. As in [85], we model a global surface as a combination of local patches. Note that these patches are different from those used in the point cloud reconstruction, since we expect them to deform. To avoid the complexity of the non-linear model of [85], and to keep our formulation convex, we use a linear local model, where the shape of a patch $\mathbf{Y}_i$ is computed as a linear combination of N_s deformation modes $\lambda_j\ ,\ 1 \leq j \leq N_s$, which we can write

$$\mathbf{Y}_i = \mathbf{Y}_i^0 + \Lambda\mathbf{c}_i\ , \tag{3.13}$$

where Λ is the matrix whose columns are the deformation modes, $\mathbf{Y}_i^0$ is the mean shape of patch i, and $\mathbf{c}_i$ is the vector of its mode coefficients. Thanks to the local deformation models, this method is applicable to meshes of any shape, be it rectangular, circular, triangular, or any other.

In practice, these modes are obtained by applying Principal Component Analysis (PCA) to a set of inextensible patches deformed by randomly setting the angles between their facets. Since the deformation modes obtained with PCA are orthonormal, the coefficients $\mathbf{c}_i$ that define a patch shape can be directly computed from $\mathbf{Y}_i$ as $\mathbf{c}_i = \Lambda^T \left(\mathbf{Y}_i - \mathbf{Y}_i^0\right)$. This, in contrast with the standard use of linear deformation models, lets us express our deformation model directly in terms of the mesh vertex coordinates. Furthermore, we use all the modes, which lets us represent any complex shape of a patch,and we regularize the projection of the shape in the modes space by minimizing

$$\left\|\Sigma^{-1/2}\mathbf{c}_i\right\| = \left\|\Sigma^{-1/2}\Lambda^T \left(\mathbf{Y}_i - \mathbf{Y}_i^0\right)\right\| \ , \tag{3.14}$$

which penalizes overly large mode weights, and where Σ is a diagonal matrix containing the eigenvalues of the training data covariance matrix. This lets us define the global regularization term

$$E_r(\mathbf{Y}) = \sum_{i=1}^{N_d} \left\|\Sigma^{-1/2}\Lambda^T \left(\mathbf{Y}_i - \mathbf{Y}_i^0\right)\right\| \ , \tag{3.15}$$

by summing the measure of Eq. 3.14 over all N_d overlapping patches in the mesh. This regularization can be inserted into our convex optimization problem, which then becomes

$$\begin{aligned} \underset{\mathbf{V},\gamma}{\text{maximize}} \quad & w_s\gamma - \|\mathbf{MY} - \gamma\mathbf{Q}\| - w_r E_r(\mathbf{Y}) \\ \text{subject to} \quad & \|\mathbf{y}_k - \mathbf{y}_j\| \leq l_{j,k} \ , \ \forall (j,k) \in \mathcal{E} \\ & \gamma_{low} \leq \gamma \leq \gamma_{up} \ , \end{aligned} \tag{3.16}$$

where w_r is a regularization weight. In practice, because the shape of the mesh is initially far from matching that of the point cloud, we iteratively compute the barycentric coordinates of the points on the surface and solve the optimization problem of Eq. 3.16 using the available solver SeDuMi [91].

3.2.4.2 Enforcing Consistency over Multiple Frames

While, in most cases, the mesh reconstruction presented in the previous section is sufficient to obtain accurate shapes, we can further take advantage of having a video sequence to enforce consistency across the frames. In the previous formulation nothing constrains the barycentric coordinates of a point to be the same in every frame where it appears. We now show that such constraints can be introduced in our framework. This lets us reconstruct multiple frames simultaneously, which stabilizes the individual results in a way that is similar to what bundle adjustment methods do.

The only additional requirement is to be able to identify the reconstructed points in order to match them across different frames. This requirement is trivially fulfilled when all points have been reconstructed using the same support frame. With multiple support frames, such an identification can easily be obtained by additionally matching points across the different support frames. Given the identity of all points, we only need to compute barycentric coordinates once for each point, instead of in all frames as before. For points shared between several frames, this is done in the frame that gave the minimum point-to-surface distance.

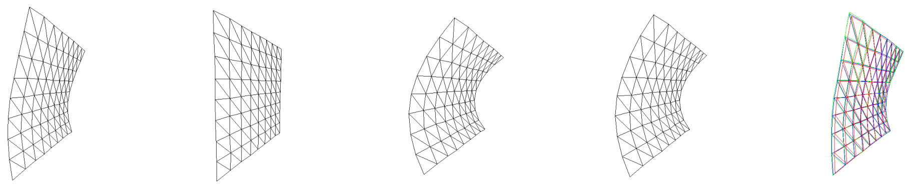
This lets us rewrite the optimization problem of Eq. 3.16 in terms of the vertex coordinates in the N_f frames of a sequence as

$$\begin{aligned} \underset{\mathbf{Y}^{1,\ldots,N_f},\gamma^{1,\ldots,N_f}}{\text{maximize}} \quad & \sum_{t=1}^{N_f} \left(w_s\gamma^t - \|\mathbf{M}^t\mathbf{Y}^t - \gamma^t\mathbf{Q}^t\| - w_r E_r(\mathbf{V}^t) \right) \\ \text{subject to} \quad & \|\mathbf{y}_k^t - \mathbf{y}_j^t\| \le l_{j,k} \ , \ \forall (j,k) \in \mathcal{E} \ , \ \forall t \in [1, N_f] \\ & \gamma_{low} \le \gamma^t \le \gamma_{up} \ , \forall t \in [1, N_f] \ , \end{aligned} \tag{3.17}$$

where $\mathbf{Y}^t$, γ^t, $\mathbf{Q}^t$ and $\mathbf{M}^t$ are similar quantities as in Eq. 3.16 but for frame t. As in the single frame case, we iteratively solve this problem and recompute the barycentric coordinates of the unique points.

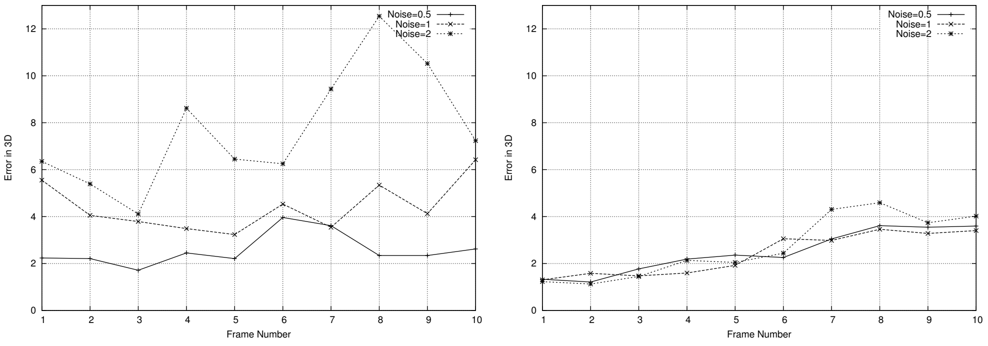
3.3 Results

We first applied our approach to synthetic data to quantitatively evaluate its performance. We obtained the meshes of Figure 3.5 by capturing the deformations of a piece of paper using a



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

Figure 3.5: Changing the support frame. (a-d) 4 of the synthetic meshes we used for our experiments. **(e)**. The three meshes recovered using (a) as the input frame and (b,c,d) in turn as the support frame. Note how similar they are.



(a) (b)

Figure 3.6: Robustness to noise. (a) Mean distance of the vertices to the ground truth using the first frame as the support frame and the others as the input frame. The curves correspond to gaussian noise of variance 0.5, 1.0, and 2.0 added to the correspondences. **(b)** Results using the same correspondences and the method of [84], which, unlike ours, requires *a priori* knowledge of the shape in the support frame.

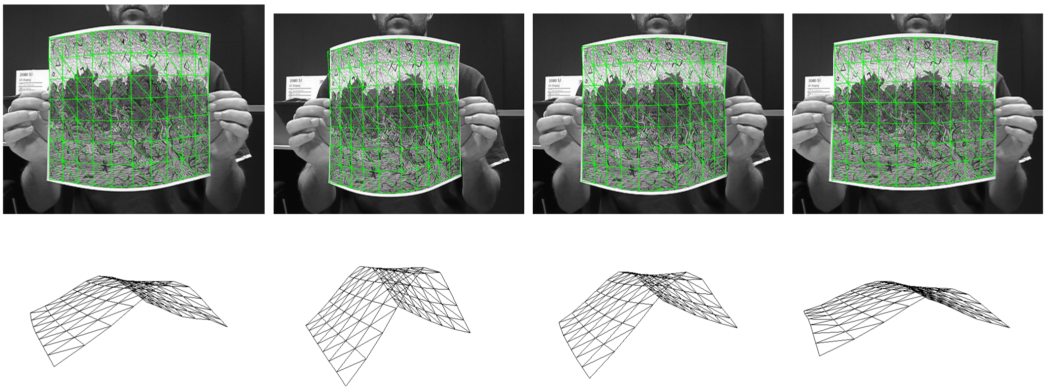


Figure 3.7: Deforming piece of paper. Top row Reconstructed 3D meshes reprojected into successive images. **Bottom row** The same meshes seen from a different viewpoint.

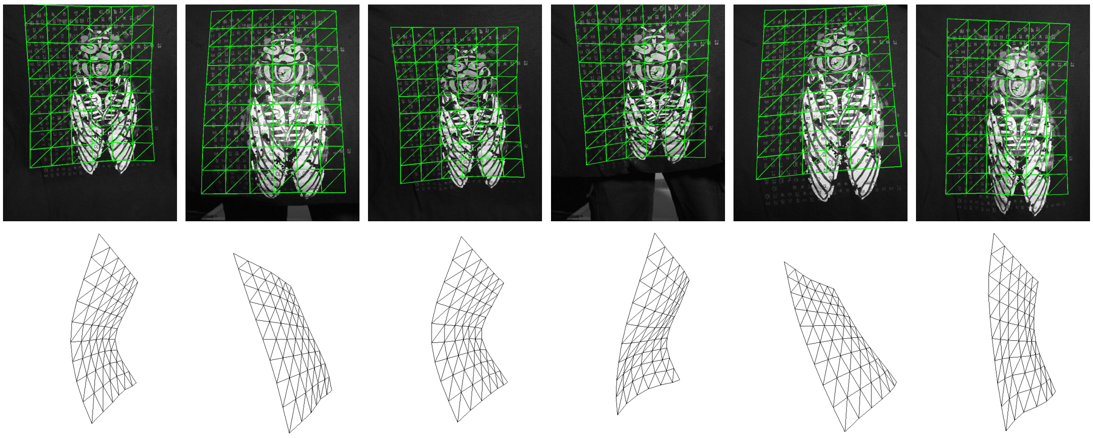


Figure 3.8: Deforming Tshirt. Top row Reconstructed 3D meshes reprojected into successive images. **Bottom row** The same meshes seen from a different viewpoint.

Vicon^{tm} optical motion capture system. We then used those to create synthetic correspondences by randomly sampling the mesh facets and projecting them using a known projection matrix and adding varying amounts of noise to the resulting image coordinates.

In Figure 3.5(e), we superpose the reconstructions obtained for the same input image using different images as the support frame and the ground truth mesh, without noise. Note how well superposed the reconstructed surfaces are, thus indicating the insensitivity of our approach to the specific choice of support frame. The mean distances between the recovered vertices and those of the ground truth mesh vary from 3.8 to 5.6, which is quite small with respect to 20, the length of the mesh edges before deformation.

We then used the first frame as the support frame and all the others in turn as the input frame. In the graph of Figure 3.6(a), each curve represents the mean distance between the reconstructed mesh vertices and their true positions in successive frames for a specific noise level in the correspondences. As evidenced by Figure 3.5, a mean error of 2 is very small and one of 5 remains barely visible. For comparison purposes, we implemented the method of [84] that relies on knowing the exact shape in one frame. At low noise levels, the results using the same correspondences are comparable, which is encouraging since our approach does not imply any *a priori* knowledge of the shape in any frame. At higher noise levels, however, the performance of our approach degrades faster, which is normal since we solve a much less constrained problem.

In practice, since SIFT provides inliers whose mean error is less than 2 pixels and since we use a robust estimator, this does not substantially affect our reconstructions. To demonstrate this, in Figures 3.7 and 3.8, we show results on real video sequences of a deforming piece of paper and a t-shirt.

3.4 Conclusion

We have presented an approach to deformable surface 3D reconstruction that overcomes most limitations of state-of-the-art techniques. We can recover the shape of a non-rigid surface while requiring neither points to be tracked throughout a whole video sequence nor a reference image in which the surface shape is known. We only need a pair of images displaying the surface in two different configurations and with enough texture to establish correspondences. We believe this to be both a minimal setup for which a correspondence-based 3D shape recovery technique could possibly work and a practical one for real-world applications.

While texture-based methods, such as the one proposed in this chapter, have proved effective in surface reconstruction tasks, they are ill-equipped to handle partially-textured surfaces. In Chapter 4, we propose a novel approach to recovering the 3D shape of a deformable surface from a monocular input by exploring shading as well as the textural information.

CHAPTER

FOUR

RECONSTRUCTION OF LOCALLY TEXTURED SURFACES

4.1 Introduction

Most recent approaches to monocular non-rigid 3D shape recovery rely on exploiting point correspondences and work best when the whole surface is well-textured. The alternative is to rely either on contours or shading information, which has only been demonstrated in very restrictive settings. Here, we propose a novel approach to monocular deformable shape recovery that can operate under complex lighting and handle partially textured surfaces. At the heart of our algorithm are a learned mapping from intensity patterns to the shape of local surface patches and a principled approach to piecing together the resulting local shape estimates. We validate our approach quantitatively and qualitatively using both synthetic and real data.

Many algorithms have been proposed to recover the 3D shape of a deformable surface from either single views or short video sequences. The most recent approaches rely on using point correspondences that are spread over the entire surface [28, 31, 70, 80, 90, 93, 107, 122], which requires the surface to be well-textured. Others avoid this requirement by exploiting contours, but can only handle surfaces such as a piece of paper where the boundaries are well defined [35, 50, 69, 121]. Some take advantage of shading information, but typically only to disambiguate the information provided by the interest points or the contours [116]. This is largely because most traditional shape-from-shading techniques can only operate under restrictive assumptions regarding lighting environment and surface albedo.

In this chapter, we propose a novel approach to recovering the 3D shape of a deformable surface from a monocular input by taking advantage of shading information in more generic contexts. This includes surfaces that may be fully or partially textured and lit by arbitrarily many light sources. To this end, given a lighting model, we propose to learn the relationship between a shading pattern and the corresponding local surface shape. At run time, we first use this knowledge to recover the shape of surface patches and then enforce spatial consistency between the patches to produce a global 3D shape.

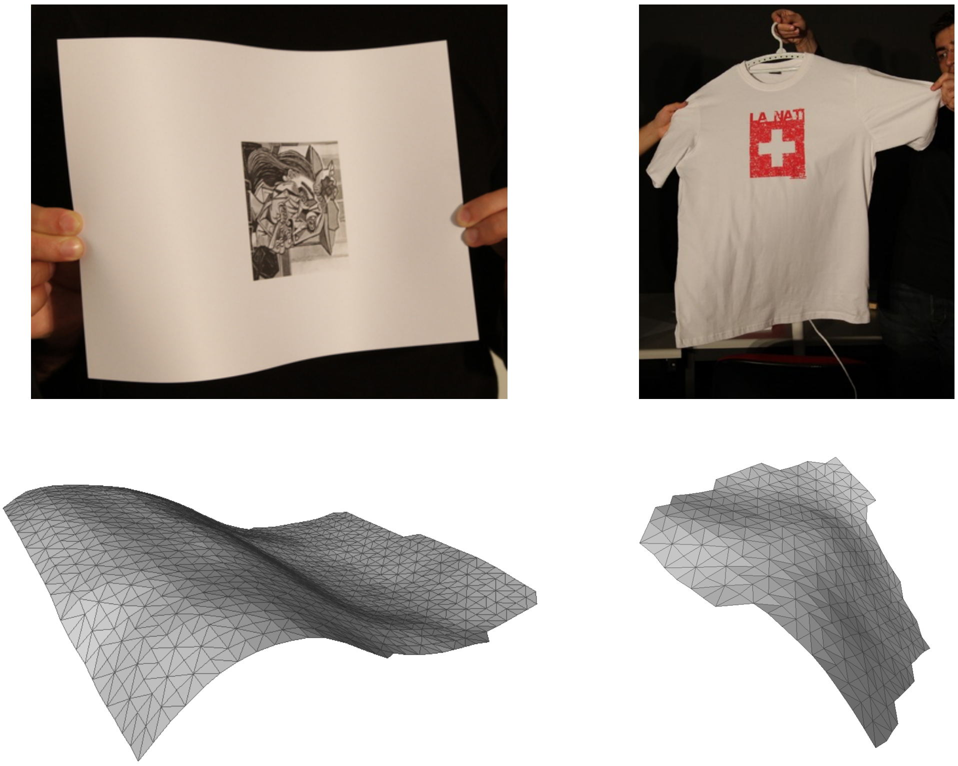


Figure 4.1: 3D reconstruction of two poorly-textured deformable surfaces from single images.

More specifically, we represent surface patches as triangulated meshes whose deformations are parametrized as weighted sums of deformation modes. We use spherical harmonics to model the lighting environment, and calibrate this model using a light probe. This lets us shade and render realistically deforming surface patches that we use to create a database of pairs of intensity patterns and 3D local shapes. We exploit this data set to train Gaussian Process (GP) mappings from intensity patterns to deformation modes. Given an input image, we find featureless surface patches and use the GPs to predict their potential shapes, which usually yields several plausible interpretations per patch. We find the correct candidates by linking each individual patch with its neighbors in a Markov Random Field (MFR). Finally, we generate a global 3D surface by fitting a larger mesh to the resulting set of 3D patches.

We exploit texture information to constrain the global 3D reconstruction and add robustness. To this end, we estimate the 3D shape of textured patches using a correspondence-based technique [80] and add these estimates into the Markov Random Field. In other words, instead of treating texture as noise as in many shape-from-shading approaches, we exploit it as an additional source of information. In short, our contribution is an approach to shape-from-shading that can operate in a much broader context than earlier ones: We can handle indifferently weak or full perspective cameras; the surfaces can be partially or fully textured; we can handle any lighting environment that can be approximated by spherical harmonics; there is no need to pre-segment the surface and we return an exact solution as opposed to one up to a scale factor. While some earlier methods address subsets of these problems, we are not aware of any that tackles them all. We demonstrate the effectiveness of our approach on synthetic and real images, and show that it outperforms state-of-the-art texture-based shape recovery and shape-from-shading techniques.

4.2 Method Overview

Our goal is to recover the 3D shape of deforming surfaces such as those shown in Figure 4.1 from a single *input image*, given a *reference image* in which the shape is known, a calibrated camera, and a lighting model. We assume that the surface albedo is constant, except at textured regions, and measure it in the reference image. Our approach relies on several insights:

- The deformations of local surface patches are simpler to model than those of the whole surface.
- For patches that are featureless, one can learn a relationship between gray-level variations induced by changes in surface normals and 3D shape that holds even when the lighting is complex.
- For patches that fall on textured parts of the surface, one can use preexisting correspondence-based techniques [80].

This patch-based approach allows the use of different techniques for different patches depending on the exact nature of the underlying image. In practice, the local reconstruction

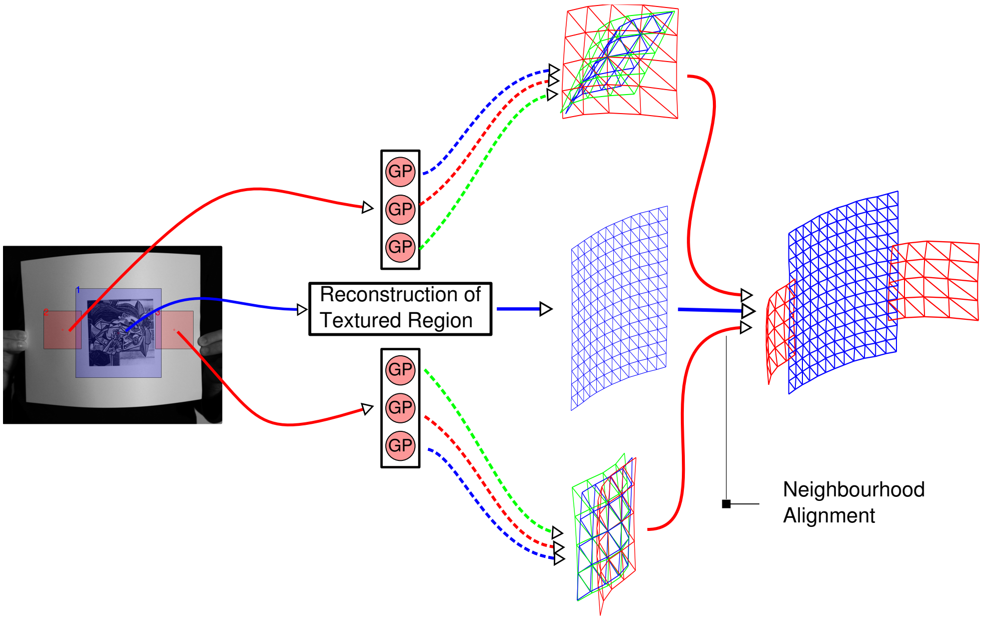


Figure 4.2: Algorithmic flow. We partition the image into patches, some of which are labeled as textured and others as featureless. We compute the 3D shape of textured patches such as the blue one by establishing point correspondences with a reference image in which the shape is known. We use Gaussian Processes trained on synthetic data to predict plausible 3D shapes for featureless patches such as the red ones. Finally, neighborhood alignment of the patches is done using a Markov Random Field to choose among all possible local interpretations those that are globally consistent.

problems may have several plausible solutions and obtaining a global surface requires a final step to enforce global geometric consistency across the reconstructed patches. The algorithm corresponding to our approach is depicted by Figure 4.2. Its two key steps are the estimation of local 3D surface shape from gray level intensities across image patches followed by the enforcement of global geometric consistency. We outline them briefly below and discuss them in more details in the two following sections.

4.2.1 Estimating the Shape of Local Patches

While we can reconstruct the 3D shape of textured patches by establishing correspondences between the feature points they contain and those points in the reference image [80], this can obviously not be done for featureless ones. For those, we infer shape from shading-induced gray-level variations. Since there is no simple algebraic relationship between intensity patterns and 3D shape when the lighting is complex, we use a Machine Learning approach to establish one.

More specifically, we learn GP mappings from intensity variations to surface deformations using a training set created by rendering a set of synthetically deformed 3D patches shaded using the known lighting model. As we will see, this is a one-to-many mapping since a given intensity pattern can give rise to several interpretations.

4.2.2 Enforcing Overall Geometric Consistency

Because there can be several different interpretations for each patch, we must select the ones that result in a consistent global 3D shape. To this end, we link the patches into an MRF that accounts for dependencies between neighboring ones. Finding the maximum a posteriori state of the MRF then yields a consistent set of local interpretations.

Although not strictly necessary, textured patches, which can be reconstructed accurately in most cases, help better constrain the process. In essence, they play the role of boundary conditions, which are always helpful when performing shape-from-shading type computations.

4.3 Estimating Local Shape

As outlined above, our method begins by reconstructing local surface patches from intensity profiles, which we do using a statistical learning approach. To this end, we calibrate the scene lighting, create a training database of deformed 3D patches and corresponding intensity profiles, and use GPs to learn the mapping between them.

4.3.1 Generating Training Data

Since shading cues are specific to a given lighting environment, we begin by representing it in terms of spherical harmonics coefficients that we recover using a spherical light probe. As scene irradiance is relatively insensitive to high frequencies in the lighting, for Lambertian objects lit by far lighting sources, we can restrict ourselves to the first nine such coefficients [76]. In practice, this has proved sufficient to operate in an everyday environment, such as our office pictured in Figure 4.3, which is lit by large area lights and extended light sources.

