THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Towards Reliable and Accurate Global Structure-from-Motion

JOSÉ PEDRO IGLESIAS

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Towards Reliable and Accurate Global Structure-from-Motion

JOSÉ PEDRO IGLESIAS
ISBN 978-91-7905-863-0

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5329
ISSN 0346-718X

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000

Printed by Chalmers Reproservice
Gothenburg, Sweden, May 2023

To my father.

Abstract

Reconstruction of objects or scenes from sparse point detections across multiple views is one of the most tackled problems in computer vision. Given the coordinates of 2D points tracked in multiple images, the problem consists of estimating the corresponding 3D points and cameras' calibrations (intrinsic and pose), and can be solved by minimizing reprojection errors using bundle adjustment. However, given bundle adjustment's nonlinear objective function and iterative nature, a good starting guess is required to converge to global minima.

Global and Incremental Structure-from-Motion methods appear as ways to provide good initializations to bundle adjustment, each with different properties. While Global Structure-from-Motion has been shown to result in more accurate reconstructions compared to Incremental Structure-from-Motion, the latter has better scalability by starting with a small subset of images and sequentially adding new views, allowing reconstruction of sequences with millions of images. Additionally, both Global and Incremental Structure-from-Motion methods rely on accurate models of the scene or object, and under noisy conditions or high model uncertainty might result in poor initializations for bundle adjustment. Recently pOSE, a class of matrix factorization methods, has been proposed as an alternative to conventional Global SfM methods. These methods use VarPro - a second-order optimization method - to minimize a linear combination of an approximation of reprojection errors and a regularization term based on an affine camera model, and have been shown to converge to global minima with a high rate even when starting from random camera calibration estimations.

This thesis aims at improving the reliability and accuracy of global SfM through different approaches. First, by studying conditions for global optimality of point set registration, a point cloud averaging method that can be used when (incomplete) 3D point clouds of the same scene in different coordinate systems are available. Second, by extending pOSE methods to different Structure-from-Motion problem instances, such as Non-Rigid SfM or radial distortion invariant SfM. Third and finally, by replacing the regularization term of pOSE methods with an exponential regularization on the projective depth of the 3D point estimations, resulting in a loss that achieves reconstructions with accuracy close to bundle adjustment.

Keywords: Structure-from-Motion, 3D reconstruction, camera calibration, bundle adjustment, global SfM, non-rigid SfM, radial distortion, matrix factorization, pOSE, point set registration.

List of Publications

This thesis is based on the following publications:

[A] **José Pedro Iglesias**, Carl Olsson, Fredrik Kahl, "Point Set Registration and Global Optimality". Conference on Computer Vision and Pattern Recognition 2020.

[B] **José Pedro Iglesias**, Carl Olsson, "Radial Distortion Invariant Factorization for Structure from Motion". International Conference on Computer Vision 2021.

[C] **José Pedro Iglesias**, Amanda Nilsson, Carl Olsson, "expOSE: Accurate Initialization Free Projective Factorization using Exponential Regularization". Conference on Computer Vision and Pattern Recognition 2023.

[D] **José Pedro Iglesias**, Carl Olsson, Marcus Valtonen Örnhag, "Accurate Optimization of Weighted Nuclear Norm for Non-Rigid Structure from Motion". European Conference on Computer Vision 2020.

[E] Marcus Valtonen Örnhag, **José Pedro Iglesias**, Carl Olsson, "Bilinear Parameterization for Non-Separable Singular Value Penalties". Conference on Computer Vision and Pattern Recognition 2021.

Other publications by the author, not included in this thesis, are:

[F] Lucas Brynte, Viktor Larsson, **José Pedro Iglesias**, Carl Olsson, Fredrik Kahl, "On the tightness of semidefinite relaxations for rotation estimation". Journal of Mathematical Imaging and Vision, 2022.

[G] Pedro Miraldo, **José Pedro Iglesias**, "A Unified Model for Line Projections in Catadioptric Cameras With Rotationally Symmetric Mirrors". Conference on Computer Vision and Pattern Recognition, 2022.

Acknowledgments

What amazing 5 years these have been! I'd like to start by thanking Carl Olsson, my supervisor, for giving me this opportunity at the beginning of 2018. I've learned immensely from you and none of this would have been possible without your guidance. I'd equally like to thank Fredrik Kahl for being my co-supervisor and leading a research group filled with brilliant and talented people. A special thanks to the people that I've collaborated with during my studies, Marcus, Amanda, the Eye-Tracking team at Meta Research Labs, and especially Pedro Miraldo, who introduced me to the research world about eight years ago and has been a mentor and a friend ever since.

To the professors, researchers and postdocs in the group, Christopher, Torsten, Ida, Che-Tsung, Axel, James, Huu, Erik, David, and to my former and current PhD student colleagues, Carl, Måns, Jennifer, Lucas, Kunal, Xixi, Georg, Yaroslava, Rasmus, Roman, Dorian, Josef, Victor, you are some of the most intelligent, passionate and talented people I've ever met and it's truly humbling to talk and learn from you week after week. I will never forget the deep conversation about the most random topics during our fika breaks, you make the department a happier place to be at. I wish you all the best in your studies and careers, and can't wait to see all the great things the future will bring you.

On a personal level, a big shout-out and thanks to all my friends, in Sweden, Portugal, and all over the world. I feel incredibly fortunate that you are just too many to name here, but I hope I demonstrate every day how important every single of you is in my life. No reasonable amount of words will ever show how essential you were in all the ups and downs of this period of my life. Love you all.

Finally, to my family in Portugal, to my dear grandma, and in particular to my dad - I know how excited you were about this chapter of my life and I wish you could be here now, miss you so much. To my cat Thor, and to my mom, who I'll address in portuguese. Obrigado pela educação e amor que me deste até hoje, nada disto teria sido possível sem o teu esforço e dedicação. Estarei eternamente grato e para sempre ao teu lado.

José Pedro Iglesias
Göteborg, May 2023

Acronyms

SfM:	Strucutre-from-Motion
BA:	Bundle Adjustment
LM:	Levenberg-Marquardt
SDP:	Semidefinite Program
SVD:	Singular Value Decomposition
DLT:	Direct Linear Transform
RANSAC:	Random Sample Consensus
SIFT:	Scale Invariant Feature Transform
SURF:	Speeded Up Robust Features
FAST:	Features from Accelerated Segment Test
OSE:	Object Space Error
ROSE:	Radial Object Space Error
pOSE:	pseudo Object Space Error
RpOSE:	Radial pseudo Object Space Error
NN:	Nuclear Norm
WNN:	Weighted Nuclear Norm

Contents

Abstract i

List of Papers iii

Acknowledgements v

Acronyms vi

I Overview 1

1 Introduction 3

1.1 Thesis outline 7

1.2 Notation 8

2 Background 9

2.1 Projective Geometry 9

Points in 2D and 3D 9

Transformations in 2D and 3D 10

2.2 Camera models 12

Pinhole camera model 12

Affine camera model 14

Distortion models 15
1D Radial camera model 17
2.3 Feature Extraction and Matching 18

3 Structure-from-Motion 21
3.1 Multiview Geometry and SfM problems 21
Reconstruction Ambiguity and Degrees of Freedom 22
Triangulation and Camera resectioning 23
3.2 Bundle Adjustment 24
3.3 Global and Incremental SfM Pipeline 25
Global Structure-from-Motion 25
Minimal Solvers and RANSAC 26
Pose averaging 27
Incremental Structure-from-Motion 28
3.4 Factorization-based SfM 29
Affine Factorization 30
Projective Factorization 31
pOSE models 31
3.5 Non-rigid Structure-from-Motion 33
Non-rigid Factorization 33
Low-rank Penalty Functions 35

4 Summary of included papers 39
4.1 Paper A - “Point Set Registration and Global Optimality” 40
4.2 Paper B - “Radial Distortion Invariant Factorization for Structure-from-Motion” 41
4.3 Paper C - “expOSE: Accurate Initialization Free Projective Factorization using Exponential Regularization” 42
4.4 Paper D - “Accurate Optimization of Weighted Nuclear Norm for Non-Rigid Structure-from-Motion” 44
4.5 Paper E - “Bilinear Parameterization for Non-Separable Singular Value Penalties” 45

5 Concluding Remarks and Future Work 47
5.1 Conclusions 47
5.2 Future work 48
Dealing with outliers 48

Solvability and Global Optimality 50
Incorporating learning-based methods 50

References **53**

II Papers **63**

A Point Set Registration and Global Optimality **A1**
1 Introduction A3
1.1 Notation A5
2 Problem Statement A5
3 Duality and Optimality Conditions A7
3.1 Relaxation of the Primal Problem A8
3.2 Sufficient Global Optimality Conditions A8
4 Spectral Analysis A9
4.1 Decoupling into Noise-free and Noisy Data A9
4.2 Spectral Analysis of the Decoupled Data A10
4.3 Interpretation and Evaluation A11
5 An Alternative Spectral Analysis A13
5.1 Decoupling into Noise-free and Noisy Data A13
5.2 Spectral Analysis of the Decoupled Data A14
5.3 Interpretation and Evaluation A15
6 Experiments A16
6.1 Local Optimization A17
6.2 Evaluating Theorems 1, 2 and 3 with Real Data A17
7 Conclusions A18
References A20

B Radial Distortion Invariant Factorization for Structure from Motion **B1**
1 Introduction B3
2 The 1D-radial Camera Model B5
2.1 Distortion Models B5
2.2 The 1D Radial Camera Model B6
2.3 The ML-estimate B7
2.4 A pOSE model B8

2.5 Evaluation B11
3 Connection between pOSE and ML B14
3.1 Algorithm with Refinement B15
3.2 Evaluation B15
4 Full System Outline B17
4.1 Experiments B18
5 Conclusions B19
References B19

C expOSE: Accurate Initialization Free Projective Factorization using Exponential Regularization C1
1 Introduction C3
2 Reconstruction Objectives C5
2.1 Reprojection Error and Cheirality C5
2.2 The pOSE Model C7
2.3 Exponential Regularization C8
3 Optimization with VarPro C9
3.1 The expOSE Model C10
3.2 Performance evaluation of expOSE C12
4 Robustness to Radial Distortion C13
4.1 Decoupling Tangential and Radial Errors C13
4.2 Reweighting the Error Components C14
4.3 Regularization for radial distortion invariance C16
5 Outline of Full Reconstruction Pipeline C17
5.1 Experiments C18
6 Conclusions C21
References C21

D Accurate Optimization of Weighted Nuclear Norm for Non-Rigid Structure from Motion D1
1 Introduction D3
1.1 Related Work and Contributions D5
2 Bilinear Parameterization Penalties D7
2.1 Extreme Points and Optimality D8
3 Non-square Matrices D10
4 Linear Objectives - Weighted Nuclear Norms D11

5 Experiments D11
5.1 Pseudo Object Space Error (pOSE) and Non-Rigid Structure from Motion D12
5.2 Low-Rank Matrix Recovery with pOSE errors D13
5.3 Non-Rigid Structure Recovery D14
6 Conclusions D18
References D19

E Bilinear Parameterization for Non-Separable Singular Value Penalties E1
1 Introduction E3
1.1 Framework and Contributions E5
1.2 Related Work E6
2 Overview of the Approach E8
2.1 A Continuous Relaxation E8
2.2 Bilinear Parameterization E9
2.3 Quadratic Approximation and Optimization E10
2.4 Overview of Algorithm E12
3 Main Theoretical Result E13
3.1 Theoretical Background E13
3.2 Proof of Theorem 1 . . . . E14
4 Experiments E17
4.1 Matrix Recovery E17
4.2 Application to Non-Rigid SfM E18
5 Conclusions E20
References E20

Part I

Overview

CHAPTER 1

Introduction

Vision is an essential part of the way that we interpret, experience, and navigate the world. More than any other sense, vision allows us to detect and locate friends or foes in our proximity, understand our position with respect to reference points, estimate distances between or to objects, etc.. All this data is captured by our eyes in raw form, which is then (somehow) processed by our brains into useful information. Given the richness of vision data, how can we replicate this process in our systems, such as robots, autonomous cars, or computer applications? Well, the data-capturing part is quite well solved I would argue. We have camera sensors that very closely replicate the optical processes in the eye, resulting in images that have as much information as we can see (perhaps with even higher resolution). The problem is that it's unclear what to do with this information since the inner workings of the brain are still pretty unknown to us. Besides that, camera sensors provide us with digitalized pixel data, which are very different from the analogical signals our brain processes.

While our knowledge about neuroscience and the brain keeps growing and new theories emerge, the research community tackled the problem using the tools they currently master: mathematics and geometry. Computer vision has been a popular research area since the middle of the last century. From camera models that capture how 3D points are projected into images, to multiple view tensors that determine algebraic relations between projections in different views, concepts from linear algebra, calculus, matrix and spectral analysis, polynomial algebra, optimization, and many others have been the preferred tools to solve computer vision problems. More recently, a new class of solutions emerged with the rise of deep machine learning and parallel computing. Instead of explicitly defining the functions and heuristics that determine the relations between camera, scene, and images, deep machine learning methods learn those relations through neural networks with learnable parameters from (huge amounts of) training data. These learning-based methods have been extremely successful, in particular in areas like scene understanding, object detection, image segmentation or classification, and more recently image generation, strongly outperforming conventional methods based on heuristics. However, in regard to problems with strong geometric primitives, like 3D reconstruction or pose estimation, learning-based methods are still not able to achieve the accuracy of conventional methods. The advantage of conventional methods is that they use exact relations between variables, and approximating these with neural networks results in degradation of the obtained reconstruction - at least for now.

Even though learning-based methods are definitely promising and are slowly catching up, I still believe that a lot more can be squeezed in terms of reliability and performance from conventional methods without the need to learn from large datasets. Therefore, the focus of this thesis is on 3D reconstruction and camera pose estimation, in which geometrical relations between cameras and the scene structure play an essential part. In particular, I study the Structure-from-Motion (SfM) problem, which estimates camera calibrations (both intrinsic calibration and camera poses) and 3D point coordinates from a set of 2D points tracked along multiple images, captured from different points of view (see Figure 1.1).

Structure-from-Motion has several applications which have been under society's spotlight over the last few years. One of the biggest ones is perhaps autonomous driving, in which images collected from multiple cameras placed in the vehicle can be used to localize and/or track its position. The vehicle's pose estimation from images can then be combined with other sensor data (e.g. GPS, IMU, Lidar) to increase the accuracy of the estimation. More generally, the problem of estimating pose based on images is referred to as visual localization and can also include other applications like robot navigation or augmented reality. Augmented reality headsets usually have cameras mounted on the front and sides of the device which are used for scene understanding and pose estimation. Having an accurate headset pose estimation is essential to generate virtual objects over the user's field of view in a smooth

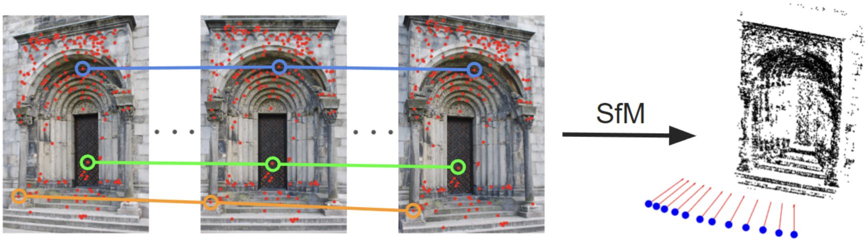

Figure 1.1: Representation of the inputs and outputs of a Structure-from-Motion (SfM) algorithm. (Left) Multiple images are captured from different viewpoints of a certain object or scene, in which keypoints (in red) are detected and matched along the images (some of the matches are represented in blue, green and yellow). (Right) These matches, or correspondences, are then fed as input to a Structure-from-Motion algorithm, which ultimately estimates the 3D coordinates of the detected keypoints (in black), along with the camera calibrations position (in blue), rotation (red arrows) and internal calibration.

and realistic way. Failing to do so makes those objects jitter or drift, resulting in a poor user experience and possibly even nausea. Structure-from-Motion methods are also frequently used for 3D scanning of objects or scenes. Typically 3D scanners are laser-based which makes the device expensive and not particularly accessible. Contrarily, camera sensors are fairly cheap and when combined with Structure-from-Motion methods provide an affordable alternative to obtain 3D scans - you can easily get a 3D model of an object of interest with your phone, for instance. Examples of such use cases include real estate, where virtual tours can be set up by 3D scanning a house or apartment and making its 3D model available online for interested buyers to explore remotely. More recently, Structure-from-Motion methods have also been used to provide training data for deep learning methods. For instance, novel view synthesis have been a hot topic in computer vision since the release of the NeRF paper in 2020. Novel view synthesis takes training data images of an object or scene from several viewpoints along with the corresponding camera calibrations and learns a neural network model that is able to accurately generate a new image from a new and unseen viewpoint. In order to learn models that are able to generate realistic images, the input camera calibrations, both camera poses and internal calibration, need to be extremely accurate and Structure-from-Motion methods are usually the preferred way to estimate them if ground-truth calibrations are not available.

