



*ECOLE DOCTORALE
DES SCIENCES POUR L'INGENIEUR*

THÈSE

Présentée à l'Université Clermont Auvergne
pour l'obtention du grade de **Docteur**

Specialité
VISION PAR ORDINATEUR

**Contributions to Plane-based and Deformable
3D Reconstruction from Monocular Images**

Toby Collins

Rapporteur Peter STURM, Directeur de Recherche, INRIA Rhône-Alpes
Rapporteur João Paulo COSTEIRA, Associate professor, Instituto Superior Técnico
Examineur Marie-Odile BERGER, Senior researcher, INRIA Nancy Grand Est
Directeur de thèse Adrien BARTOLI, Professeur, Université Clermont Auvergne



**EnCoV, Institut Pascal, UMR 6602 CNRS, Université
Clermont Auvergne, CHU de Clermont-Ferrand**
Faculté de Médecine
28 place Henri Dunant, 63000, Clermont-Ferrand
Tel: +33 4 73 17 81 23

To Millie and my parents, Adrian and Jenni. Without their love and support this thesis could not have been possible.

Abstract

Reconstructing 3D geometric information from monocular images has been one of the central areas of study in computer vision for well over four decades, with many challenges still open. There exists no universal solution, and a large taxonomy of problems and approaches covering different scenarios has grown and matured over the years. One of the largest areas of study is 3D reconstruction from image motion. This in turn has many problem flavors depending on factors such as the available motion information (*e.g.* sparse or dense motion), the reconstruction representation, the correctness of the motion information, available prior knowledge and whether or not the camera is intrinsically calibrated. This thesis describes novel technical and theoretical contributions that have advanced state-of-the-art in three of the fundamental reconstruction problems, which are intimately related.

The first reconstruction problem is to jointly estimate the poses of cameras viewing an unknown planar structure and to recover its metric structure. This problem is known as *Plane-based Structure from Motion* (PSfM). PSfM is well-understood with the perspective camera model and it has a closed-form solution using homography decomposition. However, when the structure is small and/or viewed far relative to focal length, the perspective effects diminish, and the projections tends to affine with parallel projection rays. In these situations homography decomposition fails because the problem itself becomes ill-posed. We propose a stable alternative using affine camera models which have been used extensively to reconstruct non-planar structures. However, reconstruction with planar structures is fundamentally different because the affine camera models that we can use are more restricted and reconstruction is inherently more ambiguous and non-linear. Indeed, reconstruction of planar structures is not possible with weak- and para-perspective cameras (two of the most common affine camera models). We provide a general, accurate and closed-form solution for the orthographic camera model (PSfM-O), equivalent to a weak-perspective camera with constant image scaling. The solution finds all metric structure solutions and camera poses, which was not possible with previous solutions. It produces very accurate results and there does not appear to be a significant benefit to refining the solutions with bundle adjustment. We also present a new theoretical analysis that deepens our understanding of the problem. The main result is a complete geometric characterization of degenerate scenes. This analysis tells us both the geometric configurations that are required to solve the problem, and quasi-degenerate configurations where the problem is theoretically solvable but a small amount of noise will cause highly unstable results.

The second reconstruction problem is to estimate the 3D pose of a perspective camera relative to a known planar object from point correspondences. This is a well-known problem and we refer to it as *Plane-based Pose Estimation with a Perspective camera* (PPE-P). However, this problem is still open because there does not exist a fast closed-form solution that is statistically optimal. The fastest previous approaches decompose the object-to-image homography matrix. However, they have two main limitations. Firstly, they are not particularly accurate especially with strong noise. Secondly, they fail in practice when perspective effects are small, such as pose estimation with Augmented Reality (AR) markers. Slower and more accurate methods exist, with the most accurate being iterative reprojection error refinement with *e.g.* Levenberg-Marquardt (LM). We present a novel analytical solution called *Infinitesimal Plane-based Pose estimation* (IPPE) that is substantially more accurate than previous homography decomposition methods, and just as fast. IPPE is also much more stable when the perspective effects diminish. Furthermore, if the object points are arranged in a regular pattern such as the corners of a square, results from IPPE are extremely accurate and there is little benefit in refining its solutions with LM. Such point configurations are a very common use case with

e.g. AR markers.

The third reconstruction problem is Shape from Template (SfT) with a perspective camera and when a key intrinsic is unknown *a priori*: the camera’s focal length. SfT is the problem of jointly registering a deformable 3D object with a monocular image and reconstructing its 3D shape using a template. The template gives the object’s 3D shape in a known rest position, acquired using *e.g.* dense Multi-view Stereo (MVS). The template also constrains how the object can physically deform from its rest position, which is required to make SfT well-posed. SfT is solved by estimating the deformable transformation of the template to camera coordinates, typically using point matches between the template’s surface and the image. The point matches can be found using existing sparse approaches (*i.e.* keypoint matching) or dense approaches. SfT has various applications and in particular AR with deforming objects. A very common assumption is that the object deforms quasi-isometrically (*i.e.* limited stretching or shrinking). This assumption can make SfT well-posed and it is valid for many real-world objects made of paper, cardboard, thick rubber, leather, cloth and tightly-woven fabric.

SfT has been extensively studied with a calibrated perspective camera and quasi-isometric deformation. However, a dedicated camera calibration process is required, typically using images of a rigid calibration target such as a checkerboard. This is limiting and cumbersome in real world applications. We show that SfT can be solved accurately with quasi-isometric deformation and when one of the key intrinsics is unknown: the camera’s focal length. We solve focal length and the template’s deformation simultaneously from one or more images. We refer to this problem as *focal length and Shape-from-Template* (fSfT). fSfT has important practical use because for many real cameras, the remaining intrinsics (principal point, skew, aspect ratio) can take standard known values and lens distortion can be neglected. fSfT is not trivial because it is a large-scale non-convex problem with no known closed-form solution. We present two novel and complementary methods to solve fSfT in practice: an analytical method and an optimization-based method. The analytical method uses motion within one or more local surface regions. A non-holonomic partial differential equation is constructed for each local region that has a unique and linear focal length solution. We then apply robust averaging to aggregate focal length solutions from multiple local regions, which removes outliers and mitigates noise. The advantage of the analytical method is its very low computational cost and it permits the analysis of fSfT in terms of well-posedness and degenerate configurations. In contrast, the optimization-based method solves fSfT with iterative minimization of a large-scale non-convex cost function. Unlike the analytical method, it uses all available geometric constraints from the template, and in practice it is substantially more accurate than the analytical method. The cost function combines a data cost from point correspondences, and a deformation cost that penalizes non-isometric deformation. Designing and efficiently optimizing the cost function is not straightforward. We make several contributions in these regards. Firstly, we carefully normalize the cost function, which allows the same hyper-parameters, such as the weight of the deformation cost, to be used for a wide variety of problem configurations. Hyper-parameter tuning is a known issue in prior methods that solve SfT with cost optimization and the normalization techniques we propose could also be used in the calibrated setting. Secondly, we show that the weight of the deformation cost, which is a critical hyper-parameter, can be automatically set in an unsupervised manner based on the problem’s conditioning number. Thirdly, we show that the cost has a relatively wide basin of convergence and there is no need to initialize with a highly accurate focal length or deformation estimate. In practice, initialization can be achieved with either the analytical method or by focal length sampling. In the latter case, only a small number of samples are required (typically no more than three samples that correspond to short, medium and long focal lengths). We also give a mechanism in the optimization process to avoid redundant

search from different focal length samples. Fourthly, we extend the optimization-based approach to handle multiple images sharing a common unknown focal length. We show that focal length and the template’s deformations for all views can be jointly optimized with super-linear convergence using the Schur complement. The optimization process has linear computational cost and linear memory overhead in the number of images, allowing it to scale well to a large number of images. We test our fSfT methods on 12 public datasets and we show a significant benefit for solving fSfT with multiple views compared to a single view.

In summary, this thesis presents novel contributions in three reconstruction problems that are different but they are indeed related. Specifically, the contributions for each problem are unified by three common themes that run throughout the thesis. These are *(i)* novel closed-form solutions from motion equations, *(ii)* theoretical problem analysis using the closed-form solutions, and *(iii)* combining closed-form solutions with non-convex optimization to achieve highly accurate reconstructions.

Keywords: Pose estimation, Shape-from-Template, Structure-from-Motion, Non-Rigid Structure-from-Motion, As-Rigid-As-Possible, Degeneracy, Critical Configurations

Contents

1	Introduction and Thesis Contributions	1
1.1	Personal context	3
1.2	Computer vision context	3
1.2.1	Monocular reconstruction	3
1.2.2	Reconstruction and image registration	5
1.2.3	Geometry-based algorithms and the era of deep learning	5
1.3	Problems tackled in this thesis, motivation, and contribution summary	9
1.3.1	Chapter 3: Plane-based Structure-from-Motion with Affine cameras (PSfM-A)	9
1.3.2	Chapter 4: Plane-based Pose Estimation with a Perspective camera (PPE-P)	11
1.3.3	Chapter 5: Focal length and Shape-from-Template (fSfT)	15
1.4	Thesis organization and relation to publications	19
2	Background and Related Works	21
2.1	Review of camera models	23
2.1.1	Perspective cameras	23
2.1.2	Affine cameras	24
2.2	Problem relationships	26
2.3	Plane-based Pose Estimation (PPE)	31
2.3.1	Homography decomposition methods with perspective cameras	31
2.3.2	Failure of homography decomposition methods in quasi-affine conditions	32
2.3.3	Perspective- n -Point methods	32
2.3.4	Extensions of PnP to partially calibrated or uncalibrated cameras	34
2.4	Plane-based Structure-from-Motion (PSfM)	34
2.4.1	Solutions with perspective cameras	34
2.4.2	Solutions with affine cameras	36
2.5	Shape-from-Template (SfT)	36
2.5.1	Template modeling	36
2.5.2	Registration	39
2.5.3	Closed-form SfT-P solutions	40
2.5.4	Optimization-based SfT-P solutions	40
2.5.5	CNN-based solutions	40

3	Plane-based Structure-from-Motion with Affine Cameras	43
3.1	Problem setup and solution overview	45
3.1.1	Scene geometry and notation	45
3.1.2	Projection model instantiation	46
3.1.3	Why previous stratified methods cannot solve PSfM-O	46
3.1.4	Overview of proposed solution	47
3.2	PSfM-O technical solution	48
3.2.1	Upgrade constraints	48
3.2.2	Upgrade parameterization	49
3.2.3	Computing affine structure from point correspondences	49
3.2.4	Exact-PSfM-O: An optimal PSfM-O solution with three views	50
3.2.5	Approx-PSfM-O: A least-squares PSfM-O solution with three or more views	52
3.3	Theoretical problem analysis	56
3.3.1	Section overview	56
3.3.2	Definitions	56
3.3.3	New theoretical results for PSfM-O	57
3.3.4	New theoretical results for PSfM-WP and PSfM-PP	59
3.3.5	Summary of the differences between stratified SfM with affine cameras for planar versus non-planar structures	60
3.4	Empirical evaluation	61
3.4.1	Method comparison summary	61
3.4.2	MOVA: The fallback method	62
3.4.3	Error metrics	62
3.4.4	Experiments with simulated data	63
3.4.5	Experiments with real data	65
3.5	Conclusion	70
4	Infinitesimal Plane-based Pose Estimation	73
4.1	IPPE motivation: Overcoming the limits of previous PPE-P methods	75
4.2	Methodology	76
4.2.1	Definitions	76
4.2.2	Approach overview	76
4.2.3	Problem statement	77
4.2.4	Solution	78
4.3	IPPE theoretical analysis	82
4.3.1	Theorems	82
4.3.2	Pose disambiguation using reprojection error	83
4.4	Experimental evaluation with simulated data	84
4.4.1	Simulation setup	84
4.4.2	Well-posed and ill-posed conditions	85
4.4.3	Summary of experimental parameters and error metrics	85
4.4.4	IPPE versus PHD methods	86
4.4.5	IPPE versus PnP methods	90
4.4.6	IPPE Versus P3P/RPnP with virtual correspondences	98
4.5	Experimental evaluation with real data	98

4.5.1	Pose estimation from keypoint matches	98
4.5.2	Pose estimation of checkerboards	100
4.5.3	Pose estimation of Augmented Reality markers	102
4.6	Conclusion	105
5	Focal Length and Shape-from-Template	107
5.1	Solving fSfT analytically	109
5.1.1	Section overview	109
5.1.2	PDE formulation	109
5.1.3	Instantiation with the pinhole model	109
5.1.4	Local weak-perspective solution	110
5.1.5	Focal length solution	111
5.1.6	Implementation details	112
5.1.7	Degeneracy analysis	113
5.2	Solving single-view fSfT with non-convex optimization	114
5.2.1	Section overview	114
5.2.2	Problem modeling	115
5.2.3	Cost normalization and weight selection	119
5.2.4	Cost optimization	126
5.2.5	Initialization	129
5.3	Multi-view fSfT	131
5.3.1	Overview	131
5.3.2	Robust focal length averaging	132
5.3.3	Multi-view fSfT optimization	132
5.3.4	Efficient optimization with the Schur complement	133
5.3.5	Instantiation for multi-view fSfT	135
5.4	Experimental results	135
5.4.1	Overview	135
5.4.2	Dataset descriptions	136
5.4.3	Evaluation metrics	141
5.4.4	Single-View fSfT evaluation	142
5.4.5	Results visualizations	159
5.4.6	Multi-view fSfT evaluation	173
5.5	Conclusion	183
6	Conclusions and Future Work	185
6.1	Review of common thesis themes	186
6.1.1	Theme 1: Closed-form solutions using motion model relaxation	186
6.1.2	Theme 2: Theoretical problem analysis from closed-form solutions	187
6.1.3	Theme 3: Solution refinement with non-convex optimization	187
6.2	Future directions of research	188
6.2.1	Plane-based Structure from Motion with Affine cameras	188
6.2.2	Perspective Plane-based Pose Estimation	189
6.2.3	Focal length and Shape-from-Template	191
	Appendices	195

A Appendices	197
A.1 Chapter 1 appendices	197
A.1.1 Published works at EnCoV	197
A.2 Chapter 2 appendices	201
A.2.1 Failure of Zhang’s method with affine motion	201
A.2.2 Decomposing an affine camera projection matrix with Equation (2.7)	202
A.3 Chapter 3 appendices	202
A.3.1 2D Affine scene reconstruction from point correspondences with missing data	202
A.3.2 Proof of Theorem 1	202
A.3.3 Proof of Theorem 2	207
A.3.4 Proof of Theorems 3 to 8	209
A.4 Chapter 4 appendices	211
A.4.1 Proofs of theorems	211
Bibliography	217

Acronyms, Abbreviations and Notation Guide

General acronyms and abbreviations

AR *Augmented Reality*

ARAP *As-Rigid-As-Possible*

BA *Bundle Adjustment*

CAS *Computer Assisted Surgery*

CAD *Computer Aided Design*

CNN *Convolutional Neural Networks*

CT *Computed Tomography*

DoF *Degree-of-Freedom*

FEM *Finite Element Method*

GN *Gauss-Newton*

GT *Ground-truth*

HD *Homography Decomposition*

HE *Homography Estimation*

ICP *Iterative Closest Point*

IID *Independent and Identically Distributed*

IRLS *Iteratively Reweighted Least-Squares*

LLS *Linear Least-Squares*

LM *Levenberg-Marquardt*

MDH *Maximum Depth Heuristic*

MIS *Minimally Invasive Surgery*

ML *Maximum Likelihood*

MRI *Magnetic Resonance Imaging*

MVS *Multi-View Stereo*

NADA *Non-Artificially Degenerate Algorithm*

P, PP, WP and O Abbreviations for the perspective, para-perspective, weak-perspective, and orthographic camera models respectively. We use WP synonymously with the Scaled Orthographic camera.

PDE *Partial Differential Equation*

PHD *Perspective Homography Decomposition*

POPL *Point Of Perspective Linearization*

RANSAC *RANdom SAmples Consensus*

SDP *Semi-Definite Programming*

SfS *Shape-from-Shading*

SLAM *Simultaneous Localization and Mapping*

SOCP *Second-Order Cone Programming*

SVD *Singular Value Decomposition*

TPS *Thin-Plate Splines*

wrt *with respect to*

Glossary of monocular reconstruction problems

The following is a list of monocular reconstruction problems that are relevant to this thesis. For each problem, we include the abbreviation used in this thesis, a brief description, and we also give representative references of algorithms that have tackled the problem from the research literature. These problems are all connected, which is discussed and illustrated graphically in §2.2.

PnP *Perspective-n-Point* [[DD92](#); [Nis03](#); [WH05](#); [LMF09](#); [LXX12](#); [HR11](#); [Zhe+13](#); [KLS14](#); [Nak15](#); [Wie+18](#); [Nak16](#)]. The problem of estimating the pose of an intrinsically calibrated perspective camera relative to a known rigid structure using a general number n of non-collinear point correspondences with $n \geq 3$. The problem is also called perspective camera resection or absolute pose estimation.

P3P *Perspective-3-Point* [[DRL89](#); [FB81](#); [Gao+03](#); [Har+91](#); [QL99](#); [Har+94](#); [KR17](#); [Gru41](#)]. This specializes PnP with 3 points.

P4P *Perspective-4-Point* [[Hor+89](#); [QL99](#); [Tri99b](#); [WH05](#)]. This specializes PnP with 4 points.

-
- PE-X** *Pose Estimation from n points with camera model X .* The problem of estimating the pose of a camera relative to a known rigid structure using $n \geq 3$ non-colinear point correspondences. The camera model is denoted by ‘ X ’. PE-P indicates using an intrinsically calibrated perspective camera and it is synonymous with PnP. Published algorithms exist for para-perspective (PE-PP) [Hor+97; BCP15], Weak-perspective (PE-WP) [DD95; Hor+97; BCP15] and orthographic (PE-O) [BCP15; Ste18]. In PE-P, there are 6 unknown camera DoFs per image (camera poses). In PE-PP, PE-WP and PE-O there are 8, 6 and 5 unknown camera DoFs per image. We use PE-A to denote solving PE with an unspecified affine camera. PE with generalized cameras have also been considered with known and non-parallel point rays [CC04; Nis04a; SP08; KLS14]
- PPE-X** *Plane-based Pose Estimation from n points.* This specializes PE-X with co-planar points. PPE-X can be solved with a PE-X algorithm that handles co-planar points as a special case, or with an algorithm that is dedicated to the case of co-planar points. Dedicated algorithms exist for PPE-P [Stu00; Zha00; CB14a], PE-PP [Hor+97; BC18], PE-WP [ODD96; Hor+97; BC18] and PPE-O [CB17; BC18; CB17; Ste18].
- UPPE-P** *Uncalibrated Plane-based Pose Estimation from n points with a perspective camera* [Tsa87; SM99; Zha00; HS97; Bou00; Zha16]. This relaxes PPE-P where one or more of the camera intrinsics are unknown *a priori*. This problem’s main application is camera calibration using a planar calibration target.
- SfM** *Structure-from-Motion.* The problem of reconstructing a rigid scene from multiple monocular images using motion. Reconstruction generally requires estimating the scene’s metric structure and camera poses.
- SfM-X** *Structure-from-Motion using camera model X .* Published algorithms exist for SfM-P (known intrinsics) [Lon87; Tri+00; HZ04; Özy+17], SfM-PP [KM98; Qua94], SfM-WP [KM98; Qua94] and SfM-O [Ull79; KM98; TK92; Qua94; Ull79; HL89; HB86; TJK10; MC09].
- USfM-P** *Uncalibrated Structure-from-Motion with a perspective camera* [FLM92; Tri97; Fau95; PG99; LV96; PKG98]. This relaxes SfM-P when one or more camera intrinsics are unknown (also known as auto- or self calibration).
- PSfM-X** *Plane-based Structure-from-Motion using camera model X .* This specializes SfM-X when structure is co-planar. Published algorithms exist for PSfM-P [FL88; ZH96; MV07] and PSfM-O [TJK10; HL89; HB86; TJK10; CB17]. PSfM-PP and PSfM-WP are unsolvable as explained in §3.1.2.
- UPSfM-P** *Uncalibrated Plane-based Structure-from-Motion* [MC02; Tri98; HZ04; GS03; Men+08]. This relaxes PSfM-P where one or more camera intrinsics are unknown.
- SfT** *Shape-from-Template.* The problem of registering and reconstructing a deformable 3D surface from one or more monocular images using an isometric or quasi-isometric template that can deform with limited stretching or shrinking. SfT generalizes PE to deformable objects.
- SfT-X** *SfT using camera model X .* Published algorithms exist for SfT-P (known intrinsics) [SHF07; PHB11; CPB14a; Bar+15; CB15; Yu+15; Par+15; SF09; Mag+15; Bru+10; Liu+16b; MBC12a; Ost+12; MBC11; SUF08; Pum+18; Gol+18; Fue+18; Fue+21]. SfT-PP [Bar+15], SfT-WP [BC13b; BPC13; PBC13] and SfT-O [TJK10; CB10b].

- USfT-P** *Uncalibrated Shape-from-Template* [BC13b; BPC13]. This relaxes SfT-P where one or more camera intrinsics are unknown *a priori*.
- fSfT** *Focal length and Shape-from-Template* [BC13b; BPC13]. This specializes USfT-P where the camera’s focal length is unknown and constant in a set of images. We do not include the suffix ‘P’ in fSfT because fSfT implies using a perspective camera.
- NRSfM** *Non-Rigid Structure-from-Motion*. The problem of registering and reconstructing a deformable structure from multiple monocular images using motion. Unlike SfT, in NRSfM there is no template and therefore a 3D reference shape of the structure is not known *a priori*. NRSfM generalizes SfM to deformable objects.
- IsoSfM** *Isometric Structure-from-Motion*. IsoSfM specializes NRSfM where the structure deforms isometrically or quasi-isometrically (limited stretching or shrinking).
- NRSfM-X** *NRSfM using camera model X*. Published algorithms exist for NRSfM-P (known intrinsics) [Agu+16; PPB20; VA12; CPB14b; PPB18; RYA14; Var+09b; Chh+16], NRSfM-WP [BC13b; BPC13; PBC13; BHB00; Bra01; DSA07; THB08; Fay+09; Akh+09; GM11; Kum19] and NRSfM-O [VA12; TJK10; CB10a]. NRSfM-PP has not been studied.
- IsoSfM-X** *IsoSfM using camera model X*. Published algorithms exist for IsoSfM-P [PPB20; VA12; CPB14b; PPB18; RYA14; Var+09b] and IsoSfM-O [TJK10; CB10a]. No published algorithms exist for IsoSfM-PP or IsoSfM-WP.
- UIsoSfM-P** *Uncalibrated Isometric Structure-from-Motion* [Pro+18; PBP18]. This relaxes IsoSfM-P where one or more camera intrinsics are unknown *a priori*.
- fIsoSfM** *Focal length and Isometric Structure-from-Motion* [Pro+18; PBP18]. fIsoSfM specializes UIsoSfM-P where focal length is unknown and constant in a set of images.

Mathematical notation guide

Matrices are in upright upper-case bold, vectors are in upright lower-case bold and scalars are in lower-case italic. Sets are in upper-case calligraphic. Table 1 gives the notation guide.

Symbol	Meaning
\mathbf{v}_k	k^{th} element of a vector \mathbf{v}
\mathbf{A}_{ij}	Element at row i , column k of a matrix \mathbf{A}
$[\mathbf{A}]_{K \times L}$	Top-left $K \times L$ sub-matrix of \mathbf{A}
$\mathbf{0}_{K \times L}$	$K \times L$ matrix of all-zeros
$\mathbf{1}_{K \times L}$	$K \times L$ matrix of all-ones
$\mathbf{I}_{K \times K}$	$K \times K$ identity matrix
$\mathbf{I}_{K \times L}$	$K \times L$ top-left sub-matrix of $\mathbf{I}_{\max(K,L) \times \max(K,L)}$
$\hat{\mathbf{A}}$	Estimate of a matrix \mathbf{A} from noisy measurements
$G(\mathbf{A})$	Gramian of a matrix \mathbf{A} : $G(\mathbf{A}) = \mathbf{A}^\top \mathbf{A}$
$\text{stk}(\mathbf{A}_1, \dots, \mathbf{A}_M)$	Row-wise stacking of matrices $(\mathbf{A}_1, \dots, \mathbf{A}_M)$: $\text{stk}(\mathbf{A}_1, \dots, \mathbf{A}_M) = [\mathbf{A}_1^\top, \dots, \mathbf{A}_M^\top]^\top$
$\text{vec}(\mathbf{A})$	Conversion of matrix \mathbf{A} to a column-vector by stacking its elements in column order
$s_k(\mathbf{A})$	k^{th} largest singular value of \mathbf{A}
$\lambda_k(\mathbf{A})$	k^{th} largest eigenvalue of a positive semidefinite matrix \mathbf{A}
$\mathcal{V}_k(\mathbf{P})$	Set of unit eigenvectors of \mathbf{P} with eigenvalue $\lambda_k(\mathbf{P})$
$\mathcal{S}_{2 \times 3}$	Stiefel group of dimensions 2×3 ($\mathbf{M} \in \mathcal{S}_{2 \times 3} \Leftrightarrow \mathbf{M}\mathbf{M}^\top = \mathbf{I}_2$)
$\mathcal{SS}_{2 \times 2}$	Sub-Stiefel group of dimensions 2×2 ($\mathbf{A} \in \mathcal{SS}_{2 \times 2} \Leftrightarrow \exists \mathbf{M} \in \mathcal{S}_{2 \times 3}$ s.t. $\mathbf{A} = [\mathbf{M}]_{2 \times 2}$)
$\mathcal{G}_{2 \times 2}$	Gramian of $\mathcal{SS}_{2 \times 2}$ ($\mathbf{G} \in \mathcal{G}_{2 \times 2} \Leftrightarrow \exists \mathbf{A} \in \mathcal{SS}_{2 \times 2}$ such that $\mathbf{A}^\top \mathbf{A} = \mathbf{G}$)
SO_n	Special orthogonal group of dimension n
SE_n	Special Euclidean group of dimension s

Table 1: Symbol definitions and notation guide

Chapter 1

Introduction and Thesis Contributions

Chapter summary and organization

This chapter presents the thesis context from a personal standpoint and within the field of Computer vision, in particular 3D reconstruction using motion information in monocular images. We introduce three closely related reconstruction problems that are studied and solved in this thesis. These are (i) Structure from Motion with planar structures, (ii) pose estimation with planar structures and (iii) reconstruction and camera calibration with deformable surfaces. We give general background, applications, open challenges of each problem and we summarize the main technical and theoretical contributions that have been achieved for each problem in this thesis. We list the peer-reviewed publications originating from this work and we summarize the subsequent thesis chapters.

1.1 Personal context

The work of this thesis represents a selection of research that I, Toby Collins, conducted while at the Endoscopy and Computer Vision (EnCoV) group, where I worked between October 2009 and October 2016. I was initially contracted as a Masters-level computer vision researcher where in the first few years I built up my research skills in the lab’s two central research areas: fundamental 3D computer vision and computer-assisted surgery with computer vision. My research output grew over time and my duties broadened to supervising Masters projects, supporting the research of several EnCoV Ph.D students on the ERC Project Flexible, and then to co-supervise an EnCoV Ph.D student on this grant with Prof. Bartoli (Mathias Gallardo, graduated in 2018). I left EnCoV in October 2016 to join the Institut de Recherche contre les Cancers de l’Appareil Digestif (IRCAD) to continue my research focusing exclusively on computer-assisted surgery and clinical translation. I continued my collaboration with Prof. Bartoli in this area by co-supervising another EnCoV Ph.D student (Richard Modrzejewski, graduated in 2020). I co-authored 41 papers during my time at EnCoV in many of the top computer vision and medical imaging conferences and journals, which are listed in Appendix §A.1.1. Among these works, a selection of my research was used to create this document, which is a regular Ph.D thesis composed of three contribution chapters. We decided to focus this thesis on fundamental computer vision research and in particular monocular 3D reconstruction of rigid and deformable surfaces. This restricted the scope and quantity to a standard thesis while also forming a cohesive and self-contained body of work. This thesis should therefore be judged based solely on its contents as a regular thesis with three contribution chapters, without considering my other research.

1.2 Computer vision context

1.2.1 Monocular reconstruction

Computer vision is a very active and broad research area in computer science whose general goal is to develop computational systems and a theory for interpreting image and video data. Computer vision intersects various disciplines, in particular computational geometry, machine learning, probability theory, statistics, physics, signal processing, robotics, computer graphics and optimization theory. It is also composed of several specialisms that focus on different vision problems, such as object detection, image segmentation, object tracking, image retrieval, image registration and reconstruction. Significant advances in recent years have led to computer vision algorithms permeating deeply within our society, ranging from the recognition of human faces in online social media and urban surveillance systems, reconstruction of cities in 3D from satellite images for GPS navigation systems, vision-based control of robots and drones, to enabling Augmented Reality (AR) applications running on smartphones and automatic recognition of pre-cancerous lesions in medical exams such as colonoscopy.

Of the many goals in computer vision, reconstructing 3D geometry from monocular images has been one of the central areas of study for well over four decades, and it has many open challenges to this day. The general task is to reconstruct 3D entities visible to the camera, such as points, curves, lines, depth-maps or surfaces, and reconstruct the 3D motion of these entities with respect to the camera from its images. Because camera projection represents a loss of depth information, reconstruction can only be solved by combining image data with prior knowledge. This may include knowledge about camera projection, the relationship between camera movement and image movement through multi-view projective geometry, knowing if the structure is rigid, or photometric knowledge

that connects pixel brightness and surface reflectance to 3D shape. This knowledge is represented in one of two main ways: either using hand-designed geometric or photometric models, or represented with general-purpose machine learning models such as CNNs trained from examples.

Reconstruction problems have been approached in many ways over the years with a very wide range of techniques and problem formulations. There exists no universal reconstruction method that solves monocular reconstruction in all cases. Instead, a taxonomy of problems and approaches has expanded over the years that encompasses different problem settings. One of the most extensively studied settings is reconstruction exclusively from motion data. This in turn has many problem flavors depending on factors such the available motion information (*e.g.* sparse or dense motion), the correctness of the motion information, whether or not the camera is intrinsically calibrated, and available prior knowledge. The many possible combinations has lead to a vast amount of research, algorithms and literature base. *Structure-from-Motion* (SfM) is one of the major categories that uses the assumption of scene rigidity. The task of SfM is to simultaneously reconstruct the scene’s metric structure and the poses of the cameras viewing the structure from image motion. There now exists various mature SfM algorithms that have translated into commercial products, and progress has been coupled with a deep theoretical understanding [HZ04; FLP01]. Unlike SfM, reconstruction of rigid objects whose structure is known *a priori* can be achieved with a single image, where the goal is to estimate the 3D pose of the object from the camera’s viewpoint, known as *pose estimation*¹.

Reconstruction of deformable structures is also a major and open research area in computer vision that has been extensively studied over the decades. It also has a broad range of applications including AR, deformation capture, 3D model building, video post-production and computer-assisted surgery (CAS). However, it is generally much harder compared to rigid reconstruction, because of the significantly larger search space and significantly weaker geometric constraints. It is also much harder to perform theoretical analysis with deformable reconstruction problems, such as determining problem well-posedness, critical configurations and geometric ambiguities. Nevertheless, much progress has been made in deformable reconstruction in spite of a sparse theoretical analysis.

To make deformable reconstruction well-posed, the prior knowledge used in many works, including this thesis, is embodied in a *template*. The template provides a geometric model of the object (often called the *reference shape*), and it also constrains how the object can physically deform from its reference shape. The reference shape can be acquired by various means. For example, a computer assisted design (CAD) model, a 3D scanner, or from dense multi-view stereo (MVS) with a monocular camera viewing the object at rest. The physical constraints are generally implemented using models from, or inspired by mechanics such as mass-spring systems or finite element models (FEMs), The approach of solving monocular deformable reconstruction with a template is often called *Shape-from-Template* (SfT) in the literature or equivalently *template-based monocular deformable reconstruction*. We use SfT in this thesis. Similarly to rigid reconstruction, SfT has been studied with several different problem formulations, including different camera models, using single image or video inputs, different visual cues (motion, shading, and contours in particular) and different physical deformation models. The literature on SfT has many proposed algorithms, yet SfT is not considered a solved problem because of the inherent challenges in both modeling the problem and solving it.

¹‘Pose estimation’ is also commonly used to mean estimating the non-rigid pose of a human body. We specifically use pose estimation to mean estimating the pose of a rigid structure in this thesis. ‘Absolute pose’ is also commonly used and it is synonymous with pose estimation of a rigid structure.

1.2.2 Reconstruction and image registration

There is an intimate relationship between monocular reconstruction problems and *image registration*, which itself is a fundamental and large topic in computer vision with many open challenges. Image registration is about determining the spatial alignment of one or more structures of interest in images. It connects with reconstruction in two ways: Firstly, it supplies the motion data required by a reconstruction algorithm. Secondly, the reconstruction algorithm necessarily makes assumptions about the scene as prior knowledge, and this in turn constrains registration. Therefore there is a two-way relationship between reconstruction and registration. A classical example where this relationship is exploited is to estimate the relative pose of an intrinsically calibrated camera from two views of a rigid scene. The fact that the scene is rigid imposes constraints on registration expressed by the epipolar geometry and the essential matrix [Lon81; Nis04b]. This is exploited using Random Sampling and Consensus (RANSAC) [FB81]. In the classical pipeline, first an imperfect registration is computed by matching a sparse set of points in each image using texture similarity. Matching is never usually perfect because of incorrectly matched points with ambiguous texture. Registration has therefore been achieved for a subset of matches, but we do not know the sub-set, hence we do not know the registration. This is resolved by cycling between registration and reconstruction: In each cycle a small set of matches is randomly selected, a reconstruction hypothesis is made from them by estimating the essential matrix (equivalent to estimating the relative camera pose), then the hypothesis is tested by how well it can register the other matches. The process continues until a reconstruction and registration are found with high likelihood. RANSAC is one famous example of many approaches where registration and reconstruction feed into one another.

In deformable reconstruction with SfT, the overarching goal is to achieve both reconstruction (*i.e.* estimating the template’s deformed 3D shape) and registration (estimating the spatial alignment between the template and the image). This is required for most SfT applications such as AR and discussed in §1.3.3.1.

1.2.3 Geometry-based algorithms and the era of deep learning

The algorithms presented in this thesis are categorized as geometry-based 3D computer vision algorithms. Since the publication of the works in this thesis, deep learning has emerged as an important alternative approach to 3D computer vision. Deep learning found its first successes in semantic computer vision tasks such as image classification, image segmentation and object detection. It has produced unprecedented results in all those tasks, and in the past 5 years it has also shown great potential for solving 3D vision tasks that have not traditionally been considered direct applications of machine learning. These include monocular depth estimation [EPF14; Min+21], stereo disparity estimation [ZL15; Zho+20], camera pose [KGC15; SF19], camera relative pose [Mel+17b; SF19] and human body pose [Cao+19; Wan+21] estimation. Camera calibration [Bog+18; Hol+18], SfT [Pum+18; Gol+18; Fue+21], SfM with independent depth and pose estimation networks [Zho+17b; YS18; Ran+19] or with a combined network [Demon; TT19; TD20], and NRSfM [KL19; Sid+20; KL20] have also been tackled with deep learning. We now give some perspectives about these developments and discuss why geometry-based approaches, including our work in this thesis, remain important today and into the future. We also discuss how the combination of geometry and deep learning can offer us the advantages of both approaches.

Geometry-based approaches Geometry-based approaches to 3D vision are characterized by the use of geometric abstractions (or geometric primitives) such as points, lines, planes, mesh faces and surface patches. These approaches assemble and solve geometric equations that relate unknown properties (typically 3D position, displacement or deformation) with geometric features extracted from images. Geometry-based approaches have a very long history in computer vision dating back to the 1960s, and they are popular for 3D vision because they have compact and intuitive representations for modeling, synthesizing, compressing, registering and reconstructing physical structures.

Geometry-based approaches may incorporate elements of probability and machine learning because features extracted from images necessarily carry uncertainty. This can originate from image noise, quantization, the aperture problem and geometric approximation. In practice, often very simple and general uncertainty models are sufficient. For example, point position uncertainty of keypoints such as SIFT [Low04a] can be well approximated with IID Gaussian additive noise, which has been shown to work well in countless research works such as bundle adjustment [Tri+00]. For problems that involve only reprojection constraints (typically when reconstructing rigid structures), noise standard deviation drops out of the equations, and the optimal solution simply minimizes the L2 reprojection error. Consequently, the only hyper-parameter of the noise model (standard deviation) becomes irrelevant. This ability to use a general noise model with little or no hyper-parameter estimation is an important advantage of geometry-based approaches. In contrast, for problems involving deformable reconstruction, the required deformation prior may either be learned from data (statistical deformation modeling) or based on knowledge of the surface material (physical deformation modeling as in this thesis). In both cases, a hyper-parameter is often used to balance the relative influence of the deformation prior to avoid fitting to noise, which requires estimation. This may either be estimated manually or learned from training data. Unlike end-to-end deep learning, the number of hyper-parameters to tune is typically very small when the geometry-based method is designed well, so the amount of required training data can be very small.

In general, there are five important characteristics of geometry-based approaches that are not available in end-to-end deep learning-based approaches: *(i)* Geometry-based approaches can provide simple and explainable algorithms that are often computationally inexpensive, especially for problems that admit closed-form solutions (which includes all problems studied in this thesis). *(ii)* Geometry-based approaches are excellent tools to analyze theoretical aspects of a problem, and analyze algorithms for solving that problem. Analysis includes establishing minimal cases, solution multiplicity, geometric conditions that are necessary or sufficient for well-posedness, and problem/algorithm stability as a function of scene geometry or noise. Such analysis is fundamental to predict or diagnose failures in real-world applications. *(iii)* Geometry-based approaches can give mathematically rigorous algorithms with formal guarantees, including a guarantee of returning all solutions that explain the data. *(iv)* Geometry-based approaches decouple geometric feature extraction from down-stream geometric reasoning, benefiting from problem ‘divide-and-conquer’. A variety of geometric feature extraction algorithms can be used, and better algorithms can be swapped into the pipeline without making significant alterations. *(v)* Many geometry-based approaches are very general and work for a wide range of applications because they do not depend heavily on training data for learning complex functions or probability distributions.

End-to-end deep learning-based approaches In recent years, the field of computer vision has been revolutionized by the success of deep learning and in particular Convolutional Neural Networks (CNNs). CNNs are artificial neural networks consisting in convolutional layers that compute trans-

lational equivariant outputs known as feature maps using learnable convolutional filters. CNNs have been around since the late 90s when they were mainly used for image classification and in particular handwritten character recognition [LeC+98]. However, it took many years for CNNs to gain wide-spread popularity. One of the main catalysts was AlexNet [KSH12] in 2012; a CNN for image classification that considerably outperformed ‘shallow’ methods using hand-crafted features on the ImageNet Large Scale Visual Recognition Challenge [Rus+15]. Since then, deep learning and CNNs have been applied to a vast array of vision tasks and in many cases with great success. There are five main reasons for this. (i) The availability of large-scale training datasets, required to learn useful deep features. (ii) The availability of open source deep learning frameworks, in particular Tensorflow from Google and Pytorch from Facebook, that provide high-level programming abstractions which make it simple to create, train, adapt, share and fine-tuning deep neural networks in Python. (iii) The wide availability of consumer-grade graphics cards capable of fast neural network training and inference on standard workstation computers and low-cost devices. (iv) The success of training deep neural networks with backpropagation using batch or stochastic gradient descent often without needing careful initialization [Du+19]. (v) The maturation of good neural network design principles and components that work well in many problems. For example, rectified linear units (ReLUs), batch normalization, residual skip connections, encoder / decoder architectures and Generative Adversarial Networks (GANs) [Goo+14]. The combination of these five factors have generated a sweeping revolution in computer vision in recent years.

All the 3D vision problems mentioned above have been tackled with an ‘end-to-end’ deep learning approach. The idea is to train a deep neural network to approximate the function that maps problem inputs (images or videos) to outputs such as camera poses or depth-maps. The principal advantages of end-to-end deep learning are three-fold: (a) It eliminates human effort for designing hand-crafted features. (b) It reduces inductive biases associated with sub-optimal hand-crafted features. (c) It does not involve solving an optimization problem at run-time (it requires only a forward-pass of the network, which can be especially fast for CNNs using conventional graphics cards). However, end-to-end learning approaches carry some important disadvantages. Firstly, they are strongly dependent on adequate and representative training data. The first end-to-end 3D vision approaches were trained with supervised learning, requiring training data with known outputs (labels). This is often severely limiting because of the difficulty in obtaining labels. For example, PoseNet [KGC15] was the first demonstration of end-to-end deep learning for camera pose estimation. To account for the lack of ground truth camera poses, the outputs from a geometry-based SfM method were used as labels. SfT has been tackled with supervised deep end-to-end learning [Pum+18; Gol+18] where labeled data was simulated by open source animation software (Blender). However, these methods did not generalize well to real data (the so-called ‘render gap’) as shown in [Fue+21] (work co-authored by myself). Furthermore, [Pum+18; Gol+18] required training the deep neural network for each object, which severely reduced their practical value. Very recently, an end-to-end SfT method has been proposed that can handle different objects with the same network weights [Fue+21]. However, the object templates must be flat and rectangular. Thus, they do not yet match the generality of a carefully designed geometry-based SfT method such as [CB15] (work co-authored by myself).

Combining geometry and deep learning-based methods and thesis extensions To overcome the challenges of supervised learning, there has been significant success in training end-to-end deep neural networks using self-supervised learning, and specifically using image synthesis as a supervision signal. One of the seminal works was [Gar+16a] that trained a monocular depth estimation

network using rectified stereo images. The loss function measured the photometric dissimilarity between the true right image and a synthesized right image using the left image and its predicted depth map. There have been many following works in 3D vision that use image synthesis or other geometric cues such as left-right consistency to train deep neural networks without ground truth, including monocular [GAB17] and stereo [Zho+17a] depth estimation, camera relative pose estimation [ZC21], small-scale dense SfM [Zho+17b] and very recently NRSfM [Sid+20]. These approaches marks a big step forward towards making deep learning-based approaches practical and salable, and they are only possible using the mature knowledge of multi-view geometry gained by geometry-based methods. Neither SfT nor fSfT have been attempted with end-to-end self-supervised deep learning. This may be possible but it would require a carefully designed cost function that works well in general (*i.e.* with a strong minimum close to the true solution and limited hyper-parameter tuning). We believe that the work in Chapter 5, which focuses on designing such a cost function using weight normalization and automatic (unsupervised) hyper-parameter estimation could open up this possibility. Therefore, innovations in a geometry-based approach have the additional value of insights and tools for training deep neural networks with self-supervised learning.

In addition to training data acquisition, another major limitation of end-to-end approaches is that they involve networks with a vast number of weights (often in the millions), making them extremely hard to interpret. Fundamentally, they do not have any of the 5 desirable characteristics of geometry-based approaches outlined above. A promising direction that has gained momentum over the past few years is to combine deep learning for feature extraction with geometry-based methods for reconstruction. Some of the first works were for camera pose estimation [TSF18]. A deep neural network predicted point correspondences and pose was then estimated with a PnP method (EPnP) [LMF09]. The deep neural network was superior to classical methods such as SIFT, especially for poorly textured objects, while the PnP method allowed fast, closed-form pose estimation. EPnP does not handle well the two-fold pose ambiguity associated with planar structures (Chapter 4). Consequently, in such cases, EPnP could be swapped for IPPE described in Chapter 4, which correctly handles the pose ambiguity and is generally more accurate than EPnP as shown in Chapter 4. In contrast, to the best of our knowledge, no end-to-end approach is capable of finding all camera pose solutions in ambiguous cases, because they can only output one solution. Deep learning has also been combined with geometry-based methods for other reconstruction tasks with great progress, in particular SfM and SLAM, where research is extremely active [Che+20]. For example, CNN-SLAM [Tat+17] uses depth maps regressed from a deep neural network to input into an existing geometry-based monocular SLAM framework (LSD-SLAM [ESC14]), which resulted in better scene reconstructions and reduced scale drift. In SLAM, hybrid approaches [Yin+17; Bar+18; Zha+20] for solving visual odometry (camera relative pose estimation in video) that combine traditional geometry-based reconstruction with geometric data (including camera pose sets and depth-maps) from deep neural networks. These hybrid methods have been shown to outperform end-to-end visual odometry approaches especially to avoid scale drift [Che+20].

We emphasize that all the geometry-based methods presented in this thesis take image point matches as inputs. Thus, they could all be used with more recent sparse or dense deep learning-based image matching [DMR18; Dus+19; Ono+18; Rev+19; Yi+16] as a hybrid approach, and we hope to investigate the potential benefits in future work.

1.3 Problems tackled in this thesis, motivation, and contribution summary

This thesis advances state-of-the-art in three related and fundamental reconstruction and registration problems involving rigid and deformable structures. The contributions to each problem are provided in three dedicated chapters. We now summarize each problem, its applications, the open challenges and our scientific contributions. A more in-depth review of prior state-of-the-art methods for each problems, and the fundamental connections between the problems is provided in Chapter 2.

1.3.1 Chapter 3: Plane-based Structure-from-Motion with Affine cameras (PSfM-A)

1.3.1.1 Problem summary and applications

SfM is the problem of simultaneously determining a rigid scene’s metric structure and camera poses from motion information in the camera images. SfM has many well-known applications including 3D modeling, photogrammetry and environment mapping and navigation with robots or drones equipped with a monocular camera. Plane-based Structure-from-Motion (PSfM) is the special case of SfM when the object or scene to be reconstructed is planar or quasi-planar, such as a tabletop surface or a section of relatively flat land in aerial photography. PSfM has also been applied for solving Non-rigid Structure-from-Motion (NRSfM) using locally planar surface approximations [TJK10] REF. This is done by estimating the motion of local surface regions, then for each region, PSfM is used to reconstruct the region’s local geometry (depth and surface normal). From this information the full surface shape can be estimated by fusing local reconstructions. An accurate PSfM solution is a key component of this approach. We illustrate these applications in Figure 1.1.

PSfM with a calibrated perspective camera (PSfM-P) is well-understood. It can be solved in closed-form with two solutions in general from two images, by decomposing the homography matrix that registers the images [FL88; ZH96; MV07]. A third image is required to obtain a unique solution. However, when the structure is small and/or far from the camera, the perspective effects diminish and projection becomes affine with parallel projection rays. In these situations, which are common in practice with small structures, PSfM-P becomes ill-posed and all PSfM-P methods fail.

To meet this challenge, we present a stable alternative to PSfM using an affine camera (PSfM-A) that does not suffer this problem. PSfM-A is nevertheless not trivial for several reasons. Firstly, we must restrict the type of affine camera to make it well-posed. Indeed, it is not solvable with weak- or para-perspective cameras (two of the most common types of affine cameras). However, it is solvable with the orthographic camera (PSfM-O) [TJK10; HL89; HB86] up to a discrete number of solutions. There are in general two camera pose solutions per view, which correspond to flipping the structure about the camera’s optical axis. Regarding structure, in the minimal case of three views there can be up to two structure solutions [HL89]. With four or more views, structure can be solved uniquely [TJK10] but a unique solution is not guaranteed.

1.3.1.2 Limitations of prior state-of-the-art and our contributions

The earliest closed-form PSfM-O methods solved the minimal case: three non-co-linear points in three images [HL89; HB86]. However, these methods have very limited practical value because they assume the points are noiseless. The only previous closed-form PSfM-O method that handles noisy points

and a general number of views is [TJK10]. However, this has four strong limitations: (i) it cannot handle the minimal case of three views, (ii) it only handles three points, (iii) it cannot handle cases with more than one structure solution and (iv) it approximates the problem with a linear relaxation, leading to sub-optimal solutions and *artificial degeneracies*. An artificial degeneracy is a case when the problem can be solved in theory, but a method fails to solve it.

Our contributions are both in terms of a practical closed-form method that overcomes all these limits of prior methods, and new theoretical problem analysis. Regarding our method, it is the first general closed-form method to solve PSfM-O with the following characteristics: (i) it handles an arbitrary number of views including the minimal case of three views, (ii) it handles an arbitrary number of points including the minimal case of three points and including views with missing points, (iii) it finds all structure solutions and (iv) it does not make a linear relaxation and therefore it does not suffer from artificial degeneracies caused by the relaxation. The method leads to significantly more accurate results than [TJK10] in the cases when that method can be applied. The code has been released publicly at <https://github.com/tobycollins/PSfM-A> and it is freely available for academic and commercial use.

Our method works in three steps as follows. In the first step, we compute the rank-2 affine reconstruction from point correspondences. In the second step, we find the upgrade matrix that converts the affine reconstruction to a metric reconstruction by optimizing a set of non-convex upgrade constraints in the least-squares sense. This is solved in closed-form by the roots of a univariate degree-seven polynomial. In the third step, we solve camera poses by minimizing the reprojection error with global optimization. Our PSfM-O method does not require initialization, and its results are generally very accurate compared to the Maximum Likelihood (ML) estimate obtained by Bundle Adjustment (BA). While the ML estimate is statistically optimal², it cannot be obtained with guarantees because of non-convexity and an absence of a general closed-form solution.

Regarding theoretical results, there were several open and fundamental questions about PSfM-O that have been answered in this work:

1. *What are the general geometric conditions that determine if PSfM-O can and cannot be solved?*
Our main theoretical result is to give all necessary and sufficient geometric conditions to solve PSfM-O with three or more views and three or more points. With this result, we can now fully characterize when PSfM-O has a finite or infinite number of solutions thanks to the problem’s specific geometry.
2. *What are the geometric conditions for disambiguating structure with extra views (four or more)?*
We give the necessary and sufficient geometric conditions.
3. *Can the ML PSfM-O estimates ever be found in closed form?* We show that in the case of three views with three or more noisy points, the ML estimates can usually be found in closed-form with our method.
4. *What additional prior knowledge can be used to solve PSfM-A with other affine cameras?* We show that problems involving two common camera types (weak- and para-perspective) can be solved in some special cases that are practically relevant with additional scene assumptions, by converting them to PSfM-O problems.

²Unless otherwise stated, we use a reconstruction problem’s ‘ML estimate’ to mean a solution that minimizes the L_2 point correspondence reprojection error. This assumes that image noise is IID Gaussian, which is generally a good assumption that has been demonstrated many times to work well in practice [HK07]

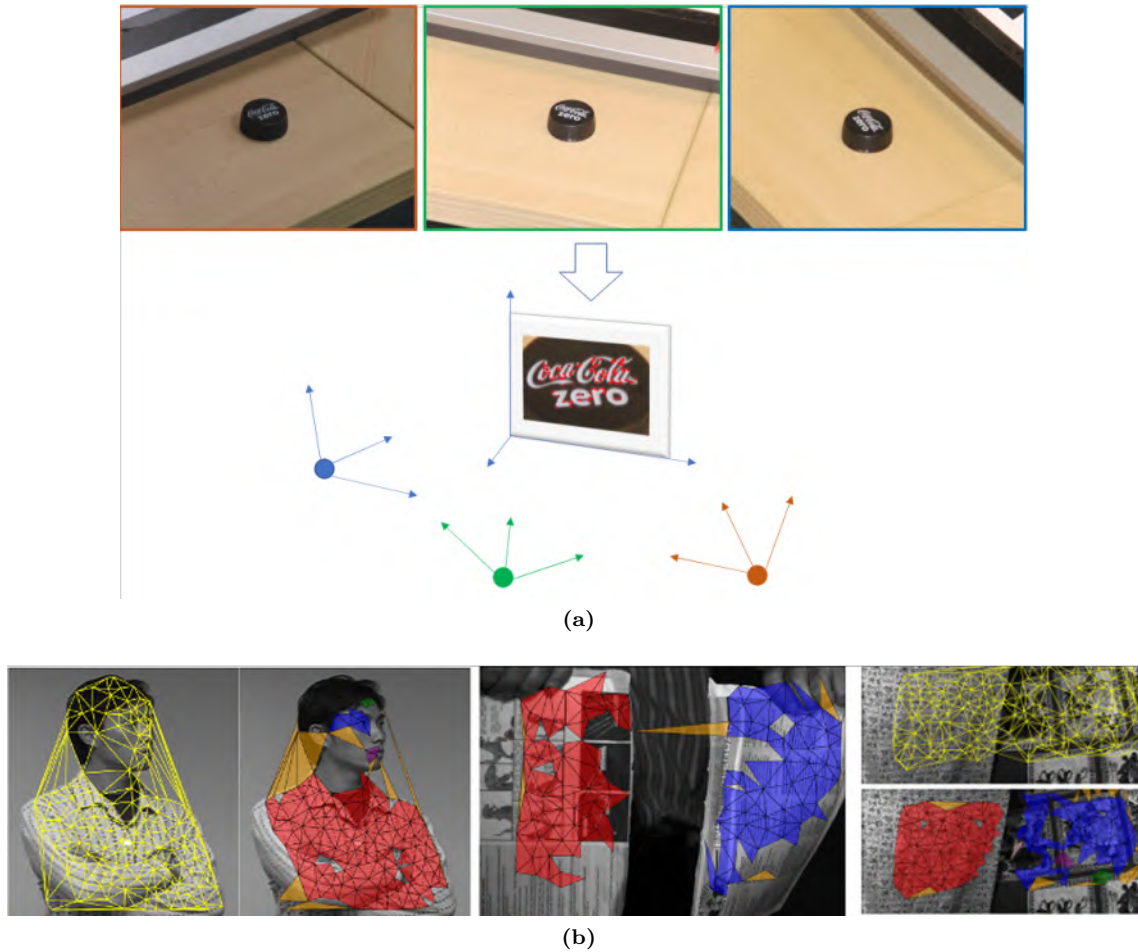


Figure 1.1: PSfM-O and example applications. (a) Camera pose estimation and surface plane rectification: The goal is to estimate camera poses relative to a small planar structure. In this example structure is a bottle top’s surface. This problem cannot be solved with PSfM-P because of the structure’s small size, leading to an ill-posed problem. In contrast, it can be solved with PSfM-O. This example is taken from the experimental section in §3.4.5.2 and the point correspondences used to compute motion are illustrated as red points. (b) Using PSfM-O to solve NRSfM from [TJK10]. Firstly, point correspondences are computed between views, then they are triangulated to create a discrete surface representation. The motion of each triangle is then used to infer the triangle’s 3D orientation in each view with a PSfM-O method. The surface can then be reconstructed up to depth and discrete ambiguities from the triangle orientations. Triangles that do not physically correspond to rigid surface elements are shown in orange. These can be detected automatically because their movement cannot be explained by rigid motion and orthographic projection.

1.3.2 Chapter 4: Plane-based Pose Estimation with a Perspective camera (PPE-P)

1.3.2.1 Problem summary and applications

PPE-P is an old and fundamental problem in computer vision with many applications including AR, camera calibration, 3D object tracking with planar markers, and scene reconstruction. We illustrate several important PPE-P applications in Figures 1.2, 1.3 and 1.4 with extended figure captions describing each application. Many PPE-P algorithms have been presented over the years whose inputs are point correspondences computed between the structure and the image. The PPE-P problem is then equivalent to the Perspective-n-Point problem in the special cases when the object points are co-planar. However, this problem is still open because the ML estimate has no known closed-

form solution (neither for PPE-P nor PnP with general object points). The ML estimate generally produces the most accurate poses using point correspondences in most real-world applications.

There have been two main categories of approaches for solving PPE-P with point correspondences. In the first approach, a closed-form solution is used to solve or optimize an algebraic problem that approximates the ML problem. There are three main kinds of methods in this category. The first kind are analytical methods [Stu00; Zha00], which are extremely fast but they are not very accurate compared to the ML estimate in real-world applications. The second kind are higher-order root finding methods, starting with the DLS algorithm [HR11] and later improved by *e.g.* [Zhe+13] with the OPnP algorithm. These solve general PnP problems including PPE-P as a special case, and they optimize in closed-form the object-space error [ENG07; LHM00]. They are generally much more accurate compared to analytical methods, but they are also much slower by at least 100 orders of magnitude. The third kind of methods solve a lower-order root finding problem, and they aim to strike a balance between computational cost and accuracy. The best prior methods in this category are EPnP [LMF09] and RPnP [LXX12].

In the second main category of approaches for solving PPE-P, gradient-based optimization is performed to optimize the reprojection error directly. The gold standard method is Levenberg-Marquardt (LM), implemented in well-known libraries such as OpenCV’s `solvePnP` method [Bra00]. The main advantage over the closed-form solutions is to achieve the most accurate results in general. Furthermore, despite being iterative, if well initialized it is also much less computationally expensive compared to the closed-form solution using higher-order root finding such as DLS and OPnP. Consequently, they have never gained widespread use for real-time vision applications, unlike LM optimization that is today the most widely-used approach.

The main limitation of gradient-based optimization with *e.g.* LM are (i) the need for a good initialization because of the problem’s non-convexity and (ii) challenges when there are more than one ‘good’ solution with similar reprojection error (solution multiplicity). This can often occur in PPE-P and we discuss it further in §1.3.2.2.

1.3.2.2 Limitations of prior state-of-the-art and our contributions

The fastest way to initialize gradient-based optimization for PPE-P has been the analytical methods of [Stu00] or [Zha00]. However, they have two main limitations: They are highly susceptible to noise especially when the number of points is small. They also become highly unstable when the perspective effects diminish. In practice, this occurs when the structure is small or viewed from a large distance to the camera, and it occurs very frequently in AR marker applications. This is illustrated in Figure 1.5 and described in the figure’s extended caption. In these conditions there can be two valid pose solutions, also known as a flip ambiguity. In such cases, [Stu00] and [Zha00] completely fail to find a single one of these solutions.

We have overcome these limits with our proposed PPE-P method called *Infinitesimal Plane-based Pose estimation* (IPPE). This method has the following desirable characteristics:

1. IPPE is an analytical solution and therefore it is extremely fast. Its computational cost is practically the same as the fastest PPE-P methods of [Stu00] and [Zha00].
2. IPPE is considerably more accurate than previous analytical methods of [Stu00] and [Zha00]. If the object points are arranged in a regular pattern such as the corners of a square, it has very similar accuracy compared to the ML estimate, and there is no clear benefit in refining its

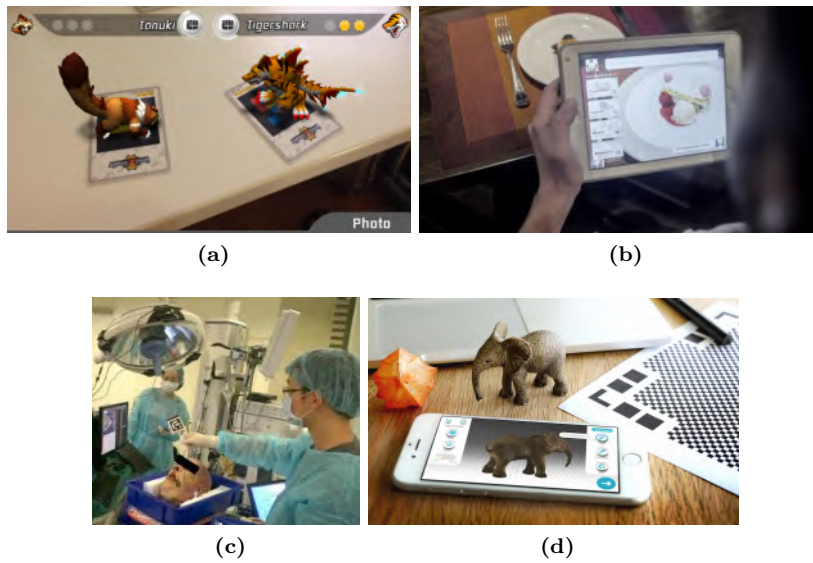


Figure 1.2: Examples of PPE-P applications. (a,b,c) are AR applications where the estimated 3D pose of a planar surface is used to automatically augment the surface with a virtual 3D model. (a) is a gaming application on the Sony PSP gaming console where 3D virtual characters fight on the surface [Nov09], (b) is a dining application where food from a menu is virtually augmented on an empty plate [Kab09]. (c) is an application of PPE-P for surgery instrument tool tracking in 3D using low-cost AR markers [Wan+19]. (d) is an application of PPE-P for scanning an object resting on a planar surface. PPE-P is used to automatically acquire the 3D pose of a smartphone camera relative to the planar surface, and using this information the object can be reconstructed in 3D using dense multi-view stereo [Eye18].

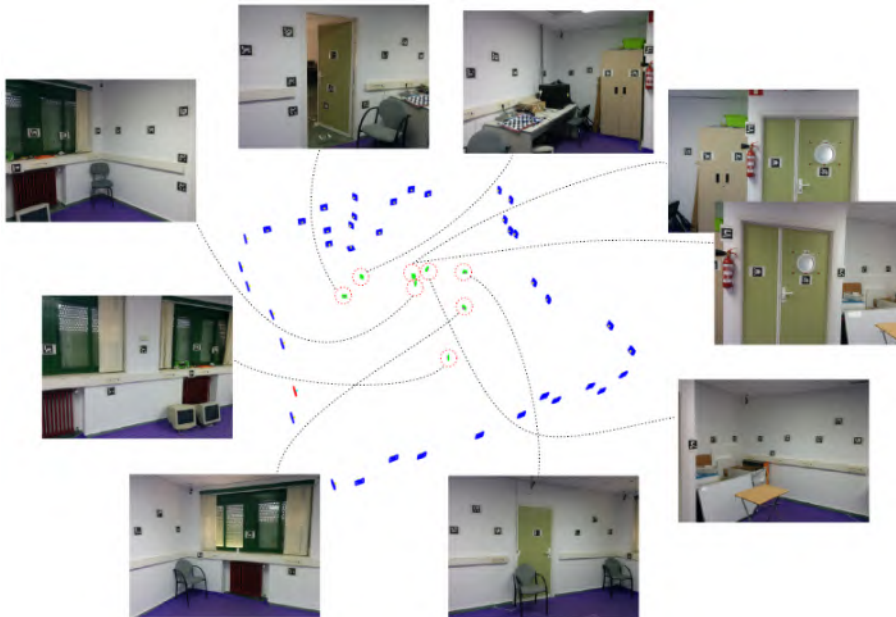


Figure 1.3: PPE-P has applications for 3D Mapping and localization using planar markers [Muñ+18]. The PPE-P method presented in this thesis (IPPE) has been used in [Muñ+18] to provide the pose of each planar marker in multiple images. The pose information is used to initialize a 3D reconstruction of the markers and cameras which is then optimized with bundle adjustment. Quoting [Muñ+18]: “In this work, we have opted for [IPPE] because of its high speed and robustness.”

solution with iterative optimization. This is a very common use case with *e.g.* AR marker pose estimation.

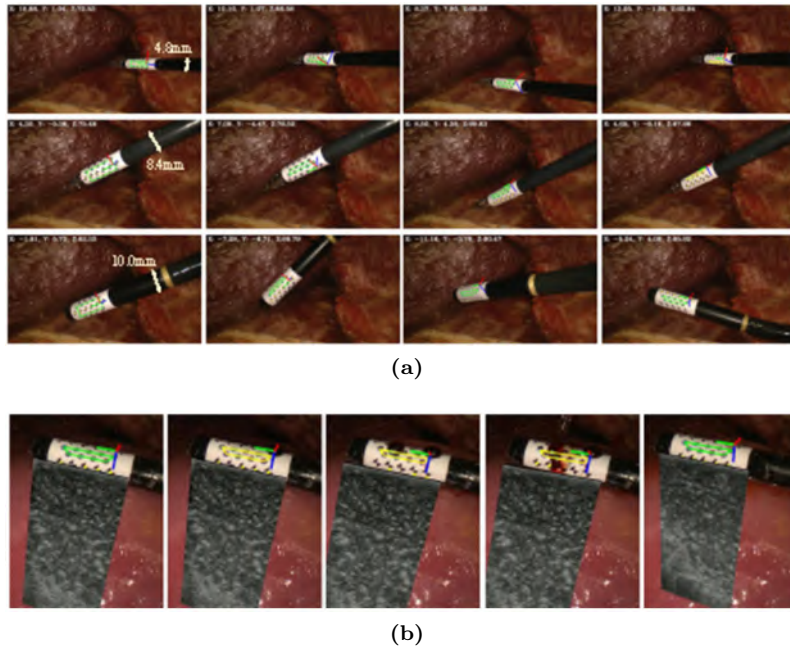


Figure 1.4: PPE-P has applications for 3D tracking of the tip of a surgical instruments in minimally invasive surgery [Zha+17]. (a) The PPE-P method presented in this thesis (IPPE) has been used in [Zha+17] to provide the pose of a marker pattern fixated on the tip of a laparoscopic ultrasound probe. IPPE was used specifically for its robustness in cases when the marker occupies a small region of the image. (b) From the probe’s 3D position, the ultrasound image (gray region) can be automatically overlaid on the laparoscopic images to show the location of hidden critical structures such as vessels or tumors.

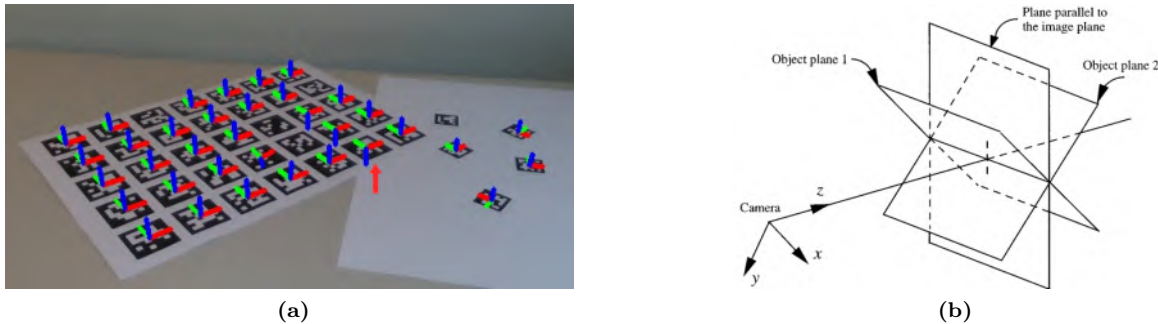


Figure 1.5: Illustration of plane-based pose estimation flip ambiguities. An ArUco marker board [Gar+16b; RMM18] made of 7×5 markers is shown in (a). Poses of each marker in the board have been computed independently using OpenCV’s solvePnP method [Bra00], solved with Levenberg-Marquardt optimization. Poses are visualized using each marker’s three principal axes projected onto the image. The poses of four markers are incorrect, one of which is indicated with the arrow. This is caused because the markers are relatively small and consequently their projection is approximately affine. In these conditions there are two plausible pose solutions indicated in (b) and reproduced from [ODD96]. The term flip ambiguity is used because the two pose solutions correspond to a flip of the object plane about the camera’s image plane. In (a), for one marker the solvePnP method has fallen into a local minimum corresponding to the wrong solution where the surface normal is pointing down instead of up in camera coordinates. In the case of the ArUco marker board it is possible to detect and correct flip ambiguities using the fact that all markers are on a planar board (spatial context). In general however when there is one marker or when markers are placed arbitrarily in a scene, distinguishing the correct pose is not trivial and an open problem.

3. IPPE is stable in ambiguous or quasi-ambiguous cases (flip ambiguities)
4. IPPE has no artificial degeneracies

No state-of-the-art PPE-P method published before IPPE fulfilled all these qualities. We show in Figures 1.3 and 1.4 two example applications from independent research groups who have used IPPE for real-time 3D pose estimation of optical markers [Muñ+18; Zha+17]. IPPE was used specifically for its combination of very high speed and robustness in quasi-affine conditions. The code is freely available for academic and commercial use, and in 2018 it was integrated in the popular OpenCV library at <https://github.com/opencv/opencv>.

1.3.3 Chapter 5: Focal length and Shape-from-Template (fSfT)

1.3.3.1 Problem summary and applications

Shape-from-Template (SfT). SfT is now a well-studied problem in Computer vision [SHF07; BHB14; PHB11; Bar+15; Ost+12; SUF08; Yu+15]. We illustrate SfT in Figure 1.6 using a deforming shoe as an example. A 3D geometrical model of a deformable object is provided in some rest state, called the *template*, which is typically represented by a textured 3D mesh. A 2D camera image is provided observing the object in an unknown deformed state, which may either be an isolated image or a frame from a video sequence. The goal of SfT is to recover the object’s unknown 3D deformation corresponding to each image. This is solved by estimating the deformable 3D transform that maps the template to camera coordinates, giving the depth of the deformed object with respect to the camera and also the registration between the template’s surface and the image. This information allows various applications including AR with deformable objects, human-computer interaction with deforming objects, and AR-guided surgery [Ost+12; CB15; Váv+17]. We illustrate these applications in Figures 1.7, 1.8, 1.9, 1.10 and 1.11 and we describe them in the extended captions.

SfT is ill posed if the template can deform arbitrarily because of the lack of depth information. It is therefore necessary to restrict the space of deformations. In SfT this is done using physical material properties. The most common property is quasi-isometry which prevents significant stretching or shrinking of the surface. This is applicable for many objects of interest, and crucially, the quasi-isometry assumption can guarantee that SfT is well-posed with a calibrated perspective camera [SHF07; Bar+15]. SfT is normally presented as a constraint satisfaction problem that aims to find the 3D deformation that best explains the image data while simultaneously respecting physical deformation constraints.

Shape-from-Template with a calibrated Perspective camera (SfT-P). SfT has been studied extensively with a perspective camera that is fully calibrated [SHF07; BHB14; PHB11; Bar+15; Ost+12; SUF08; Yu+15]. We refer to this as *Shape-from-Template with a Perspective camera* (SfT-P). Calibrated intrinsics are required to relate camera coordinates with image coordinates. However, requiring known intrinsics is an important limitation in many real-world applications. A camera may have fixed and unknown intrinsics, or time-varying and unknown intrinsics *e.g.* when the camera zooms in or out on the object. Neither situation can be handled by SfT-P methods. In the case of fixed and unknown intrinsics, a classical camera calibration is usually performed before SfT-P with a rigid calibration target such as a checkerboard [Zha00]. However, this has several limitations. The calibration process requires user interaction, it may disrupt the application, and a calibration target may not be available. Furthermore, this is only suitable when the camera intrinsics remain fixed after the calibration procedure, which is restrictive.

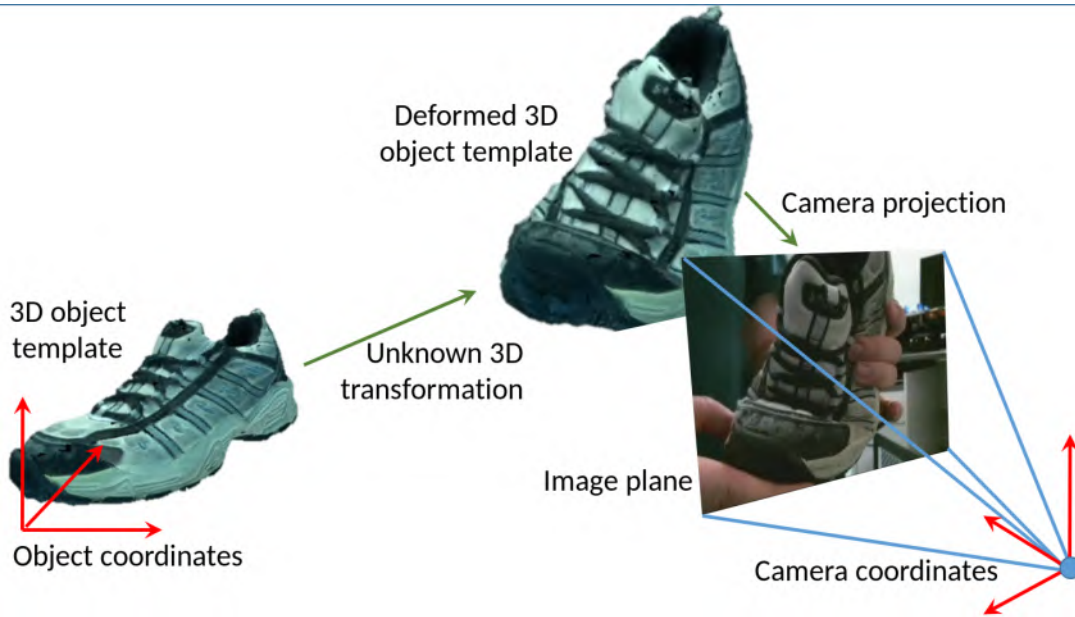


Figure 1.6: The SfT problem illustrated with a shoe being deformed in the hand. A 3D template models the shape and appearance of the shoe in a known rest position. The goal of SfT is to determine the unknown deformable 3D transform that maps the template to camera coordinates, using (i) visual information in the image and (ii) physical constraints from the object’s material properties.



Figure 1.7: An AR SfT application from [Bar+15]. Video frames of a real deforming birthday balloon are augmented with 10 virtual candles. Using SfT, a balloon template is registered and reconstructed from each image. Virtual candles are placed on the template, and thanks to the registration and reconstruction provided by SfT, the candles can be automatically augmented on the images, giving the impression they are physically stuck to the surface and oriented appropriately. Furthermore, the candles can be virtually ignited when the balloon is squeezed sufficiently, thanks to the recovered 3D shape information.

1.3.3.2 Limitations of prior state-of-the-art and our contributions

We describe the first approach to solve SfT with an uncalibrated perspective camera. We restrict this problem to the special case of unknown focal length using a quasi-isometric deformation model. We refer to this as *focal length and Shape-from-Template* (fSfT). The other intrinsic terms are assumed to either be known *a priori* or take standard values. We study fSfT mainly for pragmatic reasons: unknown focal length is a very common use case. Many modern cameras can be modeled accurately with a perspective camera with negligible lens distortion and simplified intrinsics with zero skew, aspect ratio of one and principal point at the image center. The only unknown intrinsics is the focal length that may vary according to optical or digital zoom about the image center. Consequently, solving fSfT covers a broad range of applications. We also study fSfT because unlike camera calibration with a rigid object, the much weaker geometric constraints available with a deformable object may

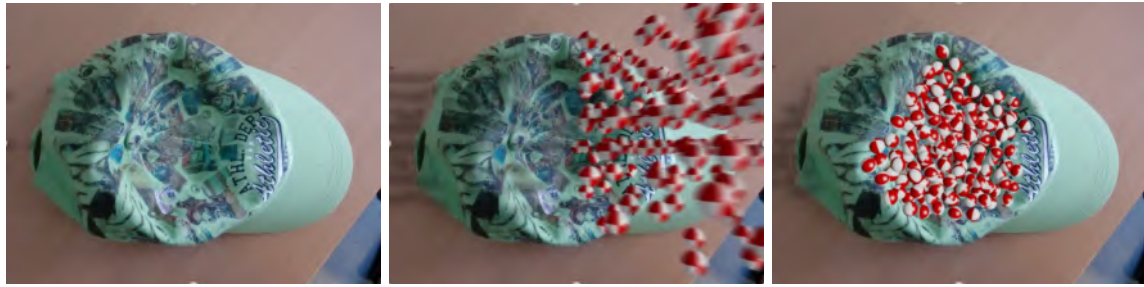


Figure 1.8: An AR SfT application from [Bar+15]. A static image of a deformed cap is augmented with virtual balls being thrown at the cap. Thanks to SfT, the reconstructed cap’s 3D shape can be used in a physics simulation engine involving the cap, table, balls and gravity. The images show three snapshots of the simulation before the balls are thrown (left), immediately after they are thrown (middle) and the final equilibrium state, where some of the balls convincingly pack themselves into the large depression on the cap’s surface.



Figure 1.9: A Human Computer Interaction (HCI) SfT application from [CB15]. The goal is to assist a computer animator to interactively deform a virtual object by manipulating a real, similar deformable object in their hands. In this example the real object is a juice bottle that is viewed by a monocular camera. Its corresponding deformation and 3D position are recovered from the camera video with SfT. The deformations are transferred in real-time to a virtual object (a milk bottle). The deformed virtual object is then visualized for the animator. Using this setup, an animator can interactively position and deform the virtual object in a scene by physically deforming a real object. This may save them considerable time compared with standard deformation editing in graphics software.

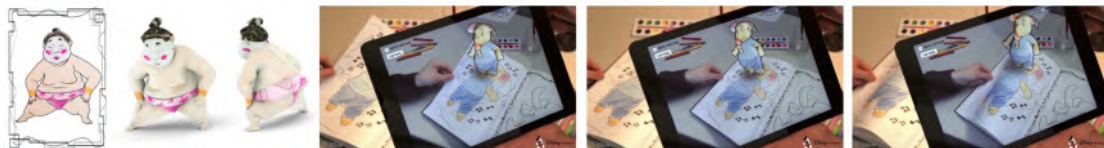


Figure 1.10: A combined HCI and AR application from [Mag+15]. A game application has been implemented on a tablet for interactively coloring a virtual 3D model using a real coloring book. A deformable page from the book is colored with a pencil and SfT is used to register images of the page with a paper template. Then the color is transferred from the template to virtual 3D cartoon model (an elephant). SfT is also used to augment the image of the paper sheet with the colored cartoon model in real-time.

make SfT with more unknown intrinsics ill- or extremely weakly-posed. Solving the case of unknown focal length is clearly the first step towards attempting other versions of SfT with an uncalibrated perspective camera.

We present novel solutions to two versions of fSfT: The first version uses a single image and point correspondences between the image and template surface. The second version extends the first version to multiple images with a common focal length. We now describe the main ideas and contributions in both versions.

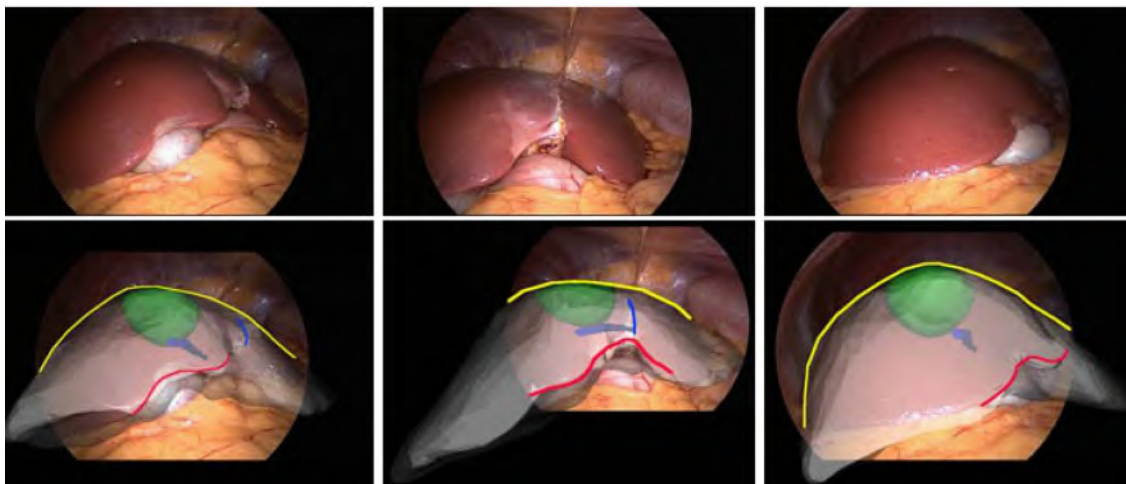


Figure 1.11: An AR SfT application from [Koo+17] to improve minimally invasive laparoscopic surgery of the liver. The template is a 3D model of a patient’s liver constructed from a pre-operative CT image. The template is registered with laparoscopic images using SfT with contour and shading constraints. Thanks to the registration and reconstruction provided, the locations of critical structures with the liver (vessels and cancerous tumors) can be virtually overlaid on the image to help guide the surgeon during surgery.

Single-view fSfT: The analytical method. We show that fSfT can be modeled with continuous differential geometry and a Partial Differential Equation (PDE) that relates focal length, motion information, and unknown 3D shape. We solve this PDE with a non-holonomic analytical approach inspired by [Bar+15]. This solves fSfT using one or more local surface regions, produces a unique solution to focal length for each region. To mitigate the influence of noise, we present a robust extension using multiple focal length estimates from different regions with robust averaging. Additionally, the analytical solution gives novel insights into fSfT well-posedness. Specifically, it gives sufficient conditions for fSfT to be well-posed and with a unique focal length solution.

Single-view fSfT: The optimization-based method. The analytical method works well for densely textured and smooth surfaces. However, it breaks down when texture is sparse or when deformation is complex, where it is difficult to estimate motion accurately. Furthermore, results are sub-optimal because the analytical method solves focal length with a local surface region taken in isolation of other regions³. Therefore it does not use all available geometric constraints. We present a second and complementary method that gives significantly more accurate results in general. We model fSfT as the optimization of a large-scale non-convex cost function $c(f, \theta)$ where f is the unknown focal length and θ is the unknown 3D deformation of the template. The form of c is similar to cost functions used in the most accurate SfT-P optimization based methods such as [CB15; Ost+12], and it combines data costs from point correspondences, and deformation costs from the template to penalize non-plausible or physically impossible deformations. We cannot optimize c with guaranteed global optimality, nevertheless we present a solution that works very well in practice. Our approach is based on local (iterative) optimization, combining a well-designed initialization strategy, careful cost modeling, and fast optimization. The principal characteristics and advantages of the approach are as follows:

³SfT methods that solve the problem using local surface regions treated independently of other regions such as [Bar+15] are often called *local methods* or *non-holonomic methods*. This should not be confused with methods that solve SfT with local optimization.

1. We model *all* deformation constraints provided by the template in the cost function using a mesh-based physical model. The results we obtain are significantly more accurate compared to the analytical method.
2. We apply normalization techniques to the cost function, which greatly reduces the need to tune cost weights. Such tuning is a known issue in cost minimization approaches, and thanks to normalization, the same weights can be used for any problem instance. In our experimental evaluation, the same weights are used in *all* test cases, covering different object shapes, mesh discretization, textures, deformations, and imaging conditions.
3. We show that an important hyper-parameter (the weight of the isometric cost) can be automatically estimated with a novel unsupervised technique, without requiring ground-truth focal length or 3D information.
4. Precise initialization is not required in general. Initialization can be performed either using the analytical method or using a very small number of focal length samples (three or fewer). We also introduce a mechanism to improve computational efficiency by avoiding repeated optimization of the same region of search space from different initializations.

Multi-view fSfT. We then seek to answer the following question: given multiple images with a common focal length, can we solve fSfT more accurately compared to using a single image? We show the answer is a clear yes. We investigate this question with two novel approaches, both demonstrating superior accuracy compared to solving fSfT with a single image. The first approach solves fSfT independent for each image and then applies robust focal length averaging. The advantages are simplicity and robustness. The second approach optimizes a cost function c' which is the multi-view extension of c . This connects geometric constraints from all images, generating a large sparse non-linear least squares problem that can be optimized with a quasi-Newton method such as Gauss-Newton (GN) or LM. However, a naive implementation quickly becomes computationally infeasible with more than a few views. We show that fSfT, and uncalibrated SfT in general, can be optimized efficiently with a quasi-Newton method with computational cost that is linear in the number of images using the Schur complement. This has been inspired by its great success in efficiently solving Bundle Adjustment in SfM with quasi-Newton methods [Tri+00].

1.4 Thesis organization and relation to publications

The thesis is organized in 6 chapters. Chapter 1 presents the thesis context from a personal standpoint concerning my research at EnCoV and within the field of Computer vision at large. This chapter introduces the three closely related reconstruction problems that are studied and solved in this thesis: *(i)* Plane-based Structure-from-Motion with Affine cameras (PSfM-A), *(ii)* Plane-based Pose Estimation with Perspective cameras (PPE-P) and *(iii)* focal length and Shape-from-Template (fSfT). General background, applications, open challenges of each problem are provided. The chapter closes with a summary of the main technical and theoretical contributions that have been achieved for each problem in this thesis. Chapter 2 provides further background details and a summary of prior state-of-the-art in each of the three reconstruction problems. Chapters 3, 4 and 5 give full details of our novel contributions to the PSfM-A, PPE-P and fSfT problems respectively. Chapter 6 concludes this thesis by distilling the similarities and differences of the three reconstruction problems and our contributions. Chapter 6 ends by discussing future research objectives stemming from this thesis.

The work on PSfM-A described in Chapter 3 was published in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) [CB17] (submitted June 2016). The work on PPE-P described in Chapter 4 was published in the International Journal of Computer Vision (IJCV) [CB14a] (submitted June 2013). The work on fSfT described in Chapter 5 was published partially in two conferences: the International Conference on Computer Vision (ICCV) [BC13a] (submitted April 2013) and Computer Vision and Pattern Recognition (CVPR) [BPC13] (submitted November 2012). Specifically, the analytical fSfT method was published in [BPC13] and the optimization-based fSfT method was described briefly in [BC13a]. Chapters 3 and 4 are very close to the PAMI and IJCV papers. Superficial modifications have been made to notation and section organization with background and conclusion material moved out and merged into the introduction, background and conclusion sections of this thesis. The empirical experiments in Chapters 3 and 4 are reproduced from the PAMI and IJCV papers. In Chapter 3, we have slightly extended the theoretical analysis to include the relationship between IPPE and weighted homography decomposition. In Chapter 5, the analytical fSfT method from the ICCV paper is described with slight modifications to unify notation and terminology and we have simplified the analytical solution derivation to link it with Chapter 3. We have significantly expanded the optimization-based fSfT method from the CVPR paper in Chapter 5. This is described in greater detail for full reproducibility, and we include six non-trivial extensions. These are as follows: *(i)* Cost normalization to significantly reduce hyper-parameter tuning. *(ii)* Unsupervised cost weight selection to automatically set the isometric cost weight (a critical hyper-parameter). *(iii)* Efficient multi-start optimization using multiple focal length samples. *(iv)* GPU-enabled optimization with Gauss-Newton, *(v)* The extension to multi-view fSfT. *(vi)* An extended experimental evaluation using 12 public datasets. For *(iv)*, techniques have been incorporated from our work on real-time Shape-from-Template with a calibrated perspective camera [CB15].

Chapter 2

Background and Related Works

Chapter summary and organization

This chapter presents background and a review of previous works in each of the monocular reconstruction problems solved in this thesis. This chapter is broken down into five main sections. In §2.1 we review the perspective and affine camera models and we discuss the theoretical and practical value of affine cameras for monocular reconstruction despite them having higher modelling error compared to perspective cameras in general. In §2.2 we connect the problems studied in this thesis and other related reconstruction problems with an innovative problem graphs that links problems via relaxations and tightening of geometric constraints. We give further specific background on plane-based pose estimation in §2.3, plane-based Structure-from-Motion in §2.4 and Shape-from-Template in §2.5.

2.1 Review of camera models

This section reviews standard definitions of perspective and affine cameras [HZ04; FLP01; FP12].

2.1.1 Perspective cameras

The projection of a 3D point $\mathbf{x} \in \mathbb{R}^3$ from camera coordinates to image coordinates using a perspective camera is defined by the projection function $\pi_P : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ as follows

$$\pi_P(\mathbf{x}) \stackrel{\text{def}}{=} [\mathbf{K}]_{2 \times 3} \text{stk}\left(\frac{1}{\mathbf{x}_3} [\mathbf{x}]_{2 \times 1}, 1\right) \quad (2.1)$$

This is parameterized by an intrinsic calibration matrix \mathbf{K} that models the linear physical characteristics of a real camera, defined as

$$\mathbf{K} \stackrel{\text{def}}{=} \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

The values f_x and f_y denote the camera's effective focal length along the x and y axes in pixels, $\mathbf{c} \stackrel{\text{def}}{=} \text{stk}(c_x, c_y)$ denotes the principal point and s denotes skew. The perspective camera reduces to the *pinhole camera* when $\mathbf{K} = \mathbf{I}_{3 \times 3}$ with the projection function

$$\pi_1(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{\mathbf{x}_3} [\mathbf{x}]_{2 \times 1} : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \quad (2.3)$$

We often use π_1 instead of π_P when \mathbf{K} is known (called an intrinsically calibrated camera) to simplify reconstruction equations. We convert to the pinhole camera by applying the affine transform \mathbf{K}^{-1} to image coordinates. Points in this transformed image are said to be in *normalized pixel coordinates*.

A 3D point $\mathbf{p} \in \mathbb{R}^3$ defined in world coordinates projects to the image point $\mathbf{q} \in \mathbb{R}^2$ as

$$\mathbf{q} = \pi_P(\mathbf{R}\mathbf{p} + \mathbf{t}) = [\mathbf{K}]_{2 \times 3} \text{stk}(\pi_1(\mathbf{R}\mathbf{p} + \mathbf{t}), 1) \quad (2.4)$$

where $\mathbf{R} \in SO_3$ and $\mathbf{t} \in \mathbb{R}^3$ give the rotation and translation from world to camera coordinates.

The perspective model can be extended to account for non-linear lens distortion effects. Several distortion models have been proposed and the most common is the Brown-Conrady's model [Bro71]. This model projects a point \mathbf{p} in world coordinates with

$$\mathbf{q} = [\mathbf{K}]_{2 \times 3} \text{stk}(d(\pi_1(\mathbf{R}\mathbf{p} + \mathbf{t})), 1) \quad (2.5)$$

where $d : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is the lens distortion function defined as

$$d(u, v) \stackrel{\text{def}}{=} (1 + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6) \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} 2\alpha_4 uv + \alpha_5(r^2 + 2u^2) \\ 2\alpha_5 uv + \alpha_4(r^2 + 2v^2) \end{pmatrix} \quad (2.6)$$

$$r^2 \stackrel{\text{def}}{=} u^2 + v^2$$

The coefficients $\alpha_{i \in [1,5]}$ denotes the distortion parameters. When the distortion parameters and \mathbf{K} are known then the effect of d and \mathbf{K} can be removed by inversion, allowing one to work with pinhole projection and normalized pixel coordinates.

2.1.2 Affine cameras

The projection of a point \mathbf{p} from world to image coordinates with an affine camera is defined as $\mathbf{q} = \mathbf{M}\text{stk}(\mathbf{p}, 1)$ where \mathbf{M} is a 2×4 *affine projection matrix*. Affine projection works well when the higher-order effects of perspective projection are small. For real cameras, this occurs when a structure is small relative to its distance to the camera, leading to co-linear projection rays. Affine projection is generally less accurate than perspective projection when modeling a real camera. Despite this fact, there are two important reasons why we should consider them for monocular reconstruction.

2.1.2.1 Reason 1: Analysis of solution ambiguities and degenerate geometries

We refer to conditions when the second-order effects of perspective projection are small as *quasi-affine conditions*. In such conditions there can exist multiple valid reconstruction hypotheses that explain the image data. Indeed, when the image data is noisy (unavoidable in practice), it can be impossible to determine which reconstruction hypothesis is correct. One clear example of this is the flip ambiguity that occurs in plane-based pose estimation as shown in Figure 1.5, where quasi-affine projection leads to two valid pose hypotheses. This contrasts plane-based pose estimation with strong perspective effects, which has only one valid pose hypothesis [Stu00; Zha00]. Consequently, studying and solving reconstruction problems with affine cameras is important to fully understand reconstruction with perspective cameras, because it reveals if solution multiplicity occurs as the perspective effects diminish, and how the solutions are related geometrically. Furthermore, we can learn what kinds of camera motion can lead to unsolvable reconstruction problems with perspective cameras as perspective effects diminish. We have done exactly this kind of analysis in Chapter 3 in the plane-based Structure-from-Motion problem, revealing new theorems for both solution multiplicity and degenerate geometries.

2.1.2.2 Reason 2: Closed-form solutions

Some reconstruction problems can be solved in closed-form with affine cameras but they cannot with perspective cameras. A classic example is non-rigid Structure-from-Motion using linear shape bases [BHB00; DLH12], which is a very successful technique in practice. The solution using an affine camera can be used as the final solution, which may be accurate enough for the application, or the solution can be used as a starting point for iterative optimization with a perspective camera.

2.1.2.3 Geometric interpretation of affine projection and affine camera types

We can interpret the geometry of the affine projection matrix in several ways. Furthermore, by restricting its DoFs, we obtain three special types of affine cameras: the orthographic, weak- and para-perspective cameras.

2.1.2.4 Interpretation 1: Affine cameras are orthographic cameras with linear image distortion

Affine projection can be seen as three sequential transforms [Qua94]: (i) a rigid transform from world coordinates to camera coordinates, (ii) an orthographic projection to the image plane and (iii) and

Affine camera	α	k	β	Total DoFs
Para-perspective	free	free	free	8
Weak-perspective	free	0	1	6
Orthographic	fixed, $\in \mathbb{R}^+$	0	1	5

Table 2.1: Definitions of the para-perspective, weak-perspective and orthographic cameras using the affine projection matrix interpretation given in Equation (2.7).

affine transform of the image plane to pixel coordinates. The decomposition of \mathbf{M} is as follows:

$$\begin{aligned} \mathbf{M} &= \alpha \begin{bmatrix} 1 & k \\ 0 & \beta \end{bmatrix} \begin{bmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (a) \\ &= \alpha \begin{bmatrix} 1 & k \\ 0 & \beta \end{bmatrix} [\mathbf{R}_{2 \times 3}, \mathbf{t}_{2 \times 1}] \quad (b) \end{aligned} \tag{2.7}$$

The terms $\mathbf{R}^o \in SO_3$ and $\mathbf{t}^o \in \mathbb{R}^3$ are the rotation and translation from world to camera coordinates. Note that the z -component of translation (depth) does not appear in Equation (2.7-b), caused by the loss of depth information by orthographic projection. Consequently, \mathbf{M} has 5 DoFs related to camera pose (3 for rotation and 2 for the x and y components of translation). The scalar $\alpha \in \mathbb{R}^+$ is the camera’s *magnification factor*, specifying the pixel size. The scalar $k \in \mathbb{R}$ denotes skew and the scalar $\beta \in \mathbb{R}^+$ denotes anisotropic scaling. We can uniquely decompose a full-rank affine projection matrix with Equation (2.7) and we provide pseudo-code in §A.2.2. From Equation (2.7), we define the orthographic, weak- and para-perspective cameras according to whether α , k and β are free DoFs. This is defined in Table 2.1, where these cameras have 5, 6 and 8 DoFs respectively.

2.1.2.5 Interpretation 2: Affine cameras are linearized perspective cameras

We can also make a para-perspective camera by linearizing π_1 about a point $\mathbf{x}' \in \mathbb{R}^3$ in camera coordinates. We refer to \mathbf{x}' as the *point-of-perspective-linearization*. The Taylor expansion of π_1 about \mathbf{x}' is

$$\pi_1(\mathbf{x}; d, \mathbf{b}) = \frac{1}{\mathbf{x}'_3} [\mathbf{I}_{2 \times 1}, -\mathbf{x}'_{2 \times 1}] \mathbf{x} - \mathbf{x}'_{2 \times 1} + \mathcal{O}_2 \tag{2.8}$$

where \mathcal{O}_2 denotes higher-order terms. We approximate π_1 with affine projection by ignoring \mathcal{O}_2 . Consequently, a point \mathbf{p} in world coordinates projects with this approximation as $\mathbf{q} \approx \mathbf{M} \text{stk}(\mathbf{p}, 1)$ where \mathbf{M} decomposes as

$$\begin{aligned} \mathbf{M} &= [\mathbf{B}\mathbf{R}, \mathbf{B}\mathbf{t} - \mathbf{b}] \quad (a) \\ \mathbf{B} &\stackrel{\text{def}}{=} \frac{1}{\mathbf{x}'_3} [\mathbf{I}_{2 \times 2}, -\mathbf{x}'_{2 \times 1}] \quad (b) \end{aligned} \tag{2.9}$$

Equation (2.9) parameterizes \mathbf{M} by a point-of-perspective-linearization and a camera pose. Note that we obtain the same \mathbf{M} by scaling \mathbf{x}'_3 by an arbitrary value s and scaling \mathbf{t} by $\frac{1}{s}$. Therefore Equation (2.9) has 8 effective DoFs: 3 for \mathbf{R} and 5 for \mathbf{x}' and \mathbf{t} . In the special case when \mathbf{x}' is fixed and oriented with the optical axis *i.e.* $\mathbf{x}' = d \text{stk}(0, 0, 1)$ for some depth DoF d , we create a weak-perspective camera with $\alpha = \frac{1}{d}$ and $k = \beta = 0$. This is verified by substituting Equation (2.9-b) into Equation (2.7) with $\mathbf{B} = [\mathbf{I}, \mathbf{0}_{2 \times 1}]$, giving a 6 DoF affine projection matrix. When we also fix d to a specific value, we create an orthographic camera with $k = \beta = 0$ and a fixed α , giving a 5 DoF affine projection matrix.

Considering affine cameras in terms of linearization error helps us understand when the cameras can be used with high accuracy for reconstruction purposes. For the para-perspective camera, we reduce

linearization error by setting the point-of-perspective-linearization \mathbf{x}' close to the structure's center $\bar{\mathbf{x}}$. We then obtain an accurate projection if the distances between structure points and $\bar{\mathbf{x}}$ is small relative to $\|\bar{\mathbf{x}}\|_2$. Of course $\bar{\mathbf{x}}$ is normally not known in advance. Therefore, to reduce linearization error using a para-perspective camera, we must jointly reconstruct the scene and estimate $\bar{\mathbf{x}}$ in each view. This is essentially a camera calibration task: we are setting the point-of-perspective-linearization, which is an intrinsic of the para-perspective camera, to $\bar{\mathbf{x}}$.

This calibration problem is simplified if the real camera has an intrinsic perspective calibration, or if the intrinsics are well approximated by canonical values. In such cases, we have $\bar{\mathbf{x}} \approx d \text{stk}(\bar{\mathbf{q}}, 1)$ where $\bar{\mathbf{q}}$ is the structure's centroid in normalized pixel coordinates (which can be measured with *e.g.* point correspondences using the intrinsic calibration), and \bar{d} is the depth of $\bar{\mathbf{x}}$. Thus, by setting $\mathbf{x}' = \bar{d} \text{stk}(\bar{\mathbf{q}}, 1)$, we have fixed 2 of the para-perspective camera's DoFs. The reconstruction problem can then be converted to an equivalent one with a weak-perspective camera with rigid coordinate transform. Specifically let $\tilde{\mathbf{R}}$ be a rotation matrix that aligns the vector $\text{stk}(\bar{\mathbf{q}}, 1)$ to the optical axis $\text{stk}(0, 0, 1)$. Next, we transform image points by the homography $\mathbf{H} = \tilde{\mathbf{R}}$, and the transformed points are then described by a weak-perspective projection. We use this technique to generalize reconstruction results in this thesis from a weak- to para-perspective cameras with known perspective intrinsics and known barycenter of the projected structure.