To populate our training database, we take advantage of the availability of a set of realistically deforming surface patches, represented by 5×5 grids of 3D points. It was acquired by attaching 3mm wide hemispherical reflective markers to pieces of cloth , which were then

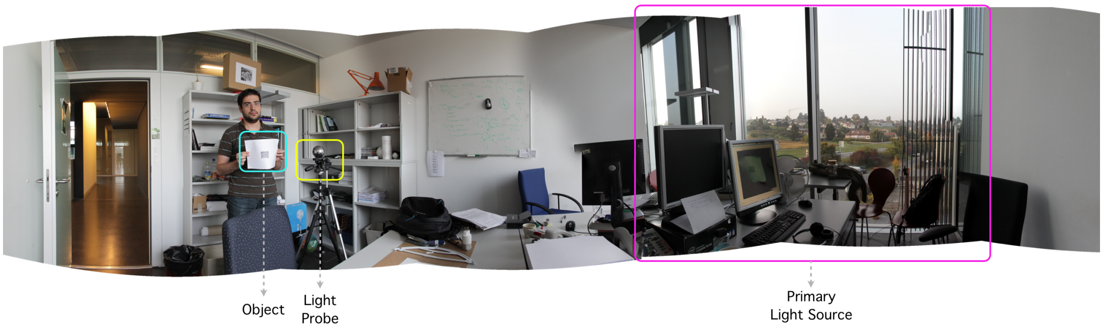


Figure 4.3: Panoramic image of the environment in which we performed our experiments

waved in front of six infrared Vicon$^{\mathrm{TM}}$ cameras to reconstruct the 3D positions of the markers. For each 3D patch, we use a standard Computer Graphics method [76] to render the patches as they would appear under our lighting model.

As a result, our training database contains pairs of 2D intensity profiles and their corresponding 3D shapes. In practice, we use 101×101 intensity patches and 5×5 3D patches, which could mean learning a mapping from an 10201-dimensional space into an 75-dimensional one. It would require data with large number of samples and be computationally difficult to achieve. Furthermore, as Lambertian surfaces evenly scatter the incoming light, they can be viewed as low-pass filters over the incident illumination. Thus, the high-frequency intensity variations tend to supply relatively little shape information and are mostly induced by noise. We therefore reduce the dimensionality of our learning problem by performing Principal Component Analysis (PCA) on both the intensity patches and the corresponding 3D deformations, and discarding high-frequency modes.

Performing PCA on the intensity patches produces an orthonormal basis of *intensity modes* and a *mean intensity patch*, as depicted by the top row of Figure 4.7. Each intensity mode encodes a structured deviation from the mean intensity patch. More formally, a square intensity patch $\mathtt{I} \in \mathbb{R}^{\mathrm{w}\times\mathrm{w}}$ of width w can be written as

$$\mathtt{I} = \mathtt{I}_0 + \sum_{i=1}^{N_I} x_i \mathtt{I}_i \ , \tag{4.1}$$

where $\mathtt{I}_0$ is the mean intensity patch, the $\mathtt{I}_i s$ are the intensity modes, the $x_i s$ are the modal weights that specify the intensity profile of the patch, and N_I denotes the number of modes. Note that, even though we learn the modes from patches of width w, we are not restricted to that size because we can uniformly scale the modes to the desired size at run-time. As a result, the mode weights will remain invariant for similar intensity profiles at different scales.

Similarly, we parametrize the shape of a 3D surface patch as the deformations of a mesh around its undeformed state. The shape can thus be expressed as the weighted sum of *deformation modes*

$$\mathbf{Y} = \mathbf{Y}_0 + \sum_{i=1}^{N_s} \Lambda_i c_i \tag{4.2}$$

where $\mathbf{Y}_0$ is the undeformed mesh configuration, the $\Lambda_i s$ are the deformation modes, the $c_i s$ are the modal weights, and N_s is the number of modes.

The modes are obtained by performing PCA over vectors of vertex coordinates from many examplars of inextensibly deformed surface patches, obtained from motion capture data [85], such as those depicted by Figure 4.7. Since they are naturally ordered by increasing levels of deformation, the first three always correspond to translations in the X, Y and Z directions and the next three to a linear approximation of rotations around the three axes. We discard the in-plane deformation modes because they do not affect local patch appearance.

This being done, for each training sample, we now have intensity modal weights $[x_1, \cdots, x_{N_I}]$ and deformation modal weights $[c_1, \cdots, c_{N_s}]$.

4.3.2 From Intensities to Deformations

Our goal is to relate the appearance of a surface patch to its 3D shape. In our context, this means using our database to learn a mapping

$$\mathcal{M} : [x_1, \cdots, x_{N_I}] \mapsto [c_1, \cdots, c_{N_s}] \tag{4.3}$$

that relates intensity weights to deformation weights, as illustrated by Figure 4.4. Given $\mathcal{M}$, the 3D shape of a patch that does not belong to the database can be estimated by computing its intensity weights as the dot product of the vector containing its intensities and the intensity modes, mapping them to deformation modes, and recovering the 3D shape from Eq. (4.2).

4.3.2.1 Gaussian Processes

Given N training pairs of intensity and deformation modes $\left[(\mathbf{x}^1, \mathbf{y}^1), \cdots, (\mathbf{x}^N, \mathbf{y}^N)\right]$, our goal is to predict the output $\mathbf{c}' = \mathcal{M}(\mathbf{x}')$ from a novel input $\mathbf{x}'$. Since the mapping from $\mathbf{x}$ to $\mathbf{c}$ is

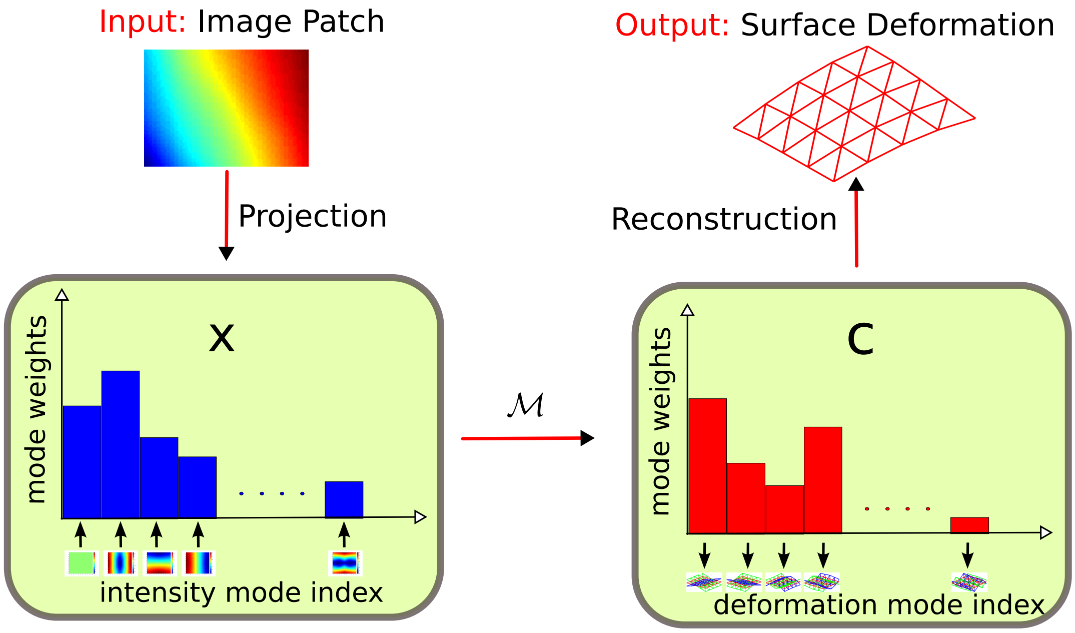


Figure 4.4: Mapping from intensity to surface deformation. Projecting an intensity patch to the set of orthogonal intensity modes produces a set of intensity modal weights $\mathbf{x}$ that describe its intensity profile. Given a mapping $\mathcal{M}$ from these weights to the deformation modal weights $\mathbf{c}$, we reconstruct the shape of the patch in 3D.

both complex and non-linear, with no known parametric representation, we exploit the GPs' ability to predict $\mathbf{c}'$ by non-linearly interpolating the training samples $\left(\mathbf{c}^1 \cdots \mathbf{c}^N\right)$.

A GP mapping assumes a Gaussian process prior over functions, whose covariance matrix `K` is built from a covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ evaluated between the training inputs. In our case, we take this function to be the sum of a radial basis function, and a noise term

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_0 \exp\left\{-\frac{\theta_1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|_2\right\} + \theta_2 \ . \tag{4.4}$$

It depends on the hyper-parameters $\Theta = \{\theta_0, \theta_1, \theta_2\}$. Given the training samples, the behavior of the GP is only function of these parameters. Assuming Gaussian noise in the observations, they are learned by maximizing $p(\mathbf{C}|\mathbf{x}^1, \cdots, \mathbf{x}^N, \Theta)p(\Theta)$ with respect to Θ, where $\mathbf{C} = [\mathbf{c}^1 \cdots \mathbf{c}^N]^T$.

At inference, given the new input intensity patch coefficients $\mathbf{x}'$, the mean prediction $\mu(\mathbf{x}')$ can be expressed as

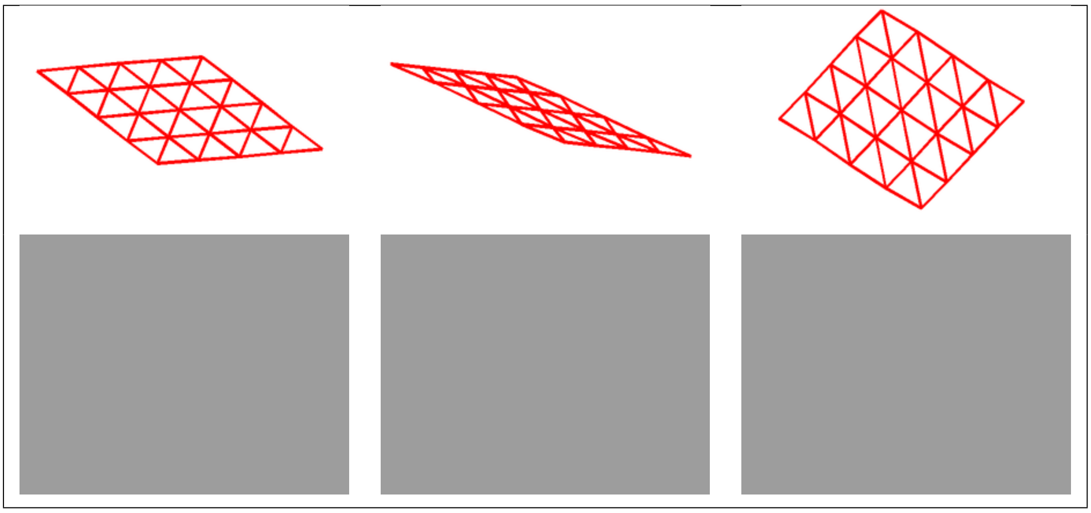
$$\mu(\mathbf{x}') \quad = \quad \mathbf{C}\mathtt{K}^{-1}\mathbf{k}(\mathbf{x}') \, , \tag{4.5}$$

where $\mathbf{k}(\mathbf{x}')$ is the vector of elements $\left[k(\mathbf{x}', \mathbf{x}^1) \cdots k(\mathbf{x}', \mathbf{x}^N)\right]$ [14]. We take $\mathbf{c}'$ to be this mean prediction.

4.3.2.2 Partitioning the Training Data

The main difficulty in learning the mapping $\mathcal{M}$ is that it is not a function. Even though going from deformation to intensity can be achieved by a simple rendering operation, the reverse is not true. As shown in Figure 4.5, many different 3D shapes can produce identical, or nearly identical, intensity profiles. These ambiguities arise from multiple phenomena, such as rotational ambiguity, convex-concave ambiguity [46], or bas-relief ambiguity [9].

As a result, many sets of deformation weights can correspond to a single set of intensity weights. Since GPs are not designed to handle one-to-many mappings, training one using all the data simultaneously produces meaningless results. Observing the ambiguous configurations reveals that the ambiguity is particularly severe when the surface patch remains planar and only undergoes rotations. Recall that the principal components of out-of-plane rotations are encoded by the first two deformation modes, which are depicted at the bottom left of Figure 4.7 and the corresponding c_1 and c_2 weights. In Figure 4.6(a) we plot the contour curves for the rendered intensities of planar patches in various orientations obtained by densely sampling c_1,c_2 space. This shows that there are infinitely many combinations of c_1 and c_2 that represent a planar patch with the same intensity. Since c_1 and c_2 encode the amount of out-of-plane rotation, a line emanating from the center of the iso-contours in the c_1,c_2 space defines a particular surface normal orientation and, within angular slices, such as those depicted by the alternating green and white quadrants of Figure 4.6(a), the surface normal of the corresponding patch remains within a given angular distance of an average orientation. We can therefore reduce the reconstruction ambiguities by splitting the c_1,c_2 space into such angular slices and learning one local GP per slice. In practice, we use 20 local GPs to cover the whole space. This resembles the clustering scheme proposed in [106], but with a partitioning scheme adapted to our problem. Other schemes, such as defining boxes in the c_1 and c_2 dimensions, would of course have been possible. However, since the dominant source of ambiguity appears to be the average surface normal that is encoded by the ratio of c_1 to c_2, we experimentally found our angular partitioning to be more efficient than others.


(a)

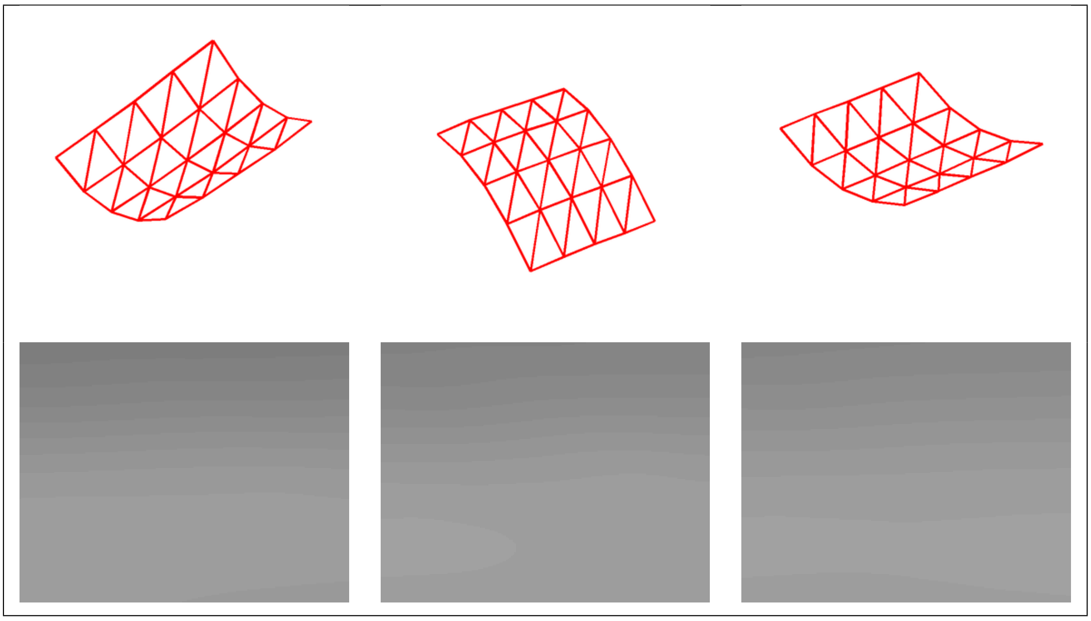
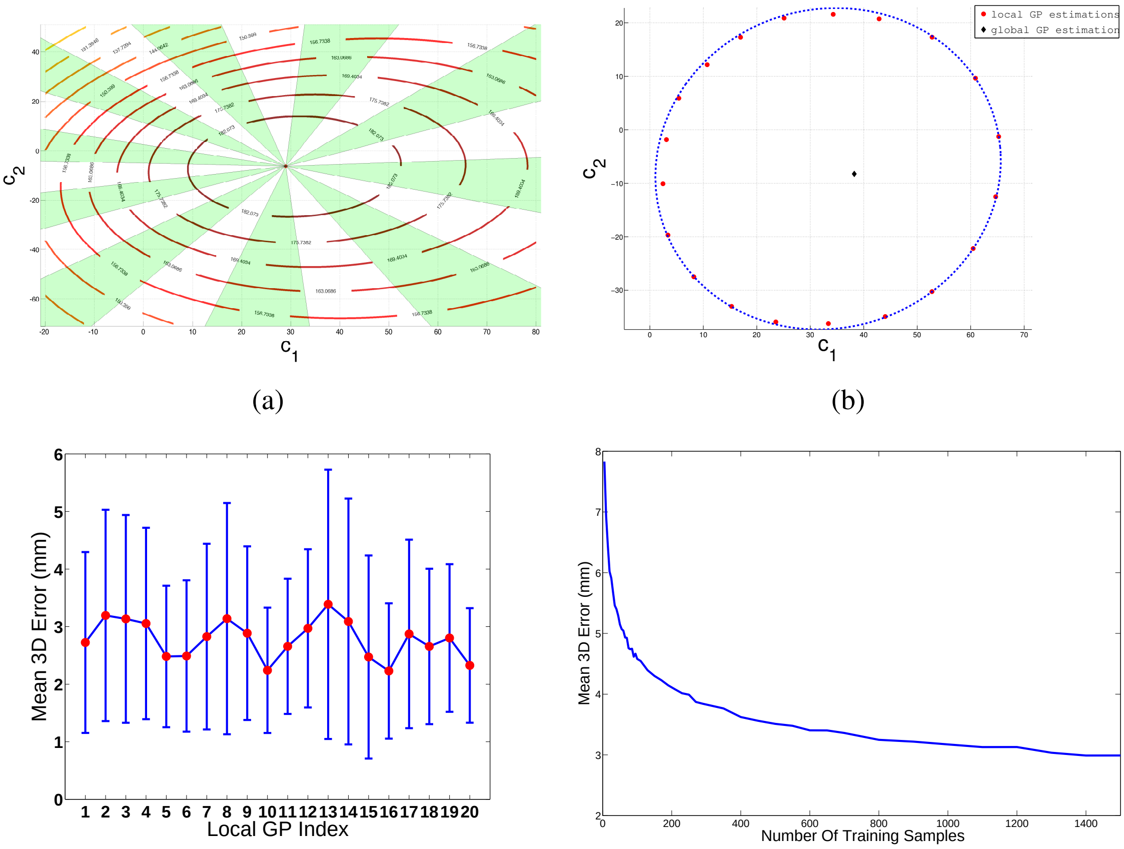

(b)

Figure 4.5: Ambiguities for flat (a) and deformed (b) surfaces. First rows Three different 3D surfaces. **Second rows** Corresponding intensity patches. Even though the 3D shapes are different, their image appearances are almost identical.

In Figure 4.6(b), we demonstrate the benefit of using local GPs over a global one to reconstruct a uniform flat patch from its intensities. The predictions from multiple GPs correctly sample the iso-intensity contour that encodes the family of all orientations producing the same



(a) (b)

(c) (d)

Figure 4.6: Single vs Multiple GPs. **(a)** Given a uniform intensity patch, there are infinitely many 3D planar patches that could have generated it. In our scheme, they are parametrized by the c_1 and c_2 weights assigned to the first two deformation modes, which encode out-of-plane rotations. The ovals represent iso-intensity values of these patches as a function of c_1 and c_2. **(b)** If we train a GP using *all* the training samples simultaneously, it will predict the same erroneous surface orientation depicted by the black dot for any uniform intensity patch. If we first partition the training samples according to angular slices shown in green and white in (a) and train a GP for each, we can predict the patch orientation shown as blue dots, which are much closer to the true orientations shown in red. **(c)** Mean and variance of the vertex-to-vertex distance between the predicted patch deformations and the ground-truth shapes for each local GP. **(d)** Accuracy of a local GP as a function of the number of training samples. GPs are accurate even when using as few as 1000 samples. In our experiments, for each local GP, we use 1400 samples on average from the training set.

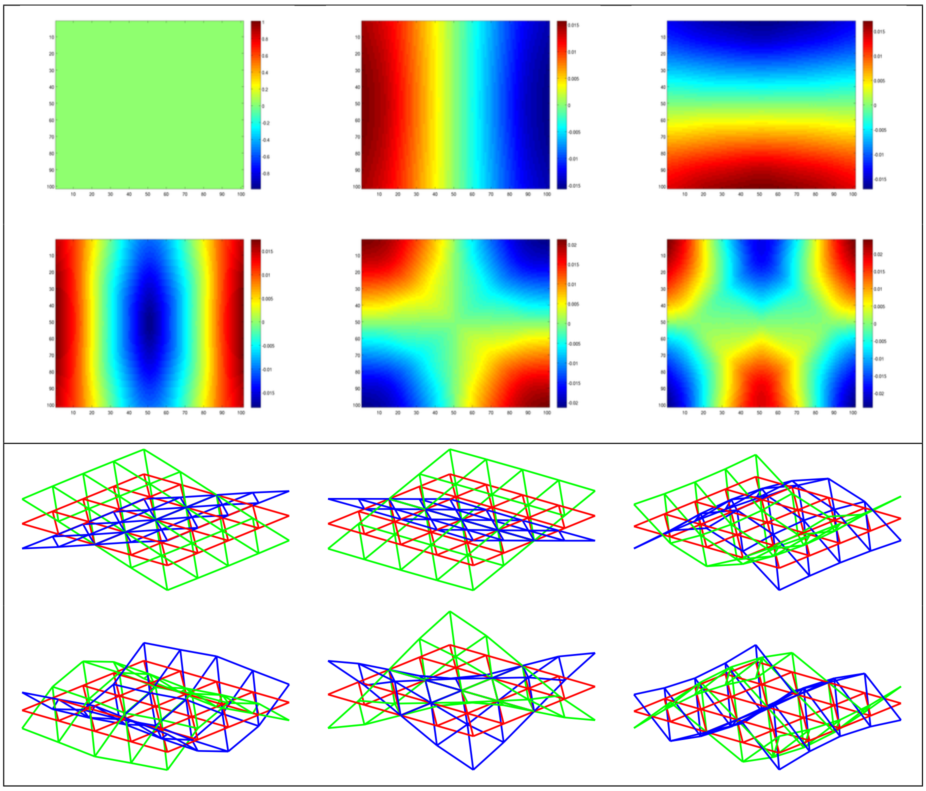


Figure 4.7: Intensity and Deformation Modes. Top Block. A subset of the low-frequency intensity modes. **Bottom Block.** A subset of the low-frequency deformation modes. The top-left and middle ones encode out-of-plane rotations and the following ones are bending modes.

intensity. In Figure 4.6(c), we consider the case of a deformed patch and plot the mean and variance values of the vertex-to-vertex distances between the prediction and ground-truth. For each slice we tested 100 unique patch deformations while training over 1000 data points. We repeated this 100 times. The average reconstruction error of 3 millimeters is small, considering that the average patch side is 100 millimeters long. This indicates that, within each partition, there is a one-to-one correspondence between intensity and deformation mode weights. Otherwise, the GP mapping could not produce this accuracy.

One attractive feature of GPs is that they can be learned from a relatively small training set. We estimate the required size empirically by measuring the accuracy of the mapping, given by the average vertex-to-vertex distance between the prediction and ground truth data, as a function of the number of training samples. For a given size, we draw 100 independent subsets of samples of that size from our training set. For each subset, we test the accuracy using 100 other instances from the test set. The resulting mean error is depicted by Figure 4.6(d).

4.3.3 Local Reconstructions from an Input Image

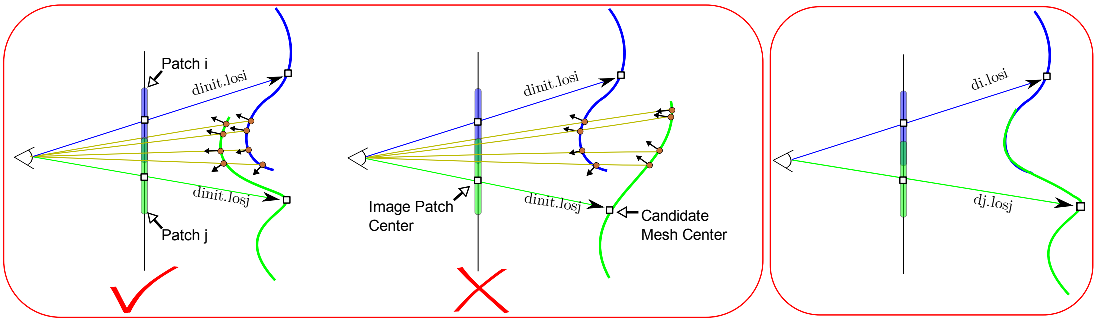
At run time, we first identify the textured patches by extracting SIFT interest points and establishing point correspondence with the reference image. They are used to recover their 3D-shape using the correspondence-based method [80]. We then scan the remainder of the image multiple times with square sliding windows of varying sizes, starting with a large one and progressively decreasing its size. During each scan, the windows whose intensity variance is greater than a threshold are discarded. The remaining ones are projected into the learned intensity mode space and retained if their mode-space distance to their nearest neighbor in the training set is smaller than a threshold. In successive scans, we ignore areas that are completely subsumed by previously selected windows. Finally, we run a connected component analysis and keep only the patches that are connected directly or indirectly to the textured one. In all our experiments we keep the maximum acceptable standard deviation of intensities in a patch to be 30 units and mode-space distance to be 10.