One possible way to solve Structure-from-Motion is through the minimization of the so-called reprojection errors. The reprojection error consists of the distance between the pixel coordinates of the 2D point detected in the image and the 2D point projected back to the image based on the estimated camera calibration and 3D point coordinates. If the reprojection error is zero, it means that the estimated model perfectly fits the measurements and consequently the reconstruction obtained is accurate. Obtaining zero reprojection error is not realistic since there is always noise in the input data (e.g. pixel discretization, uncertainty in the exact location of the 2D points in the images) and as so the problem, denoted bundle adjustment, is formulated as a minimization of reprojection errors, such that the estimated reconstruction fits as well as possible the multiple 2D points detected over all the images. Bundle adjustment does not have a closed-form solution so it relies on an iterative optimization from a starting guess for the camera calibrations and 3D points. In fact, given the nonlinearity of the reprojection errors, the starting guess needs to be sufficiently close to the desired (unknown) camera calibrations and 3D points, otherwise, the algorithm can converge to local minima. In practice, this means that some other method is needed to obtain a good enough estimation to initiate bundle adjustment, and this is where Structure-from-Motion methods come into the picture.

Structure-from-Motion methods are typically divided into two categories: Global and Incremental SfM. Global methods start by computing relative poses between pairs of images with an intersecting field of view - for instance, relative pose between images 1 and 2, 2 and 3, and so on, for all possible pairs. From the pairwise pose estimations, a global pose averaging method is applied to estimate absolute camera poses on some (sometimes arbitrary) global coordinate system. At this stage, the camera calibrations are estimated, so the 3D points can be obtained through triangulation. We will look more into the details of these steps in Section 3. The camera calibrations from the global pose averaging and 3D point from triangulation are then used as initialization to bundle adjustment. Incremental methods differ by starting with a subset of the images, typically just a pair, for which relative poses and 3D points are estimated in a similar way. After that, a new view with intersecting field of view is added, and the corresponding and already estimated 3D points are used to estimate the camera calibration of the new view. New 3D points can now be triangulated using the estimated camera calibrations. This process is repeated until all images are added. Incremental methods have the advantage to scale much better than Global methods, since estimating new camera calibrations for given 3D points and vice-versa can be solved very efficiently compared to global pose averaging. However, since only a subset of views is used sequentially, it tends to result in less accurate reconstructions.

An alternative family of Global SfM consists of factorization methods. These methods build on the fact that based on the camera model equations the 2D point detections in all views can be rearranged into a large matrix with low rank, which can be decomposed into the product of two matrices with four columns each - one corresponding to the camera calibrations and another to the (transposed) 3D points. More recently, Hong *et al.* [1] proposed a new formulation of factorization methods, pOSE models, that aims at minimizing a linear approximation of reprojection errors plus a regularization term. The pOSE problem can be efficiently optimized with VarPro, a variable projection second-order optimization algorithm, that can converge to global minima with a high success rate even when starting with random guesses for camera calibrations. Besides that, it also results in good initializations for bundle adjustment, something that was not always the case in practice with previous factorization methods for Structure-from-Motion.

This thesis aims at improving the understanding and quality of Global Structure-from-Motion methods. First by analyzing global optimality conditions of one variant of global pose averaging methods, called point set registration. In point set registration, several 3D point clouds of the same object or scene are available in each camera's local coordinate system, and the goal is to estimate an average 3D point cloud in a global coordinate system, along with the transformations that map from the local coordinate systems to the global one. In particular, we study under which conditions, such as missing data, object spatial distribution, or noise, a candidate solution is globally optimal. Second, by extending pOSE methods to different problem instances, such as images with radial distortion (radial distortion invariant SfM) and images in which the object or scene suffers temporal deformations (Non-Rigid SfM). Additionally, we also explore the use of a different regularization term that penalizes the reconstructions less than the one proposed in the original pOSE model, ultimately leading to more accurate reconstructions.

1.1 Thesis outline

The thesis is divided into two parts. Part II contains the included five papers that consist of the core of this thesis and its contributions. Part I provides the necessary background in order to be able to more easily follow the content of the papers, and is itself divided into five Chapters. Chapter 2 provides some basic knowledge regarding projective geometry and camera models, which should be enough for the reader to have a clear idea on how cameras and 3D points are represented and related in geometric computer vision. Chapter 2 finishes with an overview of feature detection and matching, which combined with camera models and 3D points provides the groundwork for understanding the Structure-from-Motion problem. Chapter 3 goes into more detail regarding Structure-from-Motion. In particular, how can the problem be formulated from the concepts introduced in Chapter 2, and what are the main approaches to tackle the problem, including their properties, advantages and disadvantages. Chapter 4 provides a summary of the five papers in Part II and relates them to the concepts in Chapter 2 and 3 of Part I. Finally, Chapter 5 concludes the work with a summary of the contributions and results of this thesis, and follows up with possible future work, both in terms of extensions of the presented papers and also what I believe could be new interesting approaches based on recent developments within the research community.

1.2 Notation

In this section, I will explain the notation used in Part I of the thesis. Scalar variables are represented as lower or upper case letters e.g. x, y, λ, N, F. N-dimensional vectors are represented as lowercase bold letters, e.g. $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Matrices are represented using upper case bold letters, e.g. $\mathbf{X}, \mathbf{P}, \mathbf{H}$. Sometimes single 3D points, which are vectors, are represented with uppercase letters as well, e.g. $\mathbf{X}$, but in those cases, the use of indexes like $\mathbf{X}_i$ or $\mathbf{X}_{ij}$ or the context of the text should make it clear if it consists of a vector or a matrix. When referring to only specific rows of a vector or matrix, a superscript $(i : j)$ is used to select rows i to j of the vector/matrix, e.g. $\mathbf{R}^{(1:2)}$ refers to the first two columns of the matrix $\mathbf{R}$, and $\mathbf{z}^{(3)}$ to the third row of the vector $\mathbf{z}$. For representing points in homogeneous coordinates a bar is added on top of the bold letter e.g. $\bar{\mathbf{x}}, \bar{\mathbf{X}}$. The vector of ordered singular values of a matrix $\mathbf{A}$ is referred to as $\sigma(\mathbf{A})$, with $\sigma_1(\mathbf{A}) \geq \sigma_2(\mathbf{A}) \geq \ldots$, where $\sigma_k(\mathbf{A})$ corresponds to the kth largest singular value of $\mathbf{A}$.

I conclude this section by noting that the notation used in the five included papers in Part II might be different from the one defined here and used in all of Part I. However, the context of each paper should make it easy to follow the notation nonetheless.

CHAPTER 2

Background

In this chapter, we will provide the necessary background information to better understand the content of the five papers presented in the second part of the thesis. In the next chapter, I will use these concepts to introduce the Structure-from-Motion problem and the different ways to solve it. The chapter is divided into three parts. In the first section, I will cover the basics of projective geometry, in particular how 2D and 3D points are represented, and what type of transformations can be applied to them. In the second section, I will go through the major camera models used in computer vision, and how effects like lens distortion can be modeled. Finally, I will take about some of the methods used to detected and track points along multiple views, something essential for 3D reconstruction and Structure-from-Motion.

2.1 Projective Geometry

Points in 2D and 3D

Let us start by considering a 2D point $\mathbf{x}$ described by the coordinates $(x, y) \in \mathbb{R}^2$. In many occasions I will represent such point using its homogeneous coordinate representation [2], i.e. $\bar{\mathbf{x}} = (x, y, 1) \in \mathbb{P}^2$ (projective space). Using homogeneous coordinates allows us to define points up to scale, meaning that $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ represent the same point in 2D if $\bar{\mathbf{x}} = \lambda\bar{\mathbf{x}}', \lambda \neq 0$. For instance, let's consider the 2D point $\mathbf{x} = (1, 3)$. It can be represented in homogeneous coordinates as $\bar{\mathbf{x}} = (1, 3, 1)$, and any point $\bar{\mathbf{x}}' = (\lambda, 3\lambda, \lambda)$ represents the exact same point in 2D since (x, y) can be recovered by dividing the first two elements of $\bar{\mathbf{x}}'$ by its third element. This representation also allows us to represent points at infinity [3] by setting the last coordinate to zero, e.g. $\bar{\mathbf{x}} = (2, 1, 0)$, a concept useful in many applications in computer vision such as camera calibration. A 2D point $\bar{\mathbf{x}} \in \mathbb{P}^2$ has 2 degrees of freedom (3 variables minus the scale ambiguity).

Similarly, a 3D point $\mathbf{X}$ is described using three coordinates $(x, y, z) \in \mathbb{R}^3$, and its homogeneous coordinates extension corresponds to $\bar{\mathbf{X}} = (x, y, z, 1) \in \mathbb{P}^3$. Just like in the 2D case, points are defined up to scale and $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}'$ define the same 3D point if $\bar{\mathbf{X}} = \lambda\bar{\mathbf{X}}', \lambda \neq 0$. This representation will be used throughout this thesis in order to represent the points in 3D that we would like to reconstruct. A 3D point $\bar{\mathbf{X}} \in \mathbb{P}^3$ has 3 degrees of freedom (4 variables minus the scale ambiguity).

Transformations in 2D and 3D

Having defined how to represent a point in 2D and 3D in projective geometry, we can now introduce the concept of transformation from $\mathbb{P}^n$ to $\mathbb{P}^n$ that can be applied to a set of points in this space. Starting with 2D, we define projective transformation or homography as

$$\bar{\mathbf{x}}' = \mathbf{H}\bar{\mathbf{x}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \bar{\mathbf{x}}. \tag{2.1}$$

For generic homographies, the only constraint on the elements h_{ij} is that $\mathbf{H}$ has to be invertible. A homography has 8 degrees of freedom (9 parameters minus scale ambiguity), and it preserves some properties of a set of points, such as colinearity [3]. Homographies are very commonly used to model transformations between planar structures. For instance, consider that we have two images of a chessboard captured from different views. The relation between all the corners of the chessboard in both images (say $\mathbf{x}_i$ and $\mathbf{x}'_i$ in images 1 and 2, respectively, for $i = 1, \ldots, N$, where N is the number of points) can be represented by a homography, i.e., $\lambda_i\bar{\mathbf{x}}'_i = \mathbf{H}\bar{\mathbf{x}}_i, \forall i = 1, \ldots, N$. This is only true since both images and the chessboard itself consist of planar surfaces.

Adding constraints to the possible values of $\mathbf{H}$ will not only reduce the degrees of freedom of the transformation but also increase the set of properties preserved by it.

For instance, an affine transformation in 2D is defined as

$$\bar{\mathbf{x}}' = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & h_{33} \end{bmatrix} \bar{\mathbf{x}} \tag{2.2}$$

where $h_{31} = h_{32} = 0$, meaning that an affine transformation has 6 degrees of freedom. In this case, besides colinearity, properties like parallelism, the ratio of areas, or centroids are equally preserved. The problem can be further constrained by considering a similarity transformation, defined as

$$\bar{\mathbf{x}}' = \begin{bmatrix} sr_{11} & sr_{12} & h_{13} \\ sr_{21} & sr_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{x}} \tag{2.3}$$

where the top-left 2×2 matrix of $\mathbf{H}$ consists of a scaled rotation $s\mathbf{R}, \mathbf{R} \in \mathcal{SO}(2)$ and $s \in \mathbb{R}$. This class of transformations additionally preserves the ratio between lengths and areas, and has 4 degrees of freedom (1 for the rotation, 1 for the scale s, and two for h_{13} and h_{23}). For the particular case $s = 1$, the transformation is denoted as Euclidean, and it additionally preserves areas and lengths. A Euclidean transformation consists of rotating and translating points in 2D, and it has 3 degrees of freedom.

The set of 3D transformations follows a similar hierarchy, with the difference that $\mathbf{H}$ is instead a 4×4 matrix, all the properties concerning lines are extended to planes, and 2D rotations in the 2D transformations are replaced by 3D rotations. A projective transformation is then defined as

$$\bar{\mathbf{X}}' = \mathbf{H}\bar{\mathbf{X}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \bar{\mathbf{X}} \tag{2.4}$$

which has 15 degrees of freedom and preserves coplanarity. The projective transformation is a core concept in Structure-from-Motion since when the intrinsic calibration of the camera is unknown, the reconstruction is estimated up to a projective transformation (this will be covered in more detail in the next Chapter). An affine transformation consists of the case in which $h_{41} = h_{42} = h_{43} = 0$, it has 12 degrees of freedom and it preserves parallelism between planes. A similarity transformation in 3D is defined as

$$\bar{\mathbf{X}}' = \begin{bmatrix} sr_{11} & sr_{12} & sr_{13} & h_{14} \\ sr_{21} & sr_{22} & sr_{23} & h_{24} \\ sr_{31} & sr_{32} & sr_{33} & h_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{X}}, \tag{2.5}$$

where the top-left 3×3 matrix of $\mathbf{H}$ consists of a scaled rotation $s\mathbf{R}, \mathbf{R} \in \mathcal{SO}(3)$ and $s \neq 0$. It has 7 degrees of freedom and additionally preserves ratios between volumes. When $s = 1$, it is also denoted as Euclidean transformation in 3D, it has 6 degrees of freedom and it consists of rotating and translating points in 3D, hence also preserving volumes.

2.2 Camera models

One of the most significant components of multiple view geometry is the camera model. In general terms, the camera model represents the mapping of 3D points of an object or scene to points in the 2D image collected with a camera sensor, i.e., a mapping from $\mathbb{P}^3$ to $\mathbb{P}^2$. There are many different ways to model this mapping, each of them corresponding to a different camera model valid under certain assumptions. In this section, I will cover some of the most common ones, such as pinhole and affine camera models. In the end, I'll also briefly explain how concepts like lens distortion can be added to such models.

Pinhole camera model

The most common way to model camera projections is by using the so-called pinhole camera model. This model takes an ideal pinhole camera assumption in which the camera aperture consists of a single point, the camera center, and for which there is no lens refraction in play, i.e., viewing rays go directly from the camera center to the 3D point (see Figure 2.1). The 2D point in the image corresponds to the intersection between the viewing ray and the image plane.