For the weak-perspective camera, we reduce linearization error by setting $\alpha = \frac{1}{\bar{d}}$. We therefore obtain an accurate projection if the structure is close to the optical axis and it has low variation in depth relative to \bar{d} . For the orthographic camera, where α is fixed, we obtain an accurate projection if (i) the structure is close to the optical axis, (ii) it has low variation in depth and (iii) \bar{d} is similar to $\frac{1}{\alpha}$.

2.1.2.6 Selecting the affine camera type

The choice of using an orthographic, weak- or para-perspective camera in a reconstruction problem is a model selection problem. On the one hand, they are models of increasing complexity that reduce perspective projection approximation error. On the other hand, increasing complexity leads to weaker geometric constraints, where we must resolve more camera DoFs. There exist problems that are only solvable with the orthographic camera, other problems that are solvable with orthographic and weak-perspective but not para-perspective cameras, and other problems that are solvable with all three types. For example, PSfM-A is solvable with the orthographic camera but not solvable with the other two types, discussed further in §2.4.2. Consequently, the right affine camera is a balance between problem well-posedness and modeling accuracy.

2.2 Problem relationships

We tackle three fundamental monocular reconstruction problems in this thesis: Plane-based Structure-from-Motion with an Orthographic camera (PSfM-O) in Chapter 3, Plane-based Pose Estimation with a Perspective camera (PPE-P) in Chapter 4 and focal length and Shape-from-Template (fSfT) in Chapter 5. They are different problems, yet they are special cases of general monocular reconstruction from motion, and in this section we present the problems in this general context.

Solving any monocular reconstruction problem using geometry requires prior knowledge and data encoded as geometric equations. A general way to organize this knowledge is into five aspects:

1. *Structure prior knowledge*: Defines what is known *a priori* about the structure's geometry and

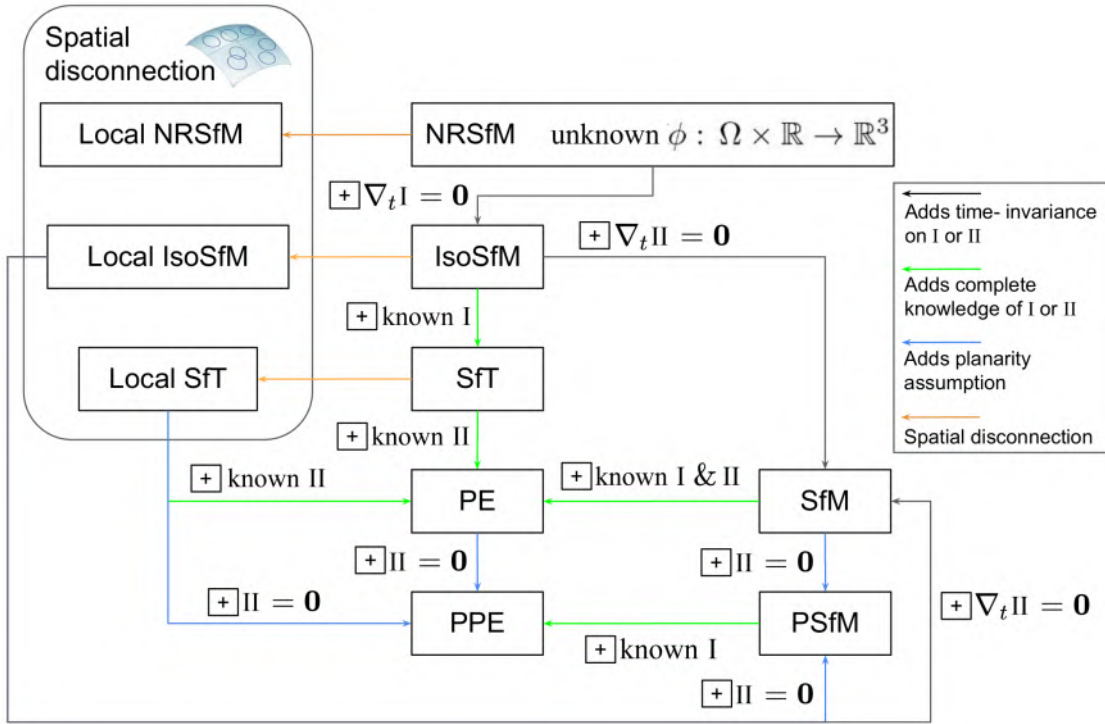


Figure 2.1: The relationships between the major classes of monocular reconstruction problems expressed as a problem graph. Different problems are formed by adding or removing constraints on the first I or second II fundamental forms of the unknown surface embedding function ϕ . The edges of the problem graph are directed with the plus symbol indicating adding the labelled constraints when the edge is traversed. These edges can also be traversed in the reverse direction, making a problem relaxation, by removing the edge's constraints.

topology in object (local) coordinates.

2. *Transformation prior knowledge:* Defines what is known *a priori* about how the structure transforms from object coordinates to camera coordinates.
3. *Camera projection prior knowledge:* Defines what is known *a priori* about how the camera projects structures from camera coordinates to image coordinates.
4. *Image motion prior knowledge:* Defines how 1, 2 and 3 constrain image motion *e.g.* constraining it to a special type of motion such as a homography or an affine transform.
5. *Motion data:* Defines how motion is measured in images, typically with point correspondences, or other forms (line correspondences, curve correspondences, optical flow or combinations).

A specific monocular reconstruction problem is created by defining the above five aspects. Then geometric equations that relate known and unknown entities can be formed and solved. We characterize PSfM-O, PPE-P and fSfT regarding these five aspects in Table 2.2 (top), highlighting the similarities and differences of each problem. Table 2.2 (bottom) characterizes three other closely related and previously studied monocular reconstruction problems with this schema. These are focal length and Plane-based Pose Estimation (fPPE), Plane-based Structure-from-Motion with a calibrated Perspective camera (PSfM-P) and Shape-from-Template with a calibrated Perspective camera (SfT-P). The problem pairs (PPE-P, fPPE), (PSfM-O, PSfM-P) and (SfT-P, fSfT) differ only in camera projection

Problem	PPE-P	PSfM-O	fSfT
Structure	known	unknown	known
prior knowledge	flat surface	flat surface	arbitrary surface
Transformation	unknown and rigid	unknown and rigid	unknown and quasi-
prior knowledge	\Rightarrow known I and II	\Rightarrow known I and II	isometric \Rightarrow known I approximately
Camera projection	perspective and	orthographic and	perspective and
prior knowledge	known	unknown optical scale	unknown focal length
Image motion			
prior knowledge	2D homography	2D affine	2D warp
Motion data	point correspondences	point correspondences	point correspondences

Problem	fPPE	PSfM-P	SfT-P
Structure	known	unknown	known
prior knowledge	2D surface	2D surface	arbitrary surface
Transformation	unknown and rigid	unknown and rigid	unknown and quasi-
prior knowledge	\Rightarrow known I and II	\Rightarrow known I and II	isometric \Rightarrow known I approximately
Camera projection	perspective and	perspective and	perspective and
prior knowledge	unknown focal length	known	known
Image motion			
prior knowledge	2D homography	2D homography	2D warp
Motion data	point correspondences	point correspondences	point correspondences

Table 2.2: (Top) characterization of the three monocular reconstruction problems studied and solved in this thesis: PPE-P, PSfM-O and fSfT. For each problem, novel closed-form solutions have been developed that relate five knowledge sources. (Bottom) characterization of related monocular reconstruction problems: fPPE, PSfM-P and SfT-P. The symbols I and II denote the first and second fundamental forms of a surface, or equivalently its metric and shape tensors respectively.

prior knowledge. Nevertheless, these differences change the problems significantly. We refer the reader to page xiv for all problem acronym definitions with representative citations of algorithms solving each problem.

We also present a unified view of these problems and how they connect to one another using continuous differential geometry constraints, shown as a problem graph in Figure 2.1. This problem graph is useful because it shows the fundamental connections that exist between various reconstruction problems, and how to convert between problems, by traversing the edges, corresponding to either adding or releasing *a priori* geometric knowledge with respect to I or II. We use general continuous functions in the problem graph to make it independent of the surface representation or discretization. We have also dropped the camera model qualifications to simplify the graph for better clarity.

We denote as $\phi(u, v, t) : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^3$ the surface embedding function that transforms a surface point (u, v) defined in a 2D parameterization space Ω to 3D reconstruction space at time t . A reconstruction problem implies that ϕ is unknown *a priori*. The first fundamental form of ϕ describes the 2D deformation of Ω induced by ϕ (equivalently deformation within the surface’s tangent plane). This is represented by the *metric tensor function* $I(u, v, t)$. The second fundamental form of ϕ describes surface curvature, represented by the *shape tensor function* $II(u, v, t)$. Different reconstruction problems are formed by applying assumptions and constraints to I and II, indicated by the arrows in Figure 2.1. The plus symbol indicates adding constraints when an edge is traversed. The green arrows represent problem tightenings in the formal mathematical sense because they convert unknown variables to known variables. We can also transition between problems along edges in the opposite direction, making a problem relaxation, by subtracting the edges’s constraints.

NRSfM is the most unconstrained and general problem. The problem chain NRSfM \rightarrow IsoSfM \rightarrow SfT \rightarrow PE \rightarrow PPE is the central downward chain in Figure 2.1, formed by adding increasing prior knowledge of I and II. NRSfM specializes to IsoSfM by adding the constraint $\nabla_t I = \mathbf{0}$, which makes

ϕ isometry-preserving by definition. IsoSfM then specializes to SfT when a template of the surface is known *a priori*. The template provides us with \mathbf{I} , which is constant over time in IsoSfM and SfT by definition. The template does not provide the shape of the surface over time, and therefore \mathbf{II} is unknown *a priori* in SfT (as with NRSfM and IsoSfM). SfT specializes to PE when we also assume that the surface does not deform with respect to the template. More generally, SfT specializes to PE when \mathbf{II} is known *a priori*. This is because when \mathbf{I} and \mathbf{II} are both known *a priori*, ϕ can be reconstructed uniquely up to 3D rigid transform (the pose of the surface). PE then specializes to PPE when we also assume $\mathbf{II} = \mathbf{0}$, constraining the surface to be flat.

There are loops in our problem graph that connect these problems with SfM and PSfM. IsoSfM specializes to SfM by assuming the surface does not deform over time, which is equivalent to adding the constraint $\nabla_t \mathbf{II} = \mathbf{0}$. SfM then specializes to PE when we assume the surface is known up to rigid pose, equivalent to assuming \mathbf{I} and \mathbf{II} are known *a priori*. SfM also specializes to PSfM by adding the constraint $\nabla_t \mathbf{II} = \mathbf{0}$, which implies the surface is flat. Finally, PSfM specializes to PPE when we know the plane’s metric structure (how ϕ distorts Ω). This is equivalent to knowing \mathbf{I} *a priori*.

We have also included in the problem graph the fact that deformable reconstruction problems can be solved with *local reconstruction* [Var+09a; CB10a; TJK10; PBC13; BC13b; CPB14b; Chh+17b; Bar+15; PPB18; RYA14]. Local reconstruction involves dividing Ω into local regions (spatial disconnection) and each is reconstructed independently. The idea is that local reconstruction can be performed using a simple geometric model, often with a closed-form solution. This is represented in Figure 2.1 by the *spatial disconnection* arrows, which represent problem relaxations because geometric constraints acting between local regions are lost. Local SfT can be converted to multi-instance PE by assuming the local regions do not change shape with respect to the template. This is equivalent to saying \mathbf{II} is known *a priori* for each region, which is supplied by the template. Local SfT can also be converted to multi-instance PPE by assuming that the local regions are flat, which is often a good approximation for surfaces with low curvature such as cloth or bending paper. As the region area tends to zero, local reconstruction with PPE becomes equivalent to reconstructing the tangent plane of ϕ [Chh+17a; Bar+15]. Local reconstruction can also be used for IsoSfM analogously with SfT. IsoSfM is converted to multi-instance SfM by assuming the shape of each local region does not change over time. IsoSfM can also be converted to multi-instance PSfM by assuming the local regions are flat. The majority of local IsoSfM methods use planar regions similarly to local SfT. An exception is [RYA14] that uses general local model discretized by a set of neighbouring points. Each region is individually reconstructed using an established sparse SfM method. It may be possible to use a combination of planar and non-planar local regions to reconstruct both smooth and non-smooth regions. However, we have not seen this yet attempted in the literature.

We can extend the problem graph to include different camera models, and calibrated and uncalibrated versions with each camera model. This is shown in Figure 2.2 with 24 problems arranged as a grid connected by edges. We include perspective cameras with known and unknown focal length, and affine camera models because they are the most relevant to this thesis. We have discussed linearizing perspective projection in §2.1.2.5 with known or default intrinsics. The linearization yielding the lowest linearization error is effectively equivalent to using a rotated weak-perspective camera. A weak perspective camera is converted to an orthographic camera by fixing the magnification factor α to a constant value. Problems in green boxes represent problems studied in this thesis (PPE-P in Chapter 3, PSfM-O in Chapter 4 and fSfT in Chapter 5). We also highlight PPE-O in green because our solution to PSfM-O contains a novel closed-form solution to PPE-O to resect the cameras. PSfM-WP is highlighted in red because it is unsolvable as discussed in §2.4.2. Problems in white boxes represent

those that have previously been studied. Problems in gray are those that have not previously been studied (only IsoSfM-WP).

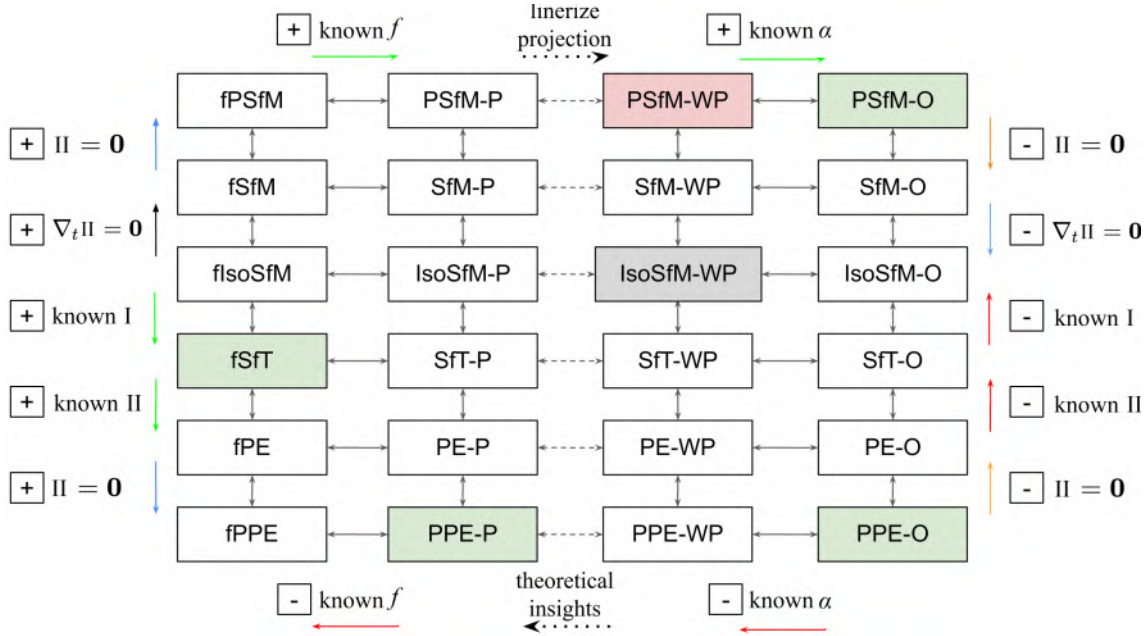


Figure 2.2: The relationships between different monocular reconstruction problems using differential geometry constraints on the metric tensor I and shape tensor II . The positive and negative symbols at each edge indicate adding or subtracting constraints.

The problem graphs in Figures 2.1 and 2.2 can be used to obtain theoretical and practical insights into these related problems:

- **Inheritance of degeneracies.** If a problem P is degenerate (ill-posed) then any relaxation of P will necessarily be degenerate. For example, PE is degenerate when structure is collinear for all camera models, therefore SfT and IsoSfM are also degenerate when structure is collinear for all camera models. Similarly, if a reconstruction problem is ill-posed then solving it with local models via surface disconnection will also be ill-posed.¹
- **Inheritance of necessary well-posedness conditions by relaxation.** A necessary condition to make problem P well-posed is a necessary condition for any problem P' that is a relaxation of P . For example, a necessary condition for fPPE is that the surface is not fronto-parallel. This also a necessary well-posedness condition for fPE, fSfT, flsoSfM and fNRSfM.
- **Inheritance of sufficient well-posedness conditions by tightening.** A sufficient well-posedness condition for problem P' is also a sufficient well-posedness condition for any problem P that is a tightening of P .
- **Algorithm application in special instances.** If we have an algorithm A that solves problem P , it can be used to solve a problem P' in special cases when there is a path from P' to P by adding constraints. For example, an SfM algorithm could be used to solve IsoSfM using a set of images where the object barely deforms. The choice between using the IsoSfM and SfM

¹It may be the case that if the reconstruction problem is ill-posed, some of the local models may be solvable. For example, in IsoSfM, if a surface consists of a static part with no motion, and a moving part, the moving part may be reconstructable with a local model, but not the static part.

algorithm is a model selection task. Another example is to use a PPE algorithm to solve local SfT when the surface does not bend significantly ($\nabla_t \Pi \approx \mathbf{0}$).

- **Algorithm application in generalized instances.** If we have an algorithm A' that solves problem P' , it may be possible to apply it for problem P if there is a path from P to P' by removing constraints. For example, an IsoSfM algorithm could be used to solve an SfM instance (*i.e.* if a deformable object does not deform in a particular set of images). However, because A' does not use all available constraints (in this case $\nabla_t \Pi = \mathbf{0}$), it is likely to be more sensitive to noise and fail in instances that could be solved by an algorithm A that uses all available constraints. The choice between using A and A' is also a model selection task.
- **Insights in quasi-affine conditions.** The linearization of perspective projection provides insights on problem behavior when perspective effects are small. This has been described in detail in §2.1.2.1.

We have focused the problem graphs around the problems studied in this thesis. It could be extended to chart other related problems such as reconstruction with multiple unknown camera intrinsics or time varying intrinsics. This may reveal new problems, for example recently, fIsoSfM has been studied [Pro+18; PBP18], however IsoSfM with unknown and time-varying intrinsics has not yet been attempted in the literature. The problem graphs could also be extended to include other deformable reconstruction problems that relax known I (the isometric assumption) to *e.g.* known angles in the surface’s tangent plane (conformal deformation) [Bar+15] or known surface area [Cas+19]. Furthermore, low-rank models could be included and new problems/settings discovered in this way, which we aim to do in follow-up work.

2.3 Plane-based Pose Estimation (PPE)

2.3.1 Homography decomposition methods with perspective cameras

A homography matrix \mathbf{H} explains the motion between a planar object defined on the plane $z = 0$ in object coordinates and the image of a perspective camera. Using normalized pixel coordinates, \mathbf{H} encodes the pose of the camera relative to the object with the decomposition

$$\mathbf{H} = \lambda \left[\begin{bmatrix} \mathbf{R} \\ \mathbf{t} \end{bmatrix} \right]_{3 \times 2}, \quad (2.10)$$

where \mathbf{R} and \mathbf{t} give the pose of the camera. In the absence of noise, \mathbf{H} decomposes uniquely with $\lambda = \|\mathbf{H}_{3 \times 2}\|_2$ and $\left[\begin{bmatrix} \mathbf{R} \\ \mathbf{t} \end{bmatrix} \right] = \frac{1}{\lambda} \mathbf{H}$. The third column of \mathbf{R} is then completed as the cross-product of the first two columns. Given a noisy homography $\hat{\mathbf{H}}$, an exact decomposition does not exist in general. Zhang [Zha00] and Sturm [Stu00] proposed slightly different methods to handle noise by minimizing an algebraic error in closed-form. In Zhang’s method, first orthonormality between columns 1 and 2 of \mathbf{R} is relaxed, denoted as \mathbf{r}_1 and \mathbf{r}_2 . These are estimated as $\hat{\mathbf{r}}_{j \in [1,2]} = \hat{\lambda}_j \hat{\mathbf{h}}_j$ with $\hat{\lambda}_{j \in [1,2]} = \frac{1}{\|\hat{\mathbf{h}}_j\|_2}$ and $\hat{\mathbf{h}}_{j \in [1,2]}$ denoting the j^{th} column of $\hat{\mathbf{H}}$. Next λ and \mathbf{t} are estimated as $\hat{\lambda} = \frac{1}{2}(\hat{\lambda}_1 + \hat{\lambda}_2)$, and $\hat{\mathbf{t}} = \hat{\lambda} \hat{\mathbf{h}}_3$. The rotation matrix is then estimated as $\hat{\mathbf{R}} = [\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_1 \times \hat{\mathbf{r}}_2]$. This is then transformed to the closest rotation matrix in the L2 sense with the SVD.

Sturm’s method differs by directly finding the pose decomposition with minimal distance in the L2 sense:

$$\min_{\hat{\lambda}, \hat{\mathbf{R}}_{3 \times 2}, \hat{\mathbf{t}}} \left\| \hat{\lambda} \hat{\mathbf{H}} - \left[\begin{bmatrix} \hat{\mathbf{R}}_{3 \times 2} \\ \hat{\mathbf{t}} \end{bmatrix} \right] \right\|_2^2 \quad \text{s.t.} \quad \left[\hat{\mathbf{R}} \right]_{3 \times 2}^\top \left[\hat{\mathbf{R}} \right]_{3 \times 2} = \mathbf{I}_2 \quad (2.11)$$

This has a unique and analytical solution computed from the SVD of $\begin{bmatrix} \hat{\mathbf{H}} \end{bmatrix}_{3 \times 2}$.

These methods perform very similarly in practice and their key advantage is negligible computational cost. However, they have three main shortcomings. Firstly they are not optimal in the ML sense because they optimize an algebraic cost. Secondly, they are relatively sensitive to noise. Thirdly, they become unstable in quasi-affine conditions. Nevertheless, because of their speed, these methods have been the preferred approach in many computer vision libraries such as OpenCV to provide a fast initial pose estimate that can be iteratively refined in the ML sense with e.g. Gauss-Newton or Levenberg-Marquardt.

2.3.2 Failure of homography decomposition methods in quasi-affine conditions

We now show briefly why the homography decomposition methods fail in quasi-affine conditions (when $\hat{\mathbf{H}}_{31}$ and $\hat{\mathbf{H}}_{32}$ tend to zero). We draw attention to this because it is a common source of error in real-world applications such as estimating the pose of AR markers. For Zhang’s method, when $\hat{\mathbf{H}}$ is affine, the rotation estimate $\hat{\mathbf{R}}_1$ before correction with the SVD has the form

$$\hat{\mathbf{R}}_1 = \begin{bmatrix} \mathbf{A} & 0 \\ & 0 \\ 00 & b \end{bmatrix} \quad (2.12)$$

for some $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ and $b \in \mathbb{R}$. Using the SVD $\hat{\mathbf{R}}_1 = \mathbf{U}\Sigma\mathbf{V}^\top$, rotation is estimated as $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^\top$. This method fails because $\hat{\mathbf{R}}_1$ is block diagonal, so \mathbf{U} and \mathbf{V} are also block diagonal, which makes $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^\top$ always a 2D rotation about the z -axis (*i.e.* rotation of the plane about the optical axis). This is simple to demonstrate and given in Appendix A.2.1.

The failure of Sturm’s method is similar. When $\hat{\mathbf{H}}$ has no perspective terms, $\begin{bmatrix} \hat{\mathbf{H}} \end{bmatrix}_{3 \times 2} = \text{stk}(\mathbf{B}, \mathbf{0}_{2 \times 1})$ for some 2×2 matrix \mathbf{B} . We define as $\mathbf{B} = \mathbf{U}_B \Sigma_B \mathbf{V}_B^\top$ and $\begin{bmatrix} \hat{\mathbf{H}} \end{bmatrix}_{3 \times 2} = \mathbf{U}_H \Sigma_H \mathbf{V}_H^\top$ the SVDs of \mathbf{B} and $\begin{bmatrix} \hat{\mathbf{H}} \end{bmatrix}_{3 \times 2}$ respectively. We then have $\mathbf{U}_H = \text{stk}(\mathbf{U}_B, \mathbf{0}_{2 \times 1})$, $\mathbf{V}_H = \text{stk}(\mathbf{V}_B, \mathbf{0}_{2 \times 1})$ and $\Sigma_H = \text{stk}(\Sigma_B, 0)$. The least squares rotation estimate is $\begin{bmatrix} \hat{\mathbf{R}} \end{bmatrix}_{3 \times 2} = \text{stk}(\mathbf{U}_B, \mathbf{0}_{2 \times 1}) \mathbf{V}_B^\top = \text{stk}(\mathbf{U}_B \mathbf{V}_B^\top, \mathbf{0}_{2 \times 1})$. Using the cross-product, the third row of $\hat{\mathbf{R}}$ is $\text{stk}(0, 0, 1)$, so $\hat{\mathbf{R}}$ is a rotation about the optical axis.

In practice, $\hat{\mathbf{H}}$ is never perfectly affine, however in quasi-affine conditions where the perspective terms in $\hat{\mathbf{H}}$ strongly diminish with respect to noise in $\hat{\mathbf{H}}$, the solutions of Zhang’s and Sturm’s methods tend towards rotations about the optical axis. These failures of Zhang’s and Sturm’s methods when the homography is affine or quasi-affine explain the poor performance for smaller planes or planes viewed further from the camera, which is shown empirically in the experimental section of Chapter 4.

2.3.3 Perspective- n -Point methods

Perspective- n -Point (PnP) is the problem of estimating the pose of an intrinsically calibrated perspective camera with respect to an object using point correspondences. This is an important and well-studied vision problem that can be applied to both planar and non-planar structures. In this section we give further background details of state-of-the-art PnP methods prior to [CB14a]. For planar objects, we can estimate the structure-to-image homography from a minimum of 4 point correspondences uniquely provided that there does not exist a subset of $n - 1$ structure points that are

colinear. Therefore the homography decomposition methods are PnP methods when the homography is computed from point correspondences.

P3P. P3P is the minimal case of PnP with 3 points and it has been studied in detail [DRL89; FB81; Gao+03; Har+91; QL99; Har+94; KR17; Har+94; Gru41]. P3P corresponds to an exact system of equations with 6 pose unknowns and 6 equations, and it reduces to a univariate quartic equation. P3P therefore has either 0, 1, 2, 3 or 4 real solutions. Its most common use is to solve PnP when point correspondences have outliers within a RANSAC-based process, where P3P provides pose hypotheses from 3 random correspondences. Various analytical P3P solutions have been proposed with the goal of increasing numerical stability and eliminating algorithm flaws. The so-called vertex permutation problem was an issue for early solutions where a permutation of the points significantly affected the solution stability [Har+94]. One of the most well-known methods was presented by [FB81] who reformulated the problem using angles and edge lengths of the triangle formed from the three points. A detailed analysis of the stability of various older P3P methods was presented in [Har+94]. [Gao+03] were the first to provide algebraic criteria for determining the number of real-valued P3P solutions. P3P is now generally considered solved.

PnP with general n When $n \geq 4$ we have a redundant system of equations with $2n$ constraints on 6 unknowns. The redundancy can be exploited to improve pose accuracy with noise. Numerous closed-form PnP algorithms have been proposed in the literature with the most competitive prior to [CB14a] being [LMF09; LXX12; HR11]. The main differences between these algorithms are speed, accuracy, sensitivity to noise, and how computational cost scales in n .

EPnP [LMF09] is a well-known closed-form PnP method with $O(n)$ computation cost. It represents each point as a weighted sum of four virtual control points and the virtual control points become the unknowns. Pose is then estimated from the virtual control points as the solution to a small algebraic system. However, EPnP has several limitations. Firstly it requires two different formulations for co-planar and non co-planar points, leading to difficulties with quasi co-planar cases. Secondly, it only returns a single solution, and it is relatively unstable without strong perspective effects with a co-planar points. Thirdly, it cannot handle the minimal case of $n = 3$. Fourthly, it is generally less accurate than the ML estimate found by iterative optimization (see below). RPnP [LXX12] improved on EPnP by formulating PnP as the solution to a redundant system of P3P equations that are satisfied in the least-squares sense. First a pair of points is located with a large separation, found by random sampling. Next $n-2$ P3P problems are generated using the point pair and each remaining point. Each P3P problem is converted into a least-squares loss by squaring its residuals. The losses are then added and the local minima are found as the roots of a 7^{th} -order polynomial. Finally, pose is estimated from each of the local minima. The approach is relatively fast. However, its main limitation is to be generally less accurate than the ML estimate. Direct Least Squares (DLS) [HR11] solves PnP as the real-roots of a 27^{th} -degree polynomial. Translation is first eliminated, then an object space loss is optimized with respect to \mathbf{R} , parameterized by 3 Cayley angles. The local minima are roots of three cubic equations in the Cayley angles, which requires the Schur complement of a sparse 120×120 action matrix and the SVD of a 27×27 matrix. DLS can produce more accurate solutions than EPnP and RPnP. However it has three limitations. Firstly it is much more computationally expensive than EPnP and it is not suitable for real-time use. Secondly, it suffers from numerical instabilities, and thirdly it has singularities arising from the Cayley parameterization (when the angle of rotation approaches 180 degrees).

PnP is also solved with iterative gradient-based optimization of a non-convex cost function. Under mild noise assumptions (I.I.D. zero-mean Gaussian noise), the ML estimate minimizes the L2 reprojection cost as

$$\left(\hat{\mathbf{R}}, \hat{\mathbf{t}}\right) = \arg \min_{\substack{\mathbf{R}^3 \in SO_3 \\ \mathbf{t} \in \mathbb{R}^3}} \sum_{i=1}^n \|\pi_P(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \hat{\mathbf{q}}_i\|^2 \quad (2.13)$$

This generally yields the most accurate pose estimates among all PnP methods in real-world use when point correspondences do not contain outliers. Such outliers can be effectively removed with RANSAC or later variants such as PROSAC [CM05]. Problem (2.13) has no known closed-form solution due to the non-convexity of both π_P and $\mathbf{R} \in SO_3$. It is therefore optimized iteratively using the cost gradient, and the gold standard method today is with Levenberg-Marquardt with efficient implementations such as OpenCV's `solvePnP` method. Indeed, this is substantially faster than DLS. This approach requires good initialization, which can be provided by a fast closed-form PnP method. For co-planar points, homography decomposition is usually used thanks to its very low computational cost. However, its relatively poor accuracy in noisy conditions leads to requiring more iterations and therefore higher optimization cost. Furthermore, the failure as perspective effects diminish leads to both very poor initialization and a failure to converge on both pose solutions.

2.3.4 Extensions of PnP to partially calibrated or uncalibrated cameras

Given a single view of a planar object with unknown pose, we can resolve pose and at most two unknown terms in the intrinsics matrix [Zha00]. This is because the object-to-image homography provides 8 independent constraints, 6 of which are required for pose. Assuming the intrinsics are constant across views, we can fully calibrate the intrinsic matrix and determine poses with a minimum of 3 non-fronto-parallel views in closed-form using an algebraic cost. The method of [Zha00] is used extensively today to calibrate a camera with views of a planar calibration target such as a checkerboard. For non-co-planar points, various single-view minimal problems have been studied where most approaches formulate and solve the problems with numerical root-finding of polynomial systems. Pose with unknown focal length (fPnP) has received the most attention [AC95; BKP08; CLS10; PAM13; Tri99a; Zhe+14; Wu15; ZK16] requiring a minimum of 4 points (which can be co-planar). fPnP has practical value because for many real cameras we can assume the principal point is at the image center and there is no skew. Other closed-form solutions have been presented in the literature with different unknown intrinsics and non-co-planar points. These include pose with unknown focal length and aspect ratio [Guo13], unknown principal point [Tri99a], and unknown radial distortion [M12; JB09; KBP13; Nak16].

2.4 Plane-based Structure-from-Motion (PSfM)

2.4.1 Solutions with perspective cameras

2.4.1.1 Homography decomposition.

A homography matrix \mathbf{H} explains the image motion of a rigid planar structure between two views. Defining this in normalized pixel coordinates, it encodes scene geometry with the decomposition

$$\mathbf{H} = \lambda \left(\mathbf{R}_{i,j} + \frac{1}{d} \mathbf{t}\mathbf{n}^\top \right) \quad (2.14)$$

where $\mathbf{R} \in SO_3$ and $\mathbf{t} \in \mathbb{R}^3$ is the relative pose, \mathbf{n} is the structure’s normal vector with respect to the first view and d is the depth of the structure in the first view. To reconstruct the scene from \mathbf{H} , we must fix the scale ambiguity between d and \mathbf{t} (usually by arbitrarily setting $d = 1$). The translation solution will then be up to scale. The decomposition of \mathbf{H} according to Equation (2.14) has a closed-form solution using the SVD [FL88; ZH96]. In general there are two geometrically feasible solutions, and a third view can be used to establish a unique solution. If \mathbf{H} is noisy there will usually exist an exact decomposition because Equation (2.14) is an exact system of 9 equations and 9 unknowns.

Homography decomposition has been generalized to $M > 2$ views to reduce the influence of noise [FL88; ZH95; MC02]. This works by stacking homographies between pairs of views to generate an $M \times M$ supercollineation matrix \mathbf{H}_S of theoretical rank 3. The rank-3 approximation of \mathbf{H}_S is computed using the SVD, from which the structure’s normal vector and relative poses can be recovered in closed-form.

Homography decomposition with one or more unknown camera intrinsics has also been investigated [MC02; Tri98; HZ04; GS03; Men+08; GS03]. However, the solutions are not closed-form and they solve with local iterative optimization. The solutions are not statistically optimal because an algebraic cost is minimized, and an initial estimate for the unknown intrinsics is required to prevent falling into a local minimum. In the case of [GS03], a closed-form solution was presented assuming one of the images of the structure is taken in a fronto-parallel position, equivalent to knowing approximately the plane’s metric structure. This makes the problem more like plane-based pose estimation with camera calibration. [Tri98] proposed an approach using the absolute quadric from five or more views assuming fixed intrinsics in all views, and minimized an algebraic error with Gauss Newton. The approach has been extended to handle varying focal lengths [HZ04; GS03; ZI02].

2.4.1.2 Bundle Adjustment.

BA is the de facto standard to achieve the most accurate solutions to SfM. This optimizes structure and camera poses iteratively to reduce a non-convex reprojection cost. BA is usually implemented with a quasi-Newton method (Levenberg-Marquardt is most often used) and implemented as iterative re-weighted least squares (IRLS). Thanks to the problem’s sparsity pattern, the IRLS problem can be solved efficiently for a large number of views with the Schur complement [Tri+00]. Mature tools now exist such as OpenMGV [Mou+16] and Ceres [AM+], and they are routinely used to solve SfM from point correspondences with and without camera self-calibration. When image noise is zero-mean IID Gaussian, the L_2 reprojection error is used, producing solutions that are statistically optimal in the ML sense. If correspondences have outliers, they can either be detected and removed before BA typically with RANSAC-based methods, or they can be handled with a robust reprojection loss, typically implemented with M-estimators such as Huber or L_1 . For planar structures, BA is often applied without exploiting structure planarity in the optimization problem. On the one hand this simplifies matters because it does not require *a priori* scene knowledge or model selection mechanisms to switch between planar and non-planar cases. On the other hand, solutions can be less accurate by not exploiting all available constraints. BA requires an initial estimate to prevent convergence on a local minimum. For planar structures, this is usually obtained by a homography decomposition method.

2.4.1.3 Failure in quasi-affine conditions.

Plane-based SfM is unsolvable or highly unstable with a perspective camera in quasi-affine conditions. This is because \mathbf{H} becomes a quasi-affine transform with 6 effective DoFs, so it loses 2 DoFs corresponding to second-order perspective effects. It therefore contains insufficient information to resolve relative pose (6 unknowns) and additional unknowns related to structure. Clearly, this implies solving the problem with unknown perspective intrinsics also becomes unsolvable. As a consequence, all methods described in this section fail in quasi-affine conditions, because the problem they are designed to solve becomes unsolvable.

2.4.2 Solutions with affine cameras

Plane-based SfM is solvable with the orthographic camera (SfM-O) but it is unsolvable with general motion with the weak- and para-perspective cameras (SfM-WP and SfM-PP). This is evident by parameter counting: the weak-perspective camera has 6 DoFs per view, so the 6 constraints from the inter-view affine homography is insufficient to resolve both the camera DoFs and unknown structure DoFs. The same is true of the para-perspective camera with 8 DoFs per view. In contrast, the orthographic camera has 5 DoFs per view, so we can accumulate structure constraints using multiple views. Equivalently, we can solve plane-based SfM with a weak-perspective camera if the weak-perspective scale factor is approximately constant in a set of views. In practice, this implies the structure has approximately similar depth in these views.

There are some previous methods that solve PSfM-O but only in special configurations. Solutions for the case of three views of three non-colinear points were presented in [HB86; HL89]. It was shown that there are in general two structure solutions (up to reflections) and each structure solution yields 8 camera pose solutions with two solutions per view. These correspond to the flip ambiguity in plane-based pose estimation shown in Figure 1.5. However, these methods have little practical value because they require the reprojection constraints to be satisfied exactly. In general they fail to find a real-valued solution when there is noise. A substantial improvement was made by [TJK10] that handles 3 noisy points and 4 or more views. A closed-form solution was developed that relaxes the problem with a linear system that is solved with LLS. However, this has several limitations: it cannot handle the minimal case of 3 views nor more than 3 noisy points, and it cannot handle cases when there is more than one structure solutions. Furthermore, the linear relaxation makes it relatively sensitive to noise. Similarly to perspective cameras, SfM-O can be optimized with BA. However, this is not a closed-form solution and it requires a good initial solution to avoid local minima.

2.5 Shape-from-Template (SfT)

2.5.1 Template modeling

Template modeling is an important aspect of any SfT method, and it is normally broken into three components: shape, appearance and deformation modeling. We now provide a brief overview of these aspects from previous works. Template modeling is required for both geometric SfT methods and recent methods using CNNs with a calibrated perspective camera.

2.5.1.1 Shape modeling

The shape model represents the object’s geometry in a rest position and it can be created from various data sources. Three common sources are (i) an SfM+MVS reconstruction using a moving monocular camera, (ii) a Computer Assisted Design (CAD) model and (iii) a textured depth-map from an RGBD camera. SfM+MVS is often preferred because a CAD model may be unavailable, and the template can be constructed with the same monocular camera as SFT. There are two main types of shape models: surface-based models [SHF07; PHB11; CPB14a; Bar+15; CB15; Yu+15] that represent only the object’s surface as a thin shell and volume-based models [Par+15] that represent both the surface and object interior. Surface models are usually sufficient for applications where only deformation of the surface is of interest, such as those described in Figures 1.7, 1.8, 1.9 and 1.10. Various surface models have been considered including meshes (most commonly) [SHF07; SF09; CB15; Yu+15; Mag+15], b-splines [Bru+10] and Thin Plate Splines [Bar+15; CPB14a]. Volume templates have advantages by modeling and recovering interior deformation. This enables important applications such as medical augmented reality, where the movement of internal structures such as vessels or tumors is the relevant information to the surgeon (Figure 1.11).

2.5.1.2 Appearance modeling

Appearance modeling is required to register the template and there are three main approaches. The first approach extracts repeatable texture features using keypoint detectors such as SIFT [Low04b] from RGB images of the object at rest. This approach can be sufficient for objects with rich, dense texture, producing a dense set of keypoints. For weakly-texture objects, keypoints are sparsely distributed, which can be sufficient for smoothly deforming objects but it may be insufficient to recover complex deformations such as folds. The second main approach uses a texture-map, modeling appearance with one or more RGB images and a dense one-to-one surface mapping. Typically texture-maps are used with triangulated meshes, allowing standard computer graphics libraries such as OpenGL to efficiently render the template with z-buffering. The third approach uses reflectance mapping [Liu+16b; MBC12a], which models the surface’s reflectance function. The purpose is to exploit the shading cue that relates shape, scene illumination, surface reflectance and the photometric response of the camera. Reflectance mapping is considerably more complex than the other two approaches and it generally requires careful control of illumination and camera response.

2.5.1.3 Deformation modeling

Deformation modeling is required to make SFT solvable by restricting the space of template deformations. Three general types of deformation models have been used: *Physical models* respect physical characteristics of the object’s material. *Smoothing models* regularize the problem with the assumption that deformation tends to be smooth. *Statistical models* restrict the deformation space using example deformations from training data.

Physical models. Physical models have been proposed with varying complexity and object specificity. Much inspiration comes from the computer graphics and simulation literature, where a wide range of models exist that range from basic algebraic ones to highly-complex non-linear elastic models from the Finite Element Method. In SFT we are limited to using models that produce tractable problems, where a more physically realistic deformation model is not guaranteed to produce better solutions than a simpler, more constraining model.

The most widely used model by far in SfT is isometry (SfT), preventing significant changes to the metric tensor. This is valid for many objects of interest, including those made of stiff cloth, rubber, paper or cardboard. Isometry has been at the center of much progress because SfT has a unique solution given a calibrated perspective camera (SfT-P) and a registration between the image and surface [Bar+15; SHF07]. Perfectly isometric deformation is not possible by physical objects so it is often imposed softly and called *quasi-isometry*. This is typically implemented with an energy function that penalizes metric tensor changes (*i.e.* strain). A related deformation model is *As-Rigid-As-Possible* (ARAP) [SA07], which is an algebraic model that encourages deformation to be locally rigid, preserving both metric and shape tensors. Quasi-isometry and ARAP impose non-convex constraints on deformation. They can be implemented with various approaches and shape representations. For meshes, a common choice is the preservation of edge-lengths between neighboring vertices. This is normally done by imposing an $L2$ penalty on the change of edge length. ARAP is usually implemented at each vertex by measuring the deviation from rigidity using the motion of neighboring vertices [CB15; Liu+16b]. For algebraic surfaces such as b-splines and TPS, isometry and quasi-isometry have been implemented using the metric tensor expressed analytically in the spline control points [Bar+15].

Other physical models have been considered more recently to handle strong non-isometric deformation. These include conformal models [Bar+15] that preserve angles on the surface, and equi-areal models [Cas+19]. These are much weaker models than isometry and lead to less well constrained SfT problems with ambiguities. For the conformal model, it was shown that, given a registration between the surface and image with known perspective intrinsics, SfT is solvable up to a discrete set of convex-concave ambiguities. For the equi-areal model, SfT is also solvable up to spatially-localized two-fold ambiguities.

Inextensible deformation is another model that has gained interest in SfT [Mor+09; Bru+10; SHF07; PHB11]. This is a convex relaxation of isometry that permits surfaces to shrink but not stretch. Despite lacking physical realism, inextensibility has proven useful as a way to solve SfT-P with a convex relaxation and a closed-form solution. More complex deformation models have been considered based on Finite Element Models (FEMs), including linear elasticity [Mal+13], mass-spring models [Hao+15], co-rotational elements [Hao+15] and Saint Venant-Kirchhoff elements [Hao+15; Col+16a]. However, in order to recover strongly non-isometric deformation, boundary conditions are required such as 3D anchors (points on the surface with known 3D positions in camera coordinates), which are often unavailable. Without them, the ability of these models to resolve strongly non-isometric deformation appears limited.

Smoothing models. Unlike physical models, smoothing models encode the assumption that deformation is generally smooth. They are usually used in conjunction with physical models to add additional regularization and to help convexify the inference problem. Smoothing models have been implemented in one of two main ways. The first is to impose it explicitly with dimensionality reduction. For example, this has been achieved with spline models by reducing the number of control points. For meshes, this has been achieved by [Ost+12] using the mesh Laplacian. The second approach is to impose smoothness with a penalty terms that penalizes non-smooth deformation. This is normally based on the thin-plate energy which penalizes strong curvature *e.g.* [Bru+10], or thin-shell energy which penalizes strong curvature change relative to the template’s rest shape *e.g.* [CB15]. The advantage of dimensionality reduction is to significantly reduce the number of unknowns. The disadvantage is that it usually reduces constraint sparsity, which may increase optimization cost. An advantage of a penalty term is that it can be locally deactivated to allow recovery of unusually high

curvature change such as surface creases [GCB16c].

Statistical models. These models learn a low-dimensional deformation space from training examples. In SfT, statistical models have first been used to learn deformation at local surface regions [SF11]. For objects that deform in predictable ways, they can capture a more compact deformation space compared to physical or smoothing models. The main disadvantage is the requirement for a training process, which may not be feasible in many real-world settings.

2.5.2 Registration

Registration between the template’s surface and the camera image is a key component in SfT. Most SfT methods can be divided into two categories: those that assume registration is provided by an external process (*decoupled registration*) [SF11; Hao+15; Bar+15; SHF07], and those that perform registration jointly with inference (*integrated registration*) [CB15; Liu+16b; Yu+15; Ost+12].

2.5.2.1 Decoupled registration

Most of the earliest SfT methods are in this category. They assume a set of point correspondences are provided as inputs. Various approaches have been used to compute point correspondences, including keypoint-based matching, feature tracking with video inputs such as KLT [TK91] and dense optical flow-based tracking such as [SBK10]. Some methods assume the correspondences are outlier-free, while others assume there is a proportion of outliers (mis-matches) that are unknown prior to reconstruction. Outliers have been handled with two main approaches. In the first approach they are detected and removed using motion consistency assumptions *e.g.* [CB14b]. In the second approach they are handled during inference using a robust penalty term that tolerates outliers *e.g.* [Bru+10].

2.5.2.2 Integrated registration

The main limitation of decoupled methods is that they must use general registration techniques that do not exploit all the constraints available from the template. Integrated methods exploit this information by inferring deformation jointly with registration. This has mostly been performed using iterative *guided matching*, where deformation estimation is interleaved with updating registration. This has been achieved with three main styles of approach. The first, called *direct approaches* [Wan+16; Yu+15] use photo-consistency constraints similar to those used in classical optical flow algorithms. It works by measuring photometric dissimilarity between the image and the registered template surface, then updating deformation to decrease photometric dissimilarity. This has been implemented by iterative re-linearization of the image. The main advantage is the use of dense texture information, however it is generally limited by a narrow convergence basin. An alternative solution was proposed in [CB15] where there was no image linearization. A render of the template is registered to the image using a technique inspired by optical flow block matching. The render is quantized into small windows and each window is matched to the image using a quasi-exhaustive search. This process can be parallelized easily on modern GPUs, making it real-time. This has the advantage of a wider basin of convergence while also using dense texture information. The third approach to integrated registration uses keypoint matching [Ost+12] where outliers are iteratively identified and eliminated using an M-estimator. The M-estimator is designed to become increasingly sensitive with each deformation iteration, to prevent inliers being rejected early [PLF08]. The main advantage is a wide convergence baseline, however it is limited by the fundamental limits of keypoint-based registration.

2.5.3 Closed-form SfT-P solutions

One of the main breakthroughs for solving SfT-P in closed-form is the convex relaxation of isometry to inextensibility [PHB11; SHF07]. The problem is transformed to finding the deformation that maximizes the depth of matched points such that the Euclidean distance between point pairs does not exceed their geodesic distance defined on the template surface. This has been solved using a greedy method [SHF07] and with second order cone programming (SOCP) using the interior point method [SHF07; PHB11]. When the perspective effects are strong and there are many points, solutions can be very accurate, however accuracy deteriorates when perspective effects and/or number of points are reduced [CPB14a]. SfT-P can also be solve in closed-form using 1st-order PDEs [Bar+15]. The PDE uses registration constraints (linking surface depth, orientation and camera projection with registration) and isometry constraints expressed by the metric tensor. This has a point-wise solution, where the depth at each surface point can be solved analytically and uniquely. The advantage of this method is very low computational cost and high parallelization, allowing real-time solutions on the CPU. However, its main limitations is that a good registration is required, which can be hard for poorly-textured surfaces. Furthermore, it does not produce optimal solutions in general because the PDE relaxes geometric constraints (it treats depth and depth gradient as independent variables). The approach can also solve conformal deformation [Bar+15] up to global depth and local convex/concave ambiguities.

2.5.4 Optimization-based SfT-P solutions

These methods take as input an initial and sub-optimal solution, and perform iterative numerical optimization of a non-convex cost function [Liu+16b; Yu+15; Ost+12; CB15; Yu+15; MBC11; SUF08]. Practically all methods use a pseudo *Maximum a Posteriori* cost function consisting of prior and data terms. The prior term encodes agreement with a deformation model as described in 2.5.1.3. Other priors such as temporal continuity may also be included when processing video data. The data term encodes agreement between the deformed model and image evidence, such as the reprojection of feature matches [SHF07; PHB11; CPB14a; Bar+15], patch-based matches [CB15] or pixel-level photo-consistency [Ngo+15; Yu+15; MBC11] or the locations of surface borders [ISF07; VA13; Col+16a; GCB16a]. The main advantages of optimization-based methods is that they can use complex cost functions with no known closed-form solution. When properly initialized, they generally produce the most accurate solutions. Initialization can be performed using a closed-form method, or in the case of video data, with the solution from the previous frame (also called *frame-to-frame tracking*). There are three main open challenges with optimization-based methods. The first is to increase the convergence baseline, to reduces the dependency on good initialization. Methods such as coarse-to-fine optimization with multi-resolution meshes [Yu+15], or advanced schemes using geometric multi-grid [CB15] have proved useful. The second challenge is to reduce the cost of optimization for real-time solutions. This has been achieved with dimensionality reduction and GPU implementations such as [CB15]. The third challenge is designing a cost function that works well in a broad range of settings without requiring fine-tuning of hyper-parameters.

2.5.5 CNN-based solutions

CNNs have been used to with great success for solving monocular reconstruction problems with deformable objects, such as 3D human pose estimation [Mar+17; GNK18], surface normal reconstruction

[BRG16; WFG15] and monocular depth estimation [EF15; Gar+16a; Liu+16a]. These works have stimulated recent progress for solving SfT with CNNs [Pum+18; Gol+18; Fue+18; Fue+21]. The main idea is to train a CNN to learn the function that maps a single RGB image with known camera intrinsics to the template’s deformation parameters. The CNNs in these works are trained using supervised learning with labeled data *i.e.* pairs of RGB images with the corresponding deformation parameters. Acquiring labeled data is a main practical challenge and it is practically impossible to obtain with real data. For this reason, these works rely heavily on simulated labeled data generated by rendering software such as Blender. On one hand, this offers a way to generate an enormous amount of training data. On the other hand, this opens up new challenges to ensure that the training data represents the variability and realism of real-world images. The so-called render gap is a term used to express the difference in realism between simulated and real data, and it affects the ability of the CNN to generalize well to real data. In SfT we additionally face the problem that the space of possible deformations can be exceptionally large, making it difficult to cover the deformation space sufficiently with training data. For this reason, these works have been shown to work with objects undergoing simple, smooth deformation with a low-dimensional deformation space such as bending paper sheets or smoothly deforming cloth. Furthermore, these works require intrinsically calibrated images. There has been some recent progress for combining labeled simulated data with partially labeled real data in order to reduce the render gap [Fue+18]. The real data is acquired by a standard RGBD camera. This data does not contain sufficient information to train the CNN with supervised learning because RGBD images provide depth information but not registration information (the spatial alignment between the RGBD image and the template’s surface). Consequently, the CNN was trained with a combination of supervised learning (to learn the template’s depth) and unsupervised learning (to learn the template’s registration). Unsupervised learning was implemented using a photometric loss similar to multi-scale normalized cross-correlation. While this work marks a good step forward to solving SfT with CNNs in the wild, it requires calibrated RGBD data, so it is not applicable for solving fSfT. Furthermore, it requires a CNN to be trained specifically for each template, which is a strong limitation in terms of practical applicability and computational resources. This directly contrasts our approach to SfT and fSfT, which does not require a computationally-intensive training process for each template, making it much easier to apply in real applications. Very recently, a CNN-based approach has been presented that eliminates the need to train for a specific template texture [Fue+21]. This is promising work, however it only works for flat, rectangular surfaces such as a sheet of paper. This contrasts our approach to fSfT which handles templates with any shape or texture.

Plane-based Structure-from-Motion with Affine Cameras

Chapter summary and organization

This chapter presents our novel closed-form solutions and theoretical analysis for Plane-based Structure-from-Motion with Affine Cameras (PSfM-A). In §1.3.1.1 of the introduction chapter we have provided the problem background, motivation and applications. In §1.3.1.2 we have provided an overview of this chapter's technical and theoretical contributions. This chapter is organized into five main sections. In §3.1 we give the geometric problem setup, we explain why PSfM can only be solved with a restricted class of affine cameras, and we give an overview of our technical solution with the orthographic camera (PSfM-O). In §3.2 we give full technical details of our two methods for solving PSfM-O. They both work by upgrading an affine scene reconstruction to a metric scene reconstruction by solving a set of non-convex upgrade constraints in closed-form. The first method (Exact-PSfM-O) solves the upgrade exactly and it handles the minimal case of three views. The second method (Approx-PSfM-O) solves the upgrade in a least-squares sense and it handles the general case of three or more views. Approx-PSfM-O is the method we use in practice, and Exact-PSfM-O is used to perform theoretical problem analysis. In §3.3 we present this as 8 new theorems that significantly expand our knowledge of the problem. We also present a comprehensive table that compares solving SfM with affine cameras for non-planar versus planar structures, detailing differences in the upgrade problems and degenerate geometries. In §3.4 we present an empirical evaluation of Approx-PSfM-O and compare it with prior state-of-the-art methods using extensive simulated and real image data. In §3.5 we conclude this chapter with a chapter summary. Open research opportunities stemming from this work are described in the thesis conclusion chapter in §6.2.1.2. In Appendix A.3.1 we provide proofs of all new theorems presented in §3.3.

3.1 Problem setup and solution overview

3.1.1 Scene geometry and notation

The scene geometry of PSfM is illustrated in Figure 3.1. We define as M the number of views in the scene indexed by $i \in \{1, 2, \dots, M\}$. We define as N the number of points in the scene indexed by $j \in \{1, 2, \dots, N\}$. We define as $\mathbf{s}_j \in \mathbb{R}^3$ the unknown location of the j^{th} point in world coordinates. We define as $\mathbf{S} \stackrel{\text{def}}{=} [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N] \in \mathbb{R}^{3 \times N}$ the unknown *structure matrix* that holds the points in world coordinates. We define the *structure plane* as the support plane that passes through the structure points in world coordinates. Without loss of generality we defined this as the plane $z = 0$. Consequently, the third row of \mathbf{S} is all-zeros. Without loss of generality we define the centroid of the structure points to be at the origin of world coordinates. We define as $\mathbf{V} \in \{0, 1\}^{M \times N}$ the binary visibility matrix where $\mathbf{V}_{ij} = 1$ if we have a correspondence for point j in view i , and $\mathbf{V}_{ij} = 0$ otherwise.

The following spectral definitions of $\mathcal{SS}_{2 \times 2}$ and $\mathcal{G}_{2 \times 2}$ are used in this chapter:

$$\begin{aligned} \mathbf{A} \in \mathcal{SS}_{2 \times 2} &\Leftrightarrow s_1(\mathbf{A}) = 1 \Rightarrow (s_2(\mathbf{A}) = |\det(\mathbf{A})|) \\ \mathbf{G} \in \mathcal{G}_{2 \times 2} &\Leftrightarrow (\mathbf{G} \in \mathcal{SS}_{2 \times 2}, \mathbf{G} \succeq \mathbf{0}) \Leftrightarrow (s_1(\mathbf{G}) = \lambda_1(\mathbf{G}) = 1, s_2(\mathbf{G}) = \lambda_2(\mathbf{G}) = \det(\mathbf{G})) \end{aligned} \quad (3.1)$$

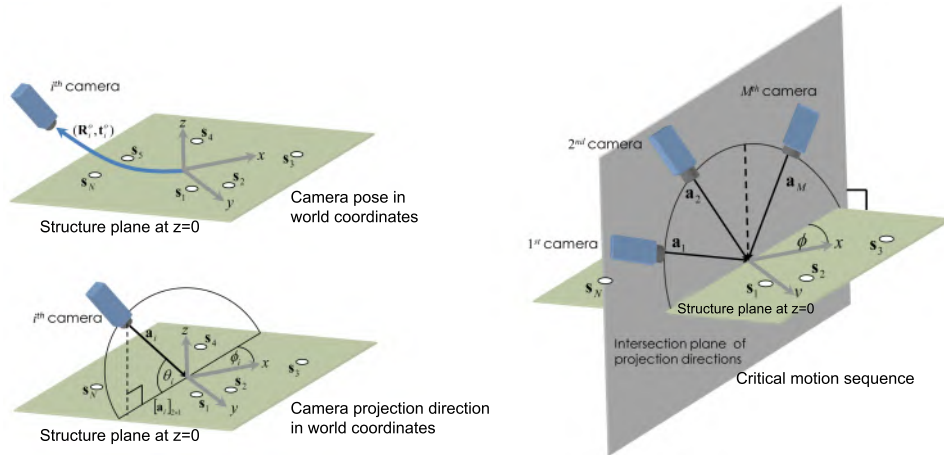


Figure 3.1: Scene geometry of PSfM with a moving affine camera viewing a planar structure (left). Constant azimuth motion is shown in the right diagram. The vector \mathbf{a}_i denotes the projection direction of the i^{th} camera. Constant azimuth motion is a critical motion sequence of PSfM-O that causes it to be ill-posed. However, it is well-posed in the general case. Furthermore, constant azimuth motion is the only critical motion sequence for PSfM-O as stated in Theorem 1.

We define as \mathbf{q}_i^j the position of the j^{th} point in the i^{th} image. We define as $\hat{\mathbf{q}}_i^j$ a noisy measurement of \mathbf{q}_i^j . We define as $\mathbf{M}_{i \in [1, M]}$ the 2×4 projection matrix of the affine camera for view i . The relationship between $\mathbf{M}_{i \in [1, M]}$, $\hat{\mathbf{q}}_i^j$ and \mathbf{s}_j^j is

$$\hat{\mathbf{q}}_i^j = \mathbf{M}_i \text{stk}(\mathbf{s}_j, 1) + \epsilon_i^j \quad (3.2)$$

where $\epsilon_i^j \in \mathbb{R}^2$ denotes error from measurement noise and camera approximation. Because the z -component of \mathbf{s}_j is 0, Equation (3.2) simplifies to

$$\hat{\mathbf{q}}_i^j = \mathbf{M}'_i \text{stk}([\mathbf{s}_j]_{2 \times 1}, 1) + \epsilon_i^j \quad (3.3)$$

where \mathbf{M}'_i is a 2×3 sub-projection matrix formed by staking columns 1, 2 and 4 of \mathbf{M}_i .

3.1.2 Projection model instantiation

To reconstruct the scene, we must resolve structure and the camera projection matrices using reprojection equations given in Equation (3.3). This first requires instantiating \mathbf{M}_i with an affine projection model. The three main types are the orthographic, weak-perspective and para-perspective models. From their definitions in Equation (2.7), \mathbf{M}'_i is defined as follow:

$$\mathbf{M}'_i = \begin{cases} \begin{bmatrix} \alpha[\mathbf{R}_i]_{2 \times 2} & [\mathbf{t}_i]_{2 \times 1} \end{bmatrix} & \text{(orthographic)} \\ \begin{bmatrix} \alpha_i[\mathbf{R}_i]_{2 \times 2} & [\mathbf{t}_i]_{2 \times 1} \end{bmatrix} & \text{(weak-perspective)} \\ \begin{bmatrix} \alpha_i \begin{bmatrix} 1 & \gamma_i \\ 0 & \beta_i \end{bmatrix} [\mathbf{R}_i]_{2 \times 2} & [\mathbf{t}_i]_{2 \times 1} \end{bmatrix} & \text{(para-perspective)} \end{cases} \quad (3.4)$$

We plug Equation (3.4) into the reprojection constraints provided by Equation (3.3) to form three different reconstruction problems: PSfM-O, PSfM-WP and PSfM-PP with the orthographic, weak-perspective and para-perspective cameras respectively.

Unlike SfM with affine cameras and non-planar structures, we cannot solve PSfM with the weak- or para-perspective cameras only from reprojection constraints. This is apparent by parameter counting. With the weak-perspective camera we have 6 unknown DoFs in \mathbf{M}'_i : 5 unknown pose DoFs plus an unknown image magnification factor $\alpha_i \in \mathbb{R}^+$. However, Equation (3.3) does not provide sufficient independent constraints to resolve these 6 unknown DoFs per view: If structure were known, then Equation (3.3) would provide at most 6 independent constraints per view on \mathbf{M}'_i . This leaves no extra constraints with which to resolve structure. With the para-perspective camera, we have 8 unknown DoFs in \mathbf{M}'_i : 5 pose DoFs, image magnification α_i , skew γ_i and aspect ratio β_i . Consequently, even if structure were known, we would not have enough constraints to resolve the 8 unknown DoFs in \mathbf{M}'_i .

In contrast, PSfM-O is solvable from reprojection constraints. The orthographic camera's image magnification factor $\alpha \in \mathbb{R}^+$ is constant for all views by definition. The absolute scale of structure is not recoverable thanks to the ambiguity between α and the structure's scale. This ambiguity is fixed by arbitrarily setting $\alpha = 1$. Consequently, SfM-O has 5 unknown camera DoFs per view: 3 for rotation and 2 for the x and y components of translation. The z component of translation is not recoverable because it is unconstrained by reprojection using an affine camera. Consequently, the reprojection equations can provide 1 extra constraint per view with which to resolve structure. For this reason, we focus this chapter on solving PSfM-O and we then introduce some extensions to adapt the solution to PSfM-WP and PSfM-PP in special cases.

3.1.3 Why previous stratified methods cannot solve PSfM-O

We now demonstrate why the classic stratified approach of [TK92], used for reconstructing non-planar structures with affine cameras, cannot work with planar structures. We show it algebraically in this section and we show it empirically in the experimental section of this chapter. The approach works in 3 steps as follows:

Step 1: Affine reconstruction. Assuming complete measurements, let $\tilde{\mathbf{q}}_i^j \stackrel{\text{def}}{=} \hat{\mathbf{q}}_i^j - \sum_{k=1}^N \hat{\mathbf{q}}_k^j$ be the zero centered correspondence of the j^{th} point in the i^{th} view. Measurements stack to form the $2M \times N$ measurement matrix $\hat{\mathbf{Q}}$ that factorises as follows:

$$\hat{\mathbf{Q}} = \begin{bmatrix} \tilde{\mathbf{q}}_1^1 & \tilde{\mathbf{q}}_1^2 & \dots & \tilde{\mathbf{q}}_1^N \\ \tilde{\mathbf{q}}_2^1 & \tilde{\mathbf{q}}_2^2 & \dots & \tilde{\mathbf{q}}_2^N \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{q}}_M^1 & \tilde{\mathbf{q}}_M^2 & \dots & \tilde{\mathbf{q}}_M^N \end{bmatrix} = \text{stk}([\mathbf{M}_1]_{2 \times 3}, [\mathbf{M}_2]_{2 \times 3}, \dots, [\mathbf{M}_M]_{2 \times 3})\mathbf{S} + \varepsilon \quad (3.5)$$

where $\varepsilon \in \mathbb{R}^{2M \times N}$ denotes measurement noise. The optimal affine reconstruction is then computed as the rank-3 approximation of $\hat{\mathbf{Q}}$ using the SVD: $\hat{\mathbf{Q}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where \mathbf{U} and \mathbf{V} are unitary matrices and $\mathbf{\Sigma}$ is the diagonal singular value matrix sorted by decreasing magnitude.

Step 2: Assemble upgrade equations. We then relate the affine reconstruction and metric reconstruction with an unknown 3×3 upgrade matrix \mathbf{Y} . Ignoring noise, this writes as follows:

$$\begin{aligned} \text{stk}([\mathbf{M}_1]_{2 \times 3}, [\mathbf{M}_2]_{2 \times 3}, \dots, [\mathbf{M}_M]_{2 \times 3}) &= [\mathbf{U}]_{2M \times 3}\mathbf{Y} & (a) \\ \mathbf{S} &= \mathbf{Y}^{-1}[\mathbf{\Sigma}]_{3 \times 3}[\mathbf{V}]_{N \times 3}^\top & (b) \end{aligned} \quad (3.6)$$

The task is then to find \mathbf{Y} using constraints from the camera model. For the orthographic camera we have $[\mathbf{M}_i]_{2 \times 3} \in \mathcal{S}_{2 \times 3} \Leftrightarrow \mathbf{U}_i\mathbf{Y} \in \mathcal{S}_{2 \times 3} \Leftrightarrow \mathbf{U}_i\mathbf{Y}\mathbf{Y}^\top\mathbf{U}_i^\top = \mathbf{I}_2$, where \mathbf{U}_i denotes the i^{th} 2×3 sub-block of $[\mathbf{U}]_{2M \times 3}$. This imposes linear equality constraints on the positive definite matrix $\mathbf{Z} \stackrel{\text{def}}{=} \mathbf{Y}\mathbf{Y}^\top$.

Step 3: Solve \mathbf{Z} with LLS and recover \mathbf{Y} . We then relax the condition that \mathbf{Z} is positive definite to it being symmetric. The upgrade constraints are then satisfied in the least-squares sense with a LLS system. This is then solved and if the solution for \mathbf{Z} is positive definite, \mathbf{Y} is then recovered with the Cholesky decomposition (up to an arbitrary and irrecoverable unitary transform). Finally, Equation (3.6) is used to recover \mathbf{S} and the camera projection matrices.

What goes wrong when structure is planar. This process works well for non-planar structures however it fails completely for planar structures, and it is extremely unstable for quasi-planar structures. For planar structures, the maximum theoretical rank of \mathbf{S} is two, so the third column of \mathbf{U} becomes all-zeros. The upgrade equations then do not fully constrain \mathbf{Z} . Indeed, they only constrain $[\mathbf{Z}]_{2 \times 2}$ as follows:

$$\mathbf{U}_i \begin{bmatrix} [\mathbf{Z}]_{2 \times 2} & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 0 \end{bmatrix} \mathbf{U}_i^\top = \mathbf{I}_2 \quad (3.7)$$

Because only 3 of the 6 DoFs of \mathbf{Z} are constrained, it cannot be recovered with this approach.

3.1.4 Overview of proposed solution

We present a fast, closed-form and stratified solution to PSfM-O that correctly handles planar structures. This is the first general solution that works with three or more views with three or more noisy point correspondences, without making a linear relaxation of the problem. It works as follows. First the 2D affine structure of the points is estimated from point correspondences. This has a closed-form solution using SVD, provided that there are no missing correspondences. Furthermore, the SVD solution is also the ML estimate. When there are missing correspondences, the 2D affine structure is

found by applying a fill-in technique followed by the SVD and iterative refinement. Next, we solve globally a set of non-convex upgrade constraints that converts the 2D affine structure to its metric structure. Finally, for each metric structure solution, the corresponding camera poses are recovered by an optimal plane-based pose estimation algorithm.

We present two variants of our approach that serve two different purposes. The first, called *Exact-PSfM-O*, solves the upgrade constraints exactly, and it is mainly used to answer core theoretical questions about the problem. The second, called *Approx-PSfM-O*, solves the upgrade constraints in a least-squares sense, and is the method we use in practice. We list here some important properties of Approx-PSfM-O:

1. Approx-PSfM-O solves the general PSfM-O problem. This is when there are three or more views, the structure has three or more points and when there are missing correspondences. No previous method could do this.
2. Approx-PSfM-O generates all solutions when there is no exact physical interpretation of the data due to noise or modeling approximation error. No previous method could do this.
3. Extensive empirical evaluation shows that there appears to be no clear benefit in using Bundle Adjustment to refine the solutions from Approx-PSfM-O. This is a remarkable result because Approx-PSfM-O does not directly optimize the reprojection error. In the special case of three points our solutions are consistently more accurate than previous state-of-the-art [TJK10] (which only handles the special case of three points and four or more views).

We also extend our approach to solve special cases of PSfM-WP and PSfM-PP. Specifically, if within the set of views we have three or more views where the depth of the structure is similar, PSfM-WP can be converted to a PSfM-O problem and solved/analyzed with our approach. If we also have an intrinsically calibrated perspective camera, then we can use it to derive a PSfM-PP problem that can also be converted to a PSfM-O problem and solved/analyzed with our approach.

3.2 PSfM-O technical solution

3.2.1 Upgrade constraints

Our approach formulates and solves *non-linear* upgrade equations from a rank-2 factorization of $\hat{\mathbf{Q}}$. We first consider the case when correspondences are measured in all views. The upgrade constraints are the same if there are missing correspondences but the method to compute the rank-2 factorization is different, as discussed in §3.2.4.4.

From Equation (3.5) and using the fact that a planar structure implies the third row of \mathbf{S} is all-zeros, the maximum theoretical rank of $\hat{\mathbf{Q}}$ is 2 and it decomposes as follows

$$\hat{\mathbf{Q}} = \text{stk}([\mathbf{M}_1]_{2 \times 2}, [\mathbf{M}_2]_{2 \times 2}, \dots, [\mathbf{M}_M]_{2 \times 2}) [\mathbf{S}]_{2 \times N} + \varepsilon \quad (3.8)$$

The two factors can be recovered up to noise and an unknown 2×2 upgrade matrix \mathbf{X} using the rank-two approximation of $\hat{\mathbf{Q}}$ with the SVD $\hat{\mathbf{Q}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. The least-squares rank-2 approximation is

$$\begin{aligned} \hat{\mathbf{Q}} &\approx \mathbf{M}^A \mathbf{S}^A \\ \mathbf{M}^A &\stackrel{\text{def}}{=} [\mathbf{U}]_{2M \times 2} [\mathbf{\Sigma}]_{2 \times 2}^{1/2} \\ \mathbf{S}^A &\stackrel{\text{def}}{=} [\mathbf{\Sigma}]_{2 \times 2}^{1/2} [\mathbf{V}]_{N \times 2}^\top \end{aligned} \quad (3.9)$$

This divides $\hat{\mathbf{Q}}$ into two factors: $\mathbf{M}^A \in \mathbb{R}^{M \times 2}$ is the affine camera factor and $\mathbf{S}^A \in \mathbb{R}^{2 \times N}$ is the affine structure factor. These relate to the camera projection and metric structure matrices with

$$\begin{aligned} \text{stk}([\mathbf{M}_1]_{2 \times 2}, [\mathbf{M}_2]_{2 \times 2}, \dots, [\mathbf{M}_M]_{2 \times 2}) &\approx \mathbf{M}^A \mathbf{X} & (a) \\ [\mathbf{S}]_{2 \times N} &\approx \mathbf{X}^{-1} \mathbf{S}^A & (b) \end{aligned} \quad (3.10)$$

We now instantiate \mathbf{M}_i with the orthographic camera for all views, to obtain constraints on \mathbf{X} . Recall that absolute scale cannot be reconstructed so we arbitrarily set the orthographic camera's magnification factor k to 1. From Equation (3.4), in the absence of noise each view provides the following constraint on \mathbf{X} :

$$[\mathbf{R}_i]_{2 \times 2} = \mathbf{M}_i^A \mathbf{X} \Leftrightarrow \mathbf{M}_i^A \mathbf{X} \in \mathcal{SS}_{2 \times 2} \Leftrightarrow s_1(\mathbf{M}_i^A \mathbf{X}) = 1 \quad (3.11)$$

where \mathbf{M}_i^A denotes the i^{th} 2×2 sub-block of \mathbf{M}^A . We can also express this constraint in terms of the Gramian matrix $\mathbf{W} \stackrel{\text{def}}{=} \mathbf{X} \mathbf{X}^\top$. From Equation (3.1) we have $\mathbf{M}_i^A \mathbf{X} \in \mathcal{SS}_{2 \times 2} \Leftrightarrow \mathbf{M}_i^A \mathbf{W} \mathbf{M}_i^{A\top} \in \mathcal{G}_{2 \times 2}$. Therefore the equivalent constraint on \mathbf{W} is

$$\lambda_1(\mathbf{M}_i^A \mathbf{W} \mathbf{M}_i^{A\top}) = 1 \quad (3.12)$$

We refer to Equation (3.12) as the *PSfM-O upgrade constraint*. This provides *one non-convex* equality constraint per view on a 2×2 positive definite upgrade matrix \mathbf{W} . By contrast, the upgrade constraint for non-planar structures (Equation (3.7)) provides *three linear* equality constraints per view on a 3×3 positive definite upgrade matrix \mathbf{Y} .

3.2.2 Upgrade parameterization

We parameterize \mathbf{W} with a vector \mathbf{w} of size 3 as follows:

$$\mathbf{W} = f(\mathbf{w}), \quad f(\mathbf{w}) \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 \\ \mathbf{w}_2 & \mathbf{w}_3 \end{bmatrix}, \quad \mathbf{w}_1 > 0, \quad \mathbf{w}_3 > 0, \quad \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2 > 0 \quad (3.13)$$

where the inequality constraints enforce positive definiteness. We then recover \mathbf{X} from \mathbf{w} up to an arbitrary and irrecoverable 2D unitary transform. This is irrecoverable because for all $\mathbf{G} \in \mathcal{S}_{2 \times 2}$, $\mathbf{W} = (\mathbf{X} \mathbf{G})(\mathbf{X} \mathbf{G})^\top$. We use the following formula to recover \mathbf{X} uniquely by fixing the unitary transform:

$$\mathbf{X} = \begin{bmatrix} \sqrt{\mathbf{w}_1 - \mathbf{w}_2^2 / \mathbf{w}_3} & \mathbf{w}_2 / \sqrt{\mathbf{w}_3} \\ 0 & \sqrt{\mathbf{w}_3} \end{bmatrix} \quad (3.14)$$

3.2.3 Computing affine structure from point correspondences

If point correspondences are complete (all points measured in all images), they are solved with the rank-2 approximation of $\hat{\mathbf{Q}}$ as described in §3.2.1. When there are missing correspondences we cannot factorize $\hat{\mathbf{Q}}$ straightforwardly with the SVD. A common strategy to perform matrix factorization with missing data is to fill-in missing entries with approximate values, compute an initial factorization with the SVD, then refine it by either gradient-based optimization or alternation. In our case, we can fill-in values using the fact that the motion of points from views i to j is a 2D affine transform, defined in

homogeneous coordinates as

$$\mathbf{A}_{ij} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{M}'_j \\ 001 \end{bmatrix} \begin{bmatrix} \mathbf{M}'_i \\ 001 \end{bmatrix}^{-1} \quad (3.15)$$

We can therefore compute \mathbf{A}_{ij} between view pairs and use it to fill-in missing correspondences. The factorization using filled-in values via the SVD is not optimal in the ML sense. We therefore polish it by iterative optimization using LM. The method we use to compute \mathbf{M}^A and \mathbf{S}^A with missing data is given as pseudo-code in Appendix A.3.1.

3.2.4 Exact-PSfM-O: An optimal PSfM-O solution with three views

3.2.4.1 Method overview

Exact-PSfM-O solves PSfM-O with three views by satisfying the upgrade constraints in Equation (3.12) exactly. Given the affine camera factor \mathbf{M}^A (which is 6×2 for three views), Exact-PSfM-O solves the following upgrade problem:

$$\boxed{\begin{array}{l} \text{find } \mathbf{w} \in \mathbb{R}^3 \text{ s.t.} \\ \left\{ \begin{array}{l} \lambda_1(\mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A\top}) = 1, \quad \forall i \in \{1, 2, 3\} \quad (a) \\ f(\mathbf{w}) \succ \mathbf{0} \quad (b) \end{array} \right. \end{array}} \quad (3.16)$$

We show that Problem (3.16) is equivalent to solving a quadratic univariate system. In the absence of noise it has two solutions in general, and with noise it has either zero, one or two solutions. For each solution we recover the upgrade matrix \mathbf{X} from \mathbf{w} using Equation (3.14) and we then estimate the plane's metric structure with $\hat{\mathbf{S}} = \text{stk}(\mathbf{X}^{-1} \mathbf{S}^A, \mathbf{0}_{1 \times N})$. For each $\hat{\mathbf{S}}$ we then resect the cameras, which has in general 2^3 solutions due to a two-fold camera pose ambiguity per view. The maximal number of scene reconstruction solutions is therefore sixteen.

3.2.4.2 Upgrade solution

We first transform the upgrade constraints to equivalent quadratic constraints using the following identity:

$$\lambda_1(\mathbf{A} \in \mathbb{R}^{2 \times 2}) = 1 \Leftrightarrow \begin{cases} \det(\mathbf{A} - \mathbf{I}_2) = 0 & (a) \\ (\lambda_2(\mathbf{A}) = \det(\mathbf{A})) \leq 1 & (b) \end{cases} \quad (3.17)$$

Equation (3.17-a) states that one of the eigenvalues of \mathbf{A} is 1 and Equation (3.17-b) states that 1 is the largest eigenvalue. Plugging the right side of Equation (3.17) into Problem (3.16) gives the equivalent problem

$$\boxed{\begin{array}{l} \text{find } \mathbf{w} \in \mathbb{R}^3 \text{ s.t.} \\ \left\{ \begin{array}{l} \det(\mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A\top} - \mathbf{I}_2) = 0 \quad \forall i \in [1, 2, 3] \quad (a) \\ \det(\mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A\top}) \leq 1 \quad \forall i \in [1, 2, 3] \quad (b) \\ f(\mathbf{w}) \succ \mathbf{0} \quad (c) \end{array} \right. \end{array}} \quad (3.18)$$

Equation (3.18-a) is a quadratic constraint on \mathbf{w} that has the following special form

$$\begin{aligned} a_i \mathbf{w}_1 + b_i \mathbf{w}_2 + c_i \mathbf{w}_3 + d_i (\mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2) &= 1 \\ \mathbf{E}_i &\stackrel{\text{def}}{=} \mathbf{M}_i^{A\top} \mathbf{M}_i^A, \quad a_i \stackrel{\text{def}}{=} [\mathbf{E}_i]_{11}, \quad b_i \stackrel{\text{def}}{=} 2[\mathbf{E}_i]_{12}, \quad c_i \stackrel{\text{def}}{=} [\mathbf{E}_i]_{22}, \quad d_i \stackrel{\text{def}}{=} -\det(\mathbf{E}_i) \end{aligned} \quad (3.19)$$

Using this special form, we can convert it to a quadratic equation in 1 variable. First we introduce the determinant of \mathbf{W} as an auxiliary variable $s \stackrel{\text{def}}{=} \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2$. The following is then equivalent to Equation (3.19):

$$\mathbf{A}_E \text{stk}(\mathbf{w}, s) = \mathbf{1}_{3 \times 1}, \quad \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2 - s = 0 \quad (3.20)$$

where \mathbf{A}_E is a 3×4 matrix holding $[a_i, b_i, c_i, d_i]$ in its i^{th} row. This system has three linear equations and one quadratic equation, so it must have either 0, 1 or 2 real solutions. We solve it by first using the linear constraints to find $\text{stk}(\mathbf{w}, s)$ up to a 1 dimensional affine subspace: $\text{stk}(\mathbf{w}, s) = \text{stk}(\mathbf{w}', s') + \alpha \mathbf{z}$, where \mathbf{z} is any unit null-vector of \mathbf{A}_E , α is an unknown scalar and $(\mathbf{w}' \in \mathbb{R}^3, s' \in \mathbb{R})$ is any solution to $\mathbf{A}_E \text{stk}(\mathbf{w}, s) = \mathbf{1}_{3 \times 1}$. We compute (\mathbf{w}', s') with the Moore-Penrose pseudo-inverse: $\text{stk}(\mathbf{w}', s') = \mathbf{A}_E^\top (\mathbf{A}_E \mathbf{A}_E^\top)^{-1} \mathbf{1}_{3 \times 1}$. We then solve α with the remaining quadratic constraint in Equation (3.20). The solution is all real roots of $a\alpha^2 + b\alpha + c$ where

$$a \stackrel{\text{def}}{=} \mathbf{z}_2^2 - \mathbf{z}_1 \mathbf{z}_3, \quad b \stackrel{\text{def}}{=} \mathbf{z}_4 - \mathbf{w}'_1 \mathbf{z}_3 + 2\mathbf{w}'_2 \mathbf{z}_2 - \mathbf{w}'_3 \mathbf{z}_1, \quad c \stackrel{\text{def}}{=} \mathbf{w}'_2^2 + 1 - \mathbf{w}'_1 \mathbf{w}'_3 \quad (3.21)$$

For each solution to α , \mathbf{w} is recovered with $\mathbf{w} = \mathbf{w}' + \alpha[\mathbf{z}]_{3 \times 1}$. We then test if the solution satisfies Equation (3.18-b,c) and if it does then it solves Problem (3.16).

We present this solution as pseudo-code in Algorithm 1. The upgrade solutions are returned in a solution set \mathcal{W} that can be of size 0, 1 or 2.

Algorithm 1 (The solution to problem (3.16))

Require: $\mathbf{M}^A \in \mathbb{R}^{6 \times 2}$ ▷ the affine camera factor with three views

- 1: **function** exact_PSFMO_upgrade(\mathbf{M}^A)
- 2: $\{\mathbf{M}_1^A, \mathbf{M}_2^A, \mathbf{M}_3^A\} \leftarrow \text{un_stk}(\mathbf{M}^A)$
- 3: $\mathcal{W} \leftarrow \emptyset$ ▷ the set of upgrade solutions
- 4: $\mathbf{E}_i \leftarrow \mathbf{M}_i^{A \top} \mathbf{M}_i^A, i \in \{1, 2, 3\}$
- 5: $\mathbf{A}_E \leftarrow \begin{bmatrix} [\mathbf{E}_1]_{11} & 2[\mathbf{E}_1]_{12} & [\mathbf{E}_1]_{22} & -\det(\mathbf{E}_1) \\ [\mathbf{E}_2]_{11} & 2[\mathbf{E}_2]_{12} & [\mathbf{E}_2]_{22} & -\det(\mathbf{E}_2) \\ [\mathbf{E}_3]_{11} & 2[\mathbf{E}_3]_{12} & [\mathbf{E}_3]_{22} & -\det(\mathbf{E}_3) \end{bmatrix}$
- 6: $\mathbf{z} \leftarrow \text{null}(\mathbf{A}_E)$ with $\mathbf{z}^\top \mathbf{z} = 1$
- 7: $\text{stk}(\mathbf{w}', s') \leftarrow \mathbf{A}_E^\top (\mathbf{A}_E \mathbf{A}_E^\top)^{-1} \mathbf{1}_{3 \times 1}$
- 8: $a \leftarrow \mathbf{z}_2^2 - \mathbf{z}_1 \mathbf{z}_3$
- 9: $b \leftarrow \mathbf{z}_4 - \mathbf{w}'_1 \mathbf{z}_3 + 2\mathbf{w}'_2 \mathbf{z}_2 - \mathbf{w}'_3 \mathbf{z}_1$
- 10: $c \leftarrow \mathbf{w}'_2^2 + s' - \mathbf{w}'_1 \mathbf{w}'_3$
- 11: $\{\alpha_1, \dots, \alpha_L\} \leftarrow \text{real_roots}(a\alpha^2 + b\alpha + c), 0 \leq L \leq 2$
- 12: **for** $l = 1$ to L **do**
- 13: $\mathbf{w} \leftarrow \mathbf{w}' + \alpha_l [\mathbf{z}]_{3 \times 1}$
- 14: **if** $(f(\mathbf{w}) > 0$ **and** $\det(\mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A \top}) \leq 1, \forall i \in \{1, 2, 3\})$ **then**, $\mathcal{W} \leftarrow \mathcal{W} \cup \{\mathbf{w}\}$
- 15: **return** \mathcal{W}

3.2.4.3 Camera resection

Given an upgrade solution from Algorithm 1, we now resect the cameras. Because the upgrade constraints are satisfied exactly, the camera rotation for view i is determined exactly from the upgraded camera factor with $[\hat{\mathbf{R}}_i]_{2 \times 2} = \mathbf{M}_i^A \mathbf{X}$. The full rotation matrix can be completed from $[\hat{\mathbf{R}}_i]_{2 \times 2}$ using orthonormality constraints. This has two solutions in general given in Algorithm 2. This two-fold rotation ambiguity corresponds to the well-known flip ambiguity when estimating the pose of an affine camera relative to a planar structure [ODD96]. The ML translation estimate is as follows:

$$[\hat{\mathbf{t}}_{i \in [1, M]}]_{2 \times 1} = \frac{1}{\sum_{j=1}^N \mathbf{V}_{ij}} \sum_{j=1}^N \mathbf{V}_{ij} \left(\hat{\mathbf{q}}_i^j - [\hat{\mathbf{R}}_i]_{2 \times 2} [\hat{\mathbf{s}}_j]_{2 \times 1} \right) \quad (3.22)$$

Recall that the z component of translation is not recoverable because we use an orthographic camera.

Algorithm 2 (3D rotation matrix completion from its top-left 2×2 sub-matrix)

Require: $\mathbf{A} \in \mathcal{SS}_{2 \times 2}$ ▷ top-left 2×2 sub-matrix of \mathbf{R}_1 and \mathbf{R}_2
 1: **function** rotation.completion(\mathbf{A})
 2: $\begin{bmatrix} r_u & r_v \\ r_v & r_w \end{bmatrix} \leftarrow \mathbf{I}_{2 \times 2} - \mathbf{A}^\top \mathbf{A}$
 3: $\mathbf{b} \leftarrow [\sqrt{r_u}, \text{sign}(r_v) r_u \sqrt{r_u}]$
 4: $\begin{bmatrix} \mathbf{c} \\ a \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^\top \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^\top \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
 5: $\mathbf{R}_1 \leftarrow \begin{bmatrix} \mathbf{A} & +\mathbf{c} \\ +\mathbf{b}^\top & a \end{bmatrix}, \mathbf{R}_2 \leftarrow \begin{bmatrix} \mathbf{A} & -\mathbf{c} \\ -\mathbf{b}^\top & a \end{bmatrix}$
 6: **return** \mathbf{R}_1 and \mathbf{R}_2

3.2.4.4 Full algorithm pseudo-code

Complete pseudo-code for Exact-PSfM-O is given in Algorithm 3, taking point correspondences in three views as inputs. This returns either 0, 1 or 2 metric structure solutions held in the set \mathcal{M} . For the k^{th} structure solution, the corresponding camera poses are held in \mathcal{R}_k and \mathcal{T}_k . The function affine_reconstruct_2D computes the 2D affine reconstruction from point correspondences as described in §3.2.3.

Algorithm 3 (Exact-PSfM-O from point correspondences)

Require: $\{\hat{\mathbf{q}}_i^j\}$ ▷ set of point correspondences with view index $i \in \{1, \dots, M\}$ and point index $j \in \{1, \dots, N\}$
 1:
 2: **function** exact.PSfM.O($\{\hat{\mathbf{q}}_i^j\}$)
 3: $(\mathbf{M}^A, \mathbf{S}^A) \leftarrow \text{affine_reconstruct_2D}(\{\hat{\mathbf{q}}_i^j\})$ ▷ solves the 2D affine scene reconstruction
 4: $\mathcal{W} \leftarrow \text{exact.PSfM.O_upgrade}\{\mathbf{M}^A\}$ ▷ solves upgrade matrices
 5: $\mathcal{S}, \mathcal{R}, \mathcal{T} \leftarrow \emptyset$ ▷ structure, rotation and translation solutions
 6: **for** $k = 1$ to $\text{size}(\mathcal{W})$ **do**
 7: $\mathbf{w} \leftarrow \mathcal{W}_k, \mathbf{X} \leftarrow \begin{bmatrix} \sqrt{\mathbf{w}_1 - \frac{\mathbf{w}_2^2}{\mathbf{w}_3}} & \frac{\mathbf{w}_2}{\sqrt{\mathbf{w}_3}} \\ 0 & \sqrt{\mathbf{w}_3} \end{bmatrix}, \hat{\mathbf{S}} \leftarrow \text{stk}(\mathbf{X}^{-1} \mathbf{S}^A, \mathbf{0}_{1 \times N})$
 8: **for** $i = 1$ to 3 **do**
 9: $[\hat{\mathbf{R}}_i]_{2 \times 2} \leftarrow \mathbf{M}_i^A \mathbf{X}$
 10: $[\hat{\mathbf{t}}_i]_{2 \times 1} \leftarrow \frac{1}{\sum_{j=1}^N \mathbf{v}_{ij}} \sum_{j=1}^N \mathbf{v}_{ij} (\hat{\mathbf{q}}_i^j - [\hat{\mathbf{R}}_i]_{2 \times 2} [\hat{\mathbf{s}}_j]_{2 \times 1})$
 11: $\mathcal{M}_k \leftarrow \hat{\mathbf{S}}, \mathcal{R}_k \leftarrow \{[\hat{\mathbf{R}}_1]_{2 \times 2}, [\hat{\mathbf{R}}_2]_{2 \times 2}, [\hat{\mathbf{R}}_3]_{2 \times 2}\}, \mathcal{T}_k \leftarrow \{[\hat{\mathbf{t}}_1]_{2 \times 1}, [\hat{\mathbf{t}}_2]_{2 \times 1}, [\hat{\mathbf{t}}_3]_{2 \times 1}\}$
 12: **return** $\mathcal{M}, \mathcal{R}, \mathcal{T}$

3.2.5 Approx-PSfM-O: A least-squares PSfM-O solution with three or more views

3.2.5.1 Method overview

Approx-PSfM-O solves metric structure with three or more views by satisfying the upgrade constraints in a least-squares sense. This contrasts Exact-PsfM-O where the upgrade constraints are satisfied exactly, which is generally not possible with noise and more than three views. We show that the least-squares upgrade matrix is found as the roots of a univariate 7^{th} order polynomial. For each upgrade solution, we resect the cameras with the corresponding metric structure using a novel method that is optimal in ML sense.

3.2.5.2 Upgrade problem

Given the affine camera factor \mathbf{M}^A , we minimize the following Sum-of-Squares cost function C based on Equation (3.18):

$$C(\mathbf{w}) \stackrel{\text{def}}{=} \sum_{i=1}^M \det^2(\mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A\top} - \mathbf{I}_2) \quad (3.23)$$

Our goal is to find *all* local minima of C in closed-form and not just its global minimum. This is necessary to handle ambiguous cases where we cannot uniquely resolve metric structure. In ambiguous cases there are more than one solution that satisfy the upgrade constraints up to noise, so the correct solution is not necessarily the one with lowest cost. Instead, by computing all local minima of C we obtain a small number of candidate upgrade matrices, which we will show is at most four in number. We then verify each candidate upgrade matrix by how well it explains the data in terms of reprojection error. All upgrade matrices that can explain the data up to noise are kept. Because $f(\mathbf{w})$ must be positive definite, the domain of C is an open set. This means that for some problems C may not have a local minimum. However, if a local minimum does exist then it must be a local minimum in the domain $\mathbf{w} \in \mathbb{R}^3$. We therefore first find all local minima of $C(\mathbf{w} \in \mathbb{R}^3)$ then we discard the local minima for which $f(\mathbf{w})$ is not positive definite.

3.2.5.3 Upgrade solution

Solution overview. Because C is quartic in \mathbf{w} , the stationary points of C are the roots of three cubic equations, equivalent to the real roots of a univariate 27^{th} -order polynomial. However, we show that C has a special structure that restrict its stationary points to the roots of a univariate 7^{th} -order polynomial. They can then be found very quickly with the SVD of a 7×7 companion matrix, giving either 1, 3, 5 or 7 real stationary points. Furthermore, because C is a Sum-of-Squares polynomial, and therefore is a positive polynomial, there must be either 1, 2, 3 or 4 real-valued local minima. We detect if a stationary point $\hat{\mathbf{w}}$ is a local minimum of C by testing $\frac{\partial^2}{\partial \mathbf{w}^2} C(\hat{\mathbf{w}}) \succ \mathbf{0}$.

Solving the local minima of C . We rewrite C as follows:

$$C(\mathbf{w}) = \sum_{i=1}^M \det^2(\mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A\top} - \mathbf{I}_2) = \|\mathbf{B} \text{stk}(\mathbf{w}, \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2) - \mathbf{1}_{M \times 1}\|_2^2 \quad (3.24)$$

with

$$\mathbf{E}_i \stackrel{\text{def}}{=} \mathbf{M}_i^{A\top} \mathbf{M}_i^A, \quad \mathbf{B} \stackrel{\text{def}}{=} \begin{bmatrix} [\mathbf{E}_1]_{11} & 2[\mathbf{E}_1]_{12} & [\mathbf{E}_1]_{22} & -\det(\mathbf{E}_1) \\ [\mathbf{E}_2]_{11} & 2[\mathbf{E}_2]_{12} & [\mathbf{E}_2]_{22} & -\det(\mathbf{E}_2) \\ \vdots & \vdots & \vdots & \vdots \\ [\mathbf{E}_M]_{11} & 2[\mathbf{E}_M]_{12} & [\mathbf{E}_M]_{22} & -\det(\mathbf{E}_M) \end{bmatrix} \quad (3.25)$$

Using the slack variable $s = \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2$, the stationary points of C are stationary points of the Lagrangian

$$L(\mathbf{w}, s, \nu) \stackrel{\text{def}}{=} \|\mathbf{B} \text{stk}(\mathbf{w}, s) - \mathbf{1}_{M \times 1}\|_2^2 + \nu (\mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2 - s) \quad (3.26)$$

where ν is a Lagrange multiplier for the constraint $s = \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2$. We now show that the stationary points of L are the roots of a 7^{th} -degree polynomial in ν . They are the solutions to $\frac{\partial}{\partial(\mathbf{w}, s, \nu)} L(\mathbf{w}, s, \nu) = \mathbf{0}_{5 \times 1}$ which writes out as follows:

$$\begin{cases} \mathbf{H} \text{stk}(\mathbf{w}, s) - \mathbf{B}^\top \mathbf{1}_{M \times 1} + \nu \text{stk}(\mathbf{F}\mathbf{w}, 1) = \mathbf{0}_{4 \times 1} & (a) \\ w_1 w_3 - w_2^2 - s = 0 & (b) \end{cases} \quad (3.27)$$

with

$$\mathbf{H} \stackrel{\text{def}}{=} \mathbf{B}^\top \mathbf{B}, \quad \mathbf{F} \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (3.28)$$

We now decompose \mathbf{H} with the QL decomposition to give $\mathbf{H} = \mathbf{Q}\mathbf{L}$ where \mathbf{Q} is a 4×4 orthogonal matrix and \mathbf{L} is a lower-triangular 4×4 matrix. Left-multiplying Equation (3.27-a) by \mathbf{Q}^\top and re-substituting $s \leftarrow w_1 w_3 - w_2^2$ gives

$$\mathbf{L} \text{stk}(\mathbf{w}, \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2) - \mathbf{Q}^\top \mathbf{B}^\top \mathbf{1}_{M \times 1} + \nu \mathbf{Q}^\top \text{stk}(\mathbf{F}\mathbf{w}, 1) = \mathbf{0}_{4 \times 1} \quad (3.29)$$

Now consider only the first three rows of Equation (3.29). Because $\mathbf{L}_{3 \times 4} \text{stk}(\mathbf{w}, \mathbf{w}_1 \mathbf{w}_3 - \mathbf{w}_2^2) = \mathbf{L}_{3 \times 3} \mathbf{w}$ (since \mathbf{L} is lower triangular) we have

$$[\mathbf{L}]_{3 \times 3} \mathbf{w} - [\mathbf{Q}]_{4 \times 3}^\top \mathbf{B}^\top \mathbf{1}_{M \times 1} + \nu [\mathbf{Q}]_{4 \times 3}^\top \text{stk}(\mathbf{F}\mathbf{w}, 1) = \mathbf{0}_{3 \times 1} \quad (3.30)$$

Therefore given a solution to ν we can recover \mathbf{w} by solving a linear system using Equation (3.30). This is given after rearrangement by

$$\begin{aligned} \mathbf{w} &= \det^{-1}(M(\nu))g(\nu) \\ M(\nu) : \mathbb{R} &\rightarrow \mathbb{R}^{3 \times 3} \stackrel{\text{def}}{=} [\mathbf{L}]_{3 \times 3} + \nu [\mathbf{Q}]_{3 \times 3}^\top \mathbf{F} \\ g(\nu) : \mathbb{R} &\rightarrow \mathbb{R}^3 \stackrel{\text{def}}{=} \text{adj}(M(\nu)) [\mathbf{I}_3 | \mathbf{0}_{3 \times 1}] (\mathbf{Q}^\top \mathbf{B}^\top \mathbf{1}_{M \times 1} - \nu \mathbf{Q}^\top [0 \ 0 \ 0 \ 1]^\top) \end{aligned} \quad (3.31)$$

where $\text{adj}(M(\nu))$ is the adjoint of $M(\nu)$. We now re-introduce the constraint from the fourth row of Equation (3.27-a):

$$\nu = a - \mathbf{h}_4 \text{stk}(\mathbf{w}, w_1 w_3 - w_2^2) \quad (3.32)$$

where a is the fourth element of $\mathbf{B}^\top \mathbf{1}_{M \times 1}$ and \mathbf{h}_4 is the fourth row of \mathbf{H} . Multiplying both sides of Equation (3.32) by $\det(M(\nu))^2$ and substituting $\det(M(\nu))\mathbf{w} \leftarrow g(\nu)$ gives after simplification:

$$\det(M(\nu))^2 \nu - \det(M(\nu))^2 a + \mathbf{h}_4 \text{stk}(\det(M(\nu))g(\nu), \det(g(\nu))) = 0 \quad (3.33)$$

Equation (3.33) defines a polynomial $p(\nu)$ in ν because $\det(M(\nu))$ and $g(\nu)$ are quadratic and cubic polynomials in ν respectively (and a and \mathbf{h}_4 are constant and known). The polynomial is non-homogeneous in general because $\det(g(\nu))$ is non-homogeneous in ν in general. The polynomial's highest order term is $\det(M(\nu))^2 \nu$, which makes it a 7th-degree polynomial in general. The roots $\{\nu_1, \dots, \nu_L\}$ of $p(\nu)$ are computed efficiently by the SVD of the associated 7×7 companion matrix. For each real root we recover the corresponding value of $\hat{\mathbf{w}}$ using Equation (3.31). We then test if it is a local minimum of C by satisfying $\frac{\partial^2}{\partial \mathbf{w}^2} C(\hat{\mathbf{w}}) \succ \mathbf{0}$. If it is a local minimum we then test if $f(\hat{\mathbf{w}})$ is positive definite. If it passes both these tests then we output $\hat{\mathbf{w}}$ as a candidate upgrade matrix.

Algorithm pseudo-code. We summarize the process to compute all local minima of C with pseudo-code in Algorithm 5. This takes as input the affine camera factor \mathbf{M}^A and it outputs the set \mathcal{W} of all candidate upgrades. The size of \mathcal{W} is either 0, 1, 2, 3 or 4.