Given a set of featureless patches and N_{GP} Gaussian Processes, one for each angular partition of the training data, we therefore predict N_{GP} shape candidates per patch represented as 5×5 meshes. We initially position them in 3D with their center at a fixed distance along the line of sight defined by the center of the corresponding image patch.

4.4 Enforcing Global Consistency

Local shape estimation returns a set $\mathcal{S}_p = \{\mathtt{S}_p^1, \cdots, \mathtt{S}_p^{N_{GP}}\}$ of plausible shape interpretations reconstructed up to a scale factor for each patch p, and a single one $\mathtt{S}_{p'}$ for each textured patch p'. To produce a single global shape interpretation, we go through the two following steps.

First, we choose one specific interpretation for each featureless patch. To this end, we use a MRF to enforce global consistency between the competing interpretations in a way that does not require knowing their scales. Second, we compute the scale of each patch, or equivalently its distance to the camera, by solving a set of linear equations. In the remainder of this section, we describe these two steps in more details.



(a)

(b)

Figure 4.8: Enforcing Shape Consistency (a) Two different instances of the evaluation of the geometric consistency of patches i and j, shown in blue and green respectively. In both cases, the predicted normals of points along the same lines of sight, drawn in yellow, are compared. Since these points have the same projections, their normals should agree. Thus, the patches on the left are found to be more consistent than those on the right. **(b)** Moving patches along their respective lines of sight. The patches i and j are moved to distances d_i and d_j from the optical center so as to minimize the distance between them in their regions of overlap.

4.4.1 Selecting one Shape Interpretation per Patch

To select the correct interpretation for individual patches, we treat each one as a node in an MRF graph. Featureless patches can be assigned one of the N_{GP} labels corresponding to the elements of $\mathcal{S}_p$, while textured ones are assigned their recovered shape label.

We take the total energy of the MRF graph to be the sum over all the featureless local patches

$$\mathtt{E} = \sum_{p} \left(\mathtt{E}_1(\mathtt{S}_p) + \frac{1}{2} \sum_{q \in \mathcal{O}(p)} \mathtt{E}_2(\mathtt{S}_p, \mathtt{S}_q) \right) , \tag{4.6}$$

where $\mathcal{O}(p)$ is the set of patches that overlap p. The unary terms $\mathtt{E}_1$ favor shapes whose shaded versions match the image as well as possible. The pairwise terms $\mathtt{E}_2$ favor geometric consistency of overlapping shapes.

In practice, we take $\mathtt{E}_1(\mathtt{S}_p)$ to be the inverse of the normalized cross correlation score between the image patch and the rendered image of the 3D shape. To evaluate the pairwise term $\mathtt{E}_2(\mathtt{S}_p, \mathtt{S}_q)$ for overlapping patches p and q, we shoot multiple camera rays from the camera center through their common projection area, as shown in Figure 4.8 (a). For each ray, we compare the normals of the two 3D shapes and take $\mathtt{E}_2(\mathtt{S}_p, \mathtt{S}_q)$ to be the mean L2 norm of the difference between the normals.

Note that both the unary and pairwise terms of Eq. (4.6) can be evaluated without knowing the scale of the patches, which is essential in our case because it is indeed unknown at this stage of the computation. We use a tree re-weighted message passing technique [45] to minimize the energy. In all of our experiments, the primal and dual programs returned the same solution [11], which indicates the algorithm converged to a global optimum even though the energy includes non sub-modular components.

4.4.2 Aligning the Local Patches

Having assigned a specific shape $\mathtt{S}_p$ to each patch, we now need to scale these shapes by moving them along their respective lines of sight, which comes down to computing the distances d_p from the optical center to the patch centers. In the camera referential, the line of sight defined by the center of patch p emanates from the origin and its direction is

$$\mathrm{los}_p = \frac{\mathbf{A}^{-1}\mathbf{u}_p}{\|\mathbf{A}^{-1}\mathbf{u}_p\|_2} \ , \tag{4.7}$$

where $\mathbf{A}$ is the 3×3 matrix of internal camera parameters and $\mathbf{u}_p$ represents the projective coordinates of the patch center.

To enforce scale consistency between pairs of overlapping patches p and q, we consider the same point samples as before, whose projections lie in the overlap area as shown in Figure 4.8 (b). Let $[x_p, y_p, z_p]^T$ and $[x_q, y_q, z_q]^T$ be the 3D coordinates of the vectors connecting such a sample to the centers of p and q, respectively. Since they project to the same image location, we must have

$$d_p \left(\mathrm{los}_p^T + [x_p, y_p, z_p]\right) = d_q \left(\mathrm{los}_q^T + [x_q, y_q, z_q]\right) \ . \tag{4.8}$$

Each sample yields one linear equation of the form of Eq. (4.8). Thus, given enough samples we can compute all the d_p up to a global scale factor by solving the resulting system of equations in the least-squares sense. If there is at least one textured patch whose depth can be recovered accurately, the global scale can be fixed and this remaining ambiguity resolved.

4.4.3 Post Processing

The alignment yields a set of overlapping 3D shapes. To make visual interpretation easier, we represent them as point clouds which are computed by linearly interpolating the z values of the vertices of all the local solutions on a uniformly sampled xy grid. For display purposes, we either directly draw these points or the corresponding Delaunay triangulation.

4.5 Results

In this section, we demonstrate our method's ability to reconstruct different kinds of surfaces. In all these experiments, we learned 20 independent GPs by partitioning the space of potential surface normals, as discussed in Section 4.3.2. For training purposes, we used 28000 surface patches, or approximately 1400 per GP. They are represented as 5×5 meshes and rendered using the calibrated experiment-specific lighting environment. The calibration and the training process jointly take approximately two hours to complete on a standard machine with a 2.4 GHz processor.

In the remainder of this section, we first use synthetic data to analyze the behavior of our algorithm. We then demonstrate its performance on real data and validate our results against ground-truth data.

Since our images contain both textured and non-textured parts, we compare our results to those obtained using our earlier technique [80] that relies solely on point correspondences to demonstrate that also using the shading information does indeed help. We also compare against pure Shape-from-Shading algorithms described in [104] and [29] that are older but, as argued in [27], still representative of the state-of-the-art, and whose implementations are available online.

4.5.1 Synthetic Images

We first tested the performance of our algorithm on a synthetic sequence created by rendering 100 different deformations of a piece of cardboard obtained using a motion capture system. Note that this is not the same sequence as the one we used for learning the intensity to deformation mapping discussed in Section 4.3. The entire sequence is rendered using the lighting parameters corresponding to a complicated lighting environment such as the one shown in Figure 4.3. To this end, we use a set of spherical harmonics coefficients computed for that particular lighting environment. In addition, the central part of the surface is artificially texture-mapped. Figure 4.9 depicts a subset of these synthetic images, the 3D reconstructions we derive from them, and 3D reconstructions obtained with our earlier texture-based method [80]. We compute 3D reconstruction errors as the mean point-to-surface distances from the

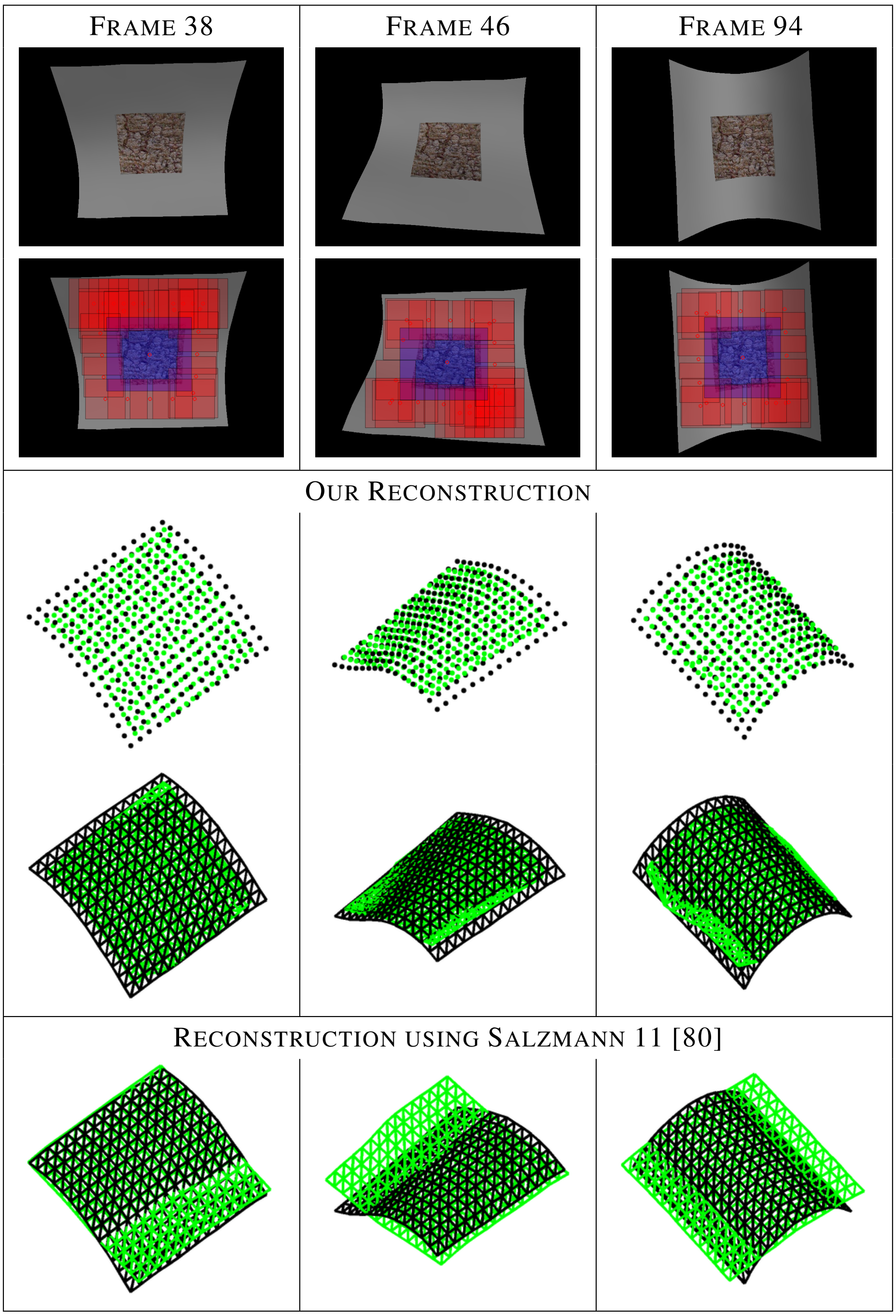


Figure 4.9: Comparisons against a texture-based method [80] on a synthetic sequence First row Input images. **Second row** Local Patches. The blue patches indicate the regions where feature correspondences are given, and the red patches are non-textured areas, selected by our patch selection algorithm. **Third row** Reconstructed point cloud (green dots) and ground-truth mesh vertices (black dots) seen from another view point. **Forth row** Estimated triangulation (green) and ground-truth triangulation (black) seen from another view point. **Fifth row** Reconstruction results using the method in [80] (green mesh) and ground-truth triangulation (black).

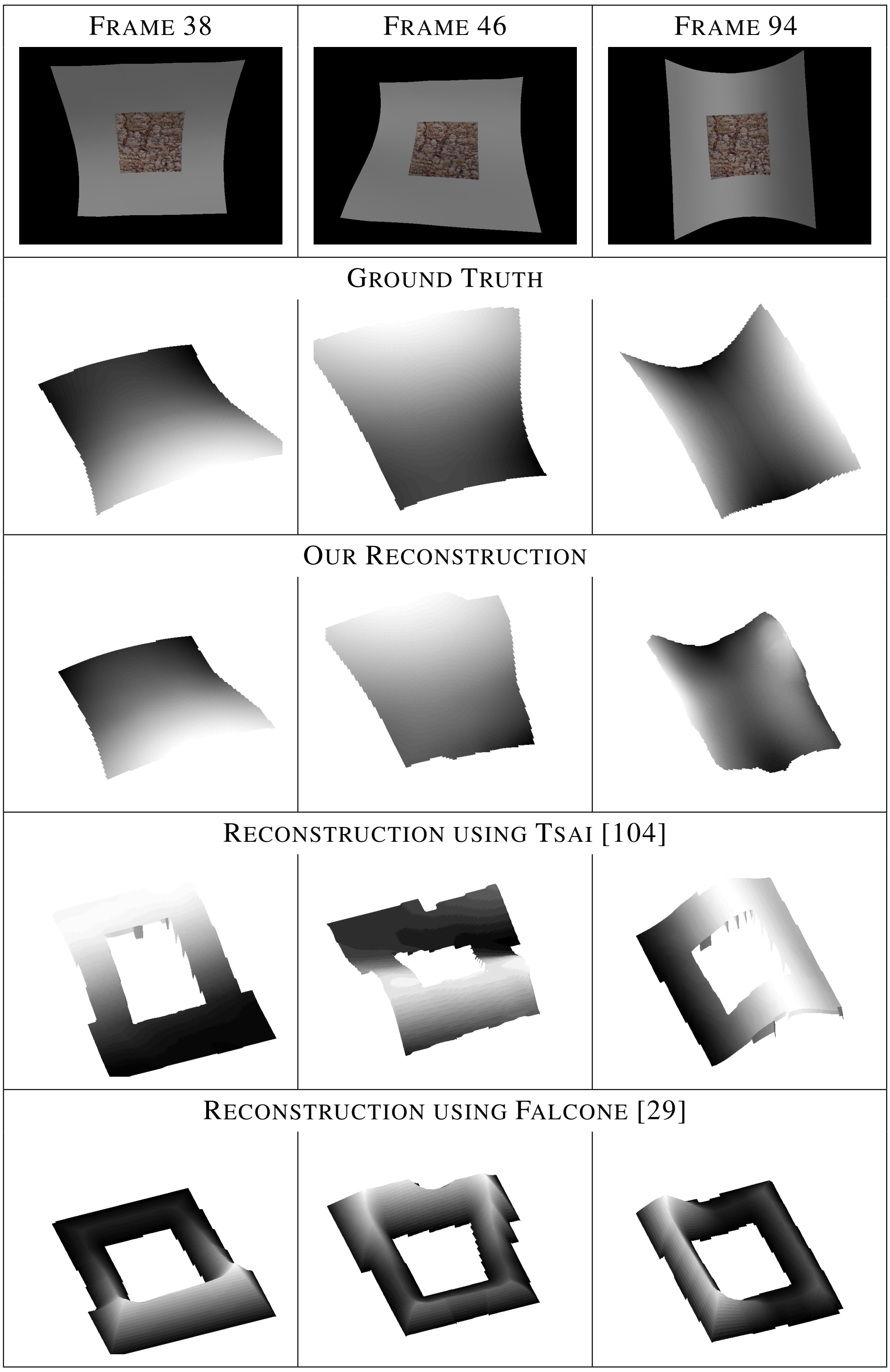

FRAME 38
FRAME 46
FRAME 94
GROUND TRUTH
OUR RECONSTRUCTION
RECONSTRUCTION USING TSAI [104]
RECONSTRUCTION USING FALCONE [29]

Figure 4.10: Comparisons against SfS methods on a synthetic sequence First row: Input images. **Second row:** Ground truth depth maps **Third row:** Depth maps computed from our reconstructions. **Forth to fifth rows:** Depth maps computed by the methods in [104], [29].

reconstructed point clouds to the ground-truth surfaces for all the frames in this sequence. The results are shown in Figure 4.11.

By combining shading and texture cues, our method performs significantly better except for flat surfaces, where both methods return similar results.In Figure 4.10, we compare our results against those of pure shape-from-shading methods [29, 104]. Our algorithm computes a properly scaled 3D surface but these methods only return a normalized depth map. For a fair comparison, we therefore computed normalized depth maps from our results. Furthermore, although our method does not require it, we provided manually drawn masks that hide the background and the textured parts of the surfaces to make the task of the Shape-from-Shading methods easier. As can be seen, in addition to being correctly scaled, our reconstructions are also considerably more accurate.

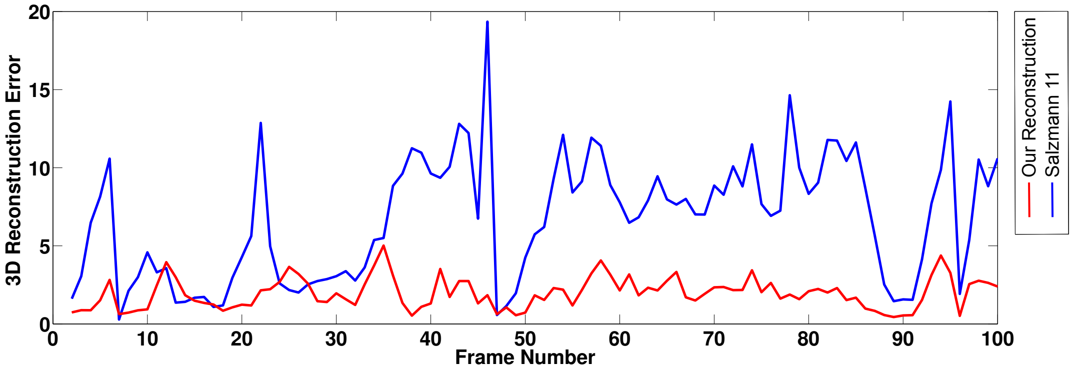


Figure 4.11: Reconstruction error of both methods for all the frames in the sequence. Note that the proposed method provides much better reconstructions, except for 6 frames in the sequence.

Because the projection of intensity patches to PCA modes can be viewed as a filtering operation, our method is robust to image noise. To demonstrate this, we introduced various level of Gaussian noise to the input image, shown in the top row of Fig. 4.12. Independently of the noise, we were able to accurately reconstruct the surface, as depicted by the blue mesh in second row of Fig. 4.12. The third and fourth rows of Fig. 4.12 show the intensity profiles of a selected patch from the noisy input images before and after the mode projection operation, respectively. As expected, using only the first few PCA modes filtered out the high frequency noise.

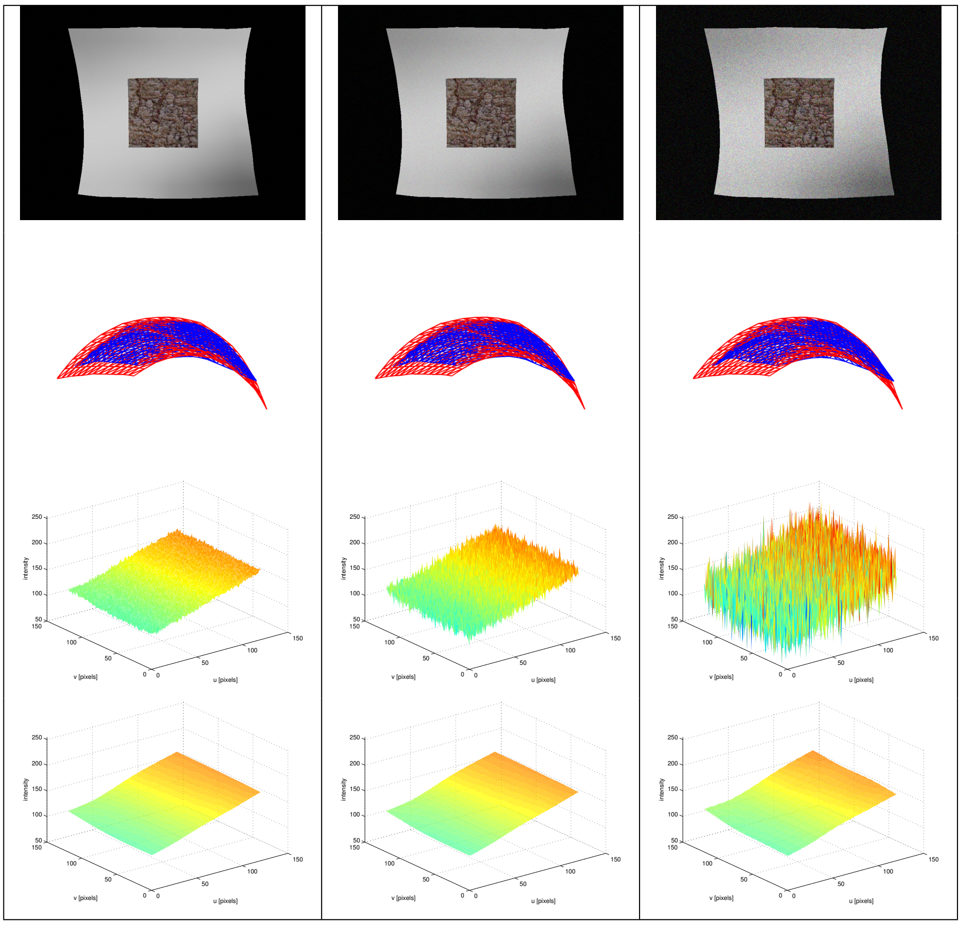


Figure 4.12: Reconstruction from noisy images. Top Row: Input images with Gaussian noise of zero mean and 3, 8, and 26 intensity variances, respectively. **Second Row:** Comparison of our reconstructions, in blue, against the ground-truth mesh in red. **Third Row:** Intensity profiles of the same patch extracted from the noisy images.**Fourth Row:** Intensity profiles of the patch after mode projection.

In theory, there is no guarantee that the reconstruction of the textured patch is correct, which could lead to reconstruction failure. An incorrect reconstruction will result in a gross error, especially since our algorithm tries to enforce global consistency with respect to this erroneous configuration. In practice, this only rarely occurs, and in all the correspondence-based experiments reported here, the algorithm we used [80] returned a valid reconstruction for the textured area. Nevertheless, our method can also handle cases when there are multiple interpretations for the textured patches by adding them as additional labels to our MRF. To demonstrate this, we generated multiple candidates for the textured patches using the sampling scheme proposed in [60]. As shown in Figure 4.13, our algorithm picks the right one from the candidate reconstructions.

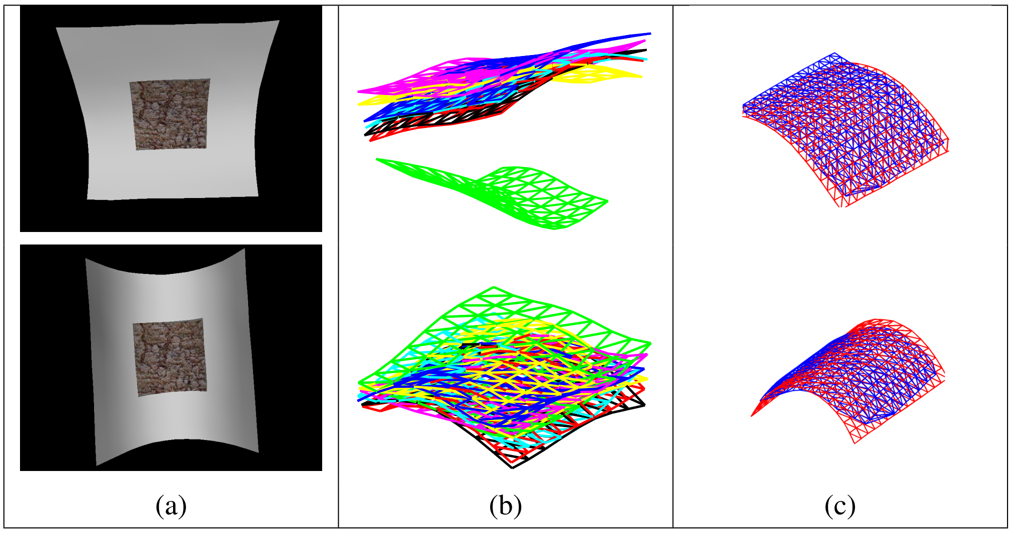

(a) (b) (c)

Figure 4.13: Reconstruction with multiple 3D hypotheses for the textured patch. (a) Input images. **(b)** 3D hypotheses for the textured patches. **(c)** 3D reconstructions of the surfaces.