The model can be mathematically represented through the following mapping between the 3D point $\mathbf{X}_j$ and the 2D image point $\mathbf{x}_j$

$$\lambda_j \bar{\mathbf{x}}_j = \mathbf{P}\bar{\mathbf{X}}_j \tag{2.6}$$

where $\lambda_j \in \mathbb{R}$ is usually unknown and referred to as projective depth, and $\mathbf{P}$ is a 3×4 camera matrix representing the intrinsic and extrinsic camera calibration. For

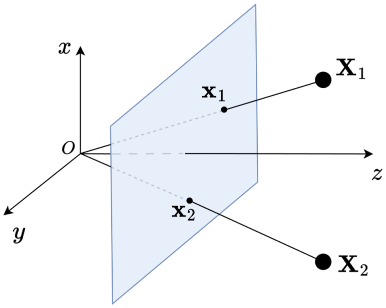

Figure 2.1: Visualization of the pinhole (perspective) camera model. The camera center is located at the origin O and the axis z determines its viewing direction or optical axis. The image plane (in light blue) is orthogonal to the optical axis and is located at a distance f (focal length) of the camera center. The ray that passes through the camera center and the 3D points is referred to as viewing ray, and its intersection with the image plane determines the 2D coordinates of the 3D point in the image. Note that any 3D point along the same viewing ray will have identical 2D coordinates.

any given pair of corresponding 2D point $\mathbf{x}_j$ and 3D point $\mathbf{X}_j$, the camera matrix is structured as $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$. The rotation $\mathbf{R}$ and the translation $\mathbf{t}$ encode the Euclidean transformation from the world coordinate frame to the camera coordinate frame (extrinsic), and $\mathbf{K}$ is the intrinsics calibration matrix

$$\mathbf{K} = \begin{bmatrix} \gamma f & s & x_c \\ 0 & f & y_c \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

that maps points to the image plane. When $\mathbf{K}$ is known, we say that the camera is calibrated. This mapping has the following parameters: principal point (x_c, y_c) that translates from the origin of the image plane to the center of the image; aspect ratio γ which controls the ratio between the height and width of a pixel (for $\gamma = 1$ we get squared pixels); skew s which determines the skewness of the pixel axes; and focal length f which rescales from cartesian units in the image plane to pixel units. In modern cameras it's common to assume that $\gamma = 1$ and $s = 0$, resulting in a simpler model. A particular case of pinhole cameras in which $\gamma = 1$, $s = 0$ and $(x_c, y_c) = (0, 0)$ are denoted as perspective camera model.

Having defined a camera model, we can use a geometric metric to measure how well the model fits the data. The most commonly used metric in computer vision is denoted as reprojection error, which compares the reprojected 3D point according to the assumed camera model with the 2D image point observed. For the pinhole camera model, the reprojection error comes down to

$$r = \left\| \mathbf{x} - \frac{1}{\mathbf{P}^{(3)}\bar{\mathbf{X}}} \mathbf{P}^{(1:2)} \bar{\mathbf{X}} \right\| \tag{2.8}$$

where λ in (2.6) is replaced by $\mathbf{P}^{(3)}\bar{\mathbf{X}}$ based on the third equation, allowing to estimate a distance metric directly in the pixel space by comparing $\mathbf{x}$ with the reprojected $\frac{1}{\mathbf{P}^{(3)}\bar{\mathbf{X}}}\mathbf{P}^{(1:2)}\bar{\mathbf{X}}$. The vector from the reprojected point to the observed point, i.e. $\mathbf{x} - \frac{1}{\mathbf{P}^{(3)}\bar{\mathbf{X}}}\mathbf{P}^{(1:2)}\bar{\mathbf{X}}$, is referred to as residual vector. Different camera models result in different reprojected 2D points as we will see later.

Even though the pinhole camera seems intuitive and very complete at first glance, in practice it's not always the best option to model our cameras. This is mainly due to two reasons that we will look into next: 1) the presence of the unknown λ_j in (2.6) adds complexity to the problem which might be unnecessary in some cases, allowing a simpler model (affine camera model) to be used instead; and 2) lens distortion are not considered by the pinhole camera model, limiting its use in real applications.

Affine camera model

An affine camera model consists of an approximation of the pinhole model around a reference 3D point, usually the centroid of a point cloud. Equivalently, you can also see the affine camera model as a constrained pinhole model, where the first three elements of the last row of $\mathbf{P}$ are set to zero. Such approximation allows us to go from the camera equation (2.6) to

$$\mathbf{x} = \mathbf{A}\mathbf{X} + \mathbf{b} \tag{2.9}$$

where $\mathbf{A} \in \mathbb{R}^{2\times 3}$ and $\mathbf{b} \in \mathbb{R}^2$ are unconstrained. A particular case of affine cameras, denoted weak-perspective, further assumes that the rows of $\mathbf{A}$ consist of the first two rows of a scaled rotation matrix. When that scale is unitary, we denote it as orthographic camera projection and can be visualized in Figure 2.2. The reprojection error for an affine camera model simply corresponds to the norm of the residual vector $\mathbf{x} - (\mathbf{A}\mathbf{X} + \mathbf{b})$, i.e,

$$r = \|\mathbf{x} - (\mathbf{A}\mathbf{X} + \mathbf{b})\| . \tag{2.10}$$

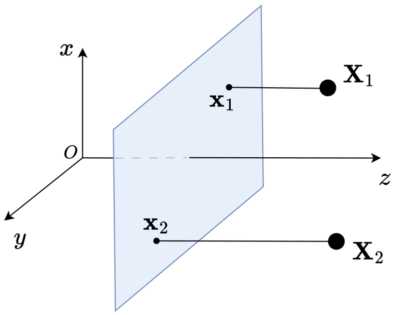

Figure 2.2: Visualization of the affine (orthographic) camera model. In this case, the viewing ray for each 3D point is parallel to the optical axis z, and its intersection with the image plane determines the 2D coordinates of the point in the image.

The accuracy of an affine camera model is similar to the one of a pinhole camera model when the distance from the camera center to the 3D points is approximately the same for all points. This is because the affine camera model assumes that the camera center is at infinity i.e. all viewing rays are parallel. In such cases where the pinhole distortion is negligible, the simplicity of the affine camera model, in particular the absence of the unknown scaling factor, allows us to formulate problems that have much more desirable properties compared to the pinhole camera model. Some examples of such properties regarding Structure-from-Motion will be covered in the following chapter.

Distortion models

As previously stated, the pinhole camera model does not account for lens distortion. This is a severe limitation for practical applications since most real cameras are affected by such distortions to a certain degree. An example of lens distortion is shown in Figure 2.3. This lens distortion effect is not always undesired since it can be used to increase the field of view of a camera sensor, as it usually can be seen in security cameras or GoPro cameras, allowing a much higher coverage of a scene with a single image compared to an ideal pinhole camera [4]–[6].

There are two main classes of methods to model lens distortion: distortion models and undistortion models [8]. Distortion models apply a distortion function to the reprojected points, allowing direct comparison in the image plane. Undistortion models assume the inverse relation, i.e., an undistortion function is applied to the

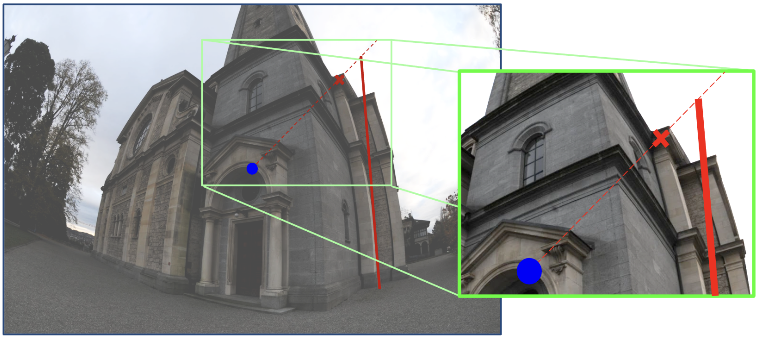

Figure 2.3: An example of an image from the Kirchenge dataset [7] with radial distortion. Without distortion, the edge of the wall represented with a red line segment would be projected to a line in the image. However, due to the presence of radial distortion, the projections (e.g. the red cross) are pushed towards the center of distortion (in blue). The distortion effect is stronger as you get further away from the center of the distortion.

image points, allowing comparison between the undistorted image point and the reprojected point. These models have the advantage that the distortion mapping is a function of the observed image points and hence independent from the camera matrix and 3D point, which are unknown in Structure-from-Motion problems.

Lens distortion can be decomposed into two components, radial and tangential distortion. In modern cameras, tangential distortion can usually be ignored since radial distortion effects are much more dominant. Radial distortion is frequently modeled as follows. Let us assume a calibrated camera setup $\mathbf{P}_c = [\mathbf{R}, \mathbf{t}]$. The undistortion model decomposes the projection into two parts, one for the undistorted projection and another for the lens distortion. For the undistorted projection (x_u, y_u), the process is similar to the pinhole camera model

$$\mathbf{x}_u = \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \frac{1}{\mathbf{P}_c^{(3)}\bar{\mathbf{X}}}\mathbf{P}_c^{(1:2)}\bar{\mathbf{X}}. \tag{2.11}$$

The relation between the normalized distorted image measurement $\mathbf{x}_d = (x_d, y_d)$, with $\bar{\mathbf{x}}_d = \mathbf{K}^{-1}\bar{\mathbf{x}}$, and the undistorted projection (x_u, y_u) is described by the following equation

$$\mathbf{x}_u = \gamma(d)\mathbf{x}_d \tag{2.12}$$

where $\gamma(d)$ is a rational function function on the magnitude $d = \|\mathbf{x}_d\|$

$$\gamma(d) = \frac{1 + c_1 d^2 + c_2 d^4 + c_3 d^6 + \ldots}{1 + k_1 d^2 + k_2 d^4 + k_3 d^6 + \ldots}. \tag{2.13}$$

The Brown-Conrady model [9], [10] is an example of a radial lens distortion model using rational functions. The model shown in (2.12) and (2.13) is an undistortion model. For distortion models, $\mathbf{x}_u$ and $\mathbf{x}_d$ are swapped in the equations (2.12) and (2.13), such that the function $\gamma(d)$ (and d) depend on $\mathbf{x}_u$ instead. Besides rational functions like (2.13), radial distortion is also represented using division models. In division models, just like the Fitzgibbon model [11], the numerator of (2.13) is 1, leaving only a polynomial on the denominator

$$\gamma(d) = \frac{1}{1 + \kappa(d)} = \frac{1}{1 + k_1 d^2 + k_2 d^4 + k_3 d^6 + \ldots}. \tag{2.14}$$

which often leads to simpler inference problems.

1D Radial camera model

Instead of explicitly modeling for radial distortion, one can simply derive a camera model which is invariant to radial distortion [12]. One way to do it is to assume that the principal point of the camera is known and coincides with the center of distortion. Under this assumption, the camera equations using the Fitzgibbon [11] undistortion model become

$$\lambda \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 + \kappa(d) \end{bmatrix} = \begin{bmatrix} f\mathbf{r}_1 & ft_1 \\ f\mathbf{r}_2 & ft_2 \\ \mathbf{r}_3 & t_3 \end{bmatrix} \bar{\mathbf{X}}. \tag{2.15}$$

where $\tilde{x} = x - x_c$ and $\tilde{y} = y - y_c$. Note that the first two equations of (2.15) are now independent from the third one, and thus are invariant to the effects of radial distortion. This suggests that one could drop the third equation completely, resulting in what is referred to as 1D radial camera model

$$\lambda \tilde{\mathbf{x}} = \mathbf{P}^{(1:2)} \bar{\mathbf{X}} \tag{2.16}$$

where $\mathbf{P}^{(1:2)} = f[\mathbf{R}^{(1:2)} \quad \mathbf{t}^{(1:2]}]$ is a 2×4 matrix. The reason why it's referred to as 1D radial camera is because, contrarily to the pinhole camera model which maps a 3D point to a 2D point, with this model a 3D point is now mapped to a line in 2D - $\lambda\tilde{\mathbf{x}}$ - that passes through the center of the image, as can be seen in Figure 2.4.

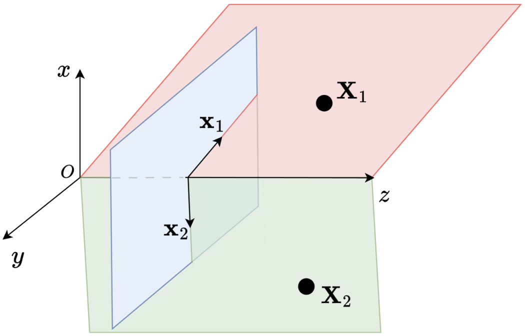

Figure 2.4: Visualization of the 1D radial camera model. As the information regarding the radial component of the projection is disregarded, each 3D point is mapped to a line passing through the center of distortion of the image. This line corresponds to the intersection between the image plane and a plane containing the optical axis z and the 3D point.

Even though we gained radial invariance by considering such a model, dropping the third equation in (2.15) means that we are throwing away valuable information about the scene. For instance, using a pinhole camera model, you need detections in at least 2 viewpoints in order to be able to triangulate a point in 3D. With a 1D radial camera model, you need at least 3 viewpoints.

2.3 Feature Extraction and Matching

Having covered how to represent 3D points and camera models, the only main component left to be able to formulate a Structure-from-Motion problem is to understand how to go from captured images from multiple viewpoints to sets of 2D points detected on those same images - the actual inputs to the SfM problem. This procedure is out-of-the scope of this thesis, but given its importance in the overall problem, I believe its basics should be explained.

In general terms, the goal is to detect the 2D projections of 3D points across all the available images. Whenever a point is detected in a pair of images, we say that we have a match or correspondence. In particular, if $\mathbf{x}$ in image 1 (with camera matrix $\mathbf{P}$) is correctly matched with $\mathbf{x}'$ in image 2 (with camera matrix $\mathbf{P}'$), it means that

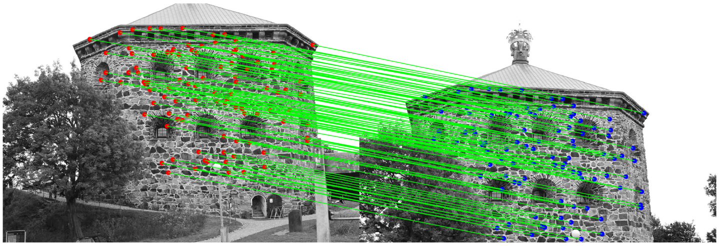

Figure 2.5: Example of sparse correspondences between two viewpoints with intersecting field-of-view. In this example with images from Skansen Kronan [13], SIFT is used to detect keypoints (red and blue), and matched keypoints are represented through the green lines connecting them.

there exists a 3D point $\mathbf{X}$ such that

$$\lambda \mathbf{x} = \mathbf{P}\mathbf{X}, \quad \lambda' \mathbf{x}' = \mathbf{P}'\mathbf{X}. \tag{2.17}$$

This is done for all available images and for as many points as possible. For each detected point, its 2D coordinates over several images are referred to as point track. There are several different methods to generate these point tracks (i.e. detect and find matches) given a set of input images. In this section I divided it into two main categories based on the input sequence: 1) the images are collected from uncorrelated viewpoints; and 2) the input images correspond to an ordered sequence of images with relatively small motion between them, e.g. video-sequence.

