THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Geometry and Learning in 3D Computer Vision

Yaroslava Lochman

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2025

Geometry and Learning in 3D Computer Vision

Yaroslava Lochman
ISBN 978-91-8103-309-0

Acknowledgements, dedications, and similar personal statements in this thesis, reflect the author's own views.

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5766
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 Digital Printing
Gothenburg, Sweden, November 2025

To my family.

Geometry and Learning in 3D Computer Vision

Yaroslava Lochman
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis focuses on studying and improving the accuracy, reliability, and efficiency of 3D vision pipelines. We leverage techniques from geometry, optimization, and deep machine learning, and we also try to explore and understand when it is suitable to combine them and when it is not, if the overall success of a 3D reconstruction system is a priority. In modern computer vision, deep neural networks are often utilized as black boxes, not only for perception but also for solving geometric problems. The performance is highly dependent on the amount and quality of the data, and the results can sometimes be surprisingly poor. Classic geometric models and optimization techniques in 3D vision are much better understood. While they are still preferred in many applications, the learning-based counterparts showcase an amazing improvement over traditional methods on certain challenging tasks.

The thesis is structured around three problems: (1) camera calibration, (2) rotation averaging, and (3) motion segmentation. For each of these problems, we analyze the weak points and failure modes of existing methods and propose new algorithms that leverage standard techniques from geometry and optimization or hybrid learning pipelines that aim to retain the interpretability of geometric models while benefiting from the expressivity and adaptability of deep neural networks.

Our contributions include: (i) a versatile pipeline for calibrating central cameras with various lens configurations that relies on simple techniques and solvers and proves to be very stable, (ii) a semidefinite program for anisotropic rotation averaging that leverages the readily-available uncertainties of the relative estimates and relies on a new convex relaxation, leading to improved reconstruction accuracy, (iii) a fast block-coordinate descent solver for anisotropic rotation averaging that achieves similar reconstruction accuracy while significantly reducing the runtime, (iv) robustification pipelines for anisotropic rotation averaging allowing gross outliers in the data, and (v) a metric learning approach addressing the challenging chicken-and-egg prob-

lem of motion segmentation via clustering in the space of trajectory feature representations, where inference is done in a fraction of a second.

Keywords: Computer vision, 3D reconstruction, camera calibration, minimal solvers, rotation averaging, global structure from motion, robust optimization, motion segmentation, trajectory clustering.

List of Publications

This thesis is based on the following publications:

[A] **Yaroslava Lochman**, Kostiantyn Liepieshov, Jianhui Chen, Michal Perdoch, Christopher Zach, James Pritts, "BabelCalib: A Universal Approach to Calibrating Central Cameras". ICCV 2021.

[B] Carl Olsson, **Yaroslava Lochman**, Johan Malmport, Christopher Zach, "Certifiably Optimal Anisotropic Rotation Averaging ". ICCV 2025.

[C] **Yaroslava Lochman**, Carl Olsson, Christopher Zach, "Fast and Robust Rotation Averaging with Anisotropic Coordinate Descent ". BMVC 2025.

[D] **Yaroslava Lochman**, Carl Olsson, Christopher Zach, "Learned Trajectory Embedding for Subspace Clustering". CVPR 2024.

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

[E] Xixi Liu, **Yaroslava Lochman**, Christopher Zach, "GEN: Pushing the Limits of Softmax-Based Out-of-Distribution Detection". CVPR 2023.

Contents

Acknowledgments

There is something truly special about being part of the research and engineering community. Being surrounded by curious people who embrace uncertainties as opportunities and strive to think outside the box and to make an impact. 5 years ago, I moved to Gothenburg to join Chalmers and WASP—the two anchor points for me to meet many talented and inspiring people. I am grateful for getting to know each one of you and sharing experiences such as doing research, studying, visiting institutions, and attending conferences together. I would particularly like to express my deepest gratitude to the most influential people. First of all, I would like to thank my supervisor, Christopher Zach, for supporting me throughout my PhD journey and helping me grow. It has been a pleasure to be part of our engaging discussions, sharing ideas, and getting your feedback and encouragement. As well as chatting about random interesting topics during breaks. I would like to equally thank my co-supervisor, Carl Olsson, for your engagement, invaluable feedback, and mathematical insights. It has been an honor to work with both of you, and I have learned so much from you. Thank you, Christopher and Carl. I am also grateful to my collaborators, with whom I had the pleasure of working on various projects and from whom I have learned as well. To the brilliant people at Chalmers, past and present members of the computer vision group: Xixi, Rasmus, Sophia, Georg B., Lucas, José, Alex, Huu, Kunal, Fredrik, Jennifer, Ida, Roman, Josef, Erik L., Victor, Sofie, both Davids, Tianyu, Bernardo, Richard, Jorge, Vilgot; and the neighboring signal processing group: Lennart, Jakob, Ji, Georg H., Erik W., Lars, David H., Adam, Carl L., Mahan. I am grateful to have been working alongside you. It has also been really nice to get to know you and do fun activities together. A special thanks to Ida for helping with the PhD defense organization. I was also lucky to meet other talented researchers, students, and friends at E2: Laura, Mattia, Moein, Rita, Gabriel, Apichai, Patrik, Alvin, Stephie, Nishant, Rikard, Alejandra, Samuel, Sabino, Ben, Kilian, Ying, Chiara, Nanami, Zhitao, and so many more. Thank you for making our workspace feel like home.

This journey would not have been possible without the love and support of my family and friends. I am deeply grateful to my partner André. I can't find the right words to express how lucky I feel to have met you. Thank you for all your support and for sharing your life with me. Love you very much. I am also grateful to my mother, Iana. Дякую за твою нескінченну підтримку,

мудрість і силу. Люблю тебе сильно. (this was in ukrainian) To my dear Gothenburg friends, thank you all for the great memories. You're making my life in a new country much less intimidating and much more vibrant. Some of you are expats too, and sharing our common experiences has been invaluable for me in establishing a new home. To my dear Ukrainian friends, whom I miss a lot: Валік, Антон, Маркіян, Богдан, Вова М., Вова Л. та ін. Дякую за всі моменти, пережиті разом. Радію, що познайомилась з вами.

Finally, I would like to thank everyone who expressed their support to me and my home country Ukraine that has been going through a lot.

Yaroslava Lochman
Gothenburg, November 2025

This work was supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Acronyms

BA:	Bundle Adjustment
CS:	Coordinate System
DoF:	Degrees of Freedom
FOV:	Field of View
ILRS:	Iteratively Reweighted Least Squares
MLE:	Maximum Likelihood Estimate
MLP:	Multilayer perceptron
NCE:	Noise-Contrastive Estimation
NN:	Neural Network
P3P:	Perspective-3-Point
RANSAC:	Random Sample Consensus
RCD:	Rotation Coordinate Descent
RGB:	Red Green Blue
RMS:	Root Mean Square
SDP:	Semidefinite Program
SfM:	Structure from Motion
SIFT:	Scale-Invariant Feature Transform
SLAM:	Simultaneous Localization And Mapping
SVD:	Singular Value Decomposition

Part I

Overview

CHAPTER 1

Introduction

Computer vision, a large subfield of computer science that deals with extracting useful information from images, is revolutionizing many different sectors. And, in particular, *3D computer vision*—a specialized area of computer vision that deals with inferring three-dimensional structure of the world from visual data, and a focus of this thesis—plays a major role in this by extracting geometric information and enabling physical interactions in space. Many exciting applications appear in, for example, healthcare and biotechnology, where vision-aided systems advance deep into the processes: robotics assistance in examination [1], [2], diagnostics [3]–[6], risk assessment, monitoring [7] and surgical assistance [8], [9]. In automotive and transportation industry, the technologies are predominantly focused on driving assistance [10]–[12], but also systems for traffic analysis and road safety. Manufacturing and assembling processes are heavily industrialized, where robotics and automation play a major role—there, 3D vision finds its applications in *e.g.* wire and cable harness assembly/disassembly [13], [14]. In agriculture, computer vision methods help with plant disease detection [15] and harvest monitoring [16]. New exciting applications keep emerging [17]–[19], and there is probably a lot more yet to appear. With that, it becomes a priority to solve computer vision

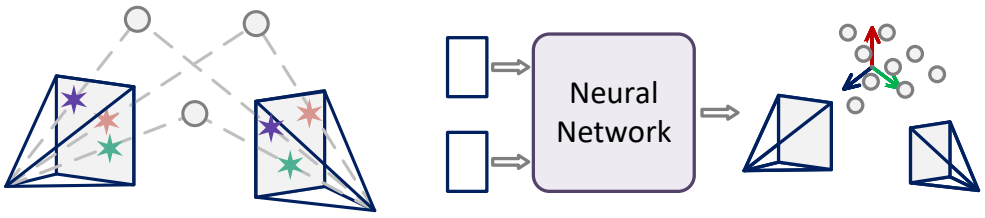
problems *reliably* and *efficiently*, especially considering many safety-critical applications. The real-world situations with many edge scenarios pose challenges in achieving such an ambitious goal, but the technological solutions are constantly evolving, pushed by the need to address these challenges.

1.1 Two sides of modern 3D computer vision

Up until around fifteen years ago, 3D vision systems primarily relied on multi-view geometry, signal processing, and optimization [20]–[25]. For instance, a major catalyst for computer vision research and applications was David Lowe's work on SIFT [26]—a highly distinctive, approximately scale- and rotation-invariant feature detector and descriptor. It enabled robust matching of images at the level of individual 2D points. Subsequent methods, such as ORB [27], offer a computationally efficient alternative, facilitating real-time processing. Alongside these developments, specialized solvers [28]–[30] were introduced for recovering 3D geometry from point matches. RANSAC [31] enabled robust model estimation. Techniques for optimizing 3D reconstruction, known as bundle adjustment [21], [32], address stability and computational feasibility of robust large-scale processing. Building on these advances, works like Bundler [33], ORB-SLAM [34] and COLMAP [35] provided practical, widely adopted pipelines for 3D reconstruction that remain influential in current research.

Then, *machine learning* and *deep machine learning* have revolutionized the field [36]–[39]. We refer to *e.g.* [40] for an excellent introduction to pattern recognition and machine learning, and *e.g.* [41], [42] that are focused on deep learning. *Neural networks* (NNs)—the key components of deep learning—are structured functions with optimizable parameters that map inputs to outputs, where we would like their output to closely resemble what we have in the data. The success of deep learning models is mainly affected by: (1) the NN architecture (*i.e.* the form of this function), (2) training (*i.e.* how are the parameters optimized), and (3) data quality and quantity. Some notable architectural examples are convolutional NNs [43], [44], graph NNs [45], and transformers [46]. A big step forward in facilitating NN optimization was backpropagation [43]. Recent works [39], [47] show that using standard models and optimizers with a carefully selected objective on extremely large and diverse datasets drastically improves performance compared to smaller datasets.

Figure 1.1: Two major paradigms in modern 3D computer vision: (left) traditional geometry-based approaches and (right) modern deep machine learning-based methods.



1.2 Is 3D reconstruction not solved yet?

Arguably, 3D reconstruction systems (that create digital three-dimensional scene models from images or other sensor data, but we focus on images) constitute a major component of 3D vision applications. We will go through the technical details of what it does, but a knowledgeable reader may ask now: is 3D reconstruction not solved yet? Indeed, various frameworks for 3D reconstruction have been proposed since long time ago [33], [35], [48]–[52]. While they have demonstrated tremendous progress, as mentioned earlier, there is still no guarantee of success in all scenarios, particularly edge cases—such a universal system does not exist yet. Although it is probably an unrealistic goal to have one (and it is unrealistic to assess that), we can always strive towards it. To be more specific, an edge case may, for instance, include images with drastic changes in fields of view, illumination and color temperature, time of the day; blurry, foggy, corrupted images, *etc.* Symmetric objects can also cause significant difficulties, as images of distinct sides may appear nearly identical—cannot blame the system though, even I sometimes get lost inside a new building where all corridors look the same. Similar problems arise for texture-less objects. Then, the use of cameras with high lens distortion poses challenges as well since these cameras are more tedious to calibrate. The presence of multiple objects moving independently introduces additional degrees of freedom and makes the optimization more challenging. These are just some examples, and more exist [53], [54]. Lastly, addressing efficiency can be as challenging when thinking about large-scale scenarios and real-time applications.

Research objectives and questions

The goal of this thesis is to take a small step towards more accurate, robust, and efficient solutions to 3D vision problems, where we think of 3D reconstruction as a main application. This thesis studies both traditional and modern methodologies (that are sketched in Figure 1.1). To approach this goal, we analyze some important components of 3D reconstruction systems that directly influence overall performance, identify current difficulties, and try to address them using the appropriate tools from a large modern 3D computer vision toolbox. This thesis, in particular, studies the topics of camera calibration, rotation averaging, and motion segmentation. While studying these problems, we formulated specific questions, namely:

- **Can we leverage existing classic techniques to address some failure cases in camera calibration and design a universal approach?** Well-established pipelines exist for camera calibration [55]–[59]. However, each one supports a limited set of models, and initialization can be a weak point causing failures, particularly for large field-of-view cameras.

- **Can we integrate pairwise estimation uncertainties into certifiably optimal rotation averaging to avoid potentially inaccurate solutions?** Relative rotation estimates obtained by optimizing two-view reconstruction objectives are known to follow an anisotropic noise model. It was also shown that taking this uncertainty into account when optimizing absolute rotations further is beneficial [60], but global optimization remains challenging.

- **Can we combine robust estimation with anisotropic inference in a meaningful way?** Robust anisotropic rotation averaging has been recently studied, demonstrating improved performance [60]. However, the underlying robust local optimization method is known to be sensitive to initialization [61]. Typical initialization is done incrementally, hence it is prone to error accumulation. Addressing both quality of initialization and the ability to escape poor local minima is of high interest.

- **Is it possible to turn a difficult chicken-and-egg problem of rigid motion segmentation into a more tractable approach via**

metric learning? Rigid motion segmentation has been studied extensively. Due to its chicken-and-egg nature, joint approaches are preferred, *i.e.* simultaneous optimization of both motion models and cluster assignments [62], [63]. However, a computationally efficient approach remained a challenge. Metric learning, on the other hand, is known to help find a useful transformation from data space to the representation space where clustering can be more tractable.

The answer to all of these questions turned out to be *yes*. We will now try to answer the corresponding questions of **"How to..."** in the remainder of this thesis.

1.3 Thesis outline

This thesis consists of two parts. The aim of Part I is to introduce the topics, formulate the research questions and the scope of the thesis, provide the bigger picture and motivation for the conducted work, offer some important background information and describe the core methodology. This Chapter covers the introduction and motivation. Chapter 2 introduces some mathematical concepts and tools that are most relevant to the thesis methodology. The knowledgeable reader can most likely skip this chapter. Chapter 3 focuses on the core contributions of the thesis within the context of 3D reconstruction, which serves as the primary application domain for the developed approaches. It begins with an overview of a typical 3D reconstruction pipeline, outlining its key components. It then introduces the specific methods and techniques proposed in this thesis, explaining how each integrates into the pipeline and contributes to addressing particular challenges. Chapter 4 summarizes the included papers and lists the author's specific contributions to each of these papers. Chapter 5 discusses the main findings, implications, and potential directions for future research. Part II consists of four research papers that form the basis of this thesis.

1.4 Notation

This section introduces some common notation used throughout the thesis. Minor variations may appear in the included papers.

In general, plain lowercase characters (x) are used to denote scalars, bold characters ($\mathbf{x}$, $\mathbf{X}$) are used to denote vectors, mono-spaced and sans-serif uppercase characters ($\mathtt{X}$, X) are used to denote matrices, where the latter ones (X) represent matrices that stack the former ones ($\mathtt{X}$) *e.g.* vertically or both vertically and horizontally. When writing explicit forms of vectors and matrices, round brackets are used for stacking the elements/coordinates, and square brackets are used for stacking the other vectors and matrices. $\mathtt{I}_3$ is the 3×3 identity matrix, I is the identity matrix whose dimensionality can be inferred from the context, or otherwise subscripts may be used. $\mathbf{0}/\mathtt{0}$ is the vector/matrix filled with zeros, and $\mathbf{1}/\mathtt{1}$—with ones, where similarly, subscripts may be used to describe dimensionalities.

Multi-view geometry A homogeneous representation of a (projected) 2D point is denoted as $\mathbf{x} \in \mathbb{P}^2$ or $\mathbf{u} \in \mathbb{P}^2$. Most commonly, $\mathbf{x}$ represents a sensor point (with world units), and $\mathbf{u}$ represents an image point (with pixel coordinates). An equivalence relation in the projective space which is equality up to scale is denoted as $\sim$. A Euclidean representation of $\mathbf{x}$ is written as $\underline{\mathbf{x}} \in \mathbb{R}^2$. The radius of the 2D point, *i.e.* its distance to the center of projection, is denoted as r. A 3D point (equivalently, a scene point) is denoted as $\mathbf{X} \in \mathbb{R}^3$, and its radial and depth components are denoted as R and Z, respectively. A point correspondence, denoted $\mathbf{a} \leftrightarrow \mathbf{b}$, is a 2-tuple, where $\mathbf{a}$ and $\mathbf{b}$ can be either points in 2D or 3D, and both are measurements of the same physical point in 3D space.

In camera geometry, the camera calibration matrix is denoted as $\mathtt{K} \in \mathbb{R}^{3\times3}$, and a projection function is in general denoted as $\pi(\cdot)$. In epipolar geometry, the fundamental matrix is $\mathtt{F} \in \mathbb{R}^{3\times3}$, and the epipolar matrix is $\mathtt{E} \in \mathbb{R}^{3\times3}$. For the relative camera pose, the 3D rotation matrix and 3D translation vector representing relative rotation and translation from the coordinate system of camera i to camera j are $\tilde{\mathtt{R}}_{ij} \in \mathrm{SO}(3)$ and $\tilde{\mathbf{t}}_{ij} \in \mathbb{R}^3$, respectively, where $\mathrm{SO}(3)$ is the *special orthogonal group*—a set of all 3D rotations, *i.e.*

$$\mathrm{SO}(3) = \{\mathtt{R} \in \mathbb{R}^{3\times3} : \mathtt{R}^\top\mathtt{R} = \mathtt{I}_3, \det(\mathtt{R}) = 1\}. \tag{1.1}$$

For the absolute camera pose, the 3D rotation matrix and 3D translation

vector representing camera rotation and translation are $\mathtt{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$, respectively. The set of all $3n \times 3$ matrices that stack n rotations vertically is denoted as

$$\mathrm{SO}(3)^n = \{\mathsf{R} = (\mathtt{R}_1^\top \ \mathtt{R}_2^\top \ \cdots \ \mathtt{R}_n^\top)^\top \ \mid \ \mathtt{R}_i \in \mathrm{SO}(3)\}. \tag{1.2}$$

We also denote the *orthogonal group* in 3D as $\mathrm{O}(3) = \{\mathtt{Q} \in \mathbb{R}^{3\times 3} : \mathtt{Q}^\top\mathtt{Q} = \mathtt{I}_3\}$, and similarly $\mathrm{O}(3)^n = \{\mathsf{Q} = (\mathtt{Q}_1^\top \ \mathtt{Q}_2^\top \ \cdots \ \mathtt{Q}_n^\top)^\top \ \mid \ \mathtt{Q}_i \in \mathrm{O}(3)\}$.

A *camera graph* or a *view graph* is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex $i \in \mathcal{V}$ represents an absolute pose $(\mathtt{R}_i, \mathbf{t}_i)$, and each edge $(i, j) \in \mathcal{E}$ represents an estimated two-view relation, *i.e.* $(i, j) \in \mathcal{E}$ if an estimate $(\tilde{\mathtt{R}}_{ij}, \tilde{\mathbf{t}}_{ij})$ of the relative pose exists.