4.5.1.1 Robustness to Lighting Environment

To show that our algorithm is robust to lighting changes, we rendered images of the same surface under three different lighting arrangements with either frontal, on left, or on right lighting. As shown in Figure 4.14, the three reconstructions that we obtained were all similar and close to the ground truth.

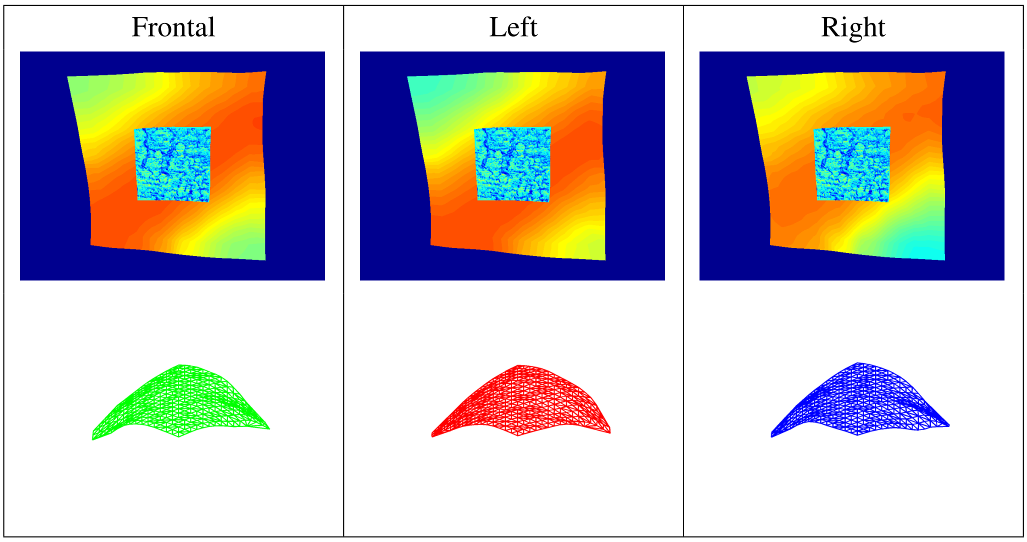


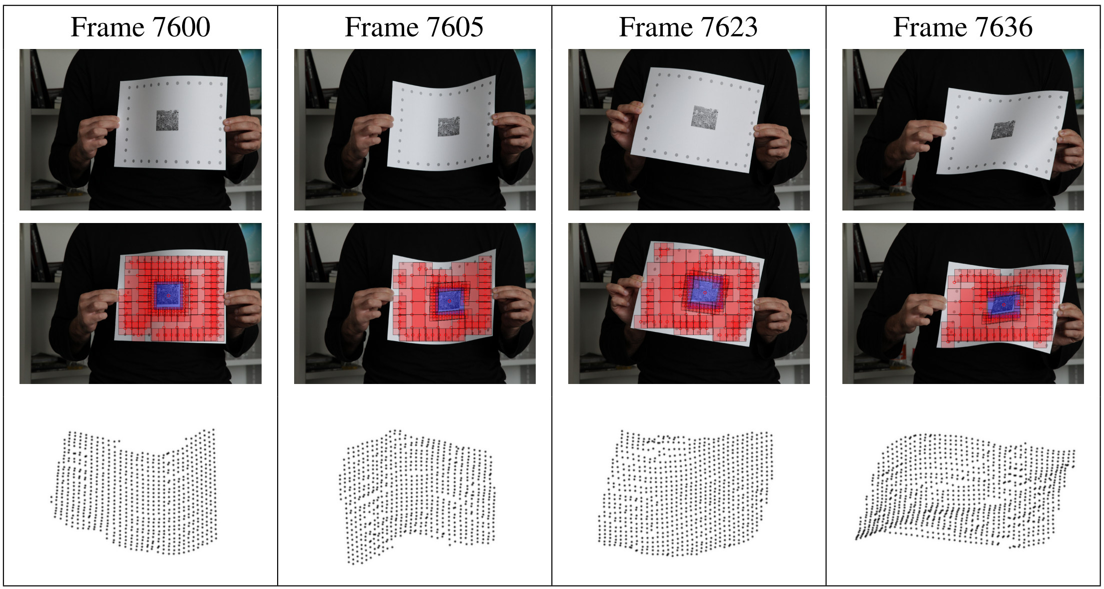
Figure 4.14: Robustness to Lighting Environment The surface is lit by three different lighting schemes. **Top row:** Intensity variation in the surface. **Bottom row:** Reconstructed surfaces.

4.5.2 Real Images

As the nature of the deformations vary considerably with respect to the surface material type, we applied our reconstruction algorithm to two surfaces with very different physical properties: the piece of paper of Figure 4.15 and the T-shirt of Figure 4.16. The deformation of the latter is significantly less constrained than that of the former. Note that because we only model the deformations of small patches that are then assembled into global surfaces, we can handle complex global deformations. However, as will be discussed below, folds that are too sharp may result in self shadowing which is not handled in our current implementation.

The real sequences were captured by a single-lens reflex (SLR) camera and recorded in raw format. The linear images were then extracted from the raw image files and the image intensities linearly scaled so that they cover most of the observable intensity range. The image resolution was approximately 5 mega-pixels.

The image patches of Section 4.3 were selected by the patch selection algorithm. In practice, we used square patches whose size ranges from 401 to 101 pixels with a 100 pixels step. We show the textured and textureless image patches selected by this procedure in the second rows of Figures 4.15 and 4.16.



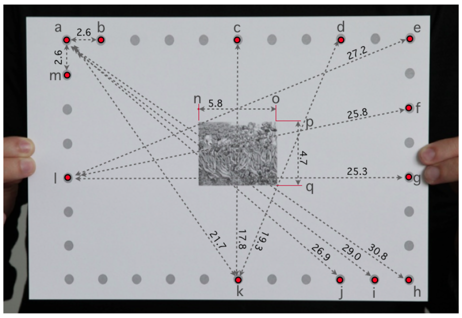
Reference Image	Point Pairs	Ground Truth	Frame Number			
			7600	7605	7623	7636
 a 2.6 b c d e 2.6 m 27.2 n 5.8 o p 25.8 f 4.7 l 25.3 g q 21.7 17.8 19.3 26.9 29.0 30.8 k j i h	a – b	2.6	2.5	2.6	2.6	3.2
	a – h	30.8	30.6	29.4	31.0	32.6
	a – i	28.9	28.2	27.9	28.9	30.1
	a – j	26.9	26.1	26.1	27.1	27.6
	a – k	21.7	21.2	22.6	22.5	22.1
	a – m	2.6	2.8	2.8	2.9	2.9
	c – k	17.8	18.0	18.0	18.4	18.9
	d – k	19.3	20.2	20.0	20.2	NA
	e – l	27.2	27.3	26.3	26.9	28.5
	f – l	25.8	26.3	24.9	25.2	27.8
	g – l	25.3	25.6	24.3	24.8	27.5
	n – o	5.8	5.9	5.9	5.9	6.1
	p – q	4.7	4.8	4.9	4.8	5.1
All distances are in cm	Avg Error		0.36	0.65	0.35	1.03

Figure 4.15: Paper Sequence. First row Input images. **Second row** Local Patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. **Third row** Reconstructed point cloud seen from another viewpoint. **Fourth row** Geodesic distances between prominent landmarks as identified on the left. Point d in frame 7636 was outside our reconstruction, which explains the missing value in the table.

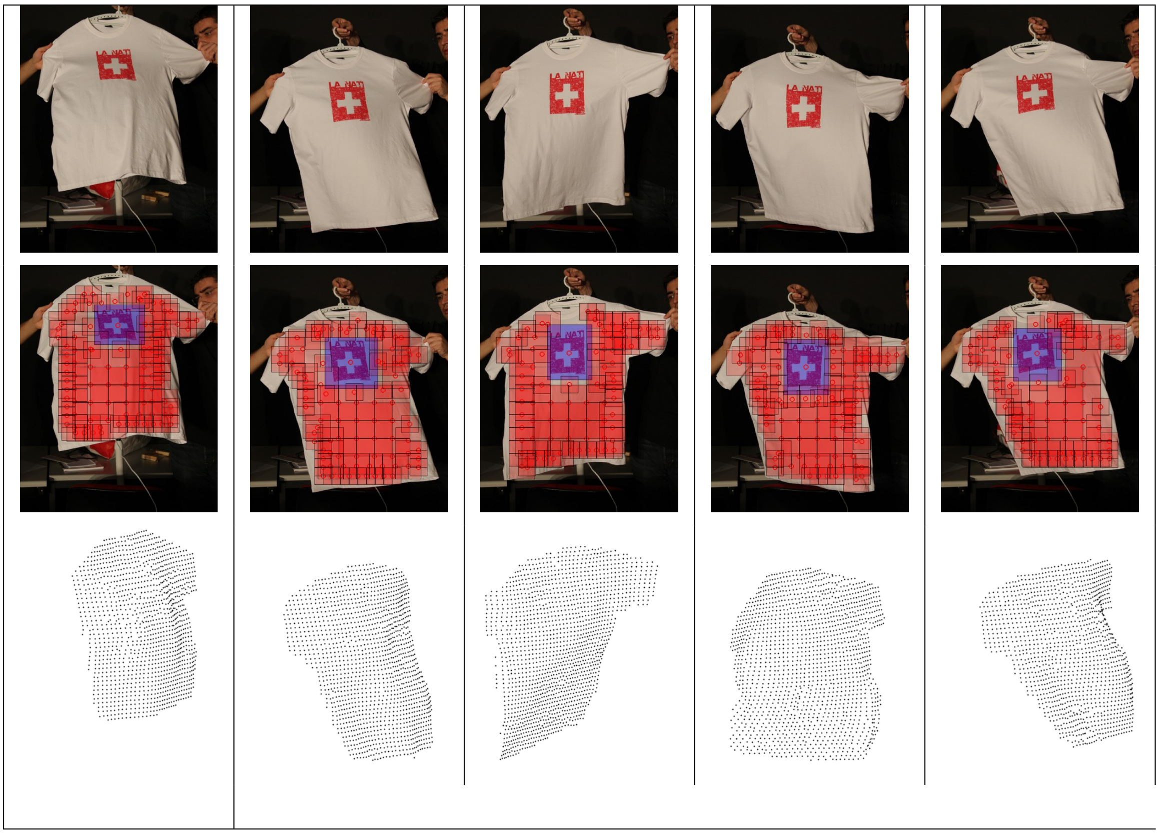


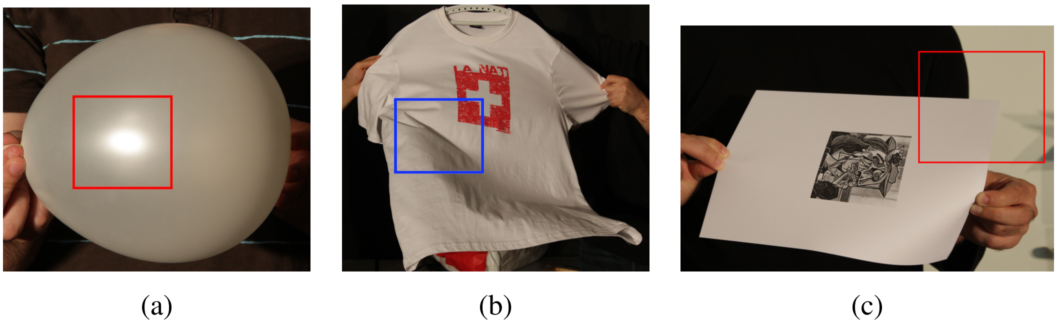
Figure 4.16: **T-Shirt Sequence.** **First row** Input images. **Second row** Local Patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. **Third row** Reconstructed point cloud seen from another viewpoint.

4.5.3 Validation

To quantitatively evaluate our algorithm's accuracy, we performed two different sets of experiments involving real data, which we detail below.

4.5.3.1 Preservation of Geodesic Distances

The geodesic distances between pairs of points, such as the circles on the piece of paper at the bottom of Figure 4.15, remain constant no matter what the deformation is because the surface is inextensible. As shown in the bottom-right table, even though we do not explicitly enforce this constraint, it remains satisfied to a very high degree, thus indicating that the global deformation is at least plausible.



(a) (b) (c)

Figure 4.17: Failure Modes. (a) Non-Lambertian surface. **(b)** Folds that create self-shadows. **(c)** Background albedo very similar to the surface.

In this example, the ground-truth geodesic distances were measured when the sheet of paper was lying flat on a table. To compute the geodesic distances on the recovered meshes, we used an adapted Gauss-Seidel iterative algorithm [16].

4.5.3.2 Comparison against Structured Light Scans

To further quantify the accuracy of our reconstructions, we captured surface deformations using a structured light scanner [115]. To this end, we fixed the shape of the same piece of paper and t-shirt as before by mounting them on a hardboard prior to scanning, as shown at the top of Figure 4.18. Because of the physical setup of the scanner, we then had to move the hardboard to acquire the images we used for reconstruction purposes. To compare our reconstructions to the scanned values, we therefore used an ICP algorithm [12] to register them together.

In the remainder of Figure 4.18, we compare the output of our algorithm to that of the same algorithms as before. These results clearly indicate that our approach to combining texture and shading cues produces much more accurate results than those of these other methods that only rely on one or the other.

4.5.4 Limitations

The main limitation of our current technique is that, outside of the truly textured regions, we assume the surface to be Lambertian and of constant albedo. As a result, we cannot reconstruct shiny objects such as the balloon shown in Figure 4.17(a). However, given the bidirectional reflectance distribution function (BRDF) of the surface points, our framework could in theory be extended to such non-Lambertian surfaces.

Like most other shape-from-shading methods, ours is ill-equipped to handle self shadows and occlusions. The effect of the latter can be mitigated to a certain extent by using temporal models. The self shadows produced by sharp folds, such as the ones shown in Figure 4.17(b), violate our basic assumptions that shading only depends on 3D shape, and could be handled by a separately trained generative model [36]. Addressing these issues is a topic for future research.

Failure may also occur when background image regions are extracted by our patch selection algorithm. Fortunately, this occurs rarely, i.e. when the background is made up of uniform regions with albedo very similar to that of the surface to be reconstructed, as shown in Figure 4.17(c). In such cases, the background patches look very similar to those of our training set and will not be filtered out.

4.6 Conclusion

We have presented an approach to monocular shape recovery that effectively takes advantage of both shading cues in non-textured areas and point correspondences in textured ones under realistic lighting conditions and under full perspective projection. We have demonstrated the superior accuracy of our approach compared to state-of-the-art techniques on both synthetic and real data.

Our framework is general enough so that each component could be replaced with a more sophisticated one. For instance, representations of the lighting environment more sophisticated than spherical harmonics could be used to create our training set. Similarly, other, potentially nonlinear parametrizations of the patch intensities and deformations could replace the current PCA mode weights.

In the next chapter, we focus on a machine learning algorithm that explicitly accounts for the constraints in its predictions. These constraints include the commonly-employed edge-length constraints which have proved effective to resolve the ambiguities in reconstructing inextensible surfaces.

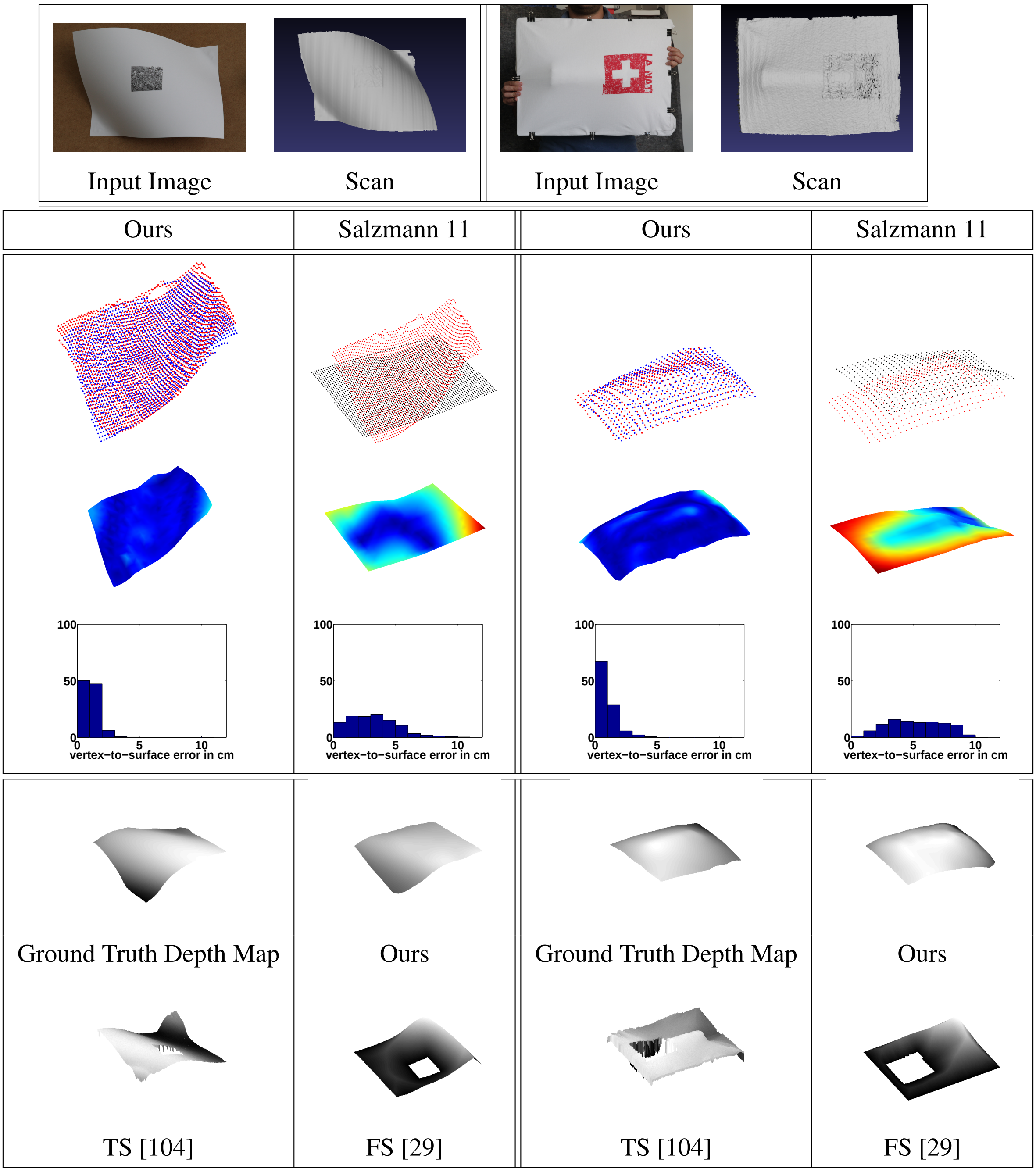


Figure 4.18: Accuracy estimation using structured light scans. Top Row. Two different surfaces and their corresponding structured light scans. **Middle Block.** From top to bottom, point clouds from the scans (red) and reconstructions by either our algorithm or that of [80] (green), reconstructed 3D surface rendered using a color going from blue to red as the vertex-to-surface distance to the ground-truth increases, and corresponding histogram of vertex-to-surface distances. **Bottom Block.** Depth maps obtained from our reconstructions and from the methods in [29, 104].

CHAPTER

FIVE

CONSTRAINED LATENT VARIABLE MODELS FOR SURFACE RECONSTRUCTION

5.1 Introduction

As we have seen in the previous chapters, latent variable models have been widely used for non-rigid pose estimation. However, as effective as they are, they suffer from the fact that they ignore prior knowledge that might be available for the specific problem at hand. In particular, nothing prevents commonly-employed latent variable models from generating configurations that violate known constraints.

In this chapter, we propose a novel non-linear latent variable model whose output explicitly accounts for the inherent constraints of the problem. To this end, we learn a non-linear mapping from the latent space to the output space such that the generated outputs comply with equality and inequality constraints expressed in terms of the problem variables. We make use of unlabeled examples to enforce the constraints, while minimizing the prediction error of labeled ones. To allow for kernel-based mappings, we introduce a primal-dual optimization framework, where the mapping is learned by sequential closed-form updates. Our approach is completely generic and could be used in many different contexts, such as image classification to impose separation of the classes, and articulated tracking to constrain the space of possible poses.

To illustrate the benefits of our model, we consider a toy case where the output is a single 3D point constrained to lie on a hemisphere. We learned a constrained latent variable model and its unconstrained version using the black dots close to the great circle of the hemisphere

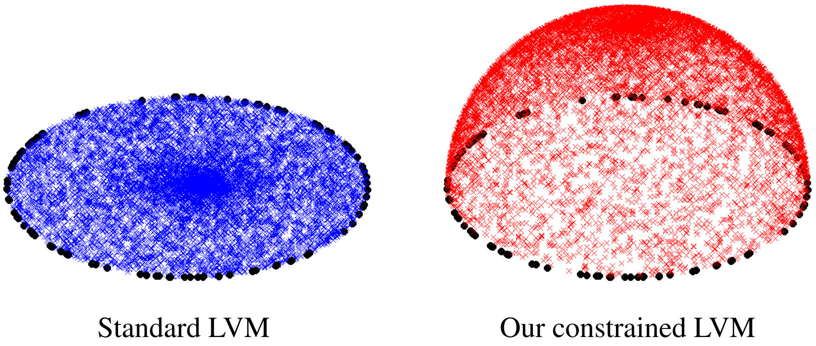


Figure 5.1: Generating 3D points on a hemisphere. (Left) Predictions from random samples in the latent space using an unconstrained latent variable model. (Right) Predictions from the same samples using our constrained latent variable model. Both models were learned using only the black dots as labeled training samples.

as labeled training examples. Figure 5.1 shows the predictions of the unconstrained and constrained models from random samples on the latent space. Note how much the predictions are improved by learning a constrained model. While this corresponds to an extreme case of poorly sampled training points, a similar scenario could easily occur locally on more complex output spaces.

In particular, this is true for the task of non-rigid surface reconstruction from monocular inputs that we address in this thesis. This is known to be a very ambiguous problem due to the fact that any point on a line of sight reprojects at the same image location, thus making depth estimation very ill-posed. Furthermore, the partial lack of identifiable texture on the surface makes the use of shape regularizers necessary to produce reasonable reconstructions. Latent variable models are commonly used for such regularization [15, 21, 30, 74, 85]. Our experimental evaluation shows that our constrained latent variable model produces more accurate reconstructions than the standard linear subspace models and the increasingly popular Gaussian Process Latent Variable Model (GPLVM) [49], which corresponds to the unconstrained version of our model. This evaluation was performed in a variety of scenarios including real images of different materials captured with the Microsoft Kinect, providing ground-truth 3D measurements.

In the context of non-rigid reconstruction, both linear [15, 18, 21, 39, 51, 103, 118] and non-linear [30, 31, 74, 85] latent variable models have been employed. However, since existing models are unable to make use of the known physical properties of the surface, they produce shapes that violate important constraints, and therefore look unnatural. An alternative approach is to directly encode the physical properties of the system. Physics-based approaches have made use both of the Finite Element Method [13, 54, 57, 63, 68, 105] and of more intuitive constraints, such as inextensibility [23, 28, 71, 90]. Unfortunately, while physics-based approaches have the advantage of explicitly encoding prior knowledge, they involve solving high-dimensional optimization problems. Furthermore, since constraints such as inextensibility are only local, the resulting methods typically require the surface to be well-textured.

Attempts at coupling latent variable models and constraints for deformable shape recovery have been made [80]. However, these methods are limited to linear subspace models and to specific constraints. Furthermore, and more importantly, they incorporate the constraints on top of a latent variable model that still allows for invalid configurations. It would seem more effective to exploit the constraints while learning the model, thus yielding a latent variable model that only generates physically-plausible deformations. This, in essence, is what we propose in this chapter.