3.2.5.4 Camera resection

Resection options. For each candidate upgrade produced by Algorithm 4, we compute the corresponding upgrade matrix \mathbf{X} using Equation (3.14), then we resect the cameras. Unlike Exact-PSfM-O, the upgraded camera factor $\mathbf{M}^A \mathbf{X}$ is *not* guaranteed to exactly correspond to orthographic projection matrices because the metric constraints have been satisfied approximately. There are two ways to deal with this. The first is to correct each upgraded camera matrix to the closest orthographic camera matrix with an algebraic metric *e.g.* the Frobenius norm. However, this correction is not

Algorithm 4 (Solves the local minima of C)

Require: $\mathbf{M}^A \in \mathbb{R}^{2M \times 2}$, $M \geq 3$ ▷ the affine camera factor for M views
 1: **function** local_minima_of_C(\mathbf{M}^A)
 2: $\{\mathbf{M}_1^A, \mathbf{M}_2^A, \dots, \mathbf{M}_M^A\} \leftarrow \text{un_stk}(\mathbf{M}^A)$
 3: $\mathcal{W} \leftarrow \emptyset$ ▷ the set of upgrade solutions
 4: $\mathbf{E}_i \leftarrow \mathbf{M}_i^{A\top} \mathbf{M}_i^A$
 5: $\mathbf{B} \leftarrow \begin{bmatrix} [\mathbf{E}_1]_{11} & 2[\mathbf{E}_1]_{12} & [\mathbf{E}_1]_{22} & -\det(\mathbf{E}_1) \\ [\mathbf{E}_2]_{11} & 2[\mathbf{E}_2]_{12} & [\mathbf{E}_2]_{22} & -\det(\mathbf{E}_2) \\ \vdots & \vdots & \vdots & \vdots \\ [\mathbf{E}_M]_{11} & 2[\mathbf{E}_M]_{12} & [\mathbf{E}_M]_{22} & -\det(\mathbf{E}_M) \end{bmatrix}$
 6: $\mathbf{H} \leftarrow \mathbf{B}^\top \mathbf{B}$, $(\mathbf{Q}, \mathbf{L}) \leftarrow \text{lq}(\mathbf{H})$, $\mathbf{p} \leftarrow \mathbf{Q}^\top \mathbf{B}^\top \mathbf{1}_{M \times 1}$, $\mathbf{F} \leftarrow \begin{bmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{bmatrix}$ ▷ lq denotes the LQ decomposition
 7: $\mathbf{c} \leftarrow P(\mathbf{L}, \mathbf{Q}, \mathbf{p})$, $\mathbf{c} \in \mathbb{R}^8$ ▷ computes coefficients of the degree 7 polynomial $p(\nu)$ in Equation (3.33).
 8: $\{\nu_1, \dots, \nu_L\} = \text{real_roots}(\mathbf{c})$, $L \in \{1, 3, 5, 7\}$ ▷ real roots of $p(\nu)$
 9: **for** $l = 1$ **to** L **do**
 10: $\mathbf{M} \leftarrow \mathbf{L}_{3 \times 3} + \nu_l \mathbf{Q}_{3 \times 3}^\top \mathbf{F}$, $\mathbf{g} \leftarrow \text{adj}(\mathbf{M}) [\mathbf{I}_3 | \mathbf{0}_{3 \times 1}] \left(\mathbf{Q}^\top \mathbf{B}^\top \mathbf{1}_{M \times 1} - \nu_l \mathbf{Q}^\top [0 \ 0 \ 1]^\top \right)$, $\hat{\mathbf{w}} \leftarrow \det(\mathbf{M})^{-1} \mathbf{g}$
 11: **if** $\frac{\partial^2}{\partial \mathbf{w}^2} C(\hat{\mathbf{w}}) > \mathbf{0}$ **then**
 12: $\mathcal{W} \leftarrow \{\hat{\mathbf{w}}\}$ ▷ $\hat{\mathbf{w}}$ is a local minimum of C
 13: **return** \mathcal{W}

optimal in terms of $L2$ reprojection error. A better solution is to take the upgraded metric structure $\hat{\mathbf{S}} = \text{stk}(\mathbf{X}^{-1} \mathbf{S}^A, \mathbf{0}_{1 \times N})$ and use it to resect the cameras with optimal plane-based pose estimation. We find that this second approach works slightly better in practice.

Optimal resection. We have developed two methods to resect the orthographic camera that are optimal in the ML sense. The first method uses Gloptipoly [HLL09] and it is the one described in [CB17]. The second method was published later [BC18] and it is approximately three orders of magnitude faster than the first method. It works by reducing the problem to an irreducible univariate sextic polynomial and it should now be the default method. Unlike the first method, the Prof. Bartoli was the main contributor of the second method. For this reason, we describe and use the first method in this Chapter. We note that in principal resection with the orthographic camera could be solved with a PnP method for generalized cameras, by providing a ray associated to each image point. However, as noted by [Ste18], prior approaches with generalized cameras do not handle parallel rays, which occurs whenever using an orthographic camera, and prior closed-form approaches are not optimal in the ML sense.

For simplicity we drop the view index. We assume there are L point correspondences in the image with $3 \leq L \leq N$. We use $\mathbf{s}_0 \in \mathbb{R}^2$ and $\mathbf{q}_0 \in \mathbb{R}^2$ to denote the point centroids of the structure and image points respectively. We first center the structure and image points by subtracting their centroids. We define the centered point sets by $\{\tilde{\mathbf{s}}_{j \in [1, L]}\}$ and $\{\tilde{\mathbf{q}}_{j \in [1, L]}\}$ respectively. The ML translation estimate between the centered point sets is zero, so we are then just left with estimating rotation. We define the unknown sub-Stiefel matrix $\mathbf{C} \stackrel{\text{def}}{=} [\hat{\mathbf{R}}]_{2 \times 2}$ and our problem is as follows:

$$\begin{aligned}
 & \arg \min_{\mathbf{C} \in \mathcal{SS}_{2 \times 2}} \sum_{j=1}^L \|\mathbf{C}[\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \dots, \tilde{\mathbf{s}}_L] - [\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \dots, \hat{\mathbf{q}}_L]\|_2^2 \\
 & = \arg \min_{\mathbf{C} \in \mathcal{SS}_{2 \times 2}} \text{trace}(\mathbf{C} \mathbf{Z} \mathbf{C}^\top - \mathbf{C} \mathbf{Y})
 \end{aligned} \tag{3.34}$$

where

$$\begin{aligned}
 \mathbf{Z} & \stackrel{\text{def}}{=} [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_L]^\top [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_L] \\
 \mathbf{Y} & \stackrel{\text{def}}{=} 2 [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_L]^\top [\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \dots, \hat{\mathbf{q}}_L]
 \end{aligned} \tag{3.35}$$

From the spectral definition of $\mathcal{SS}_{2 \times 2}$ we have $\mathbf{C} \in \mathcal{SS}_{2 \times 2} \Leftrightarrow \det(\mathbf{C}^\top \mathbf{C} - \mathbf{I}_2) = 0$, $\det(\mathbf{C}) \leq 1$. The optimization problem is therefore one of minimizing a 3-variable quadratic subject to quadratic equal-

ity and inequality constraints that can be solved globally with Gloptipoly. Given \mathbf{C} we reconstruct the two full rotation matrix solutions using Algorithm 2. We then recover the ML translation vectors using Equation (3.22).

3.2.5.5 Full algorithm pseudo-code

Complete pseudo-code for Approx-PSfM-O is given in Algorithm 3. This takes point correspondences in three or more views as inputs and it returns either 0, 1, 2 or 3 metric structure solutions held in the set \mathcal{M} . For the k^{th} structure solution, the corresponding camera poses are held in \mathcal{R}_k and \mathcal{T}_k . The function `affine_reconstruct_2D` computes the 2D affine reconstruction from point correspondences as described in §3.2.3.

Algorithm 5 (Approx-PSfM-O)

```

Require:  $\{\mathbf{q}_i^j\}$  ▷ point correspondences with view index  $i \in \{1, \dots, M\}$  and point index  $j \in \{1, \dots, N\}$ 
1: function approx_PSfM_O( $\{\mathbf{q}_i^j\}$ )
2:    $(\mathbf{M}^A, \mathbf{S}^A) \leftarrow \text{affine\_reconstruct\_2D}(\{\mathbf{q}_i^j\})$  ▷ solves the 2D affine scene reconstruction
3:    $\mathcal{W} \leftarrow \text{local\_minima\_of\_C}\{\mathbf{M}^A\}$ 
4:    $\mathcal{S}, \mathcal{R}, \mathcal{T} \leftarrow \emptyset$  ▷ structure, rotation and translation solutions
5:    $k \leftarrow 0$  ▷ the number of upgrade solutions
6:   for  $l = 1$  to  $\text{size}(\mathcal{W})$  do
7:      $\mathbf{w} \leftarrow \mathcal{W}_l$ 
8:     if  $f(\mathbf{w}) > 0$  then
9:        $k \leftarrow k + 1, \mathbf{X} \leftarrow \begin{bmatrix} \sqrt{\mathbf{w}_1 - \frac{\mathbf{w}_2^2}{\mathbf{w}_3}} & \frac{\mathbf{w}_2}{\sqrt{\mathbf{w}_3}} \\ 0 & \sqrt{\mathbf{w}_3} \end{bmatrix}, \hat{\mathbf{S}} = \text{stk}(\mathbf{X}^{-1}\mathbf{S}^A, \mathbf{0}_{1 \times N})$ 
10:      for  $i = 1$  to  $M$  do
11:         $([\hat{\mathbf{R}}_i]_{2 \times 2}, [\hat{\mathbf{t}}_i]_{2 \times 1}) \leftarrow \text{resect}(\hat{\mathbf{S}}, \{\mathbf{q}_i^1, \mathbf{q}_i^2, \dots, \mathbf{q}_i^N\})$  ▷ orthographic camera resection see §3.2.5.4
12:         $\mathcal{M}_k \leftarrow \hat{\mathbf{S}}, \mathcal{R}_k \leftarrow \{[\hat{\mathbf{R}}_1]_{2 \times 2}, [\hat{\mathbf{R}}_2]_{2 \times 2}, \dots, [\hat{\mathbf{R}}_M]_{2 \times 2}\}, \mathcal{T}_k \leftarrow \{[\hat{\mathbf{t}}_1]_{2 \times 1}, [\hat{\mathbf{t}}_2]_{2 \times 1}, \dots, [\hat{\mathbf{t}}_M]_{2 \times 1}\}$ 
13:   return  $\mathcal{M}, \mathcal{R}, \mathcal{T}$ 
    
```

3.3 Theoretical problem analysis

3.3.1 Section overview

This section presents new results that significantly extend our theoretical understanding of PSfM-O. These results are derived from our technical solutions and they are encapsulated as eight new theorems. Our main contribution is to give the necessary and sufficient geometric conditions for PSfM-O to be degenerate with complete measurements (Theorem 1). This is not trivial and we must geometrically characterize all degeneracies and prove that the characterization is complete. Our second main theoretical contribution is to show that for a general number of orthographic views there can exist up to two metric structure solutions (previously it was assumed to be unique [TJK10]). We give the necessary and sufficient geometric conditions to disambiguate structure with extra views in Theorem 2. We then extend these theorems to incomplete measurements in Theorems 3 and 4. Theorems 5 and 6 give some important theoretical guarantees for Exact-PSfM-O, and Theorems 7 and 8 present conditions for which we can solve PSfM-WP and PSfM-PP. We provide complete proofs of all theorems in Appendix A.3.1.

3.3.2 Definitions

We require some formal definitions of SfM concepts used in this section. A *metric reconstruction* is a reconstruction of the scene up to scale and a coordinate transform (also known as a gauge transform). For SfM problems involving affine cameras, the gauge transform is a rigid transform plus reflection

[Ull79]. An *ambiguity* occurs when there exists more than one metric reconstruction that can exactly satisfy the image measurements when noise is removed and the gauge transform is fixed. Ambiguities can either be *continuous*, where there is an infinite number of solutions or *discrete*, where there is a finite number of solutions. We also divide ambiguities into *structure ambiguities* and *camera resection ambiguities*. A structure ambiguity is when there is more than one structure solution and a camera resection ambiguity is when there is more than one camera pose solution for a given structure solution.

Unlike most other SfM problems, PSfM-O always has both discrete and continuous resection ambiguities. Camera rotation has a discrete two-fold ambiguity per view, corresponding to a reflection of the structure about the camera’s image plane (also called flip or Necker reversal ambiguity) as discussed in §3.2.4.3. Camera translation always has a continuous ambiguity in the depth component. Therefore PSfM-O is never well-posed in the usual sense because it never has a unique solution. Instead, we say that PSfM-O is *well-posed* if (i) structure can be solved up to discrete ambiguities, (ii) camera translation can be solved uniquely apart from its depth component, and (iii) camera rotation can be solved up to the two-fold flip ambiguity. If any one of these conditions is not satisfied then we say PSfM-O is ill-posed or equivalently *degenerate*.

In general we can break down the causes of a degeneracy into four groups. These are *critical structures*, *critical motion sequences*, *missing measurements* and *mixed*. Critical structures are when the ambiguity is caused by the structure being in a particular configuration. Critical motion sequences are when the ambiguity is caused by the camera poses being in a particular configuration. Missing measurement ambiguities are when the ambiguity is caused by one or more views having missing correspondences. Mixed ambiguities are when the ambiguity is caused by a particular combination of structure, camera poses and missing measurements.

We define an *artificial degeneracy* when the problem is well-posed but a particular algorithm cannot solve it because of its design. If an algorithm is guaranteed to not introduce an artificial degeneracy we call it a *Non-Artificially Degenerate Algorithm* (NADA). We say that a solution is *optimal* if it is the ML estimate, equivalent to minimizing the $L2$ reprojection error.

3.3.3 New theoretical results for PSfM-O

3.3.3.1 Full geometric characterization of degenerate scenes with complete measurements

This characterization concerns the camera’s projection directions, which are the directions of the camera’s optical axis in world coordinates as illustrated in Figure 3.1. We denote them by the vectors $\mathbf{a}_i \in \mathbb{R}^3$ in world coordinates where i is the view index. We can define \mathbf{a}_i using polar coordinates with

$$\mathbf{a}_i = \begin{bmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{bmatrix} \quad (3.36)$$

where $\theta \in [0, \pi]$ is the inclination angle relative to the structure plane and $\phi \in [0, 2\pi]$ is the azimuth angle. We define the azimuth angle as the anticlockwise angle between the projection of \mathbf{a} onto the structure plane and the x -axis of world coordinates.

Theorem 1. *A scene is degenerate if and only if at least one of three geometric conditions are satisfied. The first condition is when structure is co-linear. This is the only critical structure in PSfM-O. The second condition is a critical motion sequence which is when the camera projection directions lie on*

a plane that is orthogonal to the structure plane (Figure 3.1). Equivalently, this is when all camera projection directions have the same azimuth. We refer to this as **constant azimuth motion** and it is the only critical motion sequence in PSfM-O. The third condition is when there are fewer than three cameras whose projection directions are unique up to reflection about the structure plane and change of sign. There are no mixed degeneracies in PSfM-O between camera poses and structure.

Corollary 1. We can state Theorem 1 equally in terms of non-degenerate scenes using negation. A scene in PSfM-O is non-degenerate if and only if the structure points are not co-linear, the camera does not have constant azimuth motion, and there are three or more cameras with projection directions that are unique up to reflection about the structure plane and change of sign.

Corollary 2. Trivial examples of degenerate configurations are translation-only camera motion or when the camera only rotates about its optical axis. In both cases the camera projection directions are the same, violating the third condition of Theorem 1.

Corollary 3. We can solve PSfM-O even if the structure plane is orthogonal to the camera's image plane in all views. equivalent to 0° elevation angles. In these configurations, the problem is well-posed whenever structure is not co-linear and there are three or more cameras with projection directions that are unique up to sign. These configurations may not occur exactly in practical cases because it may be impossible to obtain point correspondences when the structure and camera image planes are orthogonal. However, this corollary tells us that the problem will not approach a degenerate configuration as the structure and camera image planes tend to orthogonality.

3.3.3.2 Structure uniqueness given four or more views and complete measurements

Theorem 2. Recall that PSfM-O with three views has at most two solutions for the plane's metric structure. Suppose the scene has three views $i \in \{1, 2, 3\}$, is non-degenerate and has two such solutions. Given an additional orthographic view $i = 4$ we can disambiguate structure if and only if the camera projection directions $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$ are not in a special configuration when projected onto the structure plane. Specifically the outer product of $[\mathbf{a}_4]_{2 \times 1}$ must not be an affine combination of the outer products of $[\mathbf{a}_1]_{2 \times 1}$, $[\mathbf{a}_2]_{2 \times 1}$ and $[\mathbf{a}_3]_{2 \times 1}$. Formally, structure can be disambiguated if and only if:

$$\nexists \alpha, \beta \in \mathbb{R} \text{ s.t. } [\mathbf{a}_4]_{2 \times 1} [\mathbf{a}_4]_{2 \times 1}^\top = \alpha [\mathbf{a}_1]_{2 \times 1} [\mathbf{a}_1]_{2 \times 1}^\top + \beta [\mathbf{a}_2]_{2 \times 1} [\mathbf{a}_2]_{2 \times 1}^\top + (1 - \alpha - \beta) [\mathbf{a}_3]_{2 \times 1} [\mathbf{a}_3]_{2 \times 1}^\top \quad (3.37)$$

In general, given any number of additional orthographic views, structure can be disambiguated if and only if Equation (3.37) holds for at least one of the additional views.

3.3.3.3 Generalization of Theorems 1 and 2 to missing measurements

PSfM-O problems with missing measurements can be partitioned into two types. The first (Type 1) are those where we can complete the correspondence matrix \mathbf{Q} from the incomplete measurements. The second (Type 2) are those where we cannot. Type 1 problems are equivalent to those where we can compute the structure's 2D affine reconstruction, and Type 2 problems to those where we cannot.

Theorem 3. Type 1 problems are degenerate if and only if the equivalent problem with complete measurements is degenerate. Therefore Type 1 problems do not have missing measurement degeneracies. Type 2 problems are always degenerate.

Theorem 4. Suppose the scene has three views, is non-degenerate and has two solutions for the plane's metric structure. If there is at least one additional view which has three or more correspondences

that are non-collinear on the structure plane and Equation (3.37) holds, then we can disambiguate structure. If there is at least one additional view which has two correspondences and Equation (3.37) holds then it may be possible to disambiguate structure from the foreshortening effect in some cases. If all additional views have only one point correspondence then we cannot disambiguate structure.

3.3.3.4 Theoretical Guarantees of Exact-PSfM-O

Recall that Exact-PSfM-O is our solution to PSfM-O for three views that satisfies a set of upgrade constraints exactly. Unlike previous solutions for three views [HB86; HL89], Exact-PSfM-O handles a general number of points (three or more) and has the following guarantees:

Theorem 5. *In the absence of noise Exact-PSfM-O fails to find a metric reconstruction if and only if the scene configuration is degenerate. Therefore Exact-PSfM-O is NADA.*

Theorem 6. *In the presence of noise, assume we have the scene’s rank-2 affine reconstruction approximation (which can be computed in closed-form when there are no missing measurements with the SVD). If Exact-PSfM-O has a solution, then Exact-PSfM-O finds all optimal metric reconstructions in closed-form.*

Theorem 5 is important for two reasons. The first is that Exact-PSfM-O will never fail for problems that are theoretically solvable. The second is that it allows us to systematically characterize all degeneracies, by geometrically interpreting all inputs that cause Exact-PSfM-O to fail. Theorem 6 is important because it tells us that the optimal solutions (those that minimize the reprojection error) for three views can be found in closed-form using Exact-PSfM-O. This is not true in all cases: Exact-PSfM-O will not have a solution if the affine reconstruction cannot be exactly upgraded to a metric reconstruction. In practice we find that the likelihood of Exact-PSfM-O having solutions is typically between 80-90% of the time depending on the level of noise. Consequently, Exact-PSfM-O is able to solve the problem optimally between 80-90% of the time.

3.3.4 New theoretical results for PSfM-WP and PSfM-PP

Theorem 7. *Recall that we cannot solve SfM with affine cameras and planar structure if each camera has six or more unknown DoFs (see §3.1.2). Therefore without additional constraints we cannot solve with the weak-perspective, para-perspective or axially symmetric [KSA07] affine cameras. From the affine camera interpretation in Equation (2.7) this is equivalent to saying that we cannot solve the problem if the magnification factors α_i are free DoFs and/or the terms in \mathbf{A}_i are free DoFs. However if there exist dependencies it may be possible to solve the problem. Three interesting cases are as follows:*

- *Case 1: We can isolate a subset \mathcal{I}' of three or more views where $\mathbf{A}_{i \in \mathcal{I}'}$ is known and $\alpha_{i \in \mathcal{I}'}$ is assumed to be constant.*
- *Case 2: We can isolate a subset \mathcal{I}'' of five or more views where $\mathbf{A}_{i \in \mathcal{I}''}$ and $\alpha_{i \in \mathcal{I}''}$ are unknown and assumed to be constant.*
- *Case 3: We can isolate three or more pairs of views where for each pair $(i, i') \in \{1, 2, \dots, M\}^2$, \mathbf{A}_i and $\mathbf{A}_{i'}$ are known and we assume $\alpha_i = \alpha_{i'}$.*

In Case 1, the problem of upgrading affine to metric structure is constrained only by the views in \mathcal{I}' . In the absence of noise this is exactly equivalent to upgrading with orthographic cameras using only the views in \mathcal{I}' .

Theorem 8. *Suppose the cameras have been intrinsically calibrated with the perspective camera model and we can isolate a subset \mathcal{I}' of three or more views where the distance between the camera and a planar structure is far and approximately constant. We can solve this SfM problem with weak or para-perspective cameras because this is an instance of Case 1 in Theorem 7.*

3.3.5 Summary of the differences between stratified SfM with affine cameras for planar versus non-planar structures

We finish this theoretical section with a summary of the core differences between solving SfM with affine cameras by stratification for planar and non-planar structures (Table 3.1). For non-planar structures results have been aggregated from [TK92; Qua94; Ull79].

	Non-planar structures	Planar structures
Minimal number of structure points	4	3
Maximal theoretical rank of the $2M \times N$ measurement matrix \mathbf{Q}	3	2
Unknown upgrade matrix	$\mathbf{Y} \in \mathbb{R}^{3 \times 3}$, $\text{rank}(\mathbf{Y}) = 3$	$\mathbf{X} \in \mathbb{R}^{2 \times 2}$, $\text{rank}(\mathbf{X}) = 2$
Gauge transform	3D rotation and reflection	2D rotation and reflection
Orthographic cameras		
Critical structures	Co-planar points	Co-linear points
Critical motion sequences	Translation-only motion, rotation about optical axis	Translation-only motion, rotation about optical axis, constant azimuth motion
Are there discrete structure ambiguities with complete measurements?	No	Yes
Are there discrete resection ambiguities with complete measurements?	No	Yes (two-fold ambiguous for each view)
Upgrade constraint	$\mathbf{M}_i^A \mathbf{Y} \in \mathcal{S}_{2 \times 3}$, $\mathbf{M}_i^A \in \mathbb{R}^{2 \times 3}$ is known	$\mathbf{M}_i^A \mathbf{X} \in \mathcal{S}\mathcal{S}_{2 \times 2}$, $\mathbf{M}_i^A \in \mathbb{R}^{2 \times 2}$ is known
Upgrade equation	Quadratic in \mathbf{Y} , linear in $\mathbf{Y}\mathbf{Y}^\top$	Quartic in \mathbf{X} , quadratic in $\mathbf{X}\mathbf{X}^\top$
Number of equality constraints on upgrade matrix per view	3	1
Minimal number of views required for metric upgrade	3	3
Number of distinct upgrade/structure solutions for three views without noise (up to gauge transforms)	1	1 or 2
Least-squares upgrade problem	LLS	Roots of univariate 7^{th} -order polynomial.
Number of upgrade matrices that are local minima of LS upgrade problem	0 or 1	0,1,2,3 or 4
Are the upgrade solutions for three views generally optimal in terms of reprojection error?	No	Yes
Other affine cameras		
Can we solve with the weak-perspective camera without additional information?	Yes	No
Can we solve with the weak-perspective camera with some knowledge about the camera magnification factors?	Yes	Yes
For the para-perspective camera, if the magnification factors are constant and \mathbf{A}_i is constant and unknown for all views, what is the complexity of upgrading self calibration?	\mathbf{A}_i can be trivially eliminated, and is quadratic in \mathbf{Y} , linear in $\mathbf{Y}\mathbf{Y}^\top$.	\mathbf{A}_i cannot be trivially eliminated, and is 5 quadratic equations in 5 unknowns

Table 3.1: Summary of the differences between stratified SfM with affine cameras for non-planar and planar structures

	Range of N	Range of M	Possible number of structure solutions	Are the solutions guaranteed to be planar?
Exact-PSfM-O	≥ 3	$= 3$	0,1,2	Yes
Approx-PSfM-O	≥ 3	≥ 3	0,1,2,3,4	Yes
Approx-PSfM-O(LRE)	≥ 3	≥ 3	0,1	Yes
TJK-CVPR10	$= 3$	≥ 4	0,1	Yes
TK-Factor	≥ 3	≥ 3	0,1	No
MC-CVIU09	≥ 3	≥ 3	1	No
MOVA	≥ 3	≥ 1	1	Yes

Table 3.2: Properties of methods under comparison.

3.4 Empirical evaluation

We now evaluate the accuracy of Exact-PSfM-O and Approx-PSfM-O compared to prior state-of-the-art methods with extensive simulation and real-data experiments.

3.4.1 Method comparison summary

The methods under comparison are as follows. **Exact-PSfM-O**: Proposed exact solution (§3.2.4); **Approx-PSfM-O**: Proposed approximate solution (§3.2.5); **Approx-PSfM-O(LRE)**: Proposed approximate solution but returning at most one structure solution, which is the one that produces the Lowest Reprojection Error; **TJK-CVPR10**: Solution from [TJK10]; **TK-Factor**: Solution from [TK92]; **MC-CVIU09**: solution from [MC09] using the orthographic camera model; **MOVA**: A stratified method using the *Most Orthogonal Viewpoint Approximation* heuristic (details are provided in the following section). We summarize the applicability of the methods in Table 3.2. The purpose for comparing Approx-PSfM-O(LRE) is to show how performance is affected in ambiguous cases when we force Approx-PSfM-O to select the structure solution that yields the lowest reprojection error. For the stratified methods (Exact-PSfM-O, Approx-PSfM-O, Approx-PSfM-O(LRE), MOVA and TK-factorization) we use exactly the same method to compute the structure’s affine reconstruction, given in §3.2.3.

A difficulty with comparing all methods is that for a given test input some methods may be able to produce a metric structure solution but other methods may not (*e.g.* the stratified methods may not produce a valid upgrade matrix). This makes it hard to compute and compare accuracy statistics. We deal with this by applying a fall-back method, and a method reverts to the fall-back’s solution if it does not produce a solution. The fall-back method should return a solution in all cases but is not necessarily the most accurate method. The fall-back method we use is MOVA.

A problem with TK-Factor and MC-CVIU09 is that they return a single solution to camera resection. Therefore for planar structures, even if they compute metric structure correctly their camera poses will be wrong approximately 50% of the time due to the two-fold ambiguity. To handle this fairly we resect the cameras in exactly the same way for *all* methods. This is done using our optimal method given in §3.2.5.4. Note that this requires a planar estimate of metric structure which is not guaranteed by TK-Factor and MC-CVIU09. We deal with this by converting their structure solution $\hat{\mathbf{S}}$ to the closest planar solution before resecting using the rank-two SVD of $\hat{\mathbf{S}}$.

We also evaluate the gain in accuracy by refining the best solution among all methods with Orthographic camera Bundle Adjustment (OBA), which we denote by **Best+OBA**. This is done by taking the metric structure solution among all methods with the lowest error (see below) then resect-

ing the cameras as described in §3.2.5.4. Then structure and camera poses are jointly refined until convergence by minimizing the $L2$ reprojection error using LM.

3.4.2 MOVA: The fallback method

MOVA is an approximate stratified-based solution that uses the following heuristic: if the camera orientations are distributed randomly and independently then with a sufficiently large number of views there is likely to be one that has a fronto-parallel view of the structure plane (in the limit the probability reaches 1). If we do indeed have a fronto-parallel view then the points in its image give the metric structure up to noise. In reality we are never likely to have such a view, but we can approximate metric structure using the view that is *most* fronto-parallel. This is the view where $\det^2([\mathbf{R}_i]_{2 \times 2})$ is largest. Because $\det([\mathbf{R}_i]_{2 \times 2}) = \det(\mathbf{M}_i^A \mathbf{X}) = \det(\mathbf{M}_i^A) \det(\mathbf{X})$, it is also the view where $\det^2(\mathbf{M}_i^A)$ is largest, so it can be determined entirely from the affine structure matrix. Suppose this is given by view i^* . Assuming this view is fronto-parallel we have $\mathbf{M}_{i^*}^A \mathbf{W} \mathbf{M}_{i^*}^{A\top} \approx \mathbf{I}_2$, so we can approximate the upgrade matrix by $\mathbf{W} \approx (\mathbf{M}_{i^*}^A)^{-1} (\mathbf{M}_{i^*}^A)^{-\top}$. Metric structure can then be computed by factoring \mathbf{W} as described in §3.2.1. We use MOVA as the fall-back solution because we can always compute metric structure unless for all views $\det(\mathbf{M}_i^A) = 0$ (in which case we cannot perform the matrix inversion). This happens when the structure plane normal is orthogonal to the projection direction in all views, and is a rare occurrence in practice.

3.4.3 Error metrics

We measure performance using four metrics. These are (i) *structure error*, (ii) *rotation error*, (iii) *translation error* and (iv) *success rate*. We use $\mathcal{M} = \{\hat{\mathbf{S}}_1, \dots, \hat{\mathbf{S}}_K\}$ to denote the set of K metric structure solutions produced by a method. If a method fails to compute a structure solution then $\mathcal{M} = \emptyset$. We use $\hat{\mathbf{S}}^{MOVA}$ to denote the structure solution produced by the fall-back method MOVA. We use $\mathbf{S}^{GT} \in \mathbb{R}^{2 \times N}$, $\mathcal{R}^{GT} \in \mathcal{SO}_3^M$ and $\mathcal{T}^{GT} \in \mathbb{R}^{2 \times M}$ to denote the ground truth structure, camera rotations and translations respectively.

3.4.3.1 Structure error $E_S \in \mathbb{R}^+$

Structure error is computed using the solution in \mathcal{M} that is closest to ground truth up to a similarity transform. If a method does not return a structure solution the solution from the fall-back method is used (MOVA). Formally, we define it as:

$$E_S \stackrel{\text{def}}{=} \begin{cases} \frac{1}{N} \sum_{j=1}^N \|\mathbf{S}_j^{GT} - \hat{\mathbf{S}}'_j\|_2 & \text{if } \mathcal{M} \neq \emptyset \\ E_S^{MOVA} & \text{otherwise} \end{cases} \quad \hat{\mathbf{S}}' \stackrel{\text{def}}{=} \arg \min_{\hat{\mathbf{S}} \in \mathcal{M}} \|\text{ABSOR}(\hat{\mathbf{S}}, \mathbf{S}^{GT}) - \mathbf{S}^{GT}\|_2^2 \quad (3.38)$$

The function $\text{ABSOR}(\hat{\mathbf{S}}, \mathbf{S}^{GT})$ aligns an estimate $\hat{\mathbf{S}}$ to \mathbf{S}^{GT} in the least squares sense over all 2D similarity transforms. This alignment is necessary to account for the gauge transform. Therefore $\hat{\mathbf{S}}'$ is the structure solution that has aligned best to \mathbf{S}^{GT} . The value $E_S^{MOVA} \in \mathbb{R}^+$ is the structure error from MOVA.

3.4.3.2 Rotation and translation error $E_R \in \mathbb{R}^+$, $E_T \in \mathbb{R}^+$

For each method we take the best structure solution $\hat{\mathbf{S}}'$, resect the cameras as described in §3.2.4.3, then we measure the error of the camera poses. If a method has not produced a structure solution we

use the camera poses from the fall-back method (MOVA). Let $(\hat{\mathbf{R}}_i^a, \hat{\mathbf{t}}_i^a)$ and $(\hat{\mathbf{R}}_i^b, \hat{\mathbf{t}}_i^b)$ be the camera pose estimates for view i (recall there are two from the two-fold ambiguity). The rotation error is defined as follows:

$$E_R = \begin{cases} \frac{1}{M} \sum_{i=1}^M \min [\text{ang}(\hat{\mathbf{R}}_i^a, \mathbf{R}_i^{\text{GT}}), \text{ang}(\hat{\mathbf{R}}_i^b, \mathbf{R}_i^{\text{GT}})] & \text{if } \mathcal{M} \neq \emptyset \\ E_R^{\text{MOVA}} & \text{otherwise} \end{cases} \quad (3.39)$$

The function $\text{ang}(\mathbf{R}, \mathbf{R}') : \mathcal{SO}_3^2 \rightarrow [0, 180]$ denotes the smallest angle in degrees of the rotation that maps \mathbf{R} to \mathbf{R}' . Because there are two rotation estimates per view, the error of the one with the smallest angular error is used. The value $E_R^{\text{MOVA}} \in \mathbb{R}^+$ is the rotation error from MOVA.

We measure translation error as follows. For each view we determine which of the two pose estimates has the lowest rotation error, then measure the accuracy of its corresponding translation estimate $\hat{\mathbf{t}}_i \in \mathbb{R}^2$. This is defined as

$$E_T = \begin{cases} \frac{1}{M} \sum_{i=1}^M \|\hat{\mathbf{t}}_i - \mathbf{t}_i^{\text{GT}}\|_2 & \text{if } \mathcal{M} \neq \emptyset \\ E_T^{\text{MOVA}} & \text{otherwise} \end{cases} \quad (3.40)$$

where $E_T^{\text{MOVA}} \in \mathbb{R}^+$ is the translation error from MOVA. We note here that the translation error is only useful for comparing methods when there are missing measurements. This is because when there are no missing measurements $\hat{\mathbf{t}}_i$ is the centroid of the point correspondences in view i , so it is the same for all methods. When there are missing measurements $\hat{\mathbf{t}}_i$ will depend on the particular rotation solution (see Equation (3.22)), so it will differ depending on the method.

3.4.3.3 Success rate

$E_{\text{succ}} \in [0, 100]$. We define the success rate as the percentage of instances for which a method produces a metric structure solution. The success rate of MC-CVIU09, MOVA and Best+OBA is 100%, so we only compare success rates for Exact-PSfM-O, Approx-PSfM-O, TJK-CVPR10 and TK-factor. The success rate of Approx-PSfM-O(LRE) is the same as Approx-PSfM-O so we omit it from the results.

3.4.4 Experiments with simulated data

We ran a large number of simulation experiments to test the accuracy of the methods in a variety of conditions. We generated ground truth structure matrices \mathbf{S}^{GT} by synthesizing N points positioned randomly on the structure plane $z = 0$ in world coordinates. These were drawn within the square centered at the origin. The points were then normalized so the centroid was at the origin and the mean distance from the origin was 100 units. A set of M random rotations were then generated: $\mathcal{R}^{\text{GT}} = \{\mathbf{R}_1^{\text{GT}}, \dots, \mathbf{R}_M^{\text{GT}}\}$ where $\mathbf{R}_i^{\text{GT}} \in \mathcal{SO}_3$ rotates the structure plane to camera coordinates. Similarly to [LMF09] we randomly generated these using Euler angles where each angle was assigned with uniform probability in the range $[-80, +80]$ degrees. It was unnecessary to simulate variation in the camera translations because it has no effect on a method's accuracy (because the point correspondences are simply translated in the image), so we set $\mathbf{t}_i^{\text{GT}} = \mathbf{0}_{2 \times 1}$ for all views. We also simulated variation of the camera magnification factors α_i because in real conditions the orthographic model may not hold perfectly due to variation in the scene's depths. For each view we assigned α_i with a random distribution $\alpha_i \sim \mathcal{N}(1, \sigma_k/100)$, $\sigma_k \in \mathbb{R}$. We then projected the scene points for each view and perturbed them with IID zero-mean Gaussian noise with standard deviation σ_n . To simulate missing measurements we randomly removed $\gamma\%$ of the correspondences in each view. This was done while

	N	M	σ_n	σ_k	γ
Experiment 1	[3, 40]	3	2	0	0
Experiment 2	[3, 40]	4	2	0	0
Experiment 3	[3, 40]	8	2	0	0
Experiment 4	[3, 40]	12	2	0	0
Experiment 5	3	4	2	[0, 10]	0
Experiment 6	3	8	2	[0, 10]	0
Experiment 7	50	3	2	5	[0, 70]
Experiment 8	50	8	2	5	[0, 70]

Table 3.3: Experimental parameters used in eight simulation experiments.

ensuring the scene’s affine structure could still be recovered using Algorithm 10.

We excluded from the evaluation all simulations that were poorly-conditioned, since they cannot be used to draw meaningful comparisons between the methods. This was done with the following policy. For a given simulation we first ran bundle adjustment initialized using the ground truth. If it converged far from the ground truth solution we assumed the problem was ill-conditioned and did not select it (we used a structure error threshold of 10%). We also tested whether the reprojection error had a local minimum at the point of convergence using the conditioning number of the residual error Jacobian matrix with a threshold of 1×10^{-7} . If so it was used for evaluation. We computed performance statistics over different values of the experimental parameters $\{N, M, \sigma_k, \sigma_n, \gamma\}$ by averaging over $T = 1000$ simulated scenes. We conducted eight experiments given in Table 3.3.

3.4.4.1 Results

The results of experiments one to four are shown in Figure 3.2. Each column corresponds to one experiment and the six rows show different performance statistics across the methods (the success rate, mean and median structure error, mean and median rotation error and mean reprojection error). Results for Exact-PSfM-O are shown only in the first column because it is only applicable when $M = 3$. TJK-CVPR10 is not present in experiment one and shown as a black cross in experiments two, three and four because it is applicable when $N = 3$ and $M > 3$ only. Results for translation error were not plotted because they were the same for all methods (because $\gamma = 0\%$).

With respect to success rate, when $N = 3$ points the success rate of TK-Factor is 0%. This is because when $N = 3$ the measurement matrix $\hat{\mathbf{Q}}$ never has a rank greater than two even with noise, so TK-Factor fails because it returns rank-deficient upgrade matrices that cannot be inverted. When $N > 3$, noise increases the rank of $\hat{\mathbf{Q}}$ beyond its theoretical rank, which means it is possible to find full-rank upgrade matrices using TK-Factor, which is about 65% of the time when $M = 4$ and 95% for $M = 12$. However the solutions from TK-Factor are poor compared to all other methods except MC-CVIU09, both of which perform worse than the fall-back method (*i.e.* MOVA). Structure and rotation errors for Exact-PSfM-O, Approx-PSfM-O, Approx-PSfM-O(LRE) and Best+OBA tend to decrease with more points because the effect of noise reduces. There is virtually no difference between the accuracy of Approx-PSfM-O and Best+OBA across all statistics. For $M = 3$ there is a significant difference in the structure error from Approx-PSfM-O(LRE) and Approx-PSfM-O. *This is expected because for three views structure is not in general unique. Therefore the structure solution from Approx-PSfM-O that is closest to ground truth is not necessarily the correct one.* When $M = 4$ and beyond we see that the accuracy of Approx-PSfM-O(LRE) and Approx-PSfM-O is similar. There is slight deviation for $M = 4$, which can be explained by the fact that sometimes there can be multiple

structure solutions when the cameras are in a particular configuration (see Theorem 2). With more views the likelihood of this occurring rapidly diminishes, which explains why they show the same error for $M > 4$.

The success rate of Approx-PSfM-O (and Approx-PSfM-O(LRE)) was approximately constant in all experiments and for all N at approximately 99.8%. For Exact-PSfM-O, we see in experiment one a gradual improvement in success rate from 82% to 94% as the number of points increases. This suggests that as the influence of noise decreases the chances of being able to exactly upgrade the scene’s affine structure to metric structure increases. Because the solution from Exact-PSfM-O gives the optimal solution to PSfM-O (from Theorem 6), this indicates that *we can find the optimal solutions in closed-form for three views between 82% to 94% of the time* in these cases. The reason why Exact-PSfM-O has worse performance than Approx-PSfM-O is because it fails more often, so we revert back to the fall-back method more often than with Approx-PSfM-O.

The results for experiments five and six are shown in the first two columns of Figure 3.3. For all methods we see a reduction in accuracy as the magnification factor standard deviation σ_k increases, which is due to increasing the modeling error. In experiment six we see a significant reduction in the success rate of TKJ-CVPR10 to 80.6% when $\sigma_k = 10\%$. There is also a very small drop in success rate of Approx-PSfM-O and Approx-PSfM-O(LRE) but only to 99.1% when $\sigma_k = 10\%$. The success rate of TK-Factor is 0% for all values of σ_k , which as discussed above is because it never computes an invertible upgrade matrix when $N = 3$. Unusually, we see that the mean structure error of bundle adjustment appears worse than our methods, however the median error is very similar. The problems are caused by the fact that for large k the data violate the noise model (it is no longer IID Gaussian), so we may not necessarily observe bundle adjustment giving the most accurate solutions. By contrast we see the mean errors of Approx-PSfM-O and Approx-PSfM-O(LRE) degrade gracefully with increased k . Similarly to the previous experiments we see that the accuracy of Approx-PSfM-O and Approx-PSfM-O(LRE) is indistinguishable when the number of views reaches eight, because the likelihood of there being discrete structure ambiguities diminishes considerably. As σ_k increases we see a greater difference in accuracy between TKJ-CVPR10 and Approx-PSfM-O, which indicates TKJ-CVPR10 cannot handle modeling approximation error as well as Approx-PSfM-O. The results for experiments seven and eight are shown in the last two columns of Figure 3.3. In these experiments we plot the translation error in the last two rows (recall that the translation error is only relevant when $\gamma > 0$). Again we see virtually no difference between bundle adjustment and Approx-PSfM-O. When $M = 8$ Approx-PSfM-O and Approx-PSfM-O(LRE) are indistinguishable.

In summary, these experiments show that with IID Gaussian measurement noise there is virtually no gain in the bundle adjustment solution compared to Approx-PSfM-O. This is an unusual and interesting result because bundle adjustment optimizes *both* structure and camera poses with the full reprojection error. By contrast Approx-PSfM-O estimates structure by an algebraic upgrade function (Equation (3.23)). This result tells us something quite profound about the problem. *It indicates that the optimal metric structure is extremely similar to the optimal affine structure up to an upgrade transform, and Equation (3.23) does an excellent job for finding the transform (or transforms if the problem is ambiguous).*

3.4.5 Experiments with real data

In this section we present results using real image data. We add to the methods bundle adjustment with a perspective camera (with fixed and pre-calibrated intrinsic matrices), which we call **Best+PBA**.

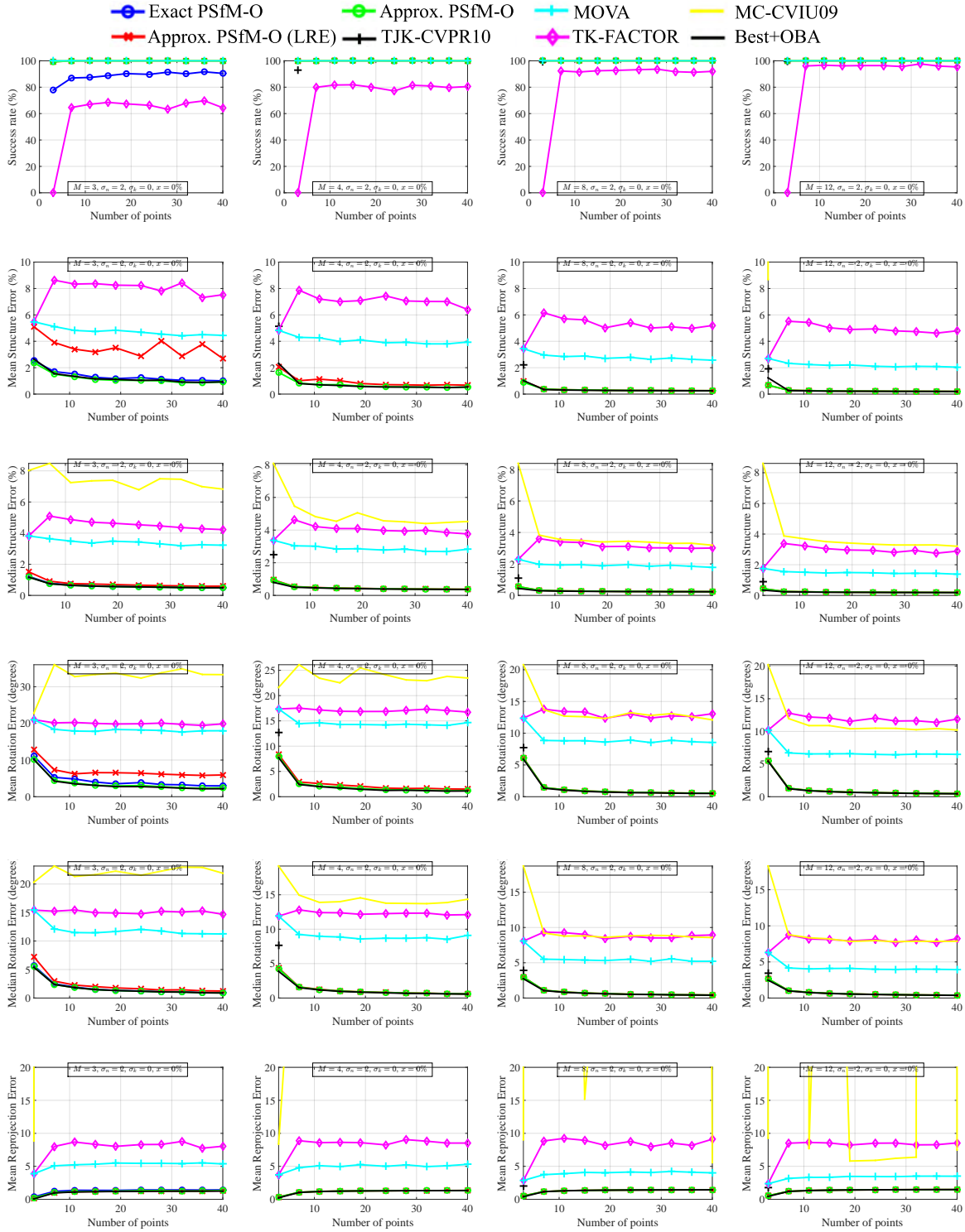


Figure 3.2: Simulation experimental results: experiments one to four with one experiment per column. Best viewed in colour.

Similarly to Best+OBA we compute this by taking the best solution to structure across all methods, but then we resect the cameras with a perspective plane-based pose estimation algorithm (IPPE from Chapter 4). We then run bundle adjustment to jointly refine the structure (which we constrain to lie

3.4. EMPIRICAL EVALUATION

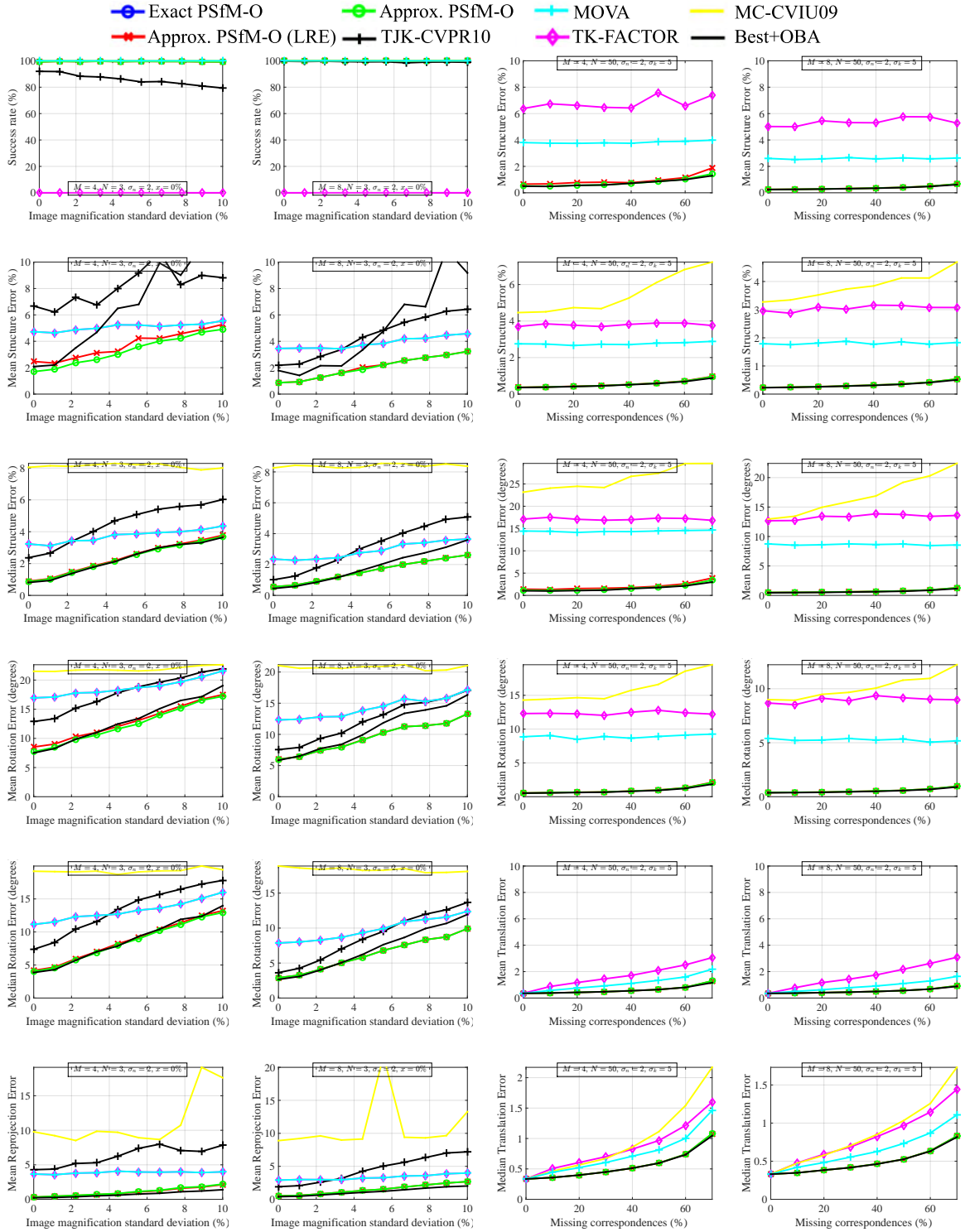


Figure 3.3: Simulation experimental results: experiments five to eight with one experiment per column. Best viewed in colour.

on the plane $z = 0$ in world coordinates) and perspective camera poses with LM.

3.4.5.1 Reconstruction with a textured planar surface

The first set of real-data experiments were performed with an unorganized collection of eight views of a textured sheet of A4 paper mounted on a flat surface (Figure 3.5). The views were taken with a Nikon D800 DSLR with a 120 mm lens with fixed focal length and image resolution of 3680×2456 px. The pattern on the paper measured 23.0×20.5 mm with an average distance to the camera of approximately 3.2 m. We intrinsically pre-calibrated the camera with Bouguet’s toolbox [Bou00] which gave an effective focal length of 1.0068×10^4 px and 1.0060×10^4 px in the x and y axes respectively (which is approximately 2.27 times the image diagonal). Feature points were computed over the pattern with SIFT [Low04b] using the VLFeat implementation [VF], which gave on average 288.3 features per view. We computed ground truth camera poses using a digital image of the paper as a 2D template which we registered in 3D to each view using a direct approach based on the DIRT toolbox [Bar08]. Correspondences and ground truth structure were determined by matching SIFT descriptors and computing the optimal positions of the features on the 2D template given the camera poses. For this we computed all inter-image homographies from the camera poses then computed putative correspondences between each pair by matching features with the closest SIFT descriptors. Correspondences were used if predicted by the homography to within 7 pixels. The correspondences were then chained to give 1842 unique points, and we then refined the points’ positions on the 2D template by minimizing the reprojection error using LM. The average number of missing correspondence per view was 64.5%.

We measured the performance of each method across two dimensions. The first dimension was the number of views M which we varied from $M = 3$ to $M = 7$. For each M we ran the methods over all possible subsets of M views. We also measured how performance varied with smaller neighborhoods of correspondences. The purpose was to investigate how methods perform as the number of point correspondences decreased. We performed this by taking each of the 1842 points in turn, and for each point we used only neighboring correspondences that were within a neighborhood radius r to that point. In our experiments we varied r between 15% and 60% of the whole pattern’s size.

The results are shown in Figure 3.4. Along each column we plot results for $M = 4$, $M = 5$ and $M = 8$ from left to right. Along each row we plot the mean and median structure error, mean and median rotation error, and success rate against the neighborhood radius. We first inspect structure error. For all M the accuracy of Best+PBA is poor but tends to improve with a larger neighborhood size. This is because for smaller neighborhoods Plane-based SfM with perspective cameras becomes poorly-conditioned. The results for TK-FACTOR look better than they actually are, which is because its success rate is so low. Therefore very often it had to revert to the solution from MOVA. We again see that across all settings there is very little difference between Approx-PSfM-O and bundle adjustment. In terms of success rate Approx-PSfM-O never dropped below 99.92%.

3.4.5.2 Reconstruction from an orbiting image sequence

The second real-data experiment involved reconstructing the top surface of a bottle cap from an ordered set of orbiting views (Figure 3.6). The image set consists of 18 1600×1800 px images taken with an automatic compact camera with fixed zoom. Views 1, 10 and 18 are shown in Figure 3.6 (first row, left). We intrinsically pre-calibrated the camera with Bouguet’s toolbox which gave an effective focal length of 1.26×10^4 px in the x and y axes (which is approximately 6.32 times the image diagonal). Points were computed using the Harris detector on the first view of the bottle cap (using default parameters), which gave 137 points (Figure 3.6 (row three, left image)). We tracked

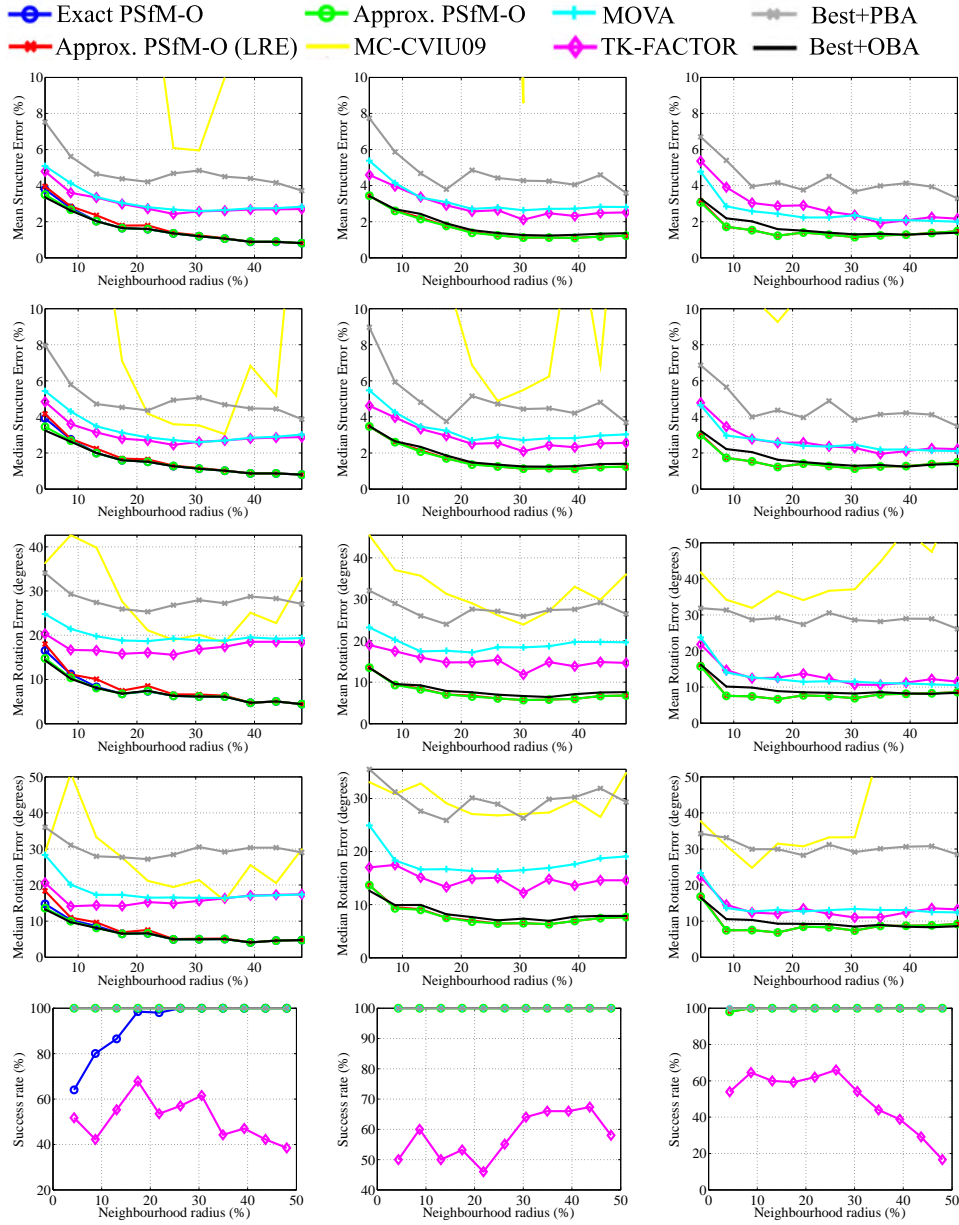


Figure 3.4: Results on the image set shown in Figure 3.5. In the three columns we show results using three, five and eight of the views. In the rows we show the corresponding performance statistics.

the points in subsequent views using KLT. To reduce tracking drop-off, for each view $i \in \{2, \dots, 18\}$ we computed an approximate homography between view 1 and i , then back-warped image i to image 1, and then ran KLT on the back-warped image. The approximate homography was computed using SIFT feature matching and RANSAC. Outliers from KLT were detected by refitting the homography with RANSAC to the KLT matches and rejecting matches with a transport error beyond 5 pixels. We used the fact that the bottle cap’s top surface is circular to rectify image 1 with a homography (Figure 3.6 (row three, left image)). Ground truth structure was computed by first computing the optimal 2D affine structure using Algorithm 10 and mapping the point’ affine structure to the rectified image. The number of missing correspondences began at 0% in image 1 and rose to 56.2% in image 18.

Similarly to the previous experiment, we measured performance by randomly sampling sub-sets of



Figure 3.5: A real test set consisting of eight unorganized 3680×2456 views of a textured flat A4 sheet of paper.

views from the full collection of 18 views, whose size we varied from 3 to 10. For each size we drew 50 random subsets and computed performance statistics over the 50 subsets. The results for mean structure error, median structure error, mean reprojection error and success rate are shown in Figure 3.6, second row. We see that PSfM-O again performs very well and there is no significant difference with bundle adjustment. The success rate of Approx-PSfM-O (and Approx-PSfM-O(LRE)) was 100% in all cases. The success rate of TK-Factor generally reduced with more views, and for 10 views was 36%. The method with lowest reprojection error was Best+PBA, however this also produced much higher structure error. This tells us the perspective camera is unsuitable for solving this problem, due to the bottle cap being too small to reliably estimate structure *and* the perspective camera projection matrices. In the last two rows of Figure 3.6 we show the reconstructed points computed from each method with a subset size of 10. The ground truth structure is shown by the set of red circles. The dark circles show the reconstructed points from a method. Because for each method we performed 50 reconstructions, we overlaid all 50 reconstructions. One can see a good clustering of the points by Approx-PSfM-O and Approx-PSfM-O(LRE) about the ground truth positions.

We also ran a second experiment to test how the methods performed by adding views in sequential order, starting from the first three views. The purpose was to see how accuracy improved as the baseline of the image set increased. The results are shown in Figure 3.6, top-right. One can see a smooth reduction error of Approx-PSfM-O, Approx-PSfM-O(LRE) and Best+OBA as the number of views (and the the baseline) increased. This demonstrates again the accuracy of our method and that there is no significant gain in accuracy by refining our solutions with bundle adjustment.

3.5 Conclusion

We have presented various technical and theoretical contributions for planar Structure-from-Motion with affine cameras. The problem is fundamentally different to SfM with non-planar structures, because the affine camera models one can use are more restricted, the upgrade constraints are non-linear and non-convex, and the problem is far more ambiguous. We have presented eight new theorems that significantly deepen our understanding of the problem. Our main theoretical result is a complete geometric characterization of degeneracies with orthographic cameras (*i.e.* the PSfM-O problem). The second main theoretical result is to show that the PSfM-O problem can have discrete structure ambiguities with a general number of views, and to give the necessary and sufficient geometric conditions for disambiguation. We have also presented three cases when SfM may be solvable with other

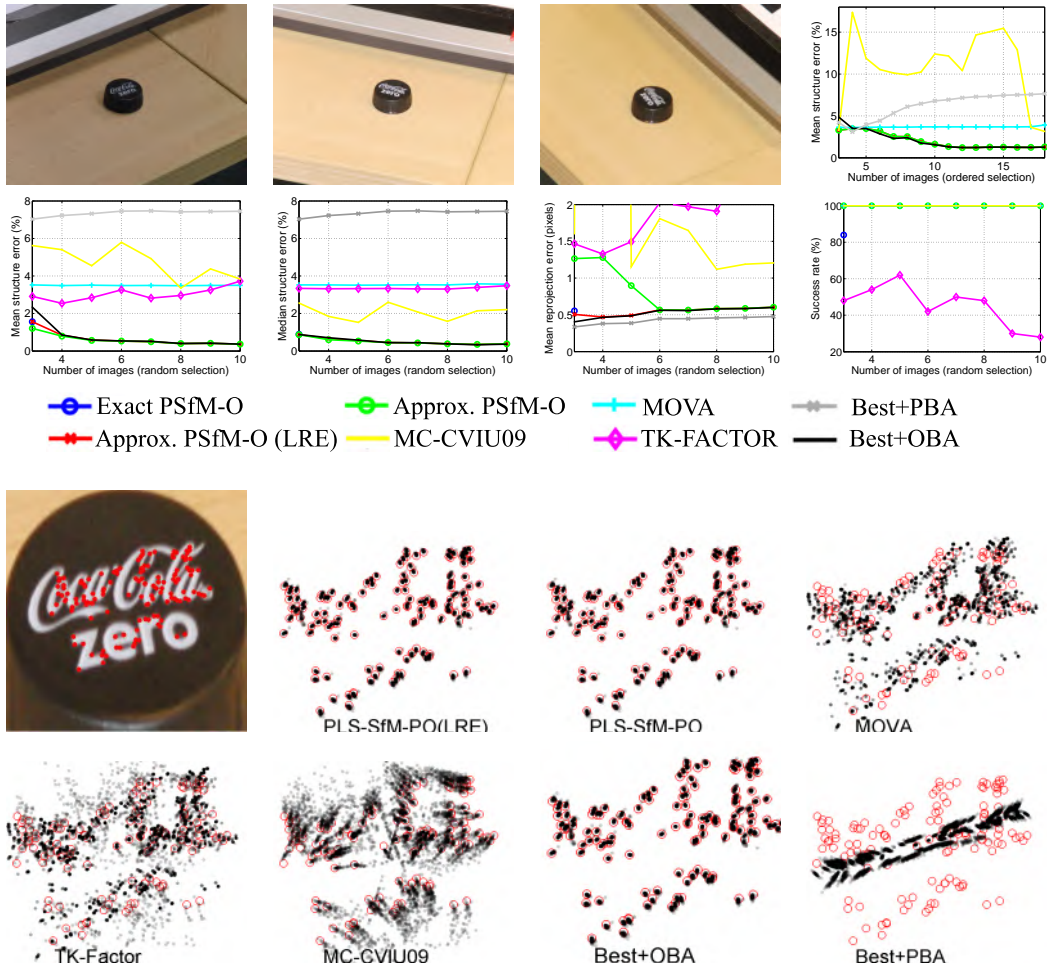


Figure 3.6: Results for reconstructing the top section of a bottle top from an orbiting image sequence (best viewed in colour).

affine cameras, which necessarily requires additional knowledge.

Our main technical contribution is Approx-PSfM-O, which solves the PSfM-O problem in its most general form. Approx-PSfM-O handles cases when there exist discrete structure ambiguities, which is not true of previous algorithms. The solutions from Approx-PSfM-O tend to be very close to locally-optimal metric reconstructions. This has been demonstrated by extensive empirical evaluation which shows that the solutions are not significantly improved by running bundle adjustment. Because Approx-PSfM-O is stratified it does not optimize the reprojection error at all stages (it does this only at the affine reconstruction and camera resection stages). The fact that the results are very close in accuracy compared to fully optimizing the reprojection error tells us two important things that were previously unknown. The first is that we can compute the plane’s metric structure very well from the optimal affine reconstruction up to an unknown upgrade transform. The second is that our upgrade cost function does an excellent job of finding the transform (or multiple transforms if the problem is ambiguous) in closed-form, even for high levels of measurement noise. In the case of three views we have some theory to explain this phenomenon (Theorem 6). Specifically, if we can exactly upgrade an optimal affine reconstruction to a metric reconstruction then the upgraded reconstruction *is* the optimal metric reconstruction. In these cases running bundle adjustment will do absolutely

nothing. Empirically we have found that we can do this for 80 to 90 percent of cases depending on noise. More theoretical analysis is required to study the precise relationship between optimizing the full reprojection error and the stratified cost functions in our method for more than three views, and we leave this to future work. This is non-trivial and requires uncertainty propagation to analyze how error in the measurements propagate to errors in the upgrade cost function. We discuss additional future research directions based on the contributions of this Chapter in Chapter 6.

Infinitesimal Plane-based Pose Estimation

Chapter summary and organization

This chapter presents Infinitesimal Plane-based Pose Estimation (IPPE): our closed-form solution to Plane-based Pose Estimation with an intrinsically calibrated perspective camera (PPE-P). In §1.3.2.1 of the introduction chapter of this thesis we have provided the PPE-P problem background, motivation for improving on-state-of-the-art and applications. In §1.3.2.2 we have provided an overview of this chapter's technical and theoretical contributions. This chapter is organized into 6 main sections. In §4.1 we motivate IPPE by summarizing the limits of previous fast PPE-P methods and why IPPE overcomes these limits. In §4.2 we give the IPPE method details. In §4.3 we analyze the method and provide several theorems about algorithm correctness, solution geometry and its connections to Perspective-3-Point (P3P) and weighted Perspective Homography Decomposition. In §4.4 we evaluate IPPE against previous state-of-the-art real-time methods with extensive simulation experiments. In §4.5 we evaluate IPPE against these methods with real datasets including pose estimation from feature matches, checkerboard corners and AR marker corners. In §4.6 we conclude this chapter. Directions for future research are detailed in §6.2.2.2 in the final chapter of this thesis.

4.1 IPPE motivation: Overcoming the limits of previous PPE-P methods

We recall that PPE-P is the problem of estimating the relative pose of a planar structure with respect to a perspective camera’s 3D coordinate frame using point correspondences. A good PPE-P method has five main characteristics:

1. Low computational cost
2. High accuracy
3. Handles the two-fold pose ambiguity in quasi-ambiguous cases (see Figure 1.5)
4. In ambiguous cases, produces solutions that are geometrically interpretable
5. Has provably no artificial degeneracies

No state-of-the-art method published before IPPE fulfilled all these criteria. Solutions with lowest computational cost were Zhang’s [Zha00] and Sturm’s [Stu00] PHD methods. These determine the plane’s pose in two stages: first a homography matrix $\hat{\mathbf{H}}$ is fitted to the correspondences, then $\hat{\mathbf{H}}$ is decomposed analytically into rotation and translation. However, these methods are relatively sensitive to noise, they are not statistically optimal, and they give poor solutions when the perspective effects diminish (*i.e.* affine conditions), occurring when the plane is small relative to its depth. Consequently, they do not satisfy items 2, 3, 4 and 5. On the other hand, the most accurate closed-form solution prior to IPPE was DLS [HR11]. However, DLS is extremely computationally demanding with $O(100)$ ms computation time in C++ on a modern PC. Consequently, DLS is not suitable for real-time vision applications. In contrast, the computation time of the PHD methods including homography estimation in C++ is approximately $O(10)\mu s$ for 4 points (where homography estimation has an analytic solution) and $O(100)\mu s$ with $4 < n < 1000$ and DLT homography estimation. Prior closed-form methods have attempted to strike a balance between computational speed and solution accuracy. The best two methods for real-time applications prior to IPPE’s publication were EPnP [LMF09] and RPnP [LXX12]. However, EPnP can be unstable with planar structures, and they are both much slower than the PHD methods.

The presented approach, IPPE, satisfies all the above 5 criteria: IPPE is considerably more accurate than the HD methods while having approximately the same very low computational cost. It handles the two-fold ambiguity, its solutions have a simple geometric relationship, and we prove that when the homography can be computed uniquely from point correspondences, IPPE is guaranteed to find the correct solution (it is a NADA algorithm). The solutions from IPPE can generally be improved by iterative reprojection error refinement with LM. However, when the object points are ‘well distributed’ such as the 4 corners of an AR marker, the solution accuracy is very close to the refined estimate, and there is no clear benefit for using refinement. Planar pose estimation with AR markers is a very common use case of regular point configurations, so this quality of IPPE has real practical benefit for reducing computation time in real-time vision applications.

4.2 Methodology

4.2.1 Definitions

Without loss of generality we define the *object plane* in object coordinates at $z = 0$. The object plane has a set of $n \geq 4$ *object points* $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$, $\mathbf{u}_{i \in [1, n]} \in \mathbb{R}^2$. Without loss of generality we assume the object points are zero-centered: $\sum_{i=1}^n \mathbf{u}_i = \mathbf{0}_{2 \times 1}$. The object points are in correspondences with a set of n *image points* $\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \dots, \hat{\mathbf{q}}_n\}$, $\mathbf{q}_{i \in [1, n]} \in \mathbb{R}^2$ defined in normalized pixel coordinates. In practice, image points detected in a real image are converted to normalized pixel coordinates using an intrinsic camera calibration, which we assume is known. We define the rigid transform that maps object coordinates to camera coordinates by the rotation $\mathbf{R} \in SO_3$ and translation $\mathbf{t} \in \mathbb{R}^3$. We define as $s(\mathbf{u}) = \mathbf{R} \text{stk}(\mathbf{u}, 0) + \mathbf{t} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ the function that transforms a point \mathbf{u} on the object plane to camera coordinates. We define as $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ the function that transforms the object plane to normalized pixel coordinates. We define as π the pinhole projection function with $\pi(x, y, z) \stackrel{\text{def}}{=} \frac{1}{z} \text{stk}(x, y)$. The relationship between w , s and π is

$$w(\mathbf{u}) \stackrel{\text{def}}{=} \pi(s(\mathbf{u})) \quad (4.1)$$

The function w is encapsulated by a homography \mathbf{H} between the object plane and camera image. Without loss of generality we assume $\mathbf{H}_{33} = 1$. The relationship between w and \mathbf{H} is

$$w(\mathbf{u}) = \pi(s(\mathbf{u})) = \frac{1}{(\mathbf{H}_{31}, \mathbf{H}_{32}, 1) \text{stk}(\mathbf{u}, 1)} [\mathbf{H}]_{2 \times 3} \text{stk}(\mathbf{u}, 1) \quad (4.2)$$

We define as $\hat{\mathbf{H}}$ an estimate of \mathbf{H} fitted to the point correspondences. We assume $\hat{\mathbf{H}}$ can be estimated uniquely, equivalent to the condition that there exists no subset of $n - 1$ object points that are collinear. In the special case of 4 correspondences, $\hat{\mathbf{H}}$ is computed exactly and it has an analytical solution. In the case of 5 or more noisy points, $\hat{\mathbf{H}}$ cannot be computed exactly, and established closed-form methods exist to compute a best-fitting homography in some sense. We test several methods in the experimental section of this chapter.

4.2.2 Approach overview

Prior PHD methods take as input a general 8-DoF homography matrix, from which the 6-DoF pose of the planar object relative to the camera is estimated. Therefore, PHD is a redundant system. The methods of [Stu00] and [Zha00] deal with this redundancy by solving the best-fitting pose by minimizing an algebraic cost which is not statistically optimal. We propose a new method that uses the redundancy in the homography to provide far better pose estimates. Our method is based on the fact that given a homography estimated from noisy correspondences, it does not predict the object-to-image transform with uniform accuracy. That is, the homography will predict the transformation better at some points on the object plane than others. Our method is based on identifying a point \mathbf{u}_0 on the object plane close to where the transform is best estimated, and then solving pose with an exact system, using local motion information at \mathbf{u}_0 , derived from the homography. We call our approach Infinitesimal Plane-based Pose Estimation (IPPE) because it can be thought of as solving pose using motion information within an infinitesimally-small region about \mathbf{u}_0 . Specifically, the motion information is 0^{th} -order (displacement) and 1^{st} -order (motion gradient) at \mathbf{u}_0 . Pose is solved analytically with this motion information, which reduces to computing the largest singular value of a 2×2 matrix.

4.2.3 Problem statement

We now use Equation (4.2) to derive 6 pose constraints using a 1st-order PDE. Consider a point \mathbf{u}_0 on the object plane that we call the *differentiation point*. We assume the differentiation point is in front of the camera. We denote as \mathbf{v} its position in normalized pixel coordinates. This is estimated from $\hat{\mathbf{H}}$ using Equation (4.2-a):

$$\mathbf{v} = w(\mathbf{u}_0) \approx \frac{1}{\begin{bmatrix} \hat{\mathbf{H}}_{31}, \hat{\mathbf{H}}_{32}, 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ 1 \end{bmatrix}} \begin{bmatrix} \hat{\mathbf{H}} \end{bmatrix}_{2 \times 3} \begin{bmatrix} \mathbf{u}_0 \\ 1 \end{bmatrix} \quad (4.3)$$

where \approx is used to denote a noisy estimate. In the special case when the differentiation point is the origin of object coordinates: $\mathbf{u}_0 = \mathbf{0}_{2 \times 1}$, \mathbf{v} simplifies to

$$\mathbf{v} \approx \text{stk} \left(\hat{\mathbf{H}}_{13}, \hat{\mathbf{H}}_{23} \right) \quad (4.4)$$

Ignoring noise, \mathbf{v} provides 2 constraints on pose from Equation (4.1)

$$w(\mathbf{u}_0) = \pi(s(\mathbf{u}_0)) = \frac{1}{[s(\mathbf{u}_0)]_3} [s(\mathbf{u}_0)]_{2 \times 1} = \mathbf{v} \quad (4.5)$$

We can also estimate from $\hat{\mathbf{H}}$ the Jacobian of w at \mathbf{u}_0 , denoted by the 2×2 matrix \mathbf{J} . This is computed as follows

$$\begin{aligned} \mathbf{J} = J_w(\mathbf{u}_0 = \text{stk}(x, y)) &\approx \frac{1}{(x\hat{\mathbf{H}}_{31} + y\hat{\mathbf{H}}_{32} + 1)^2} \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \\ \mathbf{J}_{11} &\stackrel{\text{def}}{=} \hat{\mathbf{H}}_{11} - \hat{\mathbf{H}}_{13}\hat{\mathbf{H}}_{31} + y \left(\hat{\mathbf{H}}_{11}\hat{\mathbf{H}}_{32} - \hat{\mathbf{H}}_{12}\hat{\mathbf{H}}_{31} \right) \\ \mathbf{J}_{12} &\stackrel{\text{def}}{=} \hat{\mathbf{H}}_{12} - \hat{\mathbf{H}}_{13}\hat{\mathbf{H}}_{32} - x \left(\hat{\mathbf{H}}_{11}\hat{\mathbf{H}}_{32} - \hat{\mathbf{H}}_{12}\hat{\mathbf{H}}_{31} \right) \\ \mathbf{J}_{21} &\stackrel{\text{def}}{=} \hat{\mathbf{H}}_{21} - \hat{\mathbf{H}}_{23}\hat{\mathbf{H}}_{31} + y \left(\hat{\mathbf{H}}_{21}\hat{\mathbf{H}}_{32} - \hat{\mathbf{H}}_{22}\hat{\mathbf{H}}_{31} \right) \\ \mathbf{J}_{22} &\stackrel{\text{def}}{=} \hat{\mathbf{H}}_{22} - \hat{\mathbf{H}}_{23}\hat{\mathbf{H}}_{32} - x \left(\hat{\mathbf{H}}_{12}\hat{\mathbf{H}}_{32} - \hat{\mathbf{H}}_{22}\hat{\mathbf{H}}_{31} \right) \end{aligned} \quad (4.6)$$

In the special case when $\mathbf{u}_0 = \mathbf{0}_{2 \times 1}$, \mathbf{J} simplifies to

$$\mathbf{J} \approx \begin{bmatrix} \hat{\mathbf{H}}_{11} - \hat{\mathbf{H}}_{31}\hat{\mathbf{H}}_{13} & \hat{\mathbf{H}}_{12} - \hat{\mathbf{H}}_{32}\hat{\mathbf{H}}_{13} \\ \hat{\mathbf{H}}_{21} - \hat{\mathbf{H}}_{31}\hat{\mathbf{H}}_{23} & \hat{\mathbf{H}}_{22} - \hat{\mathbf{H}}_{32}\hat{\mathbf{H}}_{23} \end{bmatrix} \quad (4.7)$$

Ignoring noise, we can use \mathbf{J} to generate 4 more pose constraints by differentiating Equation (4.5) with respect to \mathbf{u} :

$$J_w(\mathbf{u}_0) = J_\pi(s(\mathbf{u}_0)) J_s(\mathbf{u}_0) = \mathbf{J} \quad (4.8)$$

where J_f denotes the Jacobian of function f with

$$\begin{aligned} J_\pi(s(\mathbf{u}_0)) &= \frac{1}{[s(\mathbf{u}_0)]_3} \begin{bmatrix} \mathbf{I}_{2 \times 2}, & -\frac{1}{[s(\mathbf{u}_0)]_3} [s(\mathbf{u}_0)]_{2 \times 1} \end{bmatrix} \quad (a) \Leftrightarrow \\ J_s(\mathbf{u}_0) &= \gamma [\mathbf{I}_{2 \times 2}, -\mathbf{v}] \quad (b) \end{aligned} \quad (4.9)$$

where γ is the inverse-depth of \mathbf{u}_0 in camera coordinates defined as

$$\gamma \stackrel{\text{def}}{=} \frac{1}{[s(\mathbf{u}_0)]_3} \quad (4.10)$$

Equation (4.9-b) follows by substituting in Equation (4.3). We also have

$$J_s(\mathbf{u}_0 \in \mathbb{R}^2) = [\mathbf{R}]_{3 \times 2} \quad (4.11)$$

Substituting Equations (4.9-b) and (4.11) into Equation (4.8) gives the following problem:

$$\boxed{\begin{array}{l} \text{find } \gamma, \mathbf{R} \quad \text{s.t.} \\ \left\{ \begin{array}{ll} \gamma [\mathbf{I}_2, -\mathbf{v}] [\mathbf{R}]_{3 \times 2} = \mathbf{J} & (a) \\ [\mathbf{R}]_{3 \times 2}^\top [\mathbf{R}]_{3 \times 2} = \mathbf{I}_2 & (b) \\ \gamma > 0 & (c) \end{array} \right. \end{array}} \quad (4.12)$$

Equation (4.12-b) is applied because $\mathbf{R} \in SO_3 \Leftrightarrow [\mathbf{R}]_{3 \times 2}^\top [\mathbf{R}]_{3 \times 2} = \mathbf{I}_2$. Equation (4.12-b) is applied because the differentiation point is assumed to be in front of the camera.

We refer to Problem (4.12) as the *IPPE Problem*. This does not explicitly constrain the third column of \mathbf{R} . However, it is constrained implicitly because $\mathbf{R} \in SO_3$, so it is determined uniquely from $[\mathbf{R}]_{3 \times 2}$ with the cross-product:

$$\mathbf{R} = \left[[\mathbf{R}]_{3 \times 2}, [\mathbf{R}]_{3 \times 2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \times [\mathbf{R}]_{3 \times 2} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right] \quad (4.13)$$

Given a solution to γ and \mathbf{R} , we recover translation uniquely with

$$\begin{aligned} \mathbf{R} \text{stk}(\mathbf{u}_0, 0) + \mathbf{t} &= \frac{1}{\gamma} \text{stk}(\mathbf{v}, 1) \quad (a) \Rightarrow \\ \mathbf{t} &= \frac{1}{\gamma} \text{stk}(\mathbf{v}, 1) - [\mathbf{R}]_{3 \times 2} \mathbf{u}_0 \quad (b) \end{aligned} \quad (4.14)$$

4.2.4 Solution

We show that the IPPE problem can be reduced to finding the largest singular value of a 2×2 matrix. This is solved analytically as the roots of a uni-variate quadratic equation.

4.2.4.1 Simplification with a rotation change-of-variables

We define as $\mathbf{x}' \in \mathbb{R}^3$ the optical ray passing through the differentiation point in camera coordinates with unit depth:

$$\mathbf{x}' \stackrel{\text{def}}{=} \text{stk}(\mathbf{v}, 1) \quad (4.15)$$

We define as $\mathbf{R}_v \in SO_3$ any 3D rotation that rotate the z axis onto \mathbf{x}' . This can be implemented in several ways. In [CB14a] we used the minimal angular rotation with Rodriguez' formula:

$$\mathbf{R}_v = \mathbf{I}_{3 \times 3} + \sin \theta \mathbf{K}_x + (1 - \cos \theta) \mathbf{K}_x^2 \quad (4.16)$$

with the following terms

$$\begin{aligned} t &\stackrel{\text{def}}{=} \|\mathbf{v}\|_2 \\ s &\stackrel{\text{def}}{=} \|\text{stk}(\mathbf{v}, 1)\|_2 \\ \cos \theta &= \frac{1}{s} \\ \sin, \theta &= \sqrt{1 - \frac{1}{s^2}} \\ \mathbf{K}_x &= \frac{1}{t} \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{v} \\ \mathbf{v}^\top & 0 \end{bmatrix} \end{aligned} \quad (4.17)$$

where θ is the angle between the z axis and \mathbf{x}' . Rodriguez' formula has a singularity at $\theta = 0$, which occurs when $\mathbf{v} = \mathbf{0}_{2 \times 1}$. In practice it is stable when θ is extremely small, but when it is close to machine precision the singularity must be handled by setting $\mathbf{R}_v = \mathbf{I}_{3 \times 3}$. Alternatively, we can define \mathbf{R}_v without a singularity as follows:

$$\begin{aligned} \mathbf{R}_v &= \left[\frac{\mathbf{c} \times \mathbf{x}'}{\|\mathbf{c}\|_2 \|\mathbf{x}'\|_2}, \frac{\mathbf{c}}{\|\mathbf{c}\|_2}, \frac{\mathbf{x}'}{\|\mathbf{x}'\|_2} \right] & (a) \\ \mathbf{c} &\stackrel{\text{def}}{=} \text{stk}(1, 0, 0) - \frac{\mathbf{x}'_1}{\|\mathbf{x}'\|_2} \mathbf{x}' & (b) \end{aligned} \quad (4.18)$$

We verify that \mathbf{R}_v is a rotation matrix because the \mathbf{c} and \mathbf{x}' are orthogonal vectors. Either approach can be used to define \mathbf{R}_v . In our experimental section we use Rodriguez' formula, exactly replicating the results from [CB14a], but we now suggest using Equation (4.18) for simplicity and theoretical correctness.

We then solve Problem (4.12) with the change of variables $\tilde{\mathbf{R}} \stackrel{\text{def}}{=} \mathbf{R}_v^\top \mathbf{R}$. Once $\tilde{\mathbf{R}}$ is solved, we then recover \mathbf{R} with $\mathbf{R} = \mathbf{R}_v \tilde{\mathbf{R}}$. This change-of-variables reduces Equation (4.12-a) as follows:

$$\begin{aligned} \gamma [\mathbf{I}_2, -\mathbf{v}] [\mathbf{R}]_{3 \times 2} &= \mathbf{J} & \Leftrightarrow (a) \\ \gamma [\mathbf{I}_2, -\mathbf{v}] \mathbf{R}_v \begin{bmatrix} \tilde{\mathbf{R}} \end{bmatrix}_{3 \times 2} &= \mathbf{J} & \Leftrightarrow (b) \\ \gamma [\mathbf{B}, \mathbf{0}_{2 \times 1}] \begin{bmatrix} \tilde{\mathbf{R}} \end{bmatrix}_{3 \times 2} &= \mathbf{J} & \Leftrightarrow (c) \\ \gamma \mathbf{B} \begin{bmatrix} \tilde{\mathbf{R}} \end{bmatrix}_{2 \times 2} &= \mathbf{J} & \Leftrightarrow (d) \\ \gamma \begin{bmatrix} \tilde{\mathbf{R}} \end{bmatrix}_{2 \times 2} &= \mathbf{A} & (e) \end{aligned} \quad (4.19)$$

with

$$\begin{aligned} \mathbf{B} &\stackrel{\text{def}}{=} [\mathbf{I}_2, -\mathbf{v}] [\mathbf{R}_v]_{3 \times 2} (a) \\ \mathbf{A} &\stackrel{\text{def}}{=} \mathbf{B}^{-1} \mathbf{J} & (b) \end{aligned} \quad (4.20)$$

Equation (4.19-b) comes by substituting $\mathbf{R} \leftarrow \mathbf{R}_v \tilde{\mathbf{R}}$. Equation (4.19-c) comes because $[\mathbf{I}_2, -\mathbf{v}] \mathbf{R}_v = [\mathbf{B}, \mathbf{0}_{2 \times 1}]$ for some 2×2 matrix \mathbf{B} , from the definition of \mathbf{R}_v in Equation (4.18). Equations (4.19-d) and (4.19-e) then follow directly. We have therefore reduced Problem (4.12) to the decomposition of a 2×2 matrix \mathbf{A} into a positive scale term γ and a 2×2 sub-Stiefel matrix $\tilde{\mathbf{R}}_{22}$.

4.2.4.2 Solving γ

We solve the decomposition $\gamma \tilde{\mathbf{R}}_{22} = \mathbf{A}$ using the spectral definition of $\mathcal{SS}_{2 \times 2}$:

$$\begin{aligned} \tilde{\mathbf{R}}_{22} \in \mathcal{SS}_{2 \times 2} &\Leftrightarrow \exists \mathbf{U}, \mathbf{V}, \sigma \text{ s.t.} \\ \left\{ \begin{array}{l} \mathbf{U} \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} \mathbf{V}^\top = \tilde{\mathbf{R}}_{22} \\ \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_2, 0 \leq \sigma \leq 1 \end{array} \right. & (4.21) \end{aligned}$$

We denote an SVD of \mathbf{A} by $\mathbf{A} = \mathbf{U}_A [\text{diag}(\sigma_1^A, \sigma_2^A)] \mathbf{V}_A^\top$, with $\sigma_1^A > 0$, $\sigma_1^A \geq \sigma_2^A$ and $\mathbf{U}_A^\top \mathbf{U}_A = \mathbf{V}_A^\top \mathbf{V}_A = \mathbf{I}_2$. Because the singular values of a matrix are unique and unchanged by left and/or right unitary transformations, we can solve γ by equating singular values:

$$\begin{aligned} \gamma \tilde{\mathbf{R}}_{22} &= \mathbf{A} & \Leftrightarrow \\ \gamma \mathbf{U} \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} \mathbf{V}^\top &= \mathbf{U}_A \begin{bmatrix} \sigma_1^A & 0 \\ 0 & \sigma_2^A \end{bmatrix} \mathbf{V}_A^\top & \Rightarrow \\ \gamma &= \sigma_1^A & (4.22) \end{aligned}$$

The largest singular value of a 2×2 matrix can be computed analytically:

$$\gamma = \sigma_1^A = \sqrt{\frac{1}{2} \left(a_u + a_w + \sqrt{(a_u - a_w)^2 + 4a_v^2} \right)} \quad (4.23)$$

$$\begin{bmatrix} a_u & a_v \\ a_v & a_w \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{A}\mathbf{A}^\top$$

The solution to γ is therefore unique and it is always non-negative because it is a singular value.

4.2.4.3 Solving rotation

Because γ has a unique solution then $\tilde{\mathbf{R}}_{22} = \gamma^{-1}\mathbf{A}$ is a unique solution to $\tilde{\mathbf{R}}_{22}$. We then complete $\tilde{\mathbf{R}}$ from $\tilde{\mathbf{R}}_{22}$ using orthonormality constraints. This has two solutions in general and they are found using Algorithm 2 defined in Chapter 3 page 52. From either solution we reconstruct \mathbf{R} with $\mathbf{R} = \mathbf{R}_b\tilde{\mathbf{R}}$. We denote these two solutions as $\hat{\mathbf{R}}_1$ and $\hat{\mathbf{R}}_2$.

4.2.4.4 Solving translation

For each rotation solution there is a corresponding translation solution $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$. There are two approaches for solving them in closed-form. The first and fastest approach is to use Equation (4.14):

$$\hat{\mathbf{t}}_{j \in [1,2]} = \frac{1}{\gamma} \text{stk}(\mathbf{v}, 1) - \left[\hat{\mathbf{R}}_j \right]_{3 \times 2} \mathbf{u}_0 \quad (4.24)$$

The second approach to solve translation is slightly slower but we find that it performs better regarding noise. Here we exploit the fact that given a rotation solution, translation can be estimated in closed-form by minimizing the point correspondence error in object space as a LLS system. This writes as follows:

$$\hat{\mathbf{t}}_{j \in [1,2]} = \arg \min_{\mathbf{t}} \sum_{i=1}^n \left\| \left[\hat{\mathbf{R}}_j \text{stk}(\hat{\mathbf{u}}_i, 0) + \mathbf{t} \right]_{2 \times 1} - \left[\hat{\mathbf{R}}_j \text{stk}(\hat{\mathbf{u}}_i, 0) + \mathbf{t} \right]_3 \hat{\mathbf{q}}_i \right\|_2^2 \quad (4.25)$$

We then solve the associated normal equations, giving $\hat{\mathbf{t}}_j = (\mathbf{W}_j^\top \mathbf{W}_j)^{-1} \mathbf{W}_j \mathbf{b}_j$, where \mathbf{W}_j is a $2n \times 3$ matrix and \mathbf{b}_j is a $2n \times 1$ vector defined as

$$\mathbf{W}_j \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & \hat{\mathbf{q}}_1^x \\ \vdots & \vdots & \vdots \\ 1 & 0 & \hat{\mathbf{q}}_n^x \\ 0 & 1 & \hat{\mathbf{q}}_n^y \\ \vdots & \vdots & \vdots \\ 0 & 1 & \hat{\mathbf{q}}_n^y \end{bmatrix}, \quad \mathbf{b}_j \stackrel{\text{def}}{=} \begin{bmatrix} \hat{\mathbf{R}}_{13} - \hat{\mathbf{q}}_1^x \left(\hat{\mathbf{R}}_{33} - \hat{\mathbf{R}}_{31} \mathbf{u}_1^x - \hat{\mathbf{R}}_{32} \mathbf{u}_1^y \right) + \hat{\mathbf{R}}_{11} \mathbf{u}_1^x + \hat{\mathbf{R}}_{12} \mathbf{u}_1^y \\ \vdots \\ \hat{\mathbf{R}}_{13} - \hat{\mathbf{q}}_n^x \left(\hat{\mathbf{R}}_{33} - \hat{\mathbf{R}}_{31} \mathbf{u}_n^x - \hat{\mathbf{R}}_{32} \mathbf{u}_n^y \right) + \hat{\mathbf{R}}_{11} \mathbf{u}_n^x + \hat{\mathbf{R}}_{12} \mathbf{u}_n^y \\ \hat{\mathbf{R}}_{23} - \hat{\mathbf{q}}_1^y \left(\hat{\mathbf{R}}_{33} - \hat{\mathbf{R}}_{31} \mathbf{u}_1^x - \hat{\mathbf{R}}_{32} \mathbf{u}_1^y \right) \hat{\mathbf{R}}_{33} + \hat{\mathbf{R}}_{21} \mathbf{u}_1^x + \hat{\mathbf{R}}_{22} \mathbf{u}_1^y \\ \vdots \\ \hat{\mathbf{R}}_{23} - \hat{\mathbf{q}}_n^y \left(\hat{\mathbf{R}}_{33} - \hat{\mathbf{R}}_{31} \mathbf{u}_n^x - \hat{\mathbf{R}}_{32} \mathbf{u}_n^y \right) \hat{\mathbf{R}}_{33} + \hat{\mathbf{R}}_{21} \mathbf{u}_n^x + \hat{\mathbf{R}}_{22} \mathbf{u}_n^y \end{bmatrix} \quad (4.26)$$

The solution has a unique global optimum because \mathbf{W}_j has theoretical full-rank (equivalent to saying there exist two or more distinct image points). The computational overhead for solving is very small because $\mathbf{W}_j^\top \mathbf{W}_j$ is a 3×3 matrix and its inverse is very fast to compute analytically. Our preferred solution for translation is the second version as we find it is less sensitive to noise.

4.2.4.5 Algorithm summary

We now summarize our complete IPPE algorithm in pseudo-code. We break this down into two components. The first component is the solution to Problem (4.12). This takes as inputs \mathbf{v} and \mathbf{J} , and returns γ , $\hat{\mathbf{R}}_1$ and $\hat{\mathbf{R}}_2$. The pseudo-code is given in Algorithm 6. All steps involve only simple floating point operations. It is therefore extremely fast and it does not require any computational algebra library. The second component describes the complete pose estimation process from point correspondences and it is given in Algorithm 7. This takes as input point correspondences, the camera intrinsic matrix and the differentiation point. The homography is estimated from this data, then the motion data at the differentiation point (\mathbf{v} and \mathbf{J}) are evaluated, from which two rotations are estimated using Algorithm 6. Finally, translation is estimated by solving Equation (4.25) analytically.

So far we have not yet specified which homography estimation is used: it can be done in closed-form with various existing methods and we evaluate different methods in this chapter's evaluation section. We have also not yet specified the differentiation point. Indeed, its choice affects pose accuracy. We discuss this further in the following section.

Algorithm 6 IPPE: The solution to Problem (4.12)

Require: $\mathbf{v} \in \mathbb{R}^2$ and $\mathbf{J} \in \mathbb{R}^{2 \times 2}$, $\mathbf{J} \neq \mathbf{0}_{2 \times 2}$

- 1: **function** IPPE(\mathbf{v}, \mathbf{J})
 - 2: Compute \mathbf{R}_v from \mathbf{v} ▷ (Equation (4.18))
 - 3: $\mathbf{B} \leftarrow [\mathbf{I}_2, -\mathbf{v}] [\mathbf{R}_v]_{3 \times 2}$
 - 4: $\gamma \leftarrow \sigma_1^A$ ▷ the largest singular value of $\mathbf{A} = \mathbf{B}^{-1}\mathbf{J}$ (Equation (4.23))
 - 5: $\begin{bmatrix} \tilde{\mathbf{R}} \end{bmatrix}_{2 \times 2} \leftarrow \gamma^{-1}\mathbf{A}$
 - 6: $(\tilde{\mathbf{R}}_1, \tilde{\mathbf{R}}_2) \leftarrow \text{rotation_completion}(\begin{bmatrix} \tilde{\mathbf{R}} \end{bmatrix}_{2 \times 2})$ ▷ two rotation solutions (Algorithm (2))
 - 7: $\mathbf{R}_1 \leftarrow \mathbf{R}_v \tilde{\mathbf{R}}_1$, $\mathbf{R}_2 \leftarrow \mathbf{R}_v \tilde{\mathbf{R}}_2$
 - 8: **return** $\gamma, \mathbf{R}_1, \mathbf{R}_2$
-

Algorithm 7 Correspondence-based IPPE for perspective cameras

Require:

- $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$, $\mathbf{u}_{i \in [1, n]} \in \mathbb{R}^2$ ▷ set of $n \geq 4$ points on the object plane. These are zero centered:
 - $\sum_i \mathbf{u}_i = \mathbf{0}$
 - $\{\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \dots, \hat{\mathbf{q}}_n\}$, $\hat{\mathbf{q}}_{i \in [1, n]} \in \mathbb{R}^2$ ▷ correspondences in the image
 - \mathbf{K} ▷ camera intrinsic matrix
 - $\mathbf{u}_0 \in \mathbb{R}^2$ ▷ differentiation point
- 1: **function** IPPE($\{\mathbf{u}_i\}, \{\mathbf{q}_i\}, \mathbf{K}$)
 - 2: $\text{stk}(\hat{\mathbf{q}}_i, 1) \leftarrow \mathbf{K}^{-1} \text{stk}(\hat{\mathbf{q}}_i, 1) \quad \forall i \in [1, n]$ ▷ normalize correspondences
 - 3: $\hat{\mathbf{H}} \leftarrow \text{homog}(\{\mathbf{u}_i\}, \{\hat{\mathbf{q}}_i\})$ ▷ best fitting homography between $\{\mathbf{u}_i\}$ and $\{\hat{\mathbf{q}}_i\}$, $\mathbf{H}_{33} = 1$
 - 4: Evaluate \mathbf{v} from $\hat{\mathbf{H}}$ and \mathbf{u}_0 using Equation (4.3) or (4.4) if $\mathbf{u}_0 = \mathbf{0}_{2 \times 2}$
 - 5: Evaluate \mathbf{J} from $\hat{\mathbf{H}}$ and \mathbf{u}_0 using Equation (4.6) or (4.7) if $\mathbf{u}_0 = \mathbf{0}_{2 \times 2}$
 - 6: $(\gamma, \mathbf{R}_1, \mathbf{R}_2) \leftarrow \text{IPPE}(\mathbf{v}, \mathbf{J})$ ▷ solve inverse depth of differentiation point and pose rotations
 - 7: $\mathbf{t}_1 \leftarrow (\mathbf{W}_1^\top \mathbf{W}_1)^{-1} \mathbf{W}_1 \mathbf{b}_1$
 - 7: $\mathbf{t}_2 \leftarrow (\mathbf{W}_2^\top \mathbf{W}_2)^{-1} \mathbf{W}_2 \mathbf{b}_2$ ▷ solve translations with solution to (4.25)
 - 8: **return** $\{\mathbf{R}_1, \mathbf{t}_1\}, \{\mathbf{R}_2, \mathbf{t}_2\}$
-

4.3 IPPE theoretical analysis

4.3.1 Theorems

In this section we explore some theoretical aspects about IPPE and our method for solving it. We encapsulate this in 7 theorems and their proofs are provided in §A.4.1. The first two theorems are about the correctness of Algorithms 6 and 7:

Theorem 9. (*Algorithm 6 is NADA*) *Algorithm 6 is a Non-Artificially Degenerate Algorithm (NADA). Therefore, if a solution to problem Problem (4.12) exists, Algorithm 6 is guaranteed to find it.*

Theorem 10. (*Algorithm 7 is NADA*) *Algorithm 7 for solving plane-based pose estimation from point correspondences is NADA with the following inputs:*

1. *The spatial configuration of the object points is such that a unique homography can be estimated. This is equivalent to there existing no subset of $n - 1$ object points that are co-linear.*
2. *The homography estimation method is NADA e.g. the DLT.*
3. *The Jacobian of the homography at the differentiation point \mathbf{u}_0 is not the all-zeros matrix. In practice, this is always guaranteed by using a central differentiation point.*

Consequently, Algorithm 6 is guaranteed to return the correct solution for any input that satisfies the three conditions. A differentiation point that always guarantees item 3 is a *central differentiation point*, defined as the centroid of the object points. The central differentiation point has other important properties that make it a particularly good choice, explained below.