Linear algebra and convex optimization The skew-symmetric matrix of a cross-product operation $\mathbf{a} \times \mathbf{b}$ is denoted as following

$$[\mathbf{a}]_\times = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}, \tag{1.3}$$

where $\mathbf{a} = (a_x, a_y, a_z)^\top$, and is such that such that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$. Kronecker product is denoted as $\otimes$. Element-wise product is denoted as $\odot$. A function that creates a block diagonal matrix from a set of matrices is denoted as $\mathrm{blkdiag}(\cdot, \ldots, \cdot)$, *e.g.*

$$\mathrm{blkdiag}(\mathtt{A}_1, \mathtt{A}_2, \mathtt{A}_3) = \begin{bmatrix} \mathtt{A}_1 & \mathtt{0} & \mathtt{0} \\ \mathtt{0} & \mathtt{A}_2 & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \mathtt{A}_3 \end{bmatrix}. \tag{1.4}$$

A point $\mathbf{y} = \sum_i \alpha_i \mathbf{x}_i$ is a *convex combination* of the points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ if $\sum_i \alpha_i = 1$ and $\alpha_i \geq 0$, $i = 1, \ldots, n$. A *convex hull* of a set $\mathcal{C}$, denoted as $\mathrm{conv}(\mathcal{C})$, is the set of all convex combinations of points in $\mathcal{C}$

$$\mathrm{conv}(\mathcal{C}) = \left\{ \sum_i \alpha_i \mathbf{x}_i \ \mid \ \mathbf{x}_i \in \mathcal{C}, \ \sum_i \alpha_i = 1, \ \alpha_i \geq 0, \ i = 1, \ldots, n \right\}. \tag{1.5}$$

CHAPTER 2

Background

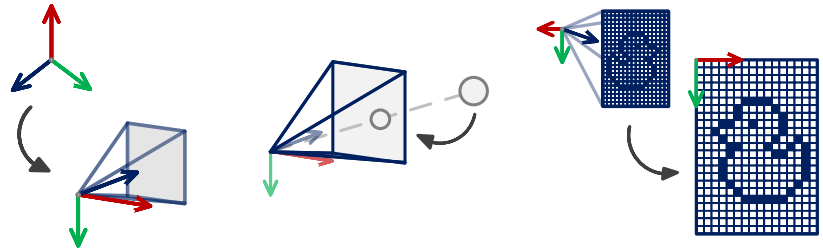
This chapter provides an overview of important tools and methodologies used throughout the thesis. It includes elements of geometry, optimization and deep machine learning used in the context of 3D computer vision.

2.1 Camera geometry

The fundamental concepts in understanding camera geometry are (1) camera pose, (2) sensor projection, and (3) sensor-to-image or pixel transformation. These three concepts are illustrated in Figure 2.1. Note that in this thesis, we only consider central cameras and subsequently central camera models, *i.e.* where a single *center of projection* (or *optical center*)—a point at which all rays of light intersect—is assumed.

A *camera pose* refers to the position and orientation of the camera in 3D relative to the world coordinate system (CS). However, often in the literature, and further in this thesis, a camera pose denotes a related entity pair: camera *rotation* and *translation* $(\mathtt{R}, \mathbf{t})$ such that $\mathbf{X} \mapsto \mathtt{R}\mathbf{X} + \mathbf{t}$ is a transformation from the world CS to the camera CS. One can therefore obtain a camera position $\mathbf{C}$ (that is also a center of projection) by noting that $\mathtt{R}\mathbf{C} + \mathbf{t} = \mathbf{0}$

Figure 2.1: Camera geometry: (left) camera pose, (center) sensor projection, and (right) sensor-to-image transformation.



giving $\mathbf{C} = -\mathtt{R}^\top \mathbf{t}$. Since the camera orientation is defined as a basis of the camera CS, it is obtained as $\mathtt{R}^\top$. It is common to refer to the camera pose(s) as *extrinsics*.

A *sensor projection* (or simply *projection*) is a mapping of a 3D point $\mathbf{X}$, with its coordinates in the camera CS, onto a 2D sensor array, where the output point $\mathbf{x}$ is referred to as a *sensor point* (or *retinal* point). The geometry of physical camera projection, where the rays of light pass through the lens and intersect the physical sensor, can be approximated very accurately [24] using various projection models of the form $\mathbf{x} = \pi(\mathbf{X})$. We discuss the most commonly used projection models in the next subsection. While physical projection happens behind the optical center producing a "flipped" image (both horizontally and vertically) which is unflipped during internal processing, for modeling purposes it is useful to imagine *a virtual plane* in front of the optical center, so that the rays intersecting this plane produce an equivalent resulting image (see Figure 2.1). Moreover, it is often convenient to assume that the virtual plane is at the distance equal to 1 from the optical center, which implies that the effects of the focal length, *i.e.* the actual distance from the optical center, are incorporated further in the sensor-to-image transformation (discussed in the next paragraph). Note also that during physical projection onto sensor, the output point coordinates may be adjusted due to discretization. Modeling rounding/quantization explicitly is in theory possible and straightforward. However, in practical 3D vision applications like 3D reconstruction it is omitted—introducing the step-functions into the gradient-based optimization would likely break the underlying framework. Instead, the image points of interest are usually assumed to be realizations of Gaussian random variables that encapsulate several types of noise including discretization noise.

A *sensor-to-image* or *pixel transformation* is a mapping of a point from the (virtual) sensor, where the units are world units, to a 2D image array, where the units are pixels. In alignment with the typical assumptions in computer vision literature, in this thesis the sensor CS is assumed to have an origin at the *principal point*—a point at which the camera's optical or Z-axis intersects the sensor plane—with the x-axis pointing right and the y-axis pointing down, while the image CS is assumed to have its origin at the top left corner of an image, similarly with the x-axis pointing right and the y-axis pointing down. The sensor-to-image mapping is formulated as $\mathbf{x} \mapsto \mathtt{K}\mathbf{x}$ where $\mathtt{K}$ is the *camera calibration matrix* having the following general form

$$\mathtt{K} = \begin{pmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.1}$$

To understand this transformation better, it is useful to decompose it *e.g.* as following

$$\mathtt{K} = \underbrace{\begin{pmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{pmatrix}}_{\text{image CS offset}} \underbrace{\begin{pmatrix} s_x & \alpha s_y & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{pixel scaling+skew}} \underbrace{\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{focal length scaling}}, \tag{2.2}$$

where f is the focal length (*e.g.* in mm), s_x and s_y are the scales to transform (from mm) to pixels, with α being the skew coefficient (for a common assumption of square pixels, $s_x = s_y$ and $\alpha = 0$), and p_x and p_y are the pixel coordinates of the principal point. Therefore $f_x = s_x f$ and $f_y = s_y f$ are the focal length values in pixels, and $s = \alpha s_y f$ is the skew coefficient in pixels. It is common to refer to the camera calibration matrix together with the projection model and its parameters as *intrinsics*.

Projection models

As mentioned earlier, the projection models are used to approximate the geometric transformation of 3D points as they are projected onto the 2D sensor array during the image formation process. This section presents an overview of the commonly used projection models. Let the 2D coordinates of the (homogeneous) sensor point $\mathbf{x}$ be $(x,\ y)^\top$, and $\mathbf{X}$ be the 3D point.

Pinhole projection

Pinhole projection, also known as *perspective* projection is a widely used camera model that assumes a single viewpoint (infinitesimally small aperture) through which light rays pass. It can be formulated as $\mathbf{x} \sim \mathbf{X}$ or $\lambda\mathbf{x} = \mathbf{X}$, where λ is the Z-component of the scene point $\mathbf{X}$, or its *depth*. The resulting 2D coordinates are obtained by dividing the X- and Y-components by the Z-component $(x,\ y)^\top = (X/Z,\ Y/Z)^\top$. The geometry of perspective projection is shown in the middle of Figure 2.1. This model approximates well pinhole cameras *i.e.* cameras with very small apertures and without a lens. It also often serves as a good starting point for modeling modern cameras that do not exhibit significant lens distortions.

Orthographic and affine projection

Orthographic projection has the following form

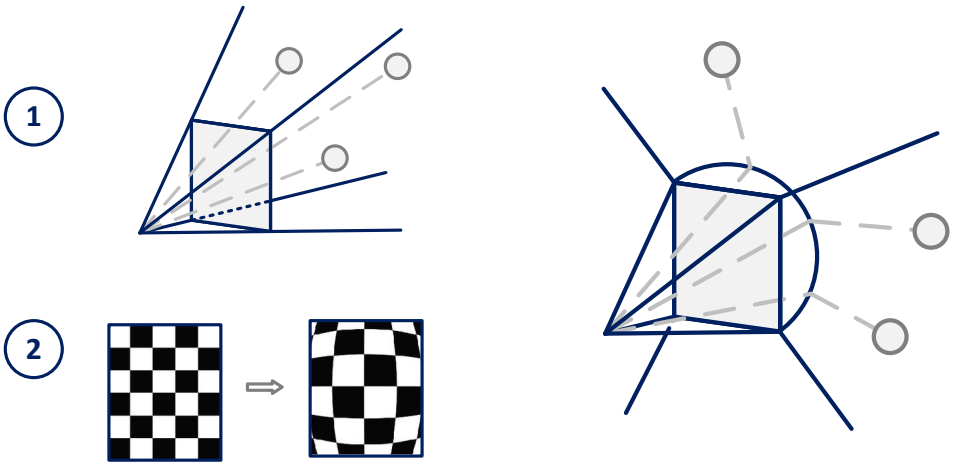
$$\mathbf{x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \tag{2.3}$$

resulting in the 2D coordinates equal to $\begin{pmatrix} x & y \end{pmatrix}^\top = \begin{pmatrix} X & Y \end{pmatrix}^\top$. It is a simplified approximation of the pinhole model that is computationally efficient and useful in scenarios where the depth variation within the scene is small relative to the distance from the camera. Orthographic projection is an instance of a more general *affine projection* $\begin{pmatrix} x & y \end{pmatrix}^\top = \mathtt{A}\mathbf{X} + \mathbf{b}$, where $\mathtt{A} \in \mathbb{R}^{2\times 3}$ ($\mathrm{rank}(\mathtt{A}) = 2$) and $\mathbf{b} \in \mathbb{R}^2$ are model parameters. Since affine cameras are linear functions (on inhomogeneous coordinates), the problem of reconstructing the 3D structure and camera geometry from 2D projections with affine model can be formulated as subspace fitting problem [64], [65] (see also Section 3.3).

Radial projections

Real-world lenses often introduce nonlinear distortions, especially near the image extent. Radial distortion and projection models account for these effects to better match the actual image formation in cameras with lenses of various configurations. Note that in this thesis, we think of distortion models (that usually augment pinhole projection) as different instances of projection

Figure 2.2: Examples of radial projection models: (left) admitting only fields of view $\ll 180^\circ$ such as ① pinhole projection followed by ② applying lens distortion model like Brown Conrady [66]; (right) allowing fields of view $> 180^\circ$ such as division model [67].



models. We also assume that the center of the radial lens distortion coincides with the principal point.

A general *radial* (or *radially-symmetric*) projection can be characterized using an implicit function $f(r, R, Z) = 0$ involving three terms: a radial component r of an image point, a radial component R and a depth component Z of a scene point to be projected. Typically, the model takes the form of either forward projection $r = \phi_\theta(R, Z)$ or backward (or back-) projection $r : rZ - R\psi_\theta(r) = 0$, where θ indicates model parameters. It can be conveniently written as

$$\begin{pmatrix} x \\ y \\ 1 + c \end{pmatrix} \sim \mathbf{X}, \tag{2.4}$$

where for the forward projection $c = \frac{Z}{R}\phi_\theta(R, Z) - 1$, and for the backward projection $c = \psi_\theta(r) - 1$.

Table 2.1: Parametric radial projection models. Models compute either radially-symmetric projection, $r = \phi_\theta(R, Z)$, or back-projection, $r : rZ - R\psi_\theta(r) = 0$, where R and Z are the radial and depth components of a scene point, and r is the distance from the center of projection of a retinal point. The right column lists functions for published models.

	Model	Params θ	Radial (Back-)Projection Function
Forward	Brown-Conrady [68]	$\{k_1, k_2\}$	$\phi_\theta(R, Z) = \frac{R}{Z} + \sum_{n=1}^{2} k_n \frac{R}{Z}^{2n+1}$
	Kannala-Brandt [66]	$\{k_1, \ldots, k_4\}$	$\phi_\theta(R, Z) = \zeta + \sum_{n=1}^{4} k_n \zeta^{2n+1}$, $\zeta = \text{atan2}(R, Z)$
	Unified Camera [69]	$\{\xi\}$	$\phi_\theta(R, Z) = R \frac{\xi+1}{\xi(\sqrt{R^2+Z^2})+Z}$
	Field of View [70]	$\{w\}$	$\phi_\theta(R, Z) = \frac{1}{w} \text{atan2}\left(2R \tan \frac{w}{2}, Z\right)$
	Extended Unified Camera [58]	$\{\alpha, \beta\}$	$\phi_\theta(R, Z) = \frac{R}{\alpha d + (1-\alpha)Z}$, $d = \sqrt{\beta R^2 + Z^2}$
	Double Sphere [59]	$\{\xi, \alpha\}$	$\phi_\theta(R, Z) = \frac{R}{\alpha d_2 + (1-\alpha) Z_2}$, $d_2 = \sqrt{R^2 + Z_2^2}$, $Z_2 = \xi\sqrt{R^2 + Z^2} + Z$
Backward	Division [67], [71]	$\{a_1, a_2, a_3\}$	$\psi_\theta(r) = 1 + \sum_{n=1}^{3} a_n r^{n+1}$
	Division-Even [72]	$\{\lambda_1, \ldots, \lambda_N\}$	$\psi_\theta(r) = 1 + \sum_{n=1}^{N} \lambda_n r^{2n}$

A list of the most commonly used radial projection models is shown in Table 2.1. Note that Brown Conrady model [66] is technically a combination of pinhole projection and Brown Conrady distortion. Because it includes pinhole projection, it can only accommodate the slight distortions present in the images and narrower, *i.e.* $\ll 180°$ fields of view, whereas the other models are able to model significant distortions (see Figure 2.2). While Table 2.1 lists parametric models only, a non-parametric function mapping radii to ray angles can also be used [73], [74].

A particular interest of this thesis is in the division model that has the

following general form of the back-projection function

$$rZ - R\psi_\theta(r) = 0 \qquad \psi_\theta(r) = 1 + \sum_{n=1}^{N} \lambda_n r^{2n}. \tag{2.5}$$

It is a simple but flexible function able to model different lenses ranging from no distortion ($\lambda_i = 0$) to significant distortion with fields-of-view greater than 180° [67] (*e.g.* fish-eye and catadioptric lenses), accurately enough. We use this model to construct a solver for camera calibration, as detailed in Section 3.4 and **Paper A**, and we argue for using it to initialize other projection models.

Other projections

We will briefly touch upon other possible projection models. These include modeling *decentering* (or *tangential*) distortion, for which the most commonly used model is (tangential part of) the Brown-Conrady distortion model [68] augmenting the pinhole projection

$$\begin{aligned} x &= \overline{x} + 2p_1\overline{xy} + p_2(\overline{r}^2 + 2\overline{x}^2) \\ y &= \overline{y} + p_1(\overline{r}^2 + 2\overline{y}^2) + 2p_2\overline{xy}, \end{aligned}$$

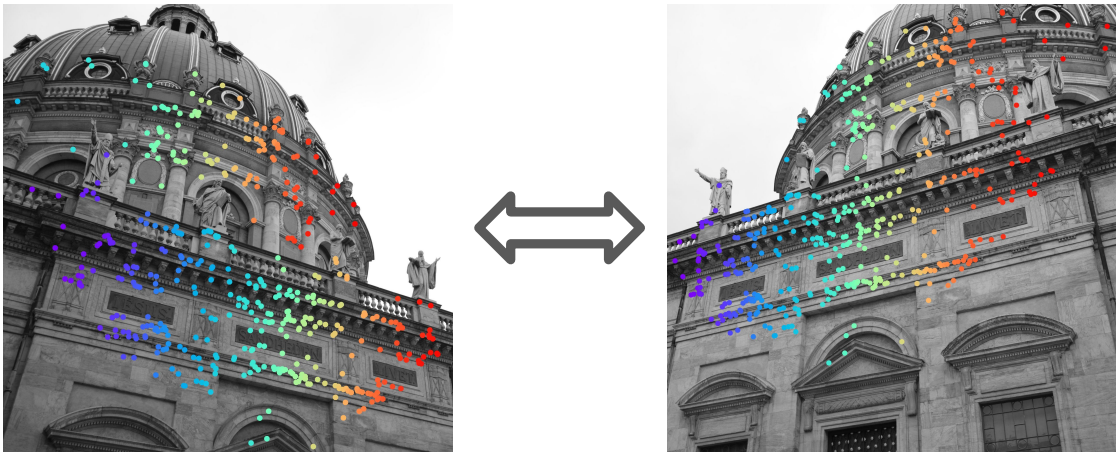
where $\overline{x} = X/Z$, $\overline{y} = Y/Z$, and $\overline{r}^2 = \overline{x}^2 + \overline{y}^2$.

So far we have discussed the parametric functions with relatively few (or no) parameters. There also exist *generic* camera models that associate each pixel with the viewing ray direction (or line for non-central cameras) using *e.g.* lookup tables [75]–[77], control points and B-spline interpolation [78], [79]. A neural network could also in principle be an interesting candidate for a generic camera model. These models are usually thought of as non-parametric or having an extreme number parameters. Generic camera models approximate real-world cameras much more accurately but at the cost of a more tedious optimization and risks of overfitting to the calibration data (if it does not contain a dense enough set of points).

2.2 Two-view geometry

In this section we look at the geometry of the pinhole projections of the same 3D point $\mathbf{X}$ in two cameras. The projections of the same 3D point are

Figure 2.3: Sparse point correspondences between images.



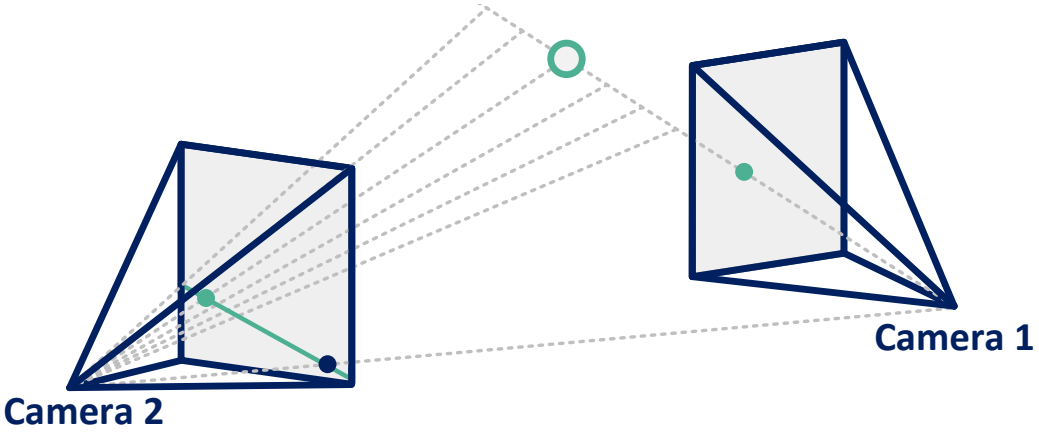
usually called *corresponding points* or *correspondences* (or *matches*)—this is the definition we assume in this thesis[1]. An example of sparse correspondences is illustrated in Figure 2.3. The search for the corresponding points, often called *matching*, is discussed further in Section 2.3. We will discuss some elements of *epipolar geometry*. Here, *epipolar* is connected to the notion of *epipole* (an image of one camera center in the other camera), *epipolar planes* (planes containing both camera centers) and *epipolar lines* (intersection of the epipolar planes with sensor/image planes of the two cameras). See Figure 2.4 for an illustration. This serves as a foundation for the two-view reconstruction pipeline described in the next section.