5.2 Learning a Constrained LVM

In this section, we present our approach to learning a latent variable model that incorporates constraints on the generated outputs. In particular, we focus on the problem of learning the mapping from a given latent space to the output space under equality and inequality constraints. Note that the latent space itself can be obtained with any available technique, such as PCA, or Isomap. Our mapping can thus be seen as a predictor from the latent space to the output space. We learn this mapping by minimizing a prediction error on labeled examples, for which the true output is known, while simultaneously enforcing constraints on unlabeled ones, for which the output is unknown. In the remainder of this section, we first derive the primal form of our learning problem. We then exploit duality to kernelize our approach, and thus be able to make use of the nonlinear kernels (e.g.,RBF) that have proven more effective than linear ones for many computer vision tasks.

5.2.1 Primal Optimization Problem

Let $\mathcal{X} \subseteq \mathbb{R}^m$ be a given latent space, and $\mathcal{Y} \subseteq \mathbb{R}^D$ be the output space of interest, such as the space of non-rigid 3D surfaces. Given a latent variable $\mathbf{x} \in \mathcal{X}$, our latent variable model can be encoded as a mapping of the form

$$\hat{\mathbf{y}} = \mathbf{W}\phi(\mathbf{x}) \ , \tag{5.1}$$

such that $\hat{\mathbf{y}} \in \mathcal{Y}$. $\mathbf{W} \in \mathbb{R}^{D\times d}$ is the parameter matrix that defines the mapping, and $\phi(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}^d$ is the feature map of the latent variable $\mathbf{x}$.

Let $\mathcal{L}$ be the set of labeled training examples containing N pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$ of latent variables $\mathbf{x}_i$ and associated continuous multi-dimensional labels (outputs) $\mathbf{y}_i$. Furthermore, let $\mathcal{U} = \mathcal{U}_E \cup \mathcal{U}_I$ be the set of unlabeled training examples $\bar{\mathbf{x}}_j$ subject to equality ($\mathcal{U}_E$) and inequality ($\mathcal{U}_I$) constraints. We formulate learning as a constrained optimization problem, where a loss function l is minimized on the labeled training set $\mathcal{L}$ subject to constraints on the unlabeled set $\mathcal{U}$. This can be written as

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{\{\mathbf{x}_i,\mathbf{y}_i\}\in\mathcal{L}} l(\mathbf{W}, \mathbf{x}_i, \mathbf{y}_i) + \gamma\mathcal{R}(\mathbf{W}) \\ \text{s. t.} \quad & C(\mathbf{W}, \bar{\mathbf{x}}_u) = 0 \qquad \forall \bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D(\mathbf{W}, \bar{\mathbf{x}}_v) \leq 0 \qquad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned} \tag{5.2}$$

where $\mathcal{R}$ is the regularizer on $\mathbf{W}$ with weight γ, and $C(\mathbf{W}, \bar{\mathbf{x}}_u)$, resp. $D(\mathbf{W}, \bar{\mathbf{x}}_v)$, is a vector function encoding all N_E equality constraints, resp. N_I inequality constraints, defined with respect to the prediction of the unlabeled data $\bar{\mathbf{x}}_u$, resp. $\bar{\mathbf{x}}_v$.

The problem of Eq. 5.2 is very general, and different loss functions, regularizers and constraints can be utilized. Here, we consider the case of the square loss, Frobenius norm regularizer, and arbitrary nonlinear constraints on the predictions. The optimization problem therefore becomes

$$\begin{aligned} \min_{\mathbf{W}} \quad & \frac{1}{2}\|\mathbf{W}\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 \\ \text{s. t.} \quad & C\left(\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right) = 0 \quad \forall \bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D\left(\mathbf{W}\phi(\bar{\mathbf{x}}_v)\right) \leq 0 \quad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned} \tag{5.3}$$

where $\phi(\mathbf{x}) = [\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_N)]$, and $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_N]$ is the matrix of labeled training outputs.

If the constraints are non-convex, so is the optimization problem in Eq. 5.3. We therefore transform it so that we can solve it as a sequence of closed-form updates. First, we rewrite the inequality constraints as equalities by introducing slack variables $\boldsymbol{\epsilon}$. This yields the optimization problem

$$\begin{aligned} \min_{\mathbf{W},\boldsymbol{\epsilon}} \quad & \frac{1}{2}\|\mathbf{W}\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon}\|_2^2 \\ \text{s. t.} \quad & C\left(\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right) = 0 \qquad\qquad \forall \bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D\left(\mathbf{W}\phi(\bar{\mathbf{x}}_v)\right) + \boldsymbol{\epsilon}^{(v)} \odot \boldsymbol{\epsilon}^{(v)} = 0 \quad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned} \tag{5.4}$$

where $\odot$ is the Hadamard (elementwise) product, $\boldsymbol{\epsilon}^{(v)}$ contains the slack variables associated with example v, and $\frac{\alpha}{2}\|\boldsymbol{\epsilon}\|_2^2$ encodes potential additional knowledge about the problem. This, for instance, is useful in conjunction with the inequality constraints of [80], where only small deviations from equalities are expected.

As a second step, we perform a first order Taylor expansion of the constraints. Given an initial solution for the parameters $\mathbf{W}$ and $\boldsymbol{\epsilon}$, we iteratively linearize the constraints around the current solution, and update the parameters by solving the linearized problem. At each iteration t of this procedure, the linearized problem can be written as

$$\begin{aligned} \min_{\delta\mathbf{W},\delta\boldsymbol{\epsilon}} \quad & \frac{1}{2}\|(\mathbf{W}_t + \delta\mathbf{W})\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 \\ & + \frac{\gamma}{2}\|\mathbf{W}_t + \delta\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 \\ \text{s. t.} \quad & C_t^{(u)} + \mathbf{G}_u\delta\mathbf{W}\phi(\bar{\mathbf{x}}_u) = 0 \qquad \forall \bar{\mathbf{x}}_u \in \mathcal{U}_E \ , \\ & D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v\delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) \\ & \quad + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)} = 0 \qquad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned} \tag{5.5}$$

where $\mathbf{W}_t$ and $\boldsymbol{\epsilon}_t$ are the current estimates of $\mathbf{W}$ and $\boldsymbol{\epsilon}$, respectively. $C_t^{(u)}$ is the value of the equality constraints for unlabeled example u at the current prediction $\hat{\mathbf{y}}_{u,t}$, and $\mathbf{G}_u$ is the $N_E \times D$ matrix containing the gradient of these constraints with respect to $\hat{\mathbf{y}}_{u,t}$. Similarly, $D_t^{(v)}$ and $\mathbf{Q}_v$ encode the value and gradient of the inequality constraints for unlabeled example v at the current prediction $\hat{\mathbf{y}}_{v,t}$. The solution to the problem in Eq. 5.6 can be obtained in closed-form by solving a linear system in $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$.

5.2.2 Kernel-based Mappings

The primal formulation of our latent variable model only allows for linear mappings from the feature map of the latent space to the output space. While some degree of non-linearity can be encoded in the feature map, it results in the rapid growth of the number of parameters to optimize. This makes our primal formulation computationally expensive and more prone to overfitting. Furthermore, for many kernels (e.g., RBF), the feature maps cannot be explicitly computed.

We therefore need to kernelize our approach to take advantage of such kernels. To this end, we exploit duality. We start by first writing the Lagrangian of the minimization problem in Eq. 5.6, and then make use of the Karush-Kuhn-Tucker (KKT) conditions to derive a solution for the Lagrange multipliers. This yields an optimization method similar to the one in Section 5.2.1, where we iteratively linearize the constraints around the current prediction of the unlabeled data, solve for the Lagrange multipliers of the dual linearized problem, and update the prediction. Importantly, we show that the Lagrange multipliers can be obtained in closed-form, thus yielding a sequence of closed-form updates similar to the one in the primal formulation.

More specifically, the Lagrangian of the minimization problem in Eq. 5.6 can be expressed as

$$\begin{aligned} L = & \frac{1}{2}\|(\mathbf{W}_t + \delta\mathbf{W})\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}_t + \delta\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 \\ & + \sum_u \left[C_t^{(u)} + \mathbf{G}_u \delta\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right]^T \boldsymbol{\lambda}_u^E \\ & + \sum_v \left[D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v \delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)}\right]^T \boldsymbol{\lambda}_v^I \, , \end{aligned}$$

where $\boldsymbol{\lambda}_u^E \in \mathbb{R}^{N_E}$ and $\boldsymbol{\lambda}_v^I \in \mathbb{R}^{N_I}$ are the Lagrange multipliers associated with the equality and inequality constraints for unlabeled examples u and v, respectively.

To find an optimal solution to our problem, we first make use of the KKT stationarity condition, which, in our case, states that the solution for $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$ must satisfy $\frac{\partial L}{\partial \delta\mathbf{W}} = 0$ and $\frac{\partial L}{\partial \delta\boldsymbol{\epsilon}} = 0$, respectively.

Claim 1 *Solving the KKT stationarity conditions yields*

$$\begin{aligned} \delta\mathbf{W} &= \mathbf{A}\mathbf{Z} - \mathbf{W}_t, \\ \delta\boldsymbol{\epsilon}^{(v)} &= -\left(\frac{1}{\alpha}\boldsymbol{\lambda}_v^I + \mathbf{1}\right) \odot \boldsymbol{\epsilon}_t^{(v)} \, , \quad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \, , \end{aligned} \tag{5.6}$$

respectively, where

$$\mathbf{A} = \left[\mathbf{M} - \sum_u \mathbf{G}_u^T \lambda e v_u \mathbf{K}_{u,:} - \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \mathbf{K}_{v,:}\right] \mathbf{B}^{-1}, \tag{5.7}$$

$$\mathbf{Z} = \left[\begin{array}{ccccccccc} \phi(\mathbf{x}) & \cdots & \phi(\bar{\mathbf{x}}_{u'}) & \cdots & \phi(\bar{\mathbf{x}}_s) & \cdots & \phi(\bar{\mathbf{x}}_{v'}) & \cdots \end{array}\right]^T,$$

$$\mathbf{B} = \mathbf{K}_{:,\mathcal{L}} \mathbf{K}_{\mathcal{L},:} + \gamma \mathbf{K}_{:,:} \ ,$$

$$\mathbf{M} = \mathbf{Y} \mathbf{K}_{\mathcal{L},:} \ ,$$

with u', s and v' the indices of the unlabeled data in $\mathcal{U}_E \backslash \mathcal{U}_I$, $\mathcal{U}_E \cap \mathcal{U}_I$, and $\mathcal{U}_I \backslash \mathcal{U}_E$, respectively.

The kernel $\mathbf{K}_{:,:}$ is defined as

$$\mathbf{K}_{:,:} = \mathbf{Z}\mathbf{Z}^T = \left[\begin{array}{c|c} \mathbf{K}_{\mathcal{L},\mathcal{L}} & \mathbf{K}_{\mathcal{L},\mathcal{U}} \\ \hline \mathbf{K}_{\mathcal{U},\mathcal{L}} & \mathbf{K}_{\mathcal{U},\mathcal{U}} \end{array}\right] , \tag{5.8}$$

and can be computed via any kernel function, e.g., RBF.

Proof: In Appendix A. □

The KKT stationarity conditions define a solution for our variables $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$ in terms of the Lagrange multipliers $\boldsymbol{\lambda}_u^E$ and $\boldsymbol{\lambda}_v^I$. To find a solution for these Lagrange multipliers, we make use of the KKT primal feasibility condition, which states that the constraints should be satisfied at the optimal value of the parameters.

Claim 2 *The solution to the constraints encoded by the KKT primal feasibility condition takes the form $\boldsymbol{\lambda} = \mathbf{S}^{-1}\mathbf{r}$, where*

$$\boldsymbol{\lambda} = \left[\begin{array}{c} \boldsymbol{\lambda}^E \\ \boldsymbol{\lambda}^I \end{array}\right] , \quad \mathbf{S} = \left[\begin{array}{c|c} \mathbf{S}^{E,E} & \mathbf{S}^{E,I} \\ \hline \mathbf{S}^{I,E} & \mathbf{S}^{I,I} \end{array}\right] , \quad \mathbf{r} = \left[\begin{array}{c} \mathbf{r}^E \\ \mathbf{r}^I \end{array}\right] ,$$

and

$$\begin{aligned}
\mathbf{S}_{u,a}^{E,E} &= \mathbf{G}_u \mathbf{G}_a^T (\mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,u}) \ , \\
\mathbf{S}_{u,b}^{E,I} &= \mathbf{G}_u \mathbf{Q}_b^T (\mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,u}), \\
\mathbf{S}_{v,a}^{I,E} &= \mathbf{Q}_v \mathbf{G}_a^T (\mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,v}), \\
\mathbf{S}_{v,b}^{I,I} &= \mathbf{Q}_v \mathbf{Q}_b^T (\mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,v}) + \delta_{v,b} \, diag\left(\frac{1}{\alpha} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)}\right) , \\
\mathbf{r}_u^E &= \mathbf{G}_u \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,u} - \mathbf{G}_u \hat{\mathbf{y}}_{u,t} + C_t^{(u)}, \\
\mathbf{r}_v^I &= \mathbf{Q}_v \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,v} - \mathbf{Q}_v \hat{\mathbf{y}}_{v,t} + D_t^{(v)} - \frac{1}{2} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)},
\end{aligned}$$

with $\delta_{v,b}$ *the Kronecker delta.*

Proof: In Appendix A. □

In the two claims above, we have shown how to obtain in closed-form the Lagrange multipliers that give the optimal solution to the problem in Eq. 5.6. Note that this requires having a prediction for the unlabeled examples $\hat{\mathbf{y}}_{v,t}$ at the current iteration t. Furthermore, at inference, to make use of our latent variable model, we need to be able to compute the prediction for a new input. To address these points, we now define the form of the prediction in our model.

Claim 3 *Prediction for any input* $\mathbf{x}_*$ *in our kernelized model can be done in closed-form, and can be written as*

$$\hat{\mathbf{y}}_* = \mathbf{A}\mathbf{K}_{:,*}\,, \tag{5.9}$$

where $\mathbf{K}_{:,*} = \mathbf{Z}\phi(\mathbf{x}_*) = \left[\begin{array}{c|c} \mathbf{K}_{*,\mathcal{L}} & \mathbf{K}_{*,\mathcal{U}} \end{array}\right]^T.$

Proof: In Appendix A. □

Since we follow the same linearization strategy as in Section 5.2.1, learning still consists of a succession of updates based on the current prediction for the unlabeled inputs. Therefore, we can derive an algorithm that iteratively linearizes the constraints around the current prediction, solves for the Lagrange multipliers and refines the prediction. This scheme is summarized in Algorithm 1. Note that each step can be done in closed-form. Note also that, even though Claim 1 defines the update $\delta\mathbf{W}$ in terms of the Lagrange multipliers, $\mathbf{W}$ is never explicitly computed in our algorithm, thus making it fully kernelized.

Algorithm 1 Learning a constrained latent variable model

```
Initialize $\hat{\mathbf{y}}_{u,0}$ and $\hat{\mathbf{y}}_{v,0}$ using an unconstrained predictor
Initialize $\boldsymbol{\epsilon}_0$ to non-zero values
for $t = 1$ to #iters do
    Compute $\mathbf{G}_u$, $C_t^{(u)}$, $\mathbf{Q}_v$, $D_t^{(v)}$ from $\hat{\mathbf{y}}_{u,t-1}$, $\hat{\mathbf{y}}_{v,t-1}$
    Compute $\mathbf{S}$ from Claim 2
    Compute $\mathbf{r}$ from Claim 2
    Compute $\boldsymbol{\lambda}_u^E$ and $\boldsymbol{\lambda}_v^I$ using $\boldsymbol{\lambda} = \mathbf{S}^{-1}\mathbf{r}$
    Compute $\mathbf{A}$ using Eq. 5.8
    Compute $\hat{\mathbf{y}}_{u,t}$ and $\hat{\mathbf{y}}_{v,t}$ using Eq. 5.9
    Compute $\boldsymbol{\epsilon}_t$ using Eq. 5.6
end for
```

5.3 Experimental Evaluation

We demonstrate the effectiveness of our constrained latent variable model at reconstructing deformable surfaces from monocular images. We compare our results to those obtained using a linear subspace model and an unconstrained version of our non-linear model, which corresponds to a GPLVM. In all cases, reconstruction is obtained by optimizing the latent variable so as to minimize an image-based loss function. Errors are given in terms of both the 3D reconstruction errors and the constraint violation. Reconstruction errors are computed as the average point-to-point distance between ground-truth and reconstructed shapes. Constraint violation is taken to be the mean value of $C(\mathbf{x}_*)$ for equalities and the mean value of $D(\mathbf{x}_*)$ for all violated inequality constraints. All the quantitative results are expressed in millimeters.

In the remainder of this section, we first describe our learning setup and the different types of constraints used in our experiments. We then present our results on synthetic data and real images of surfaces made of different materials. Our results include a quantitative evaluation of reconstructions obtained from real images captured with a Microsoft Kinect, whose output depth we treat as ground-truth.

5.3.1 Learning Setup

To learn the models, we make use of two publicly available datasets obtained with a motion capture system[1]. The first one consists of the 3D locations of markers placed as a $9{\times}9$ grid on a piece of cardboard, thus forming a square mesh of size 160×160mm with 208 edges. The second one consists of similar measurements on a piece of cloth, represented by a $9{\times}7$ mesh of size 160×120mm with 158 edges. The cardboard dataset exhibits simpler deformations than the cloth one. To obtain latent spaces for each dataset independently, we performed PCA on the 3D marker locations. We used 12 and 30 latent variables for the cardboard and cloth datasets, respectively, which covers more than 95% of the variance of the data. In all the experiments, we used an RBF kernel for our model and its unconstrained version. We set both regularization weights γ and α to 0.001.

We investigated the use of length constraints as both equalities and inequalities. Under the former, the length of the edges connecting the mesh vertices must remain constant. The latter allow these lengths to decrease to model the fact that two vertices may come closer to each other if folds appear between them, but cannot be further apart than the geodesic distance along

---

[1] Publicly available at http://cvlab.epfl.ch/data/dsr/

			Reconstruction error [mm]			
			$V = 20$	$V = 60$	$V = 100$	$V = 150$
Cardboard	Equality	PCA	31.6±4.45	31.6±4.45	31.6±4.45	31.6±4.45
		Unconstrained	4.64±0.29	4.64±0.29	4.64±0.29	4.64±0.29
		Ours	**4.48±0.31**	**4.28±0.33**	4.30±0.36	4.27±0.39
	Ineq.	PCA	31.6±4.45	31.6±4.45	31.6±4.45	31.6±4.45
		Unconstrained	4.64±0.29	4.64±0.29	4.64±0.29	4.64±0.29
		Ours	4.52±0.31	4.34±0.30	**4.25± 0.28**	**4.12±0.27**
Cloth	Equality	PCA	16.2±2.19	16.2±2.19	16.2±2.19	16.2±2.19
		Unconstrained	4.36±0.20	4.36±0.20	4.36±0.20	4.36±0.20
		Ours	4.30±0.19	4.29±0.12	4.27±0.21	4.44±0.17
	Ineq.	PCA	16.2±2.19	16.2±2.19	16.2±2.19	16.2±2.19
		Unconstrained	4.36±0.20	4.36±0.20	4.36±0.20	4.36±0.20
		Ours	**4.25±0.17**	**4.10±0.14**	**3.99± 0.17**	**3.95±0.16**

Table 5.1: Predicting shapes from known latent variables. Reconstruction error as a function of the number of unlabeled examples V for a fixed $N = 50$. Note that the constraint violation measure is different for inequalities and for equalities.

			Constraint violation [mm]			
			$V = 20$	$V = 60$	$V = 100$	$V = 150$
Cardboard	Equality	PCA	3.19±0.64	3.19±0.64	3.19±0.64	3.19±0.64
		Unconstrained	0.98±0.05	0.98±0.05	0.98±0.05	0.98±0.05
		Ours	**0.81±0.02**	**0.63± 0.01**	**0.50±0.02**	**0.41±0.02**
	Ineq.	PCA	3.43±0.76	3.43±0.76	3.43±0.76	3.43±0.76
		Unconstrained	0.94±0.04	0.94±0.04	0.94±0.04	0.94±0.04
		Ours	**0.77±0.03**	**0.56±0.02**	**0.43±0.02**	**0.34±0.02**
Cloth	Equality	PCA	3.05±0.26	3.05±0.26	3.05±0.26	3.05±0.26
		Unconstrained	1.33±0.04	1.33±0.04	1.33±0.04	1.33±0.04
		Ours	**1.20±0.04**	**1.03± 0.04**	**0.88±0.02**	**0.75±0.02**
	Ineq.	PCA	3.43±0.29	3.43±0.29	3.43±0.29	3.43±0.29
		Unconstrained	1.14±0.08	1.14±0.08	1.14±0.08	1.14±0.08
		Ours	**1.02±0.09**	**0.87±0.06**	**0.73±0.03**	**0.63±0.02**

Table 5.2: Predicting shapes from known latent variables. Constraint violation as a function of the number of unlabeled examples V for a fixed $N = 50$. Note that the constraint violation measure is different for inequalities and for equalities.

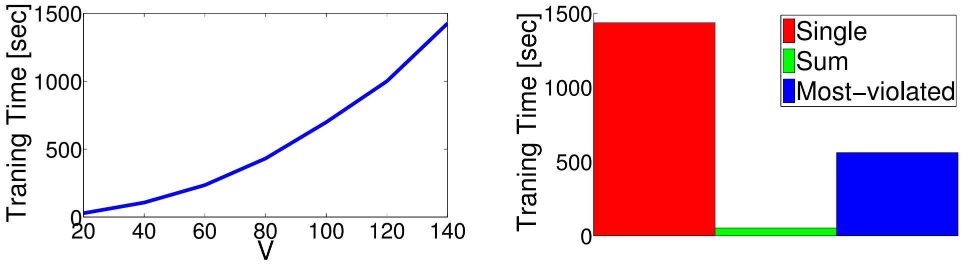
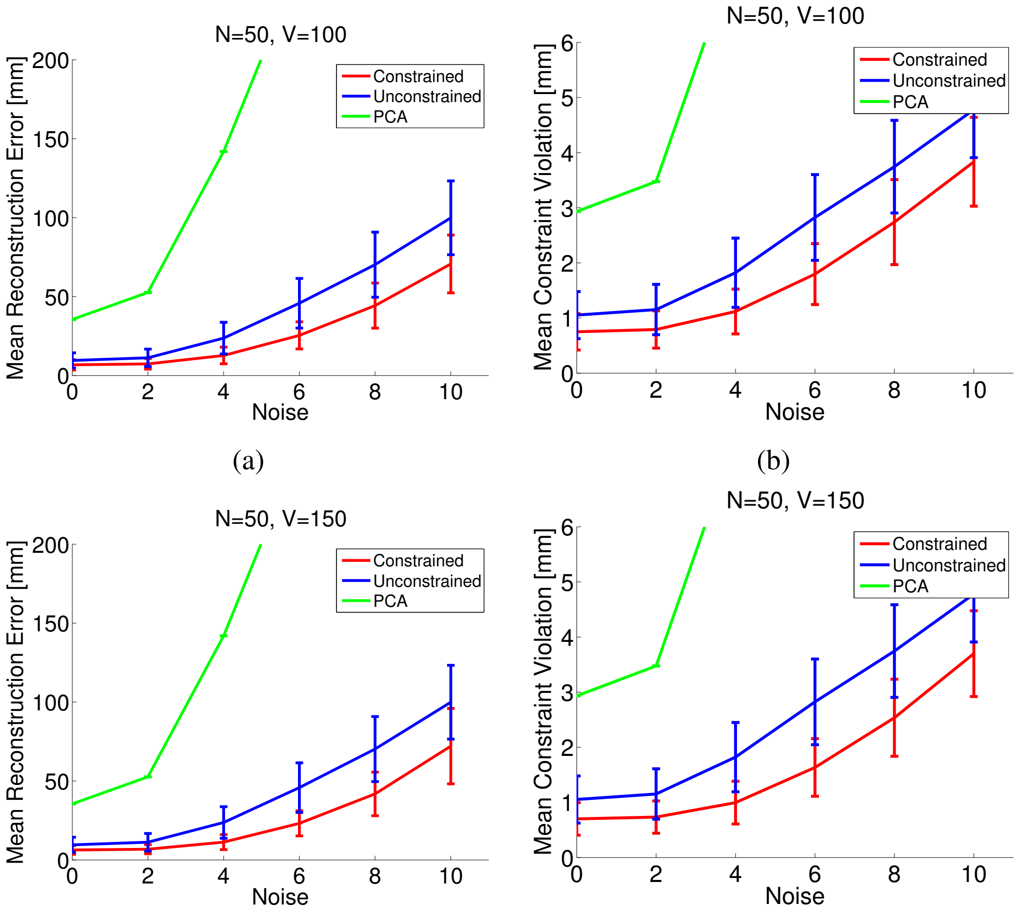


Figure 5.2: Training times. (Left) Training time as a function of the number of validation samples when using all constraints. (Right) Training times for different constraint selection strategies for a fixed number of validation samples $V = 140$.

the surface. The equalities have been shown to be appropriate for smoothly deforming surfaces, and the inequalities for surfaces undergoing more complex deformations [80]. To confirm this, we predicted shapes given the true latent variables of 500 test examples. Tables 5.1 and 5.2 depicts reconstruction and constraint errors averaged over the test samples as a function of the number of unlabeled examples V. Note that the constraint violation measure is different for inequalities and equalities. For the cardboard dataset, both constraint types perform well. For the cloth dataset where sharper folds occur, inequality constraints are more appropriate; increasing V improves both reconstruction and constraint satisfaction for inequalities, whereas it only improves constraint satisfaction for equalities. Note that our predictions are more accurate than those of the baselines.