The next theorem fully characterizes the geometry of the two pose solutions from IPPE. This is then used to explain the behavior of IPPE in *quasi-affine conditions*, which occur when the object is small and/or far to the camera, causing the homography to be quasi-affine. In these conditions there are approximately two rotation solutions to plane-based pose estimation illustrated in Figure 1.2. Quasi-affine conditions cause previous homography decomposition methods to fail, but they do not cause IPPE to fail.

Theorem 11. (*Solution geometry*) *The two rotation solutions from Algorithm 6 (and hence from Algorithm 7) are illustrated in Figure 4.1. The pose of the object plane from both solution is equal up to a reflection about a plane whose normal points along the optical ray \mathbf{x}' that passes through the differentiation point \mathbf{u}_0 . The position of \mathbf{u}_0 in camera coordinates is the same with both poses. Consequently, the two poses are identical if and only if the normal of the object plane in camera coordinates is co-linear to \mathbf{x}' .*

From Theorem 11, we can see that by changing the differentiation point we change the geometry of the pose solutions. Using this fact, we can then ensure that IPPE performs correctly in quasi-affine conditions using a central differentiation point with the following theorem:

Theorem 12. (*Correctness in quasi-affine conditions using a central differentiation point*) *Recall that in quasi-affine conditions there are two rotation matrices that can explain the data up to noise shown in Figure 1.2. These are correctly recovered using a central differentiation point, which is a direct result of Theorem 11.*

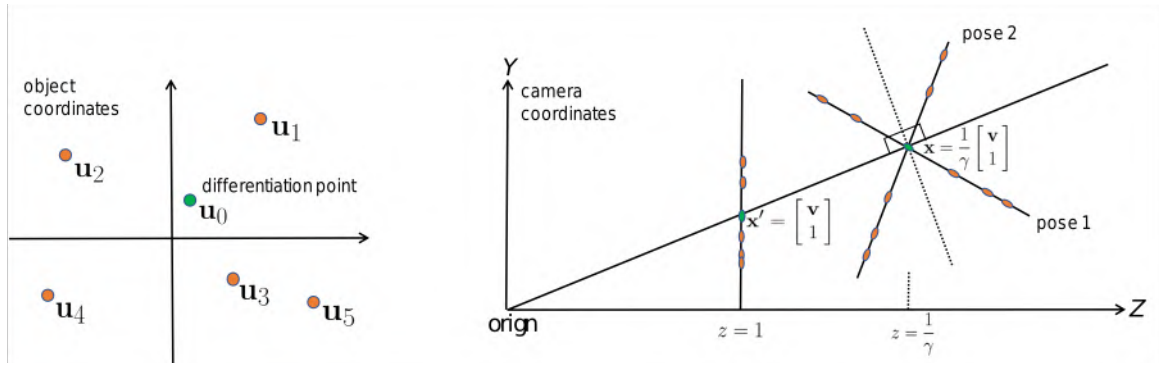


Figure 4.1: Geometric relationship between the two pose solutions produced by IPPE. We visualize camera coordinates with a cross-section in the plane $X = 0$.

The next theorem explains the benefit of using a central differentiation point in terms of errors-in-variables and error propagation:

Theorem 13. (*Errors-in-variables using a central differentiation point*) *Because of noise in the homography matrix, the choice of the differentiation point u_0 affects solution accuracy because of errors in v and J . Using a central differentiation point reduces errors-in-variables in the IPPE problem.*

Theorems ?? and 13 tell us the benefits of using a central differentiation point. Our final theorem reveals a deep relationship between IPPE and the Perspective-3-Point problem (P3P), and it helps complete our understanding of P3P when the distance between points is small relative to their depth.

Theorem 14. (*Relationship between IPPE and P3P*) *When the separation of three object points tends to zero relative to their depth, the perspective effects diminish and the P3P problem tends to the IPPE problem. Consequently, the solution geometry of P3P (which has 4 solutions in general) tends to the solution geometry of IPPE when the point separation tends to zero.*

4.3.2 Pose disambiguation using reprojection error

We discuss in this section how to establish the correct pose given by IPPE using reprojection error. For noiseless measurements we have the following theorem:

Theorem 15. *Pose can always be disambiguated with at least three non-co-linear object points and when perspective effects are non-negligible.*

The proof is straightforward and given in §??. With noise, disambiguation can be harder and requires a confidence judgment by inspecting the reprojection error of both solutions. Pose is ambiguous if the reprojection errors of each pose is similar and indistinguishable to noise. Thanks to Theorem 12, by using IPPE with a central differentiation point, the two poses correspond to the two-fold ambiguity in quasi-affine conditions, and this causes e_1 and e_2 to be similar. A decision to reject the pose with the higher reprojection error depends on the amount of perspective effects and we measure it with a likelihood ratio test. Let $e(\mathbf{R}, \mathbf{t}) : SE_3 \rightarrow \mathbb{R}^+$ be the L_2 reprojection error of a pose (\mathbf{R}, \mathbf{t}) . Without loss of generality let $e(\hat{\mathbf{R}}_1, \hat{\mathbf{t}}_1) \leq e(\hat{\mathbf{R}}_2, \hat{\mathbf{t}}_2)$. The likelihood ratio is

$$\lambda = \frac{\mathcal{L}(\mathbf{R}_1 \mathbf{t}_1)}{\mathcal{L}(\mathbf{R}_2 \mathbf{t}_2)} \quad (4.27)$$

where \mathcal{L} is the pose likelihood. For zero mean I.I.D Gaussian noise with variance σ , $\log(\lambda) = e(\hat{\mathbf{R}}_1, \hat{\mathbf{t}}_1) - e(\hat{\mathbf{R}}_1, \hat{\mathbf{t}}_1) + K$ where K depends on σ . We reject the second pose as an alternative hypothesis if $\lambda \geq c$, where c is a threshold that trades-off precision and recall. We do not set c ourselves because it is application specific. Instead we return both poses and their respective reprojection errors, and it is up to the application to decide whether or not to reject the second pose estimate.

4.4 Experimental evaluation with simulated data

In this section we give a detailed comparison of the performance of IPPE using simulation experiments. We break this section into three parts. The first part compares IPPE against PHD methods using five different methods to estimate the homography. We have found IPPE combined with Harker and O’Leary’s method [HO05] to perform the best. This performs marginally better than the normalized DLT and it is approximately the same as the ML estimate found with iterative refinement. We call this combination IPPE+HO.

In the second section we compare IPPE+HO against prior state-of-the-art PnP methods with real-time performance. We give a detailed breakdown of this comparison along two axes. The first is the number of correspondences n , which we break down into small n (*i.e.* between 4 and 10) and medium-to-large n (*i.e.* between 8 and 50). The second axis is broken down into simulations where the PPE problem is unambiguous, and simulations where it is ambiguous. Ambiguous simulations are caused by quasi-affine conditions, with two pose solutions that can reasonably explain the image data. In these cases we do not force the methods to return a single solution, but instead they can return multiple solutions. The best of these solutions with respect to ground-truth is used to measure method accuracy. In contrast, for unambiguous cases the methods are forced to return a single solution as the one with smallest L2 reprojection error, and it is only this solution which is evaluated. In the third section we compare IPPE against solving pose via P3P, using three virtual correspondences estimated from the homography.

4.4.1 Simulation setup

We use a testing framework similar to [LMF09; LXX12]. A perspective camera is setup and a planar object is transformed to camera coordinates and projected into the camera’s image. The object covers a zero-centred square region on the plane $z = 0$ with variable width w . The camera image size is 640×480 px and the intrinsic matrix is

$$\mathbf{K} = \begin{bmatrix} f & 0 & 320 \\ 0 & f & 240 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.28)$$

with focal length f using a default $f = 800$ px. We simulate poses as follows. We uniformly sample a point in the image $\hat{\mathbf{p}}$ and create the ray $\mathbf{w} = \text{stk}(\hat{\mathbf{p}}, 1)$. We then project this ray out to a random depth d drawn uniformly from the interval $d \sim U(\frac{f}{2}, 2f)$. We then compute the translation as $\mathbf{t} = d\mathbf{w}$. The rotation \mathbf{R} is computed as follows. We first create an in-plane rotation $\mathbf{R}(\theta)$, $\hat{\theta} \sim U(0, 2\pi)$. This is followed by an out-of-plane rotation $\mathbf{R}(\psi, q_x, q_y)$ with axis $\mathbf{r} = \text{stk}(q_x, q_y, 0)$, $q_x, q_y \sim U(-1, +1)$. The angle is $\psi \sim U(0, \psi_{max})$ where ψ_{max} denotes the maximum angle in radians such that the plane’s tilt angle with respect to the viewing ray is less than 80 degrees. The rotation is given by $\mathbf{R} = \mathbf{R}(\psi, q_x, q_y)\mathbf{R}(\theta)$. We then synthesize n point correspondences. Their positions in the object

plane are $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ with $\mathbf{u}_{i \in [1, n]} \sim \text{U}(-\frac{w}{2}, +\frac{w}{2})$. These points are then projected in the image via $\{\mathbf{K}, \mathbf{R}, \mathbf{t}\}$ to give their corresponding image positions $\{\mathbf{q}_1, \dots, \mathbf{q}_n\}$. To measure an algorithm’s sensitivity to noise in the image, we perturb each image point with IID zero-mean Gaussian noise with standard deviation σ_I . We also test sensitivity to noise in the object plane by perturbing each object point with noise of standard deviation σ_M . We keep only poses where all point correspondences lie in front of the camera and project within the image.

4.4.2 Well-posed and ill-posed conditions

In the special case when $\sigma_I = \sigma_M = 0$ planar pose is recoverable uniquely. When $\sigma_I > 0$ and/or $\sigma_M > 0$ there may be instances when pose estimation is ambiguous. That is, an alternative rigid hypothesis exists with similar reprojection error up to noise. It is important to separate ambiguous from unambiguous cases. In an ambiguous case a method returning a single solution may pick an incorrect pose similar to the alternative pose hypothesis. In this case it is not the algorithm which is to blame for these errors but the posedness of the problem. We therefore measure performance for each algorithm in two modes.

Mode 1 is where each algorithm returns *one* solution. The PHD methods always return one solution. IPPE, and most competitive PnP methods can return multiple solutions. In mode 1 we force these algorithms to return the solution with lowest L2 reprojection error. In order to obtain meaningful statistics we must ensure that test samples in mode 1 are sufficiently unambiguous. In §4.3.2 we have shown that pose is ambiguous when an affine homography can model the transformation between correspondences. To judge whether a problem is ambiguous, we measure how many times more likely the data is predicted by a perspective homography \mathbf{H}_p than an affine homography \mathbf{H}_a . We compute \mathbf{H}_p with the ground truth transform (\mathbf{R}, \mathbf{t}) and refine with Gauss-Newton iterations. We compute \mathbf{H}_a with a least squares fit of the correspondences (which is also the ML estimate for affine projection). We then measure the log-likelihood ratio:

$$D = l(\{\mathbf{q}_1, \dots, \mathbf{q}_n\}; \{\mathbf{u}_1, \dots, \mathbf{u}_n\}, \sigma_I, \mathbf{H}_p) - l(\{\mathbf{q}_1, \dots, \mathbf{q}_n\}; \{\mathbf{u}_1, \dots, \mathbf{u}_n\}, \sigma_I, \mathbf{H}_a) \quad (4.29)$$

where l is the data log-likelihood with IID. Gaussian noise. We judge a sample to be ambiguous if $D < \tau_a$. Only unambiguous samples are selected for testing algorithms in mode 1. A small τ_a means that more samples are rejected as being ambiguous whereas a larger τ_a means fewer. It is not critical for us to finely tune τ_a , we merely wish to select a value which eliminates cases which are clearly ambiguous to ensure that algorithms tested in mode 1 are tested in well-conditioned cases. In mode 1 we use $\tau_a = 5$. *Mode 2* is when we keep *all* samples, and allow algorithms to return multiple solutions.

4.4.3 Summary of experimental parameters and error metrics

In Table 1 we give a summary of the experimental free parameters for the simulation experiments. We denote $\{\hat{\mathbf{R}}, \hat{\mathbf{t}}\}$ to be the rotation and translation estimated by an algorithm. Similarly to previous works [LMF09; LXX12] we measure error with two metrics:

1. $RE(\hat{\mathbf{R}})$. The Rotational Error (in degrees). This is the angle of the minimal rotation needed to align $\hat{\mathbf{R}}$ to \mathbf{R} . This is computed from the angle of the axis/angle representation of $\hat{\mathbf{R}}^\top \mathbf{R}$.
2. $TE(\hat{\mathbf{t}})$. The Translation Error (%). This is the relative error in translation, given by $TE(\hat{\mathbf{t}}, \mathbf{t}) = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 / \|\mathbf{t}\|_2$.

Parameter	Meaning
f	Focal length
w	Model plane width
n	Number of correspondences
σ_I	Correspondence noise (image)
σ_M	Correspondence noise (model)
mode	Either in mode 1 or mode 2 (§4.4.2)

Table 4.1: Free parameters in the simulation experiments.

For each error metric we measure three statistics; the standard deviation, the mean and median error.

4.4.4 IPPE versus PHD methods

We start by comparing IPPE against the two existing PHD methods, which we denote by HDSt [Stu00] HDZh [Zha00]. Because these return only a single solution we perform the tests in mode 1 (*i.e.* unambiguous cases). We compare with 5 different Homography Estimation (HE) methods. This is to (*i*) assess the sensitivity of an algorithm with respect to the choice of HE method, and (*ii*) to determine which HE method leads to best pose estimates. The HE methods we test are as follows:

1. **DLT** (non iterative). The normalized Direct Linear Transform [HZ04].
2. **TAUB** (non iterative). The Taubin estimate [Tau91].
3. **HO** (non iterative). Harker and O’Leary [HO05] using Total Least Squares (TLS).
4. **ML** (iterative). ML minimizer using Gauss-Newton iterations. ML is initialized with the best non-iterative solution from DLT, TAUB and HO.

We also compare results with iterative pose refinement implemented with Gauss-Newton. We refer this as **GEOMREF**, which is initialized using estimated pose among all methods that has the lowest reprojection error. When $\sigma_M = 0$ we use the ML error as the loss function. When $\sigma_M > 0$ we use the symmetric transfer error as the loss function.

We have run a series of 5 experiments (E1 to E5) by varying the parameters in Table 1 to cover a range of problem conditions. Note that there is redundancy in scaling both f and w , therefore we keep f constant and only vary w . The parameter values for each experiment are shown in Table 2. We present summary statistics over 5,000 simulated poses in Table 4.3 to 4.7. For each HE method, we have highlighted in blue the pose estimation method which gives the lowest mean error. That method is IPPE for every experiment, for every noise level and for every HE method. This shows the considerable performance improvement IPPE has over the PHD methods across the board. Note that these tests are performed in mode 1 which are not ambiguous cases, so IPPE is not benefiting by returning two solutions. The single solution produced by IPPE with lowest reprojection error is clearly more accurate in general than the single estimate from HDZh and HDSt in all test conditions by a wide margin.

Considering differences between HE methods, TAUB consistently performs the worst for HDZh, HDSt and IPPE. We see that using the DLT gives lowest errors for HDZh and HDSt. The best performing closed-form HE method (among TAUB, DLT and HO) for IPPE is HO, which is very closely followed by DLT. HO is also the fastest method; between 5-6 times faster to compute than the DLT [HO05]. There appears to be little benefit using ML over HO for IPPE. A visual comparison of methods is shown as graphs in Figure 4.2. The five rows correspond to the five experiments, and the

4.4. EXPERIMENTAL EVALUATION WITH SIMULATED DATA

columns show mean and median errors in rotation and translation. To reduce clutter we plot results only with the best performing closed-form HE method for HDZh and HDSt (DLT). These experiments show that IPPE is able to estimate pose much better from homography matrices compared to the PHD methods.

	E1	E2	E3	E4	E5
f	800	800	800	800	800
w	200	300	200→400	250	350
n	10	5→40	12	15	8
σ_I	0→6	2	3	3.5	3.5
σ_M	0	0	0	0→7	0→5
mode	1	1	1	1	1

Table 4.2: Varying imaging conditions in simulation experiments E1-E5.

E1: RE		TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	
0.632	1.06±2.44	6.76±6.68	6.87±6.8	1.11±2.89	6.83±6.92	6.93±7.07	0.949±2.09	6.43±6.57	6.53±6.68	0.948±2.09	6.43±6.55	6.53±6.67	
1.58	2.52±5.69	12.5±10.8	12.6±11	2.63±6.8	12.7±11.6	12.8±11.8	2.23±5.2	11.9±10.6	12±10.7	2.23±5.2	11.9±10.5	12±10.7	
2.21	3.28±8.03	14.2±12.6	14.4±12.9	3.22±7.79	13.7±12.4	13.9±12.7	2.99±7.82	13±11.9	13.2±12.2	2.93±7.48	13±11.8	13.2±12.1	
3.16	4.05±7.6	15.8±13.2	16.1±13.7	4.33±9.18	15.2±13.4	15.6±14.1	3.66±7.66	14.3±12.4	14.7±13.1	3.79±8.39	14.1±12.1	14.5±12.7	
3.79	4.7±8.85	16.3±12.7	16.6±12.9	4.84±10.3	15.9±13.3	16.2±13.7	4.07±8.73	14.9±12.2	15.2±12.4	4.07±8.73	14.9±12.1	15.1±12.3	
E1: TE		TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	
0.632	0.42±0.401	3.34±4.48	3.26±4.21	0.437±0.41	3.37±4.48	3.29±4.24	0.403±0.377	3.19±4.37	3.12±4.13	0.403±0.377	3.18±4.36	3.11±4.13	
1.58	0.95±0.873	6.87±8.75	6.43±7.72	0.965±0.884	6.85±8.9	6.41±7.81	0.91±0.83	6.48±8.41	6.1±7.43	0.908±0.834	6.45±8.38	6.08±7.41	
2.21	1.2±1.12	7.98±9.87	7.32±8.53	1.2±1.17	7.98±10.2	7.29±8.77	1.13±1.07	7.45±9.61	6.86±8.33	1.12±1.06	7.41±9.55	6.83±8.25	
3.16	1.6±1.46	8.84±10.3	8.12±8.89	1.59±1.37	8.37±10.1	7.71±8.72	1.49±1.32	8.01±9.56	7.4±8.28	1.48±1.32	7.91±9.34	7.33±8.09	
3.79	1.83±1.6	9.31±10.1	8.48±8.64	1.84±1.61	9.01±10.2	8.15±8.66	1.7±1.45	8.51±9.58	7.75±8.2	1.7±1.46	8.46±9.47	7.71±8.13	

Table 4.3: IPPE versus PHD methods showing mean RE (degrees) and mean TE (%) with standard deviation : Experiment E1

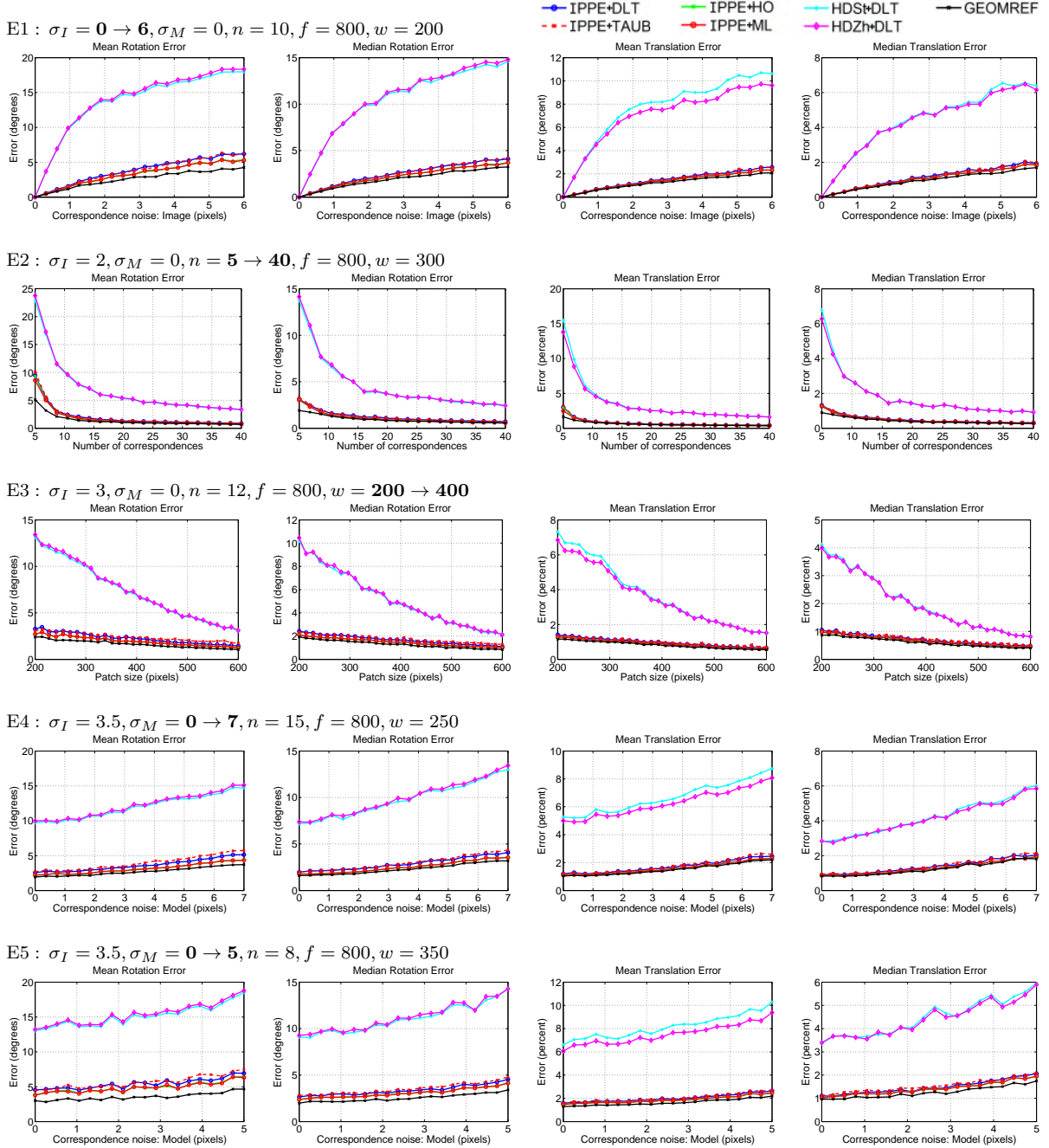


Figure 4.2: Experimental results with simulated data: Comparing the pose estimation accuracy of IPPE with PHD methods (E1-E5).

4.4. EXPERIMENTAL EVALUATION WITH SIMULATED DATA

E2: <i>RE</i>		TAUB			DLT			HO			ML		
<i>n</i>	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
5	10.1±22.1	23.4±25.7	24.2±26.8	9.46±21	22.9±25	23.8±26.3	9.33±21.3	22.3±24.6	23.1±25.7	8.59±19.2	22±24.1	22.7±25.2	
9	3.14±7.28	11.4±11.3	11.6±11.3	2.98±6.59	11.4±11.6	11.5±11.6	2.79±6.95	10.7±10.8	10.7±10.7	2.75±6.78	10.6±10.7	10.7±10.7	
14	1.75±2.01	7.55±6.79	7.67±6.88	1.79±3.41	7.05±6.74	7.14±6.79	1.51±2.84	6.7±6.32	6.78±6.38	1.55±3.19	6.66±6.34	6.74±6.4	
18	1.52±1.31	6.11±5.37	6.23±5.61	1.51±1.2	5.71±5.23	5.8±5.33	1.29±0.985	5.28±4.81	5.37±4.93	1.29±0.986	5.28±4.8	5.36±4.93	
23	1.25±1.11	4.95±4.11	5.05±4.2	1.21±0.944	4.57±4.03	4.64±4.09	1.01±0.779	4.32±3.81	4.39±3.88	1.01±0.78	4.31±3.8	4.38±3.87	
27	1.2±1.17	4.73±4.07	4.8±4.14	1.15±0.921	4.32±3.86	4.39±3.91	0.96±0.732	4.08±3.62	4.14±3.68	0.959±0.733	4.08±3.61	4.13±3.67	
E2: <i>TE</i>		TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
5	3.23±10.4	16.1±21.1	14.4±19.1	3.01±10.1	15.5±20.5	13.8±18.5	2.91±9.18	15.2±20.2	13.6±18.2	2.49±6.62	14.6±19.2	13±17.1	
9	1.14±1.25	7.85±9.83	7.22±9.01	1.15±1.22	6.07±8.63	5.65±7.49	1.07±1.13	5.67±8.23	5.26±7.14	1.07±1.14	5.62±8.07	5.23±7	
14	0.785±0.688	3.64±4.54	3.56±4.26	0.765±0.674	3.56±4.61	3.47±4.3	0.72±0.61	3.34±4.27	3.26±4.03	0.719±0.611	3.31±4.24	3.24±4.01	
18	0.669±0.581	3.05±3.73	2.99±3.51	0.663±0.601	2.83±3.48	2.77±3.28	0.61±0.541	2.66±3.3	2.62±3.13	0.61±0.54	2.64±3.3	2.6±3.12	
23	0.533±0.458	2.45±2.81	2.42±2.71	0.518±0.466	2.19±2.67	2.18±2.6	0.488±0.43	2.11±2.57	2.1±2.5	0.487±0.43	2.1±2.55	2.08±2.48	
27	0.515±0.438	2.28±2.69	2.26±2.62	0.512±0.437	2.19±2.57	2.18±2.53	0.469±0.397	2.06±2.43	2.06±2.39	0.468±0.397	2.05±2.43	2.04±2.39	

Table 4.4: IPPE versus PHD methods showing mean RE (degrees) and mean TE (%) with standard deviation : experiment E2

E3: <i>RE</i>		TAUB			DLT			HO			ML		
<i>w</i>	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
200	3.07±4.36	13.5±10.5	13.7±10.8	3.27±5.9	13.1±10.5	13.4±11	2.7±4.19	12.3±9.92	12.5±10.2	2.7±4.18	12.2±9.84	12.5±10.2	
228	3.12±5.19	12.5±10.5	12.8±10.9	2.96±4.06	11.9±10.1	12.2±10.5	2.61±4.12	11.2±9.83	11.5±10.2	2.6±4.11	11.2±9.67	11.5±10.1	
269	2.94±4.11	11.2±9.66	11.5±10	2.91±5.7	10.8±10	11±10.2	2.51±4.05	10.1±9.2	10.3±9.49	2.51±4.05	10±9.02	10.2±9.3	
297	2.77±4.72	10.6±9.33	10.8±9.44	2.75±5.33	10.1±9.28	10.2±9.4	2.33±4.59	9.5±8.71	9.63±8.79	2.32±4.59	9.49±8.81	9.64±8.9	
338	2.73±4.01	9.22±8.51	9.36±8.67	2.61±4.34	8.52±8.39	8.6±8.47	2.28±3.62	8.1±7.82	8.2±7.94	2.28±3.62	8.13±7.94	8.22±8.06	
366	2.39±3.08	8.38±7.54	8.52±7.57	2.26±3.24	7.88±7.47	8.01±7.59	1.95±2.6	7.3±6.77	7.42±6.84	1.95±2.61	7.29±6.8	7.4±6.85	
E3: <i>TE</i>		TAUB			DLT			HO			ML		
<i>w</i>	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
200	1.35±1.2	7.66±9.23	7.09±8.03	1.41±1.29	7.33±8.9	6.84±7.77	1.28±1.17	7.02±8.59	6.54±7.51	1.28±1.16	6.97±8.53	6.49±7.45	
228	1.33±1.12	6.88±8.45	6.42±7.46	1.34±1.21	6.65±8.29	6.2±7.29	1.23±1.05	6.23±7.93	5.81±7.05	1.23±1.05	6.16±7.86	5.76±7	
269	1.22±1.05	6.21±7.92	5.76±6.91	1.2±1.04	5.98±7.7	5.56±6.73	1.1±0.946	5.61±7.31	5.24±6.42	1.1±0.946	5.56±7.18	5.2±6.29	
297	1.12±1.03	5.85±7.12	5.48±6.27	1.12±1.02	5.39±6.75	5.08±5.99	1.04±0.945	5.15±6.43	4.87±5.72	1.04±0.943	5.08±6.34	4.81±5.67	
338	1.13±1.02	4.58±5.77	4.41±5.3	1.08±1	4.17±5.28	4.02±4.85	1.02±0.933	4.03±5.09	3.9±4.72	1.02±0.935	4±5.02	3.87±4.68	
366	0.993±0.869	4.04±4.73	3.9±4.37	0.985±0.868	3.91±4.9	3.82±4.59	0.906±0.803	3.67±4.47	3.59±4.22	0.905±0.8	3.65±4.45	3.57±4.21	

Table 4.5: IPPE versus PHD methods showing mean RE (degrees) and mean TE (%) with standard deviation (experiment E3)

E4: <i>RE</i>		TAUB			DLT			HO			ML		
σ_M	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
0	2.67±3.53	10.3±7.91	10.6±8.19	2.53±3.94	9.71±8.19	9.98±8.48	2.23±3.29	9.11±7.43	9.38±7.71	2.18±2.77	9.06±7.4	9.33±7.68	
0.737	2.82±3.81	10.7±8.23	10.9±8.52	2.66±3.27	9.7±8.21	9.91±8.44	2.44±5.09	9.23±7.55	9.4±7.76	2.44±5.1	9.16±7.43	9.33±7.62	
1.84	3±3.09	11.8±9.3	12±9.48	2.95±4.5	10.7±8.95	10.8±9.04	2.45±2.59	10.2±8.39	10.3±8.5	2.46±2.66	10.1±8.36	10.3±8.46	
2.58	3.37±4.15	12.3±9.42	12.6±9.67	3.24±3.92	11.2±8.99	11.5±9.22	2.8±3.38	10.6±8.36	10.8±8.55	2.79±3.38	10.5±8.29	10.8±8.47	
3.68	3.97±6.09	13.4±9.42	13.7±9.52	3.57±5.17	12.1±9.38	12.2±9.49	3.13±5.06	11.5±8.66	11.7±8.75	3.06±4.5	11.4±8.52	11.6±8.62	
4.42	4.16±4.08	14.7±9.75	15±9.98	3.9±4.36	13±8.92	13.1±9.06	3.28±2.67	12.3±8.38	12.5±8.52	3.28±2.66	12.2±8.37	12.4±8.52	
E4: <i>TE</i>		TAUB			DLT			HO			ML		
σ_M	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
0	1.22±1.06	5.85±6.91	5.54±6.16	1.19±1.06	5.26±6.47	5±5.85	1.11±0.954	5.01±6.19	4.79±5.63	1.11±0.948	5±6.14	4.78±5.57	
0.737	1.21±1.09	5.69±7.06	5.39±6.36	1.19±1.03	5.22±6.38	4.94±5.71	1.1±0.942	4.99±6.21	4.75±5.6	1.1±0.943	4.96±6.12	4.73±5.52	
1.84	1.38±1.12	6.28±7.32	5.94±6.56	1.38±1.17	5.63±6.51	5.37±5.82	1.26±1.05	5.35±6.24	5.09±5.61	1.26±1.05	5.32±6.14	5.05±5.54	
2.58	1.47±1.19	6.81±7.65	6.38±6.75	1.47±1.21	6.22±7.33	5.85±6.43	1.37±1.08	5.88±6.9	5.54±6.1	1.37±1.08	5.83±6.83	5.5±6.05	
3.68	1.75±1.71	7.52±8.22	7.03±7.29	1.66±1.33	6.56±7.29	6.2±6.46	1.55±1.23	6.4±7.14	6.05±6.38	1.55±1.25	6.31±7.02	5.98±6.28	
4.42	1.9±1.52	8.07±8.66	7.46±7.59	1.82±1.49	7.24±8.13	6.73±7.1	1.69±1.32	6.88±7.76	6.41±6.8	1.69±1.32	6.74±7.63	6.3±6.7	

Table 4.6: IPPE versus PHD methods showing mean RE (degrees) and mean TE (%) with standard deviation (experiment E4)

E5: <i>RE</i>		TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
0	4.49±9.32	13.8±13.4	14.1±13.8	4.54±10.4	13±13.4	13.2±13.7	3.83±8.71	12.4±12.6	12.6±13	3.79±8.54	12.3±12.5	12.5±12.8	
0.526	4.73±9	14.5±13.7	14.7±14	4.78±9.8	13.8±13.2	14±13.5	4.41±9.69	13.3±13	13.4±13.2	4.36±9.57	13.1±13	13.3±13.2	
1.32	4.86±9.02	14.2±13.3	14.4±13.5	4.81±10.2	13.7±13.4	13.9±13.6	4.47±9.79	13±12.8	13.1±13	4.47±9.81	12.9±12.5	13.1±12.7	
1.84	5.26±10.1	15.7±14.6	15.9±14.8	5.4±11.5	15.1±15.6	15.3±15.9	4.68±9.86	14.3±14.4	14.4±14.6	4.72±10.3	14.2±14.4	14.4±14.6	
2.63	5.59±10.9	15.6±13.2	15.9±13.6	5.57±11.6	14.9±12.8	15.2±13.2	4.78±9.91	14.3±12.7	14.6±13.2	4.93±10.6	14.1±12.4	14.4±13	
3.16	5.85±10.8	16.3±13.8	16.7±14.3	5.89±12.1	15.5±14.2	15.9±14.8	5.08±10.3	14.8±13.4	15.2±14.1	5.26±11.1	14.8±13.4	15.2±14	
E5: <i>TE</i>		TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	
0	1.6±1.55	7.23±9.51	6.61±8.21	1.58±1.54	6.61±8.74	6.07±7.55	1.45±1.39	6.41±8.61	5.91±7.46	1.46±1.39	6.35±8.46	5.87±7.34	
0.526	1.7±1.6	7.49±9.76	6.98±8.51	1.66±1.58	7.14±9.47	6.62±8.25	1.58±1.49	6.84±9.15	6.39±8	1.57±1.48	6.72±8.95	6.29±7.79	
1.32	1.81±1.74	7.49±9.43	6.93±8.13	1.75±1.67	7.14±9.36	6.67±8.15	1.62±1.5	6.86±9.25	6.43±8.02	1.63±1.49	6.82±9.06	6.38±7.87	
1.84	1.91±1.78	8.36±10.9	7.68±9.36	1.87±1.76	7.82±10.3	7.23±8.88	1.74±1.6	7.45±9.93	6.94±8.6	1.75±1.6	7.31±9.63	6.82±8.35	
2.63	1.95±1.82	8.83±10.6	8.11±9.19	1.9±1.83	8.28±9.99	7.65±8.74	1.8±1.59	7.98±9.89	7.39±8.57	1.8±1.56	7.84±9.62	7.26±8.3	
3.16	2.15±1.96	8.83±10.4	8.15±9.05	2.06±1.87	8.35±10.3	7.75±8.98	1.92±1.7	8±9.86	7.46±8.62	1.91±1.69	7.85±9.59	7.35±8.43	

Table 4.7: IPPE versus PHD methods showing mean RE (degrees) and mean TE (%) with standard deviation (experiment E5)

4.4.5 IPPE versus PnP methods

We compared IPPE against various PnP methods with simulated data. The following names are used for the compared methods:

- **IPPE+HO** (non iterative): Proposed method using homography estimated with [HO05].
- **RPP-SP** (non iterative): Schweighofer and Pinz [SP06]. This is the extension of Lu *et al.* [Lu'2000] to handle ambiguities.
- **EPnP** (non iterative): Lepetit *et al.* [LMF09].
- **RPnP** (non iterative): Li *et al.* [LXX12].
- **HDZh+DLT** (non iterative): The best performing PHD method.
- **GEOMREF** (iterative): Iterative refinement of best pose among all methods.

We start with a series of mode 1 experiments. We divided these into two parts. The first part measures performance when the number of correspondences is medium-to-large ($n = 8 \rightarrow 50$). The second part measures performance when the number of correspondences is small ($n = 4 \rightarrow 10$). We make this division to assist visualizing results as methods perform far better with larger n . The division also helps study two properties; the accuracy of an algorithm with low numbers of correspondences and how well an algorithm exploits correspondence redundancy. In total we perform 12 experiments (E6-E17). There are 6 for $n = 4 \rightarrow 10$ (E6-E11) and 6 for $n = 8 \rightarrow 50$ (E12-E17). The experimental parameters are shown in Table 4.8.

	E6	E7	E8	E9	E10	E11
f	800	800	800	800	800	800
w	300	300	300	300	300	300
n	8→50	8→50	8→50	8→50	8→50	8→50
σ_I	0.5	3	8	2	2	2
σ_M	0	0	0	0.5	3	8
Mode	1	1	1	1	1	1

	E12	E13	E14	E15	E16	E17
f	800	800	800	800	800	800
w	300	300	300	300	300	300
n	4→10	4→10	4→10	4→10	4→10	4→10
σ_I	0.5	3	8	2	2	2
σ_M	0	0	0	0.5	3	8
Mode	1	1	1	1	1	1

Table 4.8: Varying imaging conditions in simulation experiments E6-E17.

4.4.5.1 Medium-to-large n

The results for experiments E6-E11 are shown in Figure 4.3. With respect to rotation we see that across all conditions IPPE+HO is consistently the best performing method (excluding refinement with GEOMREF). There is a clear improvement in performance with respect to the next best non-iterative method (RPnP). The performance of RPP-SP with respect to mean error remains larger than IPPE+HO. With respect to median error, RPP-SP approaches but never exceeds IPPE+HO

for larger n . When n goes beyond 15 the performance of IPPE+HO is very close to GEOMREF. Turning to translation error we see a similar ranking of methods. The difference between IPPE+HO and RPP-SP is smaller than for rotation error. There is negligible difference between IPPE+HO and RPP-SP in translation performance in experiments E9-E11 (when noise increases in the model). The next best non-iterative method (RPnP) performs behind IPPE+HO and RPP-SP with respect to translation error for all experiments.

We can see that IPPE+HO is the best performing non-iterative method in the range $n = 8 \rightarrow 50$. We also see that beyond $n = 15$ the performance gains in refining the IPPE+HO solution with GEOMREF are very small in all experiments. This is true when there is correspondence noise in the image, model, or both. The same cannot be said in all experiments for the other methods. This has important practical implications as it suggests that when speed is an important priority, one can do away with iterative refinement and use the IPPE+HO solution. A rule of thumb would be when $n > 15$.

4.4.5.2 Small n

We now turn to the performance evaluation with $n = 4 \rightarrow 10$. The results are shown in Figure 4.4. Here we see that for $n \geq 6$ IPPE+HO is the best performing method (excluding GEOMREF) with respect to rotation across all conditions. For $n \geq 6$ IPPE+HO performs as well as or better than the next best method (RPP-SP) with respect to translation. For $n = 4$ IPPE+HO is outperformed by RPnP and RPP-SP. RPnP does well for $n = 4$, although there is a clear performance gap between RPnP and GEOMREF. This gap is larger for larger σ_M , indicating RPnP has difficulty with noise in the model. The performance of IPPE+HO is significantly worse at $n = 4$ than $n = 5$. The reason is two-fold. Firstly the homography is computed from 4 point correspondences, and because of the lack of redundancy the homography fits to noise. For $n > 4$ there is redundancy and this leads to considerably lower error. The second reason is that the configuration of correspondences in the model affects the sensitivity of homography estimation to noise. Because the correspondences are uniformly sampled on the model plane some configurations can lead to a poorer conditioning of the homography estimation problem. We refer the reader to [CS09] where a detailed analysis is given on the stability of homography estimation by 1st-order perturbation theory.

Experiments E12-E17 suggest that IPPE+HO should not be used when $n < 6$, as better results would be obtained with RPnP. However, this is not universally true because the performance of IPPE depends on the spatial configuration of the point. We now study the case when the object points are not drawn randomly, but where four points are located on corners of the square region: $(u, v)_1 = 1/2(w, w)$, $(u, v)_2 = \frac{1}{2}(w, -w)$, $(u, v)_3 = \frac{1}{2}(-w, -w)$, $(u, v)_4 = \frac{1}{2}(-w, w)$. This is typically the case in AR-based planar pose estimation. The remaining $n - 4$ points are positioned with uniform probability within the region. We then studied the performances in these configurations. We ran six experiments (E18-E23) using this new sampling scheme. The experimental parameters are listed in Table 4.9. These are the same as experiments E6-E11, but we have reduced the plane size from 300 to 100. The reason for this is that the new sampling scheme means the correspondences span a larger region on the model, and thus reduces the influence of noise.

The results for these experiments are shown in Figure 4.5. We see that now IPPE+HO significantly outperforms RPnP with respect to rotation and translation for all n . Furthermore, the performance of IPPE+HO is now very similar to GEOMREF in all test conditions, which is a remarkable result. The next best performing method is RPP-SP. With respect to rotation, RPP-SP is consistently

outperformed by IPPE+HO.

	E18	E19	E20	E21	E22	E23
f	800	800	800	800	800	800
w	100	100	100	100	100	100
n	4→10	4→10	4→10	4→10	4→10	4→10
σ_I	0.5	2	5	2	2	2
σ_M	0	0	0	0.5	1	3
Mode	1	1	1	1	1	1

Table 4.9: Varying imaging conditions in simulation experiments E18-E23.

4.4.5.3 Ambiguous cases

In the final set of simulation experiments we investigate algorithm performance in mode 2 (without excluding ambiguous cases). Here algorithms are permitted to return multiple solutions, and we compute error with respect to the closest solution to the ground truth. Ambiguous cases occur when the amount of perspective distortion is small, which can be controlled by reducing the plane’s size. We give the experimental parameters in in Table 4.10 using the same selection method as E18-E23 with at least four correspondences positioned on the corners of the plane. Here we have reduced the plane size to 50, which meant many ambiguous cases were included. The performance graphs are shown in Figure 4.6. Here we see a similar performance trend to E18-E23. IPPE+HO consistently does very well. It is the best performing method with respect to rotation (excluding GEOMREF) in all conditions, with a very small gap between IPPE+HO and GEOMREF. The performance gap for smaller n becomes smaller, and for $n = 4$ it is virtually indistinguishable. IPPE+HO performs as well as or better than RPP-SP in translation. HDZh and EPnP performs much worse than IPPE+HO, RPNP and RPP-SP, and their errors are beyond the axis range.

	E24	E25	E26	E27	E28	E29
f	800	800	800	800	800	800
w	50	50	50	50	50	50
n	4→10	4→10	8→40	8→10	4→10	4→10
σ_I	0.5	1	2	1	1	1
σ_M	0	0	0	0.5	1	2
mode	2	2	2	2	2	2

Table 4.10: Varying imaging conditions in simulation experiments E24-E29. These experiments tested algorithm performance in mode 2.

4.4. EXPERIMENTAL EVALUATION WITH SIMULATED DATA

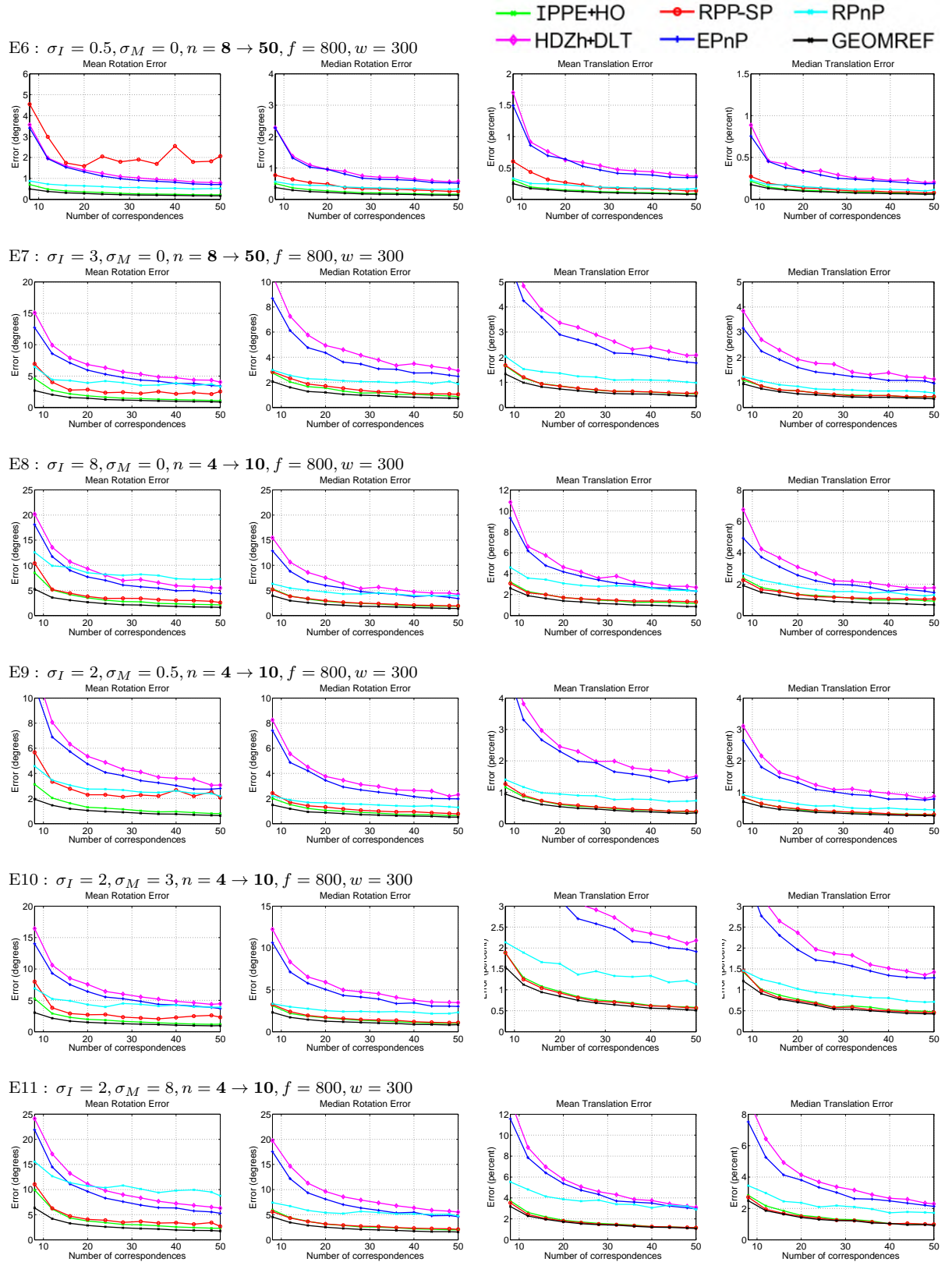


Figure 4.3: Comparing IPPE with previous state-of-the-art methods with simulation experiments E6-E11

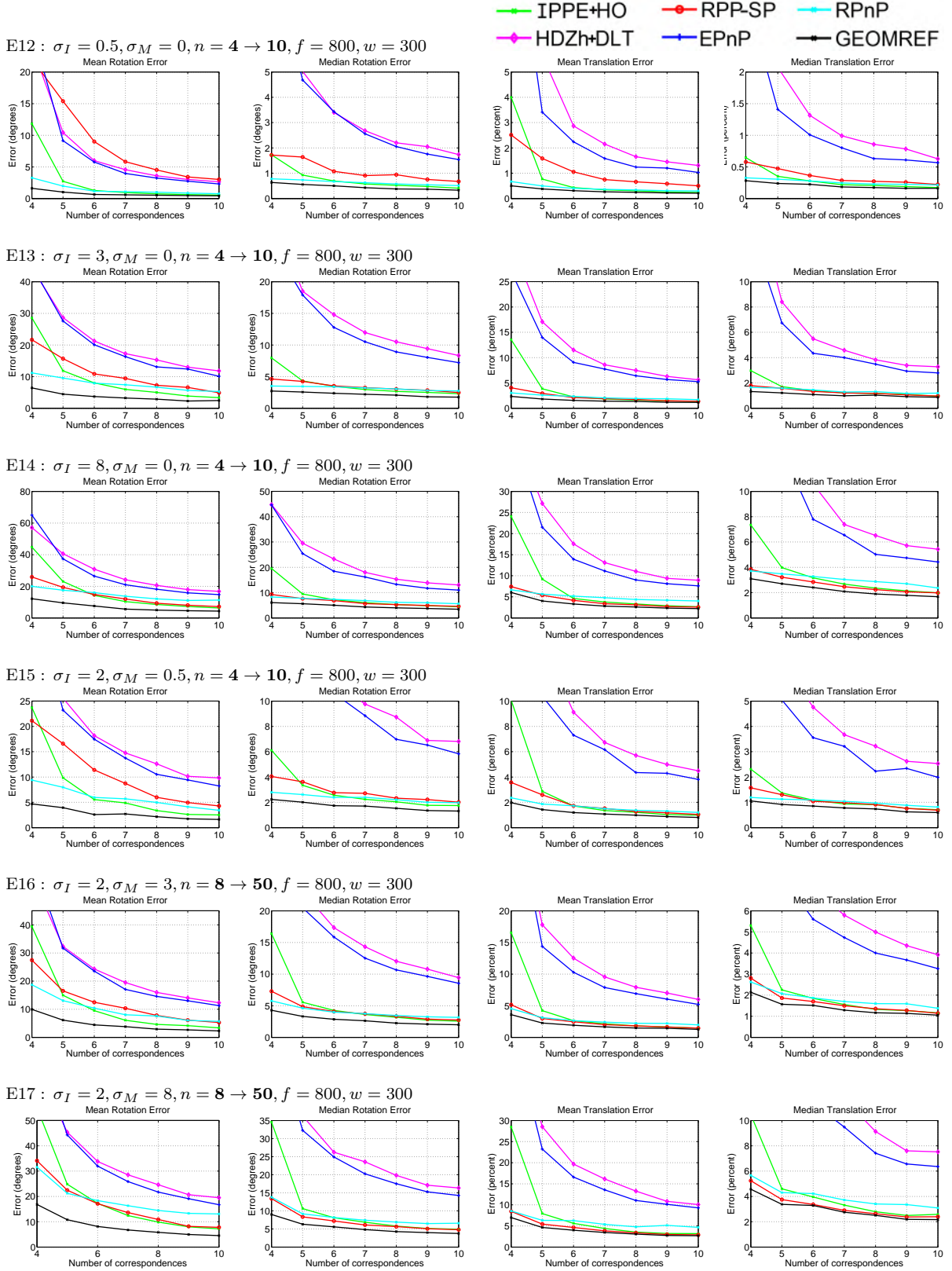


Figure 4.4: Comparing IPPE with previous state-of-the-art methods with simulation experiments E12-E17

4.4. EXPERIMENTAL EVALUATION WITH SIMULATED DATA

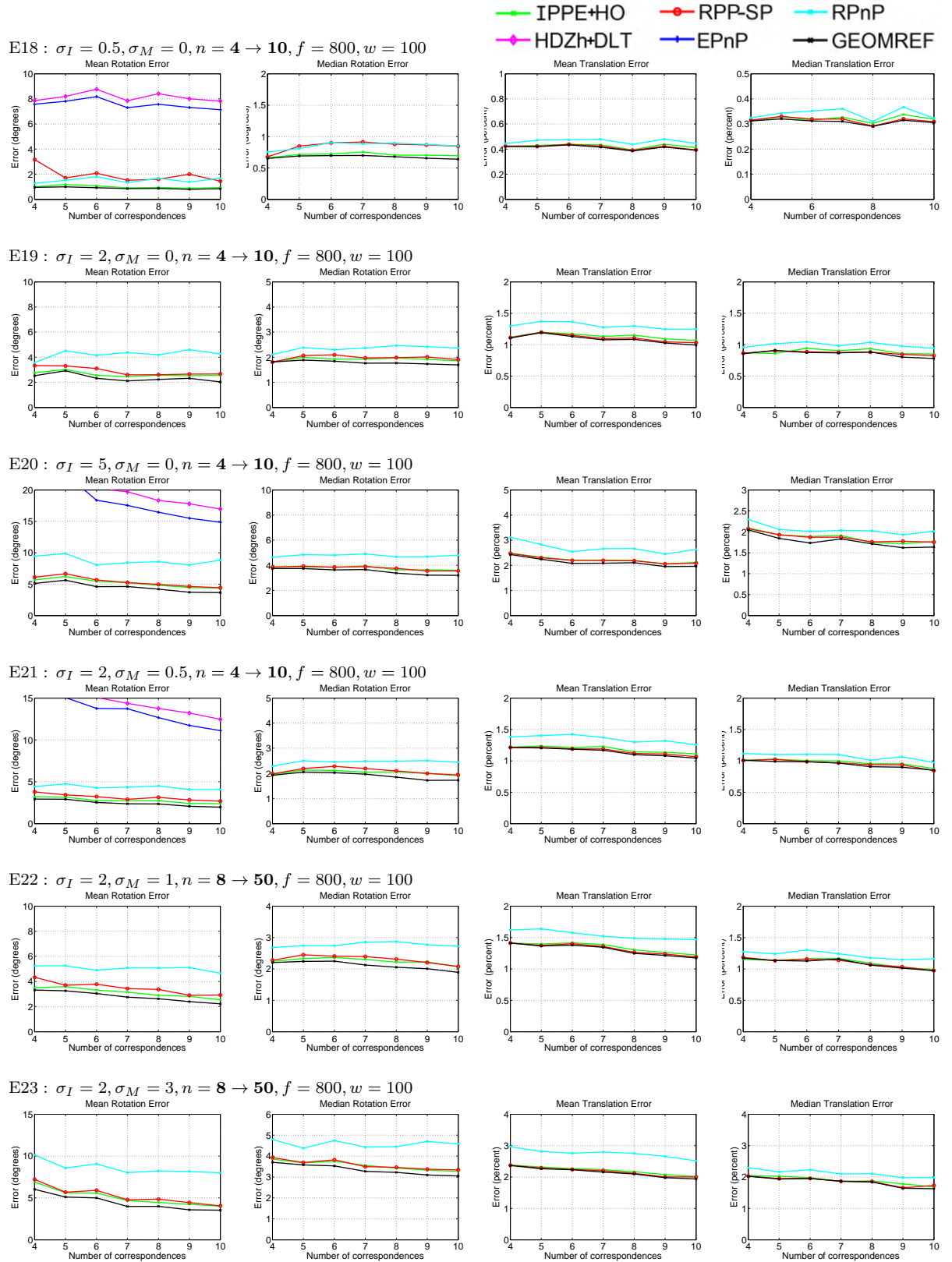


Figure 4.5: Comparing IPPE with previous state-of-the-art methods with simulation experiments E18-E23.

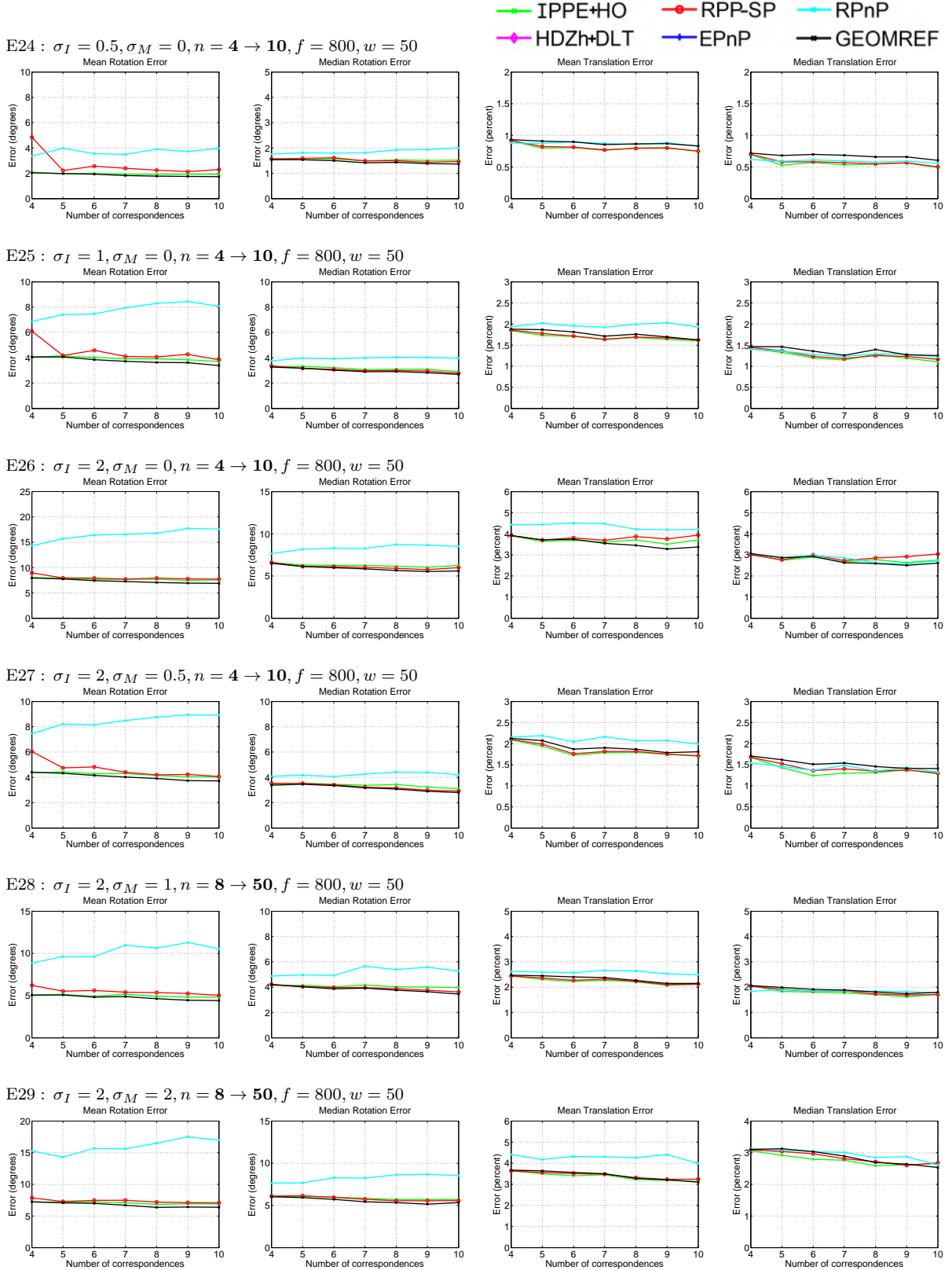


Figure 4.6: Comparing IPPE with previous state-of-the-art methods with simulation experiments E24-E28.

4.4. EXPERIMENTAL EVALUATION WITH SIMULATED DATA

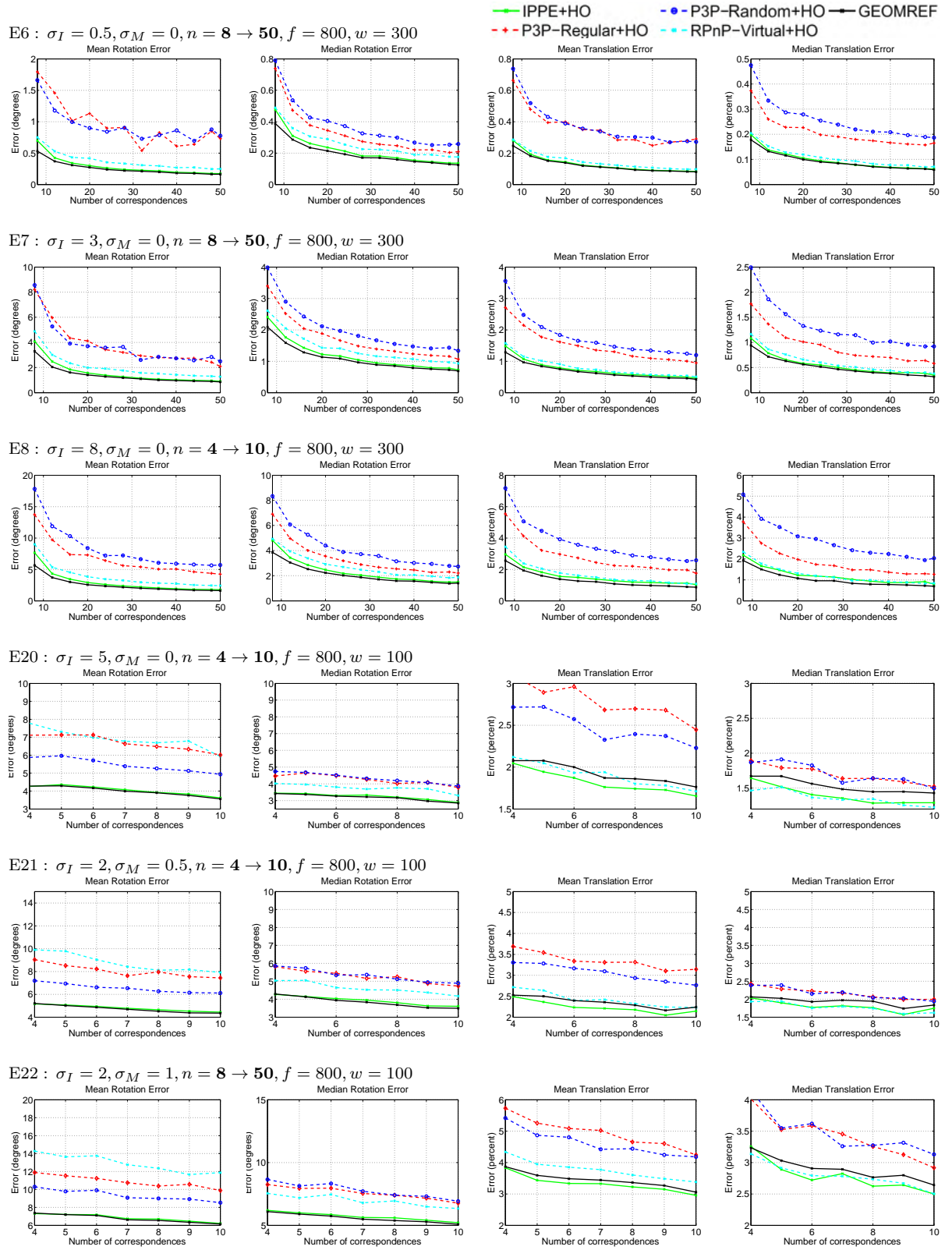


Figure 4.7: Synthetic experiments comparing IPPE with P3P using virtual point correspondences computed from the homographies.

4.4.6 IPPE Versus P3P/RPnP with virtual correspondences

In the final part of our simulation experiments we compare IPPE against P3P using virtual point correspondences. Specifically given $\hat{\mathbf{H}}$, virtual points are positioned on the object plane and their correspondences are computed by transforming them with $\hat{\mathbf{H}}$. We test three different choices for positioning the object points. These are as follows:

- **P3P-Random:** We compute the bounding box of $\{\mathbf{u}_i\}$ and position three points randomly within this box.
- **P3P-Regular:** We compute the bounding box of $\{\mathbf{u}_i\}$ and position three points on the bottom-left, top-left and top-right boundaries of this box.
- **RPnP-Virtual:** We use all the original object points.

Pose is solved for P3P-Random and P3P-Regular using the method in [Gao+03]. Pose for RPnP-Virtual is solved using RPnP (which splits the points into multiple P3P problems), but using their positions in the image predicted by $\hat{\mathbf{H}}$, rather than the measured correspondences. P3P-Random suffers from the problem that it may return zero solutions. We have found that this occurs in practice between 3-4% of the time depending on noise. To make the comparison simple we compute performance statistics for P3P-Random using only instances where it returned at least one solution. By contrast because the points in P3P-Regular are at right-angles, it is guaranteed to return at least one solution, and at most two [Gao+03].

To maintain a fair comparison we compared P3P-Random, P3P-Regular and RPnP-Virtual using the homography estimated using HO. We have found that IPPE+HO consistently performs better than P3P-Random, P3P-Regular and RPnP-Virtual across the experiments presented earlier in this section. For brevity we present the results for just for experiments E6-E8 and E18-E20 for these methods. This is given in Figure 4.7.

4.5 Experimental evaluation with real data

In this section we evaluate the algorithms on three applications involving real images. The first is to estimate the pose of a planar target from keypoint matches. The second is to estimate the pose of a planar checker-board target. The third is to estimate the pose of small planar AR markers.

4.5.1 Pose estimation from keypoint matches

In this experiment a series of images of a 120×90 mm planar test surface was photographed in normal indoor light conditions. The series comprises 28 images, three of which are shown in Figure 4.8. The camera was a Nikon D3100 DSLR with image resolution 2304×1536 pixels. The camera was calibrated with Bouguet’s calibration toolbox [Bou00] with focal length $f_x = 3204$ pixels, $f_y = 3220$ pixels. A fronto-parallel model image was constructed by undistorting and rectifying the first of these images. Correspondences between the rectified view and images were computed using VLFeat’s SIFT implementation [VF] and distinct matches were kept using Lowe’s ratio test [Low04b]. RANSAC was performed to find inlier correspondences (an inlier threshold of 5 pixels was used). This resulted in between 250-400 correspondences per image. We then computed gold standard pose estimates for each image using all inlier correspondences with GEOMREF. We compute error statistics over all 28 images and Table 4.11 shows mean RE and TE with respect to the gold standard. Given the relatively large

number of correspondences, all tested methods perform quite well. IPPE+HO obtained the lowest RE and TE.

	Mean RE (degrees)	Mean TE (%)
IPPE+HO	0.1249	0.0375
HDZh+DLT	2.8650	0.2691
RPP-SP	1.4951	1.0877
EPnP	1.9347	1.0496
RPnP	0.1850	0.2132

Table 4.11: Accuracy of algorithms on the ‘Game cover’ dataset using all correspondences. Accuracy is computed with respect to the gold standard poses from GEOMREF.

We then used this dataset to study the accuracy of the algorithms as conditions become more challenging. Specifically, when using smaller numbers of correspondences drawn from local regions of the surface. Conditions become harder as the region becomes smaller because (i) there are fewer correspondences and (ii) the problem becomes more ambiguous thanks to weaker perspective effects. Each image was processed as follows. For each correspondence, we collected all correspondences that were within a circular window of radius r mm. If there were fewer than 3 neighboring correspondences we discard the window. Otherwise we computed pose and measured pose error with respect to the gold standard pose (computed once using all correspondences across the surface). We varied r within the range [5.22...50] mm. The results are shown in Figure 4.9. In each graph we plot error on the x -axis and cumulative density on the y -axis. A point on a curve at (x, y) means that $y\%$ of all spatial windows in all images had an error below x . A better performing method has a curve with higher area-under-curve. The first and second rows of Figure 4.9 show error in rotation and translation respectively as the spatial window increases from 5.22 mm to 20.9 mm. We also give \bar{n} ; the average number of correspondences within each window. This varies from $\bar{n} = 4.64$ to $\bar{n} = 23.8$. The third and fourth rows of Figure 4.9 show errors for larger windows; r varying from 26.1mm to 41.1mm. For the smallest window size $r = 5.22$ mm all methods perform relatively poorly, including GEOMREF. This is because at this small scale the problem is severely weakly-posed and the influence of noise is very large. As r increases all methods perform better. Among the closed-form methods (IPPE+HO, HDZh+DLT, ePnP and RPnP), IPPE achieves the best results in general. The exception is the smallest neighborhood size ($r = 5.22$ mm, $\bar{n} = 4.57$) where it is slightly out-performed by RPnP. This agrees with the simulated results, where it was shown that for randomly positioned object points, RPnP does better than IPPE for $n < 5$. The performance of RPP is relatively good, however recall that unlike IPPE it is not a closed-form algorithm. The performance of HDZh+DLT is significantly worse than IPPE+HO, agreeing with the simulation experiments.



Figure 4.8: Images taken from the ‘Game cover’ dataset. Images were captured with a Nikon D3100 DSLR with image resolution 2304×1536 pixels. We used SIFT [Low04b] to compute putative feature matches with each image containing between 200-500 features.

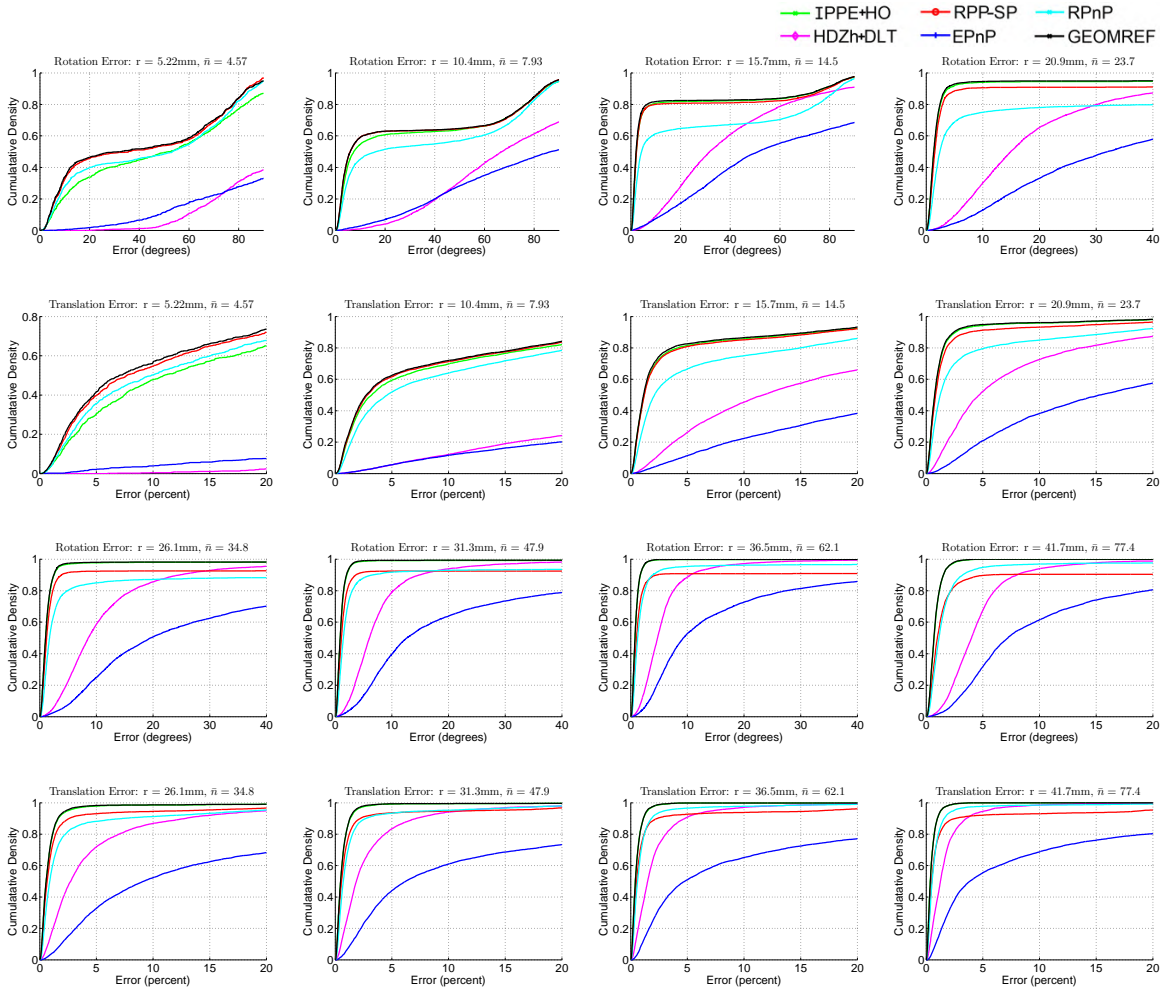


Figure 4.9: Experimental results with real data using the ‘Game cover’ dataset.

4.5.2 Pose estimation of checkerboards

Our second set of real experiments involves estimating the pose of a planar checkerboard pattern. We have experimented with two datasets. The first is a series of 20 images captured by a standard 720p smartphone camera in normal indoor lighting conditions. We used a checker surface comprising 21×30 squares each of size 9.22mm. Figure 4.10 shows three example images in this dataset. The second dataset is a publicly-available one from the Matlab Calibration Toolbox. This comprises 20 images of a 12×12 checkerboard with a square size of 30 mm.

We computed point correspondences at checker corners using the Matlab Calibration Toolbox with iterative sub-pixel accuracy refinement. There were 628 correspondences per image for the the first dataset. All methods perform well using this amount of data. To differentiate the methods we perform a similar experiment to §4.5.1 to see how well they performed with smaller checkerboards. For each image, we draw all $m \times m$ sub-checkerboards, where m was varied from 2 to 21. We then computed pose for each sub-checkerboard and compared to the gold standard pose (computed by GEOMREF using all points). Figure 4.11 shows the results for the first dataset. Here we see that IPPE+HO is virtually indistinguishable from GEOMREF, which is a very strong result and it agrees with our simulated data results where we found IPPE performs exceptionally well when the object points are arranged in

4.5. EXPERIMENTAL EVALUATION WITH REAL DATA

a regular pattern. RPP-SP is also virtually indistinguishable from GEOMREF, however this is not a closed-form method and it is between 50 and 70 times slower than IPPE+HO. HDZh+DLT and EPnP perform significantly worse than IPPE, agreeing with all previous experiments. RPnP performs better for larger checkerboard regions but it never beats IPPE. We performed a similar with checkerboard images from the Matlab Calibration Toolbox dataset, and we found the same trends, shown in Figure 4.12.

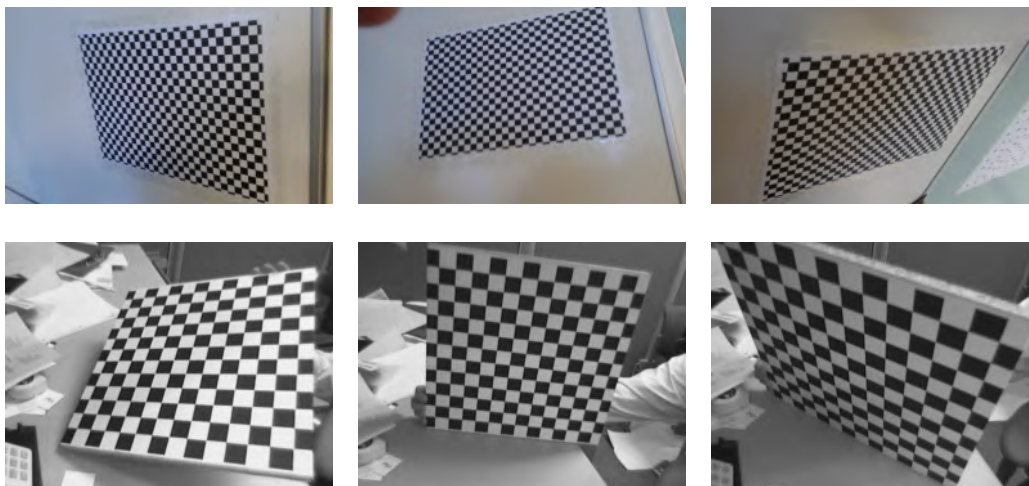


Figure 4.10: Example views of two checkerboard test surfaces. Top row: views of a 193×276 mm target captured by a 720p smartphone. Bottom row: views of a 360×360 mm target from the public dataset supplied with the Matlab Calibration Toolbox. The performance of IPPE+HO and RPP-SP is virtually indistinguishable to GEOMREF.

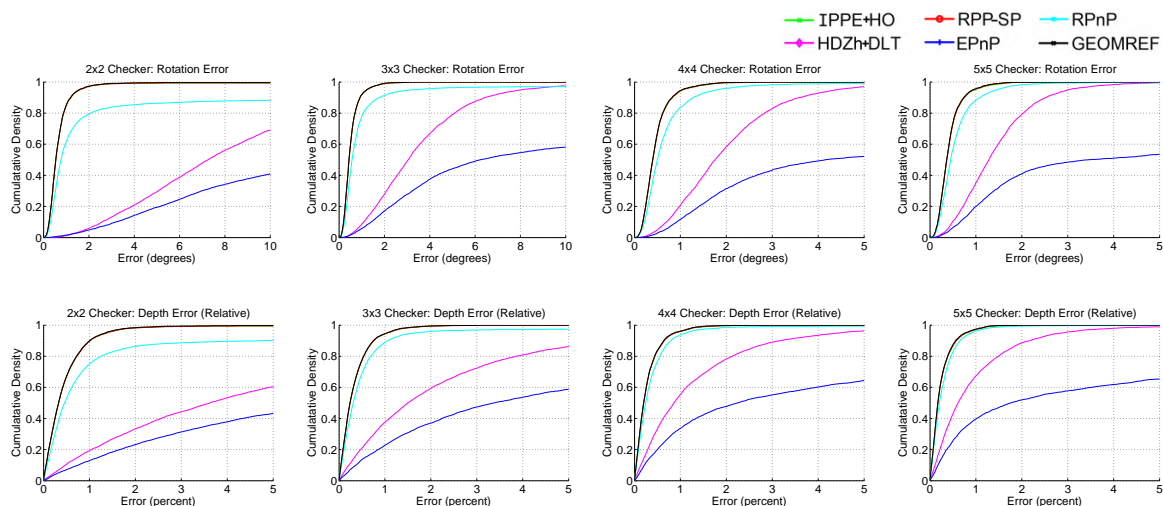


Figure 4.11: Real experiments (checkerboard pose estimation captured with 720p smartphone): Comparing pose accuracy with varying checker sizes. The performance of IPPE+HO and RPP-SP is virtually indistinguishable to GEOMREF.

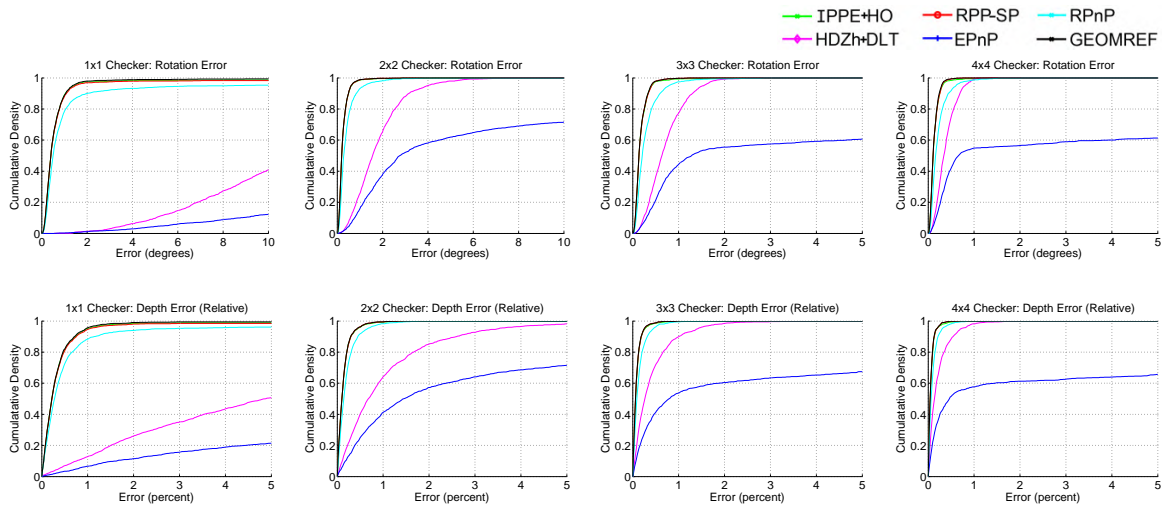


Figure 4.12: Real experiments (checkerboard pose estimation with data from the Matlab Calibration Toolbox): Comparing pose accuracy with varying checker sizes.

4.5.3 Pose estimation of Augmented Reality markers

In the last set of experiments we evaluated performance for estimating the pose of AR markers. This task typically involves the following processing pipeline: *(i)* detecting the approximate positions of markers in an image as regions that match a marker’s characteristic pattern. *(ii)* refining the four corner positions of each marker to sub-pixel accuracy. *(iii)* using the corners to estimate the 3D pose of each marker. *(iv)* (optional) pose refinement with Gauss-Newton or Levenberg-Marquardt. Here we compare the accuracy of IPPE to previous methods for solving *(iii)*. Because this problem involves 4 points, we estimate the homography exactly from point correspondences analytically.

We used the following experimental setup. The open source library ArUco [Gar+16b] was used to generate 300 uniquely-identifiable markers of width 7.90 mm. The markers were rotated by a random angle and distributed evenly over 9 A4 sheets of paper. These sheets were printed using a high-precision laser printer and corrected for non-uniform printer scaling. The papers were then fixed to a large planar background surface, by tiling them in a 3×3 grid. We ran plane-based bundle-adjustment to accurately determine the relative positions of each sheet of paper on the background. This allowed us to have a composite planar model of all 300 AR markers.

We then captured two video sequences with a 720p smartphone camera. The first one viewed the markers at close range, with an average distance between camera and markers of 52.1 cm. The second was at mid-range with the an average distance of 102.2 cm. We ran ArUco’s marker detector and rejected any video frames where fewer than 10 markers were detected (typically occurring when high motion blur was present). From the remaining frames we randomly selected 30 frames from both videos to build two test sets. Example frames from the close and medium-range datasets are shown in the top and bottom rows of Figure 4.13 respectively. We then tested the performance of the algorithms. For each image in a dataset, a gold-standard pose was computed using GEOMREF with *all* detected AR markers. The performance of an algorithm was measured by how close its could estimate the gold standard pose using a *single* AR marker. This was done with every marker in every image of a dataset and we plot the results in Figure 4.14. We compute rotation error, and also the error of the estimated depth of the center of the AR marker (in mm). Here we see that IPPE, RPnP

and RPP-SP are the best performing methods, with IPPE and RPP-SP virtually indistinguishable to GEOMREF. RPnP performs slightly worse in rotation error compared to IPPE and RPP-SP. HDZh and EPnP are significantly worse at solving this problem, and show a significant performance drop for the mid-range dataset. Even though the accuracy of IPPE, RPnP and RPP-SP is quite similar, IPPE is significantly faster because the homography is computed analytically. Therefore, IPPE computes pose *entirely analytically* using only simple floating point operations for this problem. This is not true of RPP-SP and RPnP. Note that there is a noticeable tail in errors: approximately 5% of markers have errors greater than 10 degrees with the best methods. The reason is because of tracking errors; very occasionally the corner predictions are far from their true positions (*e.g.* greater than 5 pixels), causing gradient-based corner refinement to be trapped in an incorrect local minimum.

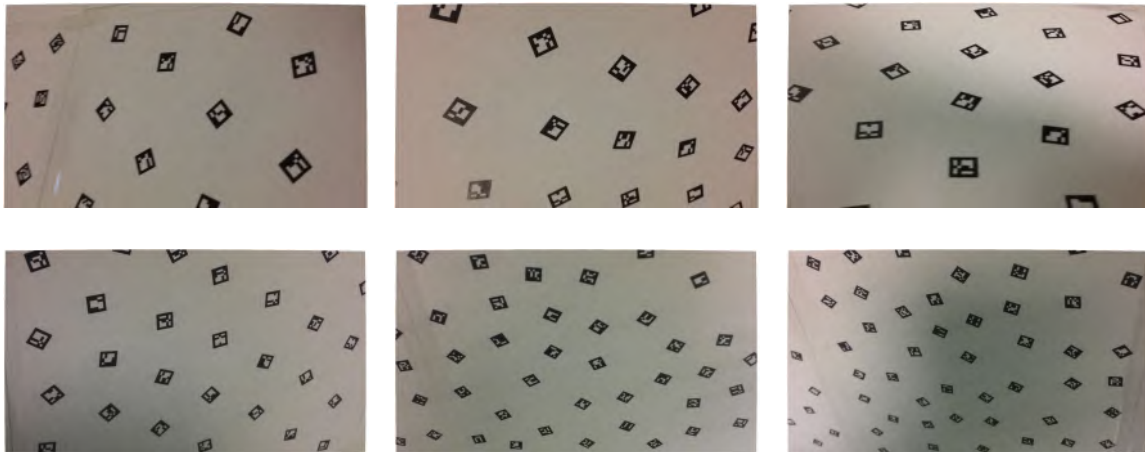


Figure 4.13: Example views of AR markers captured by a 720p smartphone. Top row: close-range views. Bottom row: medium-range views.

4.5.3.1 Timing information

We compared computation time of the methods as a function of n using Matlab implementations on a standard Intel i7-3820 desktop PC running 64-bit Matlab 2012a. For all compared algorithms we use the code provided by the authors. We use our own Matlab implementation of IPPE. Note that these are not the fastest implementations, and large speedups would be gained with C or C++ implementations. However, benchmarking all methods with Matlab gives a fair comparison and reveals how computation time scales with n . For a given n we simulated 500 randomized configurations using the simulation setup in §4.4.1. Figure 4.15 and Table 4.12 shows processing time as n varies from 4 to 650. For IPPE and HDZh with $n = 4$, we use an analytic formula to estimate the homography. RPP-SP is by far the slowest method. IPPE+HO is the fastest method. It is marginally faster than HDZh+DLT and considerably faster than EPnP, RPnP and RPP-SP. In Table 4.12 at $n = 4$ we see that IPPE is approximately 6.7 times faster than EPnP and 6.2 times faster than RPnP. IPPE is approximately 75 times faster than RPP-SP. EPnP, RPnP and IPPE are all $O(n)$ methods. We can see from Figure 4.15 that the graph's slope is considerably lower for IPPE than for EPnP and RPnP. This is because IPPE is time-bounded by the cost of computing the homography, which itself is very fast even for large n . At $n = 500$, IPPE is only about 1.5 times slower than at $n = 6$. By contrast EPnP and RPnP are approximately 4.3 and 9.6 times slower at $n = 500$ than $n = 6$.

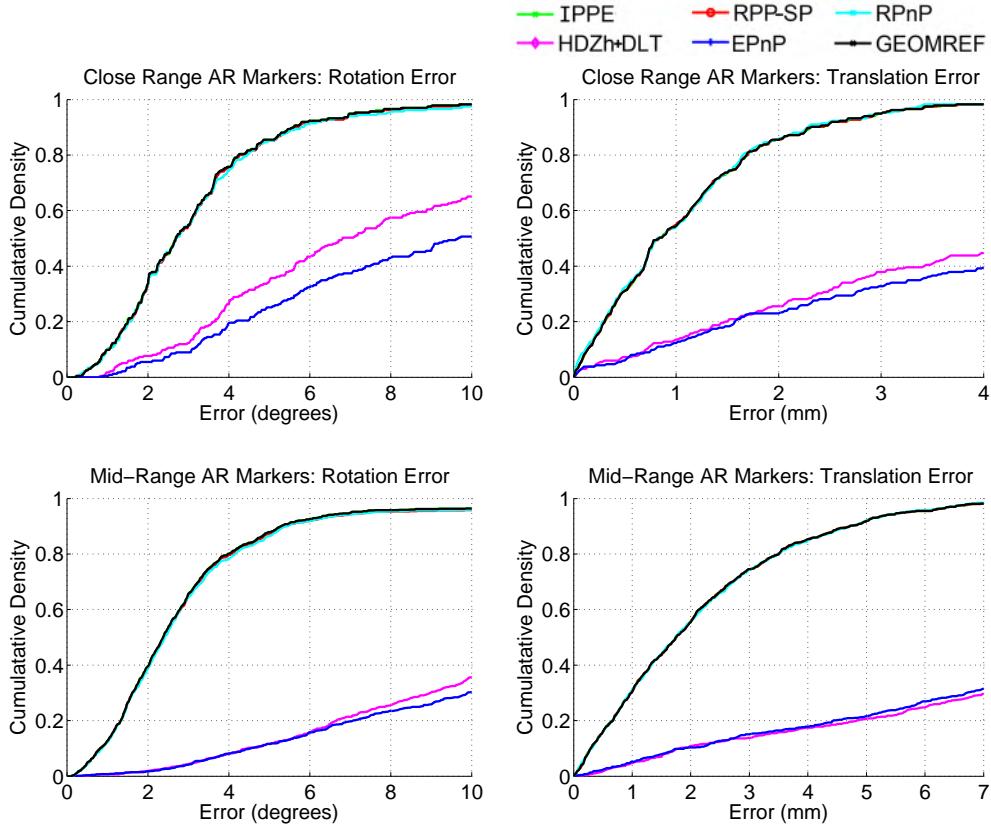


Figure 4.14: Real experiments (AR marker pose estimation). Results are divided into close-range (left column) and mid-range (right column) conditions. Because each marker has four point correspondences at its four corners, the homography is computed exactly without requiring HO or DLT methods. There is very little to distinguish IPPE, RPP-SP and RPnP in terms of accuracy, and all perform very similarly to GEOMREF. This indicates that for this application there is no real benefit in refining their pose estimates with maximum likelihood refinement. However IPPE is by far the fastest and simplest of these three methods (see Table 4.12 with $n = 4$).

n	IPPE+HO	HDZh+DLT	RPP-SP	EPnP	RPnP
4	0.150	0.261	11.101	1.012	0.940
6	0.387	0.497	14.211	0.883	0.965
10	0.398	0.517	22.444	0.929	1.011
60	0.420	0.527	51.260	1.024	1.475
160	0.494	0.605	138.508	1.705	2.908
340	0.555	0.669	258.100	2.853	5.659
500	0.602	0.715	408.362	3.760	9.205
700	0.657	0.771	483.992	4.849	13.905

Table 4.12: Computation time (in ms) of compared methods with different number of points. Benchmarking was performed on a standard Intel i7-3820 desktop PC. We used Matlab implementations provided by the authors for RPP-SP, EPnP, RPnP and HO. We used our own Matlab implementation for IPPE, HDZh and DLT.

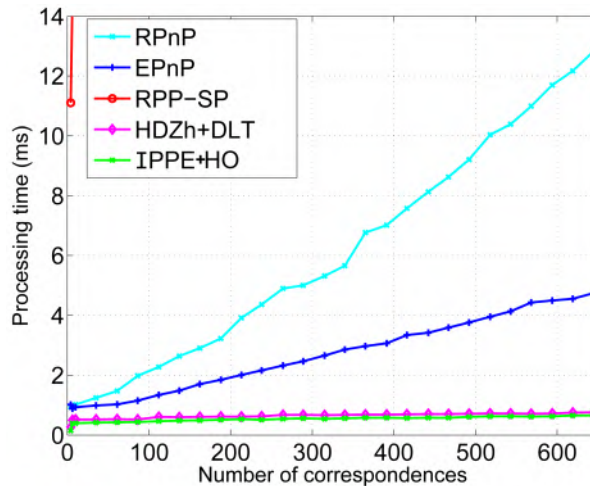


Figure 4.15: Computation time (in ms) of compared methods with different number of points. Benchmarking was performed on a standard Intel i7-3820 desktop PC. We used Matlab implementations provided by the authors for RPP-SP, EPnP, RPnP and HO. We used our own Matlab implementation for IPPE, HDZh and DLT.

4.6 Conclusion

We have presented the Infinitesimal Plane-based Pose Estimation (IPPE) algorithm that estimates the pose of a planar object with a perspective camera from a noisy homography matrix. Pose estimation is formulated as the solution to a 1st-order PDE that is solved exactly and analytically with two solutions in general. The PDE uses a differentiation point defined on the object plane, and by moving the differentiation point we modify the PDE’s errors-in-variables. The differentiation point that we use in practice approximately reduces the errors-in-variables, and it is the centroid of the object points. Extensive experimental evaluation shows that IPPE obtains better pose accuracy compared to previous state-of-the-art real-time and closed-form PnP methods in most conditions. IPPE is substantially faster than those methods and in the special case of 4 points, the homography has an analytical solution, so IPPE is computed entirely analytically from the point correspondences with only simple floating-point operations.

IPPE dramatically improves pose estimation accuracy compared to previous Perspective Homography Decomposition methods, yet is just as fast. This is a remarkable result considering it differs only by re-weighting PHD with a rank-2 weight matrix (Theorem ??). Nevertheless, we have shown this reweighting substantially changes the problem, the number of solutions and it provides solution stability in quasi-affine conditions. Furthermore, we have shown that when the points are arranged regularly (*e.g.* the corners of a square which is typical for AR marker pose estimation), IPPE is extremely accurate with little difference compared to poses estimated by iterative optimization of the L_2 reprojection error (known to give the best pose estimates in practice with mild noise assumptions). This is certainly not the case for previous PHD methods. Unlike those methods, IPPE does not break down when the perspective effects reduce. IPPE also has some attractive theoretical properties: our algorithmic solution never introduces artificial degeneracies and it gives two pose solutions with a simple geometric relationship that captures the affine pose ambiguity. For these reasons, IPPE is arguably the best method for plane-based pose estimation with regularly distributed object points,

and it is arguably the best method to initialize iterative pose refinement in other conditions using *e.g.* Gauss-Newton or Levenberg-Marquardt.

Chapter 5

Focal Length and Shape-from-Template

Chapter summary and organization

This chapter presents the first solutions to focal length and Shape-from-Template (fSfT) using single images (single-view fSfT) and multiple images (multi-view fSfT). In §1.3.3 of the introduction chapter of this thesis we have provided the problem background, motivation and applications. In §1.3.3.2, we have also provided an overview of this chapter's technical and theoretical contributions. This chapter is organized into five main sections. In §5.1 we present our analytical method for solving single-view fSfT. In §5.2 we present our optimization-based method for solving single-view fSfT, which can be initialized using either the analytical method or a novel focal length sampling approach. In §5.3 we present our solutions to multi-view fSfT with a common unknown focal length. In §5.4 we present our experimental evaluation on a range of public datasets. In §5.5 we present our conclusions. We discuss future research directions following this work in §6.2.3.2 of Chapter 6.

5.1 Solving fSfT analytically

5.1.1 Section overview

This section presents our analytical method for solving fSfT. We first present our formulation of the problem as a PDE in §5.1.2. We then instantiate it with the pinhole camera in §5.1.3. We then solve an approximation of the PDE using the weak-perspective camera in §5.1.4. From this solution we recover the focal length analytically and uniquely in §5.1.5. We provide implementation details in §5.1.6 and we provide a simple method to estimate focal length robustly using multiple PDE solutions. Finally, in §5.1.7 we analyze fSfT well-posedness and degeneracies using the analytical solution.

5.1.2 PDE formulation

We illustrate our geometric modeling of fSfT in Figure 5.1. This is closely inspired by the modeling used for solving SfT analytically with a calibrated perspective camera [Bar+15]. The template is represented by a known topological surface \mathcal{R} defined in object coordinates. The deformed template is represented by an unknown topological surface \mathcal{S} embedded in camera coordinates. We denote as $\Psi : \mathcal{R} \rightarrow \mathcal{S}$ the transformation of \mathcal{R} to \mathcal{S} . For the analytical method, we assume that Ψ is isometric, therefore Ψ does not stretch or shrink \mathcal{S} . We define as $\Omega \in \mathbb{R}^2$ the known parameterization space of the template (holding a flattened 2D version of the template). The known function $\zeta : \Omega \rightarrow \mathcal{R}$ is bijective and transforms Ω to \mathcal{R} , and the unknown function ϕ is bijective and transforms Ω to \mathcal{S} . Because Ψ is isometric, the first fundamental form of \mathcal{R} and \mathcal{S} is the same. This is satisfied if and only if $\nabla\phi^\top\nabla\phi = \nabla\zeta^\top\nabla\zeta$. Camera projection is represented by the unknown function $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. The function $\eta : \Omega \rightarrow \mathbb{R}^2$ is a known transformation from Ω to the image \mathcal{I} . In practice η can be computed by fitting a continuous 2D warp function between Ω and \mathcal{I} using point correspondences. We discuss the implementation details in §5.1.6.

Our objective is to determine ϕ and π from ζ , η , and deformation constraints (isometry) acting on ϕ and ζ . We achieve this by relating these functions to 1st-order with the following PDE:

$$\text{find}_{\phi, \pi} \begin{cases} \pi \circ \phi = \eta & (a) : 0^{th} \text{ - order reprojection} \\ (\nabla\pi \circ \phi) \nabla\phi = \nabla\eta & (b) : 1^{st} \text{ - order reprojection} \\ \nabla\phi^\top \nabla\phi = \nabla\zeta^\top \nabla\zeta & (c) : \text{isometry} \end{cases} \quad (5.1)$$

The right sides of Problem (5.1) are known. The 0th order reprojection constraints have 2 equations (one for each 2D spatial dimension), the 1st order reprojection constraints have 4 equations, and the isometry constraints have 4 equations (three of which are independent).