Essential and fundamental matrix

Let us first look at the sensor projections $\{\mathbf{x}_i\}_{i=1}^2$ in two cameras. Without loss of generality, let the world CS be the same as the camera CS of the first camera, *i.e.* such that the first projection is simply $\lambda_1\mathbf{x}_1 = \mathbf{X}$, and the second one includes the transformation from the second camera CS to the first one $\lambda_2\mathbf{x}_2 = \tilde{\mathtt{R}}\mathbf{X} + \tilde{\mathbf{t}}$. Here, $(\tilde{\mathtt{R}}, \tilde{\mathbf{t}})$ is a *relative pose*. Let us insert $\mathbf{X}$ from the first

[1] Depending on the context, corresponding points could also mean different 3D points that have the same semantic description [80], *e.g.* the top right corner of two different chairs. We do not assume this definition.

Figure 2.4: Epipolar geometry. The blue point is an epipole, in particular an image of the first camera center in the second camera. The green line is an epipolar line corresponding to the scene point (shown in green outline), namely an intersection of the epipolar plane containing the scene point with the second image plane. Knowing the imaged scene point in the first camera, we can infer that the imaged scene point in the second camera should lie on this line. In real images it will be close to but not lie on the line due to noise.



projection equation into the second one as following

$$\mathbf{x}_2 \sim \lambda_1 \tilde{\mathtt{R}} \mathbf{x}_1 + \tilde{\mathbf{t}} \tag{2.6}$$

and left-multiply both sides with $\mathbf{x}_2^\top [\tilde{\mathbf{t}}]_\times$ as below

$$\mathbf{x}_2^\top [\tilde{\mathbf{t}}]_\times \mathbf{x}_2 \sim \lambda_1 \mathbf{x}_2^\top [\tilde{\mathbf{t}}]_\times \tilde{\mathtt{R}} \mathbf{x}_1 + \mathbf{x}_2^\top [\tilde{\mathbf{t}}]_\times \tilde{\mathbf{t}}. \tag{2.7}$$

The l.h.s. vanishes (since a cross-product is orthogonal to its argument vectors), as well as the last term in the r.h.s. (as a cross product of a vector with itself). We have now managed to eliminate the underlying 3D point and simplify the relation into the following form

$$\mathbf{x}_2^\top \underbrace{[\tilde{\mathbf{t}}]_\times \tilde{\mathtt{R}}}_{\sim \mathtt{E}} \mathbf{x}_1 = 0. \tag{2.8}$$

Matrix $\mathtt{E}$ is called the *essential matrix*. It is defined up to scale—if $\mathtt{E}$ satisfies $\mathbf{x}_2^\top \mathtt{E} \mathbf{x}_1 = 0$ then $s\mathtt{E}$ does too. It has five degrees of freedom ($\mathbf{t}$ has three parameters, so does $\mathtt{R}$, giving six in total, but scale ambiguity removes one). The relation (2.8) is considered in calibrated scenarios (*i.e.* when $\mathtt{K}_i$ are estimated

beforehand, see *e.g.* Section 3.4) since the sensor points need to be obtained first (as $\mathbf{x}_i = \mathtt{K}_i^{-1}\mathbf{u}_i$) to verify that $\mathbf{x}_2^\top \mathtt{E}\mathbf{x}_1 \approx 0$.

By inserting $\mathbf{x}_i = \mathtt{K}_i^{-1}\mathbf{u}_i$ into (2.8), we obtain a two-view relation between the corresponding pixel image points

$$\mathbf{u}_2^\top \underbrace{\mathtt{K}_2^{-1\top}\mathtt{E}\mathtt{K}_1^{-1}}_{\sim \mathtt{F}} \mathbf{u}_1 = 0. \tag{2.9}$$

Matrix $\mathtt{F}$ is called the *fundamental matrix*. Similarly to the essential matrix $\mathtt{E}$, matrix $\mathtt{F}$ is defined up to scale. It can also be seen that $\mathtt{F}$ is in general rank-2 (due to $[\tilde{\mathbf{t}}]_\times$), therefore it has seven degrees of freedom [24]. The relation (2.9) is often considered in uncalibrated scenarios (*i.e.* when the calibration matrices $\mathtt{K}_i$ are not known a priori nor pre-estimated) since the image points can be used directly to verify $\mathbf{u}_2^\top \mathtt{F}\mathbf{u}_1 \approx 0$.

In the next section, we discuss the role of epipolar geometry in 3D reconstruction, in particular from two views, in more detail.

2.3 Two-view reconstruction

In *two-view reconstruction*, the goal is to recover the relative camera pose and the 3D coordinates of the scene points from two-view point correspondences. It is often done through an intermediate recovery of the epipolar geometry—as seen in equations (2.8) and (2.9), epipolar geometry holds for any pair of corresponding points (independently of the coordinates of $\mathbf{X}$), so it only depends on the extrinsics, and in the uncalibrated case also on $\mathtt{K}_i$, which allows to use the above relations in 3D reconstruction pipelines that estimate camera geometry *before* recovering the geometric structure of the 3D scene.

Recovering epipolar geometry To obtain $\mathtt{E}$ from a set of correspondences $\{\mathbf{x}_{1,j} \leftrightarrow \mathbf{x}_{2,j}\}_{j=1}^m$ (where m has to be ≥ 5, since $\mathtt{E}$ has five degrees of freedom), we can construct linear constraints on $\mathrm{vec}(\mathtt{E})$ from (2.8) as following

$$(\mathbf{x}_{1,j} \otimes \mathbf{x}_{2,j})^\top \mathrm{vec}(\mathtt{E}) = 0 \qquad j = 1, \ldots, m \tag{2.10}$$

In the noiseless case, these constraints are satisfied exactly. In practice, the points $\{\mathbf{x}_{i,j}\}$ are noisy, so the goal is to minimize the *algebraic errors* $r_j(\mathtt{E}) = (\mathbf{x}_{1,j} \otimes \mathbf{x}_{2,j})^\top \mathrm{vec}(\mathtt{E})$. Additionally, $\mathtt{E}$ has to satisfy the essential matrix constraints [24] that can be written as $\mathtt{E}\mathtt{E}^\top\mathtt{E} - \frac{1}{2}\mathrm{tr}(\mathtt{E}\mathtt{E}^\top)\mathtt{E} = \mathtt{0}$ [30]. Efficient specialized solvers exist that recover $\mathtt{E}$ from minimal (five) [30], [81]

and non-minimal (≥ 5) [82] number of correspondences. Similarly, specialized solvers exist for $\mathtt{F}$ [24], [28], [29]. In uncalibrated scenarios, it is common to assume initially that $s_x = s_y$, $\alpha = 0$, and that $(p_x, p_y)^\top$ is at the image center, leaving only one unknown for intrinsics, namely focal length in pixels, which allows designing solvers for recovering $\mathtt{E}$ together with the focal length [81].

Point correspondences often contain outliers that must be filtered out. This is typically done using RANSAC-style outlier detection methods (that will be covered in more detail in Section 2.5). An important component of these methods is model hypothesis evaluation. Algebraic errors r_j, while being computationally cheap, give us little (geometric) understanding of how good matrix $\mathtt{E}$ is [83]. A more geometrically meaningful metric is *e.g.* the sum of distances to the epipolar lines [84], [85] or its first-order approximation, Sampson error [86]–[88] that is more computationally efficient. It can also be used for further model refinement (see Section 2.5).

While epipolar geometry does not depend on the structure, various structure configurations make it more difficult to recover epipolar geometry. An important example is presence of a dominant scene plane [89]—not an uncommon scenario for manmade structures—that may cause degenerate point configurations *w.r.t.* the model [24], *i.e.* not containing enough information to uniquely define the model. This can be handled *e.g.* by integrating model selection [35], [90], [91].

Recovering relative pose and structure Let us assume that the essential matrix $\mathtt{E}$ is successfully recovered. Then we would like to solve the "inverse" problem of recovering the relative camera rotation $\tilde{\mathtt{R}}$ and translation $\tilde{\mathbf{t}}$ from $\mathtt{E}$ such that $\mathtt{E} \sim [\tilde{\mathbf{t}}]_\times \tilde{\mathtt{R}}$. Note that translation $\tilde{\mathbf{t}}$ can in fact be recovered only up to scale since $[s\tilde{\mathbf{t}}]_\times \tilde{\mathtt{R}}$ also satisfies the essential matrix constraints in (2.8). We can rescale $\mathtt{E}$ to assume the following SVD

$$\mathtt{E} = \mathtt{U}\,\mathrm{diag}(1, 1, 0)\mathtt{V}^\top, \tag{2.11}$$

such that $\det(\mathtt{U}\mathtt{V}^\top) = 1$. Define an auxiliary matrix $\mathtt{W}$

$$\mathtt{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.12}$$

Then the relative camera rotation is either $\mathtt{U}\mathtt{W}\mathtt{V}^\top$ or $\mathtt{U}\mathtt{W}^\top\mathtt{V}^\top$, and the direction of the relative translation vector is either $\mathbf{u}_{(3)}$ (the third column of $\mathtt{U}$) or

$-\mathbf{u}_{(3)}$ [24]. The ambiguity is resolved by recovering the 3D points (see *Triangulation* in Section 3.2), and checking for their plausibility which is usually defined using *cheirality condition* (the 3D point satisfies the cheirality condition if it is in front of both cameras) and often by verifying whether the 3D point is also not too far from both cameras and/or the center of gravity. In practice, some noisy point correspondence might still be present in the data causing implausibility of a subset of the reconstructed 3D points, therefore the relative pose with the most plausible 3D points is chosen. Finally, the relative camera pose $\{\tilde{\mathtt{R}}, \tilde{\mathbf{t}}\}$ and the 3D points $\{\mathbf{X}_j\}$ (and optionally intrinsics) are refined via minimization of the "gold-standard" sum of reprojection errors [24]

$$\min_{\substack{\tilde{\mathtt{R}}\in\mathrm{SO}(3)\\ \tilde{\mathbf{t}},\{\mathbf{X}_j\}}} \sum_j \rho\left(\|\mathtt{K}\pi\left(\mathbf{X}_j\right) - \mathbf{u}_{1,j}\|\right) + \rho\left(\|\mathtt{K}\pi\left(\tilde{\mathtt{R}}\mathbf{X}_j + \tilde{\mathbf{t}}\right) - \mathbf{u}_{2,j}\|\right). \tag{2.13}$$

We discuss this kind of optimization in more detail in Sections 2.5 and 3.2.

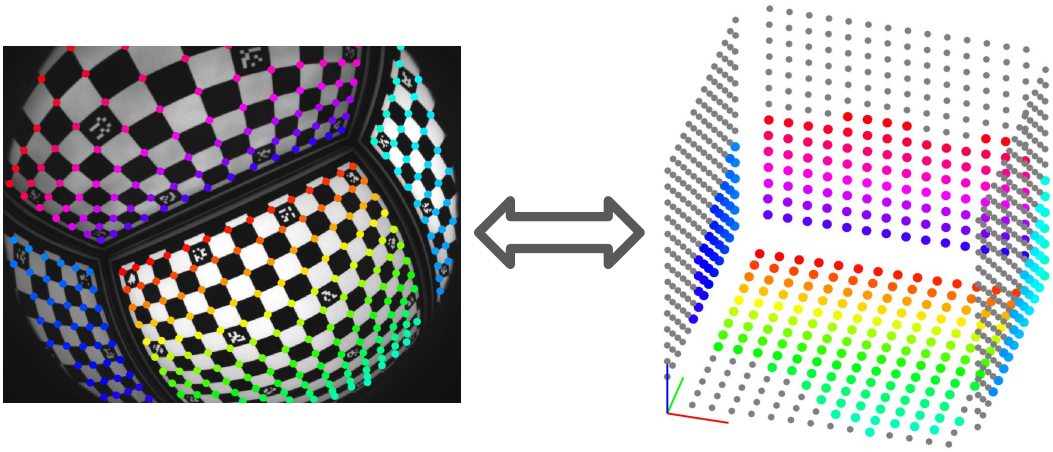
Searching for corresponding points

How can we get the corresponding points $\{\mathbf{u}_{1,j} \leftrightarrow \mathbf{u}_{2,j}\}$ (shown in Figure 2.3) needed for the two-view reconstruction pipeline described above? We are only given the images. In general, this is a challenging problem—changes in viewpoint, lighting, or weather conditions can cause drastic differences in the RGB values of image points. It is therefore an active area of research [92]–[95]. The search for point correspondences has traditionally been done via keypoint detection and description—with notable examples such as handcrafted SIFT [26], ORB [27], or learning-based approaches SuperPoint [96], AffNet [97], D2-Net[98], DeDoDe [99], [100]. This is followed by matching—for example, via nearest neighbor search, or with neural networks like SuperGlue [36] or LightGlue [101]. These methods produce sparse matches. A parallel track of works proposes to learn (semi-)dense image matching end-to-end (such as LoFTR [102], DKM [103], RoMA [37]) demonstrating remarkable results. The dense matches can be "summarized" to reduce the computational cost [104].

2.4 Single-view geometry of planar scenes with radial distortion

Another useful relation can be established between planar scene points and their corresponding sensor projections where a non-zero radial distortion is assumed [105]. The relation is established through the so-called *radial fundamental matrix*. It can be particularly useful for offline camera calibration, where planar 3D targets with known patterns are used to establish 2D-3D correspondences (see Figure 2.5). Camera calibration pipeline will be discussed in more detail in Section 3.4.

Figure 2.5: Point correspondences between planar scene points and their projections.



Radial fundamental matrix

Let the scene point $\mathbf{X}$ belong to a plane with the normal direction $\mathbf{n}$. Without loss of generality, let the world CS be such that its origin is on the plane and the third coordinate axis is $\mathbf{n}$. Then $\mathbf{X} = \begin{pmatrix} X & Y & 0 \end{pmatrix}^\top$. The world-to-camera transformation together with radial projection give the following relation

$$\begin{pmatrix} x \\ y \\ 1+c \end{pmatrix} \sim \mathtt{R}\mathbf{X} + \mathbf{t}, \tag{2.14}$$

which can also be written as

$$\mathbf{x} + \mathbf{c} \sim \mathtt{H}\mathbf{y}, \tag{2.15}$$

where $\mathbf{c} = \begin{pmatrix} 0 & 0 & c \end{pmatrix}^\top$, $\mathbf{y} = \begin{pmatrix} X & Y & 1 \end{pmatrix}^\top$, $\mathtt{H} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$, with $\mathbf{r}_1$ and $\mathbf{r}_2$ being the first two columns of $\mathtt{R}$. Left-multiplying both sides with $\mathbf{x}^\top [\mathbf{c}]_\times$ gives

$$\mathbf{x}^\top [\mathbf{c}]_\times \mathbf{x} + \mathbf{x}^\top [\mathbf{c}]_\times \mathbf{c} \sim \mathbf{x}^\top [\mathbf{c}]_\times \mathtt{H}\mathbf{y}, \tag{2.16}$$

and since l.h.s. vanishes, we get the following relation

$$\mathbf{x}^\top \underbrace{[\mathbf{c}]_\times \mathtt{H}}_{\mathtt{E}_\mathtt{H}} \mathbf{y} = 0. \tag{2.17}$$

In particular, matrix $\mathtt{E}_\mathtt{H}$ can be simplified into

$$\mathtt{E}_\mathtt{H} \sim \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} = \begin{pmatrix} -r_{21} & -r_{22} & -t_2 \\ r_{11} & r_{12} & t_1 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.18}$$

We could estimate $\mathtt{E}_\mathtt{H}$ from $\{\mathbf{x}_j \leftrightarrow \mathbf{X}_j\}$, but in typical applications, the camera calibration matrix $\mathtt{K}$ is not known. Let us then establish a relation between pixel image points and scene points $\{\mathbf{u}_j \leftrightarrow \mathbf{y}_j\}$, where $\mathbf{u}_j = \mathtt{K}\mathbf{x}_j$. Left-multiplying both sides of (2.15) with $\mathtt{K}$ gives

$$\mathbf{u} + c\ \mathbf{e} \sim \mathtt{KH}\mathbf{y}, \tag{2.19}$$

where $\mathbf{e} = \begin{pmatrix} p_x & p_y & 1 \end{pmatrix}^\top$ is the principal point (which we assume is equal to the center of projection/distortion). Left-multiplying both sides with $\mathbf{u}^\top [\mathbf{e}]_\times$ gives

$$\mathbf{u}^\top [\mathbf{e}]_\times \mathbf{u} + c\mathbf{u}^\top [\mathbf{e}]_\times \mathbf{e} \sim \mathbf{u}^\top [\mathbf{e}]_\times \mathtt{KH}\mathbf{y}, \tag{2.20}$$

and since l.h.s. vanishes again, we obtain

$$\mathbf{u}^\top \underbrace{[\mathbf{e}]_\times \mathtt{KH}}_{\sim \mathtt{F}_\mathtt{H}} \mathbf{y} = 0. \tag{2.21}$$

$\mathtt{F}_\mathtt{H}$ is an instance of a fundamental matrix [105]—it is similarly defined up to scale and is generally rank-2. Here, we call it a *radial fundamental matrix*. $\mathtt{F}_\mathtt{H}$ can be recovered from point correspondences $\{\mathbf{u}_j \leftrightarrow \mathbf{X}_j\}$ using the same approaches designed for estimating the fundamental matrix in two-view scenarios (see Section 2.3).

Recovering partial intrinsics and extrinsics It is possible to recover some intrinsics and some extrinsics from $\mathtt{F}_\mathtt{H}$, namely $\mathbf{e}$, $\mathtt{R}$, t_1, and t_2. The center of projection is a left null space of $\mathtt{F}_\mathtt{H}$ [105]. To recover $\mathtt{R}$, t_1, and t_2, it is useful to decompose $\mathtt{K}$ into the offset matrix and the scaling matrices following (2.2) in Section 2.1. An important observation is how $[\mathbf{e}]_\times$ affects the offset matrix

$$[\mathbf{e}]_\times \begin{pmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ -p_y & p_x & 0 \end{pmatrix} = [\mathbf{e}]_\times \operatorname{diag}(1, 1, 0). \tag{2.22}$$

Inserting (2.22) into (2.21) gives the following decomposition of the radial fundamental matrix

$$\begin{aligned} \mathtt{F}_\mathtt{H} &\sim [\mathbf{e}]_\times \operatorname{diag}(1, 1, 0) \begin{pmatrix} s_x & \alpha s_y & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathtt{H} \\ &\sim [\mathbf{e}]_\times \begin{pmatrix} s_x/s_y & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 \end{pmatrix}, \end{aligned} \tag{2.23}$$

which can be simplified further if square pixels are assumed

$$\mathtt{F}_\mathtt{H} \sim [\mathbf{e}]_\times \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.24}$$

which relates elements of $\mathtt{F}_\mathtt{H}$ to the partial pose. Further details, as part of the contributions of **Paper A**, are provided in Section 3.4.

Establishing 2D-3D correspondences for calibration

As mentioned earlier, in offline camera calibration, planar targets with known patterns are used. Hence, the problem of establishing 2D-3D correspondences is simplified into the problem of detecting and disambiguating these known patterns. Most methods use checkerboard patterns [55], [106]–[110] that are typically augmented with fiducial markers like ARTags [111], AprilTags [112], ArUco [113] (also illustrated in Figure 2.5). Other commonly used patterns include circular points [56], [107], [114], deltille grids with DelTags [115] and star patterns [79]. The modern approaches can achieve very high detection and localization accuracy for images of varying quality (including blur, uneven or poor illumination, occlusion, and strong geometric distortion) [79], [115].

2.5 Optimization methods

This section discusses a set of optimization techniques that are highly relevant to 3D vision applications.