Our implementation can handle up to 60K constraints on a standard PC. However, this still limits us in the number of unlabeled examples that we can use. To increase this number, we implemented a different strategy to encode the constraints, which involves summing over individual ones. This yields new constraints of the form $\tilde{C}(\mathbf{x}) = \sum_j C_j(\mathbf{x}) = 0$, and reduces the number of constraints for each unlabeled sample, which lets us use more of them. In practice, we define these sums of constraints as the sums of all individual vertical or horizontal constraints on the rectangular grid, which amounts to preserving the length of a complete horizontal or vertical line as opposed to that of individual edges.

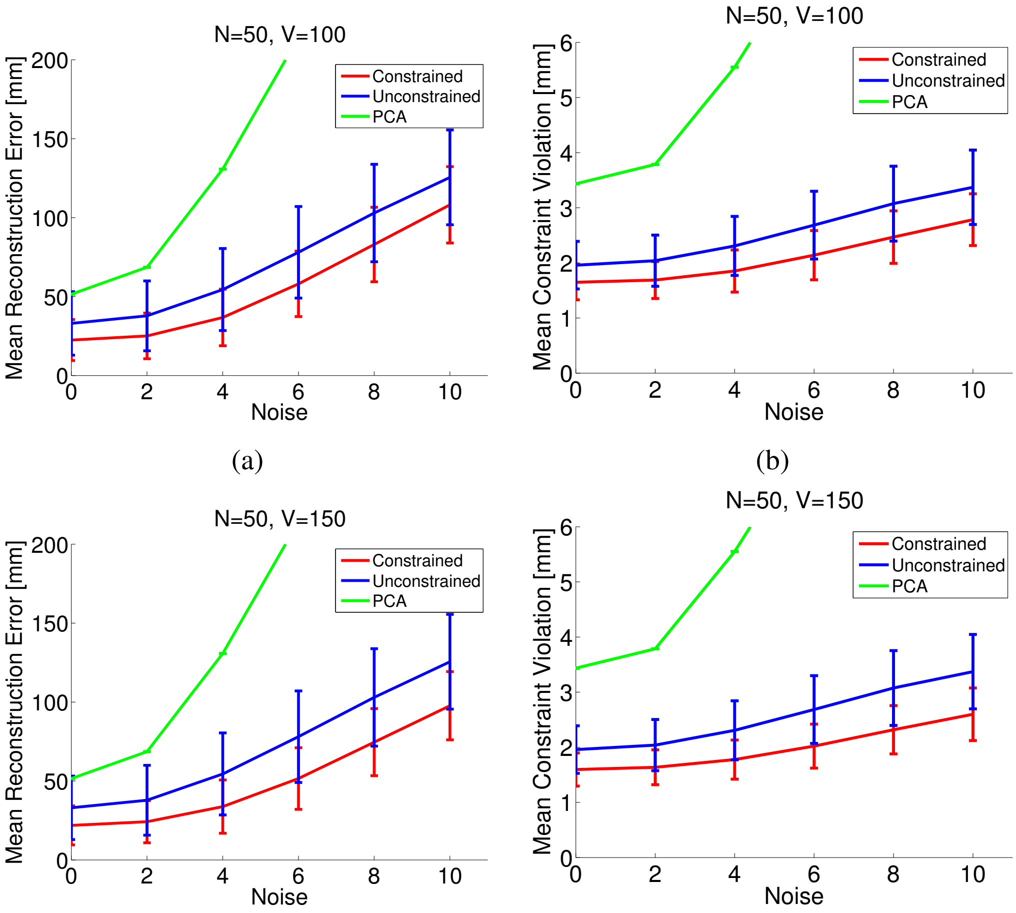
The constraints we use are sparse in nature, since each one only depends on two mesh vertices. Thus, the linear system to obtain the Lagrange multipliers is sparse as well, which allows for the use of efficient sparse solvers. Figure 5.2(left) depicts training time as a function of the number of unlabeled examples for the cardboard dataset. To improve efficiency, we can



(a) (b)

(c) (d)

Figure 5.3: Reconstructing a piece of cardboard from synthetic data. (a,b) Reconstruction error and constraint violation as a function of image noise for $N = 50$ and $V = 100$. (c,d) Similar errors for $N = 50$ and $V = 150$. Our model was trained using equality constraints.



(a) (b)

(c) (d)

Figure 5.4: Reconstructing a piece of cloth from synthetic data. (a,b) Reconstruction error and constraint violation as a function of image noise for $N = 50$ and $V = 100$. (c,d) Similar errors for $N = 50$ and $V = 150$. Our model was trained using inequality constraints.

also rely on different strategies to account for constraints. Summing constraints, as described above, is one such strategy. Another one consists in adding the most violated constraints to a set of active constraints at each learning iteration. Training times for these different strategies are given in Figure 5.2(right). Note that summing constraints decreases the training time dramatically since it effectively reduces the number of constraints for the same number of unlabeled samples. Iteratively adding constraints to an active set also noticeably reduces training time. However, since the final iterations exploits all the constraints, this strategy does not allow us to use more samples.

5.3.2 Synthetic Data

We first used the two motion capture datasets to generate synthetic data. To this end, we sampled the barycentric coordinates of the ground-truth meshes and projected the resulting 3D points with a known camera, thus creating 2D image measurements. We then added Gaussian noise with standard deviation ranging between 0 and 10 to these measurements. At test time, we optimized the latent variables, as well as the global rotation and translation, so that the predicted 3D shape minimizes the reprojection error with respect to the noisy image measurements. For both datasets, we learned the models with $N = 50$ labeled examples and either $V = 100$ or $V = 150$ unlabeled ones, and tested them on 300 samples. For each noise value, we used 5 different train/test partitions. Figs. 5.3 and 5.4 depict errors as a function of the image noise standard deviation for the cardboard and cloth datasets, respectively. Note that our constrained model consistently outperforms PCA and the unconstrained model for both reconstruction error and constraint violation. Error bars on the plots represent ± 1 standard deviation over the 5 different partitions. Note that the PCA model was learned from all the data, and therefore does not depend on the partition. This remains a valid comparison, since only the baseline has access to more data. Figure 5.5 depicts similar errors for our different constraint selection strategies. Note that, while being faster to train, summing constraints yields less accurate reconstructions for a given V.

5.3.3 Real Images with Ground-truth

To evaluate our model's accuracy in realistic conditions, we performed experiments where the images were captured with a Microsoft Kinect, which also provides us with ground-truth 3D

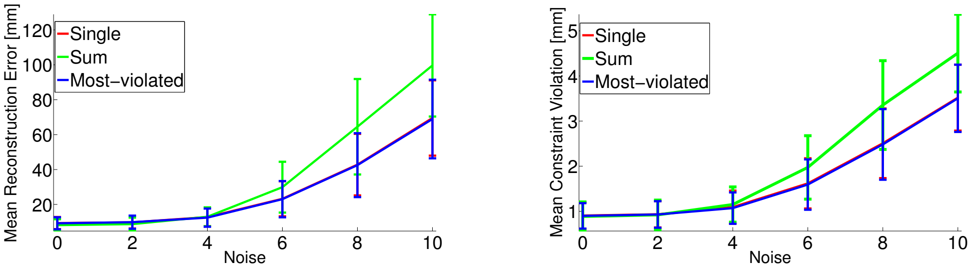


Figure 5.5: Reconstructing a piece of cardboard using different constraint selection strategies. Reconstruction error and constraint violation as a function of noise for $N = 100$ and $V = 100$. The curves for the single and most-violated strategies overlap.

	Equality		Inequality	
	Reconstr. Error [mm]	Constraint Viol. [mm]	Reconstr. Error [mm]	Constraint Viol. [mm]
PCA	11.68 ± 0.00	1.71 ± 0.00	11.68 ± 0.00	2.09 ± 0.00
Unconstrained	9.35 ± 1.03	1.10 ± 0.23	9.35 ± 1.03	0.96 ± 0.09
Ours	**7.23 ± 0.76**	**0.78 ± 0.03**	**8.03 ± 0.56**	**0.71 ± 0.13**

Table 5.3: Reconstructing a piece of paper. Reconstruction error was computed with respect to the Kinect ground-truth.

information, only used for evaluation purposes. We captured deformations of two different materials: a piece of paper and a t-shirt. As before, we used reprojection error as image loss, but used SIFT to compute the correspondences between a reference image in which the 3D shape is known and the other images of the sequence. For both materials, we learned the models with $N = 20$ labeled and $V = 50$ unlabeled examples from the cardboard dataset, and, as before, used 5 different training sets. Figure 5.6 depicts images of both sequences with our reconstructions. Tables 5.3 and 5.4 show errors averaged over the frames of the sequence when using either equality, or inequality constraints. Here, reconstruction error was computed between the predicted 3D locations of the feature points and their ground-truth Kinect locations. As with synthetic data, our model outperforms the baselines for both constraint types.

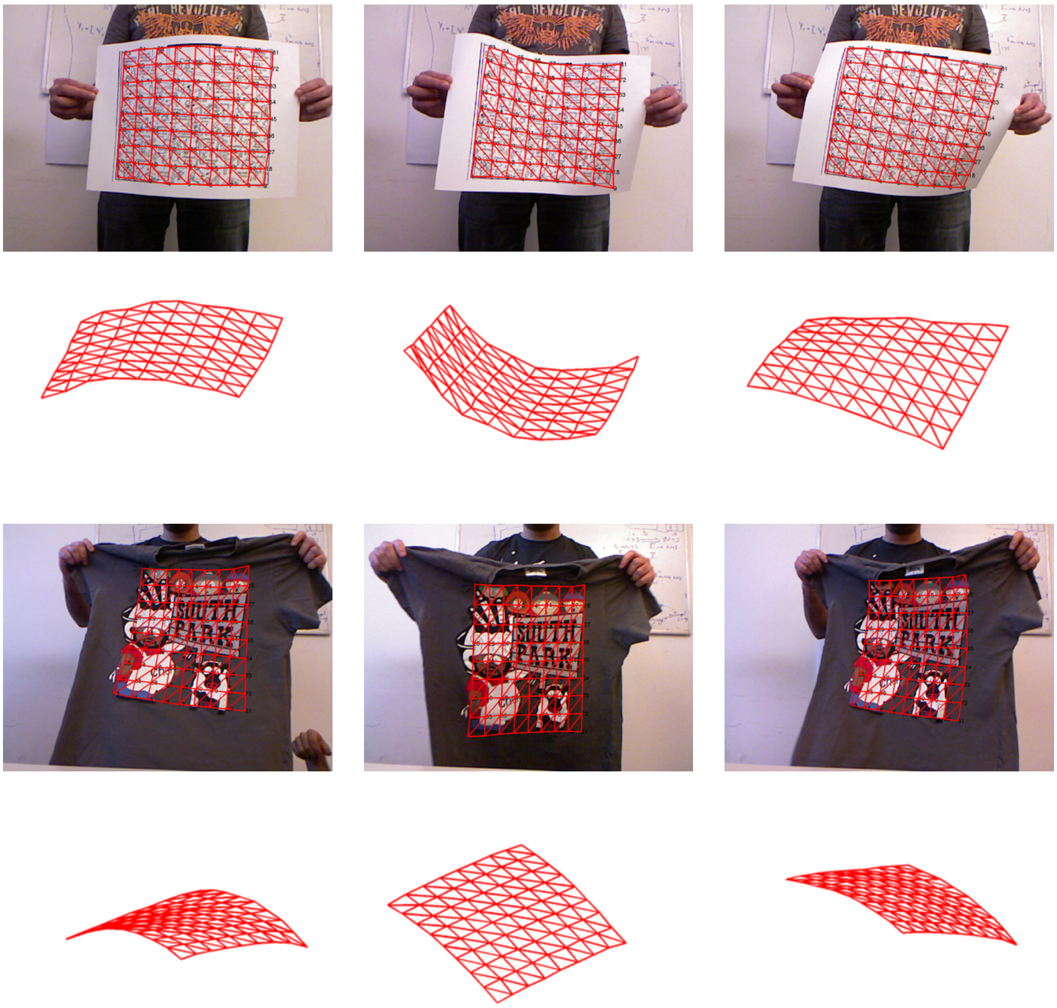


Figure 5.6: Real images with ground-truth. (Top two rows) Images of a deforming piece of paper with reconstructed meshes seen from a different viewpoint. (Bottom two rows) Similar images for a deforming t-shirt. In both cases, we show our reconstructions reprojected on the images.

	Equality		Inequality	
	Reconstr. Error [mm]	Constraint Viol. [mm]	Reconstr. Error [mm]	Constraint Viol. [mm]
PCA	18.44±0.00	0.97±0.00	18.44±0.00	0.84±0.00
Unconstrained	15.50±1.78	0.92±0.22	15.50±1.78	0.75±0.14
Ours	**14.79 ± 0.84**	**0.73 ± 0.05**	**14.35±0.90**	**0.60±0.07**

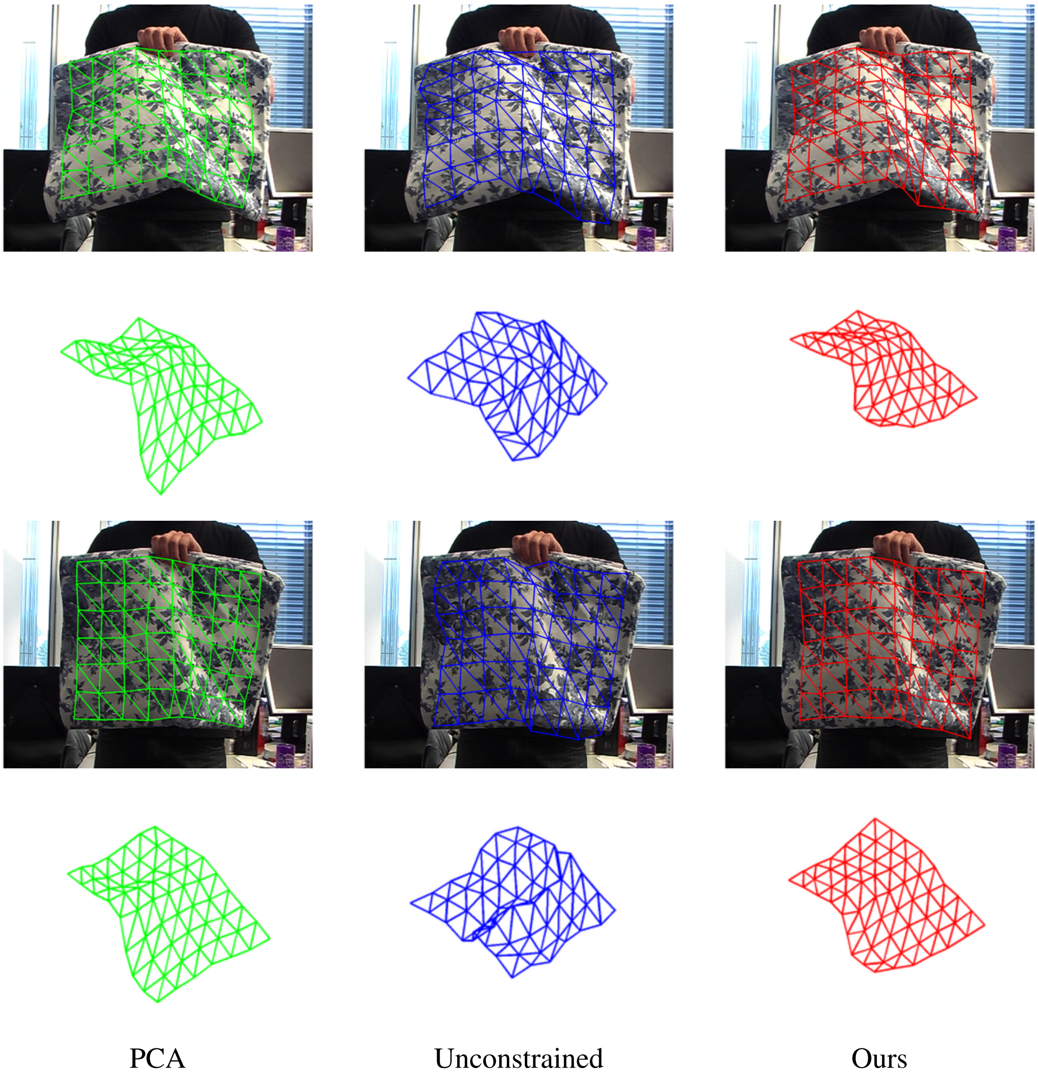
Table 5.4: Reconstructing a deforming t-shirt. Reconstruction error was computed with respect to the Kinect ground-truth.

5.3.4 Real Images without Ground-truth

For qualitative evaluation, we also applied our model to reconstructing two sequences of deforming cloth surfaces. In both cases, we used $N = 500$ labeled examples. Since the deformations in the first sequence are relatively simple, we could use a small number of unlabeled examples (V=150), and thus exploit all individual edge equality constraints when learning our model. For the second sequence, which contains more complex deformations, we used sums of constraints which let us employ more unlabeled examples (V=1500). In both experiments, the image-based loss was taken as the normalized cross-correlation between the texture under the optimized mesh and the texture in a reference image. In Figure 5.7, we compare our results with those obtained with the baselines for the first sequence. Note that the shapes reconstructed with our model better correspond to the ones in the images. Figure 5.8 depicts our reconstructions on the second sequence. However, we encourage the reader to look at the full comparison in the videos given as supplementary material. Since we have no ground-truth for these sequences, we can only evaluate constraint violation. Figure 5.9 depicts this error for all the frames in the sequences. Note that our method clearly outperforms the baselines in terms of constraint satisfaction.

5.4 Conclusion

In this chapter, we have introduced a constrained latent variable model that encodes prior knowledge about the desired output in the form of equality and inequality constraints. We have shown that our approach can be kernelized, thus allowing for the use of non-linear kernels that have proven effective in many computer vision tasks. Using both synthetic and real data,



PCA Unconstrained Ours

Figure 5.7: Reconstructing a deforming bed sheet. Comparison of our reconstructions with those of two baselines for two frames of the sequence. In each frame, we show the reconstructed mesh reprojected on the original image, as well as a side view of the mesh.

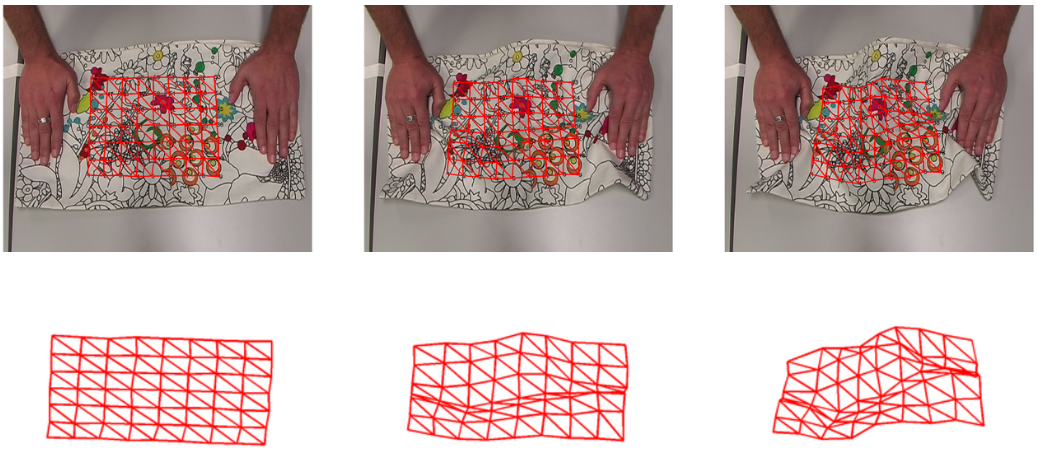
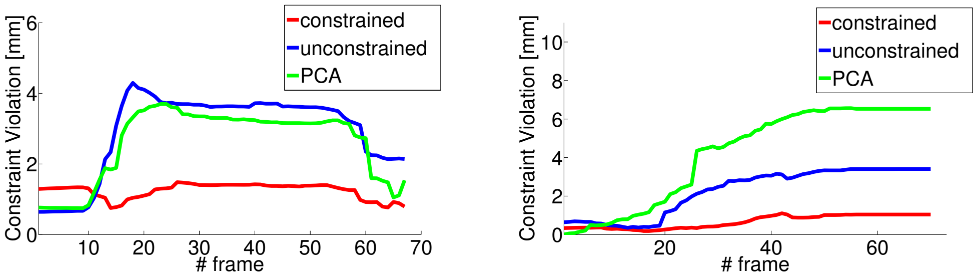


Figure 5.8: Reconstructing a deforming cushion cover. Reconstructions obtained with our model for 3 frames reprojected on the original image, and seen from a different viewpoint.



(a) (b)

Figure 5.9: Constraint violation. Comparison of our model with two baselines for (a) the bed sheet and (b) the cushion sequences.

we have demonstrated that our model outperforms the commonly-employed ones for the purpose of monocular 3D surface reconstruction, which is such an ambiguous problem that using constraints effectively is a requirement for success. Furthermore, our formalism is extremely general.

CHAPTER

SIX

CONLUDING REMARKS

In this thesis we have presented various approaches for monocular 3D deformable surface reconstruction problem. It is known to be an ambiguous problem when using feature points or shading patterns observed on an input image. Existence of multiple 3D shapes that give visually identical projections on the input image makes the problem challenging. In addition to the inherent ambiguities associated with solving an inverse problem, image noise makes the task even more difficult. Throughout the thesis, we have explored several means of resolving these ambiguities in the surface reconstruction process by simultaneously exploiting various kinds of image information and using priors and constraints.

In Chapter 3, we have presented an approach to deformable surface 3D reconstruction that overcomes most limitations of state-of-the-art techniques. We have shown that we could recover the shape of a non-rigid surface while requiring neither points to be tracked throughout a whole video sequence nor a reference image in which the surface shape is known. To this end, we only need a pair of images displaying the surface in two different configurations and enough texture to establish correspondences. We believe this to be both a minimal setup for which a correspondence-based 3D shape recovery technique could possibly work and a practical one for real-world applications.

While texture-based methods, such as the one proposed in Chapter 3, have proven effective in surface reconstruction tasks, they are ill-equipped to handle partially-textured surfaces. As our second contribution, in Chapter 4, we proposed a novel approach to recovering the 3D shape of a deformable surface from a monocular input by taking advantage of shading information in more generic contexts than conventional SfS methods. This includes surfaces that may be fully or partially textured and lit by arbitrarily many distant light sources. To this end, given a lighting model, we learn the relationship between a shading pattern and the corresponding local surface shape. At run time, we first use this knowledge to recover the shape of surface patches and then enforce spatial consistency between the patches to produce a global 3D shape. We have demonstrated the superior accuracy of our approach compared to state-of-the-art techniques on both synthetic and real data.