The first category consists of detecting points in an image with interesting photometric features, such as edges or corners. The reason why these points, referred to as keypoints, are considered interesting is that there are strong image gradients at their location, which corresponds to the standard feature descriptor used. Given that images usually contain much more content than just edges and corners, such as low-texture regions, keypoint features are typically sparse as shown in Figure 2.5. The most widely used keypoint features include SIFT [14], SURF [15], FAST [16] or Harris Corner [17]. Each method builds a descriptor for each keypoint based on the image gradients and/or pixel values at a neighborhood of the keypoint itself. More recent methods replace the heuristics of hand-crafted methods like SIFT by learning a good feature space for keypoints from training data. Examples of learning-based detectors include LIFT [18], SuperPoint [19] or LoFTR [20]. By learning from training data, learning-based detectors result in descriptors that are less sensitive to pixel

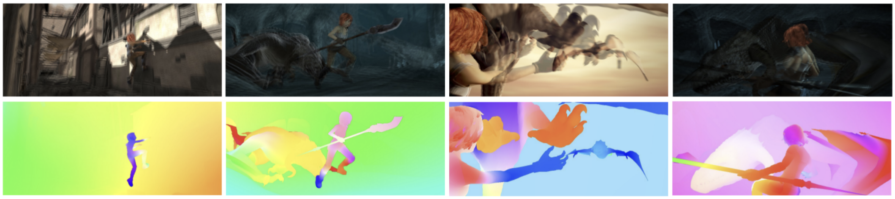

Figure 2.6: Examples of dense correspondences from video sequences. (Top) Overlaid image of two consecutive image frames of 4 different sequences in the Sintel dataset [24]. (Bottom) Ground-truth dense flow fields of the two consecutive image frames depicted above. Each color is mapped to a different flow field direction. Image adapted from [25].

values and more to the semantic information of the image itself, hence generalizing better for different light conditions or weather. Once the keypoints in a pair of images are detected, they are matched typically using nearest neighbors search [21], [22], i.e., each keypoint in image A is matched to the keypoint in image B with the most similar descriptor. Learning-based methods for matching sets of keypoint detections like SuperGlue [23] have also been proposed recently.

The second category consists of densely matching points from one image to another. These are typically referred to as optical flow [26] and usually follow the assumption that small displacements occur between images, i.e., a 2D point in image 1 can be found in a neighborhood of the same location in image 2 (see Figure 2.6). Conventional methods consist of hand-crafted optimization problems that minimize per-pixel photometric errors (corresponding image regions should be similar) with a regularization term that imposes plausible locations of the pixel on the second image. Just like with sparse detectors, more recently learning-based methods have been replacing the heuristics that were used in conventional methods with impressive results. Some examples of optical flow networks are FlowNet [25], [27], RAFT [28] or PWC-Net [29]. An extensive comparison of different learning-based optical flow methods can also be found in [30]. The advantage of learning-based flow estimation is the fact that they can generalize better for larger displacements or different light conditions, just like in the case of sparse detectors. The fact that optical flow provides dense correspondences can also result in more accurate camera calibration estimation, at the cost of increased problem size since detections per image can go from hundreds for sparse detections to thousands or even millions for dense correspondences.

CHAPTER 3

Structure-from-Motion

In this chapter I will properly formulate the Structure-from-Motion problem and some of its subproblems, just e.g. triangulation, camera resectioning, and pose averaging, based on the primitives presented in the previous chapter. Besides introducing the problem, I'll also discuss bundle adjustment and motivate why global and incremental Structure-from-Motion methods are valuable, with particular focus on global factorization-based methods and pOSE formulations given their relevance to this thesis. At the end of this chapter, I will also briefly extend the problem to non-rigid cases, i.e., Structure-from-Motion problem instances in which the scene is deformable over time.

3.1 Multiview Geometry and SfM problems

Let us assume now that we have collected n points tracks across F images using one of the methods described in Section 2.3. Using (2.6) for a pinhole camera model, we can write the camera equations for each of the detected point in each view as

$$\lambda_{ij}\bar{\mathbf{x}}_{ij} = \mathbf{P}_i\bar{\mathbf{X}}_j, \quad i = 1,\dots,F \text{ and } j = 1,\dots,N \tag{3.1}$$

where $\mathbf{P}_i$ is the 3×4 camera matrix of the ith camera, $\bar{\mathbf{X}}_j$ the homogeneous representation of the jth 3D point, $\bar{\mathbf{x}}_{ij}$ its projection into the ith camera, and λ_{ij} is some scalar.

Generically, the Structure-from-Motion problem [3] consists of jointly estimating the camera matrices $\mathbf{P}_i$, 3D points $\mathbf{X}_j$ and projective depths λ_{ij} that better fit the camera model (3.1) (or any other camera model used), for a given set of input 2D point tracks $\mathbf{x}_{ij}$. Given that each camera matrix [1] and 3D point has 11 and 3 degrees of freedom, respectively, and that we get an extra λ_{ij} per 2D image point, the total number of unknown parameters to be estimated is $11F + 3N + NF$ (assuming all points are visible in all views). However, as we will see next, the degrees of freedom of the problem are lower.

Reconstruction Ambiguity and Degrees of Freedom

One thing to consider regarding Structure-from-Motion is the presence of projective ambiguity in the solutions for $\mathbf{P}_i$ and $\mathbf{X}_j$. Let us assume that we know the correct $\mathbf{P}_i^* = \mathbf{K}^*[\mathbf{R}^* \quad \mathbf{t}^*]$ and $\mathbf{X}_j^*$ (along with λ_{ij}^*) that satisfy (3.1). Then any solution $\mathbf{P}_i = \mathbf{P}_i^*\mathbf{H}$ and $\bar{\mathbf{X}}_j = \mathbf{H}^{-1}\bar{\mathbf{X}}_j^*$, where $\mathbf{H} \in \mathbb{R}^{4\times4}$ is a projective transformation, also fits (3.1) since $\mathbf{P}_i\bar{\mathbf{X}}_j = \left(\mathbf{P}_i^*\mathbf{H}\right)\left(\mathbf{H}^{-1}\bar{\mathbf{X}}_j^*\right) = \mathbf{P}_i^*\bar{\mathbf{X}}_j^*$. In the case where the cameras $\mathbf{P}_i$ are calibrated, i.e. $\mathbf{K}_i$ is known, $\mathbf{H}$ is a similarity transformation. This means that when solving the Structure-from-Motion problem we might obtain any of the possible solutions $\mathbf{P}_i$ (and $\mathbf{X}_j$) but since both $\mathbf{P}_i^*$ and $\mathbf{H}$ are unknown, there is no way to directly obtain $\mathbf{P}_i^*$. There are, however, ways to estimate $\mathbf{H}$ and go from a solution $\mathbf{P}_i$ to $\mathbf{P}_i^*$ by taking advantage of the structure of $\mathbf{P}_i^*$ and projective geometry concepts like the plane at infinity. Such methods are usually referred to as auto-calibration or self-calibration [3], [31] and deeply depend on the assumed camera model and prior scene knowledge.

In the more general case with uncalibrated cameras, given that a projective transformation has 15 degrees of freedom, the Structure-from-Motion problem has $11F + 3N + NF - 15$ degrees of freedom. Since each 2D point gives us 3 equations as per (3.1), we can estimate how many cameras/points we need in order to solve the problem by evaluating $3FN \geq 11F + 3N + NF - 15$. For instance, for the two-view problem $F = 2$, we get that we need $N \geq 7$ 3D points.

We refer to projective reconstruction as the reconstruction consisting of the estimated sets of $\mathbf{P}_i$ and $\mathbf{X}_j$ that suffer from projective ambiguity. After performing

[1] If the intrinsic calibration $\mathbf{K}_i$ for each camera is known, then the number of unknown parameters per camera is reduced to 6.

auto-calibration, i.e. obtaining $\mathbf{H}$ for an estimated $\mathbf{P}_i$, the desired $\bar{\mathbf{X}}_j^* = \mathbf{H}\bar{\mathbf{X}}_j$ and $\mathbf{P}_i^* = \mathbf{P}_i\mathbf{H}^{-1}$ can be retrieved, along with its intrinsic and extrinsic calibration. This is referred to as Euclidean reconstruction.

Triangulation and Camera resectioning

In this section, we will look into two subproblems based on (3.1) that can arise when either the camera matrix or the 3D points are known.

Let us now consider the triangulation problem, in which camera matrices $\mathbf{P}_i$ are known and we want to estimate the 3D points $\bar{\mathbf{X}}_j$. By having multiple observations of a point $\bar{\mathbf{X}}$ in different views, using (3.1) we can build a system of equations such as

$$\begin{bmatrix} \mathbf{P}_1 & -\bar{\mathbf{x}}_1 & 0 & \dots & 0 \\ \mathbf{P}_2 & 0 & -\bar{\mathbf{x}}_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_F & 0 & 0 & \dots & -\bar{\mathbf{x}}_F \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}} \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_F \end{bmatrix} = \mathbf{0}. \tag{3.2}$$

This problem (or a smaller version of it in which the λ_i are removed) can be solved using Direct Linear Transforms (DLT) [3], [32]. Note that there are $4 + F$ unknowns and 3 equations per 2D observation, meaning that we need $F \geq 2$ in order to be able to triangulate a 3D point. Additionally, since $\mathbf{P}_i^*$ is assumed to be known, there is no projective ambiguity in the obtained solution, i.e., the estimated 3D point corresponds to the correct $\mathbf{X}^*$ and we obtain a Euclidean reconstruction.

The converse problem, i.e. known 3D points and unknown camera matrix, is referred to as camera resectioning and can be solved similarly with DLT for the system

$$\begin{bmatrix} \mathcal{D}(\bar{\mathbf{X}}_1) & -\bar{\mathbf{x}}_1 & \mathbf{0}_{3\times1} & \dots & \mathbf{0}_{3\times1} \\ \mathcal{D}(\bar{\mathbf{X}}_2) & \mathbf{0}_{3\times1} & -\bar{\mathbf{x}}_2 & \dots & \mathbf{0}_{3\times1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{D}(\bar{\mathbf{X}}_N) & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \dots & -\bar{\mathbf{x}}_N \end{bmatrix} \begin{bmatrix} \mathbf{P}^{(1)^T} \\ \mathbf{P}^{(2)^T} \\ \mathbf{P}^{(3)^T} \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \mathbf{0}. \tag{3.3}$$

for a camera $\mathbf{P}$ and where

$$\mathcal{D}(\bar{\mathbf{X}}_j) = \begin{bmatrix} \bar{\mathbf{X}}_j^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \bar{\mathbf{X}}_j^T & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \bar{\mathbf{X}}_j^T \end{bmatrix}. \tag{3.4}$$

Note that there are $12 + N$ unknowns and 3 equations per 2D observation, meaning that we need $N \geq 6$ in order to be able to estimate an uncalibrated camera matrix. If the intrinsic parameters of the cameras are known, instead of the DLT system (3.3), one can use Perspective-n-Point (PnP) solvers [33], [34] that directly estimate camera rotations and translations from known 3D points and 2D projections. For both cases, just like in the triangulation case, since we are assuming that $\mathbf{X}_j^*$ are known, there is no projective/similarity ambiguity.

3.2 Bundle Adjustment

A natural way to try to estimate the camera matrices and 3D points that fit (3.1) is to minimize reprojection errors [3], [35]. For the pinhole camera model, this results in the following optimization problem

$$\underset{\mathbf{P}_1,\ldots,\mathbf{P}_F,\bar{\mathbf{X}}_1,\ldots,\bar{\mathbf{X}}_N}{\text{minimize}} \quad \sum_{i=1}^{F}\sum_{j=1}^{N} w_{ij} \left\| \mathbf{x}_{ij} - \frac{1}{(\mathbf{P}_i^{(3)}\bar{\mathbf{X}}_j)} \mathbf{P}_i^{(1:2)}\bar{\mathbf{X}}_j \right\|^2 .$$

where $w_{ij} = \{0, 1\}$ specifies whether the observation of point j is available in image i. The optimization , referred to as Bundle Adjustment (BA), consists of a nonlinear least squares problem and is typically solved for camera matrices and 3D points jointly using Levenberg-Marquardt (LM) [3], [35], an iterative 2nd order optimization method that applies a trust region approach to the Gauss-Newton method. Since the reprojection error is nonlinear, in order to apply LM we need to, in each iteration, approximate the residual $\mathbf{x}_{ij} - \frac{1}{(\mathbf{P}_i^{(3)}\bar{\mathbf{X}}_j)}\mathbf{P}_i^{(1:2)}\bar{\mathbf{X}}_j$ by its first order Taylor expansion around the current cameras and 3D points estimations.

Being a second-order method, it requires a matrix inversion of size (at least) $(11F + 3N) \times (11F + 3N)$ for the uncalibrated case, which given the computational complexity of matrix inversion, does not scale well with the number of points and camera views. There are a few ways to circumvent this however, including 1) applying bundle adjustment to a subset of the points/cameras only and calculating the remaining using triangulation/camera resection; 2) alternate optimization between cameras and 3D points, i.e., repetitions of camera resectioning and triangulation starting from an initial guess; and 3) apply sparse matrix methods and take advantage of the structure of the matrices in the problem to invert smaller matrices using Schur complement trick [36], for instance.

The major issue of bundle adjustment consists of its sensitivity to initialization. As previously mentioned, bundle adjustment is an iterative optimization problem, and as so requires some initial guess of the camera matrices and 3D points. In fact, in most practical cases it has been long observed that bundle adjustment does not converge to global minima unless the initial guess is within its close neighborhood. This narrow basin of convergence can be attributed to the effect of the division by $z_{ij} = \mathbf{P}_i^{(3)}\bar{\mathbf{X}}_j$ on each residual, which results in a non-convex loss and additionally creates a cost barrier between negative and positive values of z_{ij}. Consequently, if an initialization of camera matrices and 3D points results in negative z_{ij} it would be extremely unlikely for the optimization to recover from it.

Given this undesirable convergence property of the BA, it is necessary to precede it with some other method (or sequence of methods) that provides a more robust initialization that can then be refined by BA. For this purpose, there are two main classes of methods, Global and Incremental SfM, each with different properties and approaches to the problem of recovering initial guesses for cameras and 3D points.

3.3 Global and Incremental SfM Pipeline

Global Structure-from-Motion

The first class of methods consists of a pipeline of different modules that divide the overall problem of estimating cameras and 3D points into sub-problems solved sequentially. The conventional pipeline for Global Structure-from-Motion typically considers calibrated cameras and can be vaguely described by the following steps:

1. Pairwise camera pose estimation: from the 2D correspondences between two images, estimate essential matrix and retrieve relative camera poses from it (up to scale) [3]. This is done for all pairs of images with overlapping 2D detections. Usually, frameworks like RANSAC and minimal solvers are used in order to ignore mismatches in the input 2D correspondences;

2. Pose averaging: from the relative/pairwise poses from the previous step, solve an averaging problem that computes global camera poses. This can be decomposed into some subproblem, e.g. rotation averaging, as we will see next;

3. Triangulation: from the global camera poses and input 2D correspondences, triangulate 3D points using multi-view geometry;

4. Refine camera poses and 3D points using Bundle Adjustment.

This pipeline can vary depending on the particular problem instance (i.e. constraints and/or assumed priors), but these steps capture the overall idea. Some of the subproblems mentioned here were already briefly introduced before (e.g. triangulation, bundle adjustment), but pairwise estimations and pose averaging might sound completely new to some readers, so I will just briefly introduce them as well in the context of Structure-from-Motion.

Minimal Solvers and RANSAC

A minimal solver consists of a solver that uses the minimum amount of input data possible to fit a certain model. For instance, let's consider the case of triangulation of a 3D point given 2D observations in multiple images. As we covered before, in order to triangulate a point we need at least 2 views, assuming a pinhole camera model. This means that a solver would take 2 observations of that point in 2 images and compute the solution for the 3D point $\bar{\mathbf{X}}$. However, by eliminating the projective depths in both camera equations and fixing the scale of $\bar{\mathbf{X}}$ (by setting the last value to 1, for instance), one gets that only 3 equations (one from one view, and two from the other) are needed to estimate $\bar{\mathbf{X}}$. A solver that only uses these 3 equations would be considered a minimal solver for the triangulation problem. In the case of Global Structure-from-Motion, minimal solvers are usually used to estimate essential/fundamental matrices [3] between a pair of image views. These matrices encode a geometrical relationship (relative pose) between the images that can be estimated from point correspondences between them. In the case of calibrated setup, five point correspondences are needed to estimate the essential matrix [37]. For uncalibrated cameras, 7 correspondences are needed to estimate the fundamental matrix, however, the normalized 8-point algorithm [38], which uses 8 correspondences, is often chosen given its simplicity and robustness (in fact, it can also be used to approximate calibrated cameras, followed by a correction step).