Nonlinear least squares

Many problems in 3D vision are formulated as estimating the unknown parameters $\boldsymbol{\theta}$ of the pre-defined generating model by fitting it to data [24]. The typical assumption is that the residuals $\mathbf{r}_i(\boldsymbol{\theta})$ are realizations of the i.i.d. Gaussian random vectors $\mathbf{r}_i \sim \mathcal{N}(\mathbf{0}, \mathsf{I})$, and the goal is to obtain a maximum likelihood estimate (MLE)

$$\max_{\boldsymbol{\theta}} \prod_i p\left(\mathbf{r}_i(\boldsymbol{\theta})\right), \tag{2.25}$$

which is equivalent to minimizing the squared residual norms

$$\min_{\boldsymbol{\theta}} \tfrac{1}{2} \sum_i \|\mathbf{r}_i(\boldsymbol{\theta})\|^2. \tag{2.26}$$

The residuals $\mathbf{r}_i$ often take the form of the differences between the measurements and the corresponding outputs of the parametric model. For example, in bundle adjustment, structure or pose estimation (see Section 3.2), or camera calibration (Section 3.4), the differences are between the 2D points—the keypoints detected in the images and the corresponding projections of the 3D points; in rotation averaging (Section 3.5), the differences are between rotations—the input relative rotations and the computed relative rotations using the sought-after absolute rotations; in 3D point cloud registration, the differences are between the 3D points—points from the source point cloud and from the target point cloud transformed into the source coordinate system. In most cases, the generating model is nonlinear. For instance, the pinhole projection of the 3D point is $\chi\left(\mathtt{K}, \mathtt{R}, \mathbf{X}, \mathbf{t}\right) = \mathtt{K}\frac{\mathtt{R}\mathbf{X}+\mathbf{t}}{\mathtt{R}_{3,:}\mathbf{X}+t_z}$. Hence we are interested in the optimization toolbox solving nonlinear least squares problems [116]. This is typically one of the approximate second-order methods such as Gauss-Newton (that uses the first-order Taylor approximation of the residual terms $\mathbf{r}_i$ around the current iterate) or its trust region modification Levenberg-Marquardt [117]–[119]. In large-scale optimization problems, exploiting the problem structure is essential for achieving computational efficiency. Sparsity is common in reconstruction problems and it can be leveraged

when implementing a parameter update. Other important aspects, including matrix inversion, parameterization, and problem-specific structures, are discussed in [21], [120].

Robust nonlinear least squares

Robust nonlinear least-squares optimization finds its practical applications in numerous vision problems [21], [52], [121]–[127] since dealing with outliers is unavoidable when working with real data. To formulate the optimization, we let $x_i(\boldsymbol{\theta}) = \|\mathbf{r}_i(\boldsymbol{\theta})\|$ be the l_2 norms of the individual residual vectors $\mathbf{r}_i(\boldsymbol{\theta})$, and write the new robust objective that we aim to optimize

$$\min_{\boldsymbol{\theta}} \sum_i \rho\left(x_i(\boldsymbol{\theta})\right), \tag{2.27}$$

where $\rho(\cdot)$ is the robust kernel. We refer to Table 2.2 for the most commonly used kernel functions and Section 9.4 in [83] for an in-depth overview.

Table 2.2: Robust kernel functions and their related functions, and comparison to the non-robust l_2 kernel.

Name	$\rho(x)$	$\kappa(w)$	$\overline{w}(x) = \rho'(x)/x$
l_1	$\lvert x\rvert$	$\frac{1}{w}$	$\frac{1}{\lvert x\rvert}$
Geman-McClure	$\frac{\tau^2 x^2}{2(x^2+\tau^2)}$	$\frac{\tau^2}{2}(\sqrt{w}-1)^2$	$\frac{\tau^4}{(x^2+\tau^2)^2}$
Cauchy/Lorentzian	$\frac{\tau^2}{2}\log\left(1+\frac{x^2}{\tau^2}\right)$	$\frac{\tau^2}{2}\left(w-\log(w)-1\right)$	$\frac{\tau^2}{\tau^2+x^2}$
Huber	$\begin{cases}\frac{1}{2}x^2 \\ \tau\lvert x\rvert - \frac{1}{2}\tau^2\end{cases}$	$\frac{1}{2}\tau^2\left(\frac{1}{w}-1\right)$	$\begin{cases}1 & \lvert x\rvert \le \tau \\ \frac{\tau}{\lvert x\rvert} & \lvert x\rvert > \tau\end{cases}$
Tukey's biweight	$\begin{cases}\frac{\tau^2}{6}\left(1-\left(1-\frac{x^2}{\tau^2}\right)^3\right) \\ \frac{\tau^2}{6}\end{cases}$	$\frac{\tau^2}{6}(\sqrt{w}-1)^2(2\sqrt{w}+1)$	$\begin{cases}\left(1-\frac{x^2}{\tau^2}\right)^2 & \lvert x\rvert \le \tau \\ 0 & \lvert x\rvert > \tau\end{cases}$
Truncated quadratic	$\frac{1}{2}\min\{\tau^2, x^2\}$	$\frac{1}{2}\tau^2(1-w)$	$\begin{cases}1 & \lvert x\rvert \le \tau \\ 0 & \lvert x\rvert > \tau\end{cases}$
Smooth truncated quadratic ($p>1$)	$\begin{cases}\frac{1}{2}x^2\left(1-\frac{p-1}{p}\left(\frac{x^2}{\tau^2}\right)^{\frac{1}{p-1}}\right) \\ \frac{\tau^2}{2p}\end{cases}$	$\frac{1}{p}\tau^2\left(w-1\right)^p$	$\begin{cases}1-\left(\frac{x^2}{\tau^2}\right)^{\frac{1}{p-1}} & \lvert x\rvert \le \tau \\ 0 & \lvert x\rvert > \tau\end{cases}$
l_2	$\frac{1}{2}x^2$	0	1

It was shown [21], [122], [124], [128] that the robustified problem can be cast as an instance of nonlinear least squares, which is compelling since we can use the same optimization toolbox as introduced in the previous section. One of the most common ways to implement this approach in computer vision is via *iteratively reweighted least squares* (IRLS) [121], [128]. IRLS has been extensively studied in various applications [129]–[135]. This robustification technique is relatively inexpensive and, if a sufficiently good initialization is provided, IRLS can refine it into a good local minimum. However, if the initialization is not good enough, IRLS shows limited ability in escaping poor local minima [124], [136], which can make the underlying pipelines less reliable. To address this issue, several techniques are proposed that prevent overconfident updates, one of them is called *generalized majorization-minimization* (GeMM) [61], [136]. Define the new problem of jointly optimizing the main parameters $\boldsymbol{\theta}$ together with additional variables $\{w_i\}$

$$\min_{\boldsymbol{\theta},\{w_i\}} \sum_i \overline{\rho}\left(x_i(\boldsymbol{\theta}), w_i\right), \tag{2.28}$$

where $\overline{\rho}(x, w) = \frac{1}{2}wx^2 + \kappa(w)$, and $\kappa(\cdot)$ is a *bias function*. In the literature, using $\overline{\rho}(x, w)$ is referred to as *lifting the kernel* since the new objective is augmented with additional "latent" variables w_i, often seen as confidence weights for the corresponding residual terms x_i. For example, if w_i is close to zero, x_i can be considered an outlier. The choice of bias $\kappa(\cdot)$, which can also be seen as the weight regularization function, determines the form of the kernel $\rho(\cdot)$ such that when the objective in (2.28) is optimized only with respect to the weights $\{w_i\}$, the value coincides with the value of the objective in (2.27) (see Table 2.2 for some examples). The new objective (2.28) generally gives an upper bound [137] of the original objective in (2.27). Then, the objective of IRLS is the surrogate function

$$\min_{\boldsymbol{\theta}} \tfrac{1}{2} \sum_i w_i^* \cdot x_i^2(\boldsymbol{\theta}) \qquad \text{s.t.} \quad w_i^* = \overline{w}\left(x_i(\boldsymbol{\theta}^*)\right), \tag{2.29}$$

where $\overline{w}(x) = \frac{\rho'(x)}{r}$ is sometimes called an IRLS weight function (also shown in Table 2.2), and $\boldsymbol{\theta}^*$ are the optimal parameters. It has the same optimality conditions [124] as the original objective (2.27). Note, however, that the optimal parameters $\boldsymbol{\theta}^*$ are not known a-priori. For that reason, IRLS employs iterative updates of the weights w_i based on the current solution followed by

updating the main parameters based on the current weights

$$w_i^{(t)} \leftarrow \overline{w}\left(x_i\left(\boldsymbol{\theta}^{(t-1)}\right)\right) \tag{2.30}$$

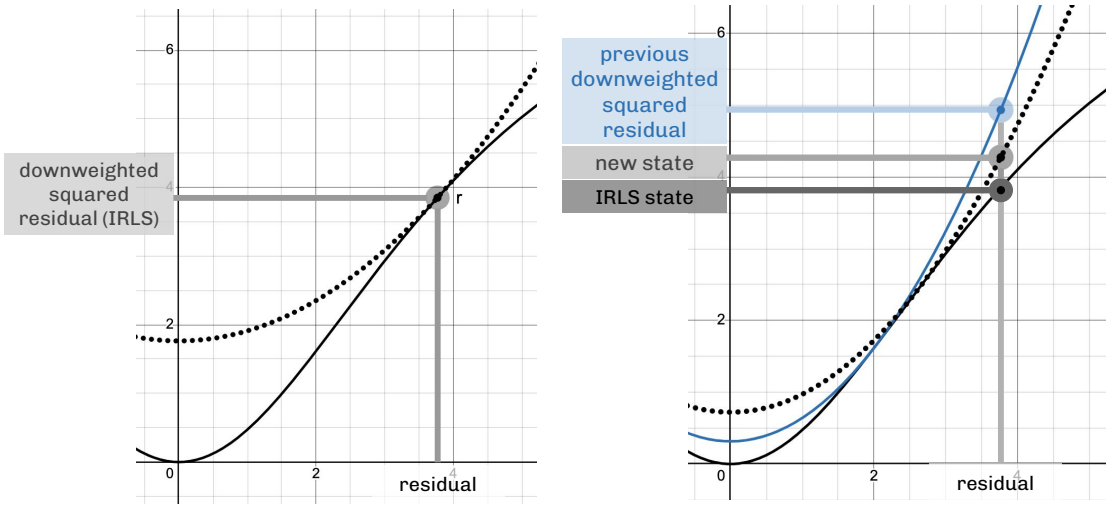
$$\boldsymbol{\theta}^{(t)} \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_i w_i^{(t)} \cdot x_i^2(\boldsymbol{\theta}), \tag{2.31}$$

starting *e.g.* from $w_i^{(0)} = 1$.

In IRLS, the "optimal" weights w_i given by $\overline{w}\left(x_i^{(t)}\right)$ satisfy the *touching* condition

$$\sum_i \overline{\rho}\left(x_i^{(t)},\ w_i\right) = \sum_i \rho\left(x_i^{(t)}\right). \tag{2.32}$$

Figure 2.6: Choosing the weights in robust optimization with IRLS *vs.* GeMM. In IRLS (left), the weights are chosen to satisfy the touching condition, whereas in GeMM (right) the weights are slightly adjusted towards the IRLS state while still guaranteeing to reduce the objective.



GeMM instead iteratively adjusts the weights guided by both the IRLS state and the previous state—it searches for the weights to satisfy the following

condition

$$\sum_i \overline{\rho}\left(x_i^{(t)}, w_i\right) \leq \eta \sum_i \rho\left(x_i^{(t)}\right) + (1-\eta) \sum_i \overline{\rho}\left(x_i^{(t)}, w_i^{(t-1)}\right). \tag{2.33}$$

This ensures that the upper bound objective value is only sufficiently improved [136], which is easy to see when the r.h.s. of (2.33) is re-written as following

$$\sum_i \overline{\rho}\left(x_i^{(t)}, w_i^{(t-1)}\right) - \eta \underbrace{\sum_i \overline{\rho}\left(x_i^{(t)}, w_i^{(t-1)}\right) - \rho\left(x_i^{(t)}\right)}_{\geq 0}, \tag{2.34}$$

since $\overline{\rho}(x, w) \geq \rho(x)$ for any w. The weights are therefore adjusted only slightly towards the "optimal"/IRLS state (see Figure 2.6) while still guaranteeing a reduction in the objective value, enabling "exploration". This prevents the algorithm from overconfident updates that could otherwise have led to a poor local minimum.

RANSAC

To provide a good initialization for robust local optimization, a method called Random Sample Consensus (RANSAC) [138] is often used. RANSAC is an iterative algorithm that filters outliers in the data by searching for an underlying generating model in a heuristic way. What this means is that at each iteration, RANSAC samples a pre-defined number of data points that is sufficient to estimate the underlying generating model. Such a model could, for example, be the fundamental matrix estimated from a set of point correspondences. The estimated model is evaluated against all data points by computing the model-to-data error. In the example with the fundamental matrix, the evaluation is typically done by computing the Sampson error [86], [87]. The inliers are identified by thresholding the corresponding errors. RANSAC keeps track of the model with the highest number of inliers and runs for sufficiently many iterations to ensure that the found model is indeed good with the pre-defined confidence. In particular, a well-known RANSAC termination criterion is reaching $\lceil \frac{\log(1-p)}{\log(1-\eta^s)} \rceil$ iterations, where p is the confidence (say, 0.999), η is the inlier ratio of the best-so-far model (that approximates the true inlier ratio), and s is the sample size. Note that increasing s will decrease the probability of an all-inlier sample η^s which leads to a higher number of iterations required to

run, slowing down the algorithm. Therefore, unless there is a domain knowledge that the inlier ratio is high, it is often beneficial to reduce s as much as possible, which leads to the notion of minimal solvers. A *minimal solver* is an algorithm that estimates a model using the smallest possible number of data points needed (which is equal to the degrees of freedom of the model). Developing minimal solvers is an active area of research: the five-point solver [30] is perhaps the most well-known example of a minimal solver, and other examples include partially calibrated [139] and uncalibrated [24] fundamental matrix estimation, homography estimation [24], [140], general techniques [141] and automatic minimal solver generation methods [142], [143].

One controversial assumption that RANSAC makes implicitly is that the more inliers the model has the better (closer to the true) the model is. In reality, since all datapoints are corrupted by noise, the generated models and estimated inliers are perturbed as well, and the effects aggravate for smaller sample sizes [144]. In **Paper A**, we use a variant of RANSAC called locally optimized RANSAC [144] that implements the model refinement (using all inliers, in contrast to the initial minimal sample) for every new best-so-far model. The locally optimized model, that is hopefully less biased, is therefore used to determine the inlier ratio and subsequently check the termination criteria. Other variations include MLESAC [145], DEGENSAC [146], PROSAC [147], CONSAC [148] and many others [149].

Semidefinite programming

Many 3D vision problems are formulated as non-convex optimization problems for which it is computationally difficult or infeasible to find a global optimum. Some problems, however, can be relaxed into convex optimization problems. In particular, we are interested in formulating a convex relaxation which has the form of the *semidefinite program* (SDP) [23]—optimization of the linear objective function over the convex cone of positive semidefinite matrices subject to a combination of linear constraints. We can formalize an SDP as following

$$\begin{aligned} &\min_{\mathtt{X} \succeq 0} \ \langle \mathtt{C}, \mathtt{X} \rangle \\ &\text{s.t. } \langle \mathtt{A}_i, \mathtt{X} \rangle = b_i \quad i = 1, \ldots, m, \end{aligned} \tag{2.35}$$

where $\mathtt{X}$ is the unknown positive semidefinite matrix, $\mathtt{C}$ is the cost matrix, and matrices $\mathtt{A}_i$ and constants b_i form the linear constraint functions.

Many SDP relaxations are proposed for problems involving rotations [150]: point cloud and mesh registration [151]–[153], absolute pose estimation [154], rotation averaging [155]–[157], extrinsic calibration of multiple sensors from per-sensor egomotion also known as hand-eye calibration [158], [159], pose graph optimization [160]–[162].

SDPs are attractive since they provide a framework for obtaining certifiably optimal solutions and can be solved at least using general purpose solvers [163], [164]. For many problems, either empirical or theoretical guarantees [165], [166] were obtained. The scalability of SDPs is, however, limited. Solving large SDPs efficiently is therefore an active area of research [167]–[173].

Application to rotation averaging

Let us showcase the approach on a problem of rotation averaging with chordal distances that is particularly relevant to **Paper B** and **Paper C** (see also more details in Section 3.5). The problem can be compactly written in the following matrix form

$$\min_{\mathsf{R}\in\mathrm{SO}(3)^n} -\langle \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle, \tag{2.36}$$

where R vertically stacks n unknown rotation matrices $\mathsf{R} = (\mathtt{R}_1^\top \ \mathtt{R}_2^\top \ \cdots \ \mathtt{R}_n^\top)^\top$, and for now we assume that N is some cost matrix. Section 3.5 explains how to obtain this form.

The constraints in (2.36) are $\mathtt{R}_i^\top \mathtt{R}_i = \mathtt{I}$ (quadratic) and $\det(\mathtt{R}_i) = 1$ (cubic) for $i = 1, \ldots, n$. One way to relax this problem is by removing the determinant constraints [156], [165] which results in the following quadratically constrained quadratic program (QCQP)

$$\min_{\mathsf{R}} \ -\langle \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle \tag{2.37}$$

$$\text{s.t. } \mathtt{R}_i^\top \mathtt{R}_i = \mathtt{I}_3 \qquad i = 1, \ldots, n. \tag{2.38}$$

While the resulting QCQP is still non-convex, it can be relaxed into an SDP by *e.g.* taking the Lagrangian dual [23] twice. We will sketch the derivation below.

The dual variables for (2.38) are 3×3 symmetric matrices denoted Υ_i, each corresponding to the orthogonality constraint on $\mathtt{R}_i$. The Lagrangian function

can then be compactly written as

$$L(\mathsf{R};\Upsilon) = -\langle \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle - \langle \Upsilon, \mathsf{I} - \mathsf{R}\mathsf{R}^\top \rangle = \langle \Upsilon - \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle - \operatorname{tr}(\Upsilon), \tag{2.39}$$

where $\Upsilon = \operatorname{blkdiag}(\Upsilon_1, \ldots, \Upsilon_n)$. We would like to analyze $\min_{\mathsf{R}} L(\mathsf{R};\Upsilon)$. If $\Upsilon - \mathsf{N} \succeq 0$ then $\langle \Upsilon - \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle = 0$ is the minimum value hence $\min_{\mathsf{R}} L(\mathsf{R},\Upsilon) = -\operatorname{tr}(\Upsilon)$, otherwise the function is unbounded (in particular, from below). Therefore the dual problem of (2.37) is

$$\max_{\Upsilon:\Upsilon-\mathsf{N}\succeq 0} \min_{\mathsf{R}} L(\mathsf{R};\Upsilon) = \max_{\Upsilon:\Upsilon-\mathsf{N}\succeq 0} -\operatorname{tr}(\Upsilon). \tag{2.40}$$

The next step is to obtain the dual of (2.40). Let us denote the dual variables for $\Upsilon - \mathsf{N} \succeq 0$ as X, which is a symmetric matrix. The Lagrangian of (2.40) is

$$L_2(\Upsilon;\mathsf{X}) = -\operatorname{tr}(\Upsilon) + \langle \Upsilon - \mathsf{N}, \mathsf{X} \rangle = \langle \Upsilon, \mathsf{X} - \mathsf{I} \rangle - \langle \mathsf{N}, \mathsf{X} \rangle. \tag{2.41}$$

Note that $\langle \Upsilon, \mathsf{X} - \mathsf{I} \rangle = \sum_i \langle \Upsilon_i, \mathtt{X}_{ii} - \mathtt{I}_3 \rangle$, where $\mathtt{X}_{ii}$ are 3×3 block diagonal blocks of X. If $\mathtt{X}_{ii} = \mathtt{I}_3$ for all i, then $\langle \Upsilon, \mathsf{X} - \mathsf{I} \rangle$ is always zero, otherwise it is unbounded (in particular, from above). Therefore the bidual problem is

$$\min_{\mathsf{X}\succeq 0} \max_{\Upsilon} L_2(\Upsilon;\mathsf{X}) = \min_{\substack{\mathsf{X}\succeq 0 \\ \mathtt{X}_{ii}=\mathtt{I}_3}} -\langle \mathsf{N}, \mathsf{X} \rangle, \tag{2.42}$$

which is an SDP. In the so-called isotropic rotation averaging, where N only contains relative rotations, [156], [165] derived the conditions for strong duality between (2.36) and (2.42), which allow recovering a global optimum of (2.36). In anisotropic rotation averaging, where N contains relative rotations pre-multiplied by matrices encoding uncertainty, we show that this relaxation is not tight. This is discussed in more detail in Section 3.5 and **Paper B**.