5.1.3 Instantiation with the pinhole model

We assume π is a perspective projection where the only unknown intrinsic is the focal length f . The other intrinsics (principal point, skew, aspect ratio and lens distortion) are assumed to either be known or to take default values. For many real-world cameras we can assume no lens distortion, principal point at the image center, no skew and an aspect ratio of 1. The effects of the intrinsics other than focal length can be ‘undone’ by applying a corresponding image warp as a preprocessing step. The camera projection function then reduces to the pinhole model:

$$\pi(x, y, z) = \frac{f}{z}(x, y) \quad (5.2)$$

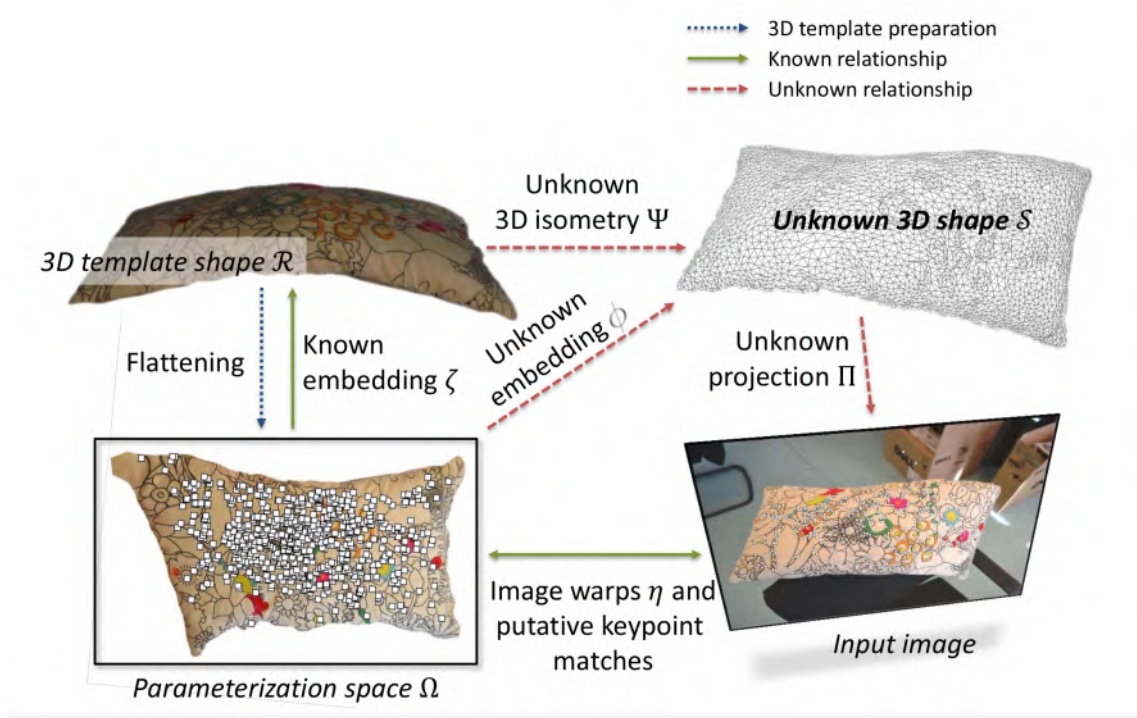


Figure 5.1: fSfT problem modeling used by the analytical method.

We substitute Equation (5.2) into Problem (5.1), giving a 1^{st} -order PDE specific to the pinhole model:

$$\text{find}_{\phi, \pi} \begin{cases} \frac{f}{\phi_z} \text{stk}(\phi_x, \phi_y) = \eta & (a) : 0^{th}\text{-order pinhole reprojection} \\ \frac{f}{\phi_z} \begin{bmatrix} 1 & 0 & -\frac{\phi_x}{\phi_z} \\ 0 & 1 & -\frac{\phi_y}{\phi_z} \end{bmatrix} \nabla \phi = \nabla \eta & (b) : 1^{st}\text{-order pinhole reprojection} \\ \nabla \phi^\top \nabla \phi = \nabla \zeta^\top \nabla \zeta & (c) : \text{isometry} \end{cases} \quad (5.3)$$

where $\text{stk}(\phi_x, \phi_y, \phi_z) \stackrel{\text{def}}{=} \phi$.

5.1.4 Local weak-perspective solution

We now solve Problem (5.3) by approximating the pinhole camera with a local weak-perspective (LWP) camera, leading to a simple closed-form solution. We call this local because the approximation is specific to a particular point on the surface. This contrasts the typical use of weak-perspective, where the projection of an entire surface is approximated by a single weak-perspective projection. This is a key factor to allow us to recover focal length.

The weak-perspective approximation of $\nabla \pi$ is equivalent to linearizing the pinhole camera about the optical center, and is as follows:

$$\nabla \pi(x, y, z) \approx \frac{f}{z} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.4)$$

Therefore we approximate Problem (5.3) with a LWP camera as follows:

$$\text{find}_{\phi, \pi} \begin{cases} \frac{f}{\phi_z} \text{stk}(\phi_x, \phi_y) = \eta & (a) : 0^{th} - \text{order pinhole reprojection} \\ \frac{f}{\phi_z} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \nabla \phi = \nabla \eta & (b) : 1^{st} - \text{order LWP reprojection} \\ \nabla \phi^\top \nabla \phi = \nabla \zeta^\top \nabla \zeta & (c) : \text{isometry} \end{cases} \quad (5.5)$$

We now solve Problem (5.5) by a relaxation: we consider ϕ and $\nabla \phi$ as independent variables. This allows us to solve the system in closed-form with a small-scale algebraic problem for any given surface point $\mathbf{p} \in \Omega$. Considering Equations (5.5-a) and (5.5-b) and with the following definitions:

$$\begin{aligned} \mathbf{A} &\stackrel{\text{def}}{=} \nabla^\top \zeta \nabla \zeta(\mathbf{p}), \mathbf{A} \in \mathcal{S}_{++}^2 && \text{(known)} \\ \mathbf{M} &\stackrel{\text{def}}{=} \nabla^\top \eta \nabla \eta(\mathbf{p}), \mathbf{A} \in \mathcal{S}_+^2 && \text{(known)} \\ \mathbf{X} &\stackrel{\text{def}}{=} \nabla \phi(\mathbf{p}), \mathbf{X} \in \mathbb{R}^{3 \times 2} && \text{(unknown)} \\ z &\stackrel{\text{def}}{=} \phi_z(\mathbf{p}), z \in \mathbb{R}^+ && \text{(unknown)} \end{aligned} \quad (5.6)$$

we have the following problem:

$$\text{find}_{f \in \mathbb{R}^+, z \in \mathbb{R}^+, \mathbf{X} \in \mathbb{R}^{3 \times 2}} \begin{cases} \frac{f}{z} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{X} = \mathbf{M} & (a) : 1^{st} \text{ order reprojection} \\ \mathbf{X}^\top \mathbf{X} = \mathbf{A} & (b) : \text{isometry} \end{cases} \quad (5.7)$$

We recognize that with rearrangement and change-of-variables, Problem (5.7) has an identical structure as the IPPE problem solved in Chapter 3. First we decompose \mathbf{A} with the Cholesky decomposition: $\mathbf{A} = \mathbf{C}\mathbf{C}^\top$, where \mathbf{C} is full-rank because \mathbf{A} is positive definite. Next we left and right-multiply Equation (5.7-b) by \mathbf{C}^{-1} and $\mathbf{C}^{-\top}$ respectively, giving $(\mathbf{X}\mathbf{C}^{-\top})(\mathbf{X}\mathbf{C}^{-\top}) = \mathbf{I}_2$. We then define variables $\mathbf{Y} \stackrel{\text{def}}{=} \mathbf{X}\mathbf{C}^{-\top}$ and $a \stackrel{\text{def}}{=} \frac{f}{z}$. We then right-multiply Equation (5.7-a) by $\mathbf{C}^{-\top}$ gives the following problem that is equivalent to Problem (5.7):

$$\text{find}_{a \in \mathbb{R}^+, \mathbf{Y} \in \mathbb{R}^{3 \times 2}} \begin{cases} a \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{Y} = \mathbf{M}\mathbf{C}^{-\top} & (a) : 1^{st} \text{ order reprojection} \\ \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_2 & (b) : \text{isometry} \end{cases} \quad (5.8)$$

This change of variables is equivalent to applying a local linear distortion to the surface in parameterization space so that its metric tensor is \mathbf{I}_2 . The problem now resembles plane-based pose estimation with the weak-perspective camera. The solution to this is already known: it is equivalent to Problem (4.12) on page 78 with the substitutions $\mathbf{J} \leftarrow \mathbf{M}\mathbf{C}^{-\top}$, $\gamma \leftarrow a$ and $[\mathbf{R}]_{3 \times 2} \leftarrow \mathbf{Y}$. This is solved in closed-form using Algorithm 6 with a unique solution for a and a two-fold solution for \mathbf{Y}

5.1.5 Focal length solution

The variable a can be evaluated at any point $\mathbf{p} \in \Omega$ whose motion gradient can be measured. We use $\alpha : \Omega \rightarrow \mathbb{R}^+$ to denote a spatially varying function that gives a at each such point. By definition α gives the ratio between focal length and surface depth. We now use α to solve f . From the definition of α and the pinhole camera in Equation (5.2) we have

$$\phi = \frac{1}{\alpha} \begin{pmatrix} \eta \\ f \end{pmatrix} \quad (5.9)$$

The 1st-order differentiation of Equation (5.9) gives

$$\nabla\phi = - \begin{pmatrix} \eta \\ f \end{pmatrix} \frac{\nabla\alpha}{\alpha^2} + \frac{1}{\alpha} \begin{pmatrix} \nabla\eta \\ \mathbf{0}^\top \end{pmatrix} \quad (5.10)$$

We then substitute Equation (5.10) into Equation (5.1 c) giving the following quadratic system in f :

$$f^2 G(\nabla\alpha) = \alpha^4 G(\nabla\zeta) - \|\eta\|_2^2 G(\nabla\alpha) - \alpha^2 G(\nabla\eta) - \alpha (\nabla\alpha^\top \eta^\top \nabla\eta + \nabla\eta^\top \eta \nabla\alpha) \quad (5.11)$$

We reduce this to a single equation by pre- and post-multiplying Equation (5.11) by $\nabla\alpha$ and $\nabla\alpha^\top$ respectively. After rearranging we obtain a unique analytical expression for f

$$f = \frac{\alpha}{\|\nabla\alpha\|_2} \sqrt{\nabla\alpha (\alpha^2 G(\nabla\zeta) - G(\nabla\eta)) \nabla\alpha^\top - \eta^\top \nabla\eta \nabla\alpha^\top + \nabla\alpha \nabla\eta^\top \eta - \|\eta\|_2^2} \quad (5.12)$$

Thus, we have shown how to solve focal length uniquely in fSfT using a local analytical solution derived from 1st-order differential geometry. We have solved focal length using a LWP approximation. Consequently, the solution has some approximation error that varies spatially. It is most accurate at points for which LWP is a good approximation of pinhole projection. These are points that project closer to the camera's optical center or those where the surface gradient in camera coordinates is smaller. We also point out that α has a 1st-order dependency on η , thus $\nabla\eta$ and therefore f is 2nd-order in η .

5.1.6 Implementation details

We can solve f with Equation (5.12) at any point on the surface for which α and $\nabla\alpha$ can be measured, and where $\|\nabla\alpha\|_2 \neq 0$ (a degenerate configuration, see §5.1.7 below). To reduce noise sensitivity, we solve f using multiple points and robust averaging.

5.1.6.1 Robust averaging

We describe a simple approach using N point correspondences computed between the template and the image. We denote these by the ordered sets $\mathcal{P} \in \mathcal{R}^N$ and $\mathcal{Q} \in \mathbb{R}^{2N}$ respectively. Each member $i \in [1, N]$ of \mathcal{P} and \mathcal{Q} corresponds to the same physical surface point up to noise.

For each point $i \in [1, N]$, we estimate f using neighboring points defined within a local radius r of $\mathcal{P}(i)$. We perform this multiple times with K different radii (a default of $k = 10$ levels spaced uniformly between $r = 5\%$ and $r = 50\%$ of the template's size). The radius size trades-off warp estimation stability (increasing as r increases) and deformation simplicity (decreasing as r increases). The best trade-off is not known in advance, explaining our use of multiple radii.

We denote the neighborhood of the i^{th} point with radius level k by the set $\mathcal{N}_{i,k} \subset [1, N]$. For each $\mathcal{N}_{i,k}$ we estimate η and ζ by fitting a local warp surrounding point i with neighbors $\mathcal{N}_{i,k}$. We denote these as $\eta_{i,k}$ and $\zeta_{i,k}$ where we index over points and neighborhood sizes. Following [PB12], the warp is estimated using a low-complexity $L2$ regularized TPS with 9 control centers. Next we evaluate α using our solution to Equation (5.8), for each point i with spatial neighborhood index k that we denote as $\alpha_{i,k} \in \mathbb{R}^+$. We then compute $\nabla\alpha_{i,k}$ from $\alpha_{i,k}$ by fitting another low-complexity TPS to $\alpha_{l \in \mathcal{N}_{i,j}}$, then we differentiate the warp to first order. Finally, using Equation (5.12) we compute $N \times K$ focal length candidates, with one candidate per point and radius level. From these candidates, we find a single focal length \hat{f} that is compatible with as many candidates as possible. This is implemented

by a sweep over focal lengths, starting at the minimal candidate f_{\min} and finishing at the maximal candidate f_{\max} . We increment the swept focal length f' by a step size $s = \frac{1}{100} (f_{\max} - f_{\min})$. During the sweep we measure consensus by counting the number of focal length candidates that are close to f' (we use as a default threshold within 15% of f'). After the sweep, we take the largest consensus set and we compute \hat{f} as the median focal length in the largest consensus set.

5.1.6.2 Obtaining point correspondences and handling mismatches

Point correspondences can be computed with generic methods such as keypoint matching using classical methods such as SIFT [Low04a], learning-based methods such as LIFT [Yi+16] or with dense matching such as large displacement optical flow with classical or learning-based methods [BBM09]. Our approach is not tied to a specific method, and we give precise details for the methods used to generate point correspondences in the experimental section. Some methods generate mismatches, which are point correspondences that do not physically correspond to the same surface point up to noise. Mismatches cause problems because they can lead to poorly estimated local warps, leading to a poorly estimate focal length. The method in §5.1.6.1 has built-in robustness to some mismatches thanks to the fact that we combine local estimates with a robust estimation process. Specifically, mismatches outside of a point’s neighborhood do not affect the warp estimation at that point. However, it is sensible to remove mismatches in advance with a dedicated method. Mature methods exist that can be used without knowledge of camera intrinsics. Possible methods include [Ost+12] where the template is fitted directly in 2D using a stiff-to-flexible annealing scheme, RANSAC-based model fitting such as [Tra+12], or methods based on motion consistency between neighboring points [PB12]. There is no single best outlier detection method, and performance depends on the method to generate the point correspondences among other factors. We detail the outlier rejection methods used in the experimental section of this chapter.

5.1.7 Degeneracy analysis

We now analyze focal length degeneracies, occurring when f cannot be solved using Equation (5.12).

Theorem 16. (*local fSfT degeneracy.*) *Focal length is solved uniquely from Equation (5.12) if and only if $\|\nabla\alpha\|_2 \neq \mathbf{0}$. Geometrically, $\|\nabla\alpha\|_2 = \mathbf{0}$ occurs when either the surface normal becomes colinear with the optical axis (i.e. fronto-parallel where $\|\nabla z\|_2 = \mathbf{0}$) or when $z \rightarrow \infty$. The projection of such points is affine with no perspective effects.*

Proof. The proof is trivial by inspecting Equation (5.12). The right side becomes singular if and only if $\|\nabla\alpha\|_2 = \mathbf{0}$. By definition $\nabla\alpha = -\frac{f}{z^2}\nabla z$, so the singularity occurs when either $\|\nabla z\|_2 = \mathbf{0}$ or when $z \rightarrow \infty$. \square

This degeneracy is also a known degeneracy in plane-based focal length calibration [SM99], which is unsolvable if a planar object is fronto-parallel or if it is very far to the camera. In our case, because the surface is deformable and f is computed from the local motion at a surface point, there are two types of degeneracies. We have a *local degeneracy* that occurs at each surface point that is fronto-parallel, and a *global degeneracy* that occurs when either the whole surface is fronto-parallel or when it is very far from the camera. In both cases projection becomes affine and there exists no algorithm that could solve fSfT.

Theorem 17. (Sufficient well-posedness conditions for fSfT using point correspondences)
 When there exists at least one surface point where (i) there is not a local fSfT degeneracy (Theorem 16), (ii) the local weak perspective projection is a good model at that point and (iii) the point neighborhood geometry allows warp gradients to be computed uniquely, and , then fSfT is theoretically solvable with a unique focal length.

Proof. Condition (i) follows directly from Theorem 16. Condition (ii) comes from the local weak perspective model used in Equation (5.4). It is well-known that this approximation becomes more accurate for points that are closer to the camera’s optical axis and/or points with a smaller angle between their surface normal and the optical axis (typically $< 30^\circ$). Condition (iii) comes from the definition of the analytical solution that depends on warp gradients $\nabla\eta$, $\nabla\zeta$ and $\nabla\alpha$. These gradients can only be computed if the spatial configuration of the point neighborhood in the 2D template domain Ω allows these warps to be computed uniquely. If for example the points are colinear in Ω then the warp gradients cannot be computed. If (i), (ii) and (iii) are satisfied then fSfT is theoretically solvable with a unique focal length using Equation (5.12). \square

The conditions in Theorem 17 are sufficient but not necessary conditions for fSfT to be well-posed. Specifically, the analytical solution may not be able to solve all possible instances of fSfT that are theoretically solvable, so it has *artificial degeneracies*. There are two main cases. Firstly, it may be difficult to determine the correct solution if few points carry the right focal length. Such points must satisfy three conditions. Firstly, they must have some tilt angle to break the fronto-parallel degeneracy. Secondly, they cannot have excessively large tilt angles that violate the weak-perspective model. Thirdly, the geometry of a point’s neighborhood must allow the warp gradients to be computed uniquely. Recall that we compute warps using the TPS model that includes $L2$ regularization. This forces a unique solution, but the estimated warps will be unreliable when the neighborhood configuration is quasi-degenerate. For example, when the neighborhood points are approximately co-linear in Ω . If only a few points exist that satisfy all three criteria, it will be difficult to distinguish their correct estimates from the incorrect estimates of the other points.

Theorem 18. (Sufficient well-posedness conditions for fSfT with point correspondences)
 If there exists a local neighborhood for which the conditions in Theorem 17 are satisfied, then fSfT is well posed if and only if SfT-P is well-posed.

Proof. The proof is trivial because the three conditions guarantee a unique solution to focal length (Theorem 17). The remaining unknowns are the deformation of the template, which is equivalent to solving SfT with an intrinsically calibrated perspective camera (SfT-P). \square

5.2 Solving single-view fSfT with non-convex optimization

5.2.1 Section overview

The solution presented in §5.1 is fast, analytical and it has been used to understand the geometric conditions for which fSfT has a unique focal length solution. However, it is sub-optimal because an important relaxation is made: the surface depth ψ_z and surface gradient $\nabla\psi$ have been decoupled and they are treated as independent unknowns. This decoupling permits an analytical solution but at the price of losing geometric constraints. In this section we present a different and complementary approach that makes no such relaxation. We solve fSfT by iterative minimization of a large-scale

non-convex cost function. As a consequence, it requires an initial estimate of the unknowns, which can either be provided by the analytical solution or by sampling a small number of focal lengths. Unlike the analytical solution, where we started from the continuous variational problem, we start the optimization-based approach with discrete modeling using triangulated meshes. This is because the cost function implementation is strongly coupled with the representation, and triangulated meshes have proven excellent representations in the SfT-P literature.

This section is organized as follows. In §5.2.2 we give our template and cost function modeling. In §5.2.3 we present cost normalization and we show how the isometric weight, which is an important hyper-parameter that strongly affects accuracy, can be set with an unsupervised method without requiring ground truth. In §5.2.4 we detail how the cost function is optimized with an efficient quasi-Newton solution using multiple initializations. In §5.2.5 we give initialization details.

5.2.2 Problem modeling

5.2.2.1 Surface and deformation parameterization

We model the template surface \mathcal{R} using a discrete triangulated surface mesh. This is a flexible and general surface representation that can model \mathcal{R} of any topology. The method used to create the mesh is not our concern and standard software packages such as Meshlab [Cig+08] can be used. The mesh consists of vertices \mathcal{V} , edges \mathcal{E} and faces \mathcal{F} . We define as $\mathbf{y}_{i \in [1, V]} \in \mathcal{R}$ the known 3D position of vertex i in object coordinates, where V is the number of vertices. We define as θ_{rest} the known 3D positions of all vertices in object coordinates:

$$\theta_{rest} \stackrel{\text{def}}{=} \text{stk}(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_V) \quad (5.13)$$

We define as $\mathbf{x}_{i \in [1, V]} \in \mathcal{S}$ the unknown 3D position of vertex i in camera coordinates, and we define as θ the unknown positions of all vertices in camera coordinates:

$$\theta \stackrel{\text{def}}{=} \text{stk}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_V) \quad (5.14)$$

The mesh has connected and non-overlapping triangle elements that model the surface piecewise linearly. We use $\mathcal{T}(j) \in [1, V]^3$ to hold the indices of the three vertices associated to the j^{th} triangle. Each triangle forms three edges and we use $\mathcal{E}(k) \in [1, V]^2$ to hold the indices of the two vertices associated to the k^{th} edge.

The transformation of any surface point $\mathbf{p} \in \mathcal{R}$ to camera coordinates is uniquely determined by θ with barycentric interpolation. We associate to \mathbf{p} a unique triangle whose three vertices i , j and k enclose \mathbf{p} . The transformation of \mathbf{p} is linear in θ and is defined as follows:

$$g(\mathbf{p}; \theta) : \mathcal{R} \rightarrow \mathcal{S} \stackrel{\text{def}}{=} w_1 \mathbf{x}_i + w_2 \mathbf{x}_j + w_3 \mathbf{x}_k \quad (5.15)$$

where $0 \leq w_1, w_2, w_3 \leq 1$ are the known barycentric weights associated with point \mathbf{p} with $w_1 + w_2 + w_3 = 1$.

5.2.2.2 Cost function modeling

What makes a good cost function? We define the cost function formally as a smooth function $c(\theta, f) : \Theta \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ that maps deformation parameters θ and focal length f to a positive real cost. Making a well-designed cost function is important and non-trivial. We want it to have three key

characteristics: (i) a single global optimum at the true solution, (ii) a wide convergence basin without plateaus and (iii) to be applicable without modification for a broad variety of problem instances (templates, deformations, viewpoints, textures *etc.*), without needing specific hyper-parameter tuning. The cost function combines two competing criteria: *data costs*, which ensures that the projection of the deformed template agrees with the information present in the image, and *deformation costs*, which penalizes deformation that is not physically feasible or plausible. We use a cost inspired by the SfT-P literature with special attention to cost normalization to achieve (i) and (ii).

Choice of data costs. All SfT-P methods exploit motion information from the object’s texture. The most common approach that we also adopt uses point correspondences computed between the template’s surface and the image. A data cost encourages a deformation that aligns the point correspondences. Additional data costs based on other cues have been considered in the SfT-P literature, particularly shading [Liu+16b; GCB16b; Mor+09]. These are very useful to handle objects with poor texture, however, they introduce new challenges in terms of modeling (scene illumination, surface reflectance, *etc.*), they may require more scene assumptions such as directional illumination, and they may introduce additional local minima. For these reasons, we only consider motion information.

Cost function form. The cost function consists of three weighted terms as follows:

$$c(\theta, f; \mathcal{P}, \mathcal{Q}) = c_{data}(\theta, f; \mathcal{P}, \mathcal{Q}) + \lambda_{iso}c_{iso}(\theta) + \lambda_{reg}c_{reg}(\theta) \quad (5.16)$$

The term c_{data} is a data term that encourages deformation to agree with N point correspondences (\mathcal{P} and \mathcal{Q}). The term $c_{iso}(\theta) : \Theta \rightarrow \mathbb{R}^+$ is a deformation cost that penalizes stretching and shrinking of \mathcal{R} (non isometric-deformation). In the graphics and mechanics literature this is usually called a *membrane energy*, but in the SfT-P literature is often called an *isometric* cost. This embodies the fact that most deformable objects, particularly man-made ones made of materials such as leather, fabric, cardboard, stiff rubber or paper, deform without significant stretching and shrinking. Similarly to SfT-P, we require c_{iso} to make fSfT well-posed. Unlike the analytical solution which assumes deformation is exactly isometric, the use of a penalty cost allows some non-isometric deformation to be modeled and recovered. The term $c_{reg}(\theta) : \Theta \rightarrow \mathbb{R}^+$ is a second deformation cost that regularizes θ by encouraging smooth deformation. This is based on a convex approximation of the thin shell energy from mechanics, and it serves the dual purpose of regularization and convexifying c , which improves the convergence basin.

There are various ways to implement each of these costs. Our choices are mainly made to reduce tuning of the cost weights. We now summarize our implementations.

Isometric cost. The isometric cost is implemented using a discrete approximation of the elastic strain energy E_{strain} of continuous surfaces [Ter+87]:

$$E_{strain} = \int_{\mathcal{R}} \|\mathbf{I}_{\mathcal{R}} - \mathbf{I}_{\mathcal{S}}\|_F^2 d\mathcal{R} \quad (5.17)$$

Where $\mathbf{I}_{\mathcal{R}}$ and $\mathbf{I}_{\mathcal{S}}$ are the first fundamental forms of \mathcal{R} and \mathcal{S} respectively. Penalizing E_{strain} encourages a deformation to preserve the first fundamental form, equivalent to penalizing non-isometric deformation (local stretching or shrinking of the surface). We use a discrete approximation of E_{strain} using a Finite Element Model (FEM) with Constant Strain Triangles (CSTs). This is a well-known

model from mechanics that is suitable for relatively stiff (*i.e.* quasi-isometric) materials. Furthermore, using a FEM with CSTs gives a *consistent discretization* of the continuous strain energy. That is, under appropriate refinement conditions and norms, it is largely invariant to the mesh discretization and it converges to the continuous energy E_{strain} . This is important for our purposes because it eliminates the need to tune the cost's weight λ_{iso} according to the mesh discretization (number of vertices, placement of vertices and the triangulation). This is not true for the majority of membrane-like costs used in the Sft-P literature which are not consistent, such as the preservation of mesh edge lengths [Ost+12; Bru+10; GCB16a] or the As-Rigid-As-Possible cost from [SA07], as discussed in [LG15]. The independence of λ_{iso} on the mesh becomes truer as the mesh becomes denser. Consequently, as the mesh becomes denser we worry less about its influence on λ_{iso} .

A CST is a triangular element whose stress and strain fields are constant in the triangle's domain. Each triangle $t \in [1, T]$ defines a planar surface region $\mathcal{R}_t \in \mathcal{R}$. This is parameterized by a linear function $\zeta_t(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ that embeds \mathcal{R}_t from 2D coordinates Ω . The first fundamental form is $\nabla^\top \zeta_t \nabla \zeta_t$, which is constant over the triangle. Similarly, we define as ϕ_t the embedding of triangle t to camera coordinates. We recall that ζ_t is known but ϕ_t is unknown. The isometric cost c_{iso} is constructed by the following discrete approximation of E_{strain} :

$$c_{iso} = \sum_{t=1}^T a_t \|G(\nabla \zeta_t) - G(\nabla \phi_t)\|_F^2 \approx E_{strain} \quad (5.18)$$

where a_t is the known surface area of the t^{th} triangle and $G(\mathbf{X}) \stackrel{\text{def}}{=} \mathbf{X}^\top \mathbf{X}$ is the Gramian operator. Because ζ_t and ϕ_t are linear within the domain of triangle t , their gradients are computed in closed-form from the vertices of triangle t . They are the solutions to the following linear equations:

$$\begin{aligned} \nabla \zeta_t (\mathbf{u}_t^i - \frac{1}{3} (\mathbf{u}_t^1 + \mathbf{u}_t^2 + \mathbf{u}_t^3)) &= (\mathbf{y}_t^i - \frac{1}{3} (\mathbf{y}_t^1 + \mathbf{y}_t^2 + \mathbf{y}_t^3)), \quad \forall i \in [1, 3] \quad (a) \\ \nabla \phi_t (\mathbf{u}_t^i - \frac{1}{3} (\mathbf{u}_t^1 + \mathbf{u}_t^2 + \mathbf{u}_t^3)) &= (\mathbf{x}_t^i - \frac{1}{3} (\mathbf{x}_t^1 + \mathbf{x}_t^2 + \mathbf{x}_t^3)), \quad \forall i \in [1, 3] \quad (b) \end{aligned} \quad (5.19)$$

where \mathbf{u}_t^i , \mathbf{y}_t^i and \mathbf{x}_t^i denote the 3D position of the i^{th} vertex of triangle t in uv , template and world coordinates respectively. To compute $\mathbf{u}_t^{i \in [1, 3]}$, we rigidly transform the triangle's vertices in object coordinates to the plane $z = 0$ and then we drop the z coordinate. This makes ζ_t isometric with the uv plane with $G(\nabla \zeta_t) = \mathbf{I}_2$. We compute $G(\nabla \phi_t)$ by inverting the linear equations in Equation (5.19-b). This makes $\nabla \phi_t$ a linear expression in $\mathbf{x}_t^{i \in [1, 3]}$, so it is a linear expression in θ by definition (Equation (5.14)). Consequently c_{iso} is a cubic (non-convex) expression in θ .

Regularization cost. The regularization cost c_{reg} is convex and it encourages smooth deformation. Various implementations could be used, and we use a simple one using the moving least squares energy [SMW06]. First the mesh is divided into overlapping *cells* where each cell describes the local motion of the mesh. The most common way to define cells is with one cell per vertex, and each cell contains the vertex and all other vertices connected by one mesh edge. The cell's motion is determined by the movement of its constituent vertices. Regularization is imposed by encouraging the cell's motion to be affine. This is implemented by determining the least-squares affine motion of the cell from object to camera coordinates, and then comparing the affine motion to the actual motion of the cell in the $L2$ sense. The cost is built by accumulating residual motion vectors for each cell and summing squared residuals. It is straightforward to show that the residuals are linear in θ , so the cost has the following form:

$$c_{reg}(\theta) \stackrel{\text{def}}{=} \|\mathbf{A}_{reg} \theta\|_2^2 \quad (5.20)$$

where \mathbf{A}_{reg} is a known $m \times 3V$ sparse matrix that does not depend on θ and $m > 3V$. The vector $\mathbf{A}_{reg}\theta$ holds the vector of residuals from all cells. The cost is zero if and only if all cells transform according to affine motion. Thanks to the cells overlapping, this cost encourages neighboring cells to transform with similar affine motion, thus penalizing non-smooth deformation.

Unlike c_{iso} , c_{reg} is not consistent, which means it depends strongly on the mesh discretization. The reason is similar for why the ARAP mesh energy is not consistent as discussed in [LG15]. Indeed, constructing a consistent and convex regularization cost with surface meshes is not trivial, and it has not been achieved before in the SfT literature. We handle this using cost normalization, detailed later in §5.2.3.1.

Data cost. The data cost is based on the point correspondence reprojection error. The position of each point $\mathcal{P}(i) \in \mathcal{P}$ on the template's surface is measured in image coordinates by the 2D point $\mathcal{Q}(i) \in \mathcal{Q}$. These are related by the reprojection equation:

$$\pi(g(\mathcal{P}(i), \theta), f) = \mathcal{Q}(i) + \epsilon_i \quad (5.21)$$

where $\epsilon_i \in \mathbb{R}^2$ is unknown noise. In the special case of independent and identically distributed (IID) Gaussian noise, the statistically optimal data cost c_{data}^l is

$$c_{data}^l(\theta, f; \mathcal{P}, \mathcal{Q}) \stackrel{\text{def}}{=} \sum_{i=1}^N \frac{1}{\sigma^2} \|\pi(g(\mathcal{P}(i); \theta), f) - \mathcal{Q}(i)\|_2^2 \quad (5.22)$$

where σ is an estimate of the noise standard deviation. The value of σ depends on several factors: the method used to generate point correspondences, image resolution and image noise. Unless σ is known, we use as default

$$\sigma = \max(w, h)/640 \quad (5.23)$$

This corresponds to a noise standard deviation of 1 pixel at VGA resolution.

Equation (5.22) is used frequently in prior SfT works with calibrated cameras, and it is suitable when the point correspondences do not contain outliers. If outliers are present they severely affect the data cost and corrupt the solution. We assume that the vast majority of outliers have been filtered in a pre-processing stage, as described in §5.1.6.2. Occasionally some outliers may remain, and as a safety guard we introduce robustness into the data cost using the Huber M-estimator ρ_h . The data cost writes as follows:

$$\begin{aligned} c_{data}(\theta, f; \mathcal{P}, \mathcal{Q}) &= \sum_{i=1}^N \frac{1}{\sigma^2} \rho(\pi(g(\mathcal{P}(i); \theta), f) - \mathcal{Q}(i)) & (a) \\ \rho(\text{stk}(x, y)) &\stackrel{\text{def}}{=} \rho_h(x) + \rho_h(y) & (b) \\ \rho_h(z) &\stackrel{\text{def}}{=} \begin{cases} \frac{1}{2}z^2 & \text{if } |z| < k \\ k(|z| - \frac{1}{2}k) & \text{otherwise} \end{cases} & (c) \end{aligned} \quad (5.24)$$

where k is the Huber constant, set to a default $k = 10\sigma$.

Residuals, Jacobian matrices and cost linearization. The terms c_{data} , c_{reg} and c_{iso} have a special form: they are each the summation of M-estimators. In the case of c_{iso} and c_{reg} the M-estimators are $L2$ and in the case of c_{data} the M-estimators are Huber. Consequently, c has the following general form:

$$\begin{aligned}
 c(\theta, f) &= \sum_{i=1}^R \rho_i(r_i(\theta, f)) \\
 \rho_i(z) \in \mathbb{R}^+ &\stackrel{\text{def}}{=} \begin{cases} \frac{1}{2}z^2 & (L2) \\ \rho_h(z) & (\text{Huber}) \end{cases}
 \end{aligned} \tag{5.25}$$

where R is the total number of residual terms and $r_i \in \mathbb{R}$ is the residual inputted to each M-estimator. We now define the following terms that are used for several purposes: to efficiently optimize c with IRWLLS using a quasi-Newton method, for weight normalization and for unsupervised selection of the isometric weight λ_{iso} :

- \mathbf{r}_Y : the vector of M-estimator residuals associated with the cost term c_Y , $Y \in \{c_{data}, c_{reg}, c_{iso}\}$
- \mathbf{J}_Y^X : the Jacobian matrix of \mathbf{r}_Y with respect to the unknowns $X \in \{\theta, f\}$
- $\mathbf{J}_Y \stackrel{\text{def}}{=} [\mathbf{J}_Y^\theta \mathbf{J}_Y^f]$: the Jacobian matrix of \mathbf{r}_Y with respect to all unknowns
- $c(\theta' + \Delta\theta, f' + \Delta f) \approx \|\mathbf{A}_c \text{stk}(\Delta\theta, \Delta f) - \mathbf{b}_c\|^2$: The linearization of c with respect to θ and f about the values f' and θ' . This system is constructed by the 1st-order approximation of c at (θ', f') . The vector \mathbf{b}_c has size R and the matrix \mathbf{A}_c has R rows.

By applying the chain rule, we see that the linearized system has the following form:

$$\mathbf{A}_c \stackrel{\text{def}}{=} \text{diag}(\mathbf{w}) \begin{bmatrix} \mathbf{J}_{data}^\theta & \mathbf{J}_{data}^f \\ \lambda_{iso} \mathbf{J}_{iso}^\theta & \mathbf{0} \\ \lambda_{reg} \mathbf{J}_{reg}^\theta & \mathbf{0} \end{bmatrix}, \mathbf{b}_c \stackrel{\text{def}}{=} \text{diag}(\mathbf{w}) \begin{bmatrix} \mathbf{r}_{data} \\ \lambda_{iso} \mathbf{r}_{iso} \\ \lambda_{reg} \mathbf{r}_{reg} \end{bmatrix} \tag{5.26}$$

where $\mathbf{w} \in \mathbb{R}^{+R}$ is the IRLS weight vector. The effect of \mathbf{w} is to reduce the influence of larger residuals depending on the chosen M-estimators. In our case, because the M-estimators for the cost terms c_{data} , c_{iso} and c_{reg} are Huber, $L2$ and $L2$ respectively, \mathbf{w} is of the form $\mathbf{w} = \text{stk}(\mathbf{w}_{data}, \mathbf{w}_{reg}, \mathbf{w}_{iso})$ where \mathbf{w}_{reg} and \mathbf{w}_{iso} are vectors of all-ones. We emphasize that \mathbf{w} should not be confused with the cost weights λ_{reg} and λ_{iso} . One can think of λ_{reg} and λ_{iso} as global weights that influence the strength of the regularization and isometric costs relative to the data cost. In contrast, \mathbf{w} re-weights the influence of point correspondences according to their residual error.

Because we use a mesh model, the Jacobians \mathbf{J}_{data}^θ , \mathbf{J}_{iso}^θ and \mathbf{J}_{reg}^θ are highly sparse because each residual depends on only a small number of mesh vertices. By contrast, the Jacobian \mathbf{J}_{data}^f is a dense $2N \times 1$ vector because all data residuals depend on focal length.

5.2.3 Cost normalization and weight selection

The cost function weights λ_{iso} and λ_{reg} are critical hyper-parameters that balance the influence of the isometric deformation cost c_{iso} and the regularization deformation cost c_{reg} respectively on c . Correct weight selection an important and non-trivial issue in many optimization-based approaches for solving registration problems. In our case, the weights must be set appropriately to ensure c has the desired behavior as described in §5.2.2.2, *What makes a good cost function?*. We have developed two strategies that eliminates the need to manually tune the weights at run-time, which is vital to have a practical solution. The first strategy is *cost normalization* for improving hyper-parameter invariance. Cost normalization can be thought of as reducing the space of possible weights \mathcal{W} that contains the optimal weights (*i.e.* the weights that lead to a cost whose minimum corresponds with the true solution). However, even with cost normalization, we will always acquire a method to actually

find good weights within \mathcal{W} . If ground truth data is available (camera intrinsics and the object’s 3D deformations), the weights can be set automatically with optimization or a grid search (we call this *supervised weight selection*). If cost normalization were to be perfect, then we could perform supervised weight selection just once, and reuse the found weights for all problem instances. However, in practice it is very difficult to achieve perfect normalization. To overcome this challenge, we have also developed an approach to automatically select the isometric weight, which is the most sensitive weight, without ground truth information. We refer to this approach as *unsupervised weight selection*. Although normalization is not strictly necessary for unsupervised weight selection, normalization helps by significantly reducing the weight search space, and therefore reduce the computation time of unsupervised weight selection.

5.2.3.1 Normalization

In the SFT-P literature weight tuning is usually performed by hand. We can reduce this problem significantly with some simple normalization techniques, allowing the same weights to be used for a broad range of inputs (templates, imaging conditions, point correspondence method *etc.*). Our normalization techniques make c strongly invariant to the following factors:

1. Template scale
2. Template discretization
3. Number of correspondences
4. Image resolution

Template scale invariance is required because the cost of deforming a larger template according to c_{iso} and c_{reg} is greater than that of a smaller template. This is undesirable. We normalize by *a priori* isotropic rescaling of the template to a canonical size, which we define as having a total surface area of 1. Template discretization invariance is important, as already discussed in §5.2.2.2, *Isometric cost*. We achieve this with two strategies. The first strategy is to use a discrete approximation of the continuous strain energy with a FEM, which achieves good discretization invariance by construction. The second strategy is to re-weight cost that depends strongly on the discretization, such as c_{reg} . We apply a rough global reweighing so that a small deformation from the rest state induces approximately the same cost irrespective of the template’s discretization. We achieve this by re-weighting a cost c_Y as follows:

$$c_Y \leftarrow \frac{1}{\|\mathbf{J}_Y^0\|_F^2} c_Y \quad (5.27)$$

where \mathbf{J}_Y^0 is the Jacobian matrix of the cost evaluated before deformation is applied (*i.e.* evaluated at the template’s natural rest state). We make c_{data} invariant to the number of point correspondences by re-weighting $c_{data} \leftarrow \frac{1}{N} c_{data}$. We have already made c_{data} invariant to the image resolution by rescaling the noise standard deviation σ by the image size, as done in Equation (5.23). Thanks to these normalization techniques, we use the same weights λ_{iso} and λ_{reg} for *all* templates and all test datasets, where mesh resolutions vary considerably from $O(100)$ to $O(1,000)$ vertices, number of point correspondences vary from $O(10)$ to $O(1,000)$, and image resolutions vary from VGA to high definition (3600×2400 pixels).

5.2.3.2 Weight selection

The above normalization techniques significantly reduce the range of suitable cost weights, because they standardize the cost magnitudes across different conditions (different templates, number of point correspondences, image resolution *etc.*). Nevertheless, we must still set λ_{iso} and λ_{reg} properly to ensure that c has in general a strong local minimum near the true solution. The weight λ_{iso} has a significant impact on this. By contrast the weight λ_{reg} governs the regularizer, used to convexify c , and its precise setting is much less important than λ_{iso} . We use a default of $\lambda_{reg} = 1e - 3$ in all tests, found experimentally.

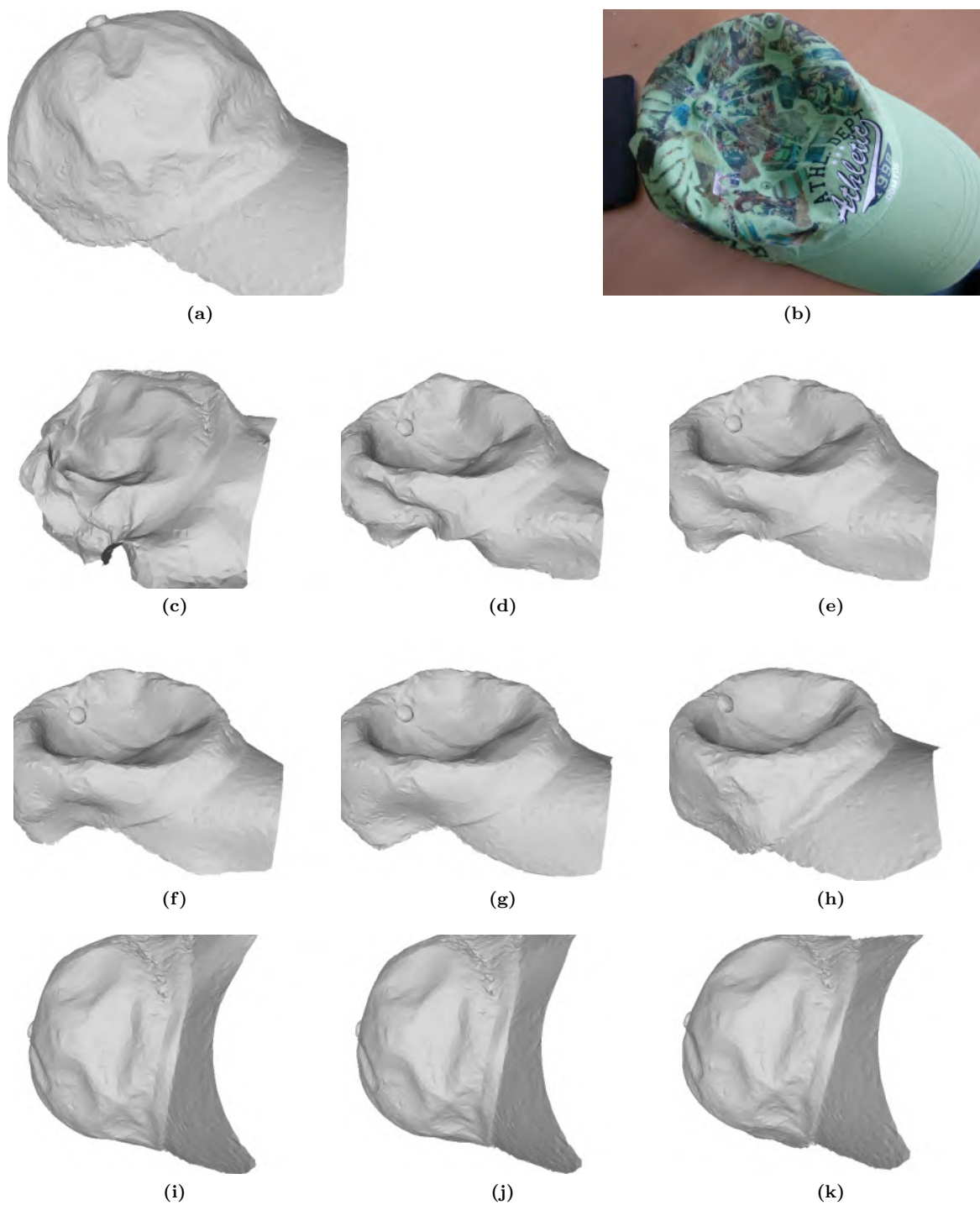


Figure 5.2: The influence of isometric weight λ_{iso} on the fSfT solution. (a) shows the template from the Cap dataset, (b) shows the first input image in the Cap dataset. (c-k) show the reconstructed 3D templates using a different isometric weight λ_{iso} . The weights increase from (c) to (k), ranging from very low (creating a highly flexible template) and very high (creating a highly stiff template). In order from (c) to (k) the weights are $\lambda_{iso} = 100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600$. The meshes are rendered from the same 3D viewpoint. The significant difference in orientation of (i), (j) and (k) compared to the others are because their high isometric weights make the template very stiff, preventing the cap's deformation being recovered. Thus, they are similar to the best-fitting rigid transformation of the template, which gives a completely wrong answer due to the strong true deformation.

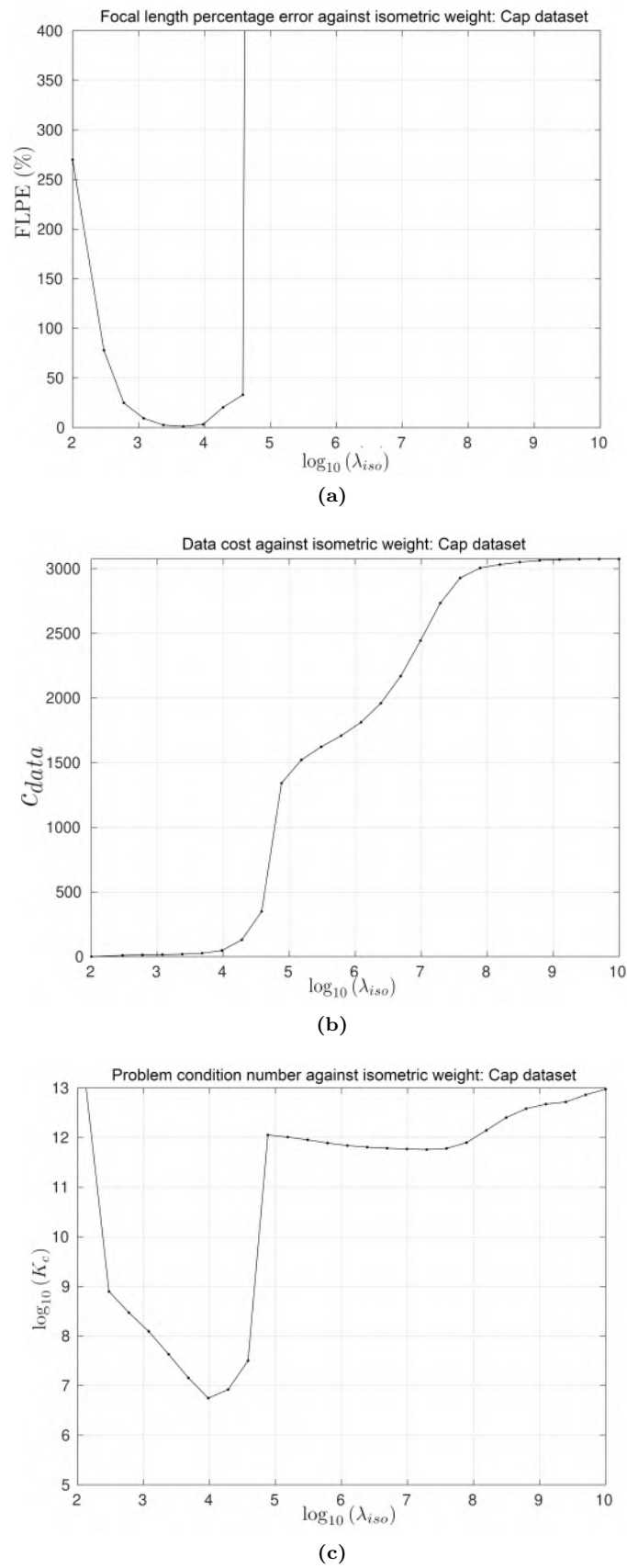


Figure 5.3: Graphs to visualize the influence of isometric weight on focal length percentage error (a), data energy (b) and problem condition (c). The graphs are plotted for the Cap dataset using the first input image.

Isometric weight sensitivity example. We illustrate the importance of selecting λ_{iso} properly with a real-world example from the public Cap dataset (full dataset details are presented in §5.4.2). In Figure 5.2 (a) we show the cap’s template mesh and in Figure 5.2 (b) we show the first input image from the Cap dataset, where significant deformation creates a large depression in the cap’s crown region. Using the point correspondences provided at the textured region, we have optimized c with local optimization (Gauss-Newton) as described in §5.2.4.4, initialized with the ground truth focal length and 3D deformation. We performed this with 26 different weights ranging from $\lambda_{iso} = 10e+2$ to $\lambda_{iso} = 10e+10$, with a multiplication step factor of 2. In Figure 5.2 (c-k) we visualize 9 reconstructions after optimization. Each image shows results with a different λ_{iso} , and the images are ordered with increasing λ_{iso} . To aid visualization we zero-centered the reconstructions and render them from a fixed virtual viewpoint. Qualitatively, we see that for smaller λ_{iso} , the Cap’s shape is poorly recovered (c-d) due to the small influence of the isometric cost. For large weights the cap undergoes practically no deformation (rigid motion) and the cap’s large depression is not recovered (i-k). This is because the isometric cost becomes very strong and it prevents deformation from the rest shape. Results with intermediate weights are visualized in Figure 5.2 (e-h) where we see plausible reconstructions. However in (e-h) there is a clear difference in shape, showing the importance of correct weight selection.

In Figure 5.3 (a) we plot the Focal Length Percentage Error (FLPE) after optimization against λ_{iso} . For a given focal length estimate \hat{f} with ground truth f^{gt} , the Focal Length Percentage Error (FLPE) is defined as follows:

$$\text{FLPE}(\hat{f}, f) \stackrel{\text{def}}{=} 100 \times \frac{|\hat{f} - f^{gt}|}{f^{gt}} \quad (5.28)$$

We see a clear global minimum in FLPE at approximately $\lambda_{iso} = 10e + 3.75$, where $\text{FLPE} = 3.2\%$. The corresponding solution for $\lambda_{iso} = 10e + 3.75$ is shown in Figure 5.2(f). This low error indicates we are able to correctly determine the focal length provided that a good isometric weight is used. We see a very large increase in FLPE for smaller weights ($< 10e + 3$) and larger weights ($> 10e + 4.5$). The valley is relatively narrow, where a range $10e + 3.50 \leq \lambda_{iso} \leq 10e + 4.0$ produces near optimal results, indicating that the range of ‘good’ weights is relatively narrow and within a factor of 5 of the optimal weight. This indicates the importance of correct weight selection. In Figure 5.3 (b) we plot the data energy c_{data} after optimization against λ_{iso} . There is a clear relationship where decreasing λ_{iso} decreases c_{data} monotonically. This occurs because for lower λ_{iso} the template is overly-flexible and it can fit to noise with a very low data cost.

Supervised weight selection. We can find the best λ_{iso} with supervision, using problem instances with known focal length and without needing any 3D information (3D template deformation). This makes it practical in SfT applications where ground truth 3D is usually hard to obtain. We have implemented this using a basic logarithmic grid search, using the Median Focal Length Percentage Error (Med-FLPE) as the objective function. Given a set of S problem instances, with focal length estimates $\{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_S\}$ and ground truth focal lengths $\{f_1^{gt}, f_2^{gt}, \dots, f_S^{gt}\}$, Med-FLPE is the median of $\{\text{FLPE}(\hat{f}_1, f_1^{gt}), \dots, \text{FLPE}(\hat{f}_S, f_S^{gt})\}$. We use the median to be robust to quasi-degenerate instances and instances where the global minimum is not found.

Unsupervised weight selection. We have also created a simple unsupervised approach that requires neither ground truth focal length nor 3D deformation. Furthermore, it has the desirable property of having no additional parameters nor thresholds. Our idea is that λ_{iso} affects the conditioning

of fSFT, and a good λ_{iso} leads to a well-conditioned problem, equivalently to where the unknowns f and θ are least sensitive to noise. We represent this relationship by the function $k_c(\lambda_{iso}) : \mathbb{R} \rightarrow \mathbb{R}^+$, giving the condition number of c as a function of λ_{iso} , where a lower value indicates better conditioning. In the unsupervised approach, we select λ_{iso} as the weight giving the best conditioned problem: $\arg \min_{\lambda_{iso}} k_c(\lambda_{iso})$.

Because c has no closed-form solution, we cannot evaluate k_c in closed form. We therefore evaluate k_c by optimizing c with iterative refinement (§5.2.4.4), linearizing c about the final solution with $c(\theta, f) \approx \|\mathbf{A}_c \text{stk}(\theta, f) - \mathbf{b}_c\|^2$ as defined in Equation (5.26), then evaluating the condition number of \mathbf{A}_c . This is defined as the ratio of the largest and smallest singular values of \mathbf{A}_c . We illustrate the approach using the cap dataset. In Figure 5.3 (b) we plot $k_c(\lambda_{iso})$ using the 26 isometric weights, and we see that there is a clear global minimum at approximately $\lambda_{iso} = 10e + 4$. As $\lambda_{iso} \rightarrow 0$ and $\lambda_{iso} \rightarrow +\infty$, we see a strong decrease in problem condition (increase in k_c). Importantly, from Figure 5.3 (a), not only is there a strong global minimum, but this minimum corresponds to a small FLPE of 4.1%. This is close to the best possible FLPE obtainable among all possible weights (3.2%). The corresponding deformation solution with $\lambda_{iso} = 10e + 4$ is shown in Figure 5.2(g).

This example suggests that λ_{iso} can be automatically set by sampling weights and using the one yielding the lowest k_c . We have tested the approach empirically with many different templates and images, and we show in the experimental section of this chapter that it is indeed a good approach to find the optimal weight. This unsupervised approach can be applied in several modes. We test the following configurations using a simple grid search over λ_{iso} :

1. *Image specific selection*: Find λ_{iso} specific to each template and each image.
2. *Template-specific selection*: Find λ_{iso} specific to a template, and apply it for all problem instances with that template.
3. *Template-generic selection*: Find a single λ_{iso} for all templates and images.

These approaches are compared in the experimental section of this chapter.

Why would this unsupervised approach work from a theoretical standpoint? We first consider the asymptotic behavior. The ratio of the largest and smallest eigenvalues of $\mathbf{A}_c^\top \mathbf{A}_c$ gives k_c where \mathbf{A}_c decomposes as follows:

$$\begin{aligned}
 \mathbf{A}_c^\top \mathbf{A}_c &\in \mathbb{R}^{(|\theta|+1) \times (|\theta|+1)} = \mathbf{A}^\top \mathbf{A} + \lambda_{reg}^2 \mathbf{B}^\top \mathbf{B} + \lambda_{iso}^2 \mathbf{C}^\top \mathbf{C} & (a) \\
 \mathbf{A} &\stackrel{\text{def}}{=} \text{diag}(\mathbf{w}_{data}) \begin{bmatrix} \mathbf{J}_{data}^\theta & \mathbf{J}_{data}^f \end{bmatrix} & (b) \\
 \mathbf{B} &\stackrel{\text{def}}{=} \text{diag}(\mathbf{w}_{reg}) \begin{bmatrix} \mathbf{J}_{reg}^\theta & \mathbf{0} \end{bmatrix} & (c) \\
 \mathbf{C} &\stackrel{\text{def}}{=} \text{diag}(\mathbf{w}_{iso}) \begin{bmatrix} \mathbf{J}_{iso}^\theta & \mathbf{0} \end{bmatrix} & (d)
 \end{aligned} \tag{5.29}$$

The matrices \mathbf{B} and \mathbf{C} are singular by definition because their last rows are all-zeros. The matrix \mathbf{A} is also always singular because the Jacobian \mathbf{J}_{data}^θ always has a non-empty kernel. Intuitively, this is because c_{data} encourages the mesh to align point correspondences from the image, and it can never fully constrain the mesh because of a depth/scale ambiguity. Therefore, one of the kernel dimensions of \mathbf{J}_{data}^θ corresponds to this ambiguity. Indeed this is the very ambiguity that is broken by using c_{iso} . If we were to scale the surface by an arbitrary factor s in camera coordinates (equivalent to scaling θ by s), the projection of points on the surface is the same, generating a null-space. As $\lambda_{iso} \rightarrow +\infty$, the eigenvalues of $\mathbf{A}_c^\top \mathbf{A}_c$ tend to the eigenvalues of $\mathbf{C}^\top \mathbf{C}$, which is singular, consequently $\lim_{\lambda_{iso} \rightarrow +\infty} k_c = +\infty$. Intuitively, the problem becomes poorly conditioned when the template becomes

extremely stiff. At first this may seem counter-intuitive but it is the result of formulating fSfT with weighted costs. Equivalently, the metric constraints, required to disambiguate surface stretching from depth, are supplied only by the isometric cost c_{iso} . In the limit there are no metric constraints, so there are an infinite number deformation/depth combinations that have the same cost, producing an ill-conditioned problem. Similarly, $\lim_{\lambda_{iso} \rightarrow 0} k_c = +\infty$ because \mathbf{B} is singular. The limits of k_c are therefore clear, but we have not yet been able to show theoretically why the global minimum of k_c should be close to the optimal isometric weight, which is non-trivial analysis. However, we show empirically that it is indeed reasonably close in the experimental section of this chapter.

5.2.4 Cost optimization

5.2.4.1 Approach overview

Our goal is to solve fSfT by as the following optimization problem P

$$P : \arg \min_{\theta, f} c(\theta, f; \mathcal{P}, \mathcal{Q}) \quad (5.30)$$

This is large-scale and non-convex problem with no known closed-form solution. Nevertheless, we propose an approach based on multi-start local optimization that proves very effective in practice. Only a very small number of initializations are required to achieve good results (*i.e.* 3 or fewer). We use (θ_S, f_S) to denote an initialization, where $k \in [1, S]$ denotes the initialization index and $S \geq 1$ is the number of initializations. Local optimization is run from each initialization, and the solution yielding the lowest overall cost is returned. There is a trade-off in having a larger S , which increase computational cost but may also increase the chances of finding the global minimum. We find that only a small S is required in practice with little benefit for $S > 3$. To reduce cost further, a mechanism is proposed to terminate repeated search of the same search region from different initializations. We first define the approach in general terms independent of the local optimization implementation and methods used to generate the initializations. We then give implementation details of these aspects.

5.2.4.2 Optimization pseudo-code

Our optimization process to solve P is summarized in pseudo-code in Algorithm 8. Each initialization is processed (either in parallel or sequentially) and local optimization is performed in two steps (lines 7 and 8). At line 6 deformation is optimized with focal length fixed, and at line 7 they are both optimized jointly. These two steps are used to improve convergence especially when the start focal length is far from the true solution. Optimization is run until termination criteria are satisfied, denoted by T_1 and T_2 respectively. When all initializations have been processed, the solution with lowest cost is taken, and a final refinement is performed with local optimization using termination criteria T_3 .

5.2.4.3 Termination criteria and search history

We maintain a search history set \mathcal{H} to prevent unnecessary repeated search from different initializations. This holds all the solutions that have been found from other initializations. During local optimization, we continually measure the distance of the current estimate \hat{f} and $\hat{\theta}$ to the closest member of \mathcal{H} using a distance function $d_{\mathcal{H}}((\hat{\theta}, \hat{f}), \mathcal{H})$ defined below. We terminate local optimization early if $d_{\mathcal{H}}((\hat{\theta}, \hat{f}), \mathcal{H}) \leq \tau_{\mathcal{H}}$ where $\tau_{\mathcal{H}}$ is a threshold. The distance function is designed to tell us when the current estimate is very likely to converge on a solution that already exists in \mathcal{H} . We have found that

Algorithm 8 fSfT Optimization**Require:**

```

     $\{(f_1, \theta_1), \dots, (f_S, \theta_S)\}$  ▷ initializations
     $c$  ▷ cost function
1: function fSfT_optimize( $\{(f_1, \theta_1), \dots, (f_S, \theta_S)\}, c$ )
2:    $c^* \leftarrow \infty$  ▷ lowest cost found so far
3:    $\mathcal{H} \leftarrow \emptyset$  ▷ search history
4:    $f^* \leftarrow 0, \theta^* \leftarrow \mathbf{0}$  ▷ best solution with cost  $c^*$ 
5:   for  $s \in [1, S]$  do
6:     initialize estimates:  $\hat{f} \leftarrow f_s, \hat{\theta} \leftarrow \theta_s$ 
7:     locally optimize  $c$  w.r.t.  $\hat{\theta}$  until stopping criteria  $T_1$  satisfied.
8:     locally optimize  $c$  w.r.t.  $\hat{\theta}$  and  $\hat{f}$  until stopping criteria  $T_2$  satisfied.
9:     update history:  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\hat{f}, \hat{\theta})\}$ 
10:    if  $c(\hat{f}, \hat{\theta}) < C^*$  then
11:       $(f^*, \theta^*) \leftarrow (\hat{f}, \hat{\theta})$ 
12:       $c^* \leftarrow c(f^*, \theta^*)$ 
13:  Final refinement: locally optimize  $c$  initialized with  $(f^*, \theta^*)$  until stopping criteria  $T_3$  satisfied.
14:  return  $f^*$  and  $\theta^*$ 

```

this is more influenced by the estimated surface orientation rather than focal length or the absolute depth of the surface. Consequently, we measure distance in terms of surface normal dissimilarity as follows:

$$d_{\mathcal{H}}((\theta, f), \mathcal{H}) = \min_{(\theta', f') \in \mathcal{H}} \max_{t \in [1, T]} \text{abs}(\angle(\mathbf{n}_t(\theta), \mathbf{n}_t(\theta'))) \quad (5.31)$$

where $\mathbf{n}_t(\theta)$ is the surface normal for triangle t generated by θ , and $\angle(\mathbf{a}, \mathbf{b})$ is the angle in degrees between vectors \mathbf{a} and \mathbf{b} .

Early termination at lines 7, 8 and 13 is done if any one of the following criteria are satisfied during local optimization:

1. **Maximum iterations:** The number of iterations τ_{step} has been performed
2. **Small parameter update:** The relative change of all unknowns is below a threshold τ_{Δ}
3. **Small cost update:** The relative change of c is below a threshold τ_c
4. **Out-of-bounds focal length:** \hat{f} is out of bounds: $\hat{f} \leq f_{min}$ or $\hat{f} \geq f_{max}$
5. **Repeated search:** The current solution is similar to one already in the search history: $d_{\mathcal{H}}((\hat{\theta}, \hat{f}), \mathcal{H}) \leq \tau_{\mathcal{H}}$.

The first three stopping criteria are standard in local optimization. The fourth criterion is used to terminate early if optimization is converging on a focal length solution that is clearly wrong. Normally this happens either when the problem is degenerate or when optimization has been very poorly initialized. We use $f_{min} = 0.1w$ and $f_{max} = 1000w$ where w is the image width.

The termination criteria T_1 , T_2 and T_3 in Algorithm 8 define values for τ_{step} , τ_{Δ} , τ_c and $\tau_{\mathcal{H}}$. We use T_1 and T_2 to terminate more aggressively compared to T_3 , which can significantly reduce computational cost. The default values are defined in Table 5.1.

	τ_{step}	τ_{Δ}	τ_c	$\tau_{\mathcal{H}}$
T_1	10	$1e-5$	$1e-5$	20
T_2	20	$1e-5$	$1e-5$	20
T_3	100	$1e-5$	$1e-5$	n/a

Table 5.1: Default termination values used in Algorithm 8

5.2.4.4 Local optimization implementation with Gauss-Newton

Why Gauss-Newton? We implement local optimization by adapting an established gradient-based approach: IRWLLS with a quasi-Newton method (Gauss-Newton). We include some enhancements to significantly reduce the computational cost for determining the step direction, detailed in this section. Gauss-Newton is used to exploit the special form of c , which is a weighted sum of M-estimators (§5.2.2.2, *Residuals, Jacobian matrices and cost linearization*). Consequently, c can be optimized with Gauss-Newton with fast (quadratic) convergence without needing to compute nor invert the problem’s Hessian matrix. For large scale problems such as ours, this would be completely prohibitive. The speedups we introduce allow us to compute the step direction in approximately 10 ms on a standard desktop PC with sub-optimal code¹. Typically convergence is achieved within 10-20 iterations. We have also tested optimization with Levenberg-Marquardt using the same speed optimizations but we found there was not a significant benefit in convergence rate.

Efficient step direction estimation. Given the current estimate of the unknowns (θ', f') , the next estimate is computed with Gauss-Newton in 4 stages:

1. Linearize c about (θ', f') using Equation (5.26).
2. Solve the linearized system to determined step directions Δ_{θ} and Δ_f
3. Apply Armijo backtracking line search to determine the step length γ
4. Produce the next estimate: $\theta' \leftarrow \theta' + \gamma\Delta_{\theta}$, $f' \leftarrow f' + \gamma\Delta_f$.

The process repeats until one or more termination criteria are satisfied as described in the previous section. We perform step 2 by assembling the normal equations and inverting the exact linear system $\mathbf{A}_c^{\top} \mathbf{A}_c \text{stk}(\Delta\theta, \Delta f) = \mathbf{A}_c^{\top} \mathbf{b}_c$. The matrix $\mathbf{A}_c^{\top} \mathbf{A}_c$ is highly sparse, however the computational cost of assembling and solving the system can be very high because a mesh may contain many vertices, with θ having *e.g.* $O(10^5)$ unknowns or above. Some preprocessing can be applied: the regularization cost is convex, so $\mathbf{J}_{reg}^{\top} \mathbf{J}_{reg}$ is constant at each Gauss-Newton iteration, and it must be computed only once. However, the other Jacobian matrices must be recomputed at each iteration. We apply two techniques to significantly reduce the computational cost.

The first technique is dimensionality reduction. We apply this when deformation is generally smooth and does not create significant folding or creasing. We implement this with linear bases as follows. We model θ with $L < 3V$ linear bases: $\theta = \mathbf{B}_{\theta} \tilde{\theta}$, where \mathbf{B}_{θ} is a known $3V \times L$ basis matrix, and $\tilde{\theta}$ is an unknown vector of size L (the basis coefficients). A smaller L gives greater dimensionality reduction. We construct \mathbf{B}_{θ} using modal analysis from [CB15] as follows. Each column of \mathbf{B}_{θ} is a right singular vector of $\mathbf{J}_{reg}^{\theta}$ sorted by increasing singular value. As L increases we introduce more bases and more deformation modes, and we are able to recover less smooth deformations. We find that a default of $L = \min(200, 3V)$ bases works well for a broad range of templates and deformations and

¹Time benchmarks are evaluated on a standard 64 bit desktop with AMD Ryzen 1700 CPU and NVidia 1080 GPU.

the deformation space is sufficiently large to capture very complex deformation. We solve the normal equations with respect to a low-dimensional update vector $\Delta_{\tilde{\theta}} \in \mathbb{R}^L$ by substituting $\mathbf{B}_{\theta}\Delta_{\tilde{\theta}} \leftarrow \Delta_{\theta}$ into the normal equations. The reduced normal equations are as follows:

$$\begin{bmatrix} \mathbf{B}_{\theta} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^{\top} \mathbf{A}_c^{\top} \mathbf{A}_c \begin{bmatrix} \mathbf{B}_{\theta} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \text{stk}(\Delta_{\tilde{\theta}}, \Delta f) = \begin{bmatrix} \mathbf{B}_{\theta} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^{\top} \mathbf{A}_c^{\top} \mathbf{b}_c \quad (5.32)$$

This has $L + 1$ unknowns and it is a dense linear system because \mathbf{B}_{θ} is dense. It can be solved very efficiently with Cholesky factorization (we use Eigen’s LDLT CPU implementation). On our benchmark computer this requires approximately 2ms. Once solved we recover Δ_{θ} with $\Delta_{\theta} = \mathbf{B}_{\theta}\Delta_{\tilde{\theta}}$.

The second technique to reduce computation time is to compute the matrix products in Equation (5.32) on an available GPU. The main bottleneck is transferring the matrices to the GPU device. Note however that \mathbf{B}_{θ} (the only dense matrix) is constant so it only needs to be transferred once. The matrix \mathbf{A}_c is highly sparse and can be transferred very quickly using a compact representation such as Compressed Sparse Row (CSR). Thanks to the GPU we can reduce the time to assemble the reduced normal equations by an order of magnitude, taking only a few milliseconds on the benchmark computer including transfer back to CPU memory for solving with Cholesky factorization.

5.2.5 Initialization

5.2.5.1 Initialization policies

The initialization set $\{(f_1, \theta_1), \dots, (f_S, \theta_S)\}$ is created with an *initialization policy*. We define a policy using lens opening angles because unlike focal lengths, they are invariant to image resolution. We convert between a focal length f and a lens opening angle ψ with the following formula:

$$\tan\left(\frac{\psi}{2}\right) = \frac{s}{2f}, \quad s \stackrel{\text{def}}{=} \max(w, h) \quad (5.33)$$

where w and h denote the image width and height in pixels respectively.

An initialization policy has two components: a set of lens opening angles Ψ_{init} and a set of SFT-P methods \mathcal{M} , used to generate the initial deformation estimate for each member of Ψ_{init} . We test two SFT-P methods: the convex relaxation using the *Maximum Depth Heuristic* (referred to as MDH) [SHF07; PHB11] and PnP (referred to as PnP). We test two ways to create Ψ_{init} : (i) the solution from the analytical method from §5.1 (with Ψ_{init} of size 1), and (ii) focal length sampling (with Ψ_{init} consisting of one or more fixed values). We have already discussed in §5.1 that the analytical solution is a local method (requiring motion to be estimated at local surface regions), which can work well when motion information is rich (*i.e.* dense texture) but it may have difficulties when texture is sparse. Focal length sampling does not suffer these limits, but it comes at the price of potentially higher computational from the larger initialization set size, and therefore more iterative optimization in Algorithm 8. Recall that this is partially mitigated using the early stopping criteria in Algorithm 8, designed to terminate early if we re-explore the same region of parameter space from different initializations.

We test different initialization policies in the experimental evaluation, to compare the analytical solution with focal length sampling, and to assess the impact of sample density on accuracy. An example initialization policy is $(\Psi_{init} = \{20, 50, 80\}, \mathcal{M} = \{MDH, PnP\})$, generating an initialization set of size $S = 6$ (3 generated with MDH using opening angles $\{20, 50, 80\}$ and 3 generated with PnP).

A policy defined as $(\Psi_{init} = \{20, 30, 40, 50, 60, 70, 80, 90\}, \mathcal{M} = \{MDH\})$ generates an initialization set of size $S = 8$ using only the MDH method.

5.2.5.2 Sampling range

In real-world applications, lens opening angle/focal length is limited by two factors: (i) the physical limits of camera hardware, and (ii) theoretical limits and well-posedness of our problem. Concerning (i), the distribution of real lens opening angles is mono-modal and it has been studied in large-scale image datasets [SSP14]. The results are reproduced in Figure 5.4. The mode of ψ is approximately 50° and the maximum is approximately 100° , equivalent to a very wide-angle focal length of approximately $f = \frac{1}{2}s$ px. This wide-angle limit sets a practical lower bound on focal length in real-world applications. By contrast, (ii) imposes a lower bound on ψ in practice. A smaller opening angle reduces the field-of-view which in turn reduces the amount of depth variation observable in the image. However, depth variation is essential to solve f . Consequently, SfT will not be solvable in real-world cases if the opening angle is very small. Consequently, we restrict the range of opening angle samples to $20^\circ \leq \psi \leq 100^\circ$ where 20° . We note that this range is more than sufficient to cover all public datasets that have been used to test SfT-P methods.

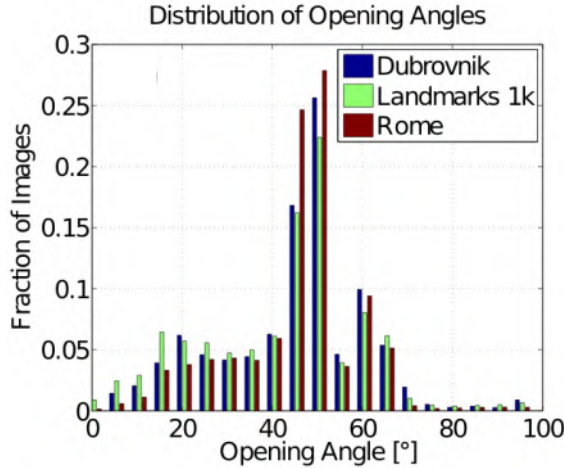


Figure 5.4: Distribution of lens opening angles for images in the Dubrovnik [LSH10], Landmarks 1k [Li+12], and Rome [LSH10] datasets. This figure is taken from [SSP14].

5.2.5.3 SfT-P method implementation

MDH. MDH is considered one of the best closed-form SfT-P methods. Given an initial opening angle ψ_s with focal length f_s , and the set of N point correspondences \mathcal{P} and \mathcal{Q} , we compute θ_s in two stages as follows. In the first stage we reconstruct the depths $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$ of \mathcal{Q} by solving a convex relaxation of SfT-P following [SHF07]. Specifically, we maximize the depth of each point such that the Euclidean distance e_{ij} between any two points $(i, j) \in [1, N]^2$ does not exceed their geodesic distance d_{ij} . The geodesic distances are known *a priori* from the template and they can be computed efficiently using *e.g.* Fast Marching [KS98]. The following SOCP problem is then solved:

$$\begin{aligned} & \max \sum_{i=1}^N z_i \quad \text{s.t.} \\ & \forall (i, j) \in \mathcal{N} \quad \left\| \frac{z_i}{f_s} \text{stk}(\mathcal{Q}(i), f_s) - \frac{z_j}{f_s} \text{stk}(\mathcal{Q}(j), f_s) \right\|_2 \leq d_{ij} \end{aligned} \quad (5.34)$$

The set \mathcal{N} defines pairs of point correspondences, constructed with a K-nearest neighbor (KNN) graph with a default of $\min(N, 15)$ neighbors. We solve Equation (5.34) quickly using the interior point method from Mosek [MOS19]. When N is not large ($N \leq 500$), this typically takes between 100 and 500 ms on the benchmark computer. If $N > 500$ we reduce the problem size by randomly sub-sampling 500 correspondences without replacement using furthest point sampling, and we ignore the remaining points. This normally has little effect on reconstruction accuracy.

In the second stage we compute θ_s from \mathcal{Z} and f_s . We solve a regularized linear least squares system that finds a smooth 3D deformation of the template mesh that fits the reconstructed point correspondences in camera coordinates. This problem is as follows:

$$\begin{aligned} \theta_s &= \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \left\| g(\mathcal{P}(i); \theta) - \frac{z_i}{f} \text{stk}(Q(i), f_s) \right\|_2^2 + \lambda c_{reg}(\theta) & (a) \\ &\Leftrightarrow \arg \min_{\theta} \|\mathbf{A}_{mdh}\theta - \mathbf{b}_{mdh}\|_2^2 & (b) \end{aligned} \quad (5.35)$$

Where λ is a regularization weight. Equation (5.35-b) is equivalent to Equation (5.35-a) where we have rearranged the problem to a standard LLS format with known terms \mathbf{A}_{mdh} and \mathbf{b}_{mdh} . The matrix \mathbf{A}_{mdh} does not depend on f_s nor \mathcal{Z} . We exploit this by solving Equation (5.35) with a factorization of \mathbf{A}_{mdh} . Importantly, the factorization can be done once and be reused for any f_s or \mathcal{Z} . We weight c_{reg} using the normalization technique described in §5.2.3.1, and we use the same λ for all problem instances (we use a default of $\lambda = 100$ in all experiments).

The factorization can be solved very quickly when the number of mesh vertices V is small (a few hundred) using sparse Cholesky factorization. However for larger meshes it become unreasonably expensive. We deal with this by applying dimensionality reduction by eliminating high-frequency deformation components from the problem. We implement this with linear bases as described in §5.2.4.4. We then solve Equation (5.35-b) for $\tilde{\theta}$ with the substitution $\mathbf{B}_{\theta}\tilde{\theta} \leftarrow \theta$. This reduces the problem to a smaller dense linear system that is solved efficiently with Cholesky factorization (we use Eigen’s LDLT implementation).

PnP. PnP estimates only the rigid pose of the template in camera coordinates given a focal length sample f_s and the point correspondences \mathcal{P} and \mathcal{Q} . When \mathcal{P} is co-planar we use IPPE (Chapter 4, otherwise we use OpenCV’s SolvePnP method. We then compute θ_s by rigidly transforming the mesh vertices to camera coordinates using the estimated pose. Despite not estimating deformation, we find that SfT-P-PnP is a surprisingly effective and fast initialization method for fSfT.

5.3 Multi-view fSfT

5.3.1 Overview

In this section we extend the work of the previous section to multiple images. We assume the true focal length f_{true} is constant and unknown in a set of M images, which is a common use case when using a video camera with either a fixed lens or when the camera operator does not change the zoom. The images can be acquired either from a set of static shots of the deformable object, or they can be selected from a video using a set of keyframes. We refer to the problem of estimating f_{true} and the M template deformations (one per image) as *Multi-view fSfT*. We explore two approaches for Multi-view fSfT: The first, called *Robust fSfT averaging*, uses robust focal length averaging from M focal length estimates computed for each view independently. The second approach, called *Multi-view*

fSfT optimization, is the multi-view extension of the optimization approach in §5.2, where focal length and all deformation parameters are optimized jointly as a single large-scale system. We show that Multi-view fSfT optimization can be solved efficiently with super-linear convergence by exploiting the problem’s sparsity pattern, with computational cost that is linear in the number of views. It therefore has the same computational cost as fSfT averaging. It also generalizes straightforwardly to *Multi-view USfT optimization*, where unknown intrinsics include other terms such as lens distortion and aspect ratio.

The approaches have both advantages and disadvantages. Both fSfT averaging and Multi-view fSfT optimization improve the estimation of focal length compared to using a single view, particularly when noise is high. Performance is generally better with Multi-view fSfT optimization compared to non-robust fSfT averaging (taking the mean focal length). This is because it links all physical constraints across images into the optimization problem. By contrast, fSfT averaging relaxes constraints because focal length is computed independently for each image. However, Multi-view fSfT optimization can be difficult because it can become trapped in a local minimum if the initialization is very poor in one or more images. By contrast, this can be handled by robust fSfT averaging.

5.3.2 Robust focal length averaging

We denote as $\mathcal{M} \stackrel{def}{=} \{(\hat{f}_1, \hat{\theta}_1), \dots, (\hat{f}_M, \hat{\theta}_M)\}$ the set of M fSfT solutions computed independently from M images. Assuming that each $\hat{f}_{i \in [1, M]}$ deviates from f_{true} by zero-mean I.I.D noise, the statistically optimal combined estimate is the mean: $\hat{f} = \frac{1}{M} \sum_{j=1}^M \hat{f}_j$. In reality I.I.D noise cannot be assumed because of outliers, caused by two problems. The first problem is when conditions of one or more views are quasi-degenerate, see §5.1.7, leading to extreme errors that invalidate the I.I.D assumption. The second problem is if one or more solutions in \mathcal{M} are at incorrect local minimum, which can also lead to extreme errors that invalidate the I.I.D assumption. We counter these issues with robust averaging. This can be implemented in several ways and we test two standard approaches. The first is median averaging, where the median focal length in \mathcal{M} is used. This is robust when up to half of the estimates in \mathcal{M} are outliers. The second approach uses Exhaustive Sampling and Consensus (ESAC). For each image $i \in [1, M]$ we compute a consensus set \mathcal{F}_i that contains all members of $\{\hat{f}_1, \dots, \hat{f}_M\}$ that are within $\tau_f\%$ of \hat{f}_i , where τ_f defines the inlier threshold. The index l for which \mathcal{F}_l is largest then taken and the final estimate \hat{f} is the average of the elements in \mathcal{F}_l . We use as default $\tau_f = 15\%$.

5.3.3 Multi-view fSfT optimization

5.3.3.1 Generalized solution

In multi-view fSfT optimization, we exploit information from multiple views to reduce sensitivity to noise and to avoid cases where single-view fSfT is quasi-degenerate. We describe this approach with a generalized cost function that can be expressed as the summation of M-estimators. In practice this covers most cost functions of interest in SfT. We also generalize over the unknown common intrinsics (focal length, lens distortion, aspect ratio *etc*). We then show that the generalized cost function can be optimized efficiently with super-linear convergence using the Shur complement. Finally, we specialize it to Multi-view fSfT, based on the cost function used by us for single-view fSfT.

We use the parameter vector \mathbf{h} to denote the unknown intrinsics. We stack the unknowns into a

vector \mathbf{y} that consists of $M + 1$ blocks:

$$\mathbf{y} \stackrel{\text{def}}{=} \text{stk}(\theta_1, \theta_2, \dots, \theta_M, \mathbf{h}) \quad (5.36)$$

where $\theta_{j \in [1, M]}$ is the deformation for the j^{th} image. We define as $c_{mv}(\mathbf{y}) : \mathbb{R}^{|\mathbf{h}| \times |\theta_1| \times \dots \times |\theta_M|} \rightarrow \mathbb{R}^+$ the multi-view cost function. We assume this takes the following general form:

$$c_{mv}(\mathbf{y}) = \sum_{j=1}^M c_{def}^j(\theta_j) + \sum_{j=1}^M c_{data}^j(\mathbf{h}, \theta_j) + c_{cam}(\mathbf{h}) \quad (5.37)$$

where c_{def}^j the total deformation cost associated to the j^{th} view, c_{data}^j is the total data cost associated to the j^{th} view, and c_{cam} is an optional cost that encodes priors on \mathbf{h} . The cost function c that we have used to solve single image fSfT (Equation (5.16)) is a special case of c_{mv} with the instantiations $M \leftarrow 1$, $\mathbf{h} \leftarrow f$, $c_{def}^1 \leftarrow \lambda_{iso} c_{iso} + \lambda_{reg} c_{reg}$, $c_{data}^1 \leftarrow c_{data}$ and $c_{cam} \leftarrow 0$.

The unknown deformation and intrinsic variables are linked by the multi-view cost function, leading to a potentially very large scale optimization problem. This situation bares some similarity with NRSfM and SfM where all unknown variables are also linked across views. In contrast, SfT methods typically solve the unknowns for each view individually, which allows the SfT method to scale trivially to an arbitrary number of views. Some SfT methods also impose temporal smoothness assumptions, which connects deformation over time and is applicable only for video data. However, this is usually implemented either by frame-to-frame initialization (the solution from the previous frame is used as an initial solution from the next frame), and/or by introducing a cost that penalizes the solution at frame t if it differs substantially from time $t - 1$ [Yu+15; CB15]. These approaches maintain a constant number of unknowns at every frame, making them computationally efficient. However, they also have limitations. Firstly, they break down if the solution at frame t is incorrect because of *e.g.* strong occlusions or a convergence failure. Secondly, they may fail with sudden camera motion or deformation. Thirdly, we assume that the unknown intrinsics do not change across the image set, so they are not connected by a time-varying function.

5.3.4 Efficient optimization with the Schur complement

We now expose the sparsity pattern of c_{mv} . We define as \mathbf{r}_{def}^j , \mathbf{r}_{data}^j and \mathbf{r}_{cam} the vector of residuals of c_{def}^j , c_{data}^j and c_{cam} respectively. The full residual vector \mathbf{r}_{mv} and associated Jacobian matrix \mathbf{J}_{mv} have the following pattern:

$$\mathbf{r}_{mv} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{r}_{def}^1 \\ \vdots \\ \mathbf{r}_{def}^M \\ \mathbf{r}_{data}^1 \\ \vdots \\ \mathbf{r}_{data}^M \\ \mathbf{r}_{cam} \end{bmatrix}, \quad \mathbf{J}_{mv} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{J}_{def}^1(\theta_1) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{def}^M(\theta_M) & \mathbf{0} \\ \mathbf{J}_{data}^1(\theta_1) & \mathbf{0} & \mathbf{0} & \mathbf{J}_{data}^1(\mathbf{h}) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{data}^M(\theta_M) & \mathbf{J}_{data}^M(\mathbf{h}) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{cam}(\mathbf{h}) \end{bmatrix} \quad (5.38)$$

where \mathbf{r}_Y^j is the residual vector of cost term Y for view j and $\mathbf{J}_Y^j(X)$ is the corresponding Jacobian matrix with respect to unknowns X . To optimize c_{mv} with super-linear convergence using either Gauss-Newton or Levenberg-Marquardt, c_{mv} is linearized and the update vector $\Delta_{\mathbf{y}} \stackrel{\text{def}}{=}$

$\text{stk}(\Delta_{\theta_1}, \dots, \Delta_{\theta_M}, \Delta_h)$ is solved with *e.g.* the normal equations. The linear system is as follows:

$$\begin{aligned} \mathbf{M}_{mv} \Delta_{\mathbf{y}} &= \mathbf{b}_{mv} & (a) \\ \mathbf{M}_{mv} &\stackrel{\text{def}}{=} \mathbf{J}_{mv}^\top \text{diag}(\mathbf{w}_{mv})^2 \mathbf{J}_{mv} + \lambda \mathbf{I} & (b) \\ \mathbf{b}_{mv} &\stackrel{\text{def}}{=} -\mathbf{J}_{mv}^\top \text{diag}(\mathbf{w}_{mv})^2 \mathbf{r}_{mv} & (c) \end{aligned} \quad (5.39)$$

where \mathbf{w}_{mv} is the IRWLLS reweighing vector and λ is the Levenberg-Marquardt damping parameter (for Gauss-Newton $\lambda = 0$). This is a very large system with the number of unknowns growing linearly in M . In general we cannot solve it efficiently with an off-the-shelf linear solver even for small M . However, thanks to its sparsity pattern we can solve it efficient using the Schur complement as follows.

The matrix \mathbf{M}_{mv} decomposes into 4 blocks as follows:

$$\mathbf{M}_{mv} = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{D} \end{bmatrix} \quad (a)$$

$$\mathbf{A} \stackrel{\text{def}}{=} \begin{bmatrix} G\left(\tilde{\mathbf{J}}_{data}^1(\theta_M)\right) + G\left(\tilde{\mathbf{J}}_{def}^1(\theta_M)\right) + \lambda \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & G\left(\tilde{\mathbf{J}}_{data}^M(\theta_M)\right) + G\left(\tilde{\mathbf{J}}_{def}^M(\theta_M)\right) + \lambda \mathbf{I} \end{bmatrix} \quad (b)$$

$$\mathbf{C} \stackrel{\text{def}}{=} \sum_{j=1}^M \tilde{\mathbf{J}}_{data}^{j\top}(\theta_j) \tilde{\mathbf{J}}_{def}^j(\theta_j) \quad (c)$$

$$\mathbf{D} \stackrel{\text{def}}{=} \sum_{j=1}^M G\left(\tilde{\mathbf{J}}_{data}(\mathbf{h})\right) + G\left(\tilde{\mathbf{J}}_{cam}(\mathbf{h})\right) + \lambda \mathbf{I} \quad (d)$$

where $\tilde{\mathbf{J}}_Y^j$ is the element-wise re-weighted version of \mathbf{J}_Y^j using the weights in \mathbf{w}_{mv} . The matrix \mathbf{A} is a large $T \times T$ block-diagonal matrix where T is the total number of unknown deformation parameters for all views. By contrast, the matrix \mathbf{D} is a small $H \times H$ matrix where H is the length of \mathbf{h} (the number of unknown camera intrinsic terms).

We now solve $\Delta_{\mathbf{h}}$ using the Schur complement. This is the solution to the following small-scale exact linear system of H unknowns:

$$(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^\top) \Delta_{\mathbf{h}} = \mathbf{b} - \mathbf{C}\mathbf{A}^{-1}\mathbf{b} \quad (5.41)$$

First, we compute the matrix $\mathbf{A}^{-1}\mathbf{C}^\top$ and vector $\mathbf{A}^{-1}\mathbf{b}$. Then the left and right sides of Equation (5.41) are assembled and $\Delta_{\mathbf{h}}$ is solved with any good direct dense linear solver. Recall that \mathbf{A} is block-diagonal from Equation (5.40), therefore we can compute these terms block-wise and efficiently:

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{C}^\top &= \begin{bmatrix} \mathbf{A}_1^{-1}\mathbf{C}_1^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_M^{-1}\mathbf{C}_M^\top \end{bmatrix} & (a) \\ \mathbf{A}^{-1}\mathbf{b}^\top &= \begin{bmatrix} \mathbf{A}_1^{-1}\mathbf{b}_1^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_M^{-1}\mathbf{b}_M^\top \end{bmatrix} & (b) \end{aligned} \quad (5.42)$$

where \mathbf{A}_j denotes the j^{th} diagonal block of \mathbf{A} , \mathbf{C}_j denotes j^{th} row block of \mathbf{C} and \mathbf{b}_j denotes the j^{th}

row block of \mathbf{b} . We compute $\mathbf{A}_j^{-1}\mathbf{C}_j^\top$ and $\mathbf{A}_j^{-1}\mathbf{b}_j$ using an appropriate factorization-based solver such as Cholesky factorization. This is efficient because the factorization needs to be done only once and it is used to solve both $\mathbf{A}_j^{-1}\mathbf{C}_j^\top$ and $\mathbf{A}_j^{-1}\mathbf{b}_j$ with back substitution. Consequently, the computational cost of assembling Equation (5.41) and solving $\Delta_{\mathbf{h}}$ is linear in M .

We then substitute $\Delta_{\mathbf{h}}$ into Equation (5.39) and we solve for the remaining unknowns $(\Delta_{\theta_1}, \Delta_{\theta_2}, \dots, \Delta_{\theta_M})$. They are the solutions to the following linear system:

$$\mathbf{A}(\Delta_{\theta_1}, \Delta_{\theta_2}, \dots, \Delta_{\theta_M}) = (\mathbf{b} - \mathbf{C}^\top \Delta_{\mathbf{h}}) \quad (5.43)$$

Using the block-diagonal structure of \mathbf{A} , this is solved efficiently as follows:

$$\Delta_{\theta_{j \in [1, M]}} = \mathbf{A}_j^{-1}\mathbf{b}_j - \mathbf{A}_j^{-1}\mathbf{C}_j\Delta_{\mathbf{h}} \quad (5.44)$$

Recall that we have already computed the terms $\mathbf{A}_j^{-1}\mathbf{C}_j$ and $\mathbf{A}_j^{-1}\mathbf{b}_j$, so the additional computational cost to solve Δ_{θ_j} is negligible.

5.3.5 Instantiation for multi-view fSfT

We have shown that Multi-view USfT can be optimized with super-linear convergence using Gauss-Newton or Levenberg-Marquardt, with a computational cost that is linear in M . We have instantiated and tested it in the special case of fSfT. We use exactly the same cost function modeling as described in the single-view case in §5.2.2.1. The cost c_{mv} is constructed by summing the single image costs as described in §5.2.2.2 over each image. We iteratively optimize c_{mv} with Gauss-Newton and backtracking line-search until convergence is detected. We initialize the system by solving fSfT for each view independently as described in §5.2. To reduce computational cost, we apply the same dimensionality reduction technique as described in §5.2.4.4 using deformation bases and the substitutions $\Delta_{\theta_j} \leftarrow \mathbf{B}\Delta_{\theta'_j}$ (recall that \mathbf{B} is the linear basis matrix and θ'_j is the unknown basis coefficients for image j). This significantly reduce the problem size and the cost of solving the linear systems. The computational cost of one multi-view Gauss-Newton iteration is practically the same as the cost of M Gauss-Newton iterations in the single-view case.

5.4 Experimental results

5.4.1 Overview

We evaluate our work with 12 public datasets that have been used to test state-of-the-art SfT methods. The datasets include video sequences and unorganized image collections of deformable man-made objects. This is a relatively large evaluation, and it significantly expands our evaluation from the conference papers [BPC13; BC13a]. The datasets have variation in the main aspects: object geometry, material, deformation, viewpoint, number of images, amount of surface texture, number of point correspondences, focal length and image resolution. We now describe the datasets. A summary of the relevant dataset statistics such as template properties and number of correspondences is provided in Tables 5.3, 5.4 and 5.5.

5.4.2 Dataset descriptions

5.4.2.1 Cap dataset [BC13b]

The Cap dataset (Figure 5.5, Table 5.3) has 15 unorganized RGB images of a baseball cap that is deformed significantly at the cap’s crown region. The provided template is built with SfM and MVS from a set of reference images of the cap in a natural rest position. The top row of Figure 5.5 shows the template mesh rendered from two viewpoints with and without texture. The bottom two rows of Figure 5.5 shows 8 representative images of the deformed cap. In the right-most images we overlay point correspondences as small red points. The images are taken from significantly different viewpoints with fixed camera intrinsics. This dataset is challenging mainly because of the strong complex deformation. This dataset does not have provided point correspondences so we computed them manually as follows. Using the images of the deformed cap, we reconstructed the deformed cap’s surface with SfM and MVS (Photoscan). We then interactively matched 266 surface points between the template and the reconstructed deformed cap’s surface using Mashlab. Finally we projected the matched points on the deformed cap’s surface into each image to generate the point correspondences².

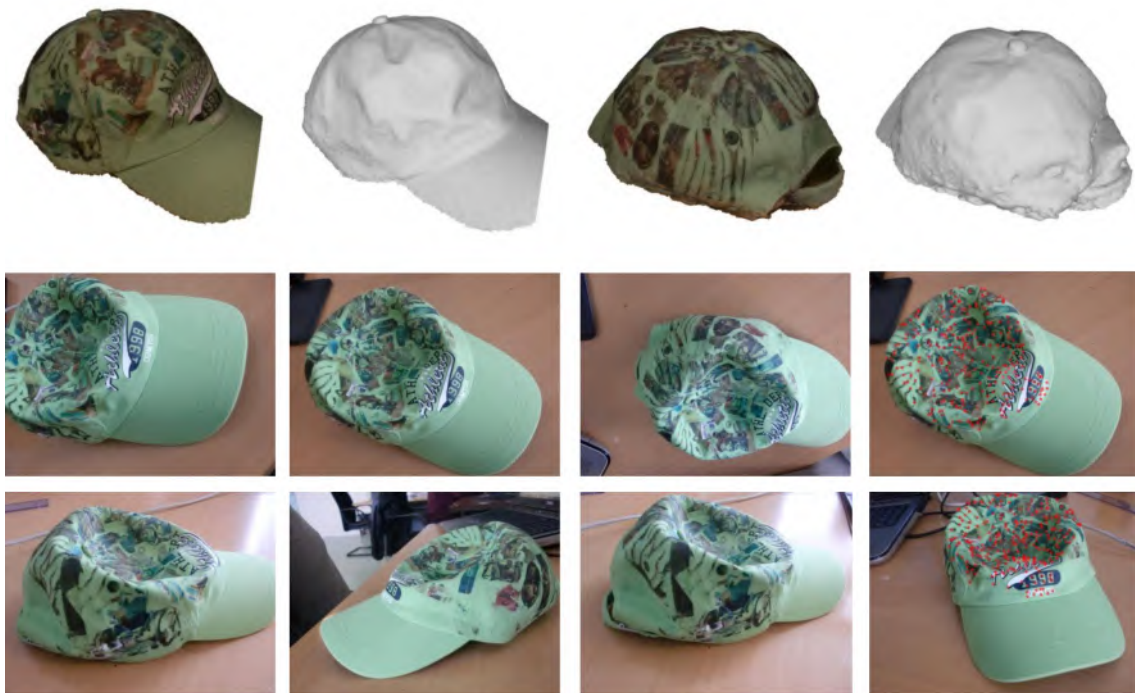


Figure 5.5: The Cap dataset [BC13b]

5.4.2.2 Spider-man dataset [Chh+17b]

The Spider-man dataset (Figure 5.6, Table 5.3) has 79 unorganized RGB images of a deforming paper magazine cover captured at 9 different zoom levels. The zoom increases by approximately a factor of 3 from level 1 to level 9. The object is imaged from significantly different viewpoints with strong deformations. We show in the left column of Figure 5.5 the template with and without texture. This

²The point correspondences are useful for other researchers using the cap dataset, so we have added them to the public dataset at http://igt.ip.uca.fr/~ab/code_and_datasets

has been constructed from a structured light scan followed by an isometric flattening of the scan onto the 2D plane. Note that the template does not cover the full surface. It has been cropped to the textured region, so its borders do not physically correspond with the borders of the object.

We show in Figure 5.6, columns 2, 3 and 4 two representative images from zoom levels 1, 6 and 9 respectively. In column 4 we overlay point correspondences as red points. We give details of the zoom levels in Table 5.2. The main challenge of this dataset is that the perspective effects diminish significantly at higher zoom levels. A dense registration is provided between the template and each image. Point correspondences are computed by extracting SIFT features in each image, and their positions on the template’s surface are determined from the dense registration. There are 1176 points per image on average with standard deviation of 468.



Figure 5.6: The Spider-man dataset [Chh+17b]

Zoom level	1	2	3	4	5	6	7	8	9
Focal length (px)	1348	1551	1808	2069	2345	2702	3131	3543	3938
Opening angles (°)	65.3	58.2	51.1	45.3	40.4	35.5	30.9	27.4	24.8

Table 5.2: Focal lengths of each of the 9 zoom levels used in the Hulk dataset.

5.4.2.3 Hulk dataset [CPB14b]

The Hulk dataset (Figure 5.7, Table 5.4) has 21 unorganized RGB images of a textured paper magazine cover. The object is captured from different viewpoints and deformations with fixed camera intrinsics, with a relatively short focal length (opening angle 66.1°). The template has 122 regularly positioned 3D vertices that are computed from the first view using a stereo camera. Point correspondences are provided for each vertex in each image. Similarly to the Spider-man template, this template does not cover the full surface and is cropped to the textured surface region, explaining the irregular border.

In Figure 5.7, top-left we show the template (only a textureless mesh is provide in this dataset). We also show 7 representative images from the Hulk dataset in Figure 5.7. This is arguably the easiest of the datasets for fSfT because of the strong perspective effects, due to the relatively short focal length, the object being smooth and there being regularly spaced point correspondences across the whole surface.

5.4.2.4 Handbag and Pillow-cover datasets [GCB17]

The Handbag dataset (Figure 5.8, Table 5.3) has 7 unorganized RGB images of a sparsely-textured handbag. There are 150 point correspondences per image. The provided template is built by scanning



Figure 5.7: The Hulk dataset [CPB14b]

object’s surface in a non-flat rest position from a single viewpoint with a structured light scanner. In Figure 5.8 we show the dataset (rows 1 and 2). The left-most images show the template with texture, and without texture with the mesh edges overlaid. In the other images in rows 1 and 2 we show 6 representative images from the Handbag dataset. In the right-most images we overlay the point correspondences as red points.

The Pillow-cover dataset is constructed in a similar way as the Handbag dataset and it has 9 unorganized RGB images of a pillow cover. There are 63 point correspondences per image. We illustrate the dataset similarly as the Handbag dataset in Figure 5.8 (rows 3 and 4). The main challenges of these datasets are poor texture, producing sparse point correspondences, and complex non-isometric deformation due to the material (fabric). The pillow cover has several views that are approximately fronto-parallel, making fSfT poorly conditioned.