Finally, in Chapter 5 we introduced a constrained latent variable model whose generated output inherently accounts for geometric constraints such as inextensibility defined on the mesh model. To this end, we learn a non-linear mapping from the latent space to the output space, which corresponds to vertex positions of a mesh model, such that the generated outputs comply with equality and inequality constraints expressed in terms of the problem variables. Since its output is encouraged to satisfy such constraints inherently, using our constrained model removes the need for computationally expensive methods that enforce them at run time. Using both synthetic and real data, we have demonstrated that our model outperforms the commonly-employed ones for the purpose of monocular 3D surface reconstruction, which is such an ambiguous problem that using constraints effectively is a requirement for success. Furthermore, our formalism is extremely general so that it could be used in many other vision tasks such as image classification and human pose recovery.

Taken together, the presented methods in this thesis headway towards dissolving the mentioned disambiguates that exist in deformable surface reconstruction from a single view. We believe that our methods represent a significant step towards making it of practical use. In the remainder of this chapter, we discuss their limitations and directions to improve them.

6.1 Future Work

There are various ways in which our proposed methods can be improved. We briefly mention the most interesting extensions below.

One way of extending our template-free reconstruction of Chapter 3 approach is to explore the use of multiple frames to handle self-occlusions. In our current implementation, points that are occluded in one of the two images cannot be reconstructed and we have to depend on surface fitting using a local deformation model to guess the shape around such points. However, since we can perform reconstruction from any two pairs of images, one can work on merging the results and filling the gaps without having to rely solely on interpolation. In other words, given an input frame, one can use any other frame in the sequence as the support frame and consider all the reconstructed point clouds simultaneously.

There is also space for improvement for the method we have presented in Chapter 4 that is useful for reconstructing partially textured surfaces while simultaneously exploiting textural and shading information. Even though our framework is general enough so that each component could be replaced with a more sophisticated one. For instance, representations of the lighting environment more sophisticated than spherical harmonics could be used to create our training set. Similarly, other potentially nonlinear parametrizations of the patch intensities and deformations could replace the current PCA mode weights. In addition, future work could focus on estimating the lighting parameters from the sequence. To this end, partial reconstructions for the textured regions of a surface would provide training data related with surface normals and corresponding intensity variations. This data would then be used to estimate the spherical harmonics coefficients describing the lighting environment.

The constrained latent variable model that we have presented in Chapter 5 can be potentially useful for other Computer Vision tasks where there are constraints on the output. In our future work, we will apply it on the human pose recovery problem where the human body is modeled with a set of connected inextensible line segments. In addition to this, we aim to integrate our method with the one presented in Chapter 4 to improve the reconstruction accuracy. To this end, we will apply our constrained learning framework to train a mapping from intensity patterns to 3D shapes.

6. CONLUDING REMARKS

APPENDIX

A

APPENDIX

To prove the claims in Chapter 5, we first rewrite the primal minimization problem that is solved at each iteration t of the constraints linearization. Recall that we have a set $\mathcal{L}$ of N labeled training pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$, and a set $\mathcal{U} = \mathcal{U}_E \cup \mathcal{U}_I$ of V unlabeled examples $\bar{\mathbf{x}}_u \in \mathcal{U}_E$ and $\bar{\mathbf{x}}_v \in \mathcal{U}_I$ on which we impose equality ($\mathcal{U}_E$) and inequality ($\mathcal{U}_I$) constraints. Our optimization problem can be written in terms of the parameter updates $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$ as

$$\begin{aligned} \min_{\delta\mathbf{W},\delta\boldsymbol{\epsilon}} \quad & \frac{1}{2}\|(\mathbf{W_t} + \delta\mathbf{W})\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W_t} + \delta\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 \\ \text{subject to} \quad & C_t^{(u)} + \mathbf{G}_u\delta\mathbf{W}\phi(\bar{\mathbf{x}}_u) = 0 \qquad \forall\bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v\delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)} = 0 \qquad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I\ , \end{aligned} \tag{A.1}$$

where $\phi(\mathbf{x}) = [\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_N)]$ contains the feature map of the labeled inputs, $\mathbf{W}_t$ and $\boldsymbol{\epsilon}_t$ are the estimates at iteration t of the parameters $\mathbf{W}$ and of the slack variables $\boldsymbol{\epsilon}$, respectively, and $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_N]$ is the matrix containing the training outputs. $C_t^{(u)}$ is the N_E-dimensional vector containing the value of the equality constraints for unlabeled example u at the current prediction $\hat{\mathbf{y}}_{u,t}$, and $\mathbf{G}_u$ is the $N_E \times D$ matrix encoding the gradient of these constraints with respect to $\hat{\mathbf{y}}_{u,t}$. Similarly, $D_t^{(v)}$ contains the value of the N_I inequality constraints for unlabeled example v at the current prediction $\hat{\mathbf{y}}_{v,t}$, and $\mathbf{Q}_v$ is the gradient of these constraints with respect to $\hat{\mathbf{y}}_{v,t}$.

The Lagrangian of this problem can be expressed as

$$\begin{aligned} L = {} & \frac{1}{2}\|(\mathbf{W_t} + \delta\mathbf{W})\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}_t + \delta\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 \\ & + \sum_u \left[C_t^{(u)} + \mathbf{G}_u\delta\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right]^T \boldsymbol{\lambda}_u^E + \sum_v \left[D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v\delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)}\right]^T \boldsymbol{\lambda}_v^I , \end{aligned} \tag{A.2}$$

where $\boldsymbol{\lambda}_u^E$ and $\boldsymbol{\lambda}_v^I$ are the Lagrange multipliers associated with the equality and inequality constraints for unlabeled examples u and v, respectively.

A solution to the original problem can be obtained by solving the Karush-Kuhn-Tucker (KKT) conditions, which, in our case, can be stated as

$$\frac{\partial L}{\partial \delta\mathbf{W}} = 0 , \tag{A.3}$$

$$\frac{\partial L}{\partial \delta\boldsymbol{\epsilon}} = 0 , \tag{A.4}$$

$$C_t^{(u)} + \mathbf{G}_u\delta\mathbf{W}\phi(\bar{\mathbf{x}}_u) = 0 , \quad \forall\bar{\mathbf{x}}_u \in \mathcal{U}_E, \tag{A.5}$$

$$D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v\delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)} = 0 , \quad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I . \tag{A.6}$$

Claim 1 *Solving the KKT stationarity conditions given in Eq.* (A.3) *and* (A.4) *yields*

$$\delta\mathbf{W} = \mathbf{A}\mathbf{Z} - \mathbf{W}_t , \tag{A.7}$$

$$\delta\boldsymbol{\epsilon}^{(v)} = -\left(\frac{1}{\alpha}\boldsymbol{\lambda}_v^I + \mathbf{1}\right) \odot \boldsymbol{\epsilon}_t^{(v)} , \quad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I , \tag{A.8}$$

respectively, where

$$\mathbf{A} = \left[\mathbf{M} - \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \mathbf{K}_{u,:} - \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \mathbf{K}_{v,:}\right] \mathbf{B}^{-1} , \tag{A.9}$$

$$\mathbf{Z} = \begin{bmatrix} \phi(\mathbf{x}) & \cdots & \phi(\bar{\mathbf{x}}_{u'}) & \cdots & \phi(\bar{\mathbf{x}}_s) & \cdots & \phi(\bar{\mathbf{x}}_{v'}) & \cdots \end{bmatrix}^T , \tag{A.10}$$

$$\mathbf{B} = \mathbf{K}_{:,\mathcal{L}}\mathbf{K}_{\mathcal{L},:} + \gamma\mathbf{K}_{:,:} ,$$

$$\mathbf{M} = \mathbf{Y}\mathbf{K}_{\mathcal{L},:} ,$$

with u', s and v' the indices of the unlabeled data in $\mathcal{U}_E \backslash \mathcal{U}_I$, $\mathcal{U}_E \cap \mathcal{U}_I$, and $\mathcal{U}_I \backslash \mathcal{U}_E$ respectively. The kernel $\mathbf{K}_{:,:}$ is defined as

$$\mathbf{K}_{:,:} = \mathbf{Z}\mathbf{Z}^T = \left[\begin{array}{c|c} \mathbf{K}_{\mathcal{L},\mathcal{L}} & \mathbf{K}_{\mathcal{L},\mathcal{U}} \\ \hline \mathbf{K}_{\mathcal{U},\mathcal{L}} & \mathbf{K}_{\mathcal{U},\mathcal{U}} \end{array}\right] , \tag{A.11}$$

and can be computed via standard kernel functions, e.g. RBF.

Proof: The KKT stationarity conditions involve looking for solutions where $\frac{\partial L}{\partial \delta \mathbf{W}} = 0$ and $\frac{\partial L}{\partial \delta \boldsymbol{\epsilon}} = 0$. The derivative of the Langrangian with respect to $\delta \mathbf{W}$ can be computed as

$$\frac{\partial L}{\partial \delta \mathbf{W}} = \left[(\mathbf{W_t} + \delta \mathbf{W})\phi(\mathbf{x}) - \mathbf{Y} \right] \phi(\mathbf{x})^T + \gamma(\mathbf{W}_t + \delta \mathbf{W}) + \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \phi(\bar{\mathbf{x}}_u)^T + \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \phi(\bar{\mathbf{x}}_v)^T \ . \tag{A.12}$$

By grouping terms and setting $\frac{\partial L}{\partial \delta \mathbf{W}} = 0$, we can write

$$\delta \mathbf{W} = \mathbf{A}\mathbf{Z} - \mathbf{W}_t \ , \tag{A.13}$$

where

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{\gamma} \left[(\mathbf{W_t} + \delta \mathbf{W})\phi(\mathbf{x}) - \mathbf{Y} \right]^T \\ \vdots \\ -\frac{1}{\gamma} \left(\mathbf{G}_{u'}^T \boldsymbol{\lambda}_{u'}^E \right)^T \\ \vdots \\ -\frac{1}{\gamma} \left(\mathbf{G}_s^T \boldsymbol{\lambda}_s^E + \mathbf{Q}_s^T \boldsymbol{\lambda}_s^I \right)^T \\ \vdots \\ -\frac{1}{\gamma} \left(\mathbf{Q}_{v'}^T \boldsymbol{\lambda}_{v'}^I \right)^T \\ \vdots \end{bmatrix}^T , \tag{A.14}$$

and $\mathbf{Z} = \begin{bmatrix} \phi(\mathbf{x}) & \cdots & \phi(\bar{\mathbf{x}}_{u'}) & \cdots & \phi(\bar{\mathbf{x}}_s) & \cdots & \phi(\bar{\mathbf{x}}_{v'}) & \cdots \end{bmatrix}^T$.

Since $\mathbf{A}$ still contains terms that depend on $\delta \mathbf{W}$, we need to solve for $\mathbf{A}$. To this end, we replace $\delta \mathbf{W}$ in the Lagrangian with its value from Eq. A.13, which yields

$$\begin{aligned} L = & \frac{1}{2}\|\mathbf{A}\mathbf{Z}\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{A}\mathbf{Z}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 & \text{(A.15)} \\ & + \sum_u \left[C_t^{(u)} + \mathbf{G}_u(\mathbf{A}\mathbf{Z} - \mathbf{W}_t)\phi(\bar{\mathbf{x}}_u) \right]^T \boldsymbol{\lambda}_u^E \\ & + \sum_v \left[D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v(\mathbf{A}\mathbf{Z} - \mathbf{W}_t)\phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)} \right]^T \boldsymbol{\lambda}_v^I \ . & \text{(A.16)} \end{aligned}$$

As before, to get a stationary point for $\mathbf{A}$, we compute the derivative of the Lagrangian, which can be written as

$$\frac{\partial L}{\partial \mathbf{A}} = \left[\mathbf{A}\mathbf{Z}\phi(\mathbf{x}) - \mathbf{Y} \right] \phi(\mathbf{x})^T \mathbf{Z}^T + \gamma \mathbf{A}\mathbf{Z}\mathbf{Z}^T + \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \phi(\bar{\mathbf{x}}_u)^T \mathbf{Z}^T + \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \phi(\bar{\mathbf{x}}_v)^T \mathbf{Z}^T \ . \tag{A.17}$$

Setting $\frac{\partial L}{\partial \mathbf{A}} = 0$ yields the linear system

$$\mathbf{A}\underbrace{[\mathbf{K}_{:,\mathcal{L}}\mathbf{K}_{\mathcal{L},:} + \gamma \mathbf{K}_{:,:}]}_{\mathbf{B}} = \underbrace{\mathbf{Y}\mathbf{K}_{\mathcal{L},:}}_{\mathbf{M}} - \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \phi(\bar{\mathbf{x}}_u)^T \mathbf{Z}^T - \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \phi(\bar{\mathbf{x}}_v)^T \mathbf{Z}^T \ , \tag{A.18}$$

where $\mathbf{B}$ is an $(N+V) \times (N+V)$ matrix, which in general has full rank if $\gamma > 0$ and $N + V \leq d$ (i.e., the dimensionality of the feature map). This lets us write

$$\mathbf{A} = \left[\mathbf{M} - \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \phi(\bar{\mathbf{x}}_u)^T \mathbf{Z}^T - \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \phi(\bar{\mathbf{x}}_v)^T \mathbf{Z}^T\right] \mathbf{B}^{-1} \ . \tag{A.19}$$

By making use of the kernel definition of Eq. A.11, we can then write

$$\mathbf{A} = \left[\mathbf{M} - \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \mathbf{K}_{u,:} - \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \mathbf{K}_{v,:}\right] \mathbf{B}^{-1} \ . \tag{A.20}$$

This, in conjunction with Eq. A.13 stating that $\delta\mathbf{W} = \mathbf{A}\mathbf{Z} - \mathbf{W}_t$, concludes the proof of the first part of Claim 1.□

To prove the second part, we study the derivative of the Langrangian with respect to $\delta\boldsymbol{\epsilon}$ for a particular unlabeled example. This derivative can be computed as

$$\frac{\partial L}{\partial \delta \boldsymbol{\epsilon}^{(v)}} = \alpha(\boldsymbol{\epsilon}_t^{(v)} + \delta\boldsymbol{\epsilon}^{(v)}) + \boldsymbol{\lambda}_v^I \odot \boldsymbol{\epsilon}_t^{(v)} \quad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ . \tag{A.21}$$

Setting $\frac{\partial L}{\partial \delta \boldsymbol{\epsilon}^{(v)}} = 0$ yields

$$\delta\boldsymbol{\epsilon}^{(v)} = -\left(\frac{1}{\alpha}\boldsymbol{\lambda}_v^I + \mathbf{1}\right) \odot \boldsymbol{\epsilon}_t^{(v)} \quad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \tag{A.22}$$

which requires $\alpha > 0$. While, strictly speaking, this condition prevents us from having completely free inequalities, in practice, a small α would only regularize the slack variable very weakly, thus effectively yielding true inequalities. This concludes the proof of the second part of Claim 1.□

Claim 2 *The solution to the constraints encoded by the KKT primal feasibility conditions in Eq. A.5 and Eq. A.6 takes the form* $\boldsymbol{\lambda} = \mathbf{S}^{-1}\mathbf{r}$*, where*

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1^E \\ \vdots \\ \boldsymbol{\lambda}_{N_E}^E \\ \boldsymbol{\lambda}_1^I \\ \vdots \\ \boldsymbol{\lambda}_{N_I}^I \end{bmatrix} , \quad \mathbf{S} = \left[\begin{array}{ccc|ccc} \mathbf{S}_{1,1}^{E,E} & \cdots & \mathbf{S}_{1,N_E}^{E,E} & \mathbf{S}_{1,1}^{E,I} & \cdots & \mathbf{S}_{1,N_I}^{E,I} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}_{N_E,1}^{E,E} & \cdots & \mathbf{S}_{N_E,N_E}^{E,E} & \mathbf{S}_{N_E,1}^{E,I} & \cdots & \mathbf{S}_{N_E,N_I}^{E,I} \\ \hline \mathbf{S}_{1,1}^{I,E} & \cdots & \mathbf{S}_{1,N_E}^{I,E} & \mathbf{S}_{1,1}^{I,I} & \cdots & \mathbf{S}_{1,N_I}^{I,I} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}_{N_I,1}^{I,E} & \cdots & \mathbf{S}_{N_I,N_E}^{I,E} & \mathbf{S}_{N_I,1}^{I,I} & \cdots & \mathbf{S}_{N_I,N_I}^{I,I} \end{array}\right] , \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_1^E \\ \vdots \\ \mathbf{r}_{N_E}^E \\ \mathbf{r}_1^I \\ \vdots \\ \mathbf{r}_{N_I}^I \end{bmatrix} , \tag{A.23}$$

and

$$
\begin{aligned}
\mathbf{S}_{u,a}^{E,E} &= \mathbf{G}_u \mathbf{G}_a^T (\mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,u}) \; , \\
\mathbf{S}_{u,b}^{E,I} &= \mathbf{G}_u \mathbf{Q}_b^T (\mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,u}) \; , \\
\mathbf{S}_{v,a}^{I,E} &= \mathbf{Q}_v \mathbf{G}_a^T (\mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,v}) \; , \\
\mathbf{S}_{v,b}^{I,I} &= \mathbf{Q}_v \mathbf{Q}_b^T (\mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,v}) + \delta_{v,b} \, diag\left(\frac{1}{\alpha} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} \right) \; , \\
\mathbf{r}_u^E &= \mathbf{G}_u \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,u} - \mathbf{G}_u \hat{\mathbf{y}}_{u,t} + C_t^{(u)} \; , \\
\mathbf{r}_v^I &= \mathbf{Q}_v \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,v} - \mathbf{Q}_v \hat{\mathbf{y}}_{v,t} + D_t^{(v)} - \frac{1}{2} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} \; ,
\end{aligned}
$$

with $\delta_{v,b}$ the Kronecker delta.

Proof: Given the solution for $\delta\mathbf{W}$ obtained from the previous claim, we can re-write the equality constraints for a particular unlabeled example as

$$
\begin{aligned}
0 &= \mathbf{G}_u \delta\mathbf{W} \phi(\bar{\mathbf{x}}_u) + C_t^{(u)} \qquad (A.24) \\
&= \mathbf{G}_u \left[\left(\mathbf{M} - \sum_{a=1}^{N_E} \mathbf{G}_a^T \boldsymbol{\lambda}_a^E \mathbf{K}_{a,:} - \sum_{b=1}^{N_I} \mathbf{Q}_b^T \boldsymbol{\lambda}_b^I \mathbf{K}_{b,:} \right) \mathbf{B}^{-1} \mathbf{Z} - \mathbf{W}_t \right] \phi(\bar{\mathbf{x}}_u) + C_t^{(u)} \qquad (A.25)
\end{aligned}
$$

By making use of the kernel definition of Eq. A.11 and of the prediction equation, this can be simplified as

$$
\mathbf{G}_u \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,u} - \sum_{a=1}^{N_E} \mathbf{G}_u \mathbf{G}_a^T \boldsymbol{\lambda}_a^E \mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,u} - \sum_{b=1}^{N_I} \mathbf{G}_u \mathbf{Q}_b^T \boldsymbol{\lambda}_b^I \mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,u} - \mathbf{G}_u \hat{\mathbf{y}}_{u,t} + C_t^{(u)} = 0 \; . \tag{A.26}
$$

Using the solutions for $\delta\boldsymbol{\epsilon}^{(v)}$ and $\delta\mathbf{W}$, we can re-write the inequality constraints for a particular unlabeled example as

$$
\begin{aligned}
0 &= \frac{1}{2} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v \delta\mathbf{W} \phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)} + D_t^{(v)} \; , \\
&= \frac{1}{2} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v \left[\left(\mathbf{M} - \sum_{a=1}^{N_E} \mathbf{G}_a^T \boldsymbol{\lambda}_a^E \mathbf{K}_{a,:} - \sum_{b=1}^{N_I} \mathbf{Q}_b^T \boldsymbol{\lambda}_b^I \mathbf{K}_{b,:} \right) \mathbf{B}^{-1} \mathbf{Z} - \mathbf{W}_t \right] \phi(\bar{\mathbf{x}}_v) \; , \\
&\quad - \left(\frac{1}{\alpha} \boldsymbol{\lambda}_v^I + \mathbf{1} \right) \odot \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + D_t^{(v)} \; . \qquad (A.27)
\end{aligned}
$$

As before, from the kernel definition of Eq. A.11 and the prediction equation, we obtain

$$
\begin{aligned}
0 &= \mathbf{Q}_v \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,v} - \sum_{a=1}^{N_E} \mathbf{Q}_v \mathbf{G}_a^T \boldsymbol{\lambda}_a^E \mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,v} - \sum_{b=1}^{N_I} \mathbf{Q}_v \mathbf{Q}_b^T \boldsymbol{\lambda}_b^I \mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,v} - \mathbf{Q}_v \hat{\mathbf{y}}_{v,t} \\
&- \left(\frac{1}{\alpha} \boldsymbol{\lambda}_v^I + \frac{1}{2} \mathbf{1} \right) \odot \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + D_t^{(v)} \; . \qquad (A.28)
\end{aligned}
$$

Combining all constraints for all unlabeled examples allows us to write the following systems of linear equations

$$\left[\begin{array}{ccc|ccc} \mathbf{S}^{E,E}_{1,1} & \cdots & \mathbf{S}^{E,E}_{1,N_E} & \mathbf{S}^{E,I}_{1,1} & \cdots & \mathbf{S}^{E,I}_{1,N_I} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}^{E,E}_{N_E,1} & \cdots & \mathbf{S}^{E,E}_{N_E,N_E} & \mathbf{S}^{E,I}_{N_E,1} & \cdots & \mathbf{S}^{E,I}_{N_E,N_I} \\ \hline \mathbf{S}^{I,E}_{1,1} & \cdots & \mathbf{S}^{I,E}_{1,N_E} & \mathbf{S}^{I,I}_{1,1} & \cdots & \mathbf{S}^{I,I}_{1,N_I} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}^{I,E}_{N_I,1} & \cdots & \mathbf{S}^{I,E}_{N_I,N_E} & \mathbf{S}^{I,I}_{N_I,1} & \cdots & \mathbf{S}^{I,I}_{N_I,N_I} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\lambda}^E_1 \\ \vdots \\ \boldsymbol{\lambda}^E_{N_E} \\ \boldsymbol{\lambda}^I_1 \\ \vdots \\ \boldsymbol{\lambda}^I_{N_I} \end{array}\right] = \left[\begin{array}{c} \mathbf{r}^E_1 \\ \vdots \\ \mathbf{r}^E_{N_E} \\ \mathbf{r}^I_1 \\ \vdots \\ \mathbf{r}^I_{N_I} \end{array}\right] , \tag{A.29}$$

where

$$\begin{aligned} \mathbf{S}^{E,E}_{u,a} &= \mathbf{G}_u \mathbf{G}_a^T (\mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,u}) , \\ \mathbf{S}^{E,I}_{u,b} &= \mathbf{G}_u \mathbf{Q}_b^T (\mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,u}) , \\ \mathbf{S}^{I,E}_{v,a} &= \mathbf{Q}_v \mathbf{G}_a^T (\mathbf{K}_{a,:} \mathbf{B}^{-1} \mathbf{K}_{:,v}) , \\ \mathbf{S}^{I,I}_{v,b} &= \mathbf{Q}_v \mathbf{Q}_b^T (\mathbf{K}_{b,:} \mathbf{B}^{-1} \mathbf{K}_{:,v}) + \delta_{v,b} \, diag\left(\frac{1}{\alpha} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} \right) , \\ \mathbf{r}^E_u &= \mathbf{G}_u \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,u} - \mathbf{G}_u \hat{\mathbf{y}}_{u,t} + C_t^{(u)} , \\ \mathbf{r}^I_v &= \mathbf{Q}_v \mathbf{M} \mathbf{B}^{-1} \mathbf{K}_{:,v} - \mathbf{Q}_v \hat{\mathbf{y}}_{v,t} + D_t^{(v)} - \frac{1}{2} \boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} . \end{aligned}$$

This concludes the proof of Claim 2. □

Claim 3 *Prediction for any input $\mathbf{x}_*$ in our kernelized model can be done in closed-form, and can be written as*

$$\hat{\mathbf{y}}_* = \mathbf{A} \mathbf{K}_{:,*} , \tag{A.30}$$

where

$$\mathbf{K}_{:,*} \;=\; \mathbf{Z} \phi(\mathbf{x}_*) = \left[\begin{array}{c|c} \mathbf{K}_{*,\mathcal{L}} & \mathbf{K}_{*,\mathcal{U}} \end{array} \right]^T ,$$

and $\mathbf{A}$ is defined in Eq. A.9.