2.6 Deep learning

As briefly mentioned earlier, deep learning is a subfield of machine learning [40] where parametric functions known as *neural networks* (or *artificial neural networks*) are used to model complex structures and solve problems by "learning" from large amounts of data [41]. Neural networks are highly structured (as compositions of simpler functions) and flexible (the number of parameters is very large). Because of that, they can "learn" to extract

useful (low-dimensional) features from high-dimensional data through optimization, in comparison to traditional machine learning methods that often rely on hand-crafted features. Thus, deep learning has become particularly attractive for processing images and 3D structures. To *train* a neural network, one needs to: (1) choose a suitable objective (*loss*)—in supervised [44] or self-supervised [47] frameworks, it would measure the deviation between the network's output (*prediction*) and the desired output (*annotation* or *label*), whereas in an unsupervised framework, it would focus on capturing structural relationships or patterns in the data *e.g.* similarly to clustering [40]; it can also be a combination of multiple losses; (2) optimize this objective, most commonly using one of the stochastic first-order methods [174]; and (3) integrate proper validation to help keep track of optimization and measure the final performance. In this thesis, we are interested in a deep learning method called *metric learning*, applied to the problem of motion segmentation. We will introduce it and discuss the approach in more detail in Section 3.7 and **Paper D**.

CHAPTER 3

3D Reconstruction of Static and Dynamic Scenes

A 3D reconstruction method aims to recover the 3D geometric structure of the scene, possibly together with its photometric characteristics, from a given set of sensor readings. The commonly used sensors are visual cameras, infrared cameras, LiDARs (light detection and ranging), and radars. The primary focus of this thesis is on cameras; however, it is possible to extend some of the presented approaches to other types of sensors (see Section 5.1). Additionally, we assume that only one camera is used to record a sequence of images. Extending the presented methods to multiple cameras is straightforward (see Section 5.1). Finally, we focus on reconstructing only the geometry of the scene, and not the photometric characteristics. The geometry of the scene often serves as a basis in photometric reconstruction [175].

Sections 3.1—3.3 establish the fundamental context for the methods developed in this thesis, while Sections 3.4—3.7 focus on the specific problems that the proposed methods address.

Section 3.1 formalizes the typical output of a reconstruction pipeline. It discusses different models to represent the geometry of a 3D scene and the role of camera geometry estimation in 3D reconstruction. Sections 3.2 and 3.3 discuss the methodologies that make up the standard pipeline for reconstructing

static 3D scenes from multiple images.

Section 3.4 discusses the first focus problem of this thesis, camera calibration. Accurate camera calibration is essential for all stages of geometric reconstruction, as errors in the intrinsics directly affect the quality of *e.g.* the 3D structure and camera poses. The initialization stage can be a weak point in calibration. This section reviews the standard calibration pipeline and introduces a method proposed in this thesis addressing initialization.

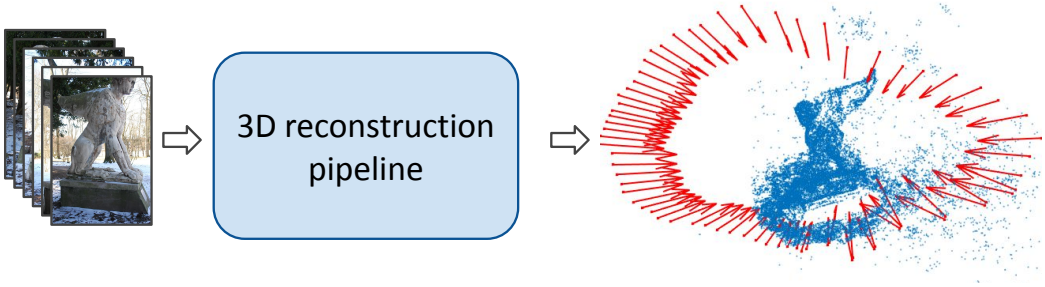
Sections 3.5 and 3.6 discuss the second focus problem of this thesis, rotation averaging. Rotation averaging plays a central role in the so-called global SfM pipelines that obtain globally consistent extrinsics from noisy pairwise relative estimates. The relative rotation estimates are anisotropically noisy because of the inherent structure of the two-view reconstruction problem. Outliers are also very common, affected by *e.g.* symmetries present in the scenes. The sections give an overview of the prior work and introduce the methods proposed in this thesis to address the aforementioned challenges.

Section 3.7 discusses the third focus problem of this thesis, motion segmentation. Unlike static scenes, dynamic scenes involve moving objects, posing additional challenges for reconstruction. Motion segmentation can be crucial in addressing these challenges. This section particularly focuses on dynamic scenes composed of multiple independently moving rigid bodies. It presents classic methodologies for simultaneous segmentation and motion model estimation and introduces a metric learning-based approach proposed in this thesis.

3.1 Scene and camera geometry

What do we mean by the geometric structure of the scene that we wish to obtain with the 3D reconstruction? The output geometry (or geometric model) can be represented in different ways, and it can be classified as either an *explicit* or *implicit* model. Popular examples of *explicit models* include: a set of 3D points or point cloud, voxels (cubic primitives), 3D line segments, polygon (typically triangle) mesh, and, related to this category, 3D Gaussians [177] that also contain photometric information. Common examples of *implicit models* include functions that for a 3D point query return: occupancy of an object at this point [178], signed distance to the object's surface where the sign indicates if the point is inside or outside (signed distance fields) [179],

Figure 3.1: Recovering the 3D geometry of the scene and cameras is the main goal of 3D reconstruction. Here we show the reconstruction for one of the standard datasets obtained by [48], [176].



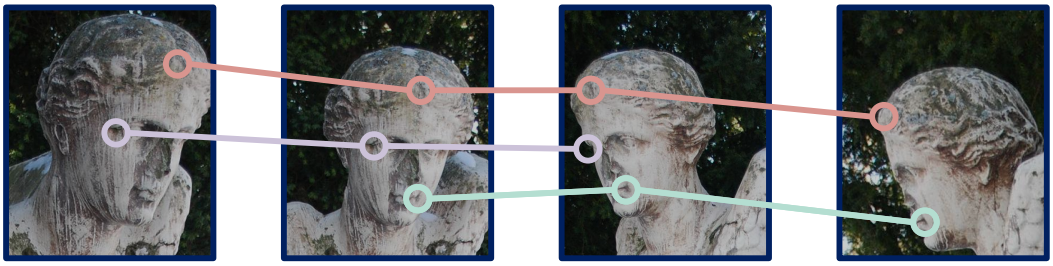
and, related to this category, neural radiance fields [175] that combine both geometric (density of the point) and photometric (color of the point) outputs while also modeling view dependency. The choice of the model typically depends on the application. It is possible to convert from one representation to another, possibly with some loss of information. In this thesis, we focus on point data, denoted $\{\mathbf{X}_j\}$. A point cloud is, in a way, the simplest geometric representation. This simplicity, however, allows obtaining elegant methods of multiple view geometry [24] that are often computationally cheap and reliable. The point cloud also provides a good base for recovering a more complex geometry and photometry of the scene [175], [177].

In 3D reconstruction, we also obtain camera geometries, namely intrinsics—the projection model $\pi(\cdot)$, its parameters, and the calibration matrix $\mathtt{K}$—as well as extrinsics—the camera poses $\{\mathtt{R}_i, \mathbf{t}_i\}$ (see Section 2.1). A common approach is to *e.g.* eliminate the scene geometry from the estimation and compute camera poses first. Therefore, accurate recovery of the camera geometry is essential for a successful scene reconstruction.

3.2 Reconstructing static scenes

Figure 3.1 illustrates what we have discussed so far, which is the goal of a general reconstruction pipeline that is of interest to this thesis. Now, let us assume that the input images are processed in a way that gives a set of corresponding 2D locations of tracked points (see Figure 3.2 for an illustration).

Figure 3.2: Point tracks: a set of corresponding 2D point locations in the input images.



We will denote this set $\Omega \subset \{1, \cdots, n\} \times \{1, \cdots, m\}$, where n is the number of images, and m is the number of unique 3D points. A pair (i, j) belongs to Ω if the j^{th} 3D point is observed in the i^{th} image as a 2D observation $\mathbf{u}_{i,j}$. There exist many challenges in estimating these *point tracks*. We discuss them in some more details in Sections 2.3 and 3.3. Let us assume a static scene, *i.e.* each 3D point $\mathbf{X}_j$ that we want to recover has the same 3D coordinates in all images $1, \ldots, n$.

Metric reconstruction

The problem of reconstructing the geometry of the static scene $\{\mathbf{X}_j\}_{j=1}^m$, $\{(\mathtt{R}_i, \mathbf{t}_i)\}_{i=1}^n$ from 2D observations $\{\mathbf{u}_{i,j}\}_{(i,j)\in\Omega}$ is typically formulated by minimizing the sum of reprojection errors

$$\min_{\substack{\{\mathbf{X}_j\},\mathtt{K},\{(\mathtt{R}_i,\mathbf{t}_i)\} \\ m R_i \in \mathrm{SO}(3)}} \sum_{(i,j)\in\Omega} \rho\left(\left\|\mathtt{K}\pi\left(\mathtt{R}_i\mathbf{X}_j + \mathbf{t}_i\right) - \mathbf{u}_{i,j}\right\|\right). \tag{3.1}$$

If optimization is non-robust, $\rho(x) = x^2$, otherwise it is a robust kernel function (see Section 2.5). Here, recall also that $\pi(\cdot)$ is the projection function, $\mathtt{K}$ is the calibration matrix, $(\mathtt{R}_i, \mathbf{t}_i)$ are camera poses, and $\mathbf{X}_j$ are the 3D coordinates of the scene points, and the objective form in (3.1) is obtained from maximum likelihood (see Section 2.5) with an assumption that the 2D residuals follow an i.i.d. standard Gaussian noise distribution. Solving (3.1) gives the so-called *metric reconstruction*. Note that it can be recovered up to an arbitrary global rotation $\mathtt{Q} \in \mathrm{SO}(3)$ (since $\mathtt{R}_i\mathbf{X}_j + \mathbf{t}_i = (\mathtt{R}_i\mathtt{Q}^\top)(\mathtt{Q}\mathbf{X}_j) + \mathbf{t}_i$),

scaling s ($\mathtt{R}_i\mathbf{X}_j + \mathbf{t}_i \sim \mathtt{R}_i(s\mathbf{X}_j) + (s\mathbf{t}_i)$), and translation $\mathbf{v}$ ($\mathtt{R}_i\mathbf{X}_j + \mathbf{t}_i = \mathtt{R}_i(\mathbf{X}_j + \mathbf{v}) + (\mathbf{t}_i - \mathtt{R}_i\mathbf{v})$). Optimization of the metric reconstruction objective (3.1) is called *bundle adjustment* (BA) [21], [180], or metric BA. This optimization is most commonly done with the local second-order optimization methods for non-linear least squares [119] and their robust versions [21] (see also Section 2.5).

Projective reconstruction

While metric reconstruction is most often the goal, in certain applications and pipelines, a *projective reconstruction* is also useful to consider if a pinhole camera is assumed. This reconstruction can be obtained as an intermediate result and later upgraded into metric reconstruction [181], [182]. Let $\mathtt{P}_i = \mathtt{K}\begin{pmatrix}\mathtt{R}_i & \mathbf{t}_i\end{pmatrix}$, where $\mathtt{P}_i$ are sometimes referred to as camera matrices. Projective reconstruction is a set of camera matrices $\{\mathtt{P}_i\}$ and the scene points in homogeneous representation $\{\hat{\mathbf{X}}_j\}$, that are defined up to projective transformation. This is because projective reconstruction has an intrinsic ambiguity—for any $\mathtt{A} \in \mathbb{R}^{4\times4}$ that is invertible, the following holds

$$\mathtt{P}_i\hat{\mathbf{X}}_j = \underbrace{\mathtt{P}_i\mathtt{A}^{-1}}_{\tilde{\mathtt{P}}_i}\underbrace{\mathtt{A}\hat{\mathbf{X}}_j}_{\tilde{\mathbf{X}}_j}, \tag{3.2}$$

meaning that the projected points produced with $(\{\mathtt{P}_i\}, \{\hat{\mathbf{X}}_j\})$ are the same as with $(\{\tilde{\mathtt{P}}_i\}, \{\tilde{\mathbf{X}}_j\})$. Hence, no matter how many points and views are observed, there is always a space of equivalent solutions producing exactly the same projections. In contrast, in metric reconstruction, as mentioned earlier, (3.2) holds only if $\mathtt{A}$ is a similarity transformation $\mathtt{A} = \begin{bmatrix} s\mathtt{Q} & \mathbf{v} \\ \mathbf{0}_3^\top & 1 \end{bmatrix}$.

The problem of projective reconstruction can be formulated as follows

$$\min_{\{\mathbf{X}_j\},\{\mathtt{P}_i\}} \sum_{(i,j)\in\Omega} \rho\left(\left\|\pi\left(\mathtt{P}_i\mathbf{X}_j\right) - \mathbf{u}_{i,j}\right\|\right), \tag{3.3}$$

where camera decomposition and hence orthonormality constraints are omitted, and $\pi(\cdot)$ is assumed to be a pinhole projection. The optimization of (3.3) is called *projective BA* [183]. It can similarly be done using local optimization methods for non-linear least squares. However, the simplified structure of the problem as compared to metric BA, namely the absence of the orthonormal-

ity constraints, makes it possible to adopt other techniques such as non-linear variable projection [183].

When some of the parameter subsets are known or fixed, we obtain two important sub-problems: triangulation and resectioning, which are discussed in the next paragraphs.

Triangulation If the camera calibration and camera poses $\{(\mathtt{R}_i, \mathbf{t}_i)\}$ are known or fixed, the problem in (3.1) reduces to searching for the optimal 3D structure $\{\mathbf{X}_j\}$ and is called *triangulation* (or *intersection*). Since 3D points are independent of each other, the problem is simplified as

$$\mathbf{X}_j = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \rho\left(\left\|\pi\left(\mathtt{R}_i\mathbf{X} + \mathbf{t}_i\right) - \mathbf{x}_{i,j}\right\|\right) \quad j = 1, \ldots, m \tag{3.4}$$

where $\mathbf{x}_{i,j} = \mathtt{K}^{-1}\mathbf{u}_{i,j}$. Apart from the typical l_2 minimization of reprojection error [85], [184], some works address different metrics. Examples are algebraic error minimization [84], midpoint method, l_1 and l_∞ minimization of angular reprojection errors [185], [186]. These provide certain advantages such as global optimality, robustness and rotation invariance.

Resectioning If the 3D point coordinates $\{\mathbf{X}_j\}$ are known or fixed, the problem in (3.1) reduces to searching for the best camera poses minimizing the reprojection error, and is called *resectioning* or *camera pose estimation*. Similarly to triangulation, this problem can be equivalently decomposed into independent sub-problems since cameras do not communicate with each other once the 3D structure is fixed. In the calibrated scenario, this gives

$$(\mathtt{R}_i, \mathbf{t}_i) = \underset{\mathtt{R}\in\mathrm{SO}(3),\mathbf{t}}{\operatorname{argmin}} \sum_j \rho\left(\left\|\pi\left(\mathtt{R}\mathbf{X}_j + \mathbf{t}\right) - \mathbf{x}_{i,j}\right\|\right) \quad i = 1, \ldots, n \tag{3.5}$$

This problem is also sometimes called Perspective n-Point (PnP) problem [24], [187], [188].

3.3 Structure from motion and other ways to initialize BA

As mentioned earlier, BA is a local optimization method and requires a good starting point. Common strategies for initialization include incremental

Table 3.1: Rough categorization of SfM techniques. The methods are also split using "/" based on the type of reconstruction returned: projective / (near) metric. "transl. avg." is translation averaging, "known rot." means known rotation problem, and "hierar." means hierarchical.

	Sequential	Global	Hybrid
Structure & Motion	*Incremental SfM* - / [33]–[35], [49]	*Factorization* [182], [183], [189]–[194] / [195]	- / hierar. [196], [197] l_∞-SLAM [198]
Motion first	*Global SfM* [199] / transl. avg. [200]–[203] known rot. [48], [52], [204], [205]		- / HARA [206] HSfM [207]
Structure first	*Structure without Motion* - / general [208], planar [209]		

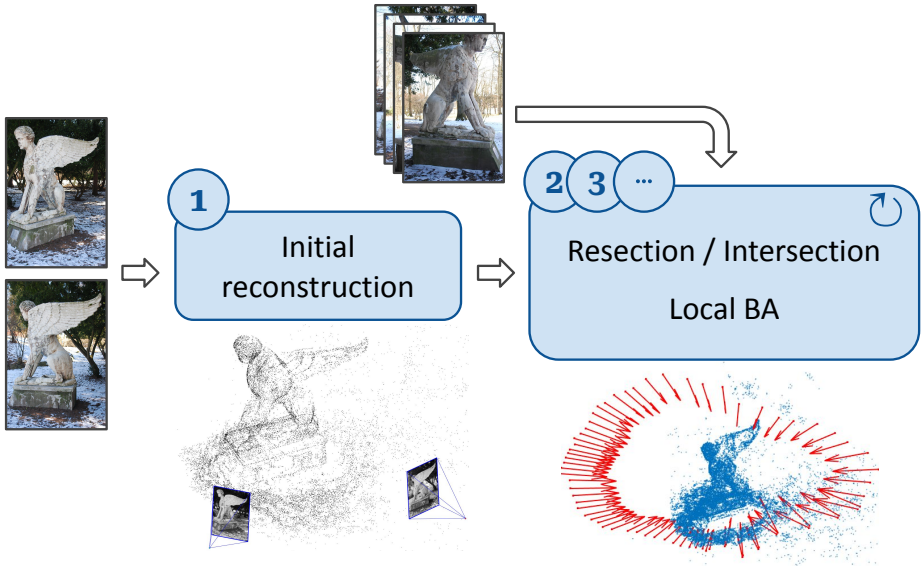
(or sequential) structure from motion, global structure from motion, and factorization-based reconstruction. The term *structure from motion* (SfM) seems to be overused in the computer vision literature, often referring to different methodologies for estimating camera motion and 3D structure. Sometimes it refers to both the initialization pipeline and the refinement stage (BA) because of the alternating nature of the underlying method (see *e.g.* incremental SfM, see Section 3.3). The name, however, suggests that SfM should refer to the family of approaches that estimate camera motion first, followed by recovering the 3D structure. There also exist *structure and motion* methods that recover both camera motions and 3D structure simultaneously, and there also exist techniques that recover structure without knowledge of the camera motion. We refer to Table 3.1 and, *e.g.*, [210] for a more thorough discussion.