5.4.2.5 Floral paper dataset and Fortune teller dataset [GCB17]

The Floral dataset (Figure 5.9 rows 1 and 2, Table 5.4) has 13 unorganized RGB images of a deforming and folding A4 sheet of paper. There are 18 point correspondences per image. The provided template is built from one held-out image using a structured light scanner. The Fortune teller dataset (Figure 5.9 rows 3 and 4, Table 5.4) is constructed in a similar way, and it has 6 unorganized RGB images of a deforming sheet of paper folded into a fortune teller game. There are 20 point correspondences per image. The templates are shown in the left-most column of Figure 5.9 and the remaining images are representative images from the datasets. The right-most images are overlaid with points correspondences in red. The main challenges of these datasets are very sparse texture and severely creased regions. Motion information from the sparse points are not sufficient to resolve the creases accurately.

5.4.2.6 Video datasets

The video datasets are Bedsheet [SF09], Bending cardboard [SUF08], Kinect paper [Var+12], Kinect t-shirt [Var+12] and Van Gogh paper [Sal+08]. These each consist of a video sequence of a deforming object and a corresponding flat template (Figure 5.10, Tables 5.4 and 5.5). The provided templates (Figure 5.10, row 1) have been constructed using the first image of each video, where the surface is quasi-flat. Regular triangulated meshes have been used to construct the templates from the first image using a rectangular region-of-interest that covers the textured region of the surface. In Figure 5.10, rows 2-7 we show representative images from each dataset underneath each template. In row 7 we overlay the images with point correspondences in red. The frames in each video are highly

5.4. EXPERIMENTAL RESULTS

	Cap	Handbag	Pillow-cover	Spider-man
Object material	Fabric	Fabric	Fabric	Paper
Template geometry	3D open	3D open	3D open	Flat open
Number of template vertices (V)	4854	1098	1368	2918
Number of template triangles (T)	9502	2063	2587	5000
Video (vid.) or image collection (col.)	col.	col.	col.	col.
Number of images (M)	15	7	9	79
Image resolution ($w \times h$)	2048 \times 1536	1280 \times 960	1280 \times 960	1728 \times 1152
Correspondences per image (N)	266	150	63	1176 \pm 468
Focal length (px)	2039	1344.0	1344.0	1348.4 \rightarrow 3937.9
Focal length (% of w)	99.6	105.0	105.0	78.0 \rightarrow 277.9
Lens opening angle ($^\circ$)	53.3	50.9	50.9	65.3 \rightarrow 24.8
Has ground truth 3D	Yes	Yes	Yes	Yes

Table 5.3: Cap, Pillow-cover, Handbag and Spider-man dataset statistics.

	Floral paper	Fortune teller	Hulk	Bending cardboard
Object material	Paper	Paper	Foam	Cardboard
Template geometry	3D open	3D open	Flat open	Flat open
Number of template vertices (V)	1248	936	122	609
Number of template triangles (T)	2342	1747	200	1120
Video (vid.) or image collection (col.)	col.	col.	col.	vid.
Number of images (M)	13	6	20	18 (87)
Image resolution ($w \times h$)	1280 \times 960	1280 \times 960	4928 \times 3264	720 \times 576
Correspondences per image (N)	18	20	20	52
Focal length (px)	1344.0	1344.0 \rightarrow 3937.9	3784.9	879.6
Focal length (% of w)	105.0	105.0 \rightarrow 277.9	76.8	122.2
Lens opening angle ($^\circ$)	50.9	50.9	66.1	44.5
Has ground truth 3D	Yes	Yes	Yes	No

Table 5.4: Floral paper, Fortune teller, Hulk and Bending cardboard dataset statistics.

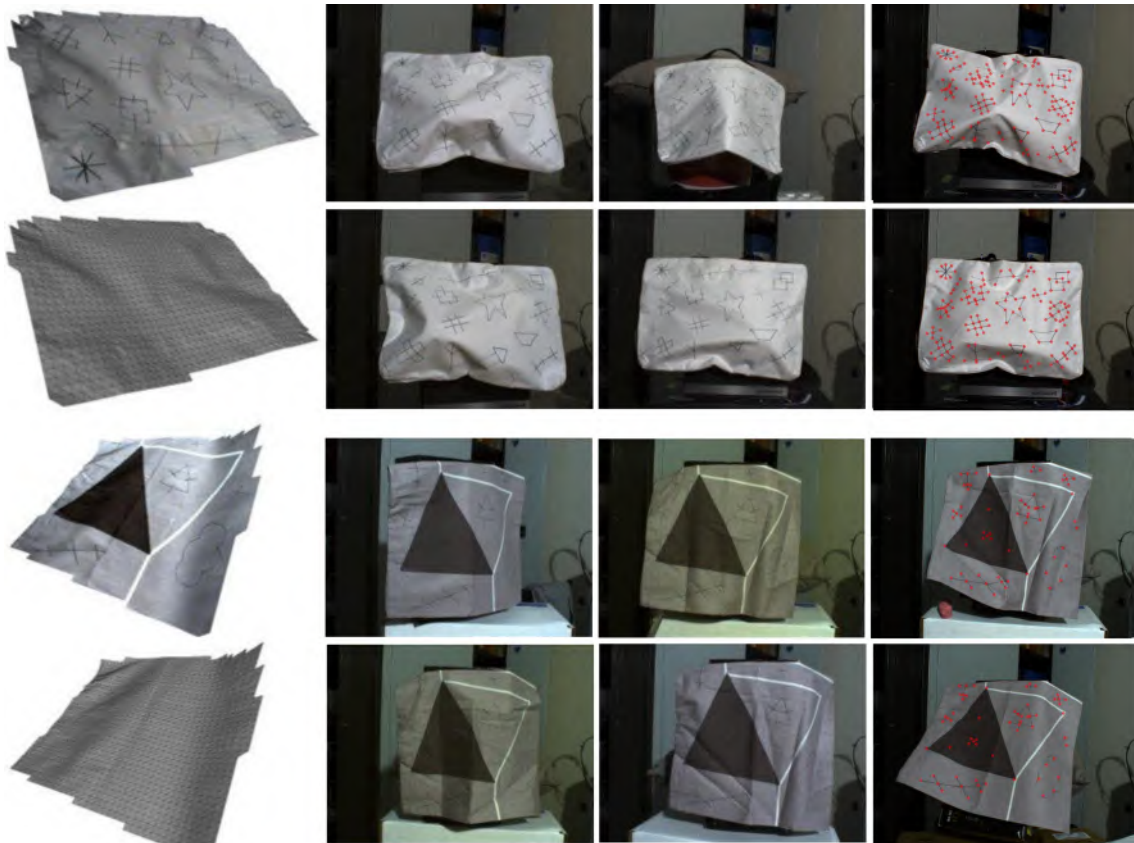


Figure 5.8: The Handbag dataset [GCB17] (rows 1 and 2) and the Pillow-cover dataset [GCB17] (rows 3 and 4).

correlated over time thanks to smooth object motion. It is therefore redundant to evaluate using every video frame. We evaluate using one image taken every 5 frames (approximately 6 frames per second). Furthermore, fSfT is not solvable using the first few frames of each video dataset because the template is flat and quasi-fronto parallel. These frames are excluded from evaluation if the template's tilt angle is below 5° . We show the number of images used per dataset in Tables 5.4 and 5.5, and the original number in brackets.

The video datasets vary in difficulty. The simplest are ones of deforming paper sheets: Kinect paper and Van Gogh paper, because they have dense texture, deformation is smooth and the objects are paper sheets and thus strongly isometric. Bending cardboard is very challenging because of very sparse texture. Bedsheet and Kinect t-shirt datasets are challenging because their fabric allows significant stretching. For all datasets there are some frames where the object is approximately fronto-parallel, so fSfT with those frames is poorly-conditioned.

The video datasets do not have provided correspondences. We computed them with a classical frame-to-frame tracker (KLT [TK91] implemented in OpenCV). Outliers were detected and removed with cross-checking. Specifically, we tracked points from the first to last images in the video, then we back-tracked them from the last to first image. A track was detected as an outlier and removed if the start and back-tracked positions differed by more than a threshold (5 pixels). KLT tracking with cross-checking works remarkably well with these datasets because the images are relatively sharp and the deformation between video frames is not very fast.

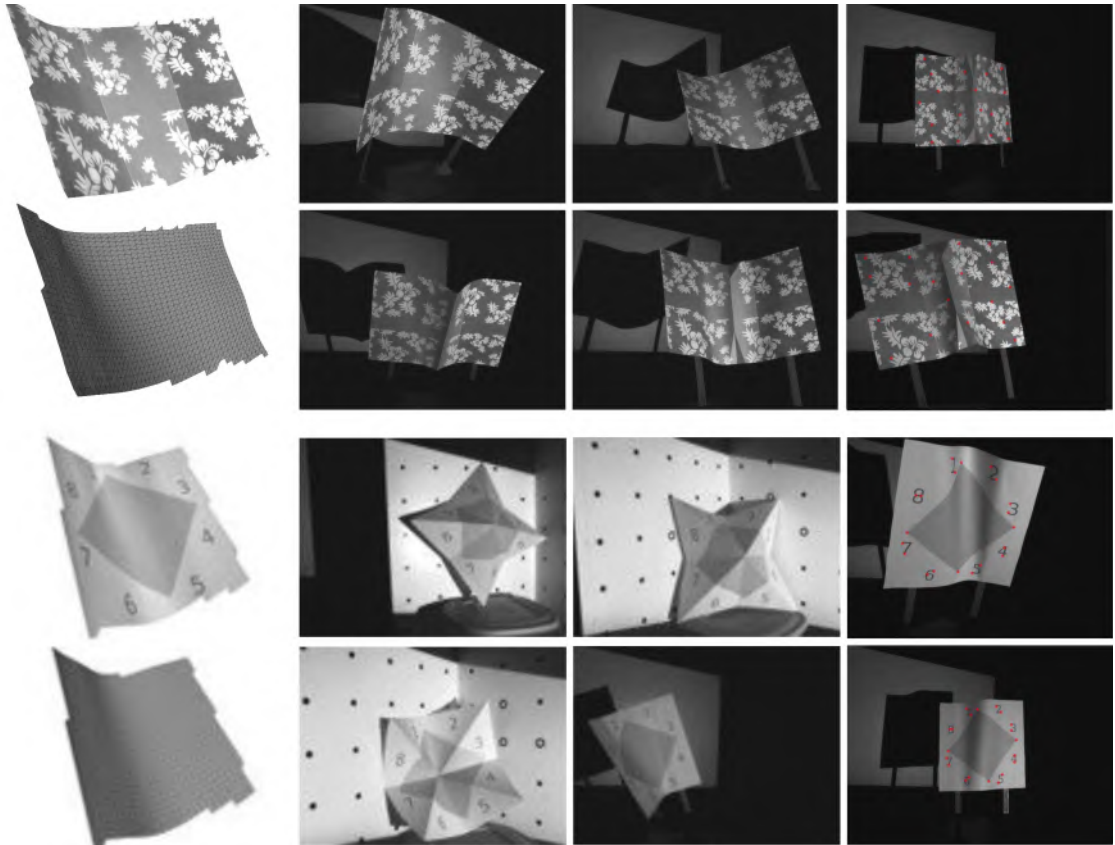


Figure 5.9: Floral paper and Fortune teller datasets [GCB17].

5.4.3 Evaluation metrics

5.4.3.1 Focal Length Percentage error (FLPE) and Shape Error (SE)

Focal length error is evaluated with Focal Length Percentage Error (FLPE) given in Equation (5.28). Deformation error is computed for all datasets with ground truth 3D as follows. For each image and each point correspondence, we evaluate the Euclidean distance between the reconstructed 3D point in camera coordinates $\hat{\mathbf{q}} \in \mathbb{R}^3$ and ground truth $\mathbf{q}^{gt} \in \mathbb{R}^3$. The *Deformation Error* (DE) is defined as follows:

$$DE(\hat{\mathbf{q}}, \mathbf{q}^{gt}) \stackrel{\text{def}}{=} \frac{100}{S} \times \|\hat{\mathbf{q}} - \mathbf{q}^{gt}\|_2 \quad (5.45)$$

A scale factor S is used to make DE independent of the template’s scale. We set this as the maximum spatial range of the template’s rest shape with respect to its 3 spatial coordinates. Consequently a DE of 1 corresponds to approximately 1% of the template’s size.

DE has been used extensively to evaluate the accuracy of SfT-P methods. However, it has an important limitation for evaluating fSfT methods because it is strongly sensitive to the accuracy of the focal length estimate, making it strongly correlated to FLPE. For example, if focal length is underestimated by 20% then the template’s reconstructed depth will generally be under-estimated by about 20% to compensate for the focal length error. This produces a large DE even if the template’s reconstructed shape is approximately correct. We handle this using a *Shape Error* (SE). First a least-squares translation t_z is computed along the camera’s optical axis to align the reconstructed 3D point

	Bedsheet	Kinect t-shirt	Kinect paper	Van Gogh paper
Object material	Fabric	Fabric	Paper	Paper
Template geometry	Flat open	Flat open	Flat open	Flat open
Number of template vertices (V)	1271	1089	1089	1189
Number of template triangles (T)	2400	2048	2048	2240
Video (vid.) or image collection (col.)	vid.	vid.	vid.	vid.
Number of images (N)	14 (68)	63 (313)	33 (100)	24 (71)
Image resolution ($w \times h$)	720×576	640×480	640×480	720×576
Correspondences per image N	1393	367	1228	4665
Focal length (px)	879.6	528.0	528.0	879.6
Focal length (% of w)	122.2	82.5	82.5	122.2
Lens opening angle ($^\circ$)	44.5	62.4	62.4	44.5
Has ground truth 3D	No	Yes	Yes	No

Table 5.5: Bedsheet, Kinect t-shirt, Kinect paper and Van Gogh paper dataset statistics.

correspondences with their ground truths. We then compute SE as follows:

$$\text{SE}(\hat{\mathbf{q}}, \mathbf{q}^{gt}) \stackrel{\text{def}}{=} \frac{100}{S} \times \|\hat{\mathbf{q}} + t_z - \mathbf{q}^{gt}\|_2 \quad (5.46)$$

5.4.3.2 Error success rate

Comparing FLPE and SE metrics is complicated because they can have extreme values, usually occurring when fSfT is weakly conditioned. Therefore non-robust summary statistics such as mean absolute error (MAE) or root mean squared error (RMSE) can be misleading. We compare metric using success rate, which is a robust statistic, defined as the proportion of instances (%) that a method returns a solution with an error less than a success threshold τ . We use FLPE-success@ τ as shorthand to denote the FLPE success rate using a threshold τ . Similarly, we use SE-success@ τ to denote the SE success rate. We use a few different thresholds to assess how often very accurate results are achieved (smaller τ) and how often results in the right ‘ballpark’ are achieved (larger τ).

5.4.4 Single-View fSfT evaluation

5.4.4.1 Summary of experiments

In §5.4.4.2 we compare the analytical and optimization-based methods, described in §5.1 and §5.2 respectively. This includes an evaluation of different initialization policies for the optimization-based method, comparing initializing with the analytical solution and focal length sampling. In §5.4.5.1 we evaluate the sensitivity of the optimization-based method to the isometric weight (a crucial hyperparameter), and we evaluate the performance of unsupervised weight selection described in §5.2.3.2.



Figure 5.10: Video datasets used for testing. Top row shows the templates for the Bedsheet, Bending cardboard, Kinect paper, Kinect t-shirt and Van Gogh paper datasets, with the template mesh overlaid. Rows 2-7 show representative images from each dataset. point correspondences are tracked through each video and row 7 shows their locations in the final video frames.

5.4.4.2 Comparison of methods: Original datasets

Recall that the optimization-based method is initialized by an initialization policy. There is an inherent trade-off between a policy with a larger initialization set (increasing computational cost) and a smaller initialization set (reducing computational cost but potentially reducing the chance of finding the global optimum). In this section we study this trade-off and compare results with the analytical method.

Tested initialization policies. We first compare three initialization policies defined with a set of opening angles Ψ and a set of SFT-P methods \mathcal{M} . These are as follows:

- Policy 1: $\Psi_{init} = \{\psi^{An}\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 2: $\Psi_{init} = \{20, 50, 80\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$

- Policy 3: $\Psi_{init} = \{20, 30, 40, 50, 60, 70, 80\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$

where ψ^{An} denotes the opening angle estimated by the analytical method. Thus, the number of initializations S for policies 1, 2 and 3 are 2, 6, and 14 respectively, with two per focal length (one for each CPSfT method applied to each focal length). The CPSfT methods are the same for these policies, so a difference in performance is explained only by the choice of Ψ_{init} .

All hyper-parameters for the analytical and optimization-based methods are set to their defaults as defined in §5.1 and §5.2 respectively. The isometric weight is fixed in this section for all datasets, and computed as described in §5.2.3.2 with grid-search performed on two randomly selected images from each dataset (24 images in total). A grid of 20 weights is used: $\{\lambda_{iso}^1, \lambda_{iso}^2, \dots, \lambda_{iso}^{j \in [1, 28]}\}$ with $\lambda_{iso}^1 = 10$ and $\lambda_{iso}^{i+1} = 2\lambda_{iso}^i$, $i \in [1, 19]$. For each image and grid weight λ_{iso}^1 , we optimize c using Algorithm 8 and policy 3. The weight λ'_{iso} yielding the lowest median FLPE is selected as the isometric weight.

Quantitative results analysis. Results are shown in Figures 5.11 and 5.12 with bar charts grouped by dataset. We also show the results of computing focal length with the analytical method, using MDH to compute deformation. This combination is denoted by **FAn+MDH**. We consider FLPE below 15% to be a good result for fSfT and FLPE below 5% to be an exceptional result. Thus, we evaluate both FLPE-success@15 and FLPE-success@5, shown in Figure 5.11(a) and (b) respectively. Similarly, Figure 5.12(a) and Figure 5.12(b) show SE-success@5% and SE-success@2% respectively. Figure 5.12(c) shows the mean number of Gauss-Newton iterations required by policies 1, 2 and 3.

We first consider FLPE-success@15 in Figure 5.11(a). We observe the following points:

1. The performance of **FAn+MDH** is very good for the Kinect and Van Gogh Paper dataset, where FLPE-success@15 is 100.0%. **FAn+MDH** achieves a relatively high FLPE-success@15 of 80.0% for the Hulk dataset. Recall that these dataset are smoothly deforming paper sheets with dense texture. These results indicate that the analytical method can estimate the focal length well in these cases.
2. For the other dataset (Spider-man, Cap, Bedsheet Kinect t-shirt, Handbag, Floral paper and Bending cardboard), **FAn+MDH** performs relatively poorly and much worse than optimization-based method (with any initialization policy). Indeed for the Cap and Bending cardboard datasets **FAn+MDH** has FLPE-success@15 of 0.0%: Therefore it was not able to find a focal length within 15% of ground truth in any of the images of those dataset. These results indicate that the analytical method does not work well in more difficult cases when texture is sparse and/or when deformation is complex.
3. There is little difference between policies 2 and 3. They achieve FLPE-success@15 of 100% for datasets with smoothly deforming, well textured objects (Spider-man, Kinect paper, Van Gogh and Hulk datasets). They achieve FLPE-success@15 above 60% for the Cap, Bedsheet, Kinect t-shirt, Handbag and Floral paper datasets. Considering the challenges associated with these datasets including strong complex and non-isometric deformation, this is a strong result. Furthermore, it indicates that (i) initialization with three fixed focal length samples (short, medium, far) achieves similar or better performance compared to policy 1, and (ii) there appears to be very little benefit in using more than three focal length samples.
4. For the Cap dataset, policy 2 has a higher success rate than policy 3. This may seem surprising because policy 3 initializes with more starts, including all starts in policy 2, so we may think

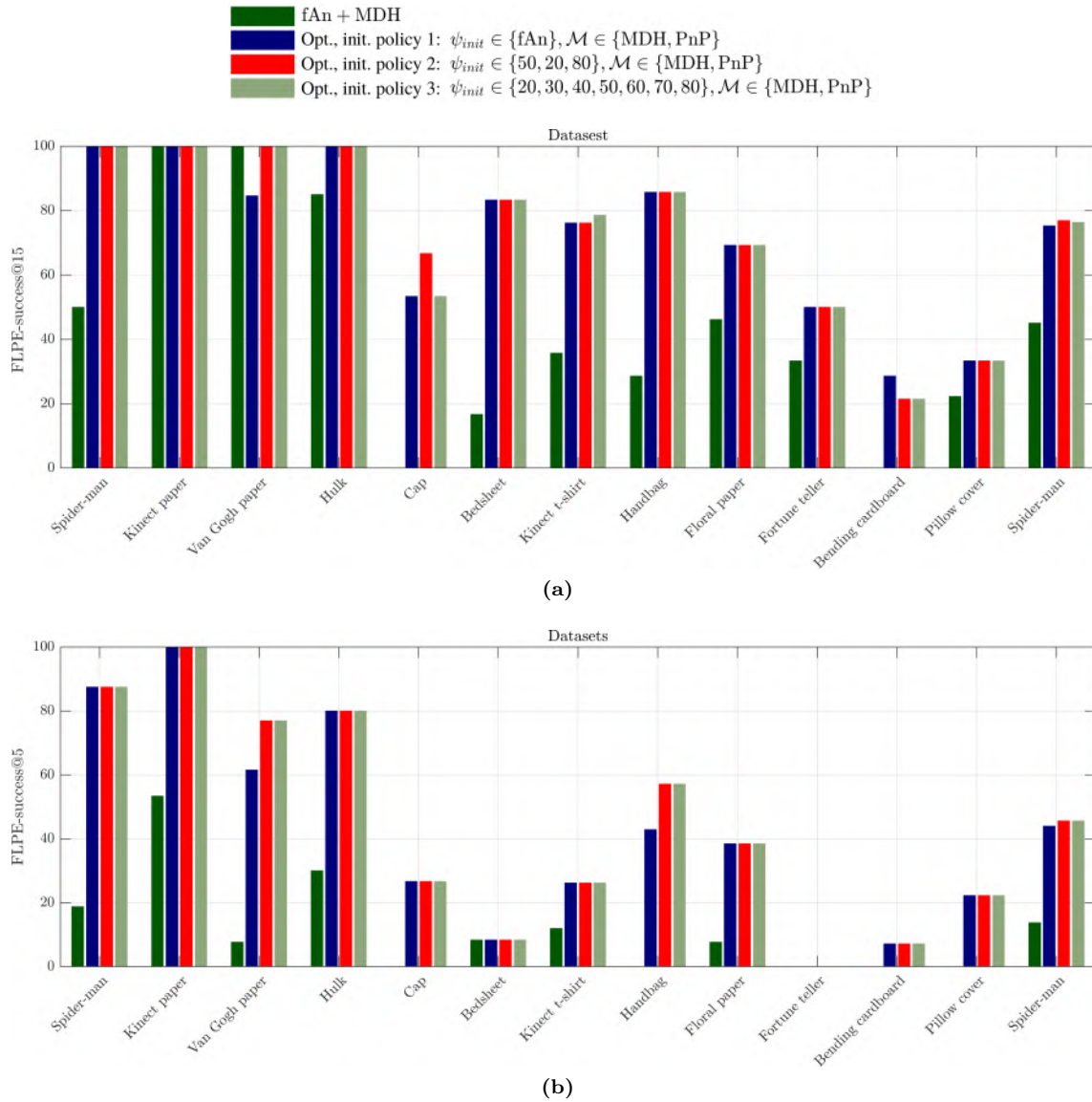


Figure 5.11: Focal Length Percentage Error (FLPE) results for the analytical method (denoted as ‘fAn+MDH’) and optimization-based method (denoted as ‘Opt.’) using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods \mathcal{M} used to initialize deformation. (a) shows FLPE success rates at 15% and (b) shows FLPE success rates at 5%.

policy 3 should always do better. This is not necessarily the case. The reason is because there exists an image in the Cap dataset with a spurious solution that has a lower cost compared to the true solution. This was located using policy 3 but not with policy 2. However, because in all other datasets the performance of policies 2 and 3 are practically identical, we see that this kind of events is extremely rare.

5. Performance is clearly strongly dataset dependent. The Bending cardboard and Pillow cover datasets have the lowest performance among all dataset. Recall that these datasets are very challenging because the Bending cardboard has extremely sparse correspondences, and the Pillow cover has many views that are approximately fronto-parallel (making fSfT poorly conditioned).

We now consider FLPE-success@5 in Figure 5.11(b). We observe the following:

6. Because of the much more stringent success threshold of 5%, we observe lower success rates for most datasets. Nevertheless, 100% success rate is achieved by the optimization-based method for the Kinect paper dataset with all initialization policies. Success rates above 78% are achieved for the Spider-man, Van Gogh paper and Hulk dataset with the optimization-based method and all initialization policies. This shows that we can solve fSfT with the optimization-based method and achieve very high accuracy (FLPE below 5%) for strongly isometric and well-textured objects.
7. For less isometric and/or weakly textured objects (those other than Spider-man, Kinect paper, Van Gogh paper and Hulk datasets), it is very challenging to solve fSfT consistently with high accuracy and FLPE below 5%.
8. Unlike FLPE-success@15, **FAn+MDH** achieves significantly lower success rates for FLPE-success@5 with the Kinect paper, Van Gogh and Hulk datasets compared to the optimization-based method. This indicates that the analytical method can achieve focal lengths in the right ballpark ($< 15\%$ error) for isometric well-textured objects, but it is not as precise as the optimization-based method.

We now consider Shape Error (SE) shown in Figure 5.12. Recall that Van Gogh, Bedsheet and Bending cardboard datasets do not have ground truth 3D information so SE cannot be measured. We observe the following points:

9. Very similar SE results are achieved for the different initialization policies for each dataset. This agrees with the FLPE results.
10. SE-success@5 is above 80% for all datasets with the optimization-based method with all initialization policies. Recall that an SE of 5 occurs when the Euclidean error at each reconstructed point is within 5% relative to the size of the object template. Thus, SE-success@5 above 80% is a strong result.
11. Unlike the FLPE results, the simpler dataset (Spider-man, Kinect paper and Hulk) do not have systematically better SE results compared to other datasets. Indeed, the Pillow cover dataset, which had the second lowest FLPE success rate among all dataset, has very similar SE-success@2 as the Hulk dataset. This highlights the intrinsic difficulty of the Pillow-cover dataset. It has little variation in depth, leading to weak perspective effects. This causes an ambiguity between the distance of the object to the camera and focal length, explaining why its shape can be estimated well but the focal length cannot.
12. The shape error of **FAn+MDH** is similar to the optimization-based method for the Hulk dataset, but it is generally significantly worse for the other datasets. Recall that the Hulk dataset has relatively strong perspective effects thanks to the short focal length and deformation is smooth with well-distributed point correspondences.

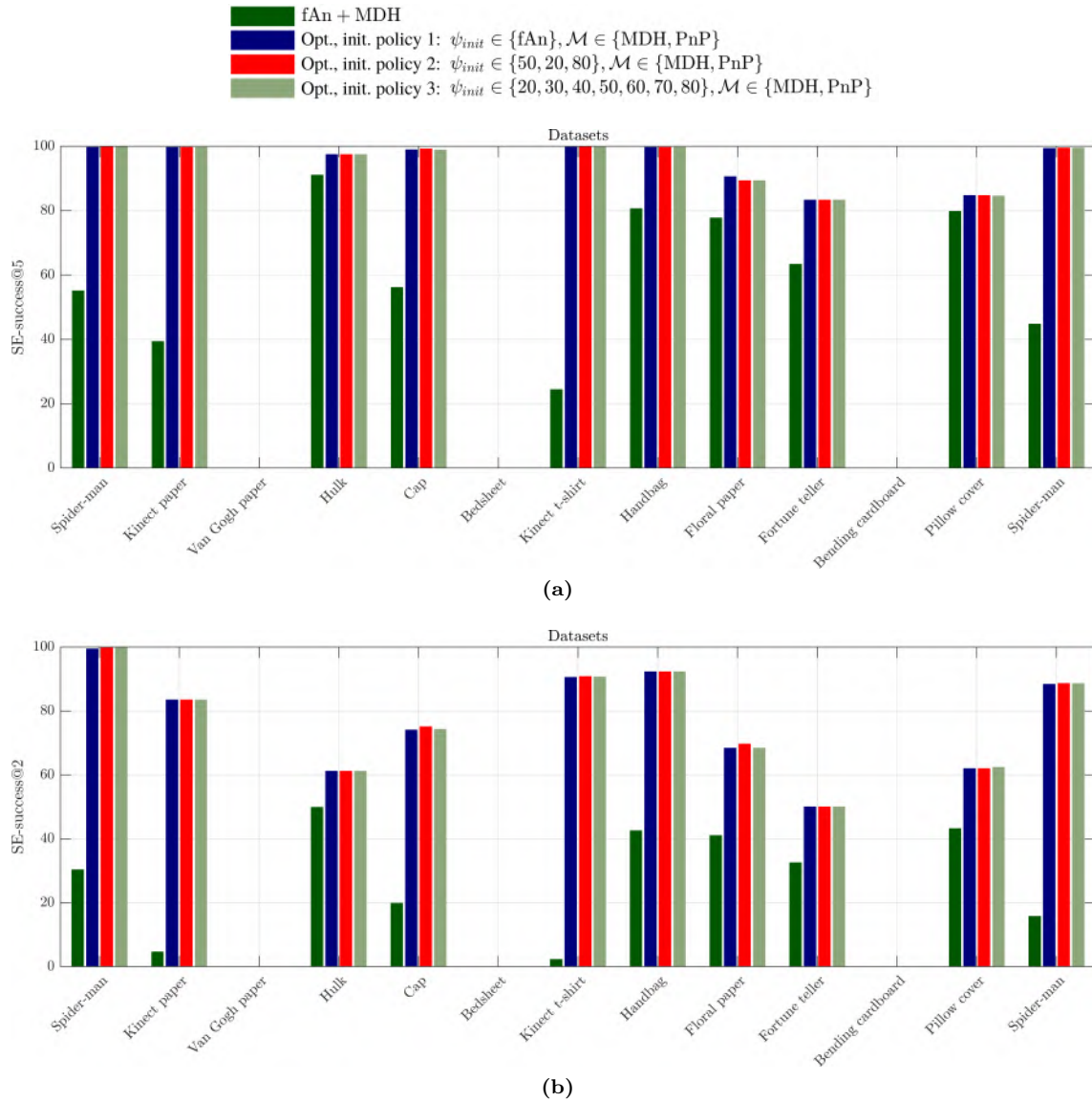


Figure 5.12: (a-b) shows the Shape Error (SE) of the method (denoted as ‘fAn+MDH’) and optimization-based method (denoted as ‘Opt.’) using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods \mathcal{M} used to initialize deformation. Van Gogh paper, Bedsheet and Bending cardboard dataset have no errors because they do not contain ground truth 3D information. For this reason there are no bars associated with them.

Results summary. The results show that the optimization-based method generally achieves far better accuracy compared to the analytical method. Therefore, in practical applications, the analytical method should be considered as a way to initialize the optimization-based method (as done in policy 1), and not as a competitive approach. Initialization with the analytical method appears to achieve similar accuracy compared to initialization with focal length sampling (policies 2 and 3). Furthermore, there appears to be no benefit in initializing with more than three focal length samples (policy 2 versus policy 3).

5.4.4.3 Further comparison of initialization policies

We now perform a deeper comparison of different initialization policies to understand (i) how sensitive performance is to the initialization policy and where the best trade-off lies between initialization set size and computational cost. To perform this evaluation, the datasets have some limitations. We first mention these and then we propose overcoming them with *dataset augmentation*. The limitations are as follows:

1. **Limited focal length variation:** With the exception of the Spider-man dataset, the lens opening angles are in the range $44.5^\circ \leq \psi \leq 62.4^\circ$. For the Spider-man dataset they are in the range $24.8^\circ \leq \psi \leq 65.3^\circ$. Therefore, we cannot assess the impact of focal length sampling when the range of focal lengths is broad.
2. **Uncontrolled noise:** We cannot assess the impact of point correspondence noise, because noise is not controlled.
3. **Unsolvable cases:** As shown in the previous section, some of the dataset (Fortune-teller, Bending cardboard, Pillow cover and Cap) have a significant proportion of images where focal length cannot be accurately resolved with neither the analytical method nor the optimization-based method. For example, from Figure 5.11(a), 65% of the images from the Pillow-cover dataset have FLPE over 15%. We believe these occur mainly when fSfT is weakly conditioned thanks to the problem’s geometry. Consequently, subtle performance differences between initialization policies can be hidden because of the large proportion of images that cannot be solved accurately no matter what initialization policy is used.
4. **Limited dataset sizes:** The non-video dataset do not contain a large number of images: all except the Spider-man dataset have 15 images or fewer. This is too small to reveal subtle differences in performance at the dataset level.

We now extend the evaluation to handle these aspects. For 1, we introduce greater focal length variation by applying simulated digital zooming (*zoom augmentation*). For 2, we apply simulated point correspondence noise to assess how performance is affected by increasing amounts of noise (*noise augmentation*). For 3, we compare performance using a subset of images for which fSfT can be solved with at least one method (*solvable filtering*) (SF). For 4, we combine datasets into *dataset groups* and compare methods with performance statistics at the dataset group level.

Zoom augmentation implementation. Zoom augmentation is implemented as follows. For each image in each dataset, we convert the point correspondences to retina coordinates then we projected them back to image coordinates using a simulated intrinsic matrix with a random focal length f_{rand} , with principal point at the image center and zero skew. We compute f_{rand} independently for each image, using an opening angle ψ_{rand} drawn with uniform probability in the range 10° to 90° , producing a wide range of focal lengths. We illustrate examples of images with simulated digital zoom for the Cap dataset in Figure 5.13.

Noise augmentation implementation. Noise augmentation is implemented to simulate increasingly adverse conditions, by adding noise to each point correspondence and by reducing the number of point correspondences. Reducing points is required because additional noise has a smaller influence on



Figure 5.13: 8 representative images from the Cap dataset where zoom augmentation is applied.

accuracy when there are many points. For each image in each dataset, we retain N' points sampled randomly and without replacement where N' is drawn uniformly in the range $[\min(N, 75), \min(N, 100)]$ where N is the original number of points in the image. Noise is added by randomly perturbing each of the retained image points by Gaussian I.I.D. noise of standard deviation σ (px). We test $\sigma = 0.16w$ and $\sigma = 0.32w$ where w is the image width. These are equivalent to a standard deviation of 1px and 2px respectively at 640×480 resolution. The latter can be considered strong noise.

Experimental setup. We define six dataset versions as follows:

- **v1:** No augmentation (original datasets)
- **v2:** Zoom augmentation and noise augmentation with $\sigma = 0.16w$
- **v3:** Zoom augmentation and noise augmentation with $\sigma = 0.32w$
- **v1+SF:** v1 with Solvable Filtering
- **v2+SF:** v2 with Solvable Filtering
- **v3+SF:** v3 with Solvable Filtering

We test 8 initialization policies. The first 3 policies are the same as defined previously and we introduce 5 new policies as follows:

- Policy 1: $\Psi_{init} = \{\psi^{An}\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 2: $\Psi_{init} = \{20, 50, 80\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 3: $\Psi_{init} = \{20, 30, 40, 50, 60, 70, 80\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 4: $\Psi_{init} = \{50\}$, $\mathcal{M} \in \{\text{MDH}\}$
- Policy 5: $\Psi_{init} = \{50\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 6: $\Psi_{init} = \{\psi^{An}\}$, $\mathcal{M} \in \{\text{MDH}\}$
- Policy 7: $\Psi_{init} = \{20, 50, 80\}$, $\mathcal{M} \in \{\text{MDH}\}$
- Policy 8: $\Psi_{init} = \{\psi^{GT}\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$

Policies 4 and 5 have one focal length sample whose opening angle is 50° , which is approximately the mode as discussed in §5.2.5.2. We compare them to evaluate the benefit of initializing with two CPSfT methods (MDH and PnP) compared with one (MDH). This is similarly done with policies 6 and 1, and policies 7 and 2.

Policies 6 and 1 have one focal length sample which is from the analytical method. Policies 7 and 2 have three focal length samples and policy 3 has 7 focal length samples. Policy 8 has one focal length sample, which is the ground truth with opening angle denoted by ψ^{GT} . Of course, we cannot use policy 8 in practice because it requires the ground truth. However, we use it to compare how well the other policies perform compared to the ideal of initializing with the ground truth focal length.

We test four dataset groups as follows where Groups 1-3 increase in difficulty:

- **Group 1** (Densely textured bending paper objects): Images from Spider-man, Kinect Paper, Van Gogh Paper, Hulk datasets
- **Group 2** (Densely textured cloth objects): Images from Cap, Bedsheet and Kinect-tshirt datasets
- **Group 3** (Sparsely textured and/or non-smooth objects): Images from Handbag, Floral-paper, Fortune-teller, Bending cardboard and Pillow-cover datasets
- **All datasets** (All datasets): Images from all datasets

We evaluate performance by averaging results within each group. Because datasets have different numbers of images and different numbers of points, datasets with many images and many points could dominate the group’s results. We handle this by macro averaging: For each dataset, we compute FLPE and SE success rates, and these are then averaged to compute the dataset group’s success rates.

Solvable Filtering (SF) implementation. So far we have compared initialization policies using a relatively large number of problem instances, including instances that may not be solvable by *any* fSfT method. This is a limitation because such instances dilute the effect of different initialization policies on the performance metrics. To deal with this, we also measure performance on a sub-set of problem instances for which the optimization-based method succeeds (we define success if the FLPE is below 15%.) To implement this, we ran the optimization-based method using *all* initializations contained in policies 1-8, and we filtered out the problem instance if its corresponding FLPE was above 15%. By only evaluating on the filtered set of problem instances, we could answer the question: How well does an initialization policy perform given that the problem is solvable using all considered initializations?

Results analysis with all datasets. Focal length results are shown in Figure 5.14, where Figure 5.14(a) shows FLPE-success@15 and Figure 5.14(b) shows FLPE-success@5 averaged across all datasets. We make the following observations:

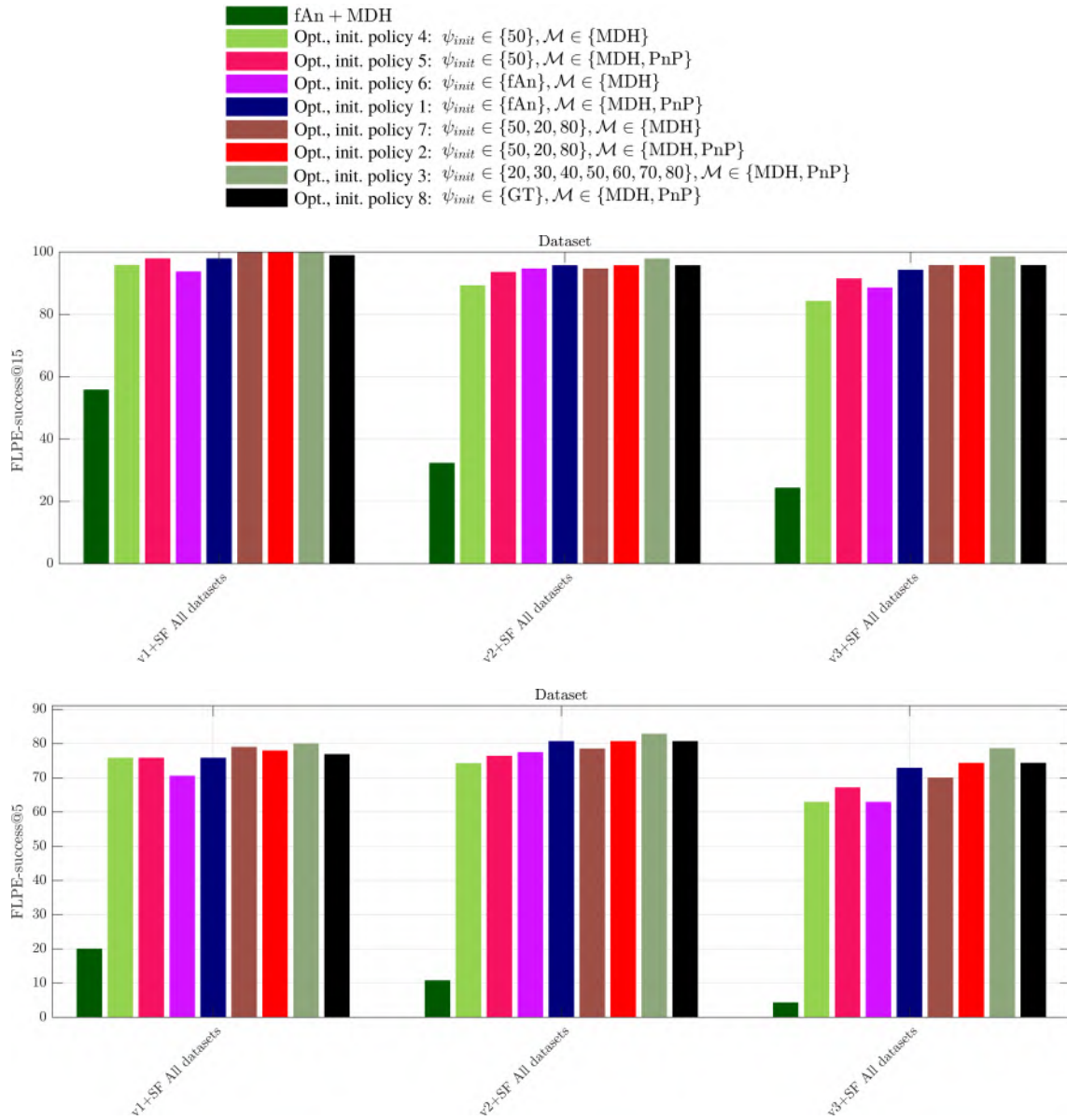


Figure 5.14: Focal Length Percentage Error performance of the optimization-based method (denoted as ‘fAn+MDH’) and optimization-based method (denoted as ‘Opt.’) using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SFT methods \mathcal{M} used to initialize deformation.

5. There is a strong trend where using more focal length samples in the initialization policy reduces focal length error. There is a slight improvement using policy 3 (with 7 samples) compared to policy 2 (with 3 samples). By contrast there is a strong benefit using policy 2 compared to policy 5 (with one sample). This illustrates diminishing returns where increasing the number of focal length samples has less of a benefit on solution accuracy.
6. The benefit of more focal length samples is less in v1+SF compared to v2+SF and v3+SF. This is because in v1+SF we do not apply zoom augmentation, so the focal lengths in v1+SF have opening angles in the range $24.8^\circ \leq \psi \leq 65.3^\circ$. In these cases the benefits of using more focal length samples is less pronounced compared to one sample at 50° .

7. There is a strong trend where using two CPSfT methods for initialization (MDH and PnP) improves performance compared to one CPSfT method (MDH). Recall that these methods operate very differently: MDH estimates deformation, and although it works well in general, there are cases when it does not estimate shape well thanks to the convex relaxation. By contrast, PnP does not estimate deformation, so the initialization it provides is the rigid pose that best fits the data. Adding the PnP solution appears to improve robustness in cases when the MDH solution cannot give a good initial estimate.
8. Initializing with the analytical method (policy 1) performs worse than initializing with a fixed opening angle of 50° (policy 5) for v1+SF. However, for v2+SF and v3+SF, we see a benefit where policies 6 and 1 outperform policies 4 and 5. Recall that v2+SF has zoom augmentation, and it has a much larger variation in opening angles compared to the original datasets without zoom augmentation (v1+SF). Therefore, when there is larger variation in opening angles, the analytical method is able to provide a better initialization compared to using a fixed opening angle of 50° . By contrast, in v1+SF, where the range of possible opening angles spans 40.5° with a midpoint at 45.0° , using a single opening angle of 50° performs better than using the opening angle from the analytical method.
9. Initializing with the ground truth focal length (policy 8) performs approximately the same as policy 2 (three focal length samples) for all dataset versions. This shows that accurate focal length initialization is not required by Algorithm 8.
10. Initializing with policy 8 performs worse in general than initializing with policy 3 (7 focal length samples). This can seem counter intuitive and we study the cause in more detail below. In short, the reason is because when we initialize with multiple focal length samples, we introduce shape diversity into the initialization set as a side effect. This diversity can help to locate the global optimum. We call this the *diverse initialization effect*.

The shape error results are shown in Figure 5.15, where Figure 5.15(a) shows SE-success@5 and Figure 5.15(b) shows SE-success@2. We observe all the same performance trends as we have observed for FLPE. The diverse initialization effect is also present.

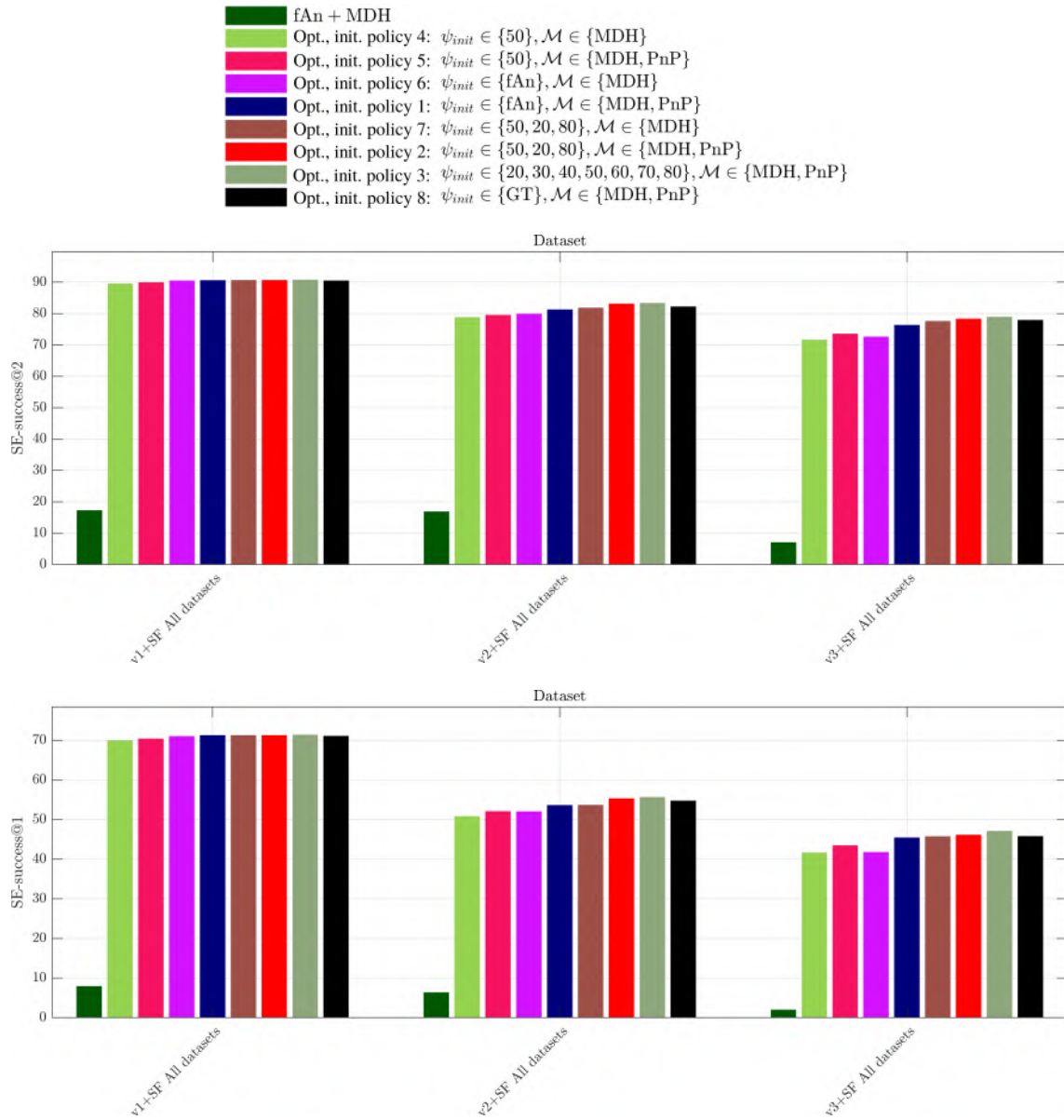


Figure 5.15: Shape Error performance of the optimization-based method (denoted as ‘fAn+MDH’) and optimization-based method (denoted as ‘Opt.’) using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods \mathcal{M} used to initialize deformation.

The diverse initialization effect. Why could initializing using several different and incorrect focal lengths improve performance compared to initializing with the ground truth focal length? We illustrate what is going on using the Van Gogh dataset (version v2), which has a clear example of the effect. In Figure 5.16 we plot FLPE against image index. The images are ordered over time with one image every 5 frames from the video. FLPE is plotted with policy 2 and policy 8. We cap FLPE to 200% to aid visualization. In this dataset the first 10 frames are quasi fronto-parallel, explaining the high errors at the start of the sequence. We illustrate 4 images from the dataset (images 11, 31, 56 and 71) in Figure 5.16, showing the increasing bending of the paper sheet and the random digital zoom applied to each image. Points in green are the point correspondences. Visualizations of the deformation solutions using policies 2 and 8 are shown below each image. These are rendered onto

the camera image plane and shaded to convey the estimated shapes. The performance of the two policies is identical except for image 56. In this image, policy 2 succeeds in finding a very good focal length with FLPE = 0.2%, however policy 8 fails with a very large FLPE of 3548%. There is a clear difference also in the estimated shape, where policy 8 is qualitatively worse than policy 2.

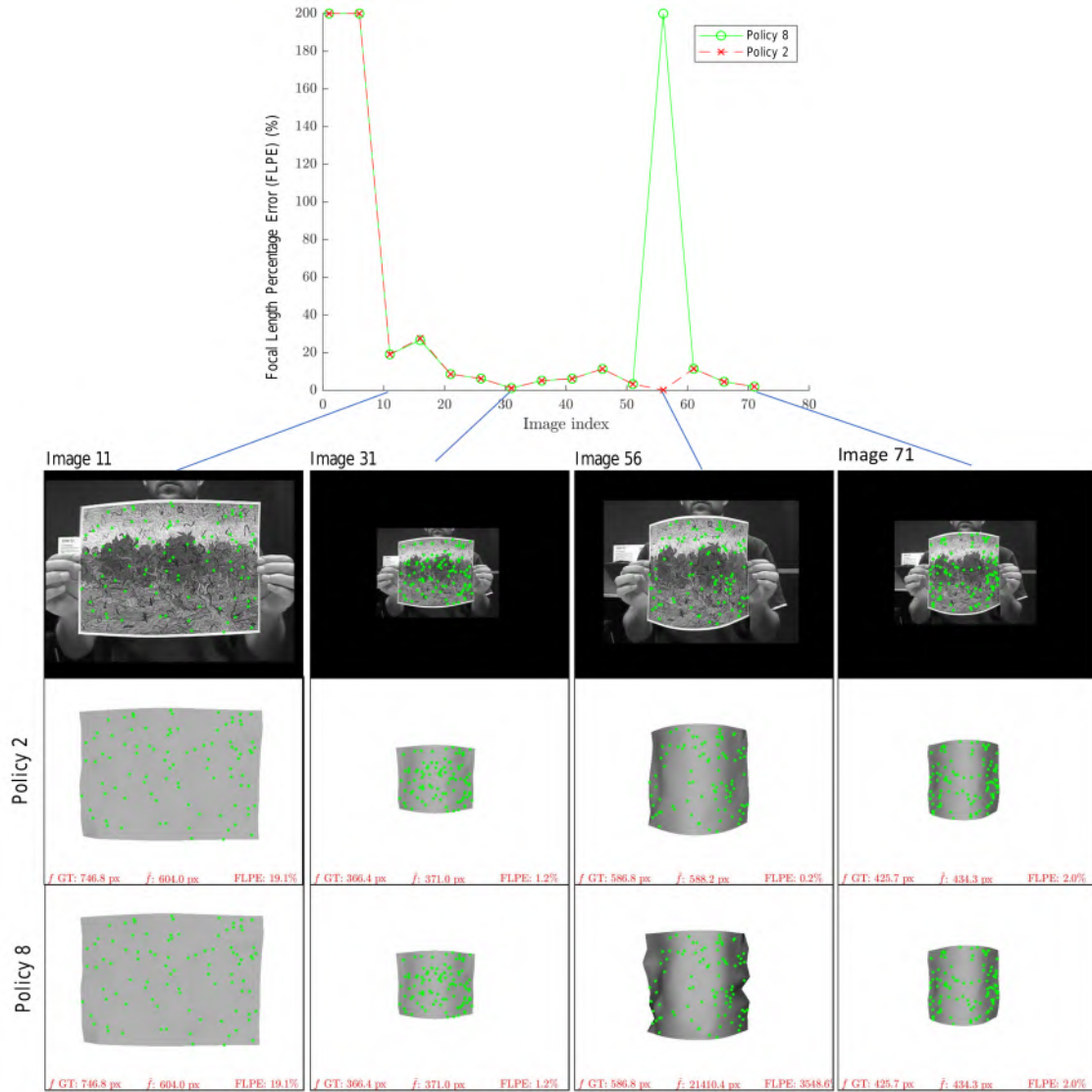


Figure 5.16: Results with different inits groups

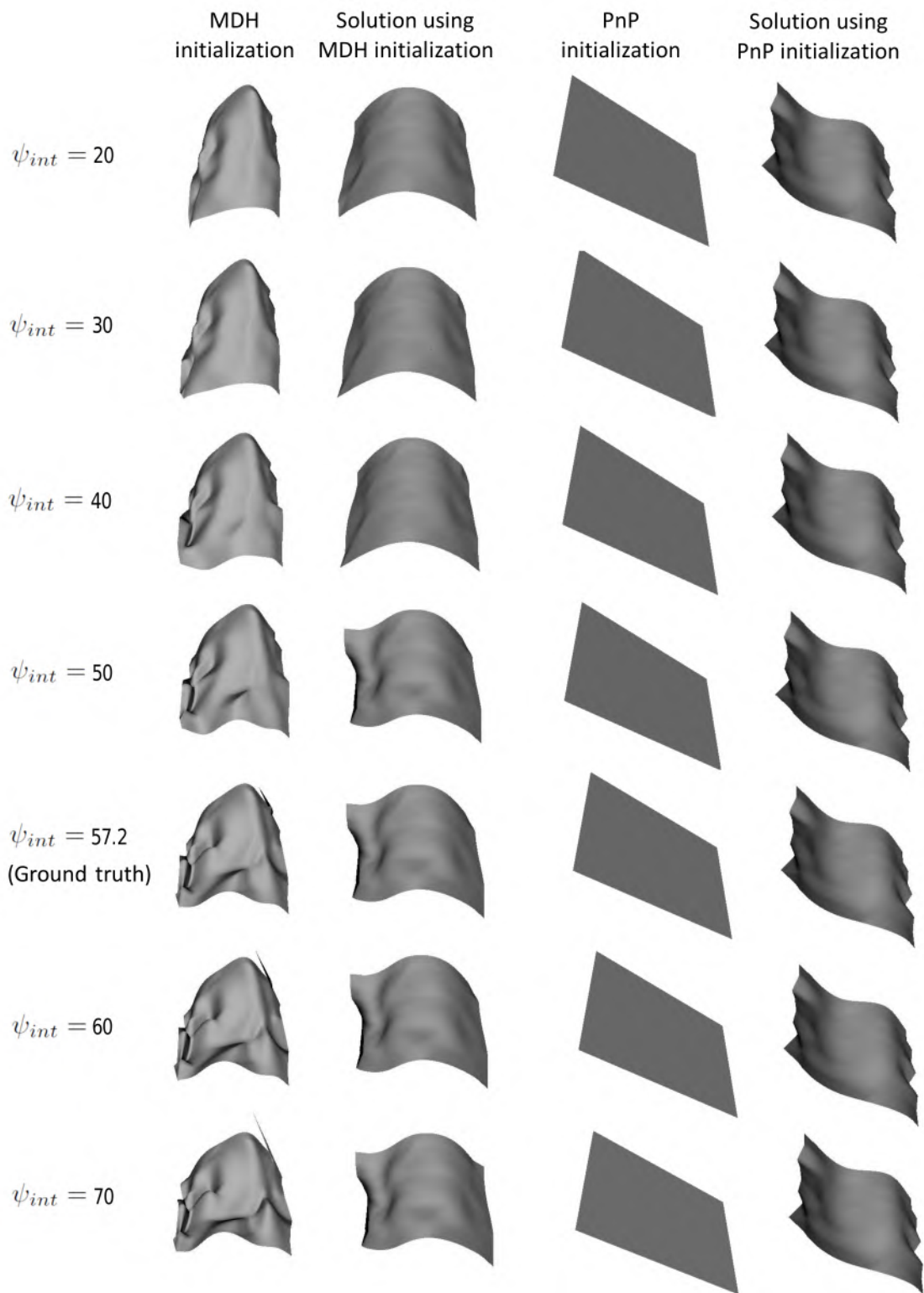


Figure 5.17: Illustration of the diverse initialization effect in the Van Gogh v2 dataset. At image 56, optimization initialized with the ground truth focal length (policy 8) fails, but it succeeds when initialized with three focal length samples (policy 2).

In Figure 5.17 we expose the underlying cause. The figure is arranged into 7 rows where each row corresponds to an initial focal length estimate, ranging from 20° to 70° opening angles at 10° intervals.

The 5th row corresponds to the ground truth focal length with opening angle of 57.2° . For each initial focal length, we show 4 renders. From left to right these are *i*) the initial deformation estimate from MDH, *ii*) the optimized fSfT solution using the MDH initial estimate, *iii*) the initial deformation estimate from PnP and *iv*) the optimized fSfT solution using the PnP initial estimate. The renders have been made with a virtual viewpoint by rotating the real camera by 40° about the camera’s *y*-axis. This helps to convey differences in shape. We see that the shape of the MDH initialization is approximately correct but it changes with focal length. Furthermore, initializing with MDH using opening angles of 20° , 30° and 40° lead to good shape estimates after optimization (second column). However, initializing with MDH and opening angles beyond 50° (including the ground truth) lead to poor shape estimates after optimization. This is because the optimization becomes trapped in a local minimum where the left side of the surface is incorrectly flipped upwards. Considering the PnP initializations (third column), these are all planar surfaces because the template’s reference shape is planar. For this particular image, no matter what initial focal length we use, none of the optimized solutions using PnP initialization is correct (fourth column). They are all trapped in the same local minimum, where the template’s left half is incorrectly flipped upwards. Flip ambiguities such as this is a recognized difficulty encountered in CPSfT [Chh+17b]. Considering 5th row in Figure 5.17 that corresponds to initializing with the ground truth focal length, we see that neither MDH nor PnP has managed to provide an initialization that converges to the correct solution. This explains the poor performance of policy 8 at frame 56 and the good performance of policy 2 (which includes 20° in the set of focal length initializations).

In conclusion, there is a hidden benefit of using multiple initial focal lengths, which is due to increasing shape diversity in the initialization set. This diversity can help locate the global minimum as illustrated in Figure 5.16 and as quantified in Figures 5.14 and 5.15.

Computation cost. We compare the computational cost of the different initialization policies by measuring the average number of optimization iterations (Gauss-Newton steps) required by Algorithm 8 using each policy. We use this instead of computation time because it is invariant to the implementation platform, and it is roughly proportional to computation time because the cost of executing each Gauss-Newton iteration is approximately constant at each iteration. The results are shown in Figure 5.18 where we observe the following:

11. There is a clear increase in computational cost using policies with a larger initialization set.
12. There is practically no difference in the computational cost of initializing using one focal length sample (policies 4 and 5) and using the analytical method’s focal length estimate (policies 6 and 1). This indicates that the number of iterations required for convergence is not highly sensitive to the accuracy of the initial focal length estimate.
13. The early termination criteria used in Algorithm 8 to avoid repeated search of solution space are proving effective. Without them, we would be seeing a doubling in the number of optimization iterations from policies using MDH to policies using both MDH and PnP. For example, the extra cost from policy 7 to policy 2 is between 22.2% and 32.7% depending on the dataset version. Without early termination the additional cost would be approximately 100%.
14. There is a slight increase in computational cost from v1 to v2 (and v1+SF to v2+SF) for all policies. This indicates that increasing noise also increases the number of iterations required for convergence.

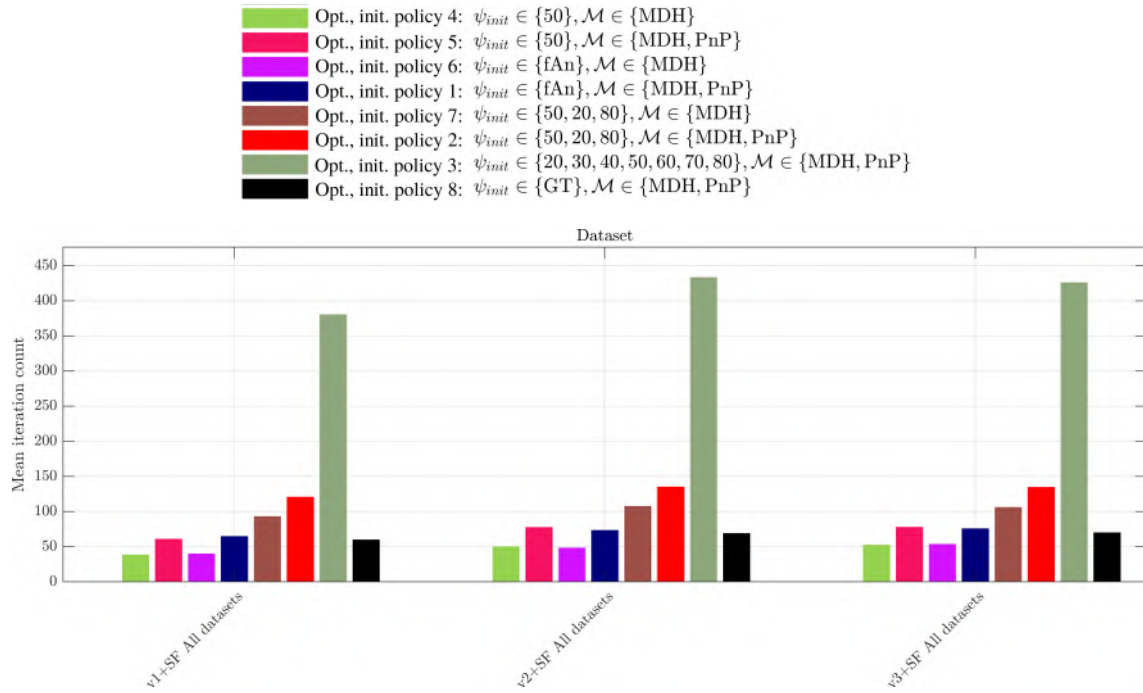


Figure 5.18: Computational cost the optimization-based method with different initialization policies. This is expressed in the average number of Gauss-Newton iterations required for Algorithm 8 to converge.

Quantitative results analysis with Groups 1, 2 and 3. We now break down FLPE results into Groups 1, 2 and 3 as defined in the *Experimental setup* paragraph. We show FLPE-success@15 results in Figure 5.19. Similar performance trends are seen in each dataset group as were found in the All dataset group (presented in Figure 5.14(a)). There are some other observations to be made. Firstly, the diverse initialization effect is apparent in Groups 1 and 3 (where policy 3 is similar or better than policy 8). However, it is not apparent in Group 2 (where policy 8 is similar or better than policy 3). The value of including PnP as an additional CPSfT method is present in all dataset groups.

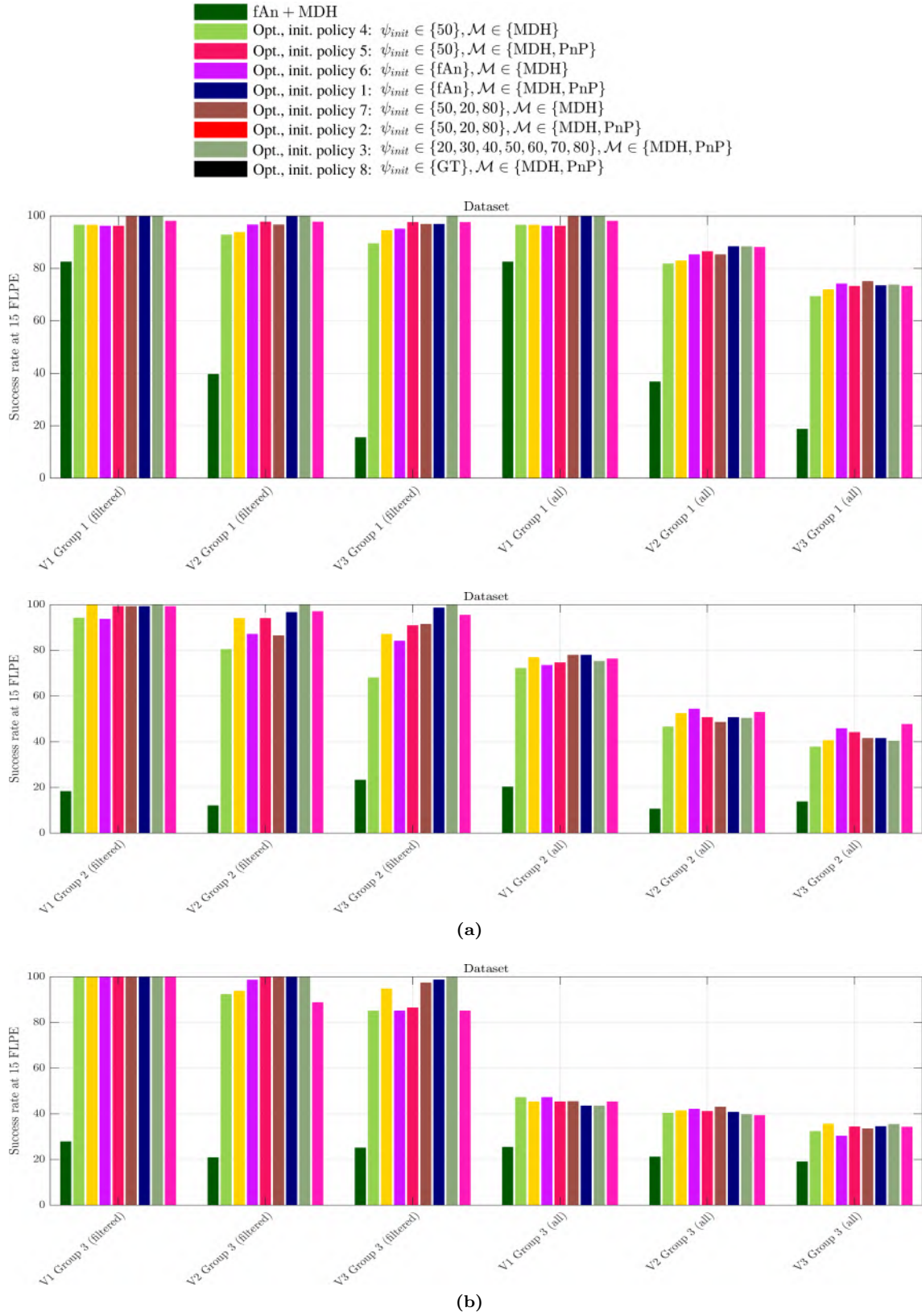


Figure 5.19: Comparison of the optimization-based solution with different initialization policies on three dataset groups defined in §5.4.4.3 *Experimental setup*. We also include the analytical method for comparison.

Summary: Which initialization policy should we use? The choice is a compromise between one with a larger initialization set (generally increasing accuracy) and a smaller initialization set (reducing computational cost). We have found there is consistent value in using two CPSfT methods (PnP and MDH) compared to just one (MDH). This is because PnP provides a cheap backup solution in cases when the MDH solution causes optimization to get trapped in a local minimum. Our early termination criteria used in Algorithm 8 significantly reduces the computational cost of using both methods compared to just one. The cost increase is between 25% and 40%, whereas without the early termination criteria the increase would be approximately 100%.

We also find that there is value in using multiple focal length samples, but in the original version of the datasets, where the range of lens opening angles is between 24.8° and 65.3° , the benefit is limited, where one sample at 50° gives very good results (policy 5). Nevertheless, we do see a benefit of using multiple focal length samples, which can be partially attributed to the diverse initialization effect. When the range of possible focal lengths is broader we see a clear benefit in more than one focal length samples, with a significant improvement from 1 to 3 samples, and a small improvement from 3 to 7 samples.

When run-time speed is more important, there is only a small performance drop between policy 3 (the most accurate policy) and policy 1 (using one focal length sample produced by the analytical method). However, there is additional overhead with using the analytical method: Using our current sub-optimal Matlab implementation takes in the order of a few seconds to run on a standard workstation. This is similar to the total cost of running the optimization-based solution with policy 2 (with sub-optimal Matlab code with GPU optimization). Thus, the value of initializing with the analytical method will only be realized with a very fast and parallelized implementation. Consequently, today policy 5 is preferred when speed is the primary concern, policy 2 meets a good middle ground between speed and accuracy when the range of possible focal lengths is large, and highest accuracy is obtained by policy 3.

5.4.5 Results visualizations

We visualize results of the optimization-based method (policy 2) in a series of figures with two datasets per figure. Results for the Spider-man and Bedsheet datasets are shown in Figure 5.20, results for the Kinect paper and Kinect tshirt datasets are shown in Figure 5.21 and results for the Van Gogh and Handbag datasets are shown in Figure 5.22. Results for the Pillow cover and Floral paper datasets are shown in Figure 5.23, results for the Pillow cover and Floral paper datasets are shown in Figure 5.24, and results for the Cap dataset are shown in Figure 5.25. Results for the Hulk dataset (where we were able to obtain one of the lowest errors among all datasets) are not shown because the dataset only provides point correspondences without matching images. Each figure is arranged in the same way. We show 4 rows of image per dataset. In the first row we show 5 representative images from the dataset. For the video datasets these are selected by uniformly sampling 5 frames over the image sequence. For the other dataset these are randomly selected. Each image is shown in grayscale and the point correspondences from the original dataset (v1) are shown as points. The points are colored by their Shape Error using jet color mapping. In the second row we show the estimated 3D shape of the template rendered from the camera’s viewpoint and shaded with a distant light source pointing in the direction of the camera’s optical axis. Below each render we give the ground truth focal length, the estimated focal length denoted by \hat{f} and the FLPE for that image. In the third row we show the same 5 images in the augmented dataset (v2). This shows the simulated zoomed images and their

respective point correspondences (recall that in v2 we down sample point correspondences and add noise, see §5.4.4.3). In the fourth row we show the estimated 3D shape of the template for each of the images in the third row, rendered from the camera’s viewpoint. Points for datasets without ground truth 3D information (Bedsheet, Van Gogh and Bending cardboard) are shown as green points.

Considering the v1 datasets (first two rows of each figure), we can see that 3D shape is generally very well estimated with almost all images having points with SE below 5%. The exceptions are the Pillow cover, Floral paper and Fortune Teller (Figures 5.23 and 5.24) where some of the images have points with SE above 5%. Nevertheless, in those cases, the general shape of these objects is captured well. We can also see that those images also correspond to a higher focal length error. Note that for the Floral paper and Fortune Teller datasets, the strong creases of the object are not recovered well. This is a consequence of the fact that correspondences are sparse and they provide insufficient information to resolve these high-frequency details. We can also see that results for v1 of the Cap dataset are very good (Figure 5.25), considering this is made of fabric that is not highly isometric and the deformation is strong and complex.

Considering the v2 datasets (second two rows of each figure), we can see that the 3D shape is also generally estimated very well. In a few of the images we see a significant increase in error compared to the v1 datasets. This has occurred two images of the Spiderman dataset (Figure 5.20), nevertheless only about 10% of the points have SE above 5%. The lack of high SE in the vast majority of these images (for both v1 and v2) indicates that our optimization-based solution has successfully avoided being trapped in local minima with incorrectly flipped regions of the surface. This pitfall is well known in CSfT, so it also manifests in fSfT: Examples of fSfT flip ambiguities have been illustrated in Figure 5.17.

5.4. EXPERIMENTAL RESULTS

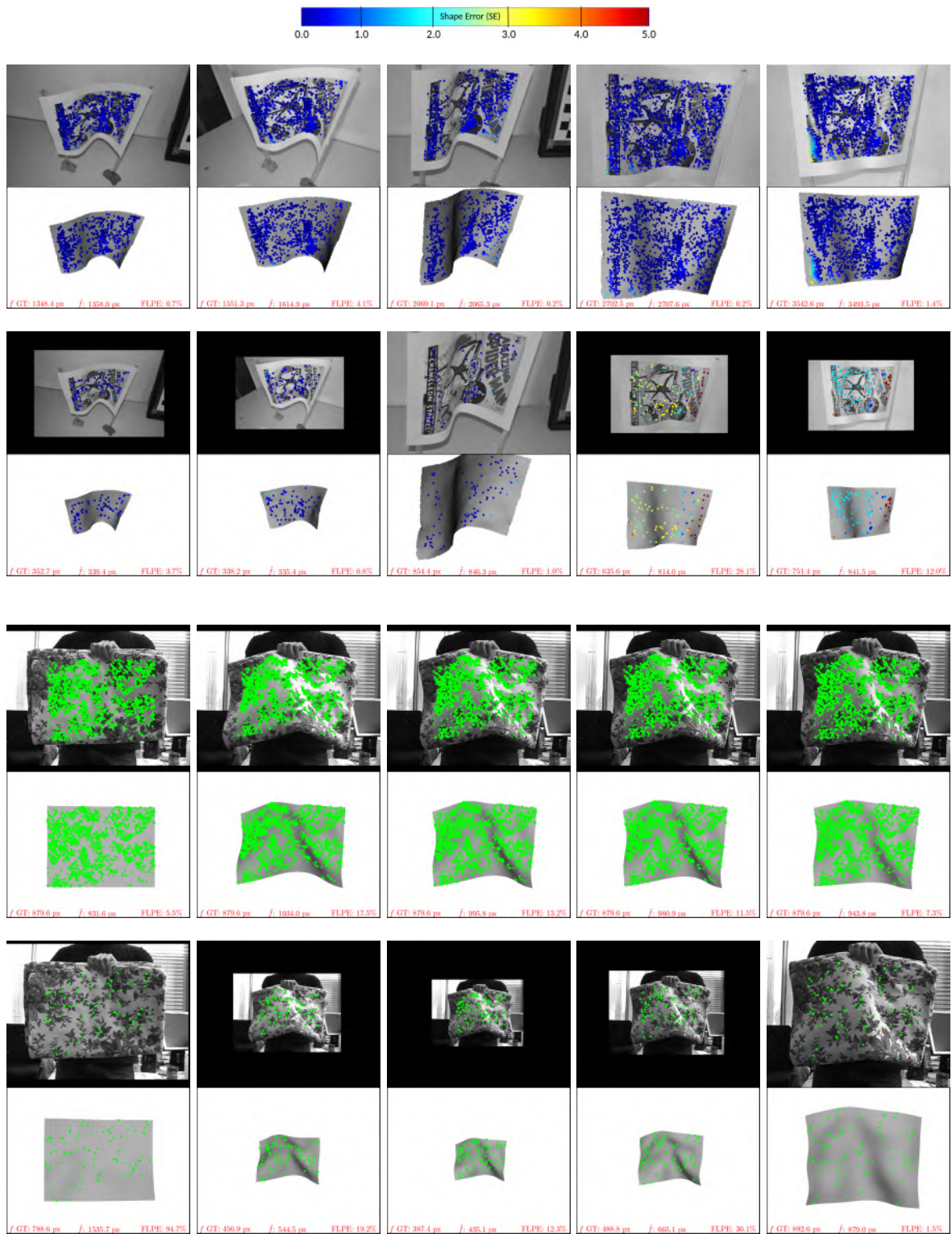


Figure 5.20: Visualizations of results with the Spider-man and Bedsheet datasets. Image rows are described from top to bottom. Row 1: 5 representative images from the Spider-man v1 dataset. Overlaid points are the point correspondences each colored according to their SE. Row 2: reconstruction of surface for each image in row 1, rendered using the deformed template from the camera viewpoint and shaded to reveal shape. Overlaid points are the correspondences on the template surface. Rows 3 and 4: results with the Spider-man v2 dataset in the same layout as rows 1 and 2. Rows 5 and 6: results with the Bedsheet v1 dataset in the same layout. This has no ground truth 3D information so points are colored in green. Rows 7 and 8: results with the Bedsheet v2 dataset in the same layout.

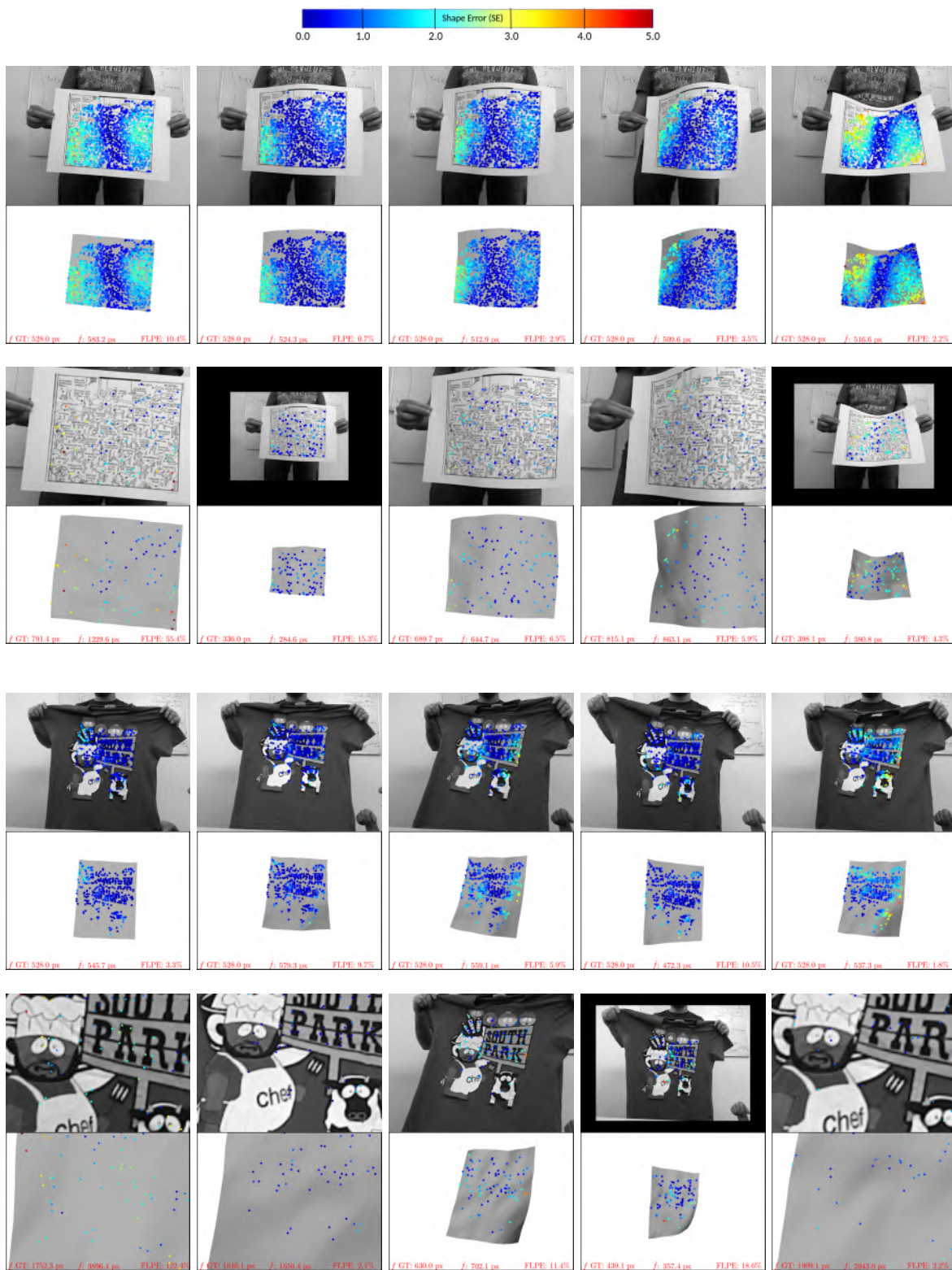


Figure 5.21: Visualizations of results with the Kinect paper and Kinect t-shirt datasets. Image rows are described from top to bottom. Row 1: 5 representative images from the Kinect paper v1 dataset. Overlaid points are the point correspondences each colored according to their SE. Row 2: reconstruction of surface for each image in row 1, rendered using the deformed template from the camera viewpoint and shaded to reveal shape. Overlaid points are the correspondences on the template surface. Rows 3 and 4: results with the Kinect paper v2 dataset in the same layout as rows 1 and 2. Rows 5 and 6: results with the Kinect t-shirt v1 dataset in the same layout. Rows 7 and 8: results with the Kinect t-shirt v2 dataset in the same layout.

5.4. EXPERIMENTAL RESULTS

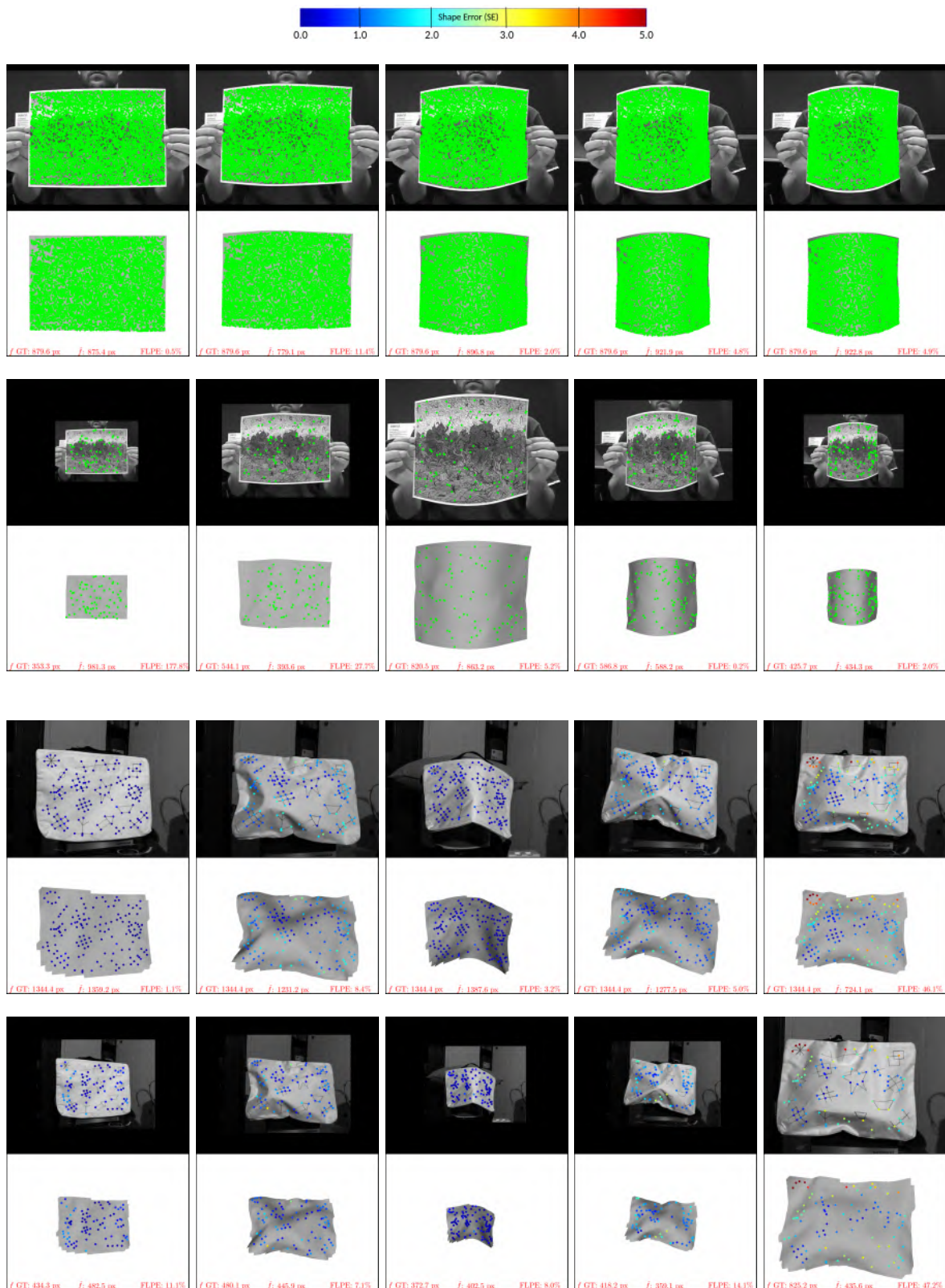


Figure 5.22: Visualizations of results with the Van Gogh and Handbag datasets. Image rows are described from top to bottom. Row 1: 5 representative images from the Van Gogh v1 dataset. Overlaid points are the point correspondences. This has no ground truth 3D information so points are colored in green. Row 2: reconstruction of surface for each image in row 1, rendered using the deformed template from the camera viewpoint and shaded to reveal shape. Overlaid points are the correspondences on the template surface. Rows 3 and 4: results with the Van Gogh v2 dataset in the same layout as rows 1 and 2. Rows 5 and 6: results with the Handbag v1 dataset in the same layout. Each point correspondence is colored according to their SE. Rows 7 and 8: results with the Handbag v2 dataset in the same layout.

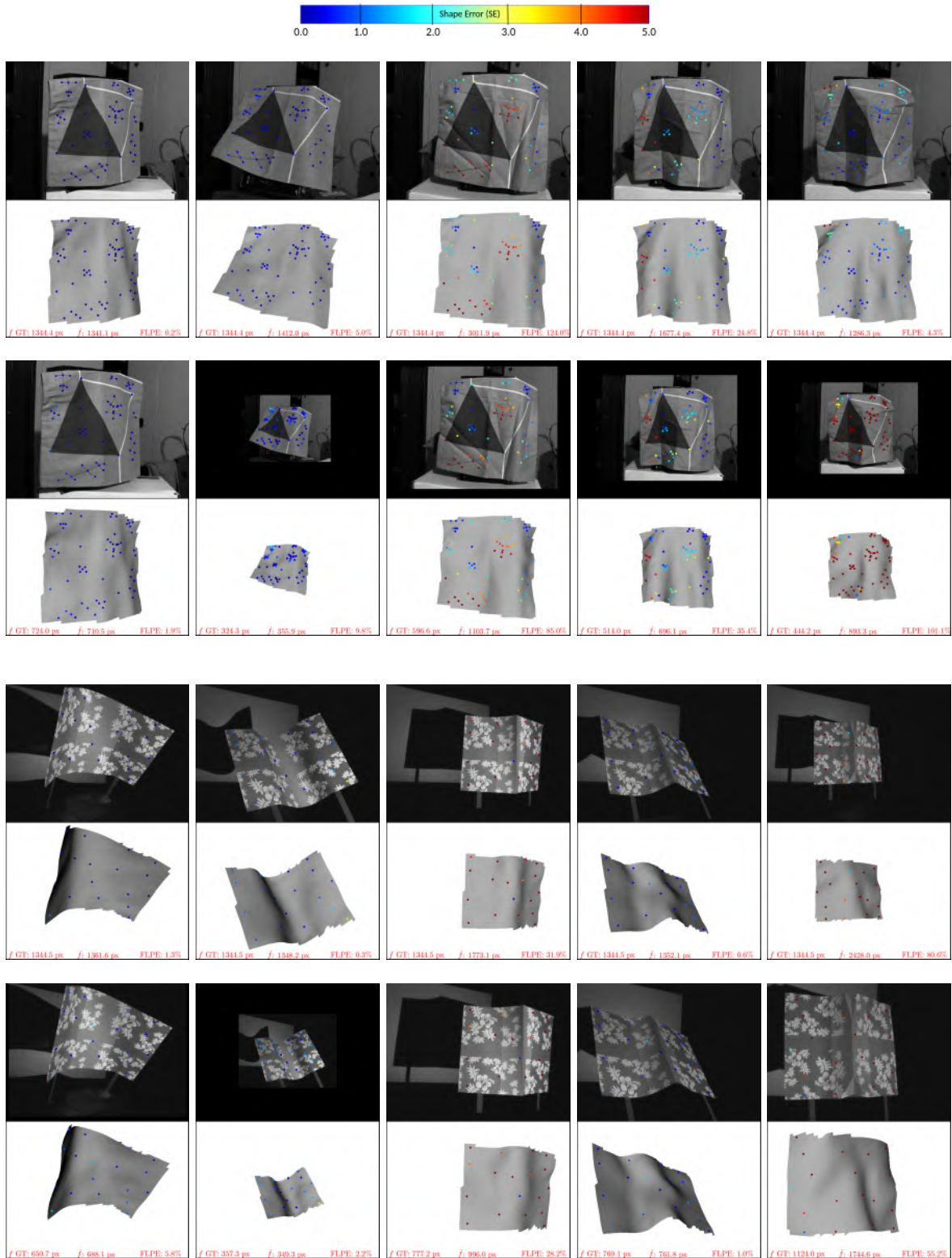


Figure 5.23: Visualizations of results with the Pillow cover and Floral paper datasets. Image rows are described from top to bottom. Row 1: 5 representative images from the Pillow cover v1 dataset. Overlaid points are the point correspondences each colored according to their SE. Row 2: reconstruction of surface for each image in row 1, rendered using the deformed template from the camera viewpoint and shaded to reveal shape. Overlaid points are the correspondences on the template surface. Rows 3 and 4: results with the Pillow cover v2 dataset in the same layout as rows 1 and 2. Rows 5 and 6: results with the Floral paper v1 dataset in the same layout. Rows 7 and 8: results with the Floral paper v2 dataset in the same layout.

5.4. EXPERIMENTAL RESULTS

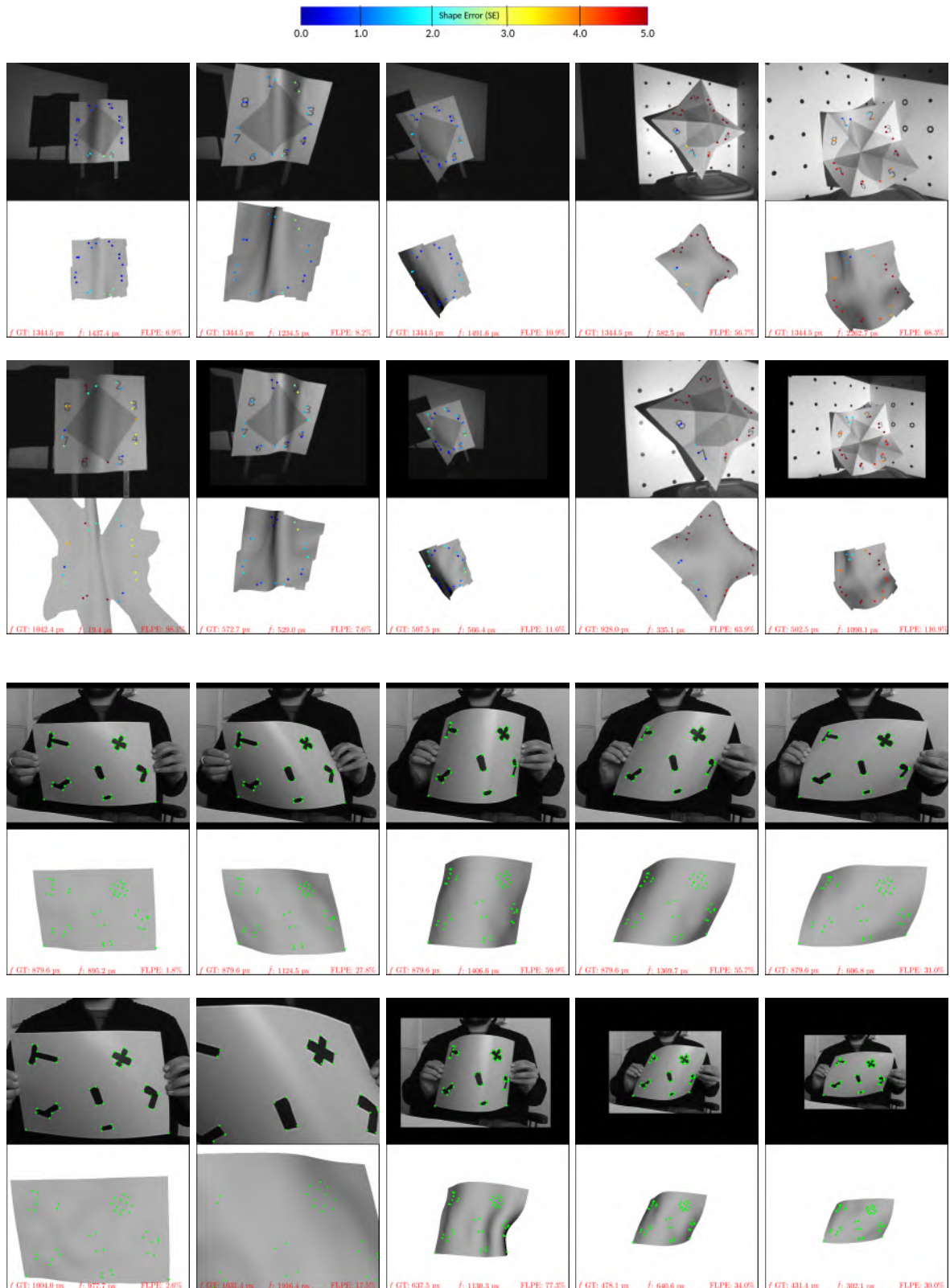


Figure 5.24: Visualizations of results with the Fortune teller and Bending cardboard datasets. Image rows are described from top to bottom. Row 1: 5 representative images from the Fortune teller v1 dataset. Overlaid points are the point correspondences each colored according to their SE. Row 2: reconstruction of surface for each image in row 1, rendered using the deformed template from the camera viewpoint and shaded to reveal shape. Overlaid points are the correspondences on the template surface. Rows 3 and 4: results with the Fortune teller v2 dataset in the same layout as rows 1 and 2. Rows 5 and 6: results with the Bending cardboard v1 dataset in the same layout. Rows 7 and 8: results with the Bending cardboard v2 dataset in the same layout. This dataset has no ground truth 3D so points are colored in green.

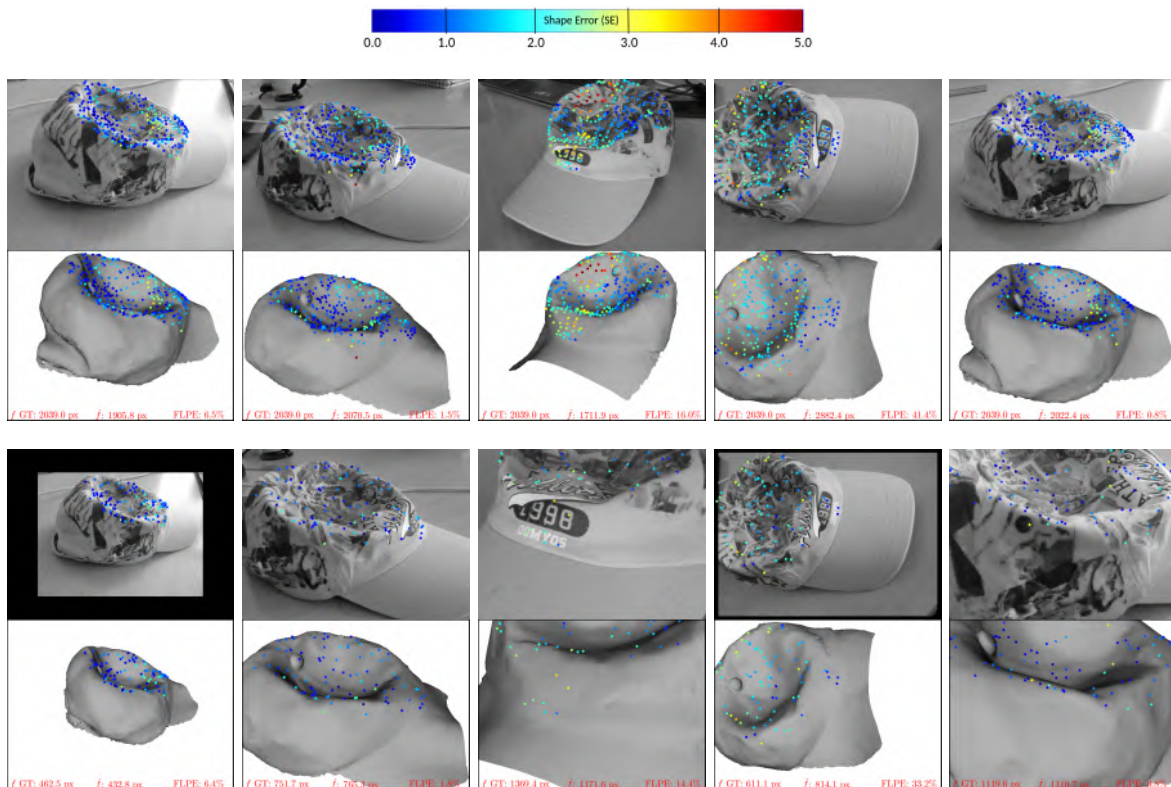


Figure 5.25: Visualizations of results with the Cap dataset. Image rows are described from top to bottom. Row 1: 5 representative images from the Cap v1 dataset. Overlaid points are the point correspondences each colored according to their SE. Row 2: reconstruction of surface for each image in row 1, rendered using the deformed template from the camera viewpoint and shaded to reveal shape. Overlaid points are the correspondences on the template surface. Rows 3 and 4: results with the Cap dataset v2 dataset in the same layout as rows 1 and 2.

5.4.5.1 Isometric weight analysis

The importance of setting correctly the isometric weight λ_{iso} for the optimization-based fSfT method has been discussed in §5.2.3. In this section we present experiments to answer the following important questions:

- How sensitive is the cost function to the isometric weight? Can a broad range of weights be used, or does it need to be carefully selected?
- Is the range of good isometric weights similar across datasets? If so, it demonstrates we have applied good cost normalization.
- Does the isometric weight that yields good focal lengths (low FLPE) also yields good 3D shapes (low SE)?
- What is the performance of the proposed unsupervised weight selection method described in §5.2.3.2?