One might wonder why should we use just a subset (minimal set in this case) to estimate a model instead of all data available. While in theory it would make sense to do so, in practice it is very common that the input data is contaminated with outliers, i.e., data that do not fit our model. Hence, by using all data, outliers would be used to estimate the model, potentially resulting in poor model fitting. To increase the method's robustness to outliers, frameworks like RANSAC [39] are used. In each iteration of RANSAC, a minimal set is sampled from the input data, and a model is estimated using the minimal solver. After that, the estimated model is used to count the number of inliers in all inputs through the use of some metric and a threshold. This process is repeated multiple times, in which of them a different minimal set is sampled and the model that resulted in more inliers from all iterations is kept (and possibly refined). By using a minimal set to estimate the model in each iteration, the chances of that set containing only inliers are increased.

While minimal solvers and RANSAC are quite useful in the conventional Structure-from-Motion pipeline, when the complexity of the underlying problem grows, the solutions provided by these methods might not be robust to noise or accurate enough, resulting in modeling errors that can then propagate to the pose averaging steps and ultimately lead to a bad initialization to bundle adjustment. Non-rigid Structure-from-Motion, which we will look into later on, is one example of such problem instances for which is hard to design a minimal solver given the additional complexity (more degrees of freedom) of non-rigid models.

Pose averaging

Pose averaging consists of estimating global camera poses from pairwise pose estimations (e.g. obtained from essential matrices). This problem is usually represented as a graph, in which the global poses are nodes and the estimated relative poses are edges between the nodes of the corresponding camera pairs. In the context of Structure-from-Motion, there are several ways to solve the problem but most of them have in common the decomposition into rotation and (some sort of) translation averaging, i.e, estimation of global camera rotations and translations from pairwise estimations.

The problem of rotation averaging has been widely studied [13], [40]–[44] and is usually formulated as finding the rotation matrices $\mathbf{R}_i \in \mathcal{SO}(3), i = 1, \dots, F$ such that fit the model $\mathbf{R}_j = \mathbf{R}_{ij}\mathbf{R}_i$, where $\mathbf{R}_{ij}$ is the relative rotation found through epipolar geometry between camera i and j. Local optimization [45] is the preferred method to solve large-scale rotation averaging, however just like for bundle adjustment, it only guarantees convergence to the nearest local minimum. Other methods like spectral decomposition [41], [42] and Semidefinite Programming [42], [43], [46] provide solutions with global optimality guarantees for low levels of noise, at the cost of higher computational expensiveness [47], [48]. Once global camera rotations are known, global camera translations (and optionally 3D points) can be estimated using

Second Order Cone Programming [49], [50] or linear methods [40], [42].

A somehow similar problem to pose averaging is point set registration, in which (incomplete) point clouds in different local coordinate frames are jointly registered to a global coordinate frame, along with an average point cloud. This problem is analogous to pose averaging in situations in which the camera sensor also has a depth channel. In this case, after creating the 2D point tracks over different views, a 3D point cloud can be obtained for each image using the depth channel measurements. The different point clouds can then be “averaged" out through point set registration, along with the global poses of the cameras.

Incremental Structure-from-Motion

In order to decrease the effect of the dimensionality problem of Global Structure-from-Motion, in particular of the pose averaging methods, one can instead solve the so-called Incremental Structure-from-Motion pipeline. As can be deduced from its name, this consists of an incremental approach to Structure-from-Motion, in which we start with a small number of views (2 for instance), and incrementally add new views and points until all views and points are solved for [47]. In general terms, the method can be described by the following steps:

1. Solve for a subset of the points and cameras (at least two views) using, for instance, some Global Structure-from-Motion method;

2. Add a new view: find a new image in which a large quantity of the 2D points has already been triangulated using previous views. This means that 2D-3D correspondences are available, so the camera matrix for the new view can be found by solving the camera resectioning problem. RANSAC can be used here to discard outliers in the 2D-3D correspondences;

3. Triangulate points in the new view: for all the remaining points in the new view that haven’t been triangulated before, check if there are corresponding detections on previous views. If yes, triangulate those points using multiview geometry;

4. Refine camera poses and 3D points using Bundle Adjustment;

5. Repeat steps 2, 3, and 4.

Again, this pipeline may vary depending on the problem priors and constraints.

The main advantage of this incremental approach is that both camera resectioning and triangulation can be solved very efficiently, so the method scales much better than its global counterpart for larger problem sizes. This nice scaling property makes Incremental Structure-from-Motion the go-to solution in most large-scale state-of-the-art software for 3D reconstruction from images [51], [52]. On the other hand, by only using a subset of views in each step, useful information about the scene might be contained in the unused views, overall resulting in estimations with high uncertainty. In the worst-case scenario, this high uncertainty might result in a bad initialization for the bundle adjustment that will not converge to the desired minima given the properties discussed in Section 3.2. In fact, it has been shown [42], [48] that global methods outperform incremental methods in terms of accuracy.

3.4 Factorization-based SfM

Given that global methods are able to output more accurate reconstructions than incremental methods, it would be desirable to boost the flexibility and scalability of the former. This leads us to global factorization optimization problems, which is a substantial part of this thesis. These methods consist of an alternative approach to the Global SfM pipeline described in the previous section, i.e., absolute camera poses and 3D points are estimated simultaneously for all images and 2D tracks. Factorization-based methods for SfM are based on the camera equations (3.1), where it was observed that measurements of a point in multiple images can be stacked vertically, resulting in

$$\begin{bmatrix} \lambda_1 \bar{\mathbf{x}}_1 \\ \lambda_2 \bar{\mathbf{x}}_2 \\ \vdots \\ \lambda_F \bar{\mathbf{x}}_F \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_F \end{bmatrix} \bar{\mathbf{X}}. \tag{3.5}$$

Similarly, observations in a single image can be stacked horizontally, resulting in the system

$$\underbrace{\begin{bmatrix} \lambda_{1,1}\bar{\mathbf{x}}_{1,1} & \lambda_{1,2}\bar{\mathbf{x}}_{1,2} & \dots & \lambda_{1,N}\bar{\mathbf{x}}_{1,N} \\ \lambda_{2,1}\bar{\mathbf{x}}_{2,1} & \lambda_{2,2}\bar{\mathbf{x}}_{2,2} & \dots & \lambda_{2,N}\bar{\mathbf{x}}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{F,1}\bar{\mathbf{x}}_{F,1} & \lambda_{F,2}\bar{\mathbf{x}}_{F,2} & \dots & \lambda_{F,N}\bar{\mathbf{x}}_{F,N} \end{bmatrix}}_{\Lambda \odot \bar{\mathbf{M}}} = \underbrace{\begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_F \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \dots & \bar{\mathbf{X}}_N \end{bmatrix}}_{\bar{\mathbf{X}}}, \tag{3.6}$$

where $\mathbf{P} \in \mathbb{R}^{(3F)\times 4}$, $\bar{\mathbf{X}} \in \mathbb{R}^{4\times N}$ are the unknown matrices containing the camera calibrations (typically uncalibrated) and 3D points, respectively, and the matrix $\Lambda \in \mathbb{R}^{F\times N}$ contains all the unknown projective scales λ_{ij}. The matrix $\bar{\mathbf{M}} \in \mathbb{R}^{(3F)\times N}$ contains the concatenations of all the 2D point tracks in homogeneous coordinates, and $\left(\Lambda \odot \bar{\mathbf{M}}\right) \in \mathbb{R}^{(3F)\times N}$ represents the scaled version of $\bar{\mathbf{M}}$ by the corresponding projective depths, i.e., each 3×1 block consists of $\lambda_{ij}\bar{\mathbf{x}}_{ij}$. The factorization problem comes from, as seen in (3.6), the matrices $\mathbf{P}$ and $\bar{\mathbf{X}}$ being indeed a rank-4 factorization of the matrix $\left(\Lambda \odot \bar{\mathbf{M}}\right)$. The issue, however, is the fact that Λ is also unknown, making it not possible to factorize $\left(\Lambda \odot \bar{\mathbf{M}}\right)$ into $\mathbf{P}$ and $\bar{\mathbf{X}}$.

There are some way to circumvent this issue as we will see next, either by having some assumptions regarding Λ, or by estimating Λ and the pair $\mathbf{P}$ and $\bar{\mathbf{X}}$ in alternating fashion.

Affine Factorization

When an affine camera model is considered, as seen in Section 2.2, the depths λ_{ij} are assumed to be equal and constant. In that case, and if all N points are visible in all F views, the system in (3.6) can be simplified to [53]

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \dots & \mathbf{x}_{1,N} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \dots & \mathbf{x}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{F,1} & \mathbf{x}_{F,2} & \dots & \mathbf{x}_{F,N} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{P}_1^{(1:2)} \\ \mathbf{P}_2^{(1:2)} \\ \vdots \\ \mathbf{P}_F^{(1:2)} \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \dots & \bar{\mathbf{X}}_N \end{bmatrix}}_{\bar{\mathbf{X}}}, \tag{3.7}$$

where $\mathbf{P} \in \mathbb{R}^{(2F)\times 4}$ is now a vertical concatenation of the first two rows of each of the camera matrices $\mathbf{P}_i$. Since the left-hand side of (3.7) is now known, one can directly obtain the factors $\mathbf{P}$ and $\bar{\mathbf{X}}$ through a rank-4 truncation of the SVD of $\mathbf{M}$. In fact, the problem can be further simplified to a rank-3 truncation of the SVD of $\mathbf{M}$ by centering the input data and eliminating the translation vector of each $\mathbf{P}_i$, as described in [53]. Solving for $\mathbf{P}$ and $\bar{\mathbf{X}}$ using SVD of $\mathbf{M}$ allows us to obtain an affine reconstruction of the scene, which can then be updated for a metric reconstruction through autocalibration and/or using additional information about the scene. One can then feed this reconstruction as initialization for bundle adjustment.

Projective Factorization

In situations in which an affine reconstruction is not accurate enough to model the scene and a projective reconstruction is needed instead, other methods need to be applied in order to solve the factorization problem in (3.6). As observed in [54], if the projective depths Λ are known, the factorization problem can be solved in a similar way to the affine factorization by replacing $\mathbf{M}$ by $\left(\Lambda \odot \bar{\mathbf{M}}\right)$. The camera matrices (now a $3F \times 4$ matrix) and the 3D points $\bar{\mathbf{X}}$ can then be obtained through a rank-4 truncation of the SVD of $\left(\Lambda \odot \bar{\mathbf{M}}\right)$. Conversely, if a factorization $\mathbf{P}$ and $\bar{\mathbf{X}}$ is known, the depths in Λ can be estimated by comparing the reprojected $\mathbf{P}_i\bar{\mathbf{X}}_j$ with $\bar{\mathbf{x}}_{ij}$. This suggests that an iterative algorithm can be used [3], where starting from an initial guess for Λ, one can perform the following steps to obtain a projective reconstruction of the scene:

1. For the current estimation of Λ, construct the matrix $\left(\Lambda \odot \bar{\mathbf{M}}\right)$ and retrieve $\mathbf{P}$ and $\bar{\mathbf{X}}$ through a rank-4 truncation of the SVD of $\left(\Lambda \odot \bar{\mathbf{M}}\right)$;

2. For all points, estimate λ_{ij} that minimizes the errors $\|\lambda_{ij}\bar{\mathbf{x}}_{ij} - \mathbf{P}_i\bar{\mathbf{X}}_j\|^2$ from $\mathbf{P}$ and $\bar{\mathbf{X}}$ obtained in the previous step, and construct Λ from the estimated λ_{ij};

3. Repeat steps 1 and 2 until convergence.

The initial guess $\lambda_{ij} = 1$ is reasonable in practice as long as the distance from the 3D points to the camera centers is approximately constant for all views. Due to scale ambiguities, normalization of the depths λ_{ij} should be done for the initial guess and after step 2. One issue with the proposed algorithm is that it has no convergence guarantees to a global minimum, meaning that it can result in poor reconstructions (maybe even not good enough to initialize bundle adjustment). An additional problem, shared with the affine factorization, is that it requires all points to be visible in all views. This constraint is not desirable in practice, since it restrains us from reconstructing sequences, like for instance, captured by a vehicle moving through the city, where points at the beginning of the sequence are not observed after some camera motion.

pOSE models

Besides the already mentioned factorization methods, many other alternating [55], [56] and splitting [57], [58] methods were proposed to solve problems such as (3.6).

However, robustness to noise, robustness to local minima (when starting from a random solution), or slow convergence near the optimum have been some of the continuous problems within these frameworks.

More recently, Hong *et al.* [1], [59] suggested a different approach for a rank-4 factorization in the context of SfM that has been shown to be very reliable in terms of robustness to initialization from a random starting solution. The core of their work consists of a reformulation of the reprojection residual vector as a linear residual on the reprojected point $\mathbf{z}_{ij} = \mathbf{P}_i\bar{\mathbf{X}}_j$, to which they call Object Space Error (OSE) defined as

$$\ell_{\text{OSE}} = \sum_{ij} w_{ij} \left\| \mathbf{z}_{ij}^{(3)} \mathbf{x}_{ij} - \mathbf{z}_{ij}^{(1:2)} \right\|^2 \tag{3.8}$$

Note that occluded points can be modeled with this framework through the use of w_{ij}, just like in the bundle adjustment. The removal of the division by $\mathbf{z}^{(3)}$ in each residual vector avoids the cost barrier issue verified with bundle adjustment around $\mathbf{z}^{(3)} = 0$, making it possible for points to flow from negative to positive values of $\mathbf{z}^{(3)}$ more easily. Using this loss to estimate $\mathbf{P}$ and $\bar{\mathbf{X}}$ would result in an unbiased estimation, however, it can be easily observed that setting $\mathbf{z}_{ij} = 0$ results in a trivial solution to the problem. To avoid such solution, the authors propose to add a loss term to the objective based on an affine camera model, defined as

$$\ell_{\text{Affine}} = \sum_{ij} w_{ij} \left\| \mathbf{x}_{ij} - \mathbf{z}_{ij}^{(1:2)} \right\|^2 . \tag{3.9}$$

The total objective of the proposed framework [1] consists of a linear combination of the OSE and affine terms, resulting in a pOSE loss defined as

$$\ell_{\text{pOSE}} = (1 - \eta)\ell_{\text{OSE}} - \eta\ell_{\text{Affine}} \tag{3.10}$$

where $\eta \in [0, 1]$ determines the weight of each term, and it is shown empirically that a low value of $\eta \approx 0.05$ results in accurate reconstructions.

The problem can be written in the general least squares form

$$\underset{\mathbf{P},\bar{\mathbf{X}}}{\text{minimize}} \quad \|\mathcal{A}(\mathbf{P}\bar{\mathbf{X}}) - \mathbf{b}\|^2 \tag{3.11}$$

where $\mathcal{A}$ is a linear operator on the elements of $\mathbf{P}\bar{\mathbf{X}}$ and $\mathbf{b}$ a vector, both based on the pOSE objective (3.10). The author proposes to solve (3.11) using VarPro [59], a second-order optimization method that uses Levenberg-Marquardt [3], [35] to update the reduced problem on $\mathbf{P}$ only, i.e., in each iteration the 3D points are solved for in closed form as a function of $\mathbf{P}$, and the Jacobians of the residuals in terms of $\mathbf{P}$ are approximated using such formulation. The authors show that not only are they able to achieve accurate reconstruction using the pOSE objective and VarPro, but also the method demonstrated remarkable robustness to initialization from random camera matrices with a convergence rate to a global minimum of above 90% in many benchmark datasets.