Incremental SfM

Incremental SfM is arguably the most common initialization pipeline in traditional 3D reconstruction. A general-purpose SfM and multi-view stereo framework called COLMAP [35] that implements incremental SfM is a very popular tool in the computer vision community. The core idea behind incremental SfM is to process the input images sequentially (see Figure 3.3). The algorithm can be roughly described as follows:

1. Two images are used to estimate the initial 3D structure and the cor-

Figure 3.3: Exemplar incremental SfM pipeline. First, initial reconstruction is obtained from two images. Then, each new image is matched against the estimated-so-far 3D structure, the corresponding camera is estimated via resection, and the new 3D structure is estimated via intersection (triangulation). Local BA is occasionally performed.



responding camera poses (see Section 2.3). Here, the camera poses are inferred from the relative pose and are therefore defined up to an arbitrary Euclidean transformation of both coordinate frames and scaling of the relative translation part.

2. Every new image is matched against the estimated-so-far 3D structure, the matching output is used to estimate the corresponding camera pose, and the 3D structure is augmented with the new structure obtained from new matches (see Section 3.2). The process is repeated until all images (or all "skeletal" images [180], [211]) are processed.

3. Bundle adjustment is performed occasionally on subsets of images to reduce potential *drift* (accumulating errors) and to refine the reconstruction obtained thus far.

It is common to use incremental SfM with both ordered and unordered image collections. The latter scenario [180], [212] requires an additional step of

selecting the order of image processing. Due to its sequential nature, however, this approach is particularly attractive in applications where data comes in the same manner *e.g.* real-time data acquisition in autonomous navigation. In relation to that, incremental SfM has a close connection to simultaneous localization and mapping (SLAM). For example, monocular visual SLAM (vSLAM) directly implements all of the components of incremental SfM. The key difference is that vSLAM is optimized for speed (in contrast to accuracy in incremental SfM), being constrained to real-time processing. Therefore, it often requires known intrinsics which can be pre-estimated with camera calibration.

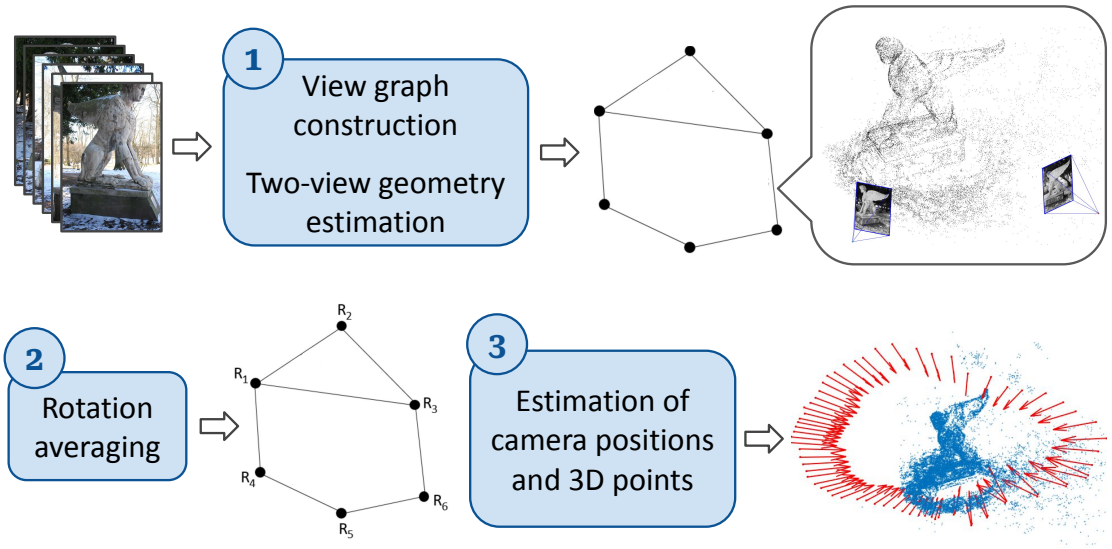
One of the challenges in incremental SfM is balancing the need to perform refinement with the need to reduce the runtime. If BA runs too often, in the most extreme case after every new camera being added to the reconstruction, the algorithm will be very slow. On the other hand, reducing the frequency of BA increases the risk of drift. Overall, to achieve a good enough initialization of the reconstruction, incremental SfM will need to run BA from time to time, which will affect runtime in more challenging scenes and larger scales [213].

Global SfM

The core idea behind *global SfM* is to process all images simultaneously [48], [205] (see Figure 3.4), in contrast to sequential processing. The global SfM algorithm can be roughly described as follows:

1. The camera graph $\mathcal{G}$ with n nodes (corresponding to n cameras) is constructed:
    - One way is to have all edges between all pairs of images/nodes.
    - Another option is to identify pairs of images with overlapping fields of view [214] and establish the edges between those pairs only; this would lead to a more computationally efficient approach.
2. The two-view reconstruction pipeline is run on all pairs of images where an edge exists (if unsuccessful, the edge is removed), and the estimated relative poses are attributed to the edges between the corresponding camera nodes.
3. Rotation averaging is performed from a set of relative poses, giving absolute camera poses (this problem will be covered in more detail in Section 3.5). If camera intrinsics are not available, rotation averaging

Figure 3.4: Exemplar global SfM pipeline. First, the view graph is constructed for the input images, and the two-view optimizations are run on all pairs of images where an edge exists, refining the view graph. Second, rotation averaging is run on the view graph, returning absolute camera orientations. The third step is obtaining camera positions and 3D points (see Section 3.3).



may be done simultaneously with calibration.

4. Finally, absolute camera positions and 3D points are estimated:

   - One option is to perform translation averaging [200]–[203], [215] followed by triangulation.
   - A more principled approach is through simultaneous estimation of camera positions and 3D points [48], [52], [176], [198], [204], [205], [216], referred to as the *known rotation problem*. *E.g.*, it was shown that under l_∞-norm (minimizing the maximum reprojection error), efficient globally optimal solutions can be obtained [217]–[220], with faster algorithms [221]–[224] proposed further. Recently, the problem was revisited with local optimization of robust lifted [202] angular errors, referred to as *global positioning* [52].

In large-scale reconstruction scenarios, a hybrid approach may be preferred. If the scene can be split into smaller sub-scenes, a global SfM can be run

on each of the sub-scenes, followed by the incremental merging of the sub-reconstructions into a single reconstruction. A tricky question is how to perform splitting "optimally" [214].

Factorization-based reconstruction

Similarly to global SfM, *factorization* (or *matrix factorization*) methods process all images simultaneously. The optimization is, however, performed on image points directly. The idea is to reformulate the projective reconstruction objective (3.3) as a factorization of the data matrix of 2D point tracks into motion and structure matrices; hence the name of the approach. If affine cameras are assumed, the objective in (3.3) can, in fact, be re-written equivalently as follows

$$\min_{\mathtt{P},\mathtt{X}} \|\mathtt{W} \odot (\mathtt{M} - \mathtt{P}\mathtt{X})\|_F^2, \tag{3.6}$$

where the data matrix $\mathtt{M} \in \mathbb{R}^{2n \times m}$ contains Euclidean image points

$$\mathtt{M} = \begin{pmatrix} \underline{\mathbf{u}}_{1,1} & \underline{\mathbf{u}}_{1,2} & \cdots & \underline{\mathbf{u}}_{1,m} \\ \underline{\mathbf{u}}_{2,1} & \underline{\mathbf{u}}_{2,2} & \cdots & \underline{\mathbf{u}}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{\mathbf{u}}_{n,1} & \underline{\mathbf{u}}_{n,2} & \cdots & \underline{\mathbf{u}}_{n,m} \end{pmatrix} \tag{3.7}$$

and may possibly contain missing data; the unknown motion $\mathtt{P} \in \mathbb{R}^{2n \times 4}$ and structure $\mathtt{X} \in \mathbb{R}^{4 \times m}$ matrices are

$$\mathtt{P} = \begin{pmatrix} \mathtt{P}_1^{(1:2)} \\ \mathtt{P}_2^{(1:2)} \\ \vdots \\ \mathtt{P}_n^{(1:2)} \end{pmatrix} \qquad \mathtt{X} = \begin{pmatrix} \hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2 & \dots & \hat{\mathbf{X}}_m, \end{pmatrix} \tag{3.8}$$

and $\mathtt{W}$ encodes visibility. In the idealistic scenario, where all scene points are visible in all images and there are no outliers, the reconstruction is obtained via SVD [65]. Otherwise, iterative methods [191], [192] are applied.

If a more realistic perspective projection is assumed, matrix factorization of the observation matrix into motion and structure does not hold anymore—the division by depth breaks linearity. The common approach is to pre-multiply

the terms in (3.3) with the depth variables and form the new objective that can compactly be written as following

$$\min_{\Lambda,\mathtt{P},\mathtt{X}} \|\mathtt{W} \odot ((\Lambda \otimes \mathbf{1}_3) \odot \mathtt{M} - \mathtt{PX})\|_F^2, \tag{3.9}$$

where the data matrix $\mathtt{M} \in \mathbb{R}^{3n \times m}$ contains homogeneous image points, the unknown motion matrix is $\mathtt{P} \in \mathbb{R}^{3n \times 4}$, $\Lambda \in \mathbb{R}^{n \times m}$ is an unknown depth matrix, and $\mathbf{1}_3$ is a vector of ones

$$\Lambda = \begin{pmatrix} \lambda_{1,1} & \lambda_{1,2} & \dots & \lambda_{1,m} \\ \lambda_{2,1} & \lambda_{2,2} & \dots & \lambda_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n,1} & \lambda_{n,2} & \dots & \lambda_{n,m} \end{pmatrix}, \qquad \mathbf{1}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{3.10}$$

To avoid trivial solutions to (3.9), *i.e.* $(\Lambda, \mathtt{P}, \mathtt{X}) = (\mathtt{0}, \mathtt{0}, \mathtt{0})$, iterative methods are proposed that start with $\lambda_{i,j} = 1$. Another branch of methods proposes to alternatively optimize the so-called object space error [182] using the fact that $\lambda_{i,j} = \mathtt{P}_i^{(3)} \mathbf{X}_j$ resulting in

$$\min_{\mathtt{P},\mathtt{X}} \sum_{(i,j)\in\Omega} \|\mathtt{P}_i^{(3)} \mathbf{X}_j \ \mathbf{u}_{i,j} - \mathtt{P}_i^{(1:2)} \mathbf{X}_j\|^2. \tag{3.11}$$

To avoid trivial solutions $\mathtt{P} = \mathtt{0}, \mathtt{X} = \mathtt{0}$, different regularization techniques [182], [194] are proposed, so that the final objective is

$$\min_{\mathtt{P},\mathtt{X}} \sum_{(i,j)\in\Omega} (1-\eta)\|\mathtt{P}_i^{(3)} \mathbf{X}_j \ \mathbf{u}_{i,j} - \mathtt{P}_i^{(1:2)} \mathbf{X}_j\|^2 + \eta \mathcal{R}\left(\mathtt{P}_i, \mathbf{X}_j, \underline{\mathbf{u}}_{i,j}\right), \tag{3.12}$$

where $\mathcal{R}$ can be *e.g.* an affine error $\mathcal{R}(\mathtt{P}, \mathbf{X}, \mathbf{m}) = \|\mathtt{P}^{(1:2)}\mathbf{X} - \mathbf{m}\|^2$ [182] or an exponential function $\mathcal{R}(\mathtt{P}, \mathbf{X}, \mathbf{m}) = \exp\left(-\frac{\mathbf{m}^\top \mathtt{P}^{(1:2)}\mathbf{X} + \mathtt{P}^{(3)}\mathbf{X}}{\sqrt{\|\mathbf{m}\|^2+1}}\right)$ [194].

Establishing point tracks

Obtaining many accurate point tracks is essential for factorization-based methods to work. Among all areas of 3D computer vision, point tracking together with image matching are where deep learning has probably shown the biggest improvements over traditional approaches, which might be due to the connection to visual perception where neural networks demonstrate outstanding

performance. Similarly to SfM, tracking methods can be divided into sequential and global. In sequential tracking, point correspondences (see Section 2.3) are established incrementally across image pairs forming tracks that can be additionally refined [225], [226]. For video sequences, optical flow-based methods can be used [227] but they are more prone to drift and cannot handle occlusions. The sequential processing can also be substituted with neural networks like TAP-Net [228], PIPs [229], [230] and TAPIR [231]. In contrast, global tracking methods formulate the problem of jointly estimating the 2D locations across all frames and are typically optimization-based like Omnimotion [232] or DINOTracker [233].

3.4 Camera calibration

Camera calibration deals with estimating intrinsics—the type and parameters of the projection function $\pi(\cdot)$ as well as the calibration matrix $\mathtt{K}$ (discussed in Section 2.1). Knowing these parameters means understanding which ray in 3D space produces which pixel in an image. This knowledge helps in many vision applications such as visual odometry and SLAM, object localization and tracking, SfM, rendering for augmented reality *etc.* It may be performed *offline* or *online*.

Offline calibration refers to obtaining intrinsics beforehand using the predetermined 3D structure, most commonly one or multiple planar targets with fiducial markers—such as checkerboards [55], AprilTags [112], ArUco [113], *etc.*—for an easy detection (see also discussion on establishing 2D-3D correspondences for calibration in Section 2.4).

Online calibration (sometimes also called *self-calibration* or *auto-calibration*) refers to estimating intrinsics (either separately or together with the other unknowns) from the same data that is used in the end application. Often times, for example in SfM scenarios, this means that the 3D structure is unknown, therefore online calibration techniques utilize other clues such as geometry of vanishing points [234]–[237] or learned from pre-calibrated data [238]–[240]).

As mentioned earlier, camera calibration is the necessary step in most 3D vision pipelines. Accurate and reliable calibration minimizes the risk of error propagation through the multiple stages of vision algorithms. For that reason, if offline calibration can be performed, it is usually preferred. And in this thesis, we focus on analyzing this approach.

Offline camera calibration pipeline

Typically, no learning is involved in offline calibration, since (1) the fiducial marker sets are specifically designed for maximizing the detection speed and accuracy [112], [115], (2) the utilized target objects are of known geometry (most commonly, planes [55], [109] or polyhedrons [115]), and (3) the geometric relation between 3D points and their projections is well understood. To describe a typical calibration pipeline, we borrow the "recommended procedure" from probably the most cited work on camera calibration—Zhang's method [55]:

1. Print a pattern and attach it to a planar surface;
2. Take many images of the model plane under different orientations by moving either the plane or the camera;
3. Detect the feature points in the images;
4. Estimate parameters of $\mathtt{K}$, $\{(\mathtt{R}_i, \mathbf{t}_i)\}$ using the closed-form solution (that minimizes an algebraic error);
5. Estimate the coefficients of the radial distortion using another closed-form (linear least-squares) solution;
6. Refine all parameters by minimizing the reprojection error (as in camera resectioning).

This pipeline works generally well with pinhole cameras with minor distortions, but not with fisheye cameras, catadioptric cameras or, in general, any camera with significant lens distortion that can not be approximated by pinhole projection model. For that reason many camera models [56], [69], [77], [79], [105], [241]–[248] and calibration frameworks [56], [69], [77], [79], [105], [242]–[248] are proposed to accommodate for different cameras.

Let us now imagine a scenario where we would like to use a camera with unknown model. Or if we want to use multiple different cameras—this is a common scenario in robotics and autonomous navigation. We would have to search for the best framework to calibrate each camera. This is a very tedious process. Moreover, if one camera has wide FOV and high lens distortion, there is a chance that the calibration framework would fail, requiring manual intervention. Upon inspection, we might find out that the initial guess was

too far for the refinement to converge to a good minimum, and we would have to manually adjust it. This is the problem addressed in **Paper A**. We provide more details about the approach in the next section.

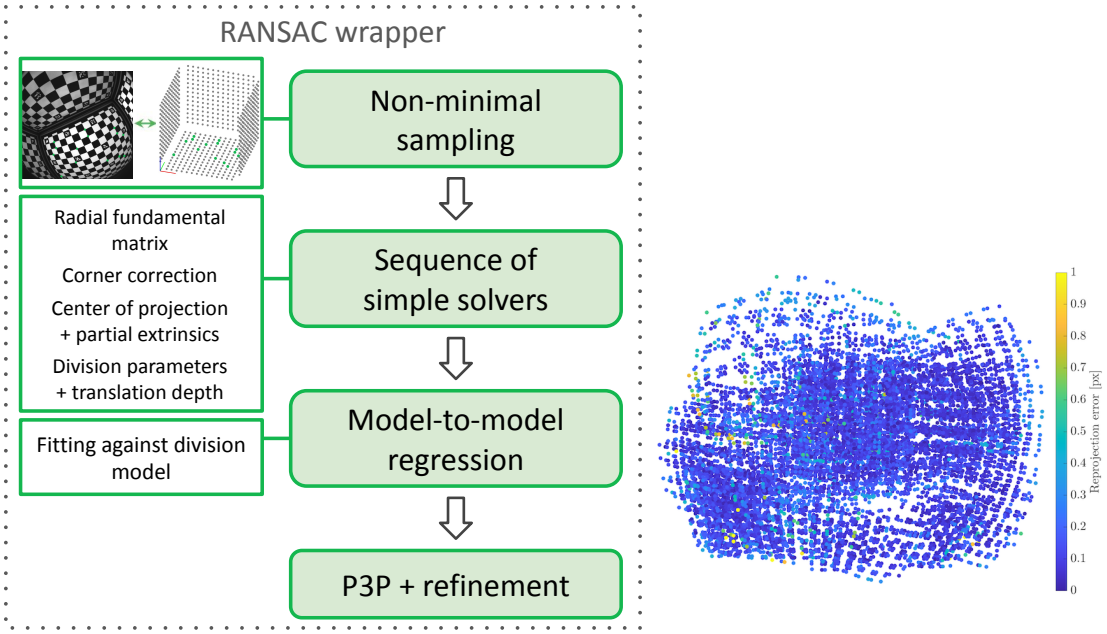
Towards universal calibration of central cameras

Cameras with wide fields of view, in particular fisheyes and catadioptric rigs [249] usually require highly nonlinear models with many parameters [59], [66], [67], [69], [250], [251], where each model comes with their own set of assumptions. The high projection distortion is also more likely to affect the accuracy of fiducials detection. Our goal is to address these difficulties, motivated by the existence of solvers for the division model [67] that, from our prior experiments, provide a sufficiently good initialization for all kinds of cameras. We therefore design a unified robust framework for calibrating any central projection camera, including narrow and wide field-of-view cameras. The method is detailed in **Paper A** and is explained below.

We look at the relation between the scene planar points and their radial projections $\mathbf{u}^\top \mathtt{F}_\mathtt{H} \mathbf{x} = 0$, where $\mathtt{F}_\mathtt{H}$ is a radial fundamental matrix [105] (see also Section 2.4 that shows how this relation can be derived). We can estimate $\mathtt{F}_\mathtt{H}$ from at least seven point-to-target point correspondences and recover the projection center by computing the left null-space of $\mathtt{F}_\mathtt{H}$. In general, the radial fundamental matrix can be expressed as $\mathtt{F}_\mathtt{H} = [\mathbf{e}]_\times \mathtt{K}\mathtt{H}$. Let us assume orthogonal and square pixels (in **Paper A**, we only assume orthogonal pixels, which is sufficient for initialization; the potentially non-square pixels are handled by sampling over the interval of the most plausible aspect ratios, from 0.5 to 2). Therefore the decomposition of $\mathtt{F}_\mathtt{H}$ can be simplified as $\mathtt{F}_\mathtt{H} \sim [\mathbf{e}]_\times \operatorname{diag}(1, 1, 0)\mathtt{H}$ (see Section 2.4). In particular, the first row of $\mathtt{F}_\mathtt{H}$ is $s \begin{pmatrix} -r_{21} & r_{22} & -t_2 \end{pmatrix}$, and the second row is $s \begin{pmatrix} r_{11} & r_{12} & t_1 \end{pmatrix}$, where s is an unknown scale. This allows us to recover the camera rotation matrix $\mathtt{R}$ and the first two components of the camera translation $\mathbf{t}$ using the quadratic constraints obtained from the orthonormality of $\mathbf{r}_1$ and $\mathbf{r}_2$.