We answer these questions empirically, by running Algorithm 8 with many candidate isometric weights, denoted by the set \mathcal{L} of size L . For each image, we take each weight $\lambda_{iso} \in \mathcal{L}$ and we optimize the cost function until convergence using Algorithm 8. We initialize optimization with policy 3 defined in §5.4.4.2. We then evaluate performance metrics (FLPE and SE) after convergence. We conduct this experiment using all images from the 12 public datasets described in §5.4.2 with a total of 301 images. We test with $L = 10$ (we run Algorithm 8 3010 times) with \mathcal{L} defined as follows:

$$\mathcal{L} = \left\{ \frac{1}{1000}\lambda'_{iso}, \frac{1}{100}\lambda'_{iso}, \frac{1}{10}\lambda'_{iso}, \frac{1}{5}\lambda'_{iso}, \frac{1}{2}\lambda'_{iso}, \lambda'_{iso}, 2\lambda'_{iso}, 5\lambda'_{iso}, 10\lambda'_{iso}, 100\lambda'_{iso} \right\} \quad (5.47)$$

where λ'_{iso} is the weight that was used in the previous section, found by minimizing the median FLPE as described in §5.4.4.2. Therefore, \mathcal{L} spans 5 orders of magnitude about λ'_{iso} . We sample more densely around λ'_{iso} for pragmatic reasons: it is highly computationally expensive to run the experiment with a dense sampling of weights, so we sample more densely around the relevant region of weight space. Following our evaluation in the previous section, we measure FLPE and SE on three dataset versions (v1, v2 and v3) defined in §5.4.4.3, *Experimental setup*. Recall that v1 corresponds to the original datasets, v2 applies zoom and moderate noise augmentation, and v3 applies zoom and strong noise augmentation.

We compare performance of selecting the isometric weight with three weight selection policies (WSPs):

- **WSP 1** (fixed weight selection): A fixed isometric weight is used for all images in all datasets. We test each weight in \mathcal{L} .
- **WSP 2** (image-specific unsupervised weight selection): An image-specific weight is used, computed using the unsupervised method described in §5.2.3.2. Recall this is done in two steps for each image. First we evaluate the problem conditioning after convergence for each weight in \mathcal{L} . Next we select the weight $\lambda_{iso}^{unsup} \in \mathcal{L}$ that has the best conditioning.
- **WSP 3** (Dataset group-specific unsupervised weight selection): We use a single isometric weight for each dataset group, computed using the unsupervised method. The purpose of this is to compare the benefit of an image-specific weight with a single weight used for all images in a

dataset group (WSP 2). We compute the dataset group’s weight as the median of its image-specific weights found using WSP2.

We first present results as bar charts in Figure 5.26. The figure is divided into four bar charts showing FLPE-success@15 (a), FLPE-success@10 (b), SE-success@5 (c) and SE-success@2 (d). Each bar chart is organized into 12 sets with one set for each dataset version (v1, v2 and v3). Within each set we plot 11 bars. The first 10 bars show the performance of WSP 1 using each weight in \mathcal{L} . The 11th bar shows the performance of WSP 2. The black filled circle shows the performance of WSP 3 (the circle is located at the weight that was automatically selected from \mathcal{L} with policy 3).

From Figure 5.26 we observe the following:

1. Concerning WSP 1, we see that the selected weight strongly influences performance. Excessively large or small weights compared to λ'_{iso} lead to poor performance. This is seen for all dataset groups and performance metrics. Performance is uni-modal with respect to the isometric weight and peaking at, or very close to λ'_{iso} . Using a fixed weight in the range $\frac{1}{2}\lambda'_{iso} \leq \lambda_{iso} \leq 2\lambda'_{iso}$ leads to similar performance where $\lambda_{iso} = \lambda'_{iso}$ is generally the best. The results also shows the importance of normalization. If for example we changed the template’s size by a factor of 10, then without scale normalization, the influence of the isometric cost (Equation 5.2.2.2) would be 10 times greater, and λ'_{iso} would not longer be a good weight: results would be equivalent to using a weight of $10\lambda'_{iso}$, with much worse results as seen in Figure 5.26. Similarly, if we were to not normalize by the number of points, a change in the number of points by a factor of 10 would also lead to significantly worse performance equivalent to $\lambda_{iso} = 1/10\lambda'_{iso}$. Recall that unlike v1, in v2 and v3 we significantly modify the number of points per image and the level of noise. The fact that the optimal weight is similar for all versions indicates good normalization.
2. Concerning WSP 2, the performance is similar to the best fixed weight. For v1 (data set version with lowest noise), performance of policy 2 is slightly lower than using a fixed weight of λ'_{iso} for policy 1. However, as noise increases (v2 and v3), policy 2 starts to outperform policy 1 especially in FLPE.
3. Concerning WSP 3, we see that one of two weights are selected: λ'_{iso} and $2\lambda'_{iso}$. Performance of these is fairly similar. However, WSP 3 is not always selecting the one with lowest error. This result indicates that the unsupervised method can find a good fixed weight of the dataset group within a factor of two of the actual best weight.

5.4. EXPERIMENTAL RESULTS

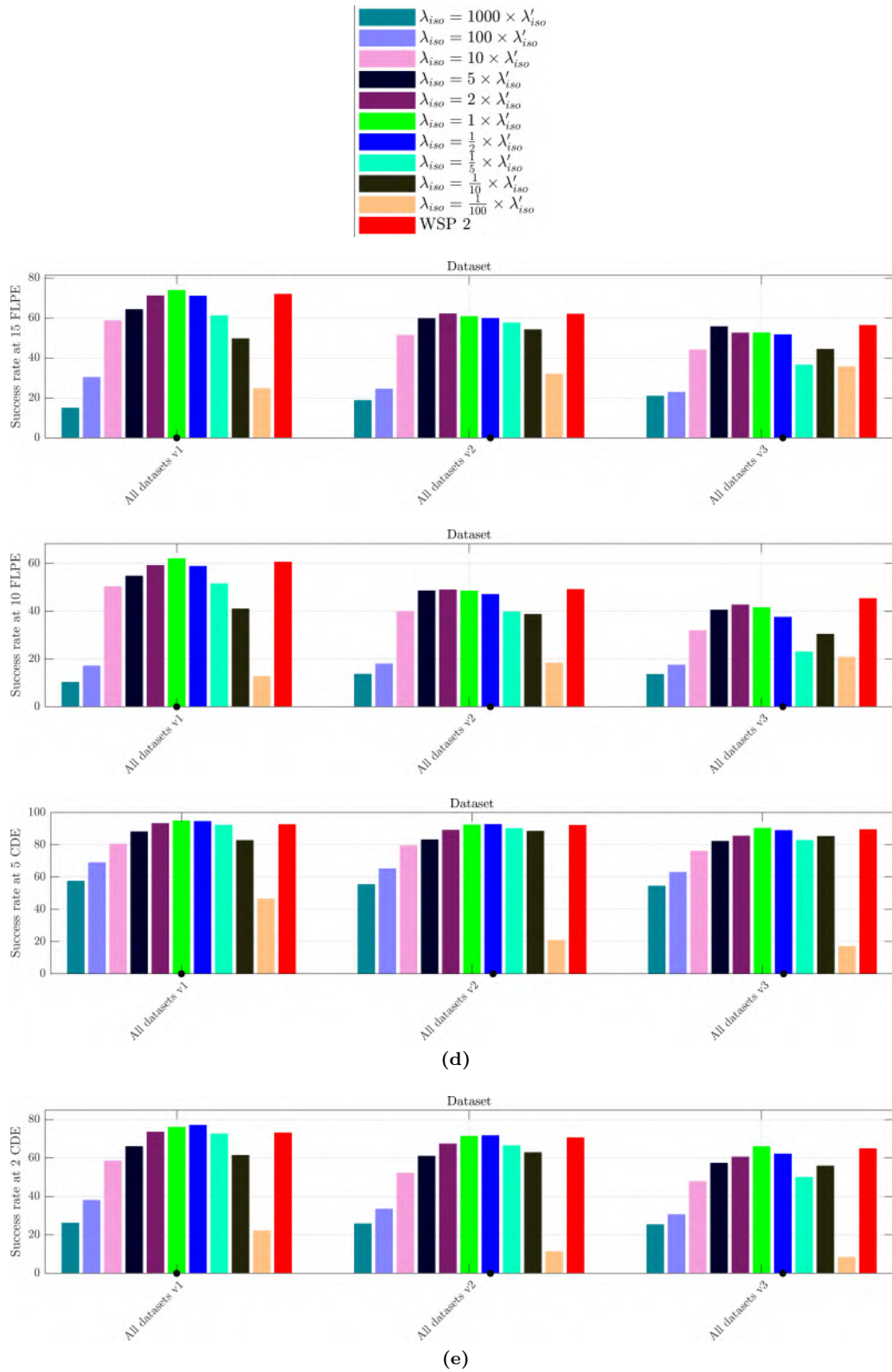


Figure 5.26: Performance of WSP 1, WSP 2 and WSP 3 averaged across all images in all datasets.

We now look at performance of the WSPs with each dataset. We show SE bar charts in Figure 5.27 and FLPE bar charts in Figure 5.28. These bar charts are arranged in exactly the same way

as Figure 5.26. Note that the bars are generally less smooth than Figure 5.26 because of the much smaller number of images associated to each dataset. Nevertheless, we observe the following trends:

4. Group 1 datasets (Spider-man, Kinect paper and Hulk) tend to have a wider range of good isometric weights. So, not only are those datasets simpler to handle because their surfaces and deformations are very smooth, but also we do not require a carefully tuned isometric weight.
5. As noise increases, the range of good isometric weights for Group 1 datasets tends to narrow and shifts slightly towards a stronger weight. We do not see a clear shift towards a stronger weight for all datasets. We believe this is because several datasets have objects with stronger and complex deformations than Group 1, and increasing the template's stiffness prevents it from deforming sufficiently. This is clearly visible with the Cap dataset, where increasing the weight from λ'_{iso} leads to a sharp drop in performance. Consequently, systematically increasing the isometric weight with increased noise is not a good policy.
6. In general, most datasets have an optimal fixed isometric weight in the range $\frac{1}{2}\lambda'_{iso} \leq \lambda_{iso} \leq 2\lambda'_{iso}$. This indicates that good normalization has been achieved across different datasets.
7. There is no consistent winner between the three weight selection policies: WSP 1 with $\lambda_{iso} = \lambda'_{iso}$, WSP 2 and WSP 3 (indicated by the black circles). Because of the considerable extra cost of WSP 2 and 3 (which requires running optimization with different isometric weights), there is not a convincing argument for using it on datasets where careful cost normalization has been applied. However, this may not always be possible as discussed below.

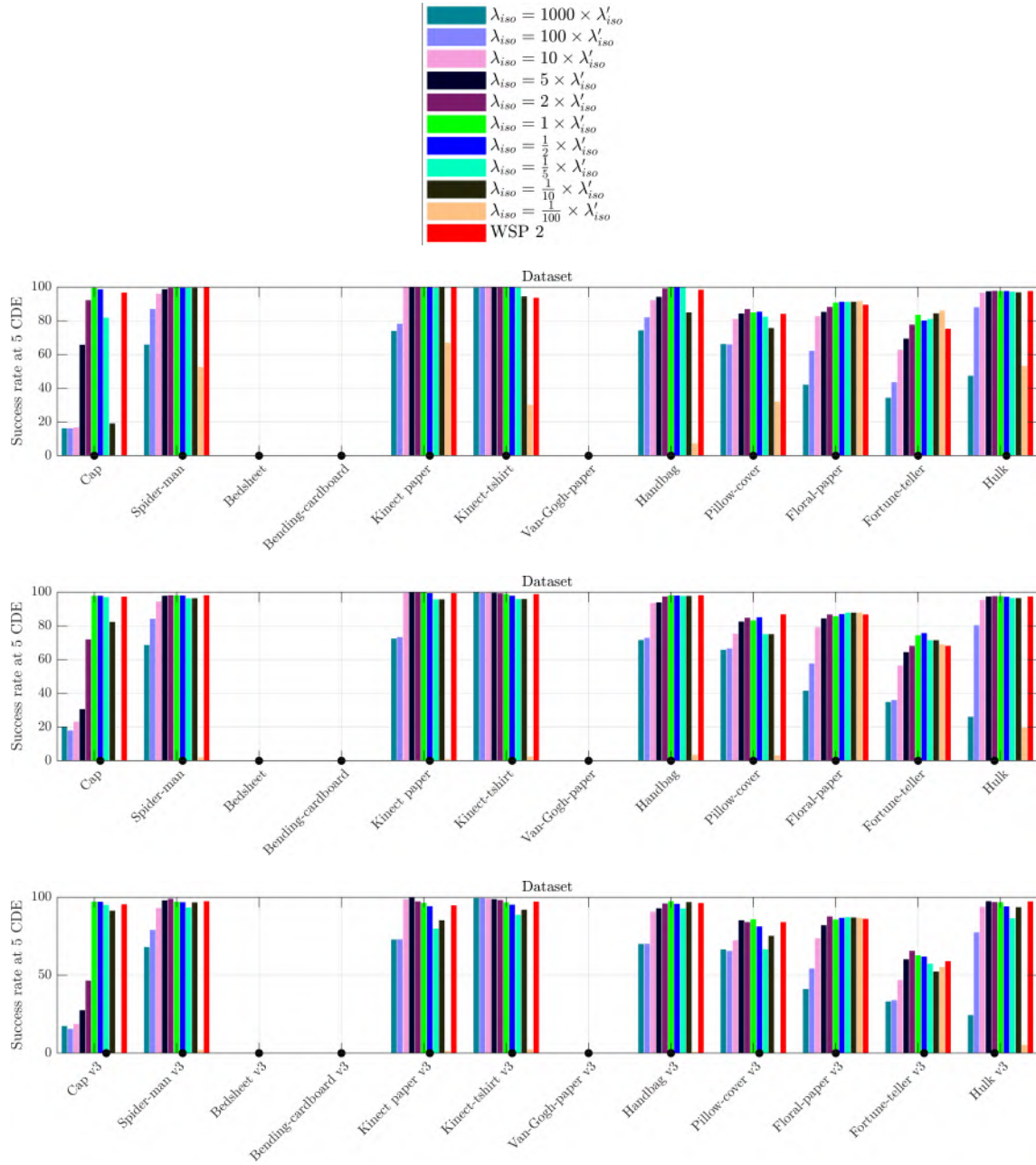


Figure 5.27: SE performance of WSP 1, WSP 2 and WSP 3 averaged across all images in ean datasets.

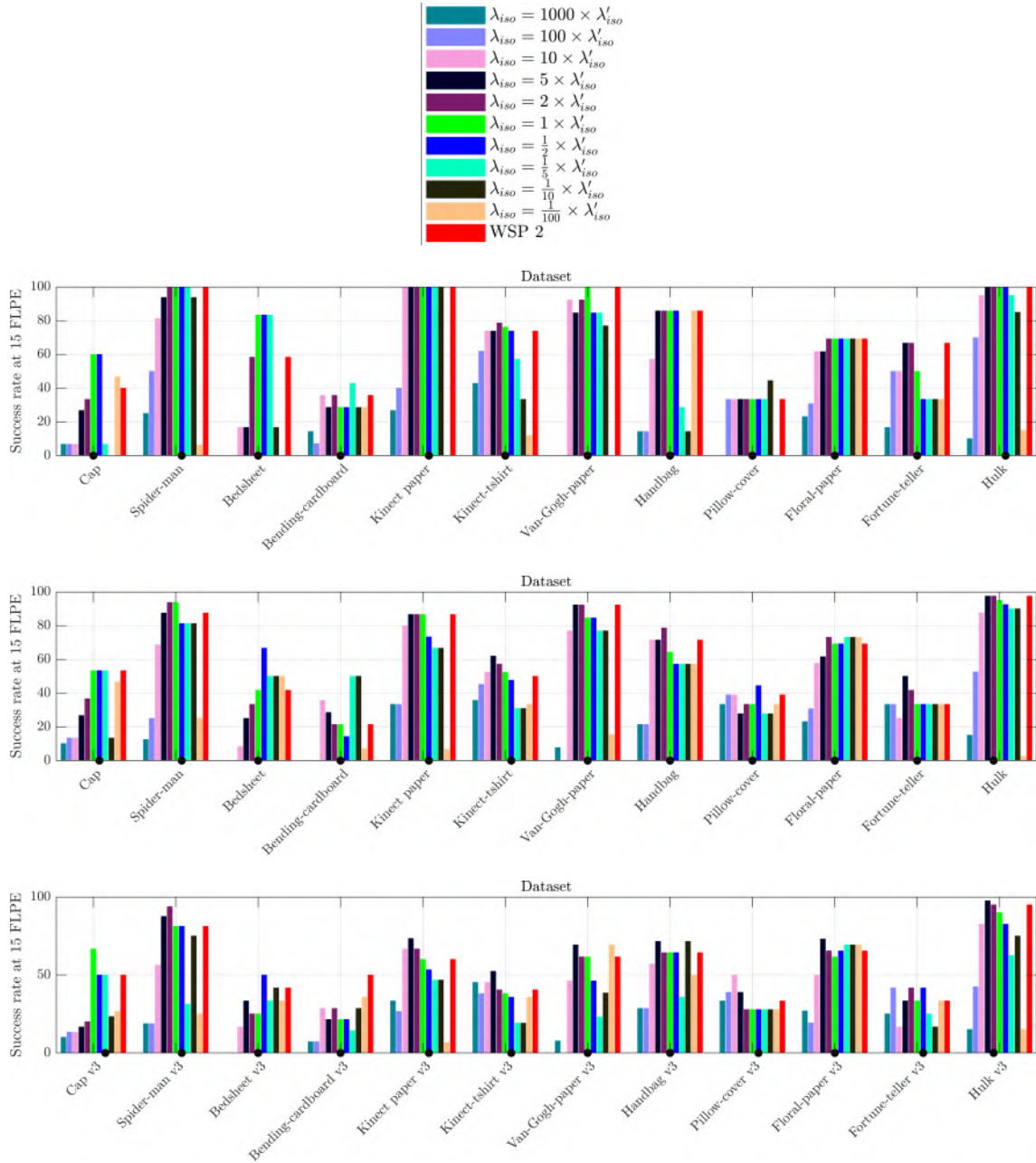


Figure 5.28: FLPE performance of WSP 1, WSP 2 and WSP 3 averaged across all images in each datasets.

Results summary. We now summarize results by answering the 4 questions asked at the beginning of this section from our observations. We recall that $\lambda'_{iso} = 1583$ is the default isometric weight that was found using a grid search in §5.2.3.2.

- *How sensitive is the cost function to the isometric weight λ_{iso} ? Can a broad range of weights be used, or does it need to be carefully selected?*

The cost function is sensitive and generally values in the range $\frac{1}{2}\lambda'_{iso}$ to $2\lambda'_{iso}$ can be used provided that normalization has been applied as described in §5.2.3. With increased noise, slightly better results can be obtained by increasing the isometric weight, however this is only clear for Group 1 objects that are relatively stiff.

- *Is the range of good isometric weights similar across datasets? If so, it demonstrates we have applied good cost normalization.*

The range of good weights is dataset specific. A broader range of weights centered around λ'_{iso} can be used for datasets of strongly isometric and well textured surfaces (Group 1).

- *Does the isometric weight that yields good focal lengths (low FLPE) also yield good 3D shapes (low SE)?*

Yes, this is the general case.

- *What is the performance of the proposed unsupervised weight selection method described in §5.2.3.2?*

It performs similarly to a fixed weight in the range $\frac{1}{2}\lambda'_{iso}$ to $2\lambda'_{iso}$. However, it appears to reduce FLPE slightly when the influence of noise is higher as shown in Figure 5.26.

Utility of unsupervised weight selection. Thanks to cost normalization, the benefit of the unsupervised weight selection method is not strongly apparent in these experiments. However, the fact that we can achieve similar or better performance compared to a carefully selected fixed weight, which requires ground truth, is noteworthy. Recall that the unsupervised method is very simple with no tuning parameters: it is able to select a good weight based only on the problem’s conditioning. It may therefore be useful in other settings where we cannot guarantee good normalization. For example, if we wanted to use a different isometric cost provided by a mechanics engine from a mechanics library such as SOFA [Fau+12], the unsupervised method could be used to automatically find the weight without ground truth. We aim to explore this aspect in future research including a more efficient method to search across weight space. Furthermore, correct weight selection is a known problem in other related problems such as CSfT, NRSfM and other registration problems involving deformable objects with physical priors. The idea of using problem conditioning as a way to select the weight automatically could be of value for those other problems, where weight selection is usually done by hand tuning.

5.4.6 Multi-view fSfT evaluation

5.4.6.1 Experimental setup

In this section we evaluate the methods for solving Multi-view fSfT described in §5.3. Multi-view fSfT solves fSfT with a set of M images of a deformable object with a common unknown focal length and a template. We evaluate the two presented approaches: robust focal length averaging (§5.3.2) and Multi-view fSfT optimization (§5.3.3).

We evaluate using the same datasets as in the previous section, excluding the Spider-man dataset because it consists of images with different camera intrinsics. We evaluate using image sets drawn randomly from each dataset with varying sizes. This allows us to measure the benefit of additional image information as M increases. We perform this with the following sampling scheme. We take each dataset whose number of images is denoted by N_D . For each image i in the dataset we make a random permutation \mathbf{p}_i of size $N_D - 1$ that stores the indices of all images except i in a random order. For the i^{th} image, we draw a set of images from the dataset of size M denoted as $\mathcal{I}_i^M \in [1, N_D]^M$. This is defined as follows:

$$\mathcal{I}_i^M = \{i, \mathbf{p}(1 : M - 1)\} \quad (5.48)$$

where $\mathbf{p}(x : y)$ denotes the elements of \mathbf{p} between indices x and y (inclusive). As we increase M , we add more images to the set without taking any images away. This allows us to establish smooth trends in performance as M increases. Each image set is processed by a method to estimate the common focal length of the image set. We evaluate performance using mean FLPE as a function of M . We cap mean FLPE at 50% to prevent it from being severely contaminated by image sets with extreme FLPE. This is reasonable because a FLPE at 50% is a significant failure. We evaluate 6 methods to estimate the common focal length of each image set. These are as follows:

- **Mean SV:** The single-view optimization method is run on each image in \mathcal{I}_i^M . The mean focal length is returned.
- **Median SV:** The single-view optimization method is run on each image in \mathcal{I}_i^M . The median focal length is returned.
- **ESAC + Mean SV:** The single-view optimization method is run on each image in \mathcal{I}_i^M . Exhaustive Sampling and Consensus (ESAC) is used to find the set of inlier focal lengths as described in §5.3.2. The mean focal length of the inlier set is returned.
- **MV:** The Multi-view fSfT optimization method is run on \mathcal{I}_i^M . This is initialized by **Median SV**.
- **ESAC + MV:** The single-view optimization method is run on each image in \mathcal{I}_i^M . ESAC is used to find the set of inlier focal lengths as described in §5.3.2. The Multi-view fSfT optimization method is run on only on the inlier set. This is initialized using the mean focal length of the inlier set.
- **SV:** The single-view fSfT optimization method is run on the first image of \mathcal{I}_i^M . This is used as a baseline to compare the improvement of the above methods with different M .

We refer to **Mean SV**, **Median SV**, **ESAC + Mean SV**, **MV** and **ESAC + MV** as *multi-view methods* because they use all images in each image set. By contrast, **SV** is a *single-view method* because it only uses the first image in each image set. For **SV**, we use initialization policy 2, which was found to be a good policy in §5.4.4.2.

We evaluate 10 different image set sizes: $M \in \{1, 2, 3, 4, 5, 6, 7, 8, 10, 12\}$. When $M = 1$, all methods are equivalent. The evaluation is therefore large scale: a total of 415×10 image sets are tested with 6 different methods. Furthermore, we test each image set with two versions: images from the original dataset (v1) and images with noise augmentation as described in §5.4.4.3 with $\sigma = 1.0$ at VGA resolution (v4). We do not use dataset versions v2 and v3 in this evaluation because they apply

zoom augmentation, so there is no common focal length in the image sets. We compute mean FLPE using the three dataset groups used in the previous section and defined in §5.4.4.3, *Experimental setup*.

5.4.6.2 Results for Group 1

Results for Group 1 are shown in Figure 5.29. We note that mean FLPE is constant in the number of images with **SV**, because **SV** always uses one image (the first one of each image set). We make the following observations from Figure 5.29:

1. As the number of images increases, all multi-view methods significantly improve on **SV**. This clearly shows that the multi-view methods are able to exploit the extra information provided in multiple images to reduce focal length error. At 12 images multi-view methods are able to attain a mean FLPE of just 1.62% for Group 1 version v1, and just 2.23% for Group 1 version v4 (where recall significant noise is added to the point correspondences). These results show that we can estimate focal length very accurately using multiple images of Group 1 datasets (smooth surfaces that are made of paper sheets whose deformation is strongly isometric).
2. The best-performing multi-view method depends on the dataset version (and therefore on the amount of point correspondence noise):
 - In v1, the best methods are **Mean SV** and **ESAC + Mean SV** (attaining very similar mean FLPE). In v4, the best performing method is **ESAC + MV**: At 12 images, **ESAC + Mean SV** reduces mean FLPE by 51.7% compared to **SV**. For v4, at 12 images **ESAC + MV** reduce mean FLPE by 71.5%.
 - Comparing **Mean SV** and **MV**, **Mean SV** performs better in v1, and **MV** performs better in v4.
 - Comparing **Mean SV** and **Median SV**, **Mean SV** performs better in v1, and **Median SV** performs better in v4.
3. We see diminishing returns for all multi-view methods as the number of images increases: The improvement from 2 images to 4 images is significantly greater than the increase from 10 images to 12 images. Diminishing returns are stronger in v4 than v1.

The fact that no single multi-view method performs best in v1 and v4 is noteworthy and it is worth understanding why.

For MV, why do we see diminishing returns earlier in v1 than in v4? We believe this is because in v4, accuracy is mainly limited by *data error* (point correspondence noise). By contrast, in v1, data error is much lower, so performance is mainly limited by *modeling error*. Modeling error is caused by geometric modeling and/or cost function modeling error: Geometric modeling errors arise by modeling surfaces as triangulated meshes, and cost function modeling arise when the true solution does not perfectly coincide with the cost function’s global minimum (caused by *e.g.* imperfect weighting of the cost terms, where perfect weighting is impossible to achieve in practices). These results indicate that when data error dominates, **MV** is able to significantly improve results with additional images. By contrast, when modeling error dominates, additional images help, but they cannot completely overcome modeling error, leading to a performance plateau. Indeed, we see that mean FLPE of **MV** in v1 does not significantly improve beyond 6 images. Thus, adding more images is not helping when

modeling error dominates. By contrast adding more images is clearly helping to overcome stronger data error in v4, and we see that the performance of **MV** is still improving at 12 images.

Why does ESAC + Mean SV have similar performance as Mean SV in version v1, but better performance in v4? This is because in v1, there are relatively few *outlier images*. We define an outlier image as one for which the focal length and/or shape estimate for that image using **SV** is highly wrong. There are more outlier images in v4 thanks to the instabilities that more noise brings to weakly conditioned images. When there are outlier images, the negative effect on **Mean SV** is natural because the mean is not a robust statistic. The fact that **ESAC + Mean SV** and **Mean SV** have similar performance in v1 indicates that ESAC is having little effect thanks to the low proportion of outlier images. In contrast, ESAC is clearly helping in v4 to remove outlier images.

Why does ESAC + MV have similar performance as MV in version v1, but better performance in v4? This is for similar reasons as the previous question. In v1, ESAC is not filtering out many outlier images because they are very infrequent, so the performance of **ESAC + MV** and **MV** is similar in v1. By contrast, in v4, where outlier images are more common, their effect is to corrupt the initialization of the multi-view optimization problem, which can lead to optimization being stuck in a local minimum. The results indicates that **MV** is not highly robust to outlier images.

Why is Mean SV better in v1 compared to MV but worse in version v4? This question is not easy to answer. Intuitively, **MV** should outperform **Mean SV** in both v1 and v4 because **MV** exploits more geometric constraints. Specifically, **MV** connects the unknown deformations for each image and the unknown focal length in a single cost function. By contrast, **Mean SV** breaks these connections (it makes a relaxation): Focal length and deformation are optimized for each image independently, then the focal length results are averaged. We believe that in version v1, the relaxation is not significantly harming results because we have seen previously that fSfT can be solved with just one image in Group 1 v1 with high accuracy. This is shown here where **SV** has a mean FLPE of just 3.45% in v1. **Mean + SV**, which simply averages the independently solved focal lengths appears to work well. This indicates that the central limit theorem is in effect and focal length estimates from **SV** have noise that is approximately I.I.D.

In contrast, for v4, where data error is much greater, **MV** is clearly doing better than **Mean + SV**. The relaxation made by **Mean + SV** is significantly impacting performance with higher levels of data error. In such cases, **MV** is able to mitigate data error better than **Mean + SV** because it exploits geometric constraints that exist between different images.

5.4.6.3 Results for Groups 2 and 3

Results for Groups 2 and 3 are shown in Figures 5.30 and 5.31. Recall that Group 2 consists in deformable objects made of smooth cloth sheets, which are significantly less isometric than the objects in Group 1. Objects in Group 2 also have significantly sparser texture than Group 1. Group 3 consists in deformable objects that have highly sparse texture and folded sheets of paper. Folding is not well handled by the cost function because it has a convex regularizer (c_{reg}) that penalizes folded solutions. This implies strong modeling error.

We make the following observations:

4. Similarly with Group 1, as the number of images increases, all multi-view methods significantly improve on **SV**. This clearly demonstrates that the proposed multi-view fSfT methods can

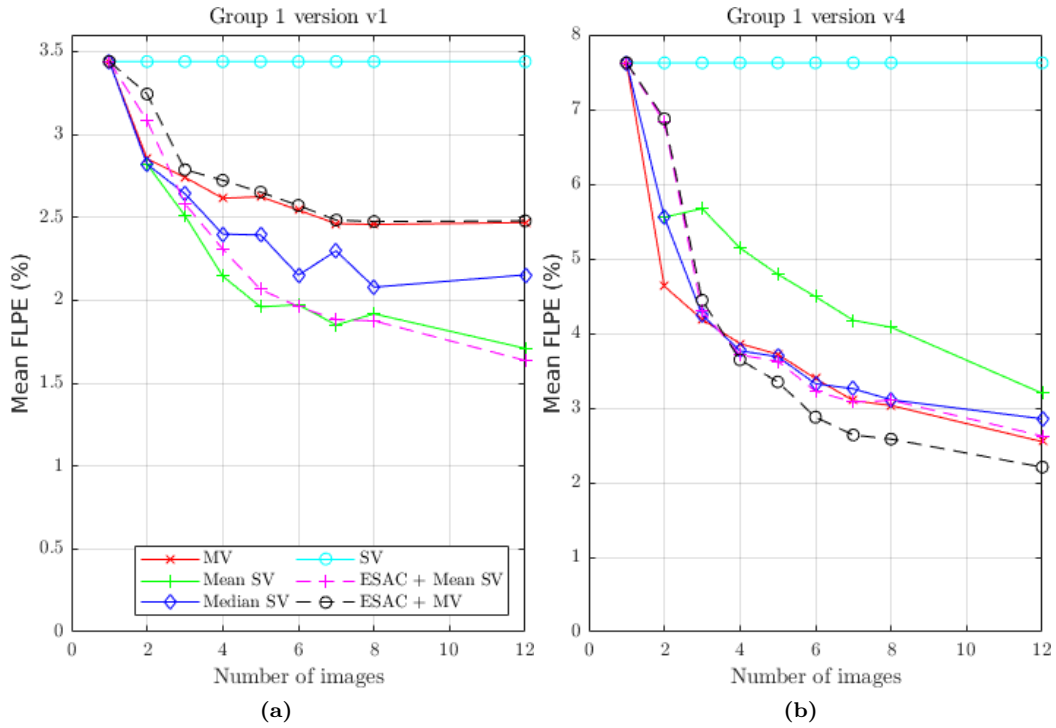


Figure 5.29: Comparison of multi-view fSFT methods using Group 1 mean FLPE

significantly improve results for all three dataset groups. Given the inherent difficulties of Groups 2 and 3, this is a strong result.

- Group 2: In v1 and 12 images, a multi-view method is able to attain a mean FLPE of 5.62%, compared to 11.82% with **SV** (an improvement of 52.4%). In v4 and 12 images, a multi-view method is able to attain a mean FLPE of 6.82%, compared to 15.91% with **SV** (an improvement of 57.1%).
 - Group 3: In version v1 and 12 images, a multi-view method is able to attain a mean FLPE of 12.32%, compared to 25.81% with **SV** (an improvement of 52.2%). In version v4 and 12 images, a multi-view method is able to attain a mean FLPE of 14.91%, compared to 27.10% for **SV** (an improvement of 50.0%).
5. Similarly to Group 1, the best-performing multi-view method depends on the dataset version. However the method rankings are different compared to Group 1:

- **ESAC + Mean SV** and **ESAC + MV** are the best methods for Groups 2 and 3 v1 with little performance difference.
- **MV** is the best method for Groups 2 and 3 v4 by a clear margin.
- **MV** is the best method for Groups 2 and 3 v1 when there are a small number of images (2-3)
- **Mean SV** is doing relatively poorly for Groups 2 and 3 in v1. This contrasts Group 1 v1 where it was among the best method. Because **ESAC + Mean SV** is performing well, the poor performance of **Mean SV** can be explained by the larger amount of outlier images in Groups 2 and 3.

- **ESAC + MV** performs worse than **MV** in v4. This strongly contrasts Group 1 where **ESAC + MV** was the best method.

A clear difference between Group 1 versus Groups 2 and 3 is the performance of **ESAC + MV** and **ESAC + SV**. For Groups 2 and 3, they are clearly performing poorly in v4. Thus, when the influence of noise is much greater, ESAC is struggling to determine the inlier set thanks to a high proportion of outlier images. This contrasts v1, where **ESAC + MV** and **ESAC + SV** are clearly outperforming **MV** and **SV**. Therefore we can conclude that ESAC is working well for low noise levels, because of lower proportion of outlier images. In contrast, when noise is high, there are far fewer inlier images, making it hard for ESAC to identify them.

We illustrate this problem with ESAC in Figure 5.32, where we show the Floral paper and Cap datasets. The Floral paper is strongly creased in approximately half of the images, and smooth in the other half (Figure 5.4). We cannot normally solve fSfT accurately for each creased image: There is neither sufficient motion information from the point correspondences to resolve the creases, nor does the deformable model support creased deformations thanks to its L2 regularization. We can see in Figure 5.32(a) that the existence of these creased images is causing problems for **Mean SV** and **MV** where performance plateaus approximately 7 images. In contrast, **ESAC + SV**, **ESAC + MV** perform well and they have not plateaued at 12 images. This shows they are effectively filtering out the outlier images. **MV** is better than ESAC with 2 images, because we cannot find an inlier set with just 2 images.

However, when we look at Figure 5.32(b) we see that **ESAC + SV** and **ESAC + MV** perform much worse compared to **MV** and their performance actually decreases with more than 7 images. The reason is that the increased noise in version v4 makes it hard to filter out the creased images with ESAC. This is because the smooth images no longer form a tight cluster of focal length estimates, thanks to noise strongly deteriorating the per-image focal length estimates.

We also show results for the Cap Dataset in Figure 5.32(c,d). Here **MV** is the best performing method in v4, and also the best performing method in v1 when the number of images is between 2 and 6. Beyond 6 images, **ESAC + MV** starts to take over. Considering Figure 5.32(d), **ESAC + MV** is consistently behind **MV** for the reasons explained earlier: ESAC is clearly having difficulty to find a good inlier set when there is stronger noise.

5.4.6.4 Individual dataset Results

We give mean FLPE for each dataset individually Table 5.6 (v1) and Table 5.7 (v4). For brevity we only show results for image sets of size 2, 5 and either 10 images or the maximum number of images in a dataset. Numbers in bold indicate the best result among methods for a given dataset and image set size. In v1, there is no winning method as discussed in detail in the previous section. FLPE below 3% is achieved by most methods for Kinect paper, Van Gogh and Hulk datasets, which are excellent results. There are clear reductions in FLPE with the Cap, Kinect t-shirt, handbag, pillow cover, Floral paper and Fortune teller datasets using the multi-view methods.

However, the Bedsheet and Bending Cardboard datasets are problematic: no multi-view method is able to significantly reduce FLPE. Therefore, they can be considered as multi-view fSfT failure modes, where there is something special about the geometry that prevent multi-view being effective. As future work, we aim to study the underlying causes and degeneracies of multi-view fSfT.

Table 5.6: FLPE of multi-view fSfT methods using v1 datasets

Dataset	Number of images	MV	Mean SV	Median SV	SV	ESAC + Mean SV	ESAC + MV
Kinect paper	2	2.92	2.79	2.79	2.89	2.79	2.92
	5	3.00	2.38	2.83	2.89	2.38	3.00
	10	2.99	2.41	2.85	2.89	2.41	2.99
Van Gogh	2	3.21	3.29	3.29	4.97	4.08	4.39
	5	3.22	1.67	2.78	4.97	1.98	3.30
	10	3.10	1.73	2.28	4.97	1.60	3.16
Hulk	2	2.42	2.38	2.38	2.45	2.38	2.42
	5	1.66	1.83	1.56	2.45	1.83	1.66
	10	1.28	1.62	1.10	2.45	1.62	1.28
Cap	2	8.40	12.81	12.81	12.24	11.81	11.63
	5	5.28	10.33	6.24	12.24	6.33	5.59
	10	3.86	9.06	4.16	12.24	3.57	3.05
Bedsheet	2	12.35	10.04	10.04	11.59	11.99	12.91
	5	12.83	9.97	10.90	11.59	11.50	13.07
	10	13.00	9.90	11.30	11.59	11.57	13.22
Kinect t-shirt	2	8.40	8.40	8.40	11.97	11.55	11.55
	5	4.19	5.59	6.17	11.97	5.61	5.40
	10	3.88	5.29	4.18	11.97	4.10	4.30
Handbag	2	11.12	11.69	11.69	11.21	10.43	10.75
	5	8.15	10.57	5.51	11.21	3.80	4.02
	7 (max)	7.37	9.85	4.98	11.21	3.80	3.95
Bending cardboard	2	36.55	34.42	34.42	37.53	37.54	37.60
	5	38.02	36.80	25.13	37.53	27.69	28.05
	10	38.62	39.40	27.04	37.53	25.52	25.96
Pillow cover	2	24.95	33.55	33.55	28.37	28.15	28.45
	5	24.38	41.58	31.34	28.37	23.95	25.03
	9 (max)	18.55	41.61	31.55	28.37	17.21	18.72
Floral paper	2	10.24	12.86	12.86	14.50	14.20	13.66
	5	8.93	12.87	6.70	14.50	5.45	4.45
	10	8.13	11.51	4.55	14.50	2.69	2.68
Fortune teller	2	15.15	19.15	19.15	20.92	19.91	21.30
	5	7.24	10.27	9.09	20.92	8.57	6.40
	6 (max)	6.74	7.00	8.92	20.92	7.60	5.20
Average	2	12.49	13.38	13.38	14.48	14.09	14.39
	5	10.63	13.08	9.84	14.42	9.01	9.09
	10	9.77	12.67	9.36	14.42	7.43	7.68

Table 5.7: FLPE of multi-view fSFT methods using v4 datasets

Dataset	Number of images	MV	Mean SV	Median SV	SV	ESAC + Mean SV	ESAC + MV
Kinect paper	2	6.93	10.22	10.22	14.72	13.82	13.35
	5	4.96	8.37	5.17	14.72	4.87	3.63
	10	3.47	6.80	3.87	14.72	3.80	2.03
Van Gogh	2	4.92	4.25	4.25	5.30	4.25	4.92
	5	5.07	4.23	4.19	5.30	4.23	5.07
	10	4.89	4.19	4.29	5.30	4.19	4.89
Hulk	2	2.03	2.20	2.20	2.85	2.41	2.35
	5	1.13	1.77	1.70	2.85	1.78	1.36
	10	0.75	1.26	1.18	2.85	1.32	0.83
Cap	2	11.90	14.79	14.79	15.13	14.87	14.40
	5	6.13	11.40	9.60	15.13	11.71	10.15
	10	4.57	10.39	5.97	15.13	9.08	7.25
Bedsheet	2	15.04	14.59	14.59	18.79	18.31	18.46
	5	13.80	12.35	15.12	18.79	18.56	17.97
	10	13.61	10.79	13.72	18.79	17.92	17.23
Kinect t-shirt	2	8.78	11.19	11.19	14.29	14.36	14.07
	5	6.59	7.11	7.79	14.29	9.60	9.11
	10	5.17	6.41	6.82	14.29	9.11	8.48
Handbag	2	10.14	12.52	12.52	15.21	14.69	14.54
	5	6.31	9.74	8.03	15.21	5.82	5.36
	8 (max)	5.49	8.64	9.18	15.21	0.16	1.41
Bending cardboard	2	25.89	28.43	28.43	29.56	29.43	29.43
	5	23.22	35.72	22.06	29.56	27.28	27.41
	10	24.44	41.35	18.91	29.56	22.72	23.13
Pillow cover	2	24.22	31.68	31.68	27.70	27.79	27.39
	5	23.74	37.81	31.72	27.70	24.14	22.23
	9 (max)	17.89	36.42	30.76	27.70	14.05	11.74
Floral paper	2	9.11	12.69	12.69	16.14	15.23	14.40
	5	7.67	15.12	7.98	16.14	11.59	9.82
	10	7.53	14.57	7.46	16.14	9.63	7.91
Fortune teller	2	21.48	24.05	24.05	31.57	31.64	31.54
	5	13.36	10.89	10.77	31.57	42.15	40.66
	6 (max)	12.20	7.35	6.77	31.57	49.79	47.39
Average	2	13.18	15.19	15.19	18.30	17.78	17.63
	5	10.18	14.05	11.29	17.39	14.70	13.89
	10	9.09	13.47	9.90	17.39	12.89	12.03

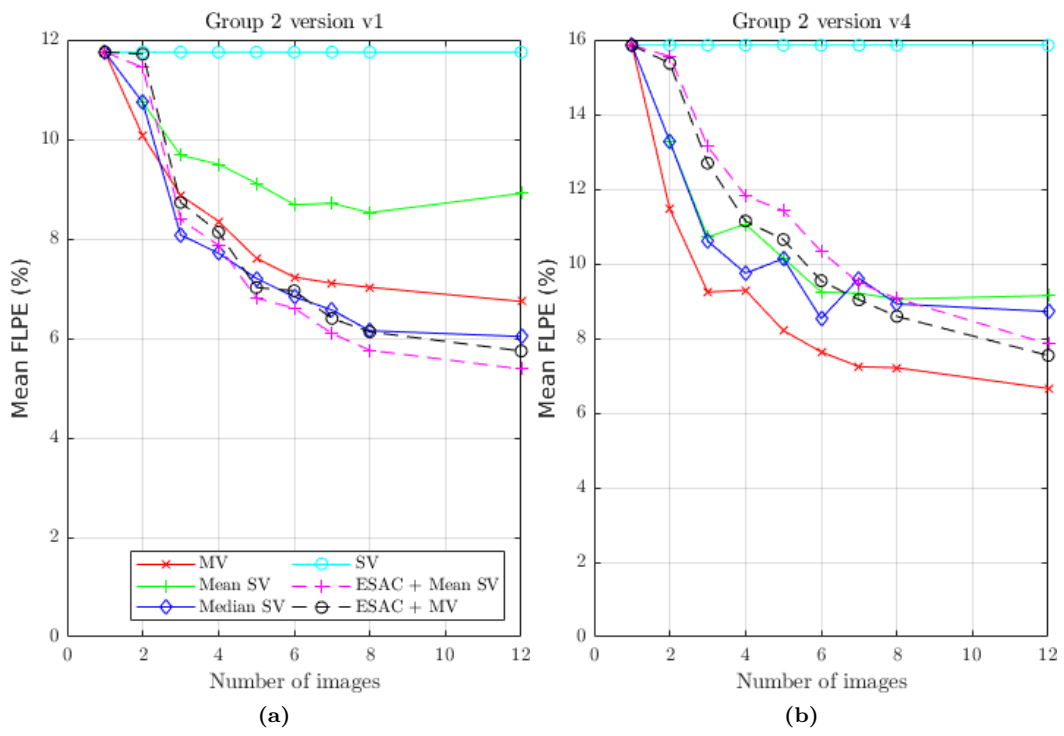


Figure 5.30: Comparison of multi-view fSFT methods using Group 2 mean FLPE

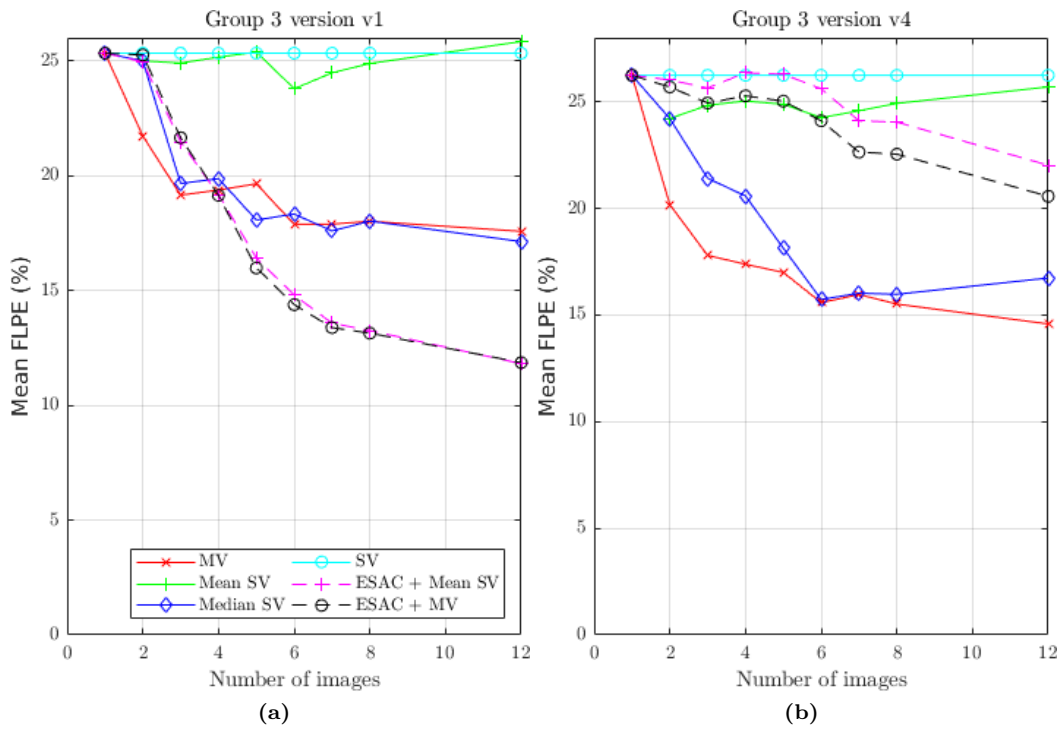


Figure 5.31: Comparison of multi-view fSFT methods using Group 3 mean FLPE

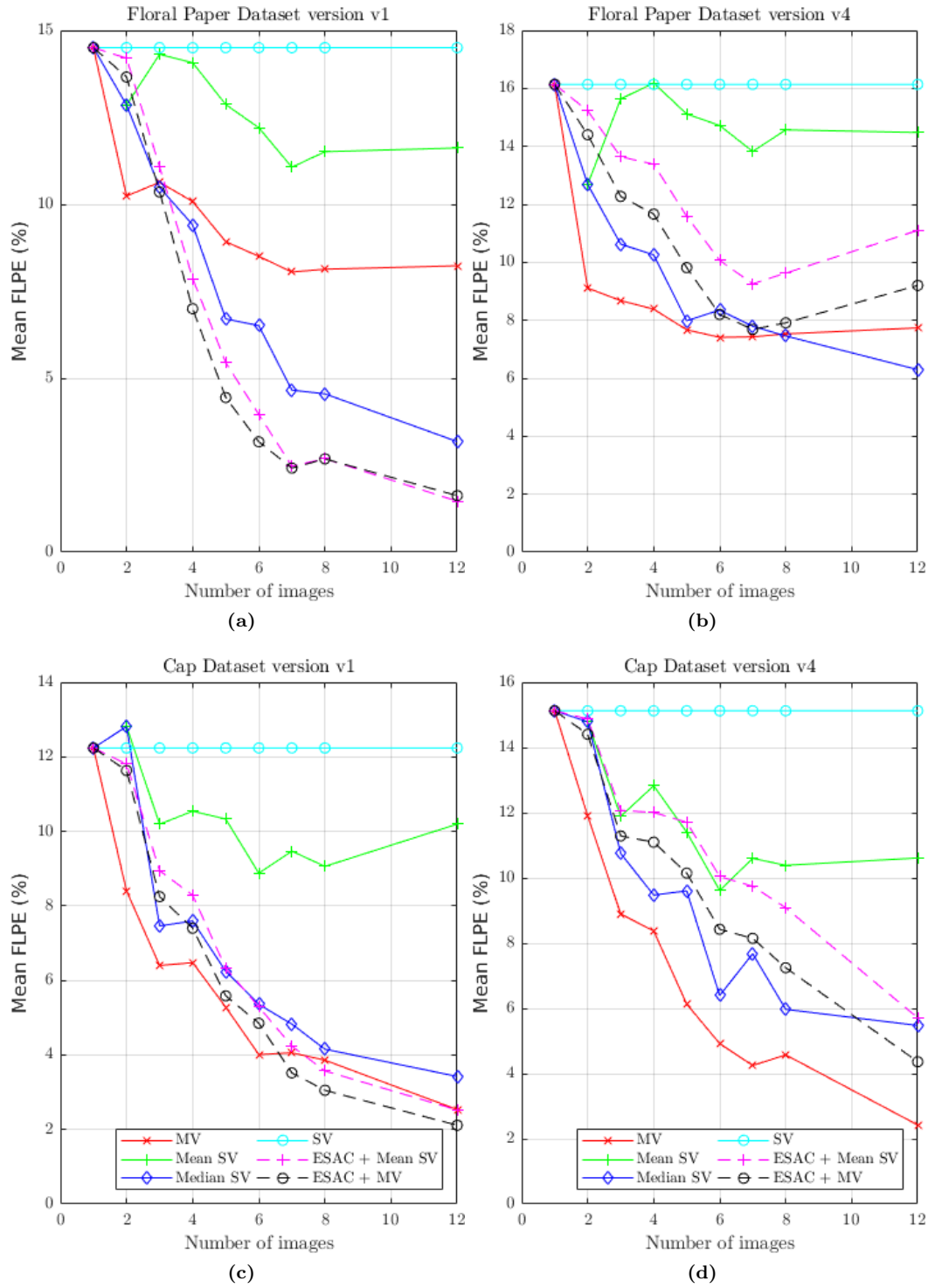


Figure 5.32: Results on original datasets: FLPE

5.4.6.5 Results perspective

The results presented in the previous two sections clearly show that the proposed multi-view fSfT methods can significantly improve focal length estimation accuracy. However, the results also paint a fairly complex picture: there is no clearly winning method in all cases, and the best method depends

on the amount of noise and the dataset. Nevertheless, we can summarize some general findings:

1. Focal length averaging (**Mean SV**) can work well for well-textured, smooth strongly isometric objects with low noise (Group 1). In these cases, focal length can be estimated relatively well for each image independently using our optimization-based method. A simple averaging improves results as the number of images increases, provided the image set has no outlier images. Outlier images are those for which per-image focal lengths have not been estimated well using single-view fSfT.
2. For all other conditions (higher noise, non-isometric objects, complex deformations, sparse correspondences, sets with outlier images), focal length averaging with the mean does not appear to work well and more sophisticated methods are required:
 - When we only have a small number of images (2-3), or when the level of noise is relatively high, the proposed multi-view optimization method (**MV**) works best in general. This is superior to **Mean SV** because it uses all geometric constraints that link unknowns across all images, yet its computational complexity is the same as Mean SV: it is linear in the number of images.
 - When the level of noise is relatively low, and we have more than 3 images, ESAC can identify the subset of inliers images. Given this subset, the focal length can be estimated with either focal length averaging or with the multi-view optimization method. There is little difference in performance between them.

The results also naturally present a new question: how to determine the best focal length solution from the multi-view methods? This is not trivial and it is left for future work. One option could be to take each focal length estimate and test it by solving CSfT on each image. The focal length yielding the lowest cost would then be selected. We have already tested this, but it does not work in general. Specifically, it becomes hard to determine the right focal length if the CSfT method fails to find the global optimum in one or more images, or there are images with high modeling error (e.g. creased surfaces). An alternative approach is to make **MV** more robust to outlier images. This could be implemented using an M-estimator to automatically deactivate the influence of such images on the cost function. This would avoid hard filtering with ESAC.

5.5 Conclusion

This chapter presents various novel contributions and an analysis of the focal length and Shape from Template (fSfT) problem. Firstly, we have shown that single-view fSfT can be solved analytically using continuous differential geometry. This has produced both a practical algorithm suitable in certain cases, and new theoretical insights about fSfT well-posedness. Secondly, we have presented the first optimization-based method for single-view fSfT, which is far superior to the analytical algorithm in terms of accuracy. It uses a carefully-designed cost function that is solved with multi-start iterative optimization. The main innovations are three-fold: *(i)* Cost term normalization, allowing us to use the same isometric weight (a critical hyper parameter) for all datasets. *(ii)* Efficient multi-start optimization that reduces repeated search of solution space. *(iii)* An unsupervised weight selection method that can automatically find the isometric weight simply by inspecting the problem's condition. We have shown that the unsupervised method performs as well as a carefully selected weight found

by supervision and requiring known focal lengths. It also provides a slight performance improvement for higher noise levels. The unsupervised method has no tuning parameters, and it may be applicable for other related problems such as SfT-P or NRSfM.

Thirdly, we have studied and solved multi-view fSfT for the first time. We have presented and compared two kinds of methods: focal length averaging, including robust and non-robust variants, and the multi-view extension of the single-view optimization-based method. We have shown that this can be optimized efficiently with non-linear least squares by exploiting the problem’s sparsity structure with the Schur complement. This has computational cost that is linear in the number of images, and it is approximately the same as solving fSfT individually for each image.

We have compared the single-view and multi-view fSfT methods on 12 public datasets, grouped into three types: Group 1 includes smooth, densely textured and strongly isometric surfaces (bending sheets of paper). These objects have been commonly used to evaluate most SfT-P methods. Group 2 includes non-smooth moderately textured cloth objects, which are less isometric than Group 1. Group 3 includes non-smooth weakly textured objects of paper and cloth, and it represents the hardest type of problem. We have performed an extensive empirical evaluation in the single-view problem with the analytical method and the multi-start optimization-based method. The latter performs well with only 1-3 focal length initializations, and we have shown that computational cost scales sub-linearly in the number of initializations, thanks to our mechanism to avoid repeated search. We have also demonstrated that our cost function is well normalized, which is rarely done in optimization methods for solving related problems such as SfT-P.

For multi-view fSfT, we have show a clear benefit in using multiple images compared to solving fSfT with a single image. The benefit is two-fold: *(i)* to cope with cases that are weakly posed with a single images, such as when the object is approximately fronto-parallel. *(ii)* to improve accuracy in the presence of data noise and modeling error. Of the proposed multi-view fSfT methods, there is no clear winner. the ESAC-based methods have shown value because they can filter out images which have a poor focal length estimate (outlier images). This also has the effect of filtering out images which have a poor deformation estimate, because one is rarely able to solve fSfT with a good focal length and a poor deformation. From the filtered images, a good focal length estimate can be made either by focal length averaging or by running the multi-view optimization method on the inlier set. However, ESAC has limitations because it works poorly when there is a small number of images or when noise is high. In those cases, the multi-view optimization method works best in general and it significantly outperforms focal length averaging for the more challenging datasets in Groups 2 and 3. We have several directions for future work in fSfT. These are discussed in the concluding chapter of this thesis in §6.2.3.2.

Chapter 6

Conclusions and Future Work

Chapter summary

This chapter concludes the thesis. We have tackled three different monocular reconstruction problems. While these problems are different, our contributions in each chapter share three central themes: (i) creating and solving closed-form solutions from motion data, (ii) analyzing the problem's solvability and well-posedness from closed-form solutions, and (iii) refining closed-form solutions with non-convex optimization (the 'initialize-then-refine' paradigm). We first discuss these themes in a general perspective. We then discuss future research challenges and opportunities originating from our work.

6.1 Review of common thesis themes

6.1.1 Theme 1: Closed-form solutions using motion model relaxation

Novel closed-form solutions have been presented in this thesis for the PPE-P, fSfT and PSfM-O problems. Although different, the solutions have the same common theme: They divide the problem into two steps: (1) estimation of image motion data (displacements and higher-order derivatives) and (2) estimation of unknown geometry from the motion data. In step 1, we have estimated motion data from point correspondences using a general motion model: In PPE-P, the general model is an 8 DoF homography, in PSfM-O it is a general 6 DoF 2D affine transform, and in fSfT it is a general 2D warp. These models are fitted to point correspondences using standard methods that have closed-form solutions. In step 2, motion data is extracted from the general motion model coefficients and used to solve the problem’s unknown geometry. Step 2 is equivalent to estimating the problem’s *specialized motion model*. In the case of PPE-P, the specialized model is a 6 DoF homography that encodes the camera pose. In the case of PSfM-O, it is a specialized affine transform with 5 unknown DoFs that encodes the camera pose (the 6th pose DoF corresponds to the structure’s depth and it is not recoverable due to orthographic projection). In the case of fSfT, it is a specialized local 2D warp with 7 unknown DoFs that encodes the pose of the tangent plane (6 DoFs) and the camera’s focal length. Our approach to each problem uses motion model relaxation because motion data is first estimated using the general motion model that may not respect all available geometric constraints. The key advantage to this approach is that both steps (motion data estimation and geometry estimation) are solved in closed-form.

However, the fact that not all geometric constraints are used to estimate motion data in the first step can be a weakness. Indeed, it is one of the reasons why the closed-form solutions are not statistically optimal in the ML sense. We have nonetheless found that our closed-form solutions for PSfM-O and PPE-P can rival the accuracy of the ML estimate, implemented with iterative optimization, in certain cases. For PPE-P, our IPPE method can achieve very similar accuracy compared to the optimized solution when the object points are arranged regularly (the corners of a square). This is a common use case for estimating the pose of AR markers, and we have found no significant improvement by refining the solution with optimization. Eliminating the need for optimization has clear benefits for an application’s run-time speed especially in cases where the poses of multiple AR markers are required. For PSfM-O and in the special case of 3 views, we have shown that our closed-form solution gives the ML estimate (Theorem 6). In general, the solution is not statistically optimal with more than 3 views, yet we found its accuracy is very similar to the optimized solution using Bundle Adjustment, rendering the need for iterative optimization questionable.

In contrast, the fSfT closed-form solution is only accurate in certain cases when the general motion model (a 2D warp) can be estimated well, which requires dense point correspondences and smooth deformation. The iteratively optimized fSfT solution gives in general far better results compared to the closed-form solution. We explain the significantly lower performance of the fSfT closed-form solution because the general motion model is much more relaxed compared to PPE-P and PSfM-O. In PPE-P, the general model has 2 extra DoFs compared to the specialized model. In PSfM-O, the general model has only 1 extra DoF compared to the specialized model. In fSfT, the general model has been implemented with a regularized Thin-Plate Spline with 9 control points hence 18 DoFs, with

11 more DoFs than the specialized motion model. In future work we aim to test different general models with fewer DoFs than the TPS, to reduce the ability of the general model to fit to noise. This may be possible with a learning-based approach using a CNN, and it may considerably improve the accuracy of the fSfT closed-form solution.

6.1.2 Theme 2: Theoretical problem analysis from closed-form solutions

In a second common theme, we present novel theoretical contributions to better understand the reconstruction problems. Specifically, we have investigated the following questions:

1. Under what geometric conditions does a problem have a unique solution?
2. If a problem has a finite solution set, how are the solutions related geometrically?
3. If a problem uses the perspective camera, how is it altered when the perspective effects diminish, leading to affine projection?
4. Does an algorithm implementation for solving the problem work in all instances (is it NADA¹)? Specifically, if the problem is theoretically solvable up to a finite number of solutions, is the algorithm guaranteed to find the solution(s)? If it cannot, the algorithm has a flaw and is said to have an *artificial degeneracy*.

Complete answers to the above questions can be difficult to obtain in most reconstruction problems. However, the answers are important to fully understand the reconstruction problems at hand, and to detect and diagnose issues with a specific algorithm. A failure to do this can lead to unforeseen problems when the algorithm is used in real-world applications. For example, in the related problem of PnP, the implementations of two algorithms (DLS and UPnP) were integrated in OpenCV but they have been subsequently removed because they were later discovered by users to fail in solvable cases (Question 4, they are not NADA algorithms). This was significant because PnP is a fundamental reconstruction problem required both within OpenCV and by users of OpenCV, and the algorithm failures could neither be explained nor predicted.

We already have fairly complete answers regarding items 1-3 for PPE-P from previous works [SM99; Zha00]. The main benefits of our analytical PPE-P solution are practical (accuracy, speed, simplicity and stability in quasi-affine conditions). We have also guaranteed that it is NADA, so it can be relied on in real-world use without unforeseen failure cases. In contrast, we have made significant advances to answer these theoretical questions for PSfM-O and fSfT, summarized in §1.3.1.2 and §1.3.3.2 respectively.

6.1.3 Theme 3: Solution refinement with non-convex optimization

In a third common theme of this thesis, we solve the reconstruction problems with non-convex iterative optimization using cost functions with the same general form: they are weighted summations of M-estimators that can be efficiently optimized with quadratic convergence using Levenberg-Marquardt or Gauss-Newton, and implemented with iterative re-weighted linear least squares. Iterative optimization requires suitable initialization, which is provided by the closed-form solutions. For the rigid problems (PPE-P and PSfM-O), iterative optimization is straightforward using existing tools. However, for

¹We recall that NADA stands for a Non-Artificially Degenerate Algorithm, which is equivalent to an algorithm that can theoretically solve all problems for which a solution exists.

fSfT, iterative optimization is not trivial because of the much larger problem space and difficulties associated with problem modeling and cost function design with a deformable object. We have made important contributions in these aspects for single and multi-view fSfT summarized in §1.3.3.2.

6.2 Future directions of research

In this section we discuss some future research directions for each problem stemming from this thesis. We first discuss how the problems have advanced since our work was published and we then discuss the important open research objectives.

6.2.1 Plane-based Structure from Motion with Affine cameras

6.2.1.1 Recent problem advances

Since our work in Chapter 4 was published in 2017 in PAMI [CB17], there has been relatively little advances on the problem. Because our solution to PSfM-O appears very close to that achieved with bundle adjustment, there appears to be limited room for improving the solution. Our theoretical understanding of PSfM-O is now very complete thanks to this work. There have been two relevant recent advances that are worth discussing on related problems. Firstly, deep learning-based methods are today state-of-the-art for densely registering images. Methods such as HomographyNet [DMR16] could easily be adapted to work with a 6 DoF affine model, and therefore motion from these CNN-based methods could be inputted into our method. This would make it more applicable when texture is poor and difficult to register with feature-based approaches. Secondly, our method first computes metric structure and then it computes camera poses by solving PPE-O for each view. Recently, we have proposed a very efficient closed-form PPE-O solution that is statistically optimal in the ML sense [BC18]. This solution should now be used to solve PPE-O instead of our initially proposed method which, although statistically optimal, is much slower.

6.2.1.2 Future research objectives

There are several open research objectives that we would like to pursue as follow-up work. We mention 4 of them.

1. Evaluate solutions with motion estimated with deep learning. We have mentioned this point earlier and it is straightforward to test. The benefit would be to significantly enlarge the algorithm’s usefulness to weakly textured surfaces for which feature-based matching fails.

2. Implement and test the extension to solve PSfM-PP. We have shown in Theorem 7 that we can solve PSfM with a para-perspective camera (PSfM-PP) in certain cases. For example, if we have an intrinsically calibrated perspective camera and the depth of the structure is approximately constant in 3 or more views (causing the image magnification factor to be approximately constant), PSfM-PP can be solved. Because the para-perspective camera has less modeling error compared to the orthographic camera, the solution could be more accurate. We aim to implement and test this idea.

3. Understand accuracy and limits with quasi-planar structures. We can apply our algorithm to structures that are not perfectly planar. For example, if the structure has some curvature or relief. In many applications such as ground plane rectification, a planar reconstruction can be sufficient. Nevertheless, the amount of non-planarity will affect solution accuracy, and there will be a tipping point when solutions that require and exploit non-planar structure will start to work better. In our experimental evaluation, the tested structures had negligible curvature and relief, for which those methods completely failed. We aim to extend our evaluation to understand the effect of non planarity on accuracy of our method, and to identify when the tipping point occurs.

4. Solving IsoSfM. As discussed in §2.2, we can convert IsoSfM to multi-instance PSfM by dividing the surface into patches and assuming that the curvature of each patch is negligible. This is valid for surfaces with smooth, low-frequency deformation such as sections of cloth or paper, and the problem can be solved by applying PSfM to each patch. This was done using the orthographic camera by [TJK10] with triangular patches, who showed it can work well even for torn surfaces. We have demonstrated that our solution considerably outperforms [TJK10] for solving PSfM-O because it does not make a linear relaxation. This indicates it could also produce better results than [TJK10] for solving IsoSfM, and it is also not limited to using triangular patches (patches with an arbitrary number of 3 or more points could be used). Furthermore, we aim to investigate two more extensions: using para-perspective camera models with a constant depth assumption to improve accuracy further, and using dense non-rigid motion predicted by a CNN-based method such as FlowNet [Dos+15] or its more recent improvements such as PWC-Net [Sun+18] and IRR-PWC [HR19].

6.2.2 Perspective Plane-based Pose Estimation

6.2.2.1 Recent problem advances

Since IPPE was published in IJCV in 2014 [CB14a], researchers have continued to search for faster and more accurate methods to solve rigid pose estimation with planar or non-planar structures along two main directions. The first direction is with deep learning-based methods, originating with PoseNet [KGC15] in 2015 and more recently with [Xia+18; Luo+18; Mel+17a; NB17; WMH17; Bra+18; TSF18; PPV19; Pen+19; ZSI19; LWJ19; CJQ20]. Of these methods, two main categories have emerged: the first category are pose regression networks such as PoseNet that directly predicts pose from an RGB image using a CNN-based encoder/decoder. The second category such as [TSF18; RL17; PPV19; Pen+19; ZSI19; LWJ19; CJQ20] are hybrid methods that combine a CNN to solve image registration (with either sparse or dense matching) and a PnP method to estimate pose from the image registration. The second category has some important advantages. It may not require training the CNN for a specific object or scene, and it handles cases when there are multiple pose solutions. Recall that this can commonly occur when the object or scene is planar and when the perspective effects diminish, leading to a two-fold pose ambiguity. Pose ambiguities are not handled by current state-of-the-art pose regression networks such as because they return only a single solution. As a consequence, IPPE may still be valuable today as a fast method to estimate a plane’s pose or poses from a registration computed by a CNN.

The second main direction developed in parallel with the CNN-based approaches are faster methods to solve PnP (and PPE-P as a special case) by minimizing the object-space error in closed-form using Gröbner bases. Notable methods are OPnP [Zhe+13] (submitted concurrently with IPPE), UPnP [KLS14], optDLS [Nak15] and GAPS [Wie+18]. The fastest methods are UPnP and GAPS, where

the computation time using an efficient C++ implementation is approximately 0.5 ms with 4 points, growing to approximately 1 ms with 1000 points [Wie+18]². These methods can achieve very accurate results that are more accurate than IPPE for general co-planar point configurations. However, they have some limits that make IPPE still competitive and relevant today. Firstly, they are not as accurate as iterative optimization with LM as shown in [Wie+18]. Consequently, LM remains the gold standard approach for solving PnP and PPE-P with highest accuracy today. Therefore, the good accuracy and very low computational cost of IPPE can make it a better choice for initializing LM compared to these more expensive methods, especially with a small number of points. In the case of 4 points the computational cost of IPPE on similar hardware is approximately 1 μ s (\approx 500 times faster than UPnP and GAPS). Secondly, for regularly distributed points, IPPE is practically as accurate as the LM solution. Thirdly, it is practically impossible to prove that those methods do not have artificial degeneracies because there is no analytical relationship between the solution and inputs. Some of these methods (DLS and UPnP) have been removed from OpenCV because there exist solvable cases that they cannot solve (artificial degeneracies), as mentioned in §6.1.2. In contrast, IPPE is NADA. For these reasons, IPPE still has advantages compared to these more recent methods.

6.2.2.2 Future research directions

There are several open research objectives that we would like to pursue as follow-up work. We mention 4 of them.

1. Understand why IPPE performs so well in the case of regularly distributed points.

There is clearly a relationship between the spatial configuration of the object points and pose accuracy. When the points are arranged as 4 corners of a square, pose accuracy is generally higher compared to other spatial configurations such as corners of a thin rectangle. This can be explained by better problem conditioning. This is intrinsic to the PPE-P problem and it is thus not a property of the IPPE algorithm. However, compared to other spatial arrangements, the square arrangement appears to make the IPPE solution extremely close to the ML estimate, which *is* a property of the IPPE algorithm. This is not true of all PPE-P methods, for example the previous homography decomposition methods of [SM99] and [Zha00]. We would like to answer why this is the case and to understand the relationship between point spatial configuration and the gap between the IPPE solution and the ML estimate. We expect this can be achieved with perturbation analysis but it is not trivial.

2. Extend IPPE to solve fPPE.

fPPE has an analytical solution that is similar to previous analytical solutions for PPE-P. First, the general 8 DoF model-to-image homography is estimated, then it is decomposed into pose (6 DoFs) and focal length (1 DoF) [Zha00]. Similarly to PPE-P, the decomposition is not exact because the general homography has 8 DoFs. We have shown that IPPE gives a better way to decompose the homography for PPE-P because of how it considers the effect of noise in the homography coefficients. It may be possible to extend IPPE to also solve fPPE analytically with better accuracy.

3. Improve RANSAC efficiency using IPPE.

RANSAC and its variants remain today among the best approaches to detect and remove wrong correspondences in PnP and PPE-P. The gold standard approach to estimate the inlier set is with minimal random sampling (3 points and P3P),

²Timing information was reported by [Wie+18] with C++ implementations executed on a single threaded 3.5 GHz PC

which may return between 0 and 4 solutions. Points are re-sampled if there is no solution, and if there is more than one solution a 4th point is randomly sampled to disambiguate pose. Pose consensus is then measured with all other points. We believe that we can improve the efficiency of this process by replacing P3P with IPPE when a 4th point is required to disambiguate pose. This is because IPPE uses all 4 points to estimate pose, and therefore the effect of noise is reduced compared to using only 3 points. This may result in fewer RANSAC iterations and consequently a more efficient solution to PPE-P with mismatched points. We believe IPPE is the ideal choice among PPE-P methods for use inside RANSAC because of its extremely low computational cost and good accuracy.

4. Evaluate IPPE with motion from a CNN-based method. We have tested IPPE using point correspondences from classical keypoint matching methods such as SIFT and SURF, which were state-of-the-art when IPPE was published. However, they have been superseded by CNN-based methods and in particular dense methods such as HomographyNet [DMR16] that directly regress the position of 4 point correspondences from two view of an arbitrary planar scene. We aim to evaluate the performance of IPPE compared to other methods using this motion data, which may have different noise characteristics.

6.2.3 Focal length and Shape-from-Template

6.2.3.1 Recent problem advances

Since the work in fSfT was published in 2013 in CVPR [BPC13] and ICCV [BC13a], we are not aware of any publications showing an improvement to the optimization-based fSfT method, nor an improvement on the closed-form solution. This thesis includes considerable extensions on those conference papers and we will be submitting them to a computer vision journal shortly. There has been some recent progress in the related problem of fIsoSfM [PBP18; Pro+18]. [Pro+18] solves fIsoSfM by relaxing the isometric assumption to the inextensibility assumption: It permits the surface to shrink but not stretch between views, allowing the problem to be modeled and solved as an SOCP problem. [PBP18] takes a very different approach and it solves fIsoSfM by approximating it as fPSfM with local planar models. This is solved using locally computed inter-view homographies derived from deformable warps derivatives. These works are very promising, having moved NRSfM forward to the uncalibrated setting. However, fIsoSfM cannot be considered solved because they make significant problem relaxations that can reduce accuracy: [Pro+18] relaxes isometry with inextensibility and [PBP18] ignores constraints acting across the whole surface.

6.2.3.2 Future research objectives

There are several open research objectives that we would like to pursue as follow-up work. We mention 7 of them.

1. Develop one multi-view method that works best in all cases. We have presented several multi-view fSfT methods using either robust focal length averaging or multi-view optimization. We have shown considerable improvement compared to single-view fSfT, but no method works best in all cases. The multi-view optimization-based method appears to work best provided that there are few outlier views. These are views where either initialization is poor or with an especially high modeling error, such as a view of a strongly folded surface that is not handled well by the deformation model (*e.g.* the Fortune teller dataset). We have proposed a basic robustification approach using RANSAC

that helps but it does not improve results in all cases. We aim to investigate better approaches that adapt the cost function to automatically recognize or reduce the influence of outlier views on the cost function.

2. Extend the optimization-based method to incremental and real-time fSfT. In many real-world applications, a camera’s focal length may be fixed over some portions of a live video then undergo smooth changes from mechanical or digital zoom. Currently this is not handled by our multi-view solutions. We aim to handle this by extending to continuous and real-time fSfT, including focal length change detection. This is a challenging problem and we will combine our work with ideas from continuous camera calibration with rigid objects such as [KS15], with for example Bayesian filtering or using an incremental version of the multi-view optimization method.

3. Extend the optimization-based method to more unknown camera intrinsics. We have shown that our multi-view optimization-based method can be applied to different unknown and fixed camera intrinsics in §5.3.3. The method is efficient for any combination of known/unknown intrinsics with linear computational cost in the number of views. We aim to evaluate this approach with different combinations, which will require constructing new datasets with different intrinsics, including significant lens distortion, since this is not available with existing public datasets. We also aim to gain empirical insights into which camera intrinsics can be reliably estimated. We will also assess the benefit of adding priors on the unknown intrinsics.

4. Solve fSfT and other uncalibrated problems in closed-form without the weak-perspective approximation. Our closed-form solution approximates perspective projection with a local weak-perspective approximation. This is valid for surface regions that either project close to the optical center or when the tilt angle of the surface is not very strong. We have since studied the equations without the weak-perspective approximation, however an analytical solution seems very challenging. It may be possible to solve with numerical root finding using Grobner bases, and we aim to study this in future work. Furthermore, it may be possible to extend this to handle an additional unknown intrinsics, such as a 1-parameter radial distortion [Fit01]. Nevertheless, we believe that additional unknown intrinsics such as this, unknown principal point or higher-order distortion terms will be very hard to solve with a closed-form method: It will only be feasible using a small surface region (to keep the number of unknowns down), which will in turn lead to poor problem conditioning. In contrast, it could be possible to solve additional unknown intrinsics using our multi-view optimization-based method because it applies all available geometric constraints from the template, connected over space (*i.e.* the whole of the template’s surface) and time (*i.e.* different views).

5. Complete the theoretical analysis of fSfT and other uncalibrated problems. From our experimental evaluation, it appears that some non-trivial scene geometries make fSfT very weakly posed or ill-posed. We experienced this with the Cardboard dataset, where there exists several images in the dataset with strong deformation that could not be solved accurately. Furthermore, we found that adding constraints from multiple images with the Cardboard dataset did not significantly improve accuracy, unlike the other datasets. We would like to know why this occurred, and ultimately to uncover the necessary and sufficient geometric conditions for fSfT to be well-posedness with either a single image or multiple images. What makes this especially challenging is that well-posedness is a

function of both deformation and the spatial layout of correspondences. The extension of this analysis to other unknown intrinsics could also be investigated.

6. Evaluate solutions with motion from a CNN-based method. Similarly to PPE and PSfM, we have estimated motion information for fSfT using point correspondences with classical feature-based methods. For surfaces with dense distinct texture, this approach works well, but for weakly textured objects this puts a fundamental limit on solution accuracy. In those cases, we expect to achieve significantly better fSfT results using dense deformable CNN-based image registration.

7. Use the proposed solution for supervised CNN-based uncalibrated SfIT. There have been some recent attempts to solve SfIT with a CNN trained end-to-end *e.g.* [Fue+18]. The main limit of these methods is to require labeled training data, usually acquired from simulated images that lack realism, commonly referred to as the ‘render gap’. We have shown that our proposed optimization-based approaches can achieve very accurate results especially in the multi-view setting. We believe it could therefore be used to supply real labeled data to train an fSfT CNN-based method. This would complement simulated data by adding realism and without requiring known focal lengths or 3D deformations from real data. We could apply it in two settings. Firstly, in a supervised setting with a robust loss to handle erroneous labels. Secondly, in an unsupervised setting, where our carefully designed cost function would serve as the unsupervised loss function.

Appendices

Appendices

A.1 Chapter 1 appendices

A.1.1 Published works at EnCoV

All the peer-reviewed published research articles that were co-authored by me from my time at EnCoV are listed below. We include articles that were published during my working contracts between 2009 and 2016 at EnCoV, articles from the Ph.D thesis of Mathias Gallardo whom I co-supervised at EnCoV and then at IRCAD with Prof. Bartoli as thesis director, and articles that were published since I left EnCoV which originated from my research at EnCoV. Articles marked by '-' indicate those whose contributions are included in the thesis.

International peer-reviewed journal articles

[Bar+12a] Adrien Bartoli, Toby Collins, Nicolas Bourdel, and Michel Canis. “Computer assisted minimally invasive surgery: is medical computer vision the answer to improving laparosurgery?” In: *Medical Hypotheses* 79.6 (2012), pp. 858–863

- [CB14a] T. Collins and A. Bartoli. “Infinitesimal plane-based pose estimation”. In: *International Journal of Computer Vision* 109.3 (2014), pp. 252–286

Forms Chapter 4 of this thesis.

[Bar+15] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. “Shape-from-Template”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.10 (2015), pp. 2099–2118

[BCP15] Adrien Bartoli, Toby Collins, and Daniel Pizarro. “Metric Corrections of the Affine Camera”. In: *Computer Vision and Image Understanding* 135.C (2015), pp. 141–156

[Bou+16a] N Bourdel, T Collins, D Pizarro, P Chauvet, C Debize, A Bartoli, et al. “First Use of Augmented Reality in Gynecology”. In: *Journal of Minimally Invasive Gynecology* 23.7 (2016), pp. 226–227

[Bou+16b] N. Bourdel, T. Collins, Daniel Pizarro-Perez, A. Bartoli, D. Da Inès, B. Perreira,

et al. “Augmented reality in gynecologic surgery: evaluation of potential benefits for myomectomy in an experimental uterine model”. In: *Surgical endoscopy* 31 (2016), pp. 456–461

[Bou+17] Nicolas Bourdel, Toby Collins, Daniel Pizarro, Clement Debize, Anne-sophie Grémeau, Adrien Bartoli, et al. “Use of augmented reality in laparoscopic gynecology to visualize myomas”. In: *Journal of Fertility and Sterility* 107.3 (2017), pp. 737–739

[Chh+17b] Ajad Chhatkuli, Daniel Pizarro, Adrien Bartoli, and Toby Collins. “A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.5 (2017), pp. 833–850

- [CB17] T. Collins and A. Bartoli. “Planar Structure-from-Motion with Affine Camera Models: Closed-Form Solutions, Ambiguities and Degeneracy Analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1237–1255

Forms Chapter 3 of this thesis.

[Cha+18] Pauline Chauvet, Toby Collins, Clement Debize, Lorraine Novais-Gameiro, Bruno Pereira, Adrien Bartoli, et al. “Augmented reality in a tumor resection model”. In: *Surgical endoscopy* 32.3 (2018), pp. 1192–1201

[BC18] Adrien Bartoli and Toby Collins. “Plane-Based Resection for Metric Affine Cameras”. In: *Journal of Mathematical Imaging and Vision* 60.7 (2018), pp. 1037–1064

[Gal+20] Mathias Gallardo, Daniel Pizarro, Toby Collins, and Adrien Bartoli. “Shape-from-template with curves”. In: *International Journal of Computer Vision* 128.1 (2020), pp. 121–165

[Col+21] T. Collins, D. Pizarro, S. Gasparini, N. Bourdel, P. Chauvet, M. Canis, et al. “Augmented Reality Guided Laparoscopic Surgery of the Uterus”. In: *IEEE Transactions on Medical Imaging* 40.1 (2021), pp. 371–380

A.1.1.1 International peer-reviewed conference articles

[Col+10] Toby Collins, Jean-Denis Durou, Pierre Gurdjos, and Adrien Bartoli. “Single view perspective shape-from-texture with focal length estimation: A piecewise affine approach”. In: *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2010)

[MCB11] Abed Malti, Toby Collins, and Adrien Bartoli. “Template-Based Deformable Shape-from-Motion from Registered Laparoscopic Images”. In: *Conference on Medical Image Understanding and Analysis (MIUA)* (2011)

[CCB11] Toby Collins, Benoit Compte, and Adrien Bartoli. “Deformable Shape-From-Motion in Laparoscopy using a Rigid Sliding Window.” In: *Conference on Medical Image Understanding and Analysis (MIUA)*. 2011, pp. 173–178

[Bar+12b] Adrien Bartoli, Y. Gérard, F. Chadebecq, and Toby Collins. “On template-based

reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012

[CB12b] Toby Collins and Adrien Bartoli. “Towards live monocular 3D laparoscopy using shading and specular information”. In: *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)*. 2012, pp. 11–21

[CB12a] Toby Collins and Adrien Bartoli. “3D reconstruction in laparoscopy with close-range photometric stereo”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2012, pp. 634–642

[MBC12b] Abed Malti, Adrien Bartoli, and Toby Collins. “Template-Based Conformal Shape-from-Motion-and-Shading for Laparoscopy”. In: *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)*. 2012, pp. 1–10

- [BPC13] A. Bartoli, D. Pizarro, and T. Collins. “A Robust Analytical Solution to Isometric Shape-from-Template with Focal Length Calibration”. In: *International Conference on Computer Vision (ICCV)*. 2013

Chapter 5 includes the analytical solution from this work.

- [BC13a] A. Bartoli and T. Collins. “Template-Based Isometric Deformable 3D Reconstruction with Sampling-Based Focal Length Self-Calibration”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013

Chapter 5 includes the optimization-based solution from this work with major extensions.

[PBC13] D. Pizarro, A. Bartoli, and T. Collins. “Isowarp and Conwarp: Warps that Exactly Comply with Weak Perspective Projection of Deforming Objects”. In: *British Machine Vision Conference (BMVC)*. 2013

[Chh+14] Ajad Chhatkuli, Adrien Bartoli, Abed Malti, and Toby Collins. “Live image parsing in uterine laparoscopy”. In: *International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2014, pp. 1263–1266

[dOr+14] Laurent d’Orazio, Adrien Bartoli, Andre Baetz, Sylvain Beorchia, Gaëlle Calvary, Yahia Chabane, et al. “Multimodal and multimedia image analysis and collaborative networking for digestive endoscopy”. In: *IRBM* 35.2 (2014), pp. 88–93

[Col+14] Toby Collins, Daniel Pizarro, Adrien Bartoli, Michel Canis, and Nicolas Bourdel. “Computer-assisted laparoscopic myomectomy by augmenting the uterus with pre-operative MRI data”. In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. 2014, pp. 243–248

[CB14b] Toby Collins and Adrien Bartoli. “Using Isometry to Classify Correct/Incorrect 3D-2D Correspondences”. In: *European Conference on Computer Vision (ECCV)*. Cham: Springer International Publishing, 2014, pp. 325–340

[CMB14] Toby Collins, Pablo Mesejo, and Adrien Bartoli. “An analysis of errors in graph-based keypoint matching and proposed solutions”. In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 138–153

[PCB15] Kristina Prokopetc, Toby Collins, and Adrien Bartoli. “Automatic detection of the uterus and fallopian tube junctions in laparoscopic images”. In: *Information Processing in Medical Imaging (IPMI)*. 2015, pp. 552–563

[Gal+15] Mathias Gallardo, Daniel Pizarro, Adrien Bartoli, and Toby Collins. “Shape-from-template in flatland”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2847–2854

[Col+15] Toby Collins, Adrien Bartoli, Nicolas Bourdel, and Michel Canis. “Segmenting the uterus in monocular laparoscopic images without manual input”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, pp. 181–189

[Par+15] Shaifali Parashar, Daniel Pizarro, Adrien Bartoli, and Toby Collins. “As-Rigid-as-Possible Volumetric Shape-from-Template”. In: *International Conference on Computer Vision (ICCV)*. USA, 2015, pp. 891–899

- [CB15] T. Collins and A. Bartoli. “Realtime Shape-from-Template: System and Applications”. In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. 2015

Chapter 5 includes the GPU speedups and dimensionality reduction technique from this work for faster iterative optimization.

[Chh+16] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. “Inextensible Non-Rigid Shape-from-Motion by Second-Order Cone Programming”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016

[Col+16a] Toby Collins, Adrien Bartoli, Nicolas Bourdel, and Michel Canis. “Robust, Real-Time, Dense and Deformable 3D Organ Tracking in Laparoscopic Videos”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2016

[GCB16a] M. Gallardo, T. Collins, and A. Bartoli. “Can we Jointly Register and Reconstruct Creased Surfaces by Shape-from-Template Accurately?” In: *European Conference on Computer Vision (ECCV)*. 2016

[GCB16b] M. Gallardo, T. Collins, and A. Bartoli. “Using Shading and a 3D Template to Reconstruct Complex Surface Deformations”. In: *British Machine Vision Conference (BMVC)*. 2016

[GCB17] M. Gallardo, T. Collins, and A. Bartoli. “Dense Non-Rigid Structure-from-Motion and Shading with Unknown Albedos”. In: *International Conference on Computer Vision (ICCV)*. 2017

A.1.1.2 International peer-reviewed workshop articles

[CB10a] Toby Collins and Adrien Bartoli. “Locally Planar and Affine Deformable Surface Reconstruction from Video”. In: *International Workshop on Vision, Modeling and Visualization (VMV)*. 2010

[MBC11] Abed Malti, Adrien Bartoli, and Toby Collins. “A pixel-based approach to template-based monocular 3D reconstruction of deformable surfaces”. In: *International Conference on Computer Vision Workshops*. 2011, pp. 1650–1657

[Kim+12] Jae-Hak Kim, Adrien Bartoli, Toby Collins, and Richard Hartley. “Tracking by detection for interactive image augmentation in laparoscopy”. In: *International Workshop on Biomedical Image Registration*. 2012, pp. 246–255

[Col+16b] Toby Collins, Pauline Chauvet, Clement Debize, Daniel Pizarro, Adrien Bartoli, Michel Canis, et al. “A System for Augmented Reality Guided Laparoscopic Tumour Resection with Quantitative Ex-vivo User Evaluation”. In: *Computer-Assisted and Robotic Endoscop (CARE)*. 2016

A.1.1.3 French peer-reviewed conference articles

[Cha+12] François Chadebecq, Christophe Tilmant, Peyras Julien, Toby Collins, and Adrien Bartoli. “Estimation de l'échelle en coloscopie monoculaire par quantification du flou optique: étude de faisabilité”. In: *Reconnaissance des Formes et Intelligence Artificielle (RFIA)*. 2012

A.2 Chapter 2 appendices

A.2.1 Failure of Zhang’s method with affine motion

Let $\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$ be a square block-diagonal matrix, $\mathbf{A} = \mathbf{U}_A \mathbf{\Sigma}_A \mathbf{V}_A^\top$ be the SVD of \mathbf{A} , $\mathbf{B} = \mathbf{U}_B \mathbf{\Sigma}_B \mathbf{V}_B^\top$ be the SVD of \mathbf{B} and $\mathbf{C} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^\top$ be the SVD of \mathbf{C} . Then

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_B \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_B \end{bmatrix} \begin{bmatrix} \mathbf{V}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_B \end{bmatrix}^\top \quad (\text{A.1})$$

The SVD of \mathbf{C} is $\mathbf{U}_C = \begin{bmatrix} \mathbf{U}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_B \end{bmatrix}$ and $\mathbf{V}_C = \begin{bmatrix} \mathbf{V}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_B \end{bmatrix}$, which are unitary and $\mathbf{\Sigma}_C = \begin{bmatrix} \mathbf{\Sigma}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_B \end{bmatrix}$ which is diagonal and non-negative.

Without loss of generality let b be positive (implying the surface is front-facing to the camera). Assigning $\mathbf{B} \leftarrow b$, we have $\mathbf{U}_B = \mathbf{1}$, $\mathbf{V}_B = \mathbf{1}$ and $\mathbf{\Sigma}_B = b$. Consequently, the SVD of a matrix in the

form of Equation (2.12) is

$$\hat{\mathbf{R}}_1 = \begin{bmatrix} & 0 \\ \mathbf{A} & \\ & 0 \\ 00 & b \end{bmatrix} = \begin{bmatrix} \mathbf{U}_A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Sigma_A & \mathbf{0} \\ \mathbf{0} & b \end{bmatrix} \begin{bmatrix} \mathbf{V}_A & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^\top \quad (\text{A.2})$$

The rotation estimate is then given by

$$\hat{\mathbf{R}} = \begin{bmatrix} \mathbf{U}_A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{V}_A & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^\top = \begin{bmatrix} \mathbf{U}_A \mathbf{V}_A^\top & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (\text{A.3})$$

which is a rotation about the z -axis.

A.2.2 Decomposing an affine camera projection matrix with Equation (2.7)

Algorithm pseudo-code is provided in Algorithm 9.

Algorithm 9 Decomposes a 2×4 affine projection matrix \mathbf{M} according to Equation (2.7)

Require: $\mathbf{M} \in \mathbb{R}^{2 \times 4}$

- 1: **function** affine_decompose(\mathbf{M})
 - 2: $\mathbf{U}\Sigma\mathbf{V}^\top \leftarrow \text{svd}([\mathbf{M}]_{2 \times 3}), \det(\mathbf{V}) = 1$
 - 3: $\begin{bmatrix} k & \alpha \\ 0 & \beta \end{bmatrix} \mathbf{Q} \leftarrow \text{lq}(\mathbf{U}\Sigma)$ ▷ LQ decomposition
 - 4: $\mathbf{R}^\circ \leftarrow \begin{bmatrix} \mathbf{Q} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \mathbf{V}^\top$
 - 5: $[\mathbf{t}^\circ]_{2 \times 1} \leftarrow \mathbf{M} [\mathbf{0}_{3 \times 1}^\top 1]^\top$
 - 6: **return** $k, \alpha, \beta, \mathbf{R}^\circ, [\mathbf{t}^\circ]_{2 \times 1}$
-

A.3 Chapter 3 appendices

A.3.1 2D Affine scene reconstruction from point correspondences with missing data

A.3.2 Proof of Theorem 1

A.3.2.1 Definitions and Theorem 1 in a Compact Form

We define $(\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ to be a 2D affine reconstruction computed from noise-free measurements, with the rank of $\tilde{\mathbf{S}}^A$ being equal to the rank of the metric structure matrix \mathbf{S} (which is at most two). We

Algorithm 10 (2D Affine scene reconstruction from point correspondences with missing data)

Require: $\{\mathbf{q}_i^j\}$ ▷ point correspondences with view index $i \in \{1, \dots, M\}$ point index $j \in \{1, \dots, N\}$

- 1: **function** affineReconstruct2D($\{\mathbf{q}_i^j\}$)
- 2: Construct a directed graph \mathcal{G} of M nodes with weighted edges $E(j, k) \in \mathbb{R}^{+M \times M}$. $E(j, k)$ is the conditioning number of the linear system for solving the Least Squares 2D affine transform from view j to k using points measured in both views.
- 3: Compute the connected components of \mathcal{G} and remove all views not connected to the largest component.
- 4: Assign the root view i^* to be the one with the shortest sum of paths from all other views.
- 5: Compute 2D affine transforms \mathbf{F}_i from i to i^* by chaining affine transforms along the shortest path to i^* .
- 6: Transfer all measured points to the root view using \mathbf{F}_i . For each point $j \in \{1, \dots, M\}$ compute its median $\mathbf{s}_j \in \mathbb{R}^{2 \times 1}$ in the root view.
- 7: Initialize the affine structure \mathbf{S}^A with \mathbf{s}_j in its j^{th} column.
- 8: Compute Least Squares 2D affine transform \mathbf{F}'_i mapping \mathbf{S}^A to measured points in i^{th} view.
- 9: Jointly refine \mathbf{F}'_i and \mathbf{S}^A to minimize the affine reconstruction reprojection error using Levenberg-Marquardt.
- 10: **return** $\mathbf{M}^A = \text{stk}([\mathbf{F}'_1]_{2 \times 2}, \dots, [\mathbf{F}'_M]_{2 \times 2}), \mathbf{S}^A$

refer to solving PSfM-O using $(\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ as the *noise-free PSfM-O problem*:

The noise-free PSfM-O problem:

find $\mathbf{w} \in \mathbb{R}^3$ s.t.

$$\begin{cases} \tilde{\mathbf{M}}_i^A f(\mathbf{w}) \tilde{\mathbf{M}}_i^{A\top} \in \mathcal{G}_{2 \times 2}, \quad \forall i \in \{1, 2, \dots, M\} & (a) \\ f(\mathbf{w}) \succ \mathbf{0} & (b) \end{cases} \quad (\text{A.4})$$

The number of solutions to problem (A.4) gives the number of metric structure solutions.

Definition 1. An input $(\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ is a degenerate input if and only if problem (A.4) has an infinite number of solutions.

Definition 2. An input $(\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ is a non-degenerate input if and only if problem (A.4) has a finite number of solutions.

We use the following terms defined in §3.3.2: trivial camera degeneracies, critical motion sequences, structural degeneracies, mixed degeneracies, artificial degeneracies and Non-artificially Degenerate Algorithms (NADAs). We also define two more geometric entities:

Definition 3. The camera Gramian matrix \mathbf{G}_i for view i :

$$\begin{aligned} \mathbf{G}_i \in \mathcal{G}_{2 \times 2} &\stackrel{\text{def}}{=} [\mathbf{R}_i]_{2 \times 2}^\top [\mathbf{R}_i]_{2 \times 2} \\ &= \mathbf{X}^\top \tilde{\mathbf{M}}_i^{A\top} \tilde{\mathbf{M}}_i^A \mathbf{X} \\ &= \mathbf{I}_2 - [\mathbf{a}_i]_{2 \times 1} [\mathbf{a}_i]_{2 \times 1}^\top \end{aligned} \quad (\text{A.5})$$

The second line is because $[\mathbf{R}_i]_{2 \times 2} = \tilde{\mathbf{M}}_i^A \mathbf{X}$ and the third line comes from the fact that \mathbf{a}_i is the third row of \mathbf{R}_i and \mathbf{R}_i is unitary. The camera Gramian matrix is important as a tool for geometrically interpreting the problem's degeneracies.

Definition 4. The scalar D with $1 \leq D \leq M$ is the number of unique camera Gramian matrices in the scene.

From Equation (A.5) we have:

$$\mathbf{G}_i = \mathbf{G}_j \Leftrightarrow [\mathbf{a}_i]_{2 \times 1} = \pm [\mathbf{a}_j]_{2 \times 1} \Leftrightarrow \mathbf{a}_i = \begin{bmatrix} \pm \mathbf{I}_2 & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & \pm 1 \end{bmatrix} \mathbf{a}_j \quad (\text{A.6})$$

This says that two camera Gramian matrices are equivalent if and only if the projection directions of the two cameras are the same up to a sign change and a reflection about the structure plane (recall the structure plane is defined in world coordinates on the plane $z = 0$). This means D is also the number of projection directions in the scene that are unique up to reflections about the structure plane and changes of sign.

The trivial camera degeneracy stated in Theorem 1 is equivalent to the condition $D < 3$. The critical motion sequence stated in Theorem 1 is when all projection directions lie on a plane which is orthogonal to the structures plane (Figure 3.1, right). This is equivalent to the condition:

$$\exists \mathbf{a} \neq \mathbf{0}_{2 \times 1} \text{ s.t. } \forall i \in \{1, 2, \dots, M\}, [\mathbf{a}_i]_{2 \times 1} \propto \mathbf{a} \quad (\text{A.7})$$

Theorem 1 is then stated compactly as follows:

$$\left(\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A \right) \text{ is degenerate} \Leftrightarrow \text{rank}(\mathbf{S}) < 2 \text{ or } D < 3 \text{ or Eq. (A.7) holds} \quad (\text{A.8})$$

This states that there is no mixed degeneracy in PSfM-O and a structural degeneracy only occurs when $\text{rank}(\mathbf{S}) < 2$ (*i.e.* the structures points being co-linear).

The reverse implication of Equation (A.8) is easy to prove and given at the end of this section. The forward implication trivially holds when $\text{rank}(\tilde{\mathbf{S}}^A) < 2$ because $\text{rank}(\tilde{\mathbf{S}}^A) = \text{rank}(\mathbf{S})$ (from the definition of $\tilde{\mathbf{S}}^A$ at the beginning of this section). The forward implication when $\text{rank}(\tilde{\mathbf{S}}^A) = 2$ is however not easy to prove. We do this first for the minimal case of $M = 3$ views. The generalisation to $M > 3$ views then follows quite easily. To ease readability we use $\bar{\mathbf{S}}^A$ to denote a rank-two affine structure matrix.

A.3.2.2 Proof of Forward Implication of Theorem 1 with $M = 3$ views

When $M = 3$ we can solve problem (A.4) with Algorithm 1. We then prove the forward implication with a hypothetical syllogism:

$$\begin{aligned} (\tilde{\mathbf{M}}^A, \bar{\mathbf{S}}^A) \text{ is degenerate} &\Rightarrow \text{Algorithm 1 fails} \\ \text{Algorithm 1 fails} &\Rightarrow D < 3 \text{ or Eq. (A.7) holds} \\ \therefore (\tilde{\mathbf{M}}^A, \bar{\mathbf{S}}^A) \text{ is degenerate} &\Rightarrow D < 3 \text{ or Eq. (A.7) holds} \end{aligned} \quad (\text{A.9})$$

The first line is true by definition (because all algorithms fail with a degenerate scene). Our task is to prove the second line. We do this with the following lemmas, recalling that in Exact-PSfM-O \mathbf{A}_E is the linear constraint matrix (see Equation (3.20)) and a , b and c are the three quadratic coefficients (see Equation (3.21)).

Lemma 1. *Exact-PSfM-O Upgrade* $\left(\tilde{\mathbf{M}}^A \right)$ fails $\Leftrightarrow \text{rank}(\mathbf{A}_E) \leq 2$ or $a = b = c = 0$

Lemma 2. $\text{rank}(\mathbf{A}_E) \leq 2 \Rightarrow (a = b = c = 0)$ does not hold

Lemma 3. $\text{rank}(\mathbf{A}_E) \leq 2$ and $\text{rank}(\mathbf{S}) = 2 \Rightarrow D < 3$ or Eq. (A.7) holds

Lemmas 1 and 2 tell us that the only time Exact-PSfM-O fails is when the matrix \mathbf{A}_E is rank-deficient. Lemma 3 tells us that when \mathbf{A}_E is rank-deficient and $\text{rank}(\mathbf{S}) = 2$ the right side of Equation (A.8) must hold, which completes the proof.