Given that this pOSE formulation can be written in the general form (3.11), it allows us to extend the framework to different problem instances in terms of modeling (e.g. point visibility, camera models, scene rigidity) and regularization (the affine term in (3.10) can be seen as regularization, but other terms can be added or replace it, as we will see). Besides this extra flexibility, pOSE formulations have similar computational complexity to bundle adjustment, since they can take advantage of the same sparsity properties and matrices' structure. On the negative side, since all data is used from the beginning, pOSE methods (and factorization methods in general) are more sensitive to outliers in the 2D point correspondences, which can seriously undermine their usefulness in real case scenarios.

3.5 Non-rigid Structure-from-Motion

So far we have been considering a rigid 3D scene, i.e., the 3D points of the scene are static between views and only the cameras are moving. However, in many cases, the scene contains moving or deforming objects which need to be modeled accordingly in other to obtain an accurate reconstruction. This problem is referred to as Non-rigid Structure-from-Motion (NRSfM) and in this section, I will briefly explain how we can incorporate non-rigidity into Global Structure-from-Motion methods. To clarify, I will focus on instances of NRSfM in which there are no two (or more) images capturing the scene at the same instant, as in Figure 3.1. In other words, each image captures the non-rigid scene at a different state of deformation. This consists of a harder problem since if two or more cameras captured the scene at each instant, one could approach the problem as rigid SfM for those subsets of cameras.

Non-rigid Factorization

One of the first attempts to address the problem was proposed by Bregler *et al.* [61]. In their work, they suggest the assumption that, for each view, the scene can be

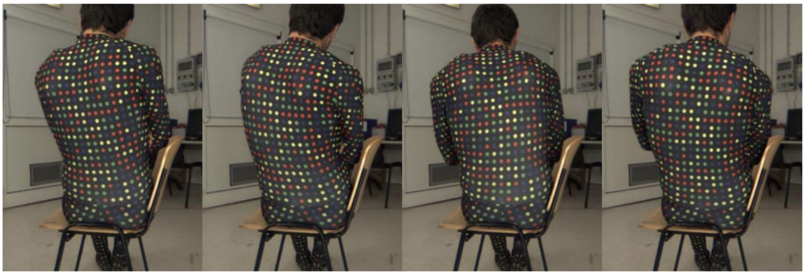

Figure 3.1: Few images from the Back dataset [60], a sequence used for Non-rigid Structure-from-Motion. A single camera captures a man moving his back while sitting on a chair. The colorful pattern of the shirt is used to easily generate the 2D point tracks needed for the SfM algorithm. For each time instant, only one viewpoint is available.

described as a linear combination of K shape basis as

$$\mathbf{X}_i = \sum_{k=1}^{K} c_{ik}\mathbf{S}_k, \quad i = 1, \ldots, F, \tag{3.12}$$

where $\mathbf{S}_k \in \mathbb{R}^{3\times N}, k = 1, \ldots, K$ are the shape bases and c_{ik} is a scalar coefficient corresponding to the kth shape basis in the ith image. Under an orthographic camera model, we can write

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \ldots & \mathbf{x}_{1,N} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \ldots & \mathbf{x}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{F,1} & \mathbf{x}_{F,2} & \ldots & \mathbf{x}_{F,N} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{(1:2)} \left(\sum_{k=1}^{K} c_{1k}\mathbf{S}_k\right) \\ \mathbf{R}_2^{(1:2)} \left(\sum_{k=1}^{K} c_{2k}\mathbf{S}_k\right) \\ \vdots \\ \mathbf{R}_F^{(1:2)} \left(\sum_{k=1}^{K} c_{Fk}\mathbf{S}_k\right) \end{bmatrix} + \begin{bmatrix} \mathbf{t}_1^{(1:2)} \\ \mathbf{t}_2^{(1:2)} \\ \vdots \\ \mathbf{t}_F^{(1:2)} \end{bmatrix} \tag{3.13}$$

and similarly to [53], the translations can be removed by subtracting the 2D centroid for each camera view, resulting in

$$\begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \ldots & \mathbf{x}_{1,N} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \ldots & \mathbf{x}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{F,1} & \mathbf{x}_{F,2} & \ldots & \mathbf{x}_{F,N} \end{bmatrix} = \underbrace{\begin{bmatrix} c_{11}\mathbf{R}_1^{(1:2)} & \ldots & c_{1K}\mathbf{R}_1^{(1:2)} \\ c_{21}\mathbf{R}_2^{(1:2)} & \ldots & c_{2K}\mathbf{R}_2^{(1:2)} \\ \vdots & \ddots & \vdots \\ c_{F1}\mathbf{R}_F^{(1:2)} & \ldots & c_{FK}\mathbf{R}_F^{(1:2)} \end{bmatrix}}_{\Pi} \underbrace{\begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_K \end{bmatrix}}_{\mathbf{S}}. \tag{3.14}$$

where $\mathbf{\Pi} \in \mathbb{R}^{2F\times 3K}$ and $\mathbf{S} \in \mathbb{R}^{3K\times N}$. As it is possible to see from (3.14), under these assumptions the problem goes from a rank-3 factorization in the rigid case to a rank-$3K$ factorization in the non-rigid case. An affine reconstruction can be obtained through SVD of the left-hand side matrix of (3.14), truncated to rank $3K$, followed by correction step that estimates $\mathbf{R}_i, i = 1, \ldots, F$ based on the rotation constraints of the left matrix on the right-hand side of (3.14).

More than a decade later, Dai *et al.* published a method [62] that constrained the problem further. One of the key contributions of the paper was the denoted "Intersection theorem", which states that the correction matrix[2] needed to find the rotation matrices after SVD must lie in the intersection between the $2K^2 - K$ dimensional null-space of a matrix and rank-3 positive semi-definite matrix cone. The problem can be solved with SDP and allows more accurate estimations of rotation matrices than [61]. Additionally, after estimating the rotation matrices, the authors propose a method to find $\mathbf{S}$ and the coefficients c_{ik} through an additional rank-K factorization of $\mathbf{X}^\sharp$, a re-arranged version of $\mathbf{X} = [\mathbf{X}_1^T, \ldots, \mathbf{X}_F^T]^T$ as defined in (3.12).

Low-rank Penalty Functions

Under the restrictive assumptions mentioned before, the methods referred to in the previous section tend to work reasonably well. However, some of those assumptions are not verified in many practical cases. In particular, the assumption that the shape can be decomposed into K shape bases is very impractical when K is not known. Additionally, one might want to give different weights to different shape bases (e.g. enforce $k < K$ bases to be dominant and the remaining to model smaller deformations), which wouldn't be possible with the previously referred approach.

Let's say that one wants to set that the scene can be defined by at most K shape basis, i.e.,

$$\begin{array}{ll} \underset{\mathbf{Z}\in\mathbb{R}^{2F\times N}}{\text{minimize}} & \|\mathbf{M} - \mathbf{Z}\|_F^2 \\ \text{such that} & \text{rank}(\mathbf{Z}) \leq 3K \end{array}. \tag{3.15}$$

Note that parameterizing $\mathbf{Z} = \mathbf{\Pi}\mathbf{S}$ with $\mathbf{\Pi} \in \mathbb{R}^{2F\times 3K}$ and $\mathbf{S} \in \mathbb{R}^{3K\times N}$ would enforce similar rank constraints by construction. A common relaxation of problems like (3.15) [63]–[66] is to replace the rank constraint by a regularization term on the singular values of $\mathbf{Z}$

$$\underset{\mathbf{Z}\in\mathbb{R}^{2F\times N}}{\text{minimize}} \quad \|\mathbf{M} - \mathbf{Z}\|_F^2 + f(\sigma(\mathbf{Z})) \tag{3.16}$$

[2] Analogous to the ambiguity transformation mentioned in Section 3.1 for the rigid case.

where $f : \mathbb{R}^{|\mathbf{Z}|} \rightarrow \mathbb{R}$ with $|\mathbf{Z}| = \max(2F, N)$. Since singular values are non-differentiable functions of $\mathbf{Z}$, splitting methods like ADMM [67] are usually preferred to solve (3.16) given that, for some choices of f, the proximal operator can be obtained in closed form. Furthermore, the formulation (3.16) happens to be convex if f is convex and absolutely symmetric [68]. One such function is the Nuclear Norm [69] $f(\sigma(\mathbf{Z})) = \|\mathbf{Z}\|_* = \sum_l \sigma_l(\mathbf{Z})$ which equally penalizes all singular values of $\mathbf{Z}$. This is the regularization chosen by [62] when trying to find a rank-K factorization of $\mathbf{X}^\sharp$ for the Non-Rigid SfM problem described in the previous section. Even though the nuclear norm has some desirable properties, such as convexity, the fact that it equally penalizes all singular values causes shrinking bias [70], [71]. It has also been shown [72] that in the context of Structure-from-Motion and under the presence of noise it usually gives a weak regularization.

In [66], the authors extend [62] by considering weighted nuclear norm as a regularization term for the factorization of $\mathbf{X}^\sharp$. The weighted nuclear gives different weights to the singular values, reducing the shrinking bias effect. The authors propose a scene-specific regularization in which the weights of the singular values depend on the input sequence and current camera rotation estimations, resulting in more accurate shape reconstructions.

All the methods mentioned so far with rank constraint relaxation are solved using a first-order method, usually ADMM, as already mentioned. Its alternating nature, regardless of its simplicity and fast iterations, can make convergence near optima slow by requiring too many steps [67]. In that regard, second-order methods like Gauss-Newton or Levenberg-Marquardt are preferred given their fast convergence in the optimum neighborhood, usually at the cost of slower iterations. The issue is that in order to be able to apply second-order methods to (3.16) a differentiable parametrization of singular values is needed. In [69] it was shown that for a matrix $\mathbf{X} = \mathbf{B}\mathbf{C}^T$ the nuclear norm can be represented as

$$\|\mathbf{X}\|_* = \min_{\mathbf{X}=\mathbf{B}\mathbf{C}^T} \frac{\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2}{2} \tag{3.17}$$

which means that the problems

$$\underset{\mathbf{Z}}{\text{minimize}} \quad \|\mathbf{M} - \mathbf{Z}\|_F^2 + \sum_l \sigma_l(\mathbf{Z}) \tag{3.18}$$

and

$$\underset{\mathbf{B},\mathbf{C}}{\text{minimize}} \quad \|\mathbf{M} - \mathbf{B}\mathbf{C}^T\|_F^2 + \frac{\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2}{2} \tag{3.19}$$

are actually equivalent. The bilinear factorization problem (3.19) can be solved with second-order methods since both terms are differentiable in $\mathbf{B}$ and $\mathbf{C}$. It is also possible to show that, even though the formulation is no longer convex given the bilinear product if a local minimum satisfies rank$(\mathbf{B}\mathbf{C}^T) < k$, with k being the number of columns of $\mathbf{B}$ and $\mathbf{C}$, then it is globally optimal [73], [74]. The works [75] and [74] also extended these results to other norms and penalties with similar conclusions, opening up the possibility of applying second-order methods to a wide range of rank penalties. Many of these concepts and results were explored in papers D and E of this thesis.

CHAPTER 4

Summary of included papers

In this Chapter, I provide a summary of the five included papers based on the content presented in Section 2 and 3. Paper A concerns the study of global optimality conditions for the point set registration problem using Lagrange duality. The remaining four papers - B, C, D, and E - concern factorization-based methods for Structure-from-Motion, each focusing on a different problem instance. Paper B extends pOSE methods for images with radial distortion by dropping the radial component of the OSE, resulting in a radial distortion invariant pOSE model. Paper C replaces the regularization term of pOSE methods, which are based on an affine camera model, with an exponential term on the projective depths of the 3D points. The proposed regularization, when combined with OSE, penalizes large depths less than pOSE, resulting in more accurate reconstruction while maintaining a wide basin of convergence. Paper D extends pOSE models to Non-rigid Structure-from-Motion by adding an additional regularization on the singular values of the reconstructed matrix. Additionally, the singular values are parameterized as functions of the bilinear factors, allowing optimization with second-order methods which ultimately leads to more accurate reconstructions than competing first-order methods. Finally, Paper E extends the results from Paper D to a wider class of low-rank non-convex penalties. In order to obtain a differentiable formulation, it replaces the non-convex penalties with a surrogate which can be approximated by a quadratic function and optimized using second-order methods.

4.1 Paper A - “Point Set Registration and Global Optimality”

In this paper, we study the global optimality conditions of the Point Set Registration problem, briefly introduced in Section 3.3. To recall, point set registration consists of aligning two or more sets of 3D points with an unknown target point cloud. Such problems can arise in the context of Structure-from-Motion with RGB-D cameras, like Microsoft Kinect, with time-of-flight technology that outputs an additional depth channel containing the depth of each image pixel. This information can be combined with the camera system calibration and 2D point correspondences to generate 3D point tracks over the images, resulting in a point cloud per image. The problem is formulated as follows

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{R},\mathbf{t}} \quad & \sum_{i,j} w_{ij} \|\mathbf{Y}_i - (\mathbf{R}_j \mathbf{X}_{ij} + \mathbf{t}_j)\|^2 \\ \text{subject to} \quad & \mathbf{R}_j \in \mathcal{SO}(3),\ j = 1, \ldots, F \end{aligned} , \tag{4.1}$$

where $\mathbf{X}_{ij}$ is the ith 3D point of the jth available point cloud. The input point clouds are denoted as source point clouds. The point $\mathbf{Y}_i$ corresponds to the ith 3D point of the target point cloud (unknown). The transformations $\{\mathbf{R}_j, \mathbf{t}_j\}$ which transform from the jth source point cloud to the target point cloud are also unknown. The variable w_{ij} determines whether the ith point is available in the jth source point cloud.

The problem is not necessarily novel, and many solutions like [76] have been proposed. The non-linear constraints in (4.1) can be relaxed, resulting in a dual problem that can be formulated as an SDP [77] which is convex and can be solved in polynomial time. However, we study the problem from a different perspective: under which condition can we guarantee that a candidate solution to the problem corresponds to the global minimum? In particular, we study conditions in terms of missing data, target point cloud spatial distribution, and source point cloud noise.

Our approach is inspired by [43], which studies global optimality conditions for the rotation averaging problem and was able to prove global optimality of a solution for low levels of noise on the source pairwise rotations. Note that the point set registration problem is inherently more complex given the effect of missing data in the source point clouds and the structure of the bipartite graph, with the target point cloud and 3D transformations as nodes. The main contributions of the paper can be summarized as: 1) application of Lagrangian duality to the point set registration problem, where given a candidate solution to the primal problem, the corresponding dual variable can be obtained in closed form. This result allows us to verify the global optimality of a local minimizer without solving the SDP (Theorem 1), which leads to a significant speedup under some conditions; 2) derivation of bounds (Theorems 2 and 3) on reconstruction errors that, if fulfilled, are sufficient to guarantee global optimality of a candidate primal solution; and 3) analysis and evaluation of the proposed bounds as functions of missing data, the spatial distribution of the estimated 3D scene, and noise on the source point clouds, using synthetic and real data.