Note that so far we were able to recover partial intrinsics and extrinsics without any assumption on the projection model. Now, in **Paper A**, we show empirically that the back-projection division model $\psi(r) = 1 + \sum_{n=1}^{N} \lambda_n r^{2n}$ (see Section 2.1) can approximate sufficiently well (hence initialize) any type of the most commonly used central projection models, including wide field-of-view cameras, such as fish eye cameras. The model is also particularly

Figure 3.5: BabelCalib overview: (left) proposed pipeline, and (right) quality of camera pose estimation on test images using BabelCalib output, where all detected points with color-coded reprojection errors are shown (yellow corresponds to higher errors, while blue corresponds to lower errors).



attractive as it allows to construct a simple linear solver [67] for recovering the model parameters λ_n together with the last component of $\mathbf{t}$ and the pixel focal length f. Finally, the poses for the remaining images are computed using P3P (Perspective-3-Point) [252] from a sample of three 2D-3D correspondences.

To accommodate different target projection models that could be used during refinement, we propose the model-to-model regression step, which is the optimization of the following non-linear least squares objective

$$\min_{\theta} \sum_{i} (\phi_\theta(r_i, z_i) - r_i)^2, \tag{3.13}$$

given a set of pairs (r_i, z_i), where $z_i = \psi(r_i)$ are the division model "depths". For most target projection models $\phi_\theta(\cdot, \cdot)$, the problem can be simplified as, in fact, linear least squares. For instance, Kannala-Brandt model $\phi_\theta(R, Z) =$

$\zeta + \sum_{n=1}^{N} k_n \zeta^{2n+1}$, $\zeta = \text{atan2}(R, Z)$ leads to the following optimization

$$\min_{\theta} \sum_{i} \left(\zeta_i + \sum_{n=1}^{N} k_n \zeta_i^{2n+1} - r_i \right)^2 \qquad \zeta_i = \text{atan2}(r_i, z_i), \tag{3.14}$$

which can be re-written as a linear least squares problem

$$\begin{bmatrix} & \vdots & \\ \zeta_i^3 & \cdots & \zeta_i^{2N+1} \\ \zeta_{i+1}^3 & \cdots & \zeta_{i+1}^{2N+1} \\ & \vdots & \end{bmatrix} \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \end{pmatrix} \approx \begin{pmatrix} \vdots \\ r_i - \zeta_i \\ r_{i+1} - \zeta_{i+1} \\ \vdots \end{pmatrix}. \tag{3.15}$$

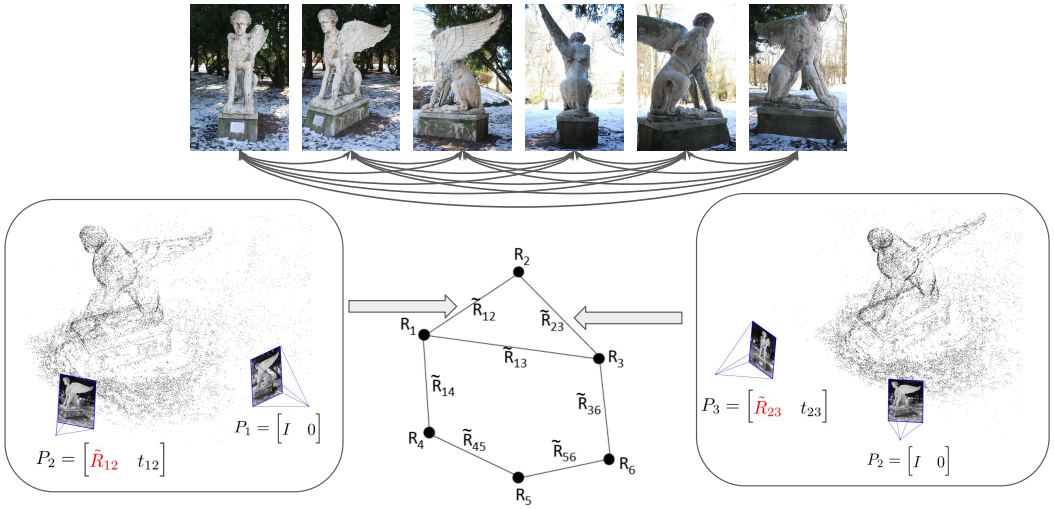
We can do the same for the Brown-Conrady [68], unified camera [69], and extended unified camera [58] models.

The overview of the pipeline is shown in Figure 3.5 to the left. As can be seen, we use the proposed sequence of solvers inside (locally optimized) RANSAC. While we expect a very high inlier ratio, it will not be 100% in general; hence, we use a robust estimation wrapper. A high inlier ratio, on the other hand, allows us to use non-minimal sample sizes (in particular, we use 14 correspondences) and refine the radial fundamental matrix by minimizing the distances to the "radial epipolar" lines (referred to as corner correction in **Paper A**). Overall, **Paper A** proposes a fully automatic calibration framework that only requires the user to choose the target camera projection model. This can also potentially be automated via model selection. We evaluate the proposed method on the downstream task of absolute pose estimation and demonstrate its state-of-the-art performance. An example of a qualitative result is illustrated in Figure 3.5 to the right.

3.5 Rotation averaging

As highlighted earlier, *rotation averaging* is an important problem that has many applications in computer vision and robotics. It is a key building block in SLAM, global and hybrid SfM pipelines. In general, the goal is to estimate a set of absolute rotations $\{\mathtt{R}_i\}$ in the common CS given the set of noisy pairwise relative rotations $\{\tilde{\mathtt{R}}_{ij}\}$ in the respective local CSs, *i.e.* such that $\tilde{\mathtt{R}}_{ij} \approx \mathtt{R}_j \mathtt{R}_i^\top$. The problem is illustrated in Figure 3.6. In the outlier-free case, the typical

Figure 3.6: Rotation averaging as part of global SfM pipelines. For a given set of images, a camera graph is constructed where nodes encode cameras and edges encode pairwise relations—relative poses estimated from pairs of images. The goal is, given the relative rotations on the edges, to obtain the absolute orientations of the cameras on the nodes.



Given: relative rotations (edges)
Relative pose solutions in respective local coordinate systems

Compute: camera orientations (nodes)
Rotations in the global coordinate system

problem formulation is minimization of the sum of all distances between pairs of rotations where the corresponding relative measurements are available

$$\min_{\{\mathtt{R}_i \in \mathrm{SO}(3)\}} \sum_{(i,j)\in\mathcal{E}} d^2(\tilde{\mathtt{R}}_{ij}, \mathtt{R}_j \mathtt{R}_i^\top), \tag{3.16}$$

where $\mathcal{E}$ is the set of pairs with available relative rotations and can also be viewed as a set of edges in a graph (such as camera graph $\mathcal{G}$), and $d(\cdot)$ is the distance metric in SO(3). Since all absolute rotations are optimized simultaneously, rotation averaging has a potential to eliminate the incrementally increasing errors that are a typical problem in sequential SfM. The distance metric $d(\cdot)$ is typically one of:

- *Angular* distance [60], [131], [253], [254], which is also a geodesic distance

in SO(3):

$$d(\mathtt{R}_1, \mathtt{R}_2) = \|\omega(\mathtt{R}_1^\top \mathtt{R}_2)\| \,(=: \phi \in [0, \pi])\,, \tag{3.17}$$

where $\omega(\cdot)$ extracts the axis-angle representation vector *i.e.* such that $\omega(e^{[\boldsymbol{\omega}]_\times}) = \boldsymbol{\omega}$, and ϕ is the residual rotation angle.

- *Chordal* distance [156], [165], [255]–[257]:

$$d(\mathtt{R}_1, \mathtt{R}_2) = \|\mathtt{R}_1 - \mathtt{R}_2\|_F \left(= 2\sqrt{2}\sin(\phi/2)\right), \tag{3.18}$$

where the relation to the residual rotation angle can be seen from expanding the squared chordal distance as following

$$\|\mathtt{R}_1 - \mathtt{R}_2\|_F^2 = \underbrace{\|\mathtt{R}_1\|_F^2}_{3} + \underbrace{\|\mathtt{R}_2\|_F^2}_{3} - 2\,\underbrace{\langle \mathtt{R}_1, \mathtt{R}_2 \rangle}_{1+2\cos(\phi)}. \tag{3.19}$$

The minimizer of the chordal distances is the MLE assuming Langevin distribution [258] (also known as Von Mises-Fisher matrix distribution) on rotation matrices.

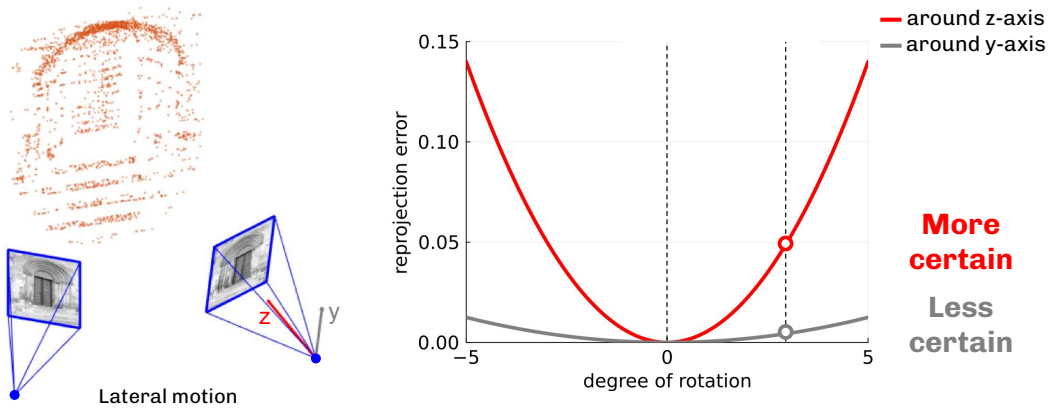
- *Quaternion* distance [259], [260]:

$$d(\mathtt{R}_1, \mathtt{R}_2) = \min(\|\mathbf{q}_1 - \mathbf{q}_2\|, \|\mathbf{q}_1 + \mathbf{q}_2\|) \,(= 2\sin(\phi/4))\,, \tag{3.20}$$

where $\mathbf{q}_1$ and $\mathbf{q}_2$ are the unit quaternion representations of $\mathtt{R}_1$ and $\mathtt{R}_2$, respectively.

For a more in-depth overview we refer to [260], [261]. Current methods mostly use various *isotropic* error measurements, *i.e.* that treat all dimensions equally. However, when considering vision applications of rotation averaging like visual SLAM or SfM, input relative rotations are not direct measurements but estimates obtained from processing image pairs via two-view optimization (discussed in Section 2.3). It was shown [157] that anisotropic errors naturally appear in two-view problems—when deviating from an optimal solution, the changes in the reprojection errors occur at different rates for different axes of rotational deviation (see Figure 3.7). The Hessian $\mathtt{H}$ of the two-view optimization can be used to approximate these rates and quantify uncertainty in the output parameters. This result can alternatively be obtained via covariance

Figure 3.7: Anisotropic noise in rotations naturally appears in two-view problems. This is a typical example of lateral motion. If an optimal relative rotation (*i.e.* that is optimized using a two-view pipeline like in Section 2.3) is slightly rotated around the z-axis, this will result in a higher reprojection error as compared to rotating around y-axis. This means a higher certainty for the angle around the z-axis and lower certainty for the angle around y-axis.



propagation to the MLE (see *e.g.* Section 5.2.2 in [24]). However, the two-view uncertainties are typically disregarded in the SfM pipelines. Zhang *et al.* [60] seem to be the first to use the pairwise estimation uncertainties in rotation averaging, where they in particular consider local optimization of angular distances. In **Paper B**, we look at the same problem but formulated using *chordal distances*. Prior work [156], [165] shows that rotation averaging with chordal distances

$$\min_{\{\mathtt{R}_i \in \mathrm{SO}(3)\}} \sum_{(i,j)\in\mathcal{E}} \|\tilde{\mathtt{R}}_{ij} - \mathtt{R}_j \mathtt{R}_i^\top\|_F^2, \tag{3.21}$$

which is a non-convex optimization problem that has many spurious local minima, can be relaxed into a convex optimization problem with conditioned guarantees of global optimality that are often satisfied in practice. We provide some key derivations that are also relevant to **Paper B**. From (3.19) we see that the terms $\|\tilde{\mathtt{R}}_{ij} - \mathtt{R}_j \mathtt{R}_i^\top\|_F^2$ can be reduced to $-\langle \tilde{\mathtt{R}}_{ij}, \mathtt{R}_j \mathtt{R}_i^\top \rangle$ (the constants are dropped as they do not affect optimization), and the problem can be

compactly written in matrix form

$$\min_{\mathsf{R}\in\mathrm{SO}(3)^n} -\langle \tilde{\mathsf{R}}, \mathsf{R}\mathsf{R}^\top \rangle, \tag{3.22}$$

where R stacks the unknown absolute rotations and $\tilde{\mathsf{R}}$ stacks the input relative rotations as following

$$\mathsf{R} = \begin{bmatrix} \mathtt{R}_1 \\ \mathtt{R}_2 \\ \vdots \\ \mathtt{R}_n \end{bmatrix} \qquad \tilde{\mathsf{R}} = \begin{bmatrix} 0 & \tilde{\mathtt{R}}_{12}^\top & \tilde{\mathtt{R}}_{13}^\top & \dots & \tilde{\mathtt{R}}_{1n}^\top \\ \tilde{\mathtt{R}}_{12} & 0 & \tilde{\mathtt{R}}_{23}^\top & \dots & \tilde{\mathtt{R}}_{2n}^\top \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathtt{R}}_{1n} & \tilde{\mathtt{R}}_{2n} & \tilde{\mathtt{R}}_{3n} & \dots & 0 \end{bmatrix}. \tag{3.23}$$

The relaxation is achieved by dropping the determinant constraints

$$\min_{\mathsf{R}\in\mathrm{O}(3)^n} -\langle \tilde{\mathsf{R}}, \mathsf{R}\mathsf{R}^\top \rangle, \tag{3.24}$$

and taking the dual twice as shown in Section 2.5. This gives the following SDP

$$\min_{\mathsf{X}\succeq 0} -\langle \tilde{\mathsf{R}}, \mathsf{X} \rangle \tag{3.25}$$

$$\text{s.t. } \mathtt{X}_{ii} = \mathtt{I}_3. \tag{3.26}$$

The relaxation was shown to be tight if all residual terms $\|\tilde{\mathtt{R}}_{ij} - \mathtt{R}_j\mathtt{R}_i^\top\|_F$ are bounded, where the bound depends on the algebraic connectivity of the camera graph $\mathcal{G}$ [156], [165]. In **Paper B** we show how to extend this convex relaxation approach to anisotropic rotation averaging setting, and we provide its overview in the next section. In **Paper C** we derive an efficient dedicated solver for the proposed formulation, which is also explained below.

Anisotropic rotation averaging

Let us show how to incorporate uncertainties in the input relative rotations into rotation averaging framework with chordal distances. The idea is to modify the objective in (3.22) while keeping its structure. The only way to do that is by replacing the data matrices $\tilde{\mathtt{R}}_{ij}$ in the objective terms $-\langle \tilde{\mathtt{R}}_{ij}, \mathtt{R}_j\mathtt{R}_i^\top \rangle$ with some other matrices, say $\mathtt{N}_{ij}$, that encode both relative rotations and uncertainties. Let $\mathtt{Q}_{ij} = \mathtt{R}_j\mathtt{R}_i^\top$ and consider a single two-view optimization problem

(hence we can for now drop the i, j indices)—let $\tilde{\mathtt{R}}$ be the local minimum, and $\Delta\boldsymbol{\omega}$ be the axis-angle vector of the residual rotation $\mathtt{Q}\tilde{\mathtt{R}}^\top$ (*i.e.* such that $\mathtt{Q} = e^{[\Delta\boldsymbol{\omega}]_\times}\tilde{\mathtt{R}}$). We obtain the following approximation for $\mathtt{M} = \frac{1}{2}\operatorname{tr}(\mathtt{H})\mathtt{I}_3 - \mathtt{H}$

$$\tfrac{1}{2}\Delta\boldsymbol{\omega}^\top\mathtt{H}\Delta\boldsymbol{\omega} \approx \operatorname{tr}(\mathtt{M}) - \langle\mathtt{M}\tilde{\mathtt{R}}, \mathtt{Q}\rangle\,. \tag{3.27}$$

To see this, let us first note that using the second-order Taylor approximation of the exponential map $\mathtt{Q}\tilde{\mathtt{R}}^\top \approx \mathtt{I}_3 + [\Delta\boldsymbol{\omega}]_\times + \frac{1}{2}[\Delta\boldsymbol{\omega}]_\times^2$ we can write

$$\begin{aligned}\langle\mathtt{M}\tilde{\mathtt{R}}, \mathtt{Q}\rangle &\approx \operatorname{tr}(\mathtt{M}) + \underbrace{\operatorname{tr}\left(\mathtt{M}\,[\Delta\boldsymbol{\omega}]_\times\right)}_{=0} + \tfrac{1}{2}\operatorname{tr}\left(\mathtt{M}\,[\Delta\boldsymbol{\omega}]_\times^2\right)\\ &= \operatorname{tr}(\mathtt{M}) - \tfrac{1}{2}\operatorname{tr}\left([\Delta\boldsymbol{\omega}]_\times^\top\,\mathtt{M}\,[\Delta\boldsymbol{\omega}]_\times\right),\end{aligned} \tag{3.28}$$

giving $\operatorname{tr}(\mathtt{M}) - \langle\mathtt{M}\tilde{\mathtt{R}}, \mathtt{Q}\rangle \approx \frac{1}{2}\operatorname{tr}\left([\Delta\boldsymbol{\omega}]_\times^\top\,\mathtt{M}\,[\Delta\boldsymbol{\omega}]_\times\right)$ up to second order terms. Inserting $\mathtt{M} = \frac{1}{2}\operatorname{tr}(\mathtt{H})\mathtt{I}_3 - \mathtt{H}$ into the approximation gives

$$\begin{aligned}\operatorname{tr}\left([\Delta\boldsymbol{\omega}]_\times^\top\left(\tfrac{1}{2}\operatorname{tr}(\mathtt{H})\mathtt{I}_3 - \mathtt{H}\right)[\Delta\boldsymbol{\omega}]_\times\right) &= \operatorname{tr}\left(\left(\mathtt{H} - \tfrac{1}{2}\operatorname{tr}(\mathtt{H})\mathtt{I}_3\right)[\Delta\boldsymbol{\omega}]_\times^2\right)\\ &= \operatorname{tr}\left(\mathtt{H}\,[\Delta\boldsymbol{\omega}]_\times^2\right) - \tfrac{1}{2}\operatorname{tr}(\mathtt{H})\operatorname{tr}\left([\Delta\boldsymbol{\omega}]_\times^2\right)\\ &= \operatorname{tr}\left(\mathtt{H}\Delta\boldsymbol{\omega}\Delta\boldsymbol{\omega}^\top\right)\\ &= \Delta\boldsymbol{\omega}^\top\mathtt{H}\Delta\boldsymbol{\omega},\end{aligned} \tag{3.29}$$

where we used that $[\mathbf{v}]_\times^2 = \mathbf{v}\mathbf{v}^\top - \mathbf{v}^\top\mathbf{v}\mathtt{I}_3$ and $\operatorname{tr}\left([\mathbf{v}]_\times^2\right) = -2\mathbf{v}^\top\mathbf{v}$. As a result, if $\mathtt{H}$ is a Hessian of the two-view optimization (hence can quantify uncertainty in $\Delta\boldsymbol{\omega}$ as a precision matrix), we can "transfer" the information about uncertainty from $\mathtt{H}$ to $\mathtt{M}$. In particular, generating rotation matrices $\tilde{\mathtt{R}}$ according to a Langevin distribution $p(\tilde{\mathtt{R}}; \mathtt{M}, \mathtt{Q}) \propto e^{\langle\mathtt{M}\tilde{\mathtt{R}}, \mathtt{Q}\rangle}$ is approximately (up to second order terms) equivalent to generating random axis-angle vectors $\Delta\boldsymbol{\omega}$ according to a Gaussian distribution $p(\Delta\boldsymbol{\omega}; \mathbf{0}, \mathtt{H}^{-1}) \propto e^{-\frac{1}{2}\Delta\boldsymbol{\omega}^\top\mathtt{H}\Delta\boldsymbol{\omega}}$. With that, the approximation in (3.27) allows to incorporate two-view uncertainties into rotation averaging while retaining the problem structure as following

$$\min_{\{\mathtt{R}_i\in\mathrm{SO}(3)\}} \sum_{(i,j)\in\mathcal{E}} -\left\langle\mathtt{M}_{ij}\tilde{\mathtt{R}}_{ij}, \mathtt{R}_j\mathtt{R}_i^\top\right\rangle, \tag{3.30}$$

or, equivalently,

$$\min_{\mathsf{R}\in\mathrm{SO}(3)^n} -\left\langle\mathsf{N}, \mathsf{R}\mathsf{R}^\top\right\rangle \tag{3.31}$$

in matrix form, where N stacks the relative rotations pre-multiplied with matrices encoding uncertainties (or zeros where they are not available)

$$\mathsf{N} = \begin{bmatrix} \mathtt{0} & (\mathtt{M}_{12}\tilde{\mathtt{R}}_{12})^\top & \cdots & (\mathtt{M}_{1n}\tilde{\mathtt{R}}_{1n})^\top \\ \mathtt{M}_{12}\tilde{\mathtt{R}}_{12} & \mathtt{0} & \cdots & (\mathtt{M}_{2n}\tilde{\mathtt{R}}_{2n})^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{M}_{1n}\tilde{\mathtt{R}}_{1n} & \mathtt{M}_{2n}\tilde{\mathtt{R}}_{2n} & \cdots & \mathtt{0} \end{bmatrix}. \tag{3.32}$$