Proof of Lemma 1 Exact-PSfM-O fails if and only if we cannot compute the upgrade matrix using Algorithm 1. This can happen for one of two reasons. The first is at Algorithm 1, line 6 and happens when \mathbf{A}_E is rank-deficient (*i.e.* $\text{rank}(\mathbf{A}_E) \leq 2$). This means we cannot compute \mathbf{z} uniquely up to scale (*i.e.* we cannot compute a 1D affine subspace for the upgrade matrix). If however \mathbf{A}_E is full-rank then we can always compute $\text{stk}(\mathbf{w}', s')$ with line 7. The second place where Algorithm 1 may fail is at line 11 and happens when all coefficients in the quadratic equation are zero: $a = b = c = 0$. This means we cannot resolve α and so we cannot resolve where in the affine subspace the upgrade matrix exists. \square

Proof of Lemma 2 We prove this by splitting the space of full-rank \mathbf{A}_E matrices into two sets and showing that in either set $a = b = c = 0$ is contradicted. Set 1 is when $\det(\mathbf{E}_i) = 0 \forall i \in \{1, 2, 3\}$. Set 2 is the complement (when $\exists i \in \{1, 2, 3\}, \det(\mathbf{E}_i) \neq 0$).

Set 1: By definition in Set 1 the fourth column of \mathbf{A}_E is all-zeros. Therefore $\text{rank}(\mathbf{A}_E) = 3 \Rightarrow \mathbf{z} = \pm[0001]^\top$. However from Equation (3.21) this implies $b = \pm 1$ which contradicts $b = 0$.

Set 2: Without loss of generality let $\det(\tilde{\mathbf{M}}_1^A) \neq 0$, which implies $\tilde{\mathbf{M}}_1^A$ is full-rank. Because the affine reconstruction is up to an arbitrary full-rank 2D affine transform, the problem does not change by redefining the factors with $\tilde{\mathbf{M}}_i^A \leftarrow \tilde{\mathbf{M}}_i^A \left(\tilde{\mathbf{M}}_1^A \right)^{-1}$ and $\mathbf{S}^A \leftarrow \tilde{\mathbf{M}}_1^A \mathbf{S}^A$. Thus without loss of generality we can assume $\tilde{\mathbf{M}}_1^A = \mathbf{I}_2$. We then have

$$\begin{aligned} (c = 0) &\Rightarrow (w'_1 w'_3 - w'^2_2 = s') & (a) \\ ([101 \ -1] \text{stk}(\mathbf{w}', s') = 1) &\Rightarrow w'_1 + w'_3 = s' & (b) \end{aligned} \tag{A.10}$$

Equation (A.10-a) comes from the definition of c in Equation (3.21). Equation (A.10-b) comes from the first linear constraint in Equation (3.20). When $c = 0$, this means the quadratic constraint in Equation (3.20) is satisfied by $s \leftarrow s'$ and $\mathbf{w} \leftarrow \mathbf{w}'$. By definition, (\mathbf{w}', s') also satisfies the linear constraints in Equation (3.20), which means $s \leftarrow s'$ and $\mathbf{w} \leftarrow \mathbf{w}'$ is a solution to Equation (3.20), and is therefore a solution to Equation (3.18). Now because $\tilde{\mathbf{M}}_1^A = \mathbf{I}_2$, we have $\det(f(\mathbf{w}') - \mathbf{I}_2) = 0$, which implies either $\lambda_1(w(\mathbf{w}')) = 1$ or $\lambda_2(w(\mathbf{w}')) = 1$. However this is contradicted by Equation (A.10-b). To see this, we can eliminate s' from the right sides of Equations (A.10-a,b) to give:

$$\begin{aligned} w'_1 + w'_3 &= w'_1 w'_3 - w'^2_2 \\ \Leftrightarrow \text{trace}(w(\mathbf{w}')) &= \det(w(\mathbf{w}')) \\ \Leftrightarrow \lambda_1(w(\mathbf{w}')) + \lambda_2(w(\mathbf{w}')) &= \lambda_1(w(\mathbf{w}'))\lambda_2(w(\mathbf{w}')) \end{aligned} \tag{A.11}$$

If $\lambda_1(w(\mathbf{w}')) = 1$ this means $1 + \lambda_2(w(\mathbf{w}')) = \lambda_2(w(\mathbf{w}'))$ which is false for all values of $\lambda_2(w(\mathbf{w}'))$. If $\lambda_2(w(\mathbf{w}')) = 1$ this means $1 + \lambda_1(w(\mathbf{w}')) = \lambda_1(w(\mathbf{w}'))$ which is false for all values of $\lambda_1(w(\mathbf{w}'))$. Therefore we have a contradiction. \square

Proof of Lemma 3 From the definition of \mathbf{A}_E in Equation (3.20) we have:

$$\begin{aligned} \text{rank}(\mathbf{A}_E) \leq 2 &\Rightarrow \exists \alpha, \beta \in \mathbb{R}, \text{ s.t. } \forall \{i, j, k\} \in \text{perm}(\{1, 2, 3\}) \\ &\left\{ \begin{array}{l} \tilde{\mathbf{M}}_i^{A\top} \tilde{\mathbf{M}}_i^A = \alpha \tilde{\mathbf{M}}_j^{A\top} \tilde{\mathbf{M}}_j^A + \beta \tilde{\mathbf{M}}_k^{A\top} \tilde{\mathbf{M}}_k^A \quad (a) \\ \det(\tilde{\mathbf{M}}_i^A) = \alpha \det(\tilde{\mathbf{M}}_j^A) + \beta \det(\tilde{\mathbf{M}}_k^A) \quad (b) \end{array} \right. \end{aligned} \quad (\text{A.12})$$

Equation (A.12-a) comes from the first three columns of \mathbf{A}_E , and Equation (A.12-b) comes from the fourth column. These equations impose linear constraints on the camera Gramian matrices:

$$\begin{aligned} \text{Eq. (A.12-a)} &\Rightarrow (\mathbf{G}_i = \alpha \mathbf{G}_j + \beta \mathbf{G}_k) \quad (a) \\ \text{Eq. (A.12-b)} &\Rightarrow (\det(\mathbf{G}_i) = \alpha \det(\mathbf{G}_j) + \beta \det(\mathbf{G}_k)) \quad (b) \end{aligned} \quad (\text{A.13})$$

This comes by pre and post-multiplying Equation (A.12-a,b) by \mathbf{X}^\top and \mathbf{X} respectively and substituting in \mathbf{G}_i using Equation (A.5). Because $\mathbf{G}_i \in \mathcal{G}_{2 \times 2}$, $\text{trace}(\mathbf{G}_i) = \lambda_1(\mathbf{G}_i) + \lambda_2(\mathbf{G}_i) = 1 + \det(\mathbf{G}_i)$, so taking the trace of the right hand of Equation (A.13-a) gives:

$$1 + \det(\mathbf{G}_i) = \alpha(1 + \det(\mathbf{G}_j)) + \beta(1 + \det(\mathbf{G}_k)) \quad (\text{A.14})$$

Subtracting Equation (A.13-b) from both sides of Equation (A.14) gives $\beta = 1 - \alpha$. We now take the right side of Equation (A.13-a) and substitute β by $(1 - \alpha)$:

$$\mathbf{G}_i = \alpha \mathbf{G}_j + (1 - \alpha) \mathbf{G}_k \Leftrightarrow [\mathbf{a}_i]_{2 \times 1} [\mathbf{a}_i]_{2 \times 1}^\top = \alpha [\mathbf{a}_j]_{2 \times 1} [\mathbf{a}_j]_{2 \times 1}^\top + (1 - \alpha) [\mathbf{a}_k]_{2 \times 1} [\mathbf{a}_k]_{2 \times 1}^\top \quad (\text{A.15})$$

The right part comes by substituting the camera Gramian matrices for the camera projection directions using Equation (A.5). For what projection directions does Equation (A.15) hold? Clearly the determinant of both sides of the second equality in Equation (A.15) must be zero. Taking the right hand side, after simplification we have:

$$\alpha(1 - \alpha) \det \begin{bmatrix} [\mathbf{a}_j]_{2 \times 1} & [\mathbf{a}_k]_{2 \times 1} \end{bmatrix} = 0 \quad (\text{A.16})$$

This holds if either $\alpha = 0$, $\alpha = 1$ or $[\mathbf{a}_j]_{2 \times 1} \propto [\mathbf{a}_k]_{2 \times 1}$. If $\alpha = 0$ then $\mathbf{G}_i = \mathbf{G}_k$, which implies $D < 3$ (*i.e.* a trival camera degeneracy). Similarly we have a trivial camera degeneracy when $\alpha = 1$. In the third case we have $[\mathbf{a}_j]_{2 \times 1} \propto [\mathbf{a}_k]_{2 \times 1}$. From Equation (A.15) we therefore have $[\mathbf{a}_i]_{2 \times 1} [\mathbf{a}_i]_{2 \times 1}^\top \propto [\mathbf{a}_j]_{2 \times 1} [\mathbf{a}_j]_{2 \times 1}^\top \propto [\mathbf{a}_k]_{2 \times 1} [\mathbf{a}_k]_{2 \times 1}^\top$, which implies Equation (A.7). \square

A.3.2.3 Proof of Theorem 1 (Forward Implication for Arbitrary M)

If the degeneracy is caused by the scene's structure (*i.e.* $\text{rank}(\mathbf{S}) < 2$) then the scene is degenerate no matter the value of M . Consider instead $\text{rank}(\mathbf{S}) = 2$. The forward implication of Equation (A.8) is proved by showing that $D \geq 3$ implies Equation (A.7) holds. When the scene is degenerate there cannot exist a subset $\mathcal{I} \in \{1, 2, \dots, M\}^3$ of three views that is non-degenerate. This is because if there were such a subset then we could compute the upgrade matrix using only the views in \mathcal{I} and the problem would be solved. When $D \geq 3$ we have at least three camera projection directions that are distinct up to sign changes and reflections about the structure plane. For all subsets of three views these camera projection directions must lie on a plane that is orthogonal to the structure plane.

It therefore follows that *all* of the camera projections must lie on this plane. This is equivalent to Equation (A.7) holding. \square

A.3.2.4 Proof of Equation (A.8) (reverse implication)

Proof that $D < 3 \Rightarrow (\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ is degenerate By definition $D < 3$ means there are fewer than three unique camera Gramian matrices. Because $\mathbf{G}_i = [\mathbf{R}_i]_{2 \times 2}^\top [\mathbf{R}_i]_{2 \times 2}$ and $\mathbf{G}_j = [\mathbf{R}_j]_{2 \times 2}^\top [\mathbf{R}_j]_{2 \times 2}$ by definition, $(\mathbf{G}_i = \mathbf{G}_j) \Leftrightarrow ([\mathbf{R}_i]_{2 \times 2} = [\mathbf{R}_j]_{2 \times 2} \mathbf{U}) \Leftrightarrow (\mathbf{M}_i^A \mathbf{X} = \mathbf{M}_j^A \mathbf{X} \mathbf{U})$ for some 2D unitary matrix \mathbf{U} . This means $\mathbf{M}_i^A \mathbf{X} \in \mathcal{SS}_{2 \times 2} \Leftrightarrow \mathbf{M}_j^A \mathbf{X} \in \mathcal{SS}_{2 \times 2}$ (*i.e.* the upgrade constraint is satisfied by view i if and only if it is satisfied by view j). Therefore given view i , view j provides no extra constraints on the upgrade matrix. Therefore when $D \leq 3$ there are fewer than three constraints on \mathbf{X} (which has 3DoFs). \square

Proof that Equation (A.7) $\Rightarrow (\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ is degenerate Without loss of generality we rotate world coordinates about the z -axis so that $\mathbf{a} = [1, 0]^\top$ (*i.e.* the azimuths of all cameras is now zero). The camera rotations are now of the form

$$\mathbf{R}_i = \begin{bmatrix} R_{2D}(\psi_i) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & 0 & \sin(\theta_i) \\ 0 & 1 & 0 \\ -\sin(\theta_i) & 0 & \cos(\theta_i) \end{bmatrix} \quad (\text{A.17})$$

$$[\mathbf{R}_i]_{2 \times 2} = R_{2D}(\psi_i) \begin{bmatrix} \cos(\theta_i) & 0 \\ 0 & 1 \end{bmatrix}$$

The 2D rotation matrix $R_{2D}(\psi_i) \in \mathcal{SS}_{2 \times 2}$ denotes a rotation of the camera's image about the camera projection direction by an angle ψ_i . The angle θ_i is the inclination angle of the i^{th} camera's projection direction (see Figure 3.1). Given any factorisation $\hat{\mathbf{Q}} = \text{stk}([\mathbf{R}_1]_{2 \times 2}, \dots, [\mathbf{R}_M]_{2 \times 2})^\top \mathbf{S}_{2 \times N}$, consider

the alternative factorisation $[\mathbf{R}'_i]_{2 \times 2} \leftarrow [\mathbf{R}_i]_{2 \times 2} \begin{bmatrix} d & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{S}'_{2 \times N} \leftarrow \begin{bmatrix} \frac{1}{d} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{S}_{2 \times N}$ for some

scalar d . For all $0 \leq d < 1$ we have $[\mathbf{R}'_i]_{2 \times 2} \in \mathcal{SS}_{2 \times 2}$, so the alternative factorisation is metric. If there exists a non-zero inclination angle $\theta_i \neq 0$ then there are an infinite number of alternative metric factorizations, so the scene is degenerate. By contrast, if there does not exist a non-zero inclination angle then all cameras have the same projection direction (which is orthogonal to the structure plane), which implies $D = 1$, and from the first paragraph the scene is also degenerate. \square

Proof that $\text{rank}(\tilde{\mathbf{S}}) < 2 \Rightarrow (\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ is degenerate When $\text{rank}(\tilde{\mathbf{S}}) < 2$ points in world coordinates are co-linear. This means the camera rotations cannot be fixed because we can rotate each camera about an axis of rotation that is co-linear with the points in world coordinates and the image measurements do not change. \square

A.3.3 Proof of Theorem 2

Without loss of generality let \mathbf{w}_1 and \mathbf{w}_2 be the two upgrade solutions using views 1,2 and 3. From Equation (3.19) we have $\mathbf{B} \text{stk}(\mathbf{w}, s) = \mathbf{1}_{M \times 1}$, where \mathbf{B} is an $M \times 4$ matrix with each row being $[a_i \ b_i \ c_i \ d_i]$, and $s = \det(f(\mathbf{w}))$. We use the following lemma:

Lemma 4. *Given four or more views we can disambiguate \mathbf{w}_1 and \mathbf{w}_2 if and only if \mathbf{B} is full-rank.*

Proof. We first prove the reverse implication. When \mathbf{B} is full-rank we can relax the quadratic constraint $s = \det(f(\mathbf{w}))$ to $s \in \mathbb{R}$, and we can then solve \mathbf{w} and s uniquely by inverting the linear system: $\text{stk}(\mathbf{w}, s) = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{1}_{M \times 1}$. We prove the reverse implication by showing that if we cannot disambiguate \mathbf{w}_1 and \mathbf{w}_2 then \mathbf{B} is rank-deficient. Let $s_1 = \det(f(\mathbf{w}_1))$ and $s_2 = \det(f(\mathbf{w}_2))$. If we cannot disambiguate \mathbf{w}_1 and \mathbf{w}_2 then $\mathbf{B} \text{stk}(\mathbf{w}_1, s_1) = \mathbf{1}_{M \times 1}$ and $\mathbf{B} \text{stk}(\mathbf{w}_2, s_2) = \mathbf{1}_{M \times 1}$. Because \mathbf{w}_1 and \mathbf{w}_2 are distinct, this implies \mathbf{B} has a nullspace which implies \mathbf{B} is rank-deficient. \square

We now use the following lemma:

Lemma 5. *Given a fourth view, \mathbf{B} is full-rank if and only if Equation (3.37) holds.*

Proof. Because we have computed upgrade solutions using the first three views, the first three rows of \mathbf{B} must be linearly independent (otherwise the PSFM-O problem using the first three views would be degenerate, see Appendix A.3.2.2). Therefore \mathbf{B} is rank deficient if and only if its fourth column is a linear combination of its first three columns. From the definition of \mathbf{B} this means

$$\begin{aligned} \text{rank}(\mathbf{B}) < 4 &\Leftrightarrow \exists \alpha, \beta, \gamma \in \mathbb{R} \text{ s.t.} \\ \left\{ \begin{array}{l} \mathbf{E}_4 = \alpha \mathbf{E}_1 + \beta \mathbf{E}_2 + \gamma \mathbf{E}_3 \\ \det(\mathbf{E}_4) = \alpha \det(\mathbf{E}_1) + \beta \det(\mathbf{E}_2) + \gamma \det(\mathbf{E}_3) \end{array} \right. & \quad (a) \quad (A.18) \end{aligned}$$

Pre and post-multiplying Equation (A.18-a,b) by \mathbf{X} and \mathbf{X}^\top , and using $\mathbf{G}_i = \mathbf{X} \mathbf{E}_i \mathbf{X}^\top$ gives

$$\begin{aligned} \text{rank}(\mathbf{B}) < 4 &\Leftrightarrow \exists \alpha, \beta, \gamma \in \mathbb{R} \text{ s.t.} \\ \left\{ \begin{array}{l} \mathbf{G}_4 = \alpha \mathbf{G}_1 + \beta \mathbf{G}_2 + \gamma \mathbf{G}_3 \\ \det(\mathbf{G}_4) = \alpha \det(\mathbf{G}_1) + \beta \det(\mathbf{G}_2) + \gamma \det(\mathbf{G}_3) \end{array} \right. & \quad (a) \quad (A.19) \end{aligned}$$

We then take the trace of both sides of Equation (A.19-a), substitute $\text{trace}(\mathbf{G}_i) \leftarrow 1 + \det(\mathbf{G}_i)$, and then subtract Equation (A.19-b), which gives $\alpha + \beta + \gamma = 1$. We then substitute $\gamma \leftarrow (1 - \alpha - \beta)$ into Equation (A.19-a) and substitute $\mathbf{G}_i \leftarrow \mathbf{I}_2 - [\mathbf{a}_i]_{2 \times 1} [\mathbf{a}_i]_{2 \times 1}^\top$, which gives

$$\begin{aligned} \text{rank}(\mathbf{B}) < 4 &\Leftrightarrow \nexists \alpha, \beta \in \mathbb{R} \text{ s.t.} \\ [\mathbf{a}_4]_{2 \times 1} [\mathbf{a}_4]_{2 \times 1}^\top &= \alpha [\mathbf{a}_1]_{2 \times 1} [\mathbf{a}_1]_{2 \times 1}^\top + \beta [\mathbf{a}_2]_{2 \times 1} [\mathbf{a}_2]_{2 \times 1}^\top + (1 - \alpha - \beta) [\mathbf{a}_3]_{2 \times 1} [\mathbf{a}_3]_{2 \times 1}^\top \end{aligned} \quad (A.20)$$

The proof is completed by negating the implication in Equation (A.20). \square

The proof of Theorem 2 is completed by generalising the result to $M > 4$ views. Lemma 5 tells us that if we have a fourth view for which Equation (3.37) holds then we can determine the correct structure solution. However, if Equation (3.37) does not hold then the rank of \mathbf{B} stays at three. From Lemma 4 this means the fourth view provides no extra constraints on the solution. We can therefore only determine the correct structure solution when we have at least one additional view for which Equation (3.37) holds. \square

A.3.4 Proof of Theorems 3 to 8

A.3.4.1 Theorems 3 and 4

Theorems 3 has two parts. For the first part, the forward implication holds trivially because if a Type 1 problem is degenerate then the equivalent problem with full measurements is degenerate, because by definition Type 1 problems are those where we can complete the rank-2 measurement matrix from the incomplete measurements. The reverse implication holds because when a system with complete measurements is degenerate then if we remove any of the measurements the problem is still degenerate. For the second part, because we cannot compute the scene's 2D affine reconstruction with a Type 2 problem (by definition) then we cannot compute the scene's 2D metric reconstruction (because all metric reconstructions are affine reconstructions).

Theorem 4 has three parts. The first part holds trivially because if there are three or more correspondences that are non-co-linear on the structure plane, then the structure-plane-to-image 2D affine transform is fully-determined. Thus any additional point correspondences are redundant, so the disambiguation problem is equivalent to disambiguating with complete measurements. The second part holds because if we have two distinct metric structures the Euclidean distance between two points will in general be different. Let $d_1 > 0, d_2 > 0$ be the Euclidean distance between the two points for structure solutions 1 and 2. Without loss of generality we assume $d_1 \leq d_2$. Due to image foreshortening with an orthographic camera the true Euclidean distance $d \in \{d_1, d_2\}$ must be equal to or exceed the Euclidean distance d_I between their positions in the image (*i.e.* $d_I \leq d$). We can disambiguate structure if and only if there exists an additional image with $d_I > d_1$. When this happens we know $d \neq d_1$ (by contradiction), so structure solution 2 is correct. By contrast suppose structure solution 1 is correct, so $d = d_1$. In this case we cannot disambiguate structure because the inequality $d_I \leq d$ is satisfied by both $d = d_1$ (because d_1 is the true distance) and $d = d_2$ (because $d_2 \geq d_1$). \square

A.3.4.2 Theorem 5

Theorem 5 requires proving Algorithm 1 fails $\Leftrightarrow (\tilde{\mathbf{M}}^A, \tilde{\mathbf{S}}^A)$ is degenerate. The forward implication has been proved by the second line of Equation (A.9) and the reverse implication has been proved by Equation (A.8). \square

A.3.4.3 Theorem 6

Let $r_A \in \mathbb{R}$ denote the reprojection error of the scene's optimal affine reconstruction and let $r_M \in \mathbb{R}$ denote the reprojection error of a metric reconstruction. There exists no metric reconstruction with $r_M < r_A$ because if we impose metric constraints on the cameras we reduce the solution space. Let r'_M denote the reprojection error of a metric reconstruction found by Exact-PSfM-O by upgrading the optimal affine reconstruction. Because solutions from Exact-PSfM-O exactly transforms the affine reconstruction to a metric reconstruction we have $r'_M = r_A$. Therefore it is not possible to find a better metric reconstruction of the scene, otherwise it would have a lower reprojection error than r_A . This means all solutions from Exact-PSfM-O must be optimal metric reconstructions.

The last part of theorem 6 follows because Exact-PSfM-O is NADA. Concretely, when Exact-PSfM-O has no solution, this means we cannot turn the optimal affine reconstruction into a metric reconstruction by transforming it with an upgrade matrix. Therefore for any upgrade matrix $\tilde{\mathbf{X}}$ the reconstructed camera factor $\tilde{\mathbf{M}}^A \tilde{\mathbf{X}}$ cannot be a metric camera factor. The individual camera

factors $\tilde{\mathbf{M}}_i^A \tilde{\mathbf{X}}$ must therefore be corrected *a posteriori* to make them members of $\mathcal{SS}_{2 \times 2}$. However the corrected solution will not be optimal because no matter how the correction is performed the upgraded structure factor $\hat{\mathbf{S}} = \tilde{\mathbf{X}}^{-1} \mathbf{S}^A$ will not be optimal in terms of reprojection error. \square

A.3.4.4 Theorem 7

Using the definition of the general affine camera in Equation (2.7), the generalization of the PSfM-O upgrade constraint to PSfM-PP is as follows:

$$\begin{aligned}
\mathbf{M}_i^A \mathbf{X} &= \alpha_i \mathbf{A}_i \mathbf{R}_i && \Leftrightarrow \\
\lambda_1 \left(\alpha_i^{-2} \mathbf{A}_i^{-1} \mathbf{M}_i^A f(\mathbf{w}) \mathbf{M}_i^{A\top} \mathbf{A}_i^{-\top} \right) &= 1 && \Leftrightarrow \\
\det \left(\mathbf{M}_i^A w(\mathbf{w}) \mathbf{M}_i^{A\top} - \alpha_i^2 \mathbf{A}_i \mathbf{A}_i^\top \right) &= 0 && (a) \\
\det \left(\mathbf{M}_i^A w(\mathbf{w}) \mathbf{M}_i^{A\top} \right) &\leq \alpha_i^2 \det^2(\mathbf{A}_i) && (b)
\end{aligned} \tag{A.21}$$

The first equivalence comes from $\alpha_i^{-1} \mathbf{A}_i^{-1} \mathbf{M}_i^A \mathbf{X} \in \mathcal{SS}_{2 \times 2}$ and the second equivalence comes from rewriting this constraint similarly as Equation (3.17). Therefore in PSfM-PP each view provides one equality constraint on \mathbf{w} (which recall has three DoFs).

For Case 1 we divide the views into two disjoint sets: \mathcal{I}' and $\mathcal{J} \stackrel{\text{def}}{=} \{1, 2, \dots, M\} \setminus \mathcal{I}'$ with $\text{size}(\mathcal{I}') \geq 3$. The views in \mathcal{J} provide no constraints on metric structure. To see this, for each view $i \in \mathcal{J}$ we have to solve camera resection by decomposing $\mathbf{A}_i^{-1} \mathbf{M}_i^A \mathbf{X}$ into $\alpha_i \mathbf{R}_i$. This provides no constraints on \mathbf{X} (and hence no constraints on \mathbf{W}) because for all $(\mathbf{A}_i, \mathbf{X})$ we can compute the decomposition by $\alpha_i = \sigma_1$ and $\mathbf{R}_i = \frac{1}{\sigma_1} \mathbf{A}_i^{-1} \mathbf{M}_i^A \mathbf{X}$ with $\sigma_1 \stackrel{\text{def}}{=} s_1(\mathbf{A}_i^{-1} \mathbf{M}_i^A \mathbf{X})$. Therefore only the views in \mathcal{I}' are relevant for constraining structure. Because $\mathbf{A}_{i \in \mathcal{I}'}$ is known and $\alpha_{i \in \mathcal{I}'}$ is constant we can effectively convert all views in \mathcal{I}' to orthographic views by transforming the points with \mathbf{A}_i^{-1} . This has the effect of undoing the ‘intrinsic’ component of the camera matrices (*i.e.* \mathbf{A}_i). Now, because α_i is assumed to be constant for all views in \mathcal{I}' this is exactly the same as using orthographic cameras with a common magnification factor $\alpha = \alpha_i$. Therefore we have converted the problem to PSfM-O where we only consider the views in \mathcal{I}' . It therefore follows that structure is solvable if and only if the equivalent PSfM-O problem is solvable, and the geometric conditions for ensuring this are given by Theorem 1.

For Case 2 we divide the views into two disjoint sets: \mathcal{I}'' and $\mathcal{J}' \stackrel{\text{def}}{=} \{1, 2, \dots, M\} \setminus \mathcal{I}''$. Similarly to Case 1 the views in \mathcal{J}' provide no constraints on structure, so we need only consider the views in \mathcal{I}'' . From Equation (A.21-a). For all $i \in \mathcal{I}''$ we can write the combined term $\alpha_i^2 \mathbf{A}_i \mathbf{A}_i^\top$ as an

unknown positive definite matrix \mathbf{V} parameterized with $\mathbf{V} = v(\mathbf{v} \in \mathbb{R}^3) = \begin{bmatrix} v_1 & v_2 \\ v_2 & v_3 \end{bmatrix}$. Equation

(A.21) provides one homogeneous quadratic constraint on six unknowns (*i.e.* \mathbf{w} and \mathbf{v}). This means that to obtain a metric reconstruction we require the size of \mathcal{I}'' to be at least 5.

For Case 3, let $\{p_1, p_2, \dots, p_P\}$ denote the set of view pairs with $p_l \in \{1, 2, \dots, P\} \in \{1, 2, \dots, M\}^2$. Let view i be a view that does not belong to a view pair (*i.e.* there is no other view that has the same magnification factor as α_i). From the same reasoning as Case 1, view i provides no constraints on structure. Therefore to determine structure we need to only deal with the views in $\{p_1, p_2, \dots, p_P\}$. To fix the scene’s scale ambiguity we can arbitrarily set the magnification factor of the first pair to 1, which means the number of unknowns is $P + 2$ (including three unknowns for \mathbf{w}). The number of equality constraints from Equation (A.21-a) is $2P$, which means to have the necessary number of

equations we must have $P \geq 2$ (which means we require 4 or more views). \square

A.3.4.5 Theorem 8

This follows from §2.1.2.5. We first take the problem with para-perspective cameras. This requires calibrating the POPP \mathbf{x}'_i for each view i . From [Hor+97] we know that linearization error is approximately minimized by setting \mathbf{x}' as the structure's 3D centroid $\mathbf{c}_i \in \mathbb{R}^3$ in view i . We parameterize $\mathbf{x}'_i = d_i \text{stk}(\mathbf{v}_i, 1)$, where $d_i \in \mathbb{R}$ is the depth of the centroid and $\mathbf{v}_i \in \mathbb{R}^2$ is its direction. A first-order approximation of \mathbf{v}_i is the vector passing through the centroid of the structure points in image i [Hor+97]. What is unknown is d_i . We know that d_i is, to first-order, inversely proportional to the camera's magnification factor α_i . Therefore if d_i is approximately constant then so is α_i . Therefore by definition we are in Case 1.

Exactly the same argument follows for weak-perspective cameras. The only difference is that \mathbf{x}'_i is constrained to lie on the optical axis (by the definition of the weak perspective camera). The depth of \mathbf{x}'_i is calibrated by setting it to the average depth of the structure in view i [Hor+97]. \square

A.4 Chapter 4 appendices

A.4.1 Proofs of theorems

Proof of Theorem 9. We do this by demonstrating that Algorithm 6 solves Problem (4.12) $\forall \mathbf{v} \in \mathbb{R}^2$ and $\forall \mathbf{J} \in \mathbb{R}^{2 \times 2}$ such that $\mathbf{J} \neq \mathbf{0}_{2 \times 2}$. We then show that $\mathbf{J} = \mathbf{0}_{2 \times 2}$ is a general degeneracy where Problem (4.12) never has a solution. For the first part, we look at all potential singularities and we show they never occur when $\mathbf{J} \neq \mathbf{0}_{2 \times 2}$ (equivalent to $\text{rank}(\mathbf{J}) \geq 1$). There are three potential singularities: At line 2 (when $\|\mathbf{c}\| = 0$ or $\|\mathbf{x}'\| = 0$), at line 4 (when \mathbf{B} is singular) and line 5 (when $\gamma = 0$). Considering line 2, $\|\mathbf{x}'\| \geq 1$ because its third element is 1. We have $\|\mathbf{c}\|_2^2 = \mathbf{x}_1^2 \mathbf{x}_2^2 + \mathbf{x}_1^2 + (\mathbf{x}_1^2 - 1)^2 \neq 0$ because $\mathbf{x}_1^2 = 0 \Leftrightarrow (\mathbf{x}_1^2 - 1)^2 \neq 0$. Considering line 4, \mathbf{B} is singular if and only if $\det(\mathbf{B}) = 0$. From the definition of \mathbf{B} , $\det(\mathbf{B}) = \frac{1}{\|\mathbf{x}'\|_2} < 0$. Considering line 5, we have shown that $\forall \mathbf{v} \text{ rank}(\mathbf{B}) = 2$ (full-rank), therefore $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{J})$. σ_1^A (the largest singular value of \mathbf{A}) must always be greater than zero when $\text{rank}(\mathbf{J}) \geq 1$. We now consider when $\mathbf{J} = \mathbf{0}_{2 \times 2}$. There exists no solution to Problem (4.12) because we have just shown that $\mathbf{J} = \mathbf{0}_{2 \times 2} \Rightarrow \text{rank}(\mathbf{J}) = 0 \Rightarrow \sigma_1^A = 0 \Rightarrow \gamma = 0$. \square

Proof of Theorem 10. Item 1 is a known result from homography estimation. If satisfied, we can compute \mathbf{H} uniquely. Item 2 says that the homography estimation method is guaranteed to find such a homography. Item 3 is required in order to solve pose using Algorithm 6. In the case when \mathbf{u}_0 is at the object points' centroid, an all-zero Jacobian at \mathbf{u}_0 would imply that all object points shrink to a single point in the image, which would then imply that the object is infinitely far to the camera. In this case, no algorithm could solve pose. When all the above 3 conditions are satisfied, we are guaranteed to solve pose without singular cases. We note that at line 7 there is potential for a singularity but it does not happen in practice because as stated in §4.2.4.4, \mathbf{W}_j is always full-rank in solvable cases. \square

Proof of Theorem 11. We first consider rotation. Inspecting Algorithm 2, $\tilde{\mathbf{R}}_1$ and $\tilde{\mathbf{R}}_2$ are related by

$$\tilde{\mathbf{R}}_1 = \begin{bmatrix} \mathbf{C} & a \\ & c \\ d e & f \end{bmatrix}, \tilde{\mathbf{R}}_2 = \begin{bmatrix} \mathbf{C} & -a \\ & -c \\ -d - e & f \end{bmatrix} \quad (\text{A.22})$$

for some $\mathbf{C} \in \mathcal{SS}_{2 \times 2}$ and $a, c, d, e, f \in \mathbf{R}$. Therefore, the rotation of an object point $\mathbf{u} \in \mathbb{R}^2$ by $\tilde{\mathbf{R}}_1$ and $\tilde{\mathbf{R}}_1$ is related by

$$\tilde{\mathbf{R}}_2 \text{stk}(\mathbf{u}, 0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tilde{\mathbf{R}}_1 \text{stk}(\mathbf{u}, 0) \quad (\text{A.23})$$

Therefore the difference between rotating by $\tilde{\mathbf{R}}_1$ or $\tilde{\mathbf{R}}_2$ is a reflection of the rotated point about the z axis. The final rotation solutions are produced at line 7 of Algorithm 2 and they are $\mathbf{R}_v \tilde{\mathbf{R}}_1$ and $\mathbf{R}_v \tilde{\mathbf{R}}_2$. By definition, \mathbf{R}_v rotates the z -axis to align it with \mathbf{x}' . The combined effect is that the rotated object points differ by a reflection about a plane whose normal is co-linear with \mathbf{x}' . This reflection will have no effect if and only if the object's normal is co-linear with \mathbf{x}' . We now consider the position \mathbf{x} of \mathbf{u}_0 in camera coordinates, given by $\mathbf{x} = \frac{1}{\gamma} \text{stk}(\mathbf{v}, 1)$. Because γ is solved uniquely by Algorithm 6, \mathbf{x} has a unique solution. □

Proof of Theorem 12. We know the pose solution geometry from Theorem 11. In quasi-affine conditions, plane-based pose estimation has a general 2-fold flip ambiguity. This corresponds to a reflection of the object about a plane whose normal is aligned with the optical ray passing through the object's center [Hor+97]. Therefore, from Theorem 11, by using a central differentiation point, the pose solution geometry from IPPE matches the 2-fold flip ambiguity in quasi-affine conditions. We are then left with answering if our algorithm can recover these pose solutions in such conditions, equivalent to when \mathbf{H}_{31} and \mathbf{H}_{32} tend to zero. When they are zero the object-to-image transform $w(\mathbf{u})$ is affine:

$$w(\mathbf{u}_0) = \mathbf{v} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \mathbf{u}_0 + \begin{bmatrix} \mathbf{H}_{13} \\ \mathbf{H}_{23} \end{bmatrix} \quad (\text{A.24})$$

Similarly, the Jacobian of w becomes constant: $J_w(\mathbf{u}_0) = \mathbf{J} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}$.

Following Theorem 9, we know that if $\mathbf{J} \neq \mathbf{0}_{2 \times 2}$ then we can recover the rotation solutions using Algorithm 6. If $\mathbf{J} = \mathbf{0}_{2 \times 2}$, then the object would vanish to a point in the image, in which case pose estimation is unsolvable. Consequently, the fact that the perspective effects diminish does not affect the ability of Algorithm 6 to recover the two rotation solutions. □

Proof of Theorem 13. We assume that point correspondences in the image have I.I.D. noise of variance σ^2 . We first consider the uncertainty in \mathbf{v} from noisy correspondences as a function of the differentiation point \mathbf{u}_0 using 1st-order uncertainty propagation. Our analysis uses the DLT method to estimate $\hat{\mathbf{H}}$ and we neglect 2nd-order effects of $\hat{\mathbf{H}}$.

Lemma 6. *Neglecting 2nd-order effects of \mathbf{H} , the \mathbf{u}_0 that minimizes the uncertainty of \mathbf{v} is the centroid of $\{\mathbf{u}_i\}$.*

Proof of Lemma 6. The DLT gives the following 1st-order approximation of $\hat{\mathbf{H}}$ from point correspondences:

$$\hat{\mathbf{H}} \approx \begin{bmatrix} \hat{\mathbf{A}}_{ML} & \hat{\mathbf{t}}_{ML} \\ \mathbf{0}^\top & 1 \end{bmatrix} \Rightarrow \quad (\text{A.25})$$

$$\mathbf{v} \approx \hat{\mathbf{A}}_{ML} \mathbf{u}_0 + \hat{\mathbf{t}}_{ML}$$

which is the least squares affine transform that maps $\{\mathbf{u}_i\}$ to $\{\hat{\mathbf{q}}_i\}$. The solutions to $\hat{\mathbf{t}}_{ML}$ and $\hat{\mathbf{A}}_{ML}$ are

$$\begin{aligned} \hat{\mathbf{t}}_{ML} &= \frac{1}{n} \mathbf{Q} \mathbf{1}_{n \times 1} \\ \hat{\mathbf{A}}_{ML} &= (\mathbf{Q} - \hat{\mathbf{t}}_{ML} \mathbf{1}_{1 \times n}) \mathbf{U}^\dagger \\ &= (\mathbf{Q} - \frac{1}{n} \mathbf{Q} \mathbf{1}_{n \times n}) \\ &= \mathbf{Q} \mathbf{B}, \mathbf{B} \stackrel{def}{=} (\mathbf{I}_{n \times n} - \frac{1}{n} \mathbf{1}_{n \times n}) \mathbf{U}^\dagger \end{aligned} \quad (\text{A.26})$$

where $\mathbf{Q} \stackrel{def}{=} \text{stk}(\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \dots, \tilde{\mathbf{q}}_n)$. To 1st-order, the movement of \mathbf{u}_0 can therefore be described by a linear transform: $f(\mathbf{u}_0) \stackrel{def}{=} \hat{\mathbf{A}}_{ML} \mathbf{u}_0$ followed by a displacement $\hat{\mathbf{t}}_{ML}$. The displacement does not depend on \mathbf{u}_0 , so we are only interested in the error propagation by f . We can write out f as follows:

$$f(\mathbf{u}_0) = \sum_{i=1}^n \begin{bmatrix} \mathbf{b}_i \mathbf{u}_0 & 0 \\ 0 & \mathbf{b}_i \mathbf{u}_0 \end{bmatrix} \hat{\mathbf{q}}_i \quad (\text{A.27})$$

where \mathbf{b}_i denotes the i^{th} row of \mathbf{B} . Because errors in $\hat{\mathbf{q}}_i$ are I.I.D. with variance σ^2 and f is linear in $\hat{\mathbf{q}}_i$, the covariance matrix Σ_f of f is

$$\begin{aligned} [\Sigma_f(\mathbf{u}_0)]_{ij} &= \begin{cases} \sum_{i=1}^n (\mathbf{b}_i \mathbf{u}_0)^2 & i = j \\ 0 & i \neq j \end{cases} \\ &= \begin{cases} (\mathbf{u}_0 - \mathbf{0})^\top \mathbf{B}^\top \mathbf{B} (\mathbf{u}_0 - \mathbf{0}) & i = j \\ 0 & i \neq j \end{cases} \end{aligned} \quad (\text{A.28})$$

Equation (A.28) shows that Σ_f is quadratic in \mathbf{u}_0 , indicating the strong influence \mathbf{u}_0 has on the certainty of its transformed position. The value of \mathbf{u}_0 that minimizes the error covariance is $\mathbf{u}_0 = \mathbf{0}_{2 \times 1}$. Recall that the object points have been zero-centered with the centroid at the origin. Therefore, the value of \mathbf{u}_0 that minimizes the error covariance is the centroid of the object points.

We now consider the uncertainty in \mathbf{J} from noisy correspondences as a function of \mathbf{u}_0 . Ignoring the 2nd-order effects of \mathbf{H} , $\mathbf{J} \approx \hat{\mathbf{A}}_{ML}$ which is independent of \mathbf{u}_0 from Equation (A.28). Consequently, neglecting 2nd-order effects of \mathbf{H} , the value of \mathbf{u}_0 that minimizes the errors-in-variables in Problem (4.12) is the centroid of $\{\mathbf{u}_i\}$. □

Proof of Theorem 14. In P3P there are three non-colinear model points $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2\}$, $\mathbf{u}_i \in \mathbb{R}^2$ and we

have estimates $\{\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2\}$, $\hat{\mathbf{q}}_i \in \mathbb{R}^2$ of their position in the image in normalized coordinates. Without loss of generality let $\mathbf{u}_0 = \mathbf{0}$. The six P3P equations are

$$\begin{aligned} \frac{1}{t_3} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} &= \hat{\mathbf{q}}_0 & (a) \\ \frac{1}{t_3 + [\mathbf{R}_{31} \ \mathbf{R}_{32}] \mathbf{u}_1} \begin{bmatrix} \mathbf{t}_1 + [\mathbf{R}_{11} \ \mathbf{R}_{12}] \mathbf{u}_1 \\ \mathbf{t}_2 + [\mathbf{R}_{21} \ \mathbf{R}_{22}] \mathbf{u}_1 \end{bmatrix} &= \hat{\mathbf{q}}_1 & (b) \\ \frac{1}{t_3 + [\mathbf{R}_{31} \ \mathbf{R}_{32}] \mathbf{u}_2} \begin{bmatrix} \mathbf{t}_1 + [\mathbf{R}_{11} \ \mathbf{R}_{12}] \mathbf{u}_2 \\ \mathbf{t}_2 + [\mathbf{R}_{21} \ \mathbf{R}_{22}] \mathbf{u}_2 \end{bmatrix} &= \hat{\mathbf{q}}_2 & (c) \end{aligned} \quad (\text{A.29})$$

When the length of \mathbf{u}_1 and \mathbf{u}_2 is small Equations (A.29-b,c) can be approximated to 1st-order with a Taylor expansion about $\mathbf{u}_{i \in [1,2]} \approx \mathbf{0}_{2 \times 1}$. After some simplification the expansion writes as

$$\begin{aligned} \frac{1}{t_3} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} &= \hat{\mathbf{q}}_0 & (a) \\ \frac{1}{t_3} \begin{bmatrix} \mathbf{I}_2 & -\frac{1}{t_3} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} \end{bmatrix} \mathbf{R}_{32} \mathbf{U} + \mathcal{O}^2 &= \mathbf{Q} & (b) \end{aligned} \quad (\text{A.30})$$

with

$$\begin{aligned} \mathbf{Q} &\stackrel{\text{def}}{=} [\hat{\mathbf{q}}_1 \ \hat{\mathbf{q}}_2] & (a) \\ \mathbf{U} &\stackrel{\text{def}}{=} [\mathbf{u}_1 \ \mathbf{u}_2] & (b) \end{aligned} \quad (\text{A.31})$$

and \mathcal{O}^2 denoting terms beyond 1st-order. When \mathcal{O}^2 is neglected Equation (A.30) approximates the P3P equations and this approximation becomes better as \mathbf{u}_1 and \mathbf{u}_2 approach the origin. We combine Equations (A.30-a,b) to give the *Infinitesimal P3P Problem*:

$$\begin{aligned} &\text{find } \mathbf{t}_3, \mathbf{R} \quad \text{s.t.} \\ &\left\{ \begin{array}{l} \frac{1}{t_3} \begin{bmatrix} \mathbf{I}_2 & -\hat{\mathbf{q}}_0 \end{bmatrix} \mathbf{R}_{32} = \mathbf{Q} \mathbf{U}^{-1} + \mathcal{O}^2 & (a) \\ \mathbf{R}_{32}^\top \mathbf{R}_{32} = \mathbf{I}_2 & (b) \\ t_3 > 0 & (c) \end{array} \right. \end{aligned} \quad (\text{A.32})$$

Note that because \mathbf{u}_1 and \mathbf{u}_2 are non-co-linear then \mathbf{U} is rank-2 and so is invertible. Problem (4.12) is identical to Problem (A.32) with $\gamma = \frac{1}{t_3}$, $\mathbf{v} = \hat{\mathbf{q}}_0$ and $\mathbf{J} = \mathbf{Q} \mathbf{U}^{-1} + \mathcal{O}_2$. When the separation of the points tends to zero, \mathcal{O}^2 tends to zero the P3P problem becomes the IPPE problem. \square

Proof of Theorem 15. We do this with contradiction. We define as \mathbf{u}_1 , \mathbf{u}_2 and $\mathbf{u}_3 \in \mathbb{R}^2$ the non-co-linear object points. Without loss of generality let $\mathbf{u}_0 = \mathbf{0}_{2 \times 1}$ and $\mathbf{u}_{i \in [1,3]} \neq \mathbf{0}_{2 \times 1}$. We define as $\mathbf{x}_{i \in [1,3]} \in \mathbb{R}^3$ the position of $\mathbf{u}_{i \in [1,3]}$ in camera coordinates with the first pose solution and $\mathbf{y}_{i \in [1,3]} \in \mathbb{R}^3$

the position of \mathbf{u}_i with the second pose solution. From Equation (4.14) we have

$$\mathbf{x}_{i \in [1,3]} = \mathbf{R}_1 \begin{bmatrix} \mathbf{u}_i \\ 0 \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} \quad (\text{A.33})$$

$$\mathbf{y}_{i \in [1,3]} = \mathbf{R}_2 \begin{bmatrix} \mathbf{u}_i \\ 0 \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$$

Pose cannot be disambiguated using the reprojection error of \mathbf{u}_i if $\mathbf{x}_i \propto \mathbf{y}_i$ (they exist along the same line-of-sight). Equivalently, we cannot disambiguate pose using reprojection error if and only if:

$$\forall i \in \{1, 2, 3\} \exists s_i \in \mathbb{R}^+ \text{ s.t.} \\ \mathbf{R}_1 \begin{bmatrix} \mathbf{u}_i \\ 0 \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = s_i \left(\mathbf{R}_2 \begin{bmatrix} \mathbf{u}_i \\ 0 \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} \right) \quad (\text{A.34})$$

Recall that $\mathbf{R}_{i \in [1,2]} = \mathbf{R}_v \tilde{\mathbf{R}}_i$ where $\tilde{\mathbf{R}}_1 = \begin{bmatrix} \mathbf{A} \\ +\mathbf{b} \end{bmatrix}$ and $\tilde{\mathbf{R}}_2 = \begin{bmatrix} \mathbf{A} \\ -\mathbf{b} \end{bmatrix}$ for some $\mathbf{A} \in \mathcal{SS}_{22}$. The

rotation \mathbf{R}_v is defined such that $\mathbf{R}_v^\top \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = [0, 0, 1]^\top$. Left-multiply both sides of Equation (A.34)

by \mathbf{R}_v^\top implies

$$\forall i \in \{1, 2, 3\} \exists s_i \in \mathbb{R}^+ \text{ s.t.} \\ \begin{bmatrix} \frac{1}{\gamma} \mathbf{A} \\ +\mathbf{b}^\top \end{bmatrix} \mathbf{u}_i + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\gamma} \end{bmatrix} = s_i \left(\begin{bmatrix} \frac{1}{\gamma} \mathbf{A} \\ -\mathbf{b}^\top \end{bmatrix} \mathbf{u}_i + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\gamma} \end{bmatrix} \right) \quad (\text{A.35})$$

The top two equations in Equation (A.35) imply

$$\mathbf{A} \mathbf{u}_i = s_i \mathbf{A} \mathbf{u}_i \forall i \in [1, 2, 3] \quad (\text{A.36})$$

Recall that $\gamma \neq 0$ by definition.

Case 1: $\mathbf{b} = \mathbf{0}_{2 \times 1}$. There is no ambiguity because $\mathbf{b} = \mathbf{0}_{2 \times 1} \Leftrightarrow \tilde{\mathbf{R}}_1 = \tilde{\mathbf{R}}_2 \Leftrightarrow \mathbf{R}_1 = \mathbf{R}_2$.

Case 2: $\mathbf{b} \neq \mathbf{0}_{2 \times 1}$ and \mathbf{A} is full-rank. Equation (A.36) implies $s_1 = s_2 = s_3 = 1$. The bottom equation in Equation (A.35) then implies $\mathbf{b}^\top \mathbf{u}_i = -\mathbf{b}^\top \mathbf{u}_i \forall i \in [1, 2, 3]$. Because $\mathbf{b} \neq \mathbf{0}_{2 \times 1}$ this implies $\mathbf{u}_1, \mathbf{u}_2$ and \mathbf{u}_3 are colinear which is a contradiction.

Case 3: $\mathbf{b} \neq \mathbf{0}_{2 \times 1}$ and \mathbf{A} is singular. Because $\mathbf{A} \in \mathcal{SS}_{22}$ it must have rank-1 with a null-vector \mathbf{w} . Let K be the number of elements in the set $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ that are co-linear with \mathbf{w} and therefore satisfy Equation (A.36). Because these elements are non-collinear then $K = 0$ or $K = 1$. If $K = 0$ Equation (A.36) implies $s_1 = s_2 = s_3 = 1$. This is a contradiction similar to case 2. If $K = 1$ then two members of $\{s_1 = s_2 = s_3\}$ must be 1. This implies two members of \mathcal{U} are co-linear which is a contradiction. \square

Bibliography

- [AC95] M. A. Abidi and T. Chandra. “A new efficient and direct solution for pose estimation using quadrangular targets: algorithm and evaluation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.5 (1995), pp. 534–538.
- [Agu+16] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. “Sequential Non-Rigid Structure from Motion Using Physical Priors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.5 (2016), pp. 979–994.
- [Akh+09] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. “Nonrigid Structure from Motion in Trajectory Space”. In: *Advances in Neural Information Processing Systems*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Vol. 21. Curran Associates, Inc., 2009.
- [AM+] Sameer Agarwal, Keir Mierle, et al. *Ceres Solver*. <http://ceres-solver.org>.
- [Bar+12a] Adrien Bartoli, Toby Collins, Nicolas Bourdel, and Michel Canis. “Computer assisted minimally invasive surgery: is medical computer vision the answer to improving laparosurgery?” In: *Medical Hypotheses* 79.6 (2012), pp. 858–863.
- [Bar+12b] Adrien Bartoli, Y. Gérard, F. Chadebecq, and Toby Collins. “On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [Bar+15] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. “Shape-from-Template”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.10 (2015), pp. 2099–2118.
- [Bar+18] Dan Barnes, Will Maddern, Geoffrey Pascoe, and Ingmar Posner. “Driven to Distraction: Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments”. In: *International Conference on Robotics and Automation*. 2018, pp. 1894–1900.
- [Bar08] Adrien Bartoli. “Groupwise Geometric and Photometric Direct Image Registration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.12 (2008), pp. 2098–2108.
- [BBM09] T. Brox, C. Bregler, and J. Malik. “Large displacement optical flow”. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

- [BC13a] A. Bartoli and T. Collins. “Template-Based Isometric Deformable 3D Reconstruction with Sampling-Based Focal Length Self-Calibration”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.
- [BC13b] Adrien Bartoli and Toby Collins. “Template-Based Isometric Deformable 3D Reconstruction with Sampling-Based Focal Length Self-Calibration”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR) (2013)*, pp. 1514–1521.
- [BC18] Adrien Bartoli and Toby Collins. “Plane-Based Resection for Metric Affine Cameras”. In: *Journal of Mathematical Imaging and Vision* 60.7 (2018), pp. 1037–1064.
- [BCP15] Adrien Bartoli, Toby Collins, and Daniel Pizarro. “Metric Corrections of the Affine Camera”. In: *Computer Vision and Image Understanding* 135.C (2015), pp. 141–156.
- [BHB00] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. “Recovering non-rigid 3D shape from image streams”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2000, pp. 690–696.
- [BHB14] F. Brunet, R. Hartley, and A. Bartoli. “Monocular Template-Based 3D Surface Reconstruction: Convex Inextensible and Nonconvex Isometric Methods”. In: *Computer Vision and Image Understanding* 125 (2014), pp. 138–154.
- [BKP08] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. “A general solution to the P4P problem for camera with unknown focal length”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008, pp. 1–8.
- [Bog+18] Oleksandr Bogdan, Viktor Eckstein, Francois Rameau, and Jean-Charles Bazin. “Deep-Calib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras”. In: *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production*. CVMP ’18. London, United Kingdom: Association for Computing Machinery, 2018.
- [Bou+16a] N Bourdel, T Collins, D Pizarro, P Chauvet, C Debize, A Bartoli, and M Canis. “First Use of Augmented Reality in Gynecology”. In: *Journal of Minimally Invasive Gynecology* 23.7 (2016), pp. 226–227.
- [Bou+16b] N. Bourdel, T. Collins, Daniel Pizarro-Perez, A. Bartoli, D. Da Inès, B. Perreira, and M. Canis. “Augmented reality in gynecologic surgery: evaluation of potential benefits for myomectomy in an experimental uterine model”. In: *Surgical endoscopy* 31 (2016), pp. 456–461.
- [Bou+17] Nicolas Bourdel, Toby Collins, Daniel Pizarro, Clement Debize, Anne-sophie Grémeau, Adrien Bartoli, and Michel Canis. “Use of augmented reality in laparoscopic gynecology to visualize myomas”. In: *Journal of Fertility and Sterility* 107.3 (2017), pp. 737–739.
- [Bou00] J.Y. Bouguet. “Matlab Camera Calibration Toolbox”. In: 2000.
- [BPC13] A. Bartoli, D. Pizarro, and T. Collins. “A Robust Analytical Solution to Isometric Shape-from-Template with Focal Length Calibration”. In: *International Conference on Computer Vision (ICCV)*. 2013.
- [Bra+18] Samarth Brahmhbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. “Geometry-Aware Learning of Maps for Camera Localization”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

- [Bra00] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).
- [Bra01] W. Brand. “Morphable 3D models from video”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 2001, pp. II–II.
- [BRG16] Aayush Bansal, Bryan Russell, and Abhinav Gupta. “Marr Revisited: 2D-3D Alignment via Surface Normal Prediction”. In: 2016, pp. 5965–5974.
- [Bro71] Duane C. Brown. “Close-range camera calibration”. In: *Photogrammetric Engineering* 37.8 (1971), pp. 855–866.
- [Bru+10] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. “Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces”. In: *Asian Conference on Computer Vision (ACCV)*. 2010.
- [Cao+19] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).
- [Cas+19] David Casillas-Perez, Daniel Pizarro, David Fuentes-Jimenez, Manuel Mazo, and Adrien Bartoli. “Equiareal Shape-from-Template”. In: *Journal of Mathematical Imaging and Vision* 61.5 (2019), pp. 607–626.
- [CB10a] Toby Collins and Adrien Bartoli. “Locally Planar and Affine Deformable Surface Reconstruction from Video”. In: *International Workshop on Vision, Modeling and Visualization (VMV)*. 2010.
- [CB10b] Toby Collins and Adrien Bartoli. “Locally Planar and Affine Deformable Surface Reconstruction from Video.” In: *VMV*. Ed. by Reinhard Koch, Andreas Kolb, and Christof Rezk-Salama. 2010, pp. 339–346.
- [CB12a] Toby Collins and Adrien Bartoli. “3D reconstruction in laparoscopy with close-range photometric stereo”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2012, pp. 634–642.
- [CB12b] Toby Collins and Adrien Bartoli. “Towards live monocular 3D laparoscopy using shading and specular information”. In: *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)*. 2012, pp. 11–21.
- [CB14a] T. Collins and A. Bartoli. “Infinitesimal plane-based pose estimation”. In: *International Journal of Computer Vision* 109.3 (2014), pp. 252–286.
- [CB14b] Toby Collins and Adrien Bartoli. “Using Isometry to Classify Correct/Incorrect 3D-2D Correspondences”. In: *European Conference on Computer Vision (ECCV)*. Cham: Springer International Publishing, 2014, pp. 325–340.
- [CB15] T. Collins and A. Bartoli. “Realtime Shape-from-Template: System and Applications”. In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. 2015.
- [CB17] T. Collins and A. Bartoli. “Planar Structure-from-Motion with Affine Camera Models: Closed-Form Solutions, Ambiguities and Degeneracy Analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1237–1255.
- [CC04] Chu-Song Chen and Wen-Yan Chang. “On pose recovery for generalized visual sensors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.7 (2004), pp. 848–861.

- [CCB11] Toby Collins, Benoit Compte, and Adrien Bartoli. “Deformable Shape-From-Motion in Laparoscopy using a Rigid Sliding Window.” In: *Conference on Medical Image Understanding and Analysis (MIUA)*. 2011, pp. 173–178.
- [Cha+12] François Chadebecq, Christophe Tilmant, Peyras Julien, Toby Collins, and Adrien Bartoli. “Estimation de l'échelle en coloscopie monoculaire par quantification du flou optique: étude de faisabilité”. In: *Reconnaissance des Formes et Intelligence Artificielle (RFIA)*. 2012.
- [Cha+18] Pauline Chauvet, Toby Collins, Clement Debize, Lorraine Novais-Gameiro, Bruno Pereira, Adrien Bartoli, Michel Canis, and Nicolas Bourdel. “Augmented reality in a tumor resection model”. In: *Surgical endoscopy* 32.3 (2018), pp. 1192–1201.
- [Che+20] Changhao Chen, B. Wang, Chris Xiaoxuan Lu, A. Trigoni, and A. Markham. “A Survey on Deep Learning for Localization and Mapping: Towards the Age of Spatial Machine Intelligence”. In: *ArXiv abs/2006.12567* (2020).
- [Chh+14] Ajad Chhatkuli, Adrien Bartoli, Abed Malti, and Toby Collins. “Live image parsing in uterine laparoscopy”. In: *International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2014, pp. 1263–1266.
- [Chh+16] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. “Inextensible Non-Rigid Shape-from-Motion by Second-Order Cone Programming”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [Chh+17a] A. Chhatkuli, D. Pizarro, A. Bartoli, and T. Collins. “A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.5 (2017), pp. 833–850.
- [Chh+17b] Ajad Chhatkuli, Daniel Pizarro, Adrien Bartoli, and Toby Collins. “A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.5 (2017), pp. 833–850.
- [Cig+08] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. “MeshLab: an Open-Source Mesh Processing Tool”. In: *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.
- [CJQ20] Song Chen, Song Jiaru, and Huang Qixing. “HybridPose: 6D Object Pose Estimation under Hybrid Representations”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 428–437.
- [CLS10] Kyuhyoung Choi, Subin Lee, and Yongduek Seo. “A Branch-and-bound Algorithm for Globally Optimal Camera Pose and Focal Length”. In: *Image and Vision Computing* 28.9 (2010), pp. 1369–1376.
- [CM05] O. Chum and J. Matas. “Matching with PROSAC - progressive sample consensus”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2005, pp. 220–226.
- [CMB14] Toby Collins, Pablo Mesejo, and Adrien Bartoli. “An analysis of errors in graph-based keypoint matching and proposed solutions”. In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 138–153.

- [Col+10] Toby Collins, Jean-Denis Durou, Pierre Gurdjos, and Adrien Bartoli. “Single view perspective shape-from-texture with focal length estimation: A piecewise affine approach”. In: *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2010).
- [Col+14] Toby Collins, Daniel Pizarro, Adrien Bartoli, Michel Canis, and Nicolas Bourdel. “Computer-assisted laparoscopic myomectomy by augmenting the uterus with pre-operative MRI data”. In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. 2014, pp. 243–248.
- [Col+15] Toby Collins, Adrien Bartoli, Nicolas Bourdel, and Michel Canis. “Segmenting the uterus in monocular laparoscopic images without manual input”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, pp. 181–189.
- [Col+16a] Toby Collins, Adrien Bartoli, Nicolas Bourdel, and Michel Canis. “Robust, Real-Time, Dense and Deformable 3D Organ Tracking in Laparoscopic Videos”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2016.
- [Col+16b] Toby Collins, Pauline Chauvet, Clement Debize, Daniel Pizarro, Adrien Bartoli, Michel Canis, and Nicolas Bourdel. “A System for Augmented Reality Guided Laparoscopic Tumour Resection with Quantitative Ex-vivo User Evaluation”. In: *Computer-Assisted and Robotic Endoscop (CARE)*. 2016.
- [Col+21] T. Collins, D. Pizarro, S. Gasparini, N. Bourdel, P. Chauvet, M. Canis, L. Calvet, and A. Bartoli. “Augmented Reality Guided Laparoscopic Surgery of the Uterus”. In: *IEEE Transactions on Medical Imaging* 40.1 (2021), pp. 371–380.
- [CPB14a] A. Chhatkuli, D. Pizarro, and A. Bartoli. “Stable Template-Based Isometric 3D Reconstruction in All Imaging Conditions by Linear Least-Squares”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.
- [CPB14b] Ajad Chhatkuli, Daniel Pizarro, and Adrien Bartoli. “Non-Rigid Shape-from-Motion for Isometric Surfaces using Infinitesimal Planarity”. In: *British Machine Vision Conference (BMVC)*. 2014.
- [CS09] Pei Chen and David Suter. “Error Analysis in Homography Estimation by First Order Approximation Tools: A General Technique”. In: *Journal of Mathematical Imaging and Vision* 33 (2009), pp. 281–295.
- [DD92] D. DeMenthon and L.S. Davis. “Exact and approximate solutions of the perspective-three-point problem”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.11 (1992), pp. 1100–1105.
- [DD95] Daniel F. Dementhon and Larry S. Davis. “Model-based object pose in 25 lines of code”. In: *International Journal of Computer Vision* 15.1 (1995), pp. 123–141.
- [DLH12] Y. Dai, H. Li, and M. He. “A simple prior-free method for non-rigid structure-from-motion factorization”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2018–2025.
- [DMR16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Deep Image Homography Estimation”. In: *ArXiv abs/1606.03798* (2016).

- [DMR18] D. DeTone, T. Malisiewicz, and A. Rabinovich. “SuperPoint: Self-Supervised Interest Point Detection and Description”. In: *Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 337–33712.
- [dOr+14] Laurent d’Orazio, Adrien Bartoli, Andre Baetz, Sylvain Beorchia, Gaëlle Calvary, Yahia Chabane, Francois Chadebecq, Toby Collins, Yann Laurillau, Laure Martins-Baltar, et al. “Multimodal and multimedia image analysis and collaborative networking for digestive endoscopy”. In: *IRBM* 35.2 (2014), pp. 88–93.
- [Dos+15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. “FlowNet: Learning Optical Flow with Convolutional Networks”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [DRL89] M. Dhome, M. Richetin, and J.-T. Lapreste. “Determination of the Attitude of 3D Objects from a Single Perspective View”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (1989), pp. 1265–1278.
- [DSA07] A. Del Bue, F. Smeraldi, and L. Agapito. “Non-Rigid Structure from Motion Using Ranklet-Based Tracking and Non-Linear Optimization”. In: *Image and Vision Computing* 25.3 (2007), pp. 297–310.
- [Du+19] Simon S. Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. “Gradient Descent Provably Optimizes Over-parameterized Neural Networks”. In: 2019.
- [Dus+19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. “D2-Net: A Trainable CNN for Joint Detection and Description of Local Features”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [EF15] D. Eigen and R. Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture”. In: *International Conference on Computer Vision (ICCV)*. 2015, pp. 2650–2658.
- [ENG07] Andreas Ess, Alexander Neubeck, and Luc Van Gool. “Generalised Linear Pose Estimation”. In: *British Machine Vision Conference (BMVC)*. Ed. by Nasir M. Rajpoot and Abhir H. Bhalerao. 2007, pp. 1–10.
- [EPF14] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network”. English (US). In: vol. 3. 2014, pp. 2366–2374.
- [ESC14] Jakob Engel, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-Scale Direct Monocular SLAM”. In: *European Conference on Computer Vision (ECCV)*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 834–849.
- [Eye18] EyeCue Vision Technologies. *Qlone*. <https://www.qlone.pro//>. 2018.
- [Fau+12] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, and Stéphane Cotin. “SOFA: A Multi-Model Framework for Interactive Physical Simulation”. In: *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Vol. 11. Studies in Mechanobiology, Tissue Engineering and Biomaterials. 2012, pp. 283–321.

- [Fau95] Olivier Faugeras. “Stratification of three-dimensional vision: projective, affine, and metric representations”. In: *Journal of the Optical Society of America A* 12.3 (1995), pp. 465–484.
- [Fay+09] J. Fayad, A. Del Bue, L. Agapito, and P.M.Q. Aguiar. “Non-rigid Structure from Motion using Quadratic Deformation Models”. In: *British Machine Vision Conference (BMVC)*. 2009.
- [FB81] Martin A. Fischler and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (1981), pp. 381–395.
- [Fit01] A. W. Fitzgibbon. “Simultaneous linear estimation of multiple view geometry and lens distortion”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2001, pp. I–I.
- [FL88] Olivier Faugeras and F. Lustman. *Motion and structure from motion in a piecewise planar environment*. Tech. rep. RR-0856. INRIA, 1988.
- [FLM92] O. D. Faugeras, Q. -T. Luong, and S. J. Maybank. “Camera self-calibration: Theory and experiments”. In: *European Conference on Computer Vision (ECCV)*. Ed. by G. Sandini. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 321–334.
- [FLP01] Olivier Faugeras, Quang-Tuan Luong, and T. Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. Cambridge, MA, USA: MIT Press, 2001.
- [FP12] David A. Forsyth and Jean Ponce. *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012.
- [Fue+18] David Fuentes-Jimenez, David Casillas-Perez, Daniel Pizarro, Toby Collins, and Adrien Bartoli. *Deep Shape-from-Template: Wide-Baseline, Dense and Fast Registration and Deformable Reconstruction from a Single Image*. arXiv:1811.07791. 2018.
- [Fue+21] David Fuentes-Jimenez, Daniel Pizarro, David Casillas-Perez, Toby Collins, and Adrien Bartoli. “Texture-Generic Deep Shape-From-Template”. In: *IEEE Access* 9 (2021), pp. 75211–75230.
- [GAB17] Clément Godard, Oisín Mac Aodha, and Gabriel J. Brostow. “Unsupervised Monocular Depth Estimation with Left-Right Consistency”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6602–6611.
- [Gal+15] Mathias Gallardo, Daniel Pizarro, Adrien Bartoli, and Toby Collins. “Shape-from-template in flatland”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2847–2854.
- [Gal+20] Mathias Gallardo, Daniel Pizarro, Toby Collins, and Adrien Bartoli. “Shape-from-template with curves”. In: *International Journal of Computer Vision* 128.1 (2020), pp. 121–165.
- [Gao+03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. “Complete Solution Classification for the Perspective-Three-Point Problem”. In: 25.8 (2003), pp. 930–943.

- [Gar+16a] Ravi Garg, B. V. Kumar, G. Carneiro, and I. Reid. “Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue”. In: *European Conference on Computer Vision (ECCV)*. 2016.
- [Gar+16b] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and R. Medina-Carnicer. “Generation of fiducial marker dictionaries using Mixed Integer Linear Programming”. In: *Pattern Recognition* 51 (2016), pp. 481–491.
- [GCB16a] M. Gallardo, T. Collins, and A. Bartoli. “Can we Jointly Register and Reconstruct Creased Surfaces by Shape-from-Template Accurately?” In: *European Conference on Computer Vision (ECCV)*. 2016.
- [GCB16b] M. Gallardo, T. Collins, and A. Bartoli. “Using Shading and a 3D Template to Reconstruct Complex Surface Deformations”. In: *British Machine Vision Conference (BMVC)*. 2016.
- [GCB16c] Mathias Gallardo, Toby Collins, and Adrien Bartoli. “Can We Jointly Register and Reconstruct Creased Surfaces by Shape-from-Template Accurately?” In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 105–120.
- [GCB17] M. Gallardo, T. Collins, and A. Bartoli. “Dense Non-Rigid Structure-from-Motion and Shading with Unknown Albedos”. In: *International Conference on Computer Vision (ICCV)*. 2017.
- [GM11] Paulo F. U. Gotardo and Aleix M. Martinez. “Kernel non-rigid structure from motion”. In: *International Conference on Computer Vision (ICCV)*. 2011, pp. 802–809.
- [GNK18] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. “DensePose: Dense Human Pose Estimation in the Wild”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7297–7306.
- [Gol+18] Vladislav Golyanik, Soshi Shimada, Kiran Varanasi, and Didier Stricker. “HDM-Net: Monocular Non-Rigid 3D Reconstruction with Learned Deformation Model”. In: 2018.
- [Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. In: *Neural Information Processing Systems Conference (NIPS)*. Vol. 27. Curran Associates, Inc., 2014.
- [Gru41] J.A. Grunert. “Das Pothenotische Problem in erweiterter Gestalt nebst ber seine Anwendungen in der Geodsie.”. In: *Grunerts Archiv fr Mathematik und Physik* (1841).
- [GS03] P. Gurdjos and P. Sturm. “Methods and geometry for plane-based self-calibration”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*. Vol. 1. 2003, pp. I–I.
- [Guo13] Yang Guo. “A Novel Solution to the P4P Problem for an Uncalibrated Camera”. In: *Journal of Mathematical Imaging and Vision* 45.2 (2013), pp. 186–198.
- [Hao+15] Nazim Haouchine, Jeremie Dequidt, Marie-Odile Berger, and Stephane Cotin. “Monocular 3D Reconstruction and Augmentation of Elastic Surfaces with Self-occlusion Handling”. In: *IEEE Transactions on Visualization and Computer Graphics* (2015), p. 14.
- [Har+91] R.M. Haralick, D. Lee, K. Ottenburg, and M. Nolle. “Analysis and solutions of the three point perspective pose estimation problem”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991, pp. 592–598.

- [Har+94] Bert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. “Review and analysis of solutions of the three point perspective pose estimation problem”. In: *International Journal of Computer Vision* 13.3 (1994), pp. 331–356.
- [HB86] D. D. Hoffman and B. M. Bennett. “The computation of structure from fixed-axis motion: rigid structures”. In: *Biological Cybernetics* 54.2 (1986), pp. 71–83.
- [HK07] Richard Hartley and Fredrik Kahl. “Critical Configurations for Projective Reconstruction from Multiple Views”. In: *International Journal of Computer Vision* 71.1 (2007), pp. 5–47.
- [HL89] T. S. Huang and C. H. Lee. “Motion and Structure from Orthographic Projections”. In: 11.5 (1989), pp. 536–540.
- [HLL09] Didier Henrion, Jean-Bernard Bernard Lasserre, and Johan Lofberg. “GloptiPoly 3: moments, optimization and semidefinite programming”. In: *Optimization Methods and Software* 24.4-5 (2009), pp. 761–779.
- [HO05] Matthew Harker and Paul O’Leary. “Computation of Homographies”. In: *British Machine Vision Conference (BMVC)*. 2005.
- [Hol+18] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matt Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. “A Perceptual Measure for Deep Single Image Camera Calibration”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2354–2363.
- [Hor+89] R. Horaud, B. Conio, O. Le Boulleux, and B. Lacolle. “An analytic solution for the perspective 4-point problem”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1989, pp. 500–507.
- [Hor+97] Radu Horaud, Fadi Dornaika, Bart Lamiroy, and S. Christy. “Object Pose: The Link between Weak Perspective, Paraperspective and Full Perspective”. In: *International Journal of Computer Vision* 22 (1997), pp. 173–189.
- [HR11] J. A. Hesch and S. I. Roumeliotis. “A Direct Least-Squares (DLS) method for PnP”. In: *International Conference on Computer Vision (ICCV)*. 2011, pp. 383–390.
- [HR19] Junhwa Hur and Stefan Roth. “Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [HS97] Janne Heikkila and Olli Silven. “A Four-Step Camera Calibration Procedure with Implicit Image Correction”. In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR ’97)*. CVPR ’97. USA: IEEE Computer Society, 1997, p. 1106.
- [HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [ISF07] Slobodan Ilic, Mathieu Salzmann, and Pascal Fua. “Implicit Meshes for Effective Silhouette Handling”. In: *International Journal of Computer Vision* 72 (2007), pp. 159–178.
- [JB09] K. Josephson and M. Byrod. “Pose estimation with radial distortion and unknown focal length”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 2419–2426.

- [Kab09] Kabaq.io. *Kabaq smartphone app*. <https://www.kabaq.io/>. 2009.
- [KBP13] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. “Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length”. In: *International Conference on Computer Vision (ICCV)*. ICCV ’13. 2013, pp. 2816–2823.
- [KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization”. In: *International Conference on Computer Vision (ICCV)*. ICCV ’15. USA, 2015, pp. 2938–2946.
- [Kim+12] Jae-Hak Kim, Adrien Bartoli, Toby Collins, and Richard Hartley. “Tracking by detection for interactive image augmentation in laparoscopy”. In: *International Workshop on Biomedical Image Registration*. 2012, pp. 246–255.
- [KL19] Chen Kong and Simon Lucey. *Deep Non-Rigid Structure from Motion*. 2019.
- [KL20] Chen Kong and Simon Lucey. “Deep Non-Rigid Structure from Motion with Missing Data”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1.
- [KLS14] Laurent Kneip, Hongdong Li, and Yongduek Seo. “UPnP: An Optimal $O(n)$ Solution to the Absolute Pose Problem with Universal Applicability”. In: *European Conference on Computer Vision (ECCV)*. Cham, 2014, pp. 127–142.
- [KM98] T. Kanade and Daniel D. Morris. “Factorization methods for structure from motion”. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 356 (1998), pp. 1153–1173.
- [Koo+17] Bongjin Koo, Erol Özgür, Bertrand Le Roy, Emmanuel Buc, and Adrien Bartoli. “Deformable Registration of a Preoperative 3D Liver Volume to a Laparoscopy Image Using Contour and Shading Cues”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2017.
- [KR17] T. Ke and S. I. Roumeliotis. “An Efficient Algebraic Solution to the Perspective-Three-Point Problem”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4618–4626.
- [KS15] Nima Keivan and G. Sibley. “Online SLAM with any-time self-calibration and automatic change detection”. In: *International Conference on Robotics and Automation* (2015), pp. 5775–5782.
- [KS98] Ron Kimmel and JA Sethian. “Computing Geodesic Paths on Manifolds”. In: *Proceedings of the National Academy of Sciences of the United States of America* 95 (1998), pp. 8431–5.
- [KSA07] Kenichi Kanatani, Yasuyuki Sugaya, and Hanno Ackermann. “Uncalibrated Factorization Using a Variable Symmetric Affine Camera”. In: *IEICE Transactions* 90-D.5 (2007), pp. 851–858.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems Conference (NIPS)*. 2012, pp. 1097–1105.
- [Kum19] Suryansh Kumar. “A Simple Prior-Free Method for Non-rigid Structure-from-Motion Factorization : Revisited”. In: *CoRR* abs/1902.10274 (2019).

- [LeC+98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [LG15] Zohar Levi and Craig Gotsman. “Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation”. In: *IEEE Transactions on Visualization and Computer Graphics* 21.2 (2015), pp. 264–277.
- [LHM00] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. “Fast and Globally Convergent Pose Estimation from Video Images”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.6 (2000), pp. 610–622.
- [Li+12] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. “Worldwide Pose Estimation Using 3D Point Clouds”. In: *European Conference on Computer Vision (ECCV)*. ECCV’12. Florence, Italy, 2012, pp. 15–29.
- [Liu+16a] Fayao Liu, Chunhua Shen, Guosheng Lin, and D. Ian Reid. “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016), pp. 2024–2039.
- [Liu+16b] Q. Liu-Yin, R. Yu, L. Agapito, A. Fitzgibbon, and C. Russell. “Better Together: Joint Reasoning for Non-rigid 3D Reconstruction with Specularities and Shading”. In: *British Machine Vision Conference (BMVC)*. 2016.
- [LMF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EPnP: An Accurate O(n) Solution to the PnP Problem”. In: *International Journal of Computer Vision* 81 (2009), pp. 155–166.
- [Lon81] H.C. Longuet Higgins. “A Computer Algorithm for Reconstructing a Scene from Two Projections”. In: *Nature* 293 (1981).
- [Lon87] H. C. Longuet-Higgins. “Readings in Computer Vision: Issues, Problems, Principles, and Paradigms”. In: ed. by Martin A. Fischler and Oscar Firschein. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987. Chap. A Computer Algorithm for Reconstructing a Scene from Two Projections, pp. 61–62.
- [Low04a] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision* 60.2 (2004), pp. 91–110.
- [Low04b] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60 (2004), pp. 91–110.
- [LSH10] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. “Location Recognition Using Prioritized Feature Matching”. In: *European Conference on Computer Vision (ECCV)*. ECCV’10. 2010, pp. 791–804.
- [Luo+18] Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. “LSTM Pose Machines”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- [LV96] Q.-T. Luong and T. Vieville. “Canonical Representations for the Geometries of Multiple Projective Views”. In: *Computer Vision and Image Understanding* 64.2 (1996), pp. 193–229.
- [LWJ19] Zhigang Li, Gu Wang, and Xiangyang Ji. “CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation”. In: *International Conference on Computer Vision (ICCV)*. 2019.

- [LXX12] Shiqi Li, Chi Xu, and Ming Xie. “A Robust $O(n)$ Solution to the Perspective-n-Point Problem”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012).
- [M12] Bujnak M. “Algebraic solutions to absolute pose problems”. In: *PhD Thesis, Czech Technical University* (2012).
- [Mag+15] S. Magnenat, D. Ngo, Fabio Zünd, Mattia Ryffel, Gioacchino Noris, Gerhard Roethlin, Alessia Marra, Maurizio Nitti, P. Fua, M. Gross, and R. W. Sumner. “Live Texturing of Augmented Reality Characters from Colored Drawings”. In: *IEEE Transactions on Visualization and Computer Graphics* 21 (2015), pp. 1201–1210.
- [Mal+13] Abed Malti, Richard Hartley, Adrien Bartoli, and Jae-Hak Kim. “Monocular Template-Based 3D Reconstruction of Extensible Surfaces with Local Linear Elasticity”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR ’13. 2013, pp. 1522–1529.
- [Mar+17] Julieta Martinez, Rayat Hossain, Javier Romero, and J. James Little. “A simple yet effective baseline for 3D human pose estimation”. In: *International Conference on Computer Vision (ICCV)*. 2017.
- [MBC11] Abed Malti, Adrien Bartoli, and Toby Collins. “A pixel-based approach to template-based monocular 3D reconstruction of deformable surfaces”. In: *International Conference on Computer Vision Workshops*. 2011, pp. 1650–1657.
- [MBC12a] A. Malti, A. Bartoli, and T. Collins. “Template-Based Conformal Shape-from-Motion-and-Shading for Laparoscopy”. In: *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)*. 2012.
- [MBC12b] Abed Malti, Adrien Bartoli, and Toby Collins. “Template-Based Conformal Shape-from-Motion-and-Shading for Laparoscopy”. In: *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)*. 2012, pp. 1–10.
- [MC02] E. Malis and R. Cipolla. “Camera Self-Calibration from Unknown Planar Structures Enforcing the Multiview Constraints between Collineations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), pp. 1268–1272.
- [MC09] M. Marques and J.P.” Costeira. “Estimating 3D shape from degenerate sequences with missing data”. In: *Computer Vision and Image Understanding* 113.2 (2009), pp. 261–272.
- [MCB11] Abed Malti, Toby Collins, and Adrien Bartoli. “Template-Based Deformable Shape-from Motion from Registered Laparoscopic Images”. In: *Conference on Medical Image Understanding and Analysis (MIUA)* (2011).
- [Mel+17a] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. “Image-Based Localization Using Hourglass Networks”. In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017, pp. 870–877.
- [Mel+17b] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. *Relative Camera Pose Estimation Using Convolutional Neural Networks*. 2017.
- [Men+08] J. F. Menudet, J. M. Becker, T. Fournel, and C. Mennessier. “Plane-based camera self-calibration by metric rectification of images”. In: *Image and Vision Computing* 26.7 (2008), pp. 913–934.
- [Min+21] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. “Deep learning for monocular depth estimation: A review”. In: *Neurocomputing* 438 (2021), pp. 14–33.

- [Mor+09] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. “Capturing 3D Stretchable Surfaces from Single Images in Closed Form”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.
- [MOS19] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0*. 2019.
- [Mou+16] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. “OpenMVG: Open multiple view geometry”. In: *International Workshop on Reproducible Research in Pattern Recognition*. 2016, pp. 60–74.
- [Muñ+18] Rafael Muñoz-Salinas, Manuel J. Marín-Jimenez, Enrique Yeguas-Bolivar, and R. Medina-Carnicer. “Mapping and localization from planar markers”. en. In: *Pattern Recognition 73* (2018), pp. 158–171.
- [MV07] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. Research Report RR-6303. INRIA, 2007.
- [Nak15] Gaku Nakano. “Globally Optimal DLS Method for PnP Problem with Cayley parameterization”. In: *British Machine Vision Conference (BMVC)*. 2015, pp. 78.1–78.11.
- [Nak16] Gaku Nakano. “A Versatile Approach for Solving PnP, PnPf, and PnPfr Problems”. In: *European Conference on Computer Vision (ECCV)*. Cham, 2016, pp. 338–352.
- [NB17] T. Naseer and W. Burgard. “Deep regression for monocular camera-based 6-DoF global localization in outdoor environments”. In: *International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1525–1530.
- [Ngo+15] Tien Dat Ngo, Sanghyuk Park, Anne Alison Jorstad, Alberto Crivellaro, Chang Yoo, and Pascal Fua. “Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture”. In: *International Conference on Computer Vision (ICCV)*. 2015.
- [Nis03] Nister. “Preemptive RANSAC for live structure and motion estimation”. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 199–206 vol.1.
- [Nis04a] D. Nister. “A minimal solution to the generalised 3-point pose problem”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2004, pp. I–I.
- [Nis04b] David Nistér. “An Efficient Solution to the Five-Point Relative Pose Problem”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6 (2004), pp. 756–777.
- [Nov09] Novarama. *Invizimals PSP Game*. <https://www.playstation.com/fr-ch/ps5/games/>. 2009.
- [ODD96] Denis Oberkampf, Daniel DeMenthon, and Larry S. Davis. “Iterative Pose Estimation Using Coplanar Feature Points.” In: *Computer Vision and Image Understanding* 63 (1996).
- [Ono+18] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. “LF-Net: Learning Local Features from Images”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Montréal, Canada: Curran Associates Inc., 2018, pp. 6237–6247.
- [Ost+12] J. Ostlund, A. Varol, T. Ngo, and P. Fua. “Laplacian Meshes for Monocular 3D Shape Recovery”. In: *European Conference on Computer Vision (ECCV)* (2012).

- [Özy+17] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. “A survey of structure from motion.” In: *Acta Numerica* 26 (2017), pp. 305–364.
- [PAM13] Adrian Penate-Sanchez, Juan Andrade-Cetto, and Francesc Moreno-Noguer. “Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.10 (2013), pp. 2387–2400.
- [Par+15] Shaifali Parashar, Daniel Pizarro, Adrien Bartoli, and Toby Collins. “As-Rigid-as-Possible Volumetric Shape-from-Template”. In: *International Conference on Computer Vision (ICCV)*. USA, 2015, pp. 891–899.
- [PB12] Daniel Pizarro and Adrien Bartoli. “Feature-Based Deformable Surface Detection with Self-Occlusion Reasoning”. In: *International Journal of Computer Vision* 97.1 (2012), pp. 54–70.
- [PBC13] D. Pizarro, A. Bartoli, and T. Collins. “Isowarp and Conwarp: Warps that Exactly Comply with Weak Perspective Projection of Deforming Objects”. In: *British Machine Vision Conference (BMVC)*. 2013.
- [PBP18] Shaifali Parashar, Adrien Bartoli, and Daniel Pizarro. “Self-Calibrating Isometric Non-Rigid Structure-from-Motion”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
- [PCB15] Kristina Prokopetc, Toby Collins, and Adrien Bartoli. “Automatic detection of the uterus and fallopian tube junctions in laparoscopic images”. In: *Information Processing in Medical Imaging (IPMI)*. 2015, pp. 552–563.
- [Pen+19] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. “PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [PG99] M. Pollefeys and L. van Gool. “Stratified self-calibration with the modulus constraint”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.8 (1999), pp. 707–724.
- [PHB11] Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. “Monocular Template-based Reconstruction of Inextensible Surfaces”. In: *International Journal of Computer Vision* 95.2 (2011), pp. 124–137.
- [PKG98] M. Pollefeys, R. Koch, and L. Gool. “Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters”. In: *International Conference on Computer Vision (ICCV)* (1998), pp. 90–95.
- [PLF08] Julien Pilet, Vincent Lepetit, and Pascal Fua. “Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation”. In: *International Journal of Computer Vision* 76.2 (2008), pp. 109–122.
- [PPB18] Shaifali Parashar, Daniel Pizarro, and Adrien Bartoli. “Isometric Non-Rigid Shape-from-Motion with Riemannian Geometry Solved in Linear Time”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10 (2018), pp. 2442–2454.
- [PPB20] Shaifali Parashar, Daniel Pizarro, and Adrien Bartoli. “Local Deformable 3D Reconstruction with Cartan’s Connections”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.12 (2020), pp. 3011–3026.

- [PPV19] Kiru Park, Timothy Patten, and Markus Vincze. “Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation”. In: *International Conference on Computer Vision (ICCV)*. 2019.
- [Pro+18] T. Probst, D. Pani Paudel, A. Chhatkuli, and L. Van Gool. “Incremental Non-Rigid Structure-from-Motion with Unknown Focal Length”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 756–771.
- [Pum+18] Albert Pumarola, Antonio Agudo, Lorenzo Porzi, Alberto Sanfeliu, Vincent Lepetit, and Francesc Moreno-Noguer. “Geometry-Aware Network for Non-Rigid Shape Prediction From a Single View.” In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018, pp. 4681–4690.
- [QL99] Long Quan and Zhongdan Lan. “Linear N-point camera pose determination”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.8 (1999), pp. 774–780.
- [Qua94] Long Quan. “Self-calibration of an Affine Camera from Multiple Views”. In: *International Journal of Computer Vision* 19 (1994), pp. 93–105.
- [Ran+19] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. “Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 12232–12241.
- [Rev+19] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. “R2D2: Repeatable and Reliable Detector and Descriptor”. In: *Neural Information Processing Systems Conference (NIPS)*. 2019.
- [RL17] Mahdi Rad and Vincent Lepetit. “BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth”. In: *International Conference on Computer Vision (ICCV)* (2017).
- [RMM18] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. “Speeded up detection of squared fiducial markers”. In: *Image and Vision Computing* 76 (2018), pp. 38–47.
- [Rus+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.
- [RYA14] Chris Russell, Rui Yu, and L. Agapito. “Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes”. In: *European Conference on Computer Vision (ECCV)*. 2014.
- [SA07] Olga Sorkine and Marc Alexa. “As-rigid-as-possible Surface Modeling”. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. SGP ’07. Barcelona, Spain: Eurographics Association, 2007, pp. 109–116.
- [Sal+08] Mathieu Salzmann, Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. “Closed-Form Solution to Non-rigid 3D Surface Registration”. In: *European Conference on Computer Vision (ECCV)*. 2008, pp. 581–594.
- [SBK10] N. Sundaram, T. Brox, and K. Keutzer. “Dense point trajectories by GPU-accelerated large displacement optical flow”. In: *European Conference on Computer Vision (ECCV)*. Lecture Notes in Computer Science. 2010.

- [SF09] M. Salzmann and P. Fua. “Reconstructing sharply folding surfaces: A convex formulation”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 1054–1061.
- [SF11] M. Salzmann and P. Fua. “Linear Local Models for Monocular Reconstruction of Deformable Surfaces”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), pp. 931–944.
- [SF19] Yoli Shavit and Ron Ferens. *Introduction to Camera Pose Estimation with Deep Learning*. 2019.
- [SHF07] M. Salzmann, R. Hartley, and P. Fua. “Convex Optimization for Deformable Surface 3D Tracking”. In: *International Conference on Computer Vision (ICCV)*. 2007.
- [Sid+20] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. “Neural Dense Non-Rigid Structure from Motion with Latent Space Constraints”. In: *European Conference on Computer Vision (ECCV)*. 2020.
- [SM99] P. F. Sturm and S. J. Maybank. “On plane-based camera calibration: A general algorithm, singularities, applications”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 1999, 432–437 Vol. 1.
- [SMW06] Scott Schaefer, Travis McPhail, and Joe Warren. “Image Deformation Using Moving Least Squares”. In: *ACM Transactions on Graphics* 25.3 (2006), pp. 533–540.
- [SP06] Gerald Schweighofer and Axel Pinz. “Robust pose estimation from a planar target”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), pp. 2024–2030.
- [SP08] G. Schweighofer and A. Pinz. “Globally Optimal $O(n)$ Solution to the PnP Problem for General Camera Models”. In: *British Machine Vision Conference (BMVC)*. 2008.
- [SSP14] Torsten Sattler, Chris Sweeney, and Marc Pollefeys. “On Sampling Focal Length Values to Solve the Absolute Pose Problem”. In: *European Conference on Computer Vision (ECCV)*. Cham, 2014, pp. 828–843.
- [Ste18] Carsten Steger. “Algorithms for the Orthographic-n-Point Problem”. In: *Journal of Mathematical Imaging and Vision* 60.2 (2018), pp. 246–266.
- [Stu00] Peter Sturm. “Algorithms for Plane-Based Pose Estimation”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2000.
- [SUF08] M. Salzmann, R. Urtasun, and P. Fua. “Local Deformation Models for Monocular 3D Shape Recovery”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.
- [Sun+18] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018, pp. 8934–8943.
- [Tat+17] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. “CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6565–6574.

- [Tau91] Gabriel Taubin. “Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991), pp. 1115–1138.
- [TD20] Zachary Teed and Jia Deng. “DeepV2D: Video to Depth with Differentiable Structure from Motion”. In: 2020.
- [Ter+87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. “Elastically Deformable Models”. In: *SIGGRAPH Comput. Graph.* 21.4 (1987), pp. 205–214.
- [THB08] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. “Nonrigid Structure-from-Motion: Estimating Shape and Motion with Hierarchical Priors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.5 (2008), pp. 878–892.
- [TJK10] Jonathan Taylor, Allan D. Jepson, and Kiriakos N. Kutulakos. “Non-rigid structure from locally-rigid motion.” In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 2761–2768.
- [TK91] Carlo Tomasi and Takeo Kanade. *Detection and Tracking of Point Features*. 1991.
- [TK92] Carlo Tomasi and Takeo Kanade. “Shape and motion from image streams under orthography: a factorization method”. In: *International Journal of Computer Vision* 9 (1992), pp. 137–154.
- [Tra+12] Quoc-Huy Tran, Tat-Jun Chin, Gustavo Carneiro, Michael S. Brown, and David Suter. “In Defence of RANSAC for Outlier Rejection in Deformable Registration”. In: *European Conference on Computer Vision (ECCV)*. 2012.
- [Tri+00] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. “Bundle Adjustment – A Modern Synthesis”. In: *Vision Algorithms: Theory and Practice*. 2000, pp. 298–375.
- [Tri97] B. Triggs. “Autocalibration and the absolute quadric”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1997, pp. 609–614.
- [Tri98] Bill Triggs. “Autocalibration from planar scenes”. In: *European Conference on Computer Vision (ECCV)*. Berlin, Heidelberg, 1998, pp. 89–105.
- [Tri99a] B. Triggs. “Camera pose and calibration from 4 or 5 known 3D points”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 1. 1999, 278–284 vol.1.
- [Tri99b] Bill Triggs. “Camera Pose and Calibration from 4 or 5 known 3D Points”. In: *International Conference on Computer Vision*. 1999.
- [Tsa87] R. Tsai. “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses”. In: *IEEE Journal on Robotics and Automation* 3.4 (1987), pp. 323–344.
- [TSF18] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. “Real-Time Seamless Single Shot 6D Object Pose Prediction”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 292–301.
- [TT19] Chengzhou Tang and P. Tan. “BA-Net: Dense Bundle Adjustment Networks”. In: 2019.
- [Ull79] Shimon Ullman. *The Interpretation of Visual Motion*. The MIT press, 1979.

- [VA12] S. Vicente and L. Agapito. “Soft Inextensibility Constraints for Template-Free Non-rigid Reconstruction”. In: *European Conference on Computer Vision (ECCV)*. 2012.
- [VA13] S. Vicente and L. Agapito. “Balloon Shapes: Reconstructing and Deforming Objects with Volume from Images”. In: *International Conference on 3D Vision*. 2013.
- [Var+09a] A. Varol, M. Salzmann, E. Tola, and P. Fua. “Template-Free Monocular Reconstruction of Deformable Surfaces”. In: *International Conference on Computer Vision (ICCV)*. 2009.
- [Var+09b] Aydin Varol, Mathieu Salzmann, Engin Tola, and Pascal Fua. “Template-Free Monocular Reconstruction of Deformable Surfaces”. In: *International Conference on Computer Vision (ICCV)*. 2009.
- [Var+12] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. “A constrained latent variable model”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [Váv+17] Petr Vávra, Jan Roman, P. Zonča, Peter Ilnát, M. Němec, Kumar Jayant, Nagy Habib, and Ahmed El-Gendi. “Recent Development of Augmented Reality in Surgery: A Review”. In: *Journal of Healthcare Engineering 2017 (2017)*, pp. 1–9.
- [VF] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*.
- [Wan+16] Xuan Wang, Mathieu Salzmann, Fei Wang, and Jizhong Zhao. “Template-Free 3D Reconstruction of Poorly-Textured Nonrigid Surfaces”. In: *European Conference on Computer Vision (ECCV)*. 2016.
- [Wan+19] J. Wang, S. Song, H. Ren, C. M. Lim, and M. Q. - Meng. “Surgical Instrument Tracking By Multiple Monocular Modules and a Sensor Fusion Approach”. In: *IEEE Transactions on Automation Science and Engineering* 16.2 (2019), pp. 629–639.
- [Wan+21] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. “Deep 3D human pose estimation: A review”. In: *Computer Vision and Image Understanding* 210 (2021), p. 103225.
- [WFG15] X. Wang, D. F. Fouhey, and A. Gupta. “Designing deep networks for surface normal estimation”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 539–547.
- [WH05] Yihong Wu and Zhanyi Hu. “PnP Problem Revisited”. In: *Journal of Mathematical Imaging and Vision* 24 (2005), pp. 131–141.
- [Wie+18] Folker Wientapper, Michael Schmitt, Matthieu Fraissinet-Tachet, and Arjan Kuijper. “A universal, closed-form approach for absolute pose problems”. In: *Computer Vision and Image Understanding* 173 (2018), pp. 57–75.
- [WMH17] Jian Wu, Liwei Ma, and Xiaolin Hu. “Delving deeper into convolutional neural networks for camera relocalization”. In: *International Conference on Robotics and Automation*. 2017, pp. 5644–5651.
- [Wu15] Changchang Wu. “P3.5P: Pose Estimation With Unknown Focal Length”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [Xia+18] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes”. In: *Robotics: Science and Systems (RSS)*. 2018.

- [Yi+16] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. “LIFT: Learned Invariant Feature Transform”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 467–483.
- [Yin+17] Xiaochuan Yin, Xiangwei Wang, Xiaoguo Du, and Qijun Chen. “Scale Recovery for Monocular Visual Odometry Using Depth Estimated with Deep Convolutional Neural Fields”. In: *International Conference on Computer Vision (ICCV)*. 2017, pp. 5871–5879.
- [YS18] Zhichao Yin and Jianping Shi. “GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- [Yu+15] R. Yu, C. Russell, N. D. F. Campbell, and L. Agapito. “Direct, Dense, and Deformable: Template-Based Non-rigid 3D Reconstruction from RGB Video”. In: *International Conference on Computer Vision (ICCV)*. 2015, pp. 918–926.
- [ZC21] Bingbing Zhuang and Manmohan Chandraker. “Fusing the Old with the New: Learning Relative Camera Pose with Geometry-Guided Uncertainty”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.
- [ZH95] Zhongfei Zhang and A.R. Hanson. “Scaled Euclidean 3D reconstruction based on externally uncalibrated cameras”. In: *Proceedings of International Symposium on Computer Vision - ISCV*. 1995, pp. 37–42.
- [ZH96] Zhongfei Zhang and Allen R. Hanson. “3D Reconstruction Based on Homography Mapping”. In: *In ARPA Image Understanding Workshop*. 1996, pp. 0249–6399.
- [Zha+17] Lin Zhang, Menglong Ye, Po-Ling Chan, and Guang-Zhong Yang. “Real-time surgical tool tracking and pose estimation using a hybrid cylindrical marker”. en. In: *International Journal of Computer Assisted Radiology and Surgery* 12.6 (2017), pp. 921–930.
- [Zha+20] Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. “Visual Odometry Revisited: What Should Be Learnt?”. In: *International Conference on Robotics and Automation*. 2020, pp. 4203–4210.
- [Zha00] Zhengyou Zhang. “A Flexible New Technique for Camera Calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), pp. 1330–1334.
- [Zha16] Zhengyou Zhang. “Camera Calibration: a Personal Retrospective”. In: *Machine Vision and Applications* 27.7 (2016), pp. 963–965.
- [Zhe+13] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Åström, and Masatoshi Okutomi. “Revisiting the PnP Problem: A Fast, General and Optimal Solution”. In: *International Conference on Computer Vision (ICCV)*. 2013, pp. 2344–2351.
- [Zhe+14] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi. “A General and Simple Method for Camera Pose and Focal Length Determination”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 430–437.
- [Zho+17a] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. “Unsupervised Learning of Stereo Matching”. In: *International Conference on Computer Vision (ICCV)*. 2017.
- [Zho+17b] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. “Unsupervised Learning of Depth and Ego-Motion from Video”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6612–6619.

- [Zho+20] Kun Zhou, Xiangxi Meng, Bo Cheng, and Cornelio Yáñez-Márquez. “Review of Stereo Matching Algorithms Based on Deep Learning”. In: *Intell. Neuroscience 2020* (2020).
- [ZI02] L. Zeinik-Manor and M. Irani. “Multiview constraints on homographies”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (2002), pp. 214–223.
- [ZK16] Yinqiang Zheng and Laurent Kneip. “A Direct Least-Squares Solution to the PnP Problem With Unknown Focal Length”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [ZL15] Jure Zbontar and Yann LeCun. “Computing the stereo matching cost with a convolutional neural network”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1592–1599.
- [ZSI19] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. “DPOD: 6D Pose Object Detector and Refiner”. In: *International Conference on Computer Vision (ICCV)*. 2019, pp. 1941–1950.