4.2 Paper B - "Radial Distortion Invariant Factorization for Structure-from-Motion"

In this paper, we extend the pOSE model with radial distortion invariance. As explained in Section 3.4, The pOSE model as proposed by [1] utilizes as objective a linear combination of an Object Space Error (OSE) (3.8) and a regularization term based on an affine camera model error (3.9), weighted by the $(1-\eta)$ and η, respectively. We have shown that the OSE, however, consists of a linear approximation of the reprojection error for the perspective camera model, hence does not consider lens distortion effects. Such limitation makes pOSE unsuitable for sequences captured by cameras with a wide field of view, which take advantage of lens radial distortion to capture a larger scene section. We propose to replace the OSE with an equivalent error based on a 1D radial camera model introduced in Section 2.2, which we refer to as Radial OSE (ROSE) and is defined as

$$\ell_{\text{ROSE}} = \sum_{ij} w_{ij} \left\| \frac{\mathbf{x}_{ij}^{\perp}}{\|\mathbf{x}_{ij}\|} \cdot \mathbf{z}_{ij} \right\|^2 \tag{4.2}$$

where $\mathbf{x}_{ij}^{\perp} = (y_{ij}, -x_{ij})$ is an orthogonal vector to $\mathbf{x}_{ij}$ and each $\mathbf{z}_{ij}$ is 2-dimensional. Our proposed loss, similarly to pOSE, consists of a linear combination of ROSE with a regularization term based on the affine camera model

$$\ell_{\text{RpOSE}} = (1-\eta)\ell_{\text{ROSE}} + \eta\ell_{\text{Affine}}. \tag{4.3}$$

and can be minimized using VarPro, maintaining a similar basin of convergence to pOSE while outperforming state-of-the-art factorization methods for sequences with radial distortion. Additionally, we show that ROSE consists of a 1st order Taylor expansion of the maximum likelihood residual $\mathbf{x}_{ij}^{\perp} \cdot \frac{\mathbf{z}_{ij}}{\|\mathbf{z}_{ij}\|}$ around $\mathbf{z}_{ij} = \mathbf{x}_{ij}$, i.e., ROSE can be generalized as

$$\ell_{\text{ROSE}} = \sum_{ij} w_{ij} \left\| \mathbf{x}_{ij}^{\perp} \cdot \frac{\mathbf{v}_{ij}}{\|\mathbf{v}_{ij}\|} + \underbrace{\left(\frac{\mathbf{x}_{ij}^{\perp}}{\|\mathbf{v}_{ij}\|} - \frac{\mathbf{x}_{ij}^{\perp} \cdot \mathbf{v}_{ij}}{\|\mathbf{v}_{ij}\|^3} \mathbf{v}_{ij} \right)}_{\mathbf{J}(\mathbf{v}_{ij})} (\mathbf{z}_{ij} - \mathbf{v}_{ij}) \right\|^2 \tag{4.4}$$

and for $\mathbf{v}_{ij} = \mathbf{x}_{ij}$ we retrieve (4.2). This result combined with the interpretation of affine regularization as a dampening term can also be generalized as

$$\ell_{\text{Affine}} = \sum_{ij} w_{ij} \|\mathbf{v}_{ij} - \mathbf{z}_{ij}\|^2, \tag{4.5}$$

results in a loss RpOSE that can be updated in an outer loop of the optimization. The estimated solution gets closer to the ML estimator as more updates of RpOSE are performed. Note that for $\mathbf{v}_{ij} = \mathbf{x}_{ij}$ we also retrieve the original affine regularization.

In summary, the main contributions of this paper are 1) a new pOSE formulation, named RpOSE, that can handle radially distorted images; 2) we show that the new formulation can be robustly optimized using VarPro converging to the globally optimal solution in the vast majority of cases from random starting solutions; 3) we show that the pOSE formulations can be seen as local approximations of reprojection error opening up the possibility of iteratively approximating the maximum likelihood formulation; and 4) we proposed a Structure-from-Motion pipeline based on RpOSE, bundle adjustment and Euclidean update step that outputs a Euclidean reconstruction from input 2D point track in an uncalibrated setup.

4.3 Paper C - "expOSE: Accurate Initialization Free Projective Factorization using Exponential Regularization"

In this paper, we start by pointing out that pOSE as originally proposed in [1] penalizes 3D points with large depths in the local camera coordinate system due to the affine regularization term in (3.10). This is because as the depth λ increases the first two coordinates of the 3D point $\mathbf{z} = \lambda\bar{\mathbf{x}}$ (for the noiseless case) also increase in magnitude, consequently increasing quadratically the error $\|\mathbf{x} - \lambda\mathbf{x}\|^2$ as the solution gets further away from the minimum located at $\lambda = 1$. The penalization of large depths caused by the regularization term of pOSE results in a slight deterioration of the reconstruction. This is noted in the original paper [1] as the authors plot 3D reconstruction for different values of η, with larger η resulting in a slightly curved reconstruction. Since pOSE is used to initialize bundle adjustment, the reconstruction obtained with pOSE can be refined, hopefully resulting in an accurate reconstruction of the scene. However, it raises the question if there could be datasets in which pOSE does not result in a good enough initialization for bundle adjustment, ultimately leading to poor reconstructions due to the affine regularization term.

We, therefore, propose an alternative regularization term defined as

$$\ell_{\text{exp}} = \sum_{ij} w_{ij} e^{-\frac{\bar{\mathbf{x}}_{ij} \cdot \mathbf{z}_{ij}}{\|\bar{\mathbf{x}}_{ij}\|}} \tag{4.6}$$

which pushes the projection $\mathbf{z}_{ij}$ along the direction $\bar{\mathbf{x}}$. This exponential regularization when combined with an OSE - which we referred to as expOSE - achieves two results: 1) it penalizes heavily negative depths, making it possible to initialize from random starting solutions (unlike bundle adjustment), and 2) does not penalize large depths like pOSE while it still counterbalances the shrinking bias of the OSE, since it will try to make the depths as large as possible. A quadratic approximation of the exponential regularization is also formulated, making it suitable to be optimized with VarPro. We show that expOSE indeed results in more accurate factorizations than pOSE while keeping a similar basin of convergence, and it even challenges the accuracy obtained after refinement with bundle adjustment in most benchmark datasets.

Additionally, we propose a generalization of the OSE error in RpOSE (Paper B), which consists of a decomposition of the original OSE in [1] into radial and tangential error components, weighted by α and $1 - \alpha$, respectively. For $\alpha = 1$ the OSE of RpOSE is recovered, while $\alpha = 0.5$ makes the OSE equal to the one presented in pOSE. This decomposition allows a trade-off between the accuracy and stability of the algorithm for sequences with radial distortion since, as mentioned in Section 2.2, the 1D radial camera model drops part of the data by ignoring errors along the radial direction, potentially leading to ill-posed problems (or close to).

The main contributions of the paper can be summarized as: 1) we investigate the pOSE models' undesirable penalization of large depths and propose expOSE; 2) we formulate a quadratic approximation of the exponential regularization term in expOSE to make it suitable for optimization with VarPro and show that, with random initialization, the model achieves convergence rates similar to pOSE with significantly higher reconstruction quality; and 3) we extend expOSE with radial distortion robustness by decomposing the Object Space Error (OSE) into radial and tangent components, and, just like with RpOSE in Paper B, propose an SfM pipeline that is able to obtain a complete and accurate Euclidean reconstruction from uncalibrated cameras.

4.4 Paper D - "Accurate Optimization of Weighted Nuclear Norm for Non-Rigid Structure-from-Motion"

In Section 3.5 we looked into how to use nuclear norm regularization in order to obtain a low-rank approximation $\mathbf{X}$ of a matrix $\mathbf{M}$. In particular, we saw that instead of minimizing directly over the elements of $\mathbf{X}$, we can minimize over factors $\mathbf{B}$ and $\mathbf{C}$ such that $\mathbf{X} = \mathbf{B}\mathbf{C}^T$ and replace the sum of singular values of $\mathbf{X}$ by $\min_{\mathbf{X}=\mathbf{B}\mathbf{C}^T} \frac{\|\mathbf{B}\|_F^2+\|\mathbf{C}\|_F^2}{2}$. First-order optimization methods are used when minimizing directly over the elements of $\mathbf{X}$, but it is well known [67] that these can have slow convergence near the minimum due to zig-zagging between level sets, and usually result in approximate solutions. By using the factors $\mathbf{B}$ and $\mathbf{C}$, we end up with a problem formulation suitable to be minimized with second-order optimization methods such as Levenberg-Marquardt or VarPro, which have better convergence properties near minima.

In the context of Structure-from-Motion, it has been shown [72] that nuclear norm regularization is not strong enough under the presence of noise. Kumar *et al.* [66] propose the use of weighted nuclear norm as a regularization on the (re-arranged) 3D structure $\mathbf{X}^\sharp$, introduced in Section 3.5. The authors claim that by weighting each singular value of $\mathbf{X}^\sharp$ differently it is possible to better capture the nature of the low dimensional space of the structure. Additionally, they proposed a method to initialize the weights of each singular value and solve the problem using ADMM.

In this paper, we propose a pOSE factorization model with weighted nuclear norm regularization that can be solved with second-order optimization methods. In particular, we show that the problems

$$\underset{\mathbf{Z}}{\text{minimize}} \quad \|\mathcal{A}(\mathbf{Z}) - \mathbf{b}\|^2 + \sum_l a_l \sigma_l(\mathbf{Z}) \tag{4.7}$$

and

$$\underset{\mathbf{B},\mathbf{C}}{\text{minimize}} \quad \|\mathcal{A}(\mathbf{B}\mathbf{C}^T) - \mathbf{b}\|^2 + \sum_l a_l \frac{\|\mathbf{B}_l\|^2 + \|\mathbf{C}_l\|^2}{2} \tag{4.8}$$

are equivalent, where $\mathbf{B}_l$ and $\mathbf{C}_l$ are the lth columns of $\mathbf{B}$ and $\mathbf{C}$, respectively, and a_l is the weights attributed to the lth largest singular value of $\mathbf{B}\mathbf{C}^T$, with $a_1 \leq a_2 \leq \dots$. We apply (4.8) to Non-Rigid Structure-from-Motion using VarPro and indeed show that using second-order optimization methods results in more accurate reconstructions, also outperforming other state-of-the-art factorization methods [62], [66].

When I joined this project the main theorems of the paper were already proved by the other authors. My contributions consisted of studying how to apply these results to Non-Rigid Structure-from-Motion and performing all experiments and evaluations, as well as writing the paper along with the other authors.

In conclusion, the main contributions of the paper are: 1) we show that the problems (4.7) and (4.8) are equivalent, and the latter can be efficiently optimized using second-order optimization methods; 2) we show that our proposed method outperforms (4.7) solved with ADMM in the context of Structure-from-Motion; and 3) comparison with other state-of-the-art factorization methods for Non-Rigid Structure-from-Motion in some benchmark datasets, with the proposed method outperforming them in terms of accuracy of the obtained reconstruction.

4.5 Paper E - "Bilinear Parameterization for Non-Separable Singular Value Penalties"

In this paper, we extend the regularization of the pOSE methods proposed in Paper D to a wider range of low-rank inducing penalties. In particular, we consider non-separable objectives of the form

$$f_h(\mathbf{Z}) = \|\mathcal{A}(\mathbf{Z}) - \mathbf{b}\|^2 + h(\sigma(\mathbf{Z})) \tag{4.9}$$

with

$$h(\sigma(\mathbf{Z})) = \sum_{l=1}^{\text{rank}(\mathbf{Z})} a_l \sigma_l(\mathbf{Z}) + b_l \tag{4.10}$$

and where it is assumed that the weights a_l and b_l are non-decreasing. Note that for $b_l = 0, \forall l$ the problem reduces to weighted nuclear norm. The regularization presented in (4.10) is not twice differentiable, so a relaxation $r_h(\sigma(\mathbf{Z}))$ is proposed by [78], which corresponds to the quadratic envelop [79] of $h(\sigma(\mathbf{Z}))$. Similarly to paper D, we prove the equivalence of the resulting problems

$$\underset{\mathbf{Z}}{\text{minimize}} \quad \|\mathcal{A}(\mathbf{Z}) - \mathbf{b}\|^2 + r_h(\sigma(\mathbf{Z})) \tag{4.11}$$

and

$$\underset{\mathbf{B},\mathbf{C}}{\text{minimize}} \quad \|\mathcal{A}(\mathbf{B}\mathbf{C}^T) - \mathbf{b}\|^2 + r_h(\gamma(\mathbf{B},\mathbf{C})) \tag{4.12}$$

where $\gamma_l(\mathbf{B},\mathbf{C})) = \frac{\|\mathbf{B}_l\|^2+\|\mathbf{C}_l\|^2}{2}$. In order to be apple to apply second-order methods, VarPro in particular, in each iteration of the optimization the regularization $r_h(\gamma(\mathbf{B},\mathbf{C}))$ is approximated by a quadratic form

$$r_h(\gamma(\mathbf{B}^t,\mathbf{C}^t)) \approx \sum_l w_l^t \frac{\|\mathbf{B}_l^t\|^2 + \|\mathbf{C}_l^t\|^2}{2} \tag{4.13}$$

where w_l^t can be efficiently computed at iteration t based on the current solutions $\mathbf{B}^t$ and $\mathbf{C}^t$ as shown in [78]. The motivation to use second-order methods is similar to the one presented in Paper D. The proposed method is applied to matrix factorization and Non-Rigid Structure-from-Motion, outperforming state-of-the-art factorization methods that rely on first-order optimization methods.

Contrary to other papers, where I had an active role in the works' idea discussion, implementation, and writing of the paper, in this work, I was mainly responsible for the experimental section of the paper.

In summary, the main contributions of this paper consist of 1) we show that (4.11) and (4.12) are equivalent; 2) we propose a quadratic approximation to (4.12) that make the problem suitable for optimization with second-order methods; and 3) we propose solving the proposed objective with a variation of VarPro that outperforms state-of-the-art methods for matrix factorization.

CHAPTER 5

Concluding Remarks and Future Work

5.1 Conclusions

In this work we investigated ways to increase the understanding, accuracy, and reliability of several different instances of Global Structure-from-Motion problems. Starting with point set registration (Paper A), we propose global optimality conditions based on Lagrange duality that allows certifying a candidate solution as globally optimal by evaluating bounds that depend on input data properties such as missing data, target point cloud spatial distribution, and noise on the source point clouds. Besides being able to certify a candidate solution as globally optimal, this work also allows us to better understand how these input data properties affect the tightness of the duality gap, and consequently the solvability of the problem with Semidefinite Programming. Regarding factorization-based Structure-from-Motion, we propose how to extend pOSE model for radial distortion invariance (Paper B) and non-rigidity (Paper D and E). Additionally, we propose to replace the regularization term of the original pOSE model with an exponential term (Paper C) that penalizes less large depths than the original formulation, resulting in an improvement in the accuracy of the estimated reconstruction. Altogether we increased the range of possible applications and accuracy of pOSE models without having to compromise their wide basin of convergence which makes them so attractive in the first place.

It is also worth mentioning that, despite the contributions of the proposed methods in this thesis and other concurrent work published over the last few years, Structure-from-Motion in general is still far from being considered a solved problem. One of the main issues with factorization-based SfM methods is their inability to deal with outliers in the input 2D point tracks. Conventional methods overcome this issue to a larger degree by estimating first pairwise relations between pairs of images with RANSAC. By using 2D point tracks as input data, factorization-based methods end up being much more sensitive to outliers, especially when combined with high levels of missing data. Furthermore, a better understanding of the solvability and stability of these methods based on the input data is needed as most results are proved empirically. For instance, it is well known that high levels of structured missing data can affect the convergence (algorithm getting stuck on local minima) or even solvability of the problem (many valid solutions might be possible to obtain due to the problem being ill-posed). However, in general, there is no clear way to measure this or to guarantee global optimality of the obtained solution.

In the next section, I will talk more in detail about some of these issues, as well as how to possibly incorporate learning-based models in order to speed up and better generalize Structure-from-Motion methods.