Proof: The prediction can be computed as

$$\hat{\mathbf{y}}_* = \mathbf{W} \phi(\mathbf{x}_*) = (\mathbf{W}_t + \delta \mathbf{W}) \phi(\mathbf{x}_*) = \mathbf{A} \mathbf{Z} \phi(\mathbf{x}_*) = \mathbf{A} \mathbf{K}_{:,*} , \tag{A.31}$$

where we made use of Eq. A.13. This concludes the proof of Claim 3. □

REFERENCES

[1] Aanaes, H. & Kahl, F. (2002). Estimation of Deformable Structure and Motion. In *Vision and Modelling of Dynamic Scenes Workshop*. 24, 30

[2] Ahmed, A. & Farag, A. (2006). A New Formulation for Shape from Shading for Non-Lambertian Surfaces. In *Conference on Computer Vision and Pattern Recognition*. 30

[3] Akhter, I., Sheikh, Y. & Khan, S. (2009). In Defense of Orthonormality Constraints for Nonrigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*. 29

[4] Bartoli, A. & Olsen, S. (2005). A Batch Algorithm for Implicit Non-Rigid Shape and Motion Recovery. In *International Conference on Computer Vision*. 24

[5] Bartoli, A. & Zisserman, A. (2004). Direct Estimation of Non-Rigid Registration. In *British Machine Vision Conference*. 20

[6] Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S. & Sayd, P. (2008). Coarse-To-Fine Low-Rank Structure-From-Motion. In *Conference on Computer Vision and Pattern Recognition*. 28, 30

[7] Bartoli, A., Y. Gerard, F.C. & T.Collins (2012). On Template-Based Reconstruction from a Single View: Analytical Solutions and Proofs of Well-Posedness for Developable, Isometric and Conformal Surfaces. In *Conference on Computer Vision and Pattern Recognition*. 28

[8] Bathe, K.J. (1982). *Finite Element Procedures in Engineering Analysis*. Prentice Hall. 20

[9] Belhumeur, P., Kriegman, D. & Yuille, A. (1999). The Bas-Relief Ambiguity. *International Journal of Computer Vision*, **35**, 33–44. 55

[10] Belkin, M. & Niyogi, P. (2001). Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, 585–591, MIT Press. 23

[11] Bertsekas, D. (1999). *Nonlinear Programming*. Athena Scientific. 61

[12] BESL, P. & MCKAY, N. (1992). A Method for Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 239–256. 71

[13] BHAT, K.S., TWIGG, C.D., HODGINS, J.K., KHOSLA, P.K., POPOVIC, Z. & SEITZ, S.M. (2003). Estimating Cloth Simulation Parameters from Video. In *ACM Symposium on Computer Animation*. 77

[14] BISHOP, C. (2006). *Pattern Recognition and Machine Learning*. Springer. 55

[15] BLANZ, V. & VETTER, T. (1999). A Morphable Model for the Synthesis of 3D Faces. In *ACM SIGGRAPH*, 187–194. 76

[16] BORNEMANN, F. & RASCH, C. (2006). Finite-Element Discretization of Static Hamilton-Jacobi Equations Based on a Local Variational Principle. *Computing and Visualization in Science*, **9**. 71

[17] BOYD, S. & VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge University Press. 26

[18] BRAND, M. (2001). Morphable 3D Models from Video. *Journal of Machine Learning Research*. 24, 30, 76

[19] BRAND, M. (2005). A Direct Method of 3D Factorization of Nonrigid Motion Observed in 2D. In *Conference on Computer Vision and Pattern Recognition*, 122–128. 29

[20] BRANDT, S., KOSKENKORVA, P., KANNALA, J. & HEYDEN, A. (2009). Uncalibrated Non-Rigid Factorisation with Automatic Shape Basis Selection. In *CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*. 30

[21] BREGLER, C., HERTZMANN, A. & BIERMANN, H. (2000). Recovering Non-Rigid 3D Shape from Image Streams. In *Conference on Computer Vision and Pattern Recognition*. 28, 76

[22] BRIDSON, R., MARINO, S. & FEDKIW, R. (2003). Simulation of Clothing With Folds and Wrinkles. In *ACM Symposium on Computer Animation*. 21

[23] BRUNET, F., HARTLEY, R., BARTOLI, A., NAVAB, N. & MALGOUYRES, R. (2010). Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces. In *Asian Conference on Computer Vision*. 77

[24] COHEN, L. & COHEN, I. (1993). Finite-Element Methods for Active Contour Models and Balloons for 2D and 3D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1131–1147. 22

[25] COURTOIS, N., KLIMOV, A., PATARIN, J. & SHAMIR, A. (2000). Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In *EUROCRYPT*. 27

[26] DEL BUE, A., SMERALDI, F. & AGAPITO, L. (2007). Non-Rigid-Structure from Motion Using Ranklet-Based Tracking and Non-Linear Optimization. *Image and Vision Computing*, **25**, 297–310. 30

[27] DUROU, J.D., FALCONE, M. & SAGONA, M. (2008). Numerical Methods for Shape from Shading : A New Survey with Benchmarks. *Computer Vision and Image Understanding*. 62

[28] ECKER, A., JEPSON, A. & KUTULAKOS, K. (2008). Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities. In *European Conference on Computer Vision*. 27, 47, 77

[29] FALCONE, M. & SAGONA, M. (1997). An Algorithm for Global Solution of the Shape-From-Shading Model. In *International Conference on Image Analysis and Processing*. xi, xii, 62, 64, 65, 73

[30] FAYAD, J., DEL BUE, A., AGAPITO, L. & AGUIAR, P.M.Q. (2009). Non-Rigid Structure from Motion Using Quadratic Deformation Models. In *British Machine Vision Conference*. 30, 76

[31] FAYAD, J., AGAPITO, L. & DEL BUE, A. (2010). Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences. In *European Conference on Computer Vision*. 30, 31, 47, 76

[32] FORSYTH, D. & ZISSERMAN, A. (1991). Reflections on Shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 671–679. 30

[33] FUA, P. & LECLERC, Y.G. (1995). Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *International Journal of Computer Vision*, **16**, 35–56. 22

[34] GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M. & GRINSPUN, E. (2007). Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*. v, 5, 6

[35] GUMEROV, N., ZANDIFAR, A., DURAISWAMI, R. & DAVIS, L. (2004). Structure of Applicable Surfaces from Single Views. In *European Conference on Computer Vision*. 26, 47

[36] HAN, M., XU, W., TAO, H. & GONG, Y. (2004). An Algorithm for Multiple Object Trajectory Tracking. In *Conference on Computer Vision and Pattern Recognition*, 864–871. 72

[37] HARTLEY, R. & SCHAFFALITZKY, F. (2004). Reconstruction from Projections Using Grassmann Tensors. In *European Conference on Computer Vision*, 363–375. 29

[38] HARTLEY, R. & VIDAL, R. (2008). Perspective Nonrigid Shape and Motion Recovery. In *European Conference on Computer Vision*. 29

[39] HARTLEY, R. & VIDAL, R. (2008). Perspective Nonrigid Shape and Motion Recovery. In *European Conference on Computer Vision*. 76

[40] HARTLEY, R. & ZISSERMAN, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press. 36

[41] HORN, B. (1975). *Obtaining Shape from Shading Information*. Mc-Graw Hill, New York. 10

[42] HORN, B. & BROOKS, M. (1989). *Shape from Shading*. MIT Press. 30

[43] JOLLIFFE, I.T. (1986). *Principal Component Analysis*. Springer-Verlag. 23

[44] KASS, M., WITKIN, A. & TERZOPOULOS, D. (1988). Snakes: Active Contour Models. *International Journal of Computer Vision*, **1**, 321–331. 20, 22

[45] KOLMOGOROV, V. (2006). Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 1568–1583. 61

[46] KOZERA, R. (1997). Uniqueness in Shape from Shading Revisited. *Journal of Mathematical Imaging and Vision*, **7**, 123–138. 55

[47] KRIEGMAN, D. & BELHUMEUR, P. (1998). What Shadows Reveal About Object Structure. In *European Conference on Computer Vision*, 399–414. 30

[48] LAWRENCE, N.D. (2004). Gaussian Process Models for Visualisation of High Dimensional Data. In *Advances in Neural Information Processing Systems*, MIT Press. 23

[49] LAWRENCE, N.D. (2005). Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, **6**, 1783–1816. 16, 76

[50] LIANG, J., DEMENTHON, D. & DOERMANN, D. (2005). Flattening Curved Documents in Images. In *Conference on Computer Vision and Pattern Recognition*, 338–345. 47

[51] LLADO, X., DEL BUE, A. & AGAPITO, L. (2010). Non-Rigid Metric Reconstruction from Perspective Cameras. *Image and Vision Computing*, **28**, 1339–1353. 28, 29, 76

[52] LOWE, D. (1999). Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision*, 1150–1157. 35

[53] MALIS, E. & VARGAS, M. (2007). Deeper Understanding of the Homography Decomposition for Vision-Based Control. Technical report, INRIA. 37

[54] MCINERNEY, T. & TERZOPOULOS, D. (1993). A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *International Conference on Computer Vision*, 518–523. 22, 77

[55] MCINERNEY, T. & TERZOPOULOS, D. (1995). A Dynamic Finite Element Surface Model for Segmentation and Tracking in Multidimensional Medical Images with Application to Cardiac 4d Image Analysis. *Computerized Medical Imaging and Graphics*, **19**, 69–83. 22

[56] MCINERNEY, T. & TERZOPOULOS, D. (1996). Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis*, **1**, 91–108. 22

[57] METAXAS, D. & TERZOPOULOS, D. (1993). Constrained Deformable Superquadrics and Nonrigid Motion Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 580–591. 77

[58] MONTAGNAT, J., DELINGETTE, H. & AYACHE, N. (2001). A Review of Deformable Surfaces: Topology, Geometry and Deformation. *Image and Vision Computing*, **19**, 1023–1040. 22

[59] MORENO-NOGUER, F., SALZMANN, M., LEPETIT, V. & FUA, P. (2009). Capturing 3D Stretchable Surfaces from Single Images in Closed Form. In *Conference on Computer Vision and Pattern Recognition*. 27, 31

[60] MORENO-NOGUER, F., PORTA, J. & FUA, P. (2010). Exploring Ambiguities for Monocular Non-Rigid Shape Estimation. In *European Conference on Computer Vision*. 31, 67

[61] NASTAR, C. (1994). Vibration Modes for Nonrigid Motion Analysis in 3D Images. In *European Conference on Computer Vision*. 21

[62] NASTAR, C. & AYACHE, N. (1993). Fast Segmentation, Tracking, and Analysis of Deformable Objects. In *International Conference on Computer Vision*, 275–279. 21

[63] NASTAR, C. & AYACHE, N. (1996). Frequency-Based Nonrigid Motion Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**. 27, 77

[64] NAYAR, S., IKEUCHI, K. & KANADE, T. (1991). Shape from Interreflections. *International Journal of Computer Vision*, **6**, 173–195. 30

[65] OREN, M. & NAYAR, S. (1996). A Theory of Specular Surface Geometry. *International Journal of Computer Vision*, **24**, 105–124. 30

[66] PENTLAND, A. (1990). Automatic Extraction of Deformable Part Models. *International Journal of Computer Vision*, **4**, 107–126. 21

[67] PENTLAND, A. (1994). Local Shading Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vi, 11

[68] PENTLAND, A. & SCLAROFF, S. (1991). Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 715–729. 21, 27, 77

[69] PERRIOLLAT, M. & BARTOLI, A. (2007). A Quasi-Minimal Model for Paper-Like Surfaces. In *BenCos: Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*. 26, 47

[70] PERRIOLLAT, M., HARTLEY, R. & BARTOLI, A. (2008). Monocular Template-Based Reconstruction of Inextensible Surfaces. In *British Machine Vision Conference*. 27, 35, 47

[71] PERRIOLLAT, M., HARTLEY, R. & BARTOLI, A. (2011). Monocular Template-Based Reconstruction of Inextensible Surfaces. *International Journal of Computer Vision*, **95**. 77

[72] PILET, J., LEPETIT, V. & FUA, P. (2008). Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation. *International Journal of Computer Vision*, **76**. 22

[73] PRADOS, E. & FAUGERAS, O. (2005). Shape from Shading: A Well-Posed Problem ? In *Conference on Computer Vision and Pattern Recognition*. 11

[74] RABAUD, V. & BELONGIE, S. (2008). Re-Thinking Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*. 30, 76

[75] RABAUD, V. & BELONGIE, S. (2009). Linear Embeddings in Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*. 30

[76] RAMAMOORTHI, R. & HANRAHAN, P. (2001). An Efficient Representation for Irradiance Environment Maps. In *ACM SIGGRAPH*. 16, 51, 52

[77] ROWEIS, S. & SAUL, L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, **290**, 2323–2326. 23

[78] SALZMANN, M. (2009). *Learning and Recovering 3D Surface Deformations*. Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne. vi, 6, 8, 9, 10, 24, 25, 27

[79] SALZMANN, M. & FUA, P. (2010). *Deformable Surface 3D Reconstruction from Monocular Images*. Morgan-Claypool Publishers. 28

[80] SALZMANN, M. & FUA, P. (2011). Linear Local Models for Monocular Reconstruction of Deformable Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 931–944. xi, xii, 15, 27, 28, 47, 49, 50, 59, 62, 63, 67, 73, 77, 79, 86

[81] SALZMANN, M., HARTLEY, R. & FUA, P. (2007). Convex Optimization for Deformable Surface 3D Tracking. In *International Conference on Computer Vision*. 26, 35

[82] SALZMANN, M., LEPETIT, V. & FUA, P. (2007). Deformable Surface Tracking Ambiguities. In *Conference on Computer Vision and Pattern Recognition*. 26

[83] SALZMANN, M., PILET, J., ILIĆ, S. & FUA, P. (2007). Surface Deformation Models for Non-Rigid 3D Shape Recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 1481–1487. 24, 27

[84] SALZMANN, M., MORENO-NOGUER, F., LEPETIT, V. & FUA, P. (2008). Closed-Form Solution to Non-Rigid 3D Surface Registration. In *European Conference on Computer Vision*. viii, 27, 28, 31, 35, 44, 45

[85] SALZMANN, M., URTASUN, R. & FUA, P. (2008). Local Deformation Models for Monocular 3D Shape Recovery. In *Conference on Computer Vision and Pattern Recognition*. 24, 27, 28, 41, 53, 76

[86] SAMARAS, D. & METAXAS, D. (1998). Incorporating Illumination Constraints in Deformable Models. In *Conference on Computer Vision and Pattern Recognition*, 322–329. 31

[87] SAMARAS, D., METAXAS, D., FUA, P. & LECLERC, Y. (2000). Variable Albedo Surface Reconstruction from Stereo and Shape from Shading. In *Conference on Computer Vision and Pattern Recognition*. 31

[88] SCHOELKOPF, B., BURGES, C. & SMOLA, A. (1999). Advances in Kernel Methods. In *Support Vector Learning*, MIT Press. 23

[89] SHAJI, A., VAROL, A., TORRESANI, L. & FUA, P. (2010). Simultaneous Point Matching and 3D Deformable Surface Reconstruction. In *Conference on Computer Vision and Pattern Recognition*. 27

[90] SHEN, S., SHI, W. & LIU, Y. (2009). Monocular 3D Tracking of Inextensible Deformable Surfaces Under L2-Norm. In *Asian Conference on Computer Vision*. 47, 77

[91] STURM, J. (1999). Using Sedumi 1.02, a Matlab Toolbox for Optimization Over Symmetric Cones. 42

[92] TADDEI, P. & BARTOLI, A. (2008). Template-based Paper Reconstruction from a Single Image is Well Posed when the Rulings are Parallel. In *CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*. 26

[93] TAYLOR, J., JEPSON, A.D. & KUTULAKOS, K.N. (2010). Non-Rigid Structure from Locally-Rigid Motion. In *Conference on Computer Vision and Pattern Recognition*. 30, 31, 47

[94] T.COLLINS & A.BARTOLI (2010). Locally Affine and Planar Deformable Surface Reconstruction from Video. In *Vision, Modeling, and Visualization*. 30, 31

[95] TENENBAUM, J., DE SILVA, V. & LANGFORD, J. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, **290**, 2319–2323. 23

[96] TERZOPOULOS, D. & METAXAS, D. (1991). Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 703–714. 22

[97] TERZOPOULOS, D., WITKIN, A. & KASS, M. (1987). Symmetry-Seeking Models and 3D Object Reconstruction. *International Journal of Computer Vision*, **1**, 211–221. 20

[98] TERZOPOULOS, D., WITKIN, A. & KASS, M. (1988). Constraints on Deformable Models: Recovering 3D Shape and Nongrid Motion. *Artificial Intelligence*, **36**, 91–123. 20

[99] TOMASI, C. & KANADE, T. (1992). Shape and Motion from Image Streams Under Orthography: A Factorization Method. *International Journal of Computer Vision*, **9**, 137–154. 28, 29

[100] TORRESANI, L. & HERTZMANN, A. (2004). Automatic Non-Rigid 3D Modeling from Video. In *European Conference on Computer Vision*, 299–312. 24

[101] TORRESANI, L., YANG, D.B., ALEXANDER, E.J. & BREGLER, C. (2001). Tracking and Modeling Non-Rigid Objects with Rank Constraints. In *Conference on Computer Vision and Pattern Recognition*, 493–500. 24

[102] TORRESANI, L., HERTZMANN, A. & BREGLER, C. (2003). Learning Non-Rigid 3D Shape from 2D Motion. In *Advances in Neural Information Processing Systems*, MIT Press. 24

[103] TORRESANI, L., HERTZMANN, A. & BREGLER, C. (2008). Nonrigid Structure-From-Motion: Estimating Shape and Motion with Hierarchical Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 878–892. 24, 29, 30, 76

[104] TSAI, P.S. & SHAH, M. (1994). Shape from Shading Using Linear Approximation. *Journal of Image and Vision Computing*, 69–82. xi, xii, 62, 64, 65, 73

[105] TSAP, L., GOLDGOF, D. & SARKAR, S. (2000). Nonrigid Motion Analysis Based on Dynamic Refinement of Finite Element Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 526–543. 77

[106] URTASUN, R. & DARRELL, T. (2008). Sparse Probabilistic Regression for Activity-Independent Human Pose Inference. In *Conference on Computer Vision and Pattern Recognition*. 55

[107] VAROL, A., SALZMANN, M., TOLA, E. & FUA, P. (2009). Template-Free Monocular Reconstruction of Deformable Surfaces. In *International Conference on Computer Vision*. 17, 30, 47

[108] VAROL, A., SALZMANN, M., FUA, P. & URTASUN, R. (2012). A Constrained Latent Variable Model. In *Conference on Computer Vision and Pattern Recognition*. 17

[109] VAROL, A., SHAJI, A., SALZMANN, M. & FUA, P. (2012). Monocular 3D Reconstruction of Sparsely Textured Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17

[110] VIDAL, R. & ABRETSKE, D. (2006). Nonrigid Shape and Motion from Multiple Perspective Views. In *European Conference on Computer Vision*. 28

[111] VOLINO, P. & MAGNENAT-THALMANN, N. (2001). Comparing Efficiency of Integration Methods for Cloth Simulation. In *Computer Graphics International*, 265–274. 21

[112] WANG, G. & WU, Q. (2010). Quasi-Perspective Projection Model: Theory and Application to Structure and Motion Factorization from Uncalibrated Image Sequences. *International Journal of Computer Vision*, **87**, 213–234. 28

[113] WANG, J., FLEET, D. & HERTZMANN, A. (2005). Gaussian Process Dynamical Models. In *Advances in Neural Information Processing Systems*. 23

[114] WEINBERGER, K.Q. & SAUL, L.K. (2004). Unsupervised Learning of Image Manifolds by Semidefinite Programming. In *Conference on Computer Vision and Pattern Recognition*. 23

[115] WEISE, T., LEIBE, B. & VAN GOOL, L. (2007). Fast 3D Scanning with Automatic Motion Compensation. In *Conference on Computer Vision and Pattern Recognition*. 71

[116] WHITE, R. & FORSYTH, D. (2006). Combining Cues: Shape from Shading and Texture. In *Conference on Computer Vision and Pattern Recognition*. 31, 47

[117] XIAO, J. & KANADE, T. (2004). Non-Rigid Shape and Motion Recovery: Degenerate Deformations. In *Conference on Computer Vision and Pattern Recognition*, 668–675. 29, 30

[118] XIAO, J. & KANADE, T. (2005). Uncalibrated Perspective Reconstruction of Deformable Structures. In *International Conference on Computer Vision*. 28, 29, 30, 76

[119] ZHANG, Z. (1994). Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision*, **13**, 119–152. 40

[120] ZHANG, Z. & HANSON, A. (1995). Scaled Euclidean 3D Reconstruction Based on Externally Uncalibrated Cameras. In *IEEE Symposium on Computer Vision*, 37–42. 37

[121] ZHANG, Z., TAN, C. & FAN, L. (2004). Restoration of Curved Document Images through 3D Shape Modeling. In *Conference on Computer Vision and Pattern Recognition*. 31, 47

[122] ZHU, J., HOI, S., STEVEN, C., XU, Z. & LYU, M. (2008). An Effective Approach to 3D Deformable Surface Tracking. In *European Conference on Computer Vision*, 766–779. 35, 47

[123] ZIENKIEWICZ, O. (1989). *The Finite Element Method*. McGraw-Hill. 20

Curriculum Vitae

Aydin Varol
Born October 13, 1983, Turkey
E-mail : aydin.varol@gmail.com

Education

2007-2012 Ph.D candidate in Computer Vision. Advisor: Prof. Pascal Fua.
Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne.
Title: Resolving Ambiguities in Monocular 3D Reconstruction of Deformable Surfaces.

2005-2007 M. Sc. in Mechanical Engineering. Advisor: Prof. Cagatay Basdogan.
Koc University, Istanbul, Turkey.

2000-2005 B. Sc. in Computer Engineering.
B. Sc. in Mechanical Engineering.
Koc University, Istanbul, Turkey.

Teaching Experience

2008-2012 Teaching Assistant, EPFL.
2005-2007 Teaching Assistant, Koc University.

Journal Publications

1. A. Varol, A. Shaji, M. Salzmann and P. Fua, Monocular 3D Reconstruction of Locally Textured Surfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June, 2012.
2. A. Varol, I. Gunev, B. Orun and C. Basdogan, Numerical Simulation of Nano Scanning in Intermittent-Contact Mode AFM under Q control, *Nanotechology*, 2008.
3. C. Basdogan, A. Kiraz, I. Bukusoglu, A. Varol and S. Doganay, Haptic Guidance for Improved Task Performance in Steering Microparticles with Optical Tweezers, *Optics Express*, Vol. 15, No. 18, 2007.
4. I. Gunev, A. Varol, S. Karaman and C.Basdogan, Adaptive Q control for Tapping-mode Nano-scanning Using a Piezo-actuated Bimorph Probe, *Review of Scientific Instruments*, Vol. 78, No. 043707, 2007.

Conference Publications

1. J. Ostlund, A. Varol, D. Ngo and P. Fua, Laplacian Meshes for Monocular 3D Shape Recovery, *European Conference on Computer Vision*, 2012.
2. A. Varol, M. Salzmann, P. Fua and R. Urtasun, A Constrained Latent Variable Model, *Conference on Computer Vision and Pattern Recognition*, 2012.
3. A. Shaji, A. Varol, P. Fua, Yashoteja, A. Jain and S. Chandran, Resolving Occlusion in Multiframe Reconstruction of Deformable Surfaces, *Computer Vision and*

Pattern Recognition Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, 2011.

4. A. Shaji, A. Varol, L. Torresani and P. Fua, Simultaneous Point Matching and 3D Deformable Surface Reconstruction, *Conference on Computer Vision and Pattern Recognition*, 2010.
5. A. Varol, M. Salzmann, E.Tola and P. Fua, Template-Free Monocular Reconstruction of Deformable Surfaces, *International Conference on Computer Vision*, 2009.
6. A. Varol, I. Gunev and C. Basdogan, A Virtual Reality Toolkit for Path Planning and Manipulation at Nano-Scale, *Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2006.

Technical Reports

1. P. Fua, A. Varol, R. Urtasun and M. Salzmann, Least-Squares Minimization Under Constraints, EPFL, Technical report, Nr. 150790, 2010.