5.2 Future work

Dealing with outliers

When dealing with outliers, I see two main types of approaches to be considered when trying to solve Structure-from-Motion with factorization-based methods like pOSE or its variants. The first one would be to consider explicitly quantifying the uncertainty of the 2D point tracks. In this case, besides the coordinates of the 2D points in each image, we can assume that we also have available a variable that quantifies the probability (based on some prior) that the 2D detection corresponds to an inlier. For instance, for the OSE, the loss can be written as

$$\ell = \sum_{ij} w_{ij} \left\| \mathbf{z}_{ij}^{(3)} \mathbf{x}_{ij} - \mathbf{z}_{ij}^{(1:2)} \right\|^2 \tag{5.1}$$

where now w_{ij} can take values between 0 (high uncertainty) and 1 (low uncertainty). Note that any other quadratic loss can be used instead of OSE since the weights can be moved inside the squared norm as

$$\ell = \sum_{ij} \left\| \sqrt{w_{ij}} \left(\mathbf{z}_{ij}^{(3)} \mathbf{x}_{ij} - \mathbf{z}_{ij}^{(1:2)} \right) \right\|^2 = \left\| \mathcal{A}(\mathbf{Z}) - \mathbf{b} \right\|^2 . \tag{5.2}$$

In this way, the problem can still be written in quadratic form and solved using VarPro, as done with the pOSE methods. The problem with this approach consists of how to properly measure the uncertainty of 2D correspondences if no good prior knowledge of the scene is available. In this direction, some previous work estimated location uncertainty for SIFT and SURF features [80], while more recent approaches like LoFTR [20] output point correspondences along with their uncertainty. Even though these uncertainties can be incorporated into SfM methods as shown in (5.2), their uncertainty estimations tend to be suboptimal for most sequences and require some type of scene-specific refinement.

A second approach would be to replace the loss with a robust function that penalizes large residuals less than the squared norm of residuals that have been considered. Some examples of robust loss functions include Truncated Least Squares (TLS), L1-norm, Huber norm [81], and many other robust kernels. Replacing the squared norm with any of these robust losses will result in a non-quadratic loss, hence second-order methods like VarPro used in most of the works in this thesis would not be applicable anymore. A possible solution is to use Iterated Reweighted Least Squares (IRLS) [82], which approximates the loss with a quadratic formula in each iteration of the optimization problem. For the OSE, for instance, IRLS would result in a loss at iteration t of the form

$$\ell^t = \sum_{ij} w_{ij}^t \left\| \mathbf{z}_{ij}^{(3)} \mathbf{x}_{ij} - \mathbf{z}_{ij}^{(1:2)} \right\|^2 \tag{5.3}$$

where w_{ij}^t will depend on the robust loss function considered and on the solution $\mathbf{z}_{ij}^{t-1}$. Depending on the robust loss function, IRLS can introduce undesirable local minima which obviously would go against one of the main benefits of pOSE-like models - its wide basin of convergence. The convergence of robust pOSE methods with IRLS or with some more recent approaches [83], [84] is something that still requires additional study and research.

In summary, figuring out better ways to incorporate uncertainties and robust loss functions into Structure-from-Motion methods is an interesting research direction that can substantially increase their usability and reliability in large-scale applications.

Solvability and Global Optimality

Understanding the solvability of Structure-from-Motion problems is a hard problem. The major sources of complexity come from missing data, measurement noise, and the scene's spatial distribution, which under specific conditions might make the problem ill-posed. Having a sense of those conditions would help us get a better understanding of, for instance, the likelihood of getting stuck on local minima and design better algorithms that can avoid them, or modify the input data (e.g. add more points/views, break the sequence into smaller and more stable subsequences) in order to avoid ill-posed configurations. Some recent works [85], [86] model the uncertainty of cameras estimations of bundle adjustment through spectral analysis of the inverse of the Hessian (or an approximation of it) of the objective function. Providing estimations and their corresponding uncertainties in an accurate way would extend the possible applications of Structure-from-Motion methods, and additionally, from these uncertainties one can get a better understanding of how certain input configurations affect the final reconstruction. However, no study or theoretical analysis of these effects has been published, to the best of my knowledge. Further research, possibly extending these results to pOSE-like models, is a possibility for future work and makes an interesting research direction.

Incorporating learning-based methods

As mentioned in the introduction chapter, learning-based methods have revolutionized computer vision over the last decade. Even though learning-based methods for 3D reconstruction and pose estimation still underperform conventional methods, there are, in my opinion, many ways in which the former can be integrated with the latter.

A first way consists of learning meaningful point correspondences, similar to the learning-based methods described in Section 2.3. When learning correspondences through a Structure-from-Motion pipeline, one could additionally use ground-truth camera poses to train correspondence networks, which would consist of a stronger signal than the photometric loss (especially in low texture scenes) and cheaper to obtain than ground-truth 2D correspondences. Furthermore, with this approach one can also learn meaningful uncertainties for the point tracks in the context of Structure-from-Motion and consequently make the method more robust to outliers, as previously discussed. A possible issue with this approach is that current SfM methods are too computationally heavy to be integrated into an end-to-end learned model, as training would become significantly slower.

Building on the reasoning of the previous paragraph, a second way to incorporate learning-based models in SfM methods is to replace part of (or the whole) conventional methods with a neural network. Examples of works that aim at exactly that for Rigid SfM are, for instance, [87], [88]. In [87], the authors propose an end-to-end approach for camera pose and depth estimation from multiple source views and one target view. For each source view, its corresponding image is fed to a feature extraction network. These features are then combined with the features of the target view in order to generate a pose cost volume and a depth cost volume. The pose and depth cost volumes are then fed to two other networks, from which one obtains relative camera poses (w.r.t. the target view) and a (dense) depth map, respectively. Alternatively, in [88] the authors take an approach more similar to conventional SfM by taking as inputs the point tracks through many views. The point tracks are fed to permutation-equivariant layers (i.e. permutation of columns and/or row of the measurement matrix $\mathbf{M}$ in (3.6) do not affect the final reconstruction). The output of the permutation-equivariant layers is then used to regress camera poses and 3D point coordinates, and reprojection error is used as training loss. Even though these methods still do not compete with conventional methods in terms of accuracy, they have must faster inference times and also are slowly reducing the performance gap, showing that Structure-from-Motion can definitely benefit from learning components. Regarding Non-Rigid SfM, a recent report [89] covers most of the state-of-the-art methods in non-rigid 3D reconstruction, some of them being learning-based. In non-rigid reconstruction, having prior knowledge about the object can help solve ambiguities related to the motion of the camera/object, or constraint the type of deformations possible. As one sequence might not contain enough information to solve all these ambiguities, learning-based models trained from large quantities of data might serve as a strong enough prior to a large class of objects.

References

[1] J. Hyeong Hong and C. Zach, “Pose: Pseudo object space error for initialization-free bundle adjustment,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] R. Szeliski. “Computer vision algorithms and applications.” (2011), [Online]. Available: `http://dx.doi.org/10.1007/978-1-84882-935-0`.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. USA: Cambridge University Press, 2003, ISBN: 0521540518.

[4] J. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon, “Motion estimation by decoupling rotation and translation in catadioptric vision,” *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 254–273, 2010, Special issue on Omnidirectional Vision, Camera Networks and Non-conventional Cameras, ISSN: 1077-3142.

[5] J. Bazin, C. Demonceaux, P. Vasseur, and I.-S. Kweon, “Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment,” *I. J. Robotic Res.*, vol. 31, pp. 63–81, Jan. 2012.

[6] B. Streckel, J.-F. Evers-Senne, and R. Koch, “Lens model selection for a markerless ar tracking system,” in *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’05)*, 2005, pp. 130–133.

[7] V. Larsson, N. Zobernig, K. Taskin, and M. Pellefeys, “Calibration-free structure-from-motion with calibrated radial trifocal tensors,” in *European Conference of Computer Vision*, 2020.

[8] V. Larsson, T. Sattler, Z. Kukelova, and M. Pollefeys, "Revisiting radial distortion absolute pose," in *International Conference on Computer Vision (ICCV)*, IEEE, 2019.

[9] A. E. Conrady, "Decentred Lens-Systems," *Monthly Notices of the Royal Astronomical Society*, vol. 79, no. 5, pp. 384–390, Mar. 1919, ISSN: 0035-8711.

[10] D. Brown, "Decentering distortion of lenses," 1966.

[11] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I.

[12] S. Thirthala and M. Pollefeys, "Radial multi-focal tensors," *International Journal of Computer Vision - IJCV*, vol. 96, Jun. 2012.

[13] O. Enqvist, F. Kahl, and C. Olsson, "Non-sequential structure from motion," in *International Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, 2011.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, ISSN: 1573-1405.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417, ISBN: 978-3-540-33833-8.

[16] D. Viswanathan, "Features from accelerated segment test (fast)," 2011.

[17] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[18] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 467–483, ISBN: 978-3-319-46466-4.

[19] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–33 712.

[20] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “LoFTR: Detector-free local feature matching with transformers,” *CVPR*, 2021.

[21] E. Fix and J. L. Hodges, “Discriminatory analysis - nonparametric discrimination: Consistency properties,” *International Statistical Review*, vol. 57, p. 238, 1989.

[22] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[23] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4937–4946.

[24] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation,” in *European Conf. on Computer Vision (ECCV)*, A. Fitzgibbon et al. (Eds.), Ed., ser. Part IV, LNCS 7577, Springer-Verlag, Oct. 2012, pp. 611–625.

[25] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[26] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981, ISSN: 0004-3702.

[27] A. Dosovitskiy, P. Fischer, E. Ilg, *et al.*, “Flownet: Learning optical flow with convolutional networks,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[28] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow (extended abstract),” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed., Sister Conferences Best Papers, International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 4839–4843.

[29] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” 2018.

[30] M. Zhai, X. Xiang, N. Lv, and X. Kong, “Optical flow and scene flow estimation: A survey,” *Pattern Recognition*, vol. 114, p. 107 861, 2021, ISSN: 0031-3203.

[31] B. Triggs, “Autocalibration and the absolute quadric,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 609–614.

[32] “Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry,” *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 2, pp. 103–107, 2015, ISSN: 0099-1112.

[33] L. Quan and Z. Lan, “Linear n-point camera pose determination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[34] X. Gao, X. Hou, J. Tang, and H.-F. Cheng, “Complete solution classification for the perspective-three-point problem,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 930–943, 2003.

[35] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis,” in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ser. ICCV ’99, Springer-Verlag, 2000, pp. 298–372.

[36] K. Konolige, “Sparse sparse bundle adjustment,” Jan. 2010, pp. 1–11.

[37] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 756–770, 2004.

[38] R. Hartley, “In defense of the eight-point algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[39] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 0001-0782.

[40] V. Govindu, “Combining two-view constraints for motion estimation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[41] D. Martinec and T. Pajdla, “Robust rotation and translation estimation in multiview reconstruction,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[42] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri, “Global motion estimation from point matches,” in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012.

[43] A. P. Eriksson, C. Olsson, F. Kahl, and T. Chin, “Rotation averaging and strong duality,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 127–135.

[44] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, “Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4597–4604.

[45] A. Chatterjee and V. M. Govindu, “Efficient and robust large-scale rotation averaging,” in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 521–528.

[46] J. Fredriksson and C. Olsson, “Simultaneous multiple rotation averaging using Lagrangian duality,” in *Asian Conference on Computer Vision*, 2012.

[47] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, “Building rome in a day,” in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 72–79.

[48] S. Zhu, R. Zhang, L. Zhou, *et al.*, “Very large-scale global sfm by distributed motion averaging,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4568–4577.

[49] F. Kahl, “Multiple view geometry and the l-infinity norm,” vol. 2, Nov. 2005, 1002–1009 Vol. 2, ISBN: 0-7695-2334-X.

[50] D. Martinec and T. Pajdla, “3d reconstruction by gluing pair-wise euclidean reconstructions, or "how to achieve a good reconstruction from bad images",” *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06)*, pp. 25–32, 2006.

[51] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: Exploring photo collections in 3d,” in *ACM siggraph 2006 papers*, 2006, pp. 835–846.

[52] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: A factorization method,” *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[54] P. F. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Proceedings of the 4th European Conference on Computer Vision-Volume II - Volume II*, ser. ECCV '96, Berlin, Heidelberg: Springer-Verlag, 1996, pp. 709–720, ISBN: 3540611231.

[55] A. Heyden, "Projective structure and motion from image sequences using subspace methods," eng, in *Proceedings of the 10th Scandinavian Conference on Image Analysis*, M. Frydrych, J. Parkkinen, and A. Visa, Eds., 1997, pp. 963–968, ISBN: 951-764-145-1.

[56] B. Triggs, "Factorization methods for projective structure and motion," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 845–851.

[57] A. D. Bue, J. M. F. Xavier, L. Agapito, and M. Paladini, "Bilinear modeling via augmented lagrange multipliers (BALM)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1496–1508, 2012.

[58] Y. Dai, H. Li, and M. He, "Projective multiview structure and motion from element-wise factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2238–2251, 2013.

[59] J. H. Hong, C. Zach, and A. Fitzgibbon, "Revisiting the variable projection method for separable nonlinear least squares problems," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5939–5947.

[60] C. Russell, J. Fayad, and L. Agapito, "Energy based multiple model fitting for non-rigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2011, pp. 3009–3016.

[61] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[62] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 101–122, 2014.

[63] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2117–2130, 2013.

[64] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *International Journal of Computer Vision*, vol. 121, Jul. 2016.

[65] T. H. Oh, Y. W. Tai, J. C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust pca: Algorithm and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 744–758, 2016.

[66] S. Kumar, "Non-rigid structure from motion: Prior-free factorization method revisited," in *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, IEEE, 2020, pp. 51–60.

[67] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[68] A. S. Lewis, "The convex analysis of unitarily invariant matrix functions," *Journal of Convex Analysis*, vol. 2, no. 1, pp. 173–183, 1995.

[69] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.

[70] L. Canyi, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[71] V. Larsson and C. Olsson, "Convex low rank approximation," *International Journal of Computer Vision*, vol. 120, no. 2, pp. 194–214, 2016.

[72] M. V. Ornhag, C. Olsson, and A. Heyden, "Bilinear parameterization for differentiable rank-regularization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[73] H. H. Bauschke, P. L. Combettes, *et al.*, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2017, vol. 2011.

[74] B. D. Haeffele and R. Vidal, "Structured low-rank matrix factorization: Global optimality, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1468–1482, 2020.

[75] F. Bach, "Convex relaxations of structured matrix factorizations," Sep. 2013.

[76] K. N. Chaudhury, Y. Khoo, and A. Singer, “Global registration of multiple point clouds using semidefinite programming,” *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 468–501, 2015.

[77] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, Mar. 1996, ISSN: 0036-1445.

[78] M. Valtonen Örnhag and C. Olsson, “A unified optimization framework for low-rank inducing penalties,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[79] M. Carlsson, “On convex envelopes and regularization of non-convex functionals without moving global minima,” *Journal of Optimization Theory and Applications, to appear*, 2019.

[80] B. Zeisl, P. Georgel, F. Schweiger, E. Steinbach, N. Navab, and G. Munich, “Estimation of location uncertainty for scale invariant feature points,” *Proceedings of the British machine vision conference*, Jan. 2009.

[81] P. J. Huber, “Robust Estimation of a Location Parameter,” *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[82] P. J. Green, “Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 46, no. 2, pp. 149–192, 1984, ISSN: 00359246.

[83] C. Zach and G. Bourmaud, “Descending, lifting or smoothing: Secrets of robust cost optimization,” in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 558–574, ISBN: 978-3-030-01258-8.

[84] C. Zach and G. Bourmaud, “Pareto meets huber: Efficiently avoiding poor minima in robust estimation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[85] M. Polic, W. Forstner, and T. Pajdla, “Fast and accurate camera covariance computation for large 3d reconstruction,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.

[86] K. Wilson and S. Wehrwein, “Visualizing spectral bundle adjustment uncertainty,” in *2020 International Conference on 3D Vision (3DV)*, 2020, pp. 663–671.

[87] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue, "Deepsfm: Structure from motion via deep bundle adjustment," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 230–247, ISBN: 978-3-030-58452-8.

[88] D. Moran, H. Koslowsky, Y. Kasten, H. Maron, M. Galun, and R. Basri, "Deep permutation equivariant structure from motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 5976–5986.

[89] E. Tretschk, N. Kairanda, M. B. R, *et al.*, *State of the art in dense monocular non-rigid 3d reconstruction*, 2023.