In **Paper B**, we show that the cost matrices $\mathtt{M}_{ij}$ are often indefinite. Because of that the standard relaxation ignoring the determinant constraints [165] is not able to recover a global solution—we show *e.g.* that for each residual term with indefinite $\mathtt{M}_{ij}$ there exists an element of O(3) but not SO(3) that leads to a lower objective value. To see that, let the eigen-decomposition of the (symmetric) matrix $\mathtt{M}$ be $\mathtt{M} = \mathtt{U}\mathtt{D}\mathtt{U}^\top$, $\mathtt{D} = \mathrm{diag}(\{d_i\})$, $\det(\mathtt{U}) = 1$, and $\tilde{\mathtt{R}} \in \mathrm{SO}(3)$. The minimizer of

$$\min_{\mathtt{Q}\in\mathcal{A}} -\langle \mathtt{M}\tilde{\mathtt{R}}, \mathtt{Q}\rangle = \min_{\mathtt{Q}\in\mathcal{A}} -\langle \mathtt{U}\mathtt{D}\mathtt{U}^\top\tilde{\mathtt{R}}, \mathtt{Q}\rangle \tag{3.33}$$

over $\mathcal{A} = \mathrm{SO}(3)$ is $\mathtt{U}\mathtt{U}^\top\tilde{\mathtt{R}} = \tilde{\mathtt{R}}$, giving $-\sum_i d_i$ as the minimum value. The minimizer over $\mathcal{A} = \mathrm{O}(3)$ is however $\mathtt{U}\,\mathrm{diag}\left(\{\mathrm{sign}(d_i)\}\right)\mathtt{U}^\top\tilde{\mathtt{R}}$, giving the minimum value of $-\langle \mathtt{D}, \mathrm{diag}\left(\{\mathrm{sign}(d_i)\}\right)\rangle = -\sum_i d_i \mathrm{sign}(d_i)$. So the minimizers coincide if all $d_i \geq 0$, but if at least one $d_i < 0$, the minimizer over O(3) will no longer belong to SO(3). Therefore the determinant constraints must be kept. We derive a new SDP relaxation for the problem (3.31) as detailed next.

New convex relaxation The idea is to introduce auxiliary variables $\mathtt{Q}_{ij} = \mathtt{R}_j\mathtt{R}_i^\top$ that are constrained to be in SO(3)

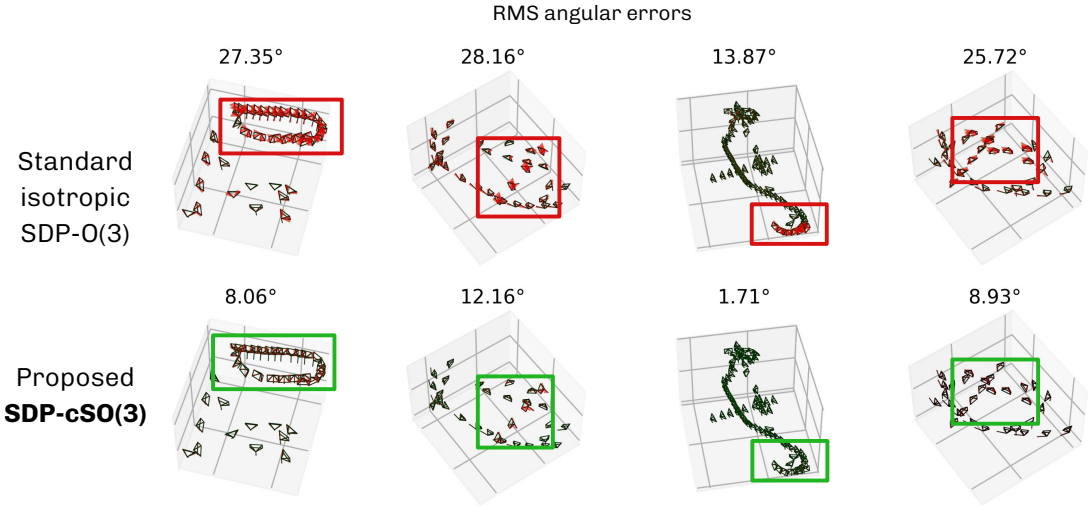
$$\min_{\mathsf{R},\mathsf{Q}\in\mathrm{SO}(3)^n} \quad -\langle \mathsf{N}, \mathsf{R}\mathsf{R}^\top\rangle \tag{3.34}$$

$$\text{s.t. } \mathtt{R}_i^\top\mathtt{R}_i = \mathtt{I}_3,\ \mathtt{R}_j\mathtt{R}_i^\top = \mathtt{Q}_{ij} \tag{3.35}$$

and to retain these SO(3) constraints until obtaining the dual problem. The dual variables for (3.35) can all be stacked into a $3n \times 3n$ symmetric matrix Υ. The Lagrangian can be simplified into

$$L(\mathsf{R}, \mathsf{Q}; \Upsilon) = \langle \Upsilon - \mathsf{N}, \mathsf{R}\mathsf{R}^\top\rangle - \mathrm{tr}(\Upsilon) - \sum_{i\neq j}\langle \Upsilon_{ij}, \mathtt{Q}_{ij}\rangle \tag{3.36}$$

Figure 3.8: The proposed SDP formulation for anisotropic rotation averaging often leads to more accurate reconstructions. These are examples of results on challenging data from ETH3D where we used SIFT keypoints with minimal filtering (ground truth camera locations are used for visualization purposes).



and its minimum *w.r.t.* R and $\mathsf{Q} \in \mathrm{SO}(3)^n$ is

$$-\operatorname{tr}(\Upsilon) + \min_{\mathsf{R}} \langle \Upsilon - \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle - \sum_{i \neq j} \min_{\mathtt{Q}_{ij} \in \mathrm{SO}(3)} \langle \Upsilon_{ij}, \mathtt{Q}_{ij} \rangle. \tag{3.37}$$

Similarly to the derivations in Section 2.5, $\langle \Upsilon - \mathsf{N}, \mathsf{R}\mathsf{R}^\top \rangle$ has minimum value 0 if $\Upsilon - \mathsf{N} \succeq 0$, otherwise is unbounded from below. For the last term we get

$$\min_{\mathtt{Q}_{ij} \in \mathrm{SO}(3)} -\langle \Upsilon_{ij}, \mathtt{Q}_{ij} \rangle = -\mathcal{I}^*_{\mathrm{SO}(3)}(\Upsilon_{ij}), \tag{3.38}$$

where $\mathcal{I}^*_{\mathrm{SO}(3)}$ is the convex conjugate of the indicator function $\mathcal{I}_{\mathrm{SO}(3)}$, *i.e.* $\mathcal{I}_{\mathrm{SO}(3)}(\mathtt{X}) = 0$ if $\mathtt{X} \in \mathrm{SO}(3)$ and ∞ otherwise. The dual problem is therefore

$$\max_{\Upsilon : \Upsilon - \mathsf{N} \succeq 0} -\operatorname{tr}(\Upsilon) - \sum_{i \neq j} \mathcal{I}^*_{\mathrm{SO}(3)}(\Upsilon_{ij}). \tag{3.39}$$

Note the difference to (2.40) which is in the additional term. When taking the dual once more (similarly to derivations in Section 2.5), this results in the additional terms of the form $\mathcal{I}^{**}_{\mathrm{SO}(3)}(\mathtt{X}_{ij})$ which are equal to $\mathcal{I}_{\mathrm{conv}(\mathrm{SO}(3))}(\mathtt{X}_{ij})$ [262] and can be written as constraints, giving the following bidual problem

$$\min_{\mathtt{X}\succeq 0} \; -\langle \mathsf{N}, \mathsf{X} \rangle \tag{3.40}$$

$$\text{s.t. } \mathtt{X}_{ii} = \mathtt{I}_3, \;\; \mathtt{X}_{ij} \in \mathrm{conv}(\mathrm{SO}(3)), \tag{3.41}$$

where we note that $\cdot \in \mathrm{conv}(\mathrm{SO}(3))$ is a semidefinite constraint [263], [264]. The resulting SDP solver is able to recover global minima. It often gives more accurate solutions than the isotropic counterparts (some qualitative results are shown in Figure 3.8).

Faster solver? The new formulation introduces a large number of additional constraints (on $\{\mathtt{X}_{ij}\}$) that grows quadratically with the number of cameras. The general purpose SDP solvers that could be used for the problem in (3.40) scale poorly with the problem size. While addressing this general issue is an active area of research [167]–[173], it is also of interest to understand if a more efficient dedicated solver leveraging the specific problem structure could be designed. This is the goal of **Paper C**. Inspired by a family of block coordinate descent methods proposed to optimize the standard chordal distances, we formulate an efficient dedicated solver that directly incorporates rotational constraints into the subproblems of the block-coordinate descent. We show that the resulting solver, called anisotropic coordinate descent (ACD), easily admits two-view uncertainties. There are two key results from the paper related to the solver development. The first result is the proposed sub-problem for the block-coordinate descent algorithm that incorporates uncertainties and SO(3) constraints. It has the following form

$$\min_{\mathsf{S}\in \mathrm{SO}(3)^{n-1}} \; -\langle \mathsf{N}_{\tilde{k},k}, \mathsf{S} \rangle \quad \text{s.t. } \begin{pmatrix} \mathtt{I}_3 & \mathsf{S}^\top \\ \mathsf{S} & \mathsf{B} \end{pmatrix} \succeq 0, \tag{3.42}$$

where the k^{th} camera is chosen as an "anchor"—all 3×3 blocks in R except for the k^{th} one are fixed giving $\mathsf{R}_{\tilde{k}} = (\mathtt{R}_1^\top \cdots \mathtt{R}_{k-1}^\top \; \mathtt{R}_{k+1}^\top \cdots \mathtt{R}_n^\top)^\top \in \mathbb{R}^{3(n-1)\times 3}$, $\mathsf{B} = \mathsf{R}_{\tilde{k}} {\mathsf{R}_{\tilde{k}}}^\top$ and $\mathsf{N}_{\tilde{k},k} \in \mathbb{R}^{3(n-1)\times 3}$ which is the k^{th} block-column of N excluding the k^{th} block-row. The reason for why we add the SO(3) constraints is that we

found that the existing isotropic method RCD [257] retains SO(3) membership throughout the iterations without enforcing it. The second result is obtaining a closed-form solution of the proposed sub-problem in (3.42)—$\mathsf{S}^* = \mathsf{R}_{\tilde{k}}\ \mathtt{R}_k^{*\top}$, where

$$\mathtt{R}_k^* = \text{project}_{\text{SO(3)}}(\mathsf{N}_k^\top \mathsf{R}) \tag{3.43}$$

$$\text{project}_{\text{SO(3)}}(\mathtt{M}) = \mathtt{U}\,\text{diag}(1, 1, \det(\mathtt{U}\mathtt{V}^\top))\mathtt{V}^\top \qquad [\mathtt{U}, \sim, \mathtt{V}] = \text{svd}(\mathtt{M}), \tag{3.44}$$

where $\mathsf{N}_k \in \mathbb{R}^{3n\times 3}$ is the k^{th} block-column of N. We also show that there is no need to update the whole matrix of unknowns in R (by setting $\mathtt{R}_k = \mathtt{I}_3$ and $\mathsf{R}_{\tilde{k}}$ to S^*). It is equivalent to updating a single unknown rotation $\mathtt{R}_k$ (that was chosen as an anchor point for the sub-problem) to $\mathtt{R}_k^*$. The proposed algorithm is very simple: (i) select a camera k and (ii) update the corresponding rotation matrix using (3.43), and repeat until convergence, as shown in Algorithm 1. While the method does not guarantee a global minimum, we find that it works well independently of the initialization strategy, and it is stable under increasing perturbations of the estimated Hessians. An example of the performance of ACD compared to the isotropic rotation averaging solver as well as the SDP solver from **Paper B** is shown in Figure 3.9.

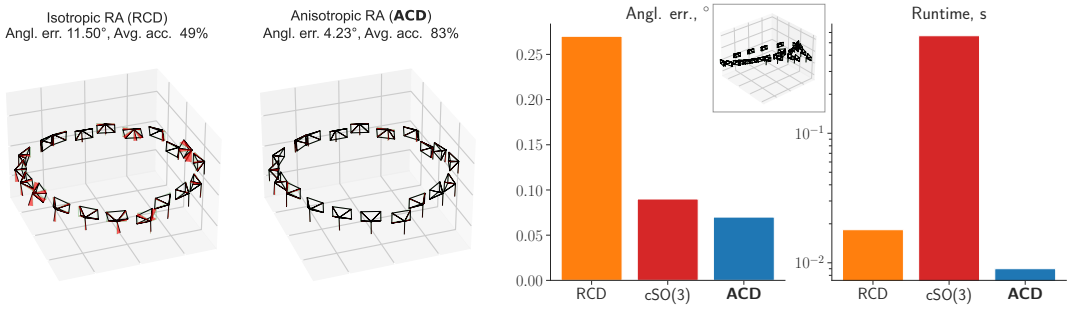
Algorithm 1: ACD algorithm.
Inputs: $\mathsf{N}, \left(\mathsf{R}^{(0)}\right)$.

for $t \in \{1, \dots\}$ **do**
  for $k \in$ *shuffle*$(\{1, \dots, n\})$ **do**
    $\mathtt{R}_i^{(t)} \leftarrow \mathtt{R}_i^{(t-1)} \quad (i \neq k)$;
    $\mathtt{R}_k^{(t)} \leftarrow \text{project}_{\text{SO(3)}}(\mathsf{N}_k^\top \mathsf{R}^{(t-1)})$;
  end
end
return R

3.6 Robust rotation averaging

Another aspect that we are interested in is robust estimation of rotations. One could always try to filter out outliers prior to the averaging step—there exist several methods doing that. For example, Govindu *et al.* [266] samples multiple minimum spanning trees from the view-graph using RANSAC [138]. Zach *et al.* [267] proposes a Bayesian inference approach to identify the likely

Figure 3.9: Performance of ACD as compared to RCD and the SDP solver: (left) on synthetic loop scene and (right) on the "boulders" dataset from ETH3D [265].



false positive pairwise geometric relations from the sampled cycles of the camera graph. Lee *et al.* [206] constructs a spanning tree in a hierarchical manner based on the consistency and reliability of the triplet constraints. Recent methods [268], [269] propose to learn to detect "doppelgangers"—images of distinct but visually similar structures—among pairs of images in the dataset. But what if some outliers were left undetected? We could figure that out either by manual inspection or after running the optimization and analyzing the reconstruction. A more principled approach is to perform robust rotation averaging via simultaneous outlier suppression and model optimization. This can be done by robustifying the residuals. The research sub-field has shown a lot of progress in robust *isotropic* averaging of a single or multiple rotations. In certain formulations under certain conditions, *e.g.* for the geodesic L1-means in SO(3) [270] or minimizers of truncated least squares [271], [272], it was shown that it is possible to obtain global minima for a single rotation averaging. For multiple rotations, some results where obtained in limited scenarios, *e.g.* [273] guarantees exact and stable recovery under a specific noise model, [272] gives theoretical guarantees although can be computationally infeasible. Then, many practical approaches are proposed. Hartley *et al.* [270] iteratively update absolute rotations by using the Weiszfeld algorithm for computing the geodesic L1-mean. Arrigoni *et al.* [274], [275] formulate the problem of decomposing the data matrix $\tilde{\mathtt{R}}$ into a rank-3 matrix $\mathtt{X}(= \mathtt{R}\mathtt{R}^\top)$ and a sparse matrix encoding outliers, and solve it via leveraging the GoDec [276] algorithm for low-rank matrix recovery in the presence of noise and outliers.

L1IRLS algorithm of Chatterjee and Govindu [131] is probably one of the

most well-known approaches to robust rotation averaging. It builds on top of the non-robust method of Govindu [253] that locally optimizes the angular distances

$$\min_{\{\mathtt{R}_i\}} \sum\nolimits_{(i,j)\in\mathcal{E}} \|\omega(\mathtt{R}_j^\top \tilde{\mathtt{R}}_{ij}\mathtt{R}_i)\|^2 \tag{3.45}$$

using the quasi-Newton solver [277]. In particular, if we let $\{\Delta\boldsymbol{\omega}_i\}$ be the incremental axis-angle updates of rotations $\{\mathtt{R}_i\}$, the approximation of the form $\omega\left(e^{[-\Delta\boldsymbol{\omega}_j]_\times}\mathtt{R}_j^\top\tilde{\mathtt{R}}_{ij}\mathtt{R}_i e^{[\Delta\boldsymbol{\omega}_i]_\times}\right) \approx \omega\left(\mathtt{R}_j{}^\top\tilde{\mathtt{R}}_{ij}\mathtt{R}_i\right) + \Delta\boldsymbol{\omega}_i - \Delta\boldsymbol{\omega}_j$ is used. Thus the iterative scheme of [277] is as following

$$\{\Delta\boldsymbol{\omega}_i^{(t)}\} \leftarrow \operatorname*{argmin}_{\{\Delta\boldsymbol{\omega}_i\}} \sum_{(i,j)\in\mathcal{E}} \left\| \omega\left(\mathtt{R}_j^{(t)\top}\tilde{\mathtt{R}}_{ij}\mathtt{R}_i^{(t)}\right) - (\Delta\boldsymbol{\omega}_j - \Delta\boldsymbol{\omega}_i)\right\|^2 \tag{3.46}$$

$$\mathtt{R}_i^{(t+1)} \leftarrow \mathtt{R}_i^{(t)} e^{\left[\Delta\boldsymbol{\omega}_i^{(t)}\right]_\times}. \tag{3.47}$$

The follow-up work [131] robustifies the objective in (3.45) by adding the IRLS weights of Geman-McClure [278] kernel. To ensure a sufficiently good initialization, they propose to first optimize an l_1 loss (in each sub-problem)

$$\{\Delta\boldsymbol{\omega}_i^{(t)}\} \leftarrow \operatorname*{argmin}_{\{\Delta\boldsymbol{\omega}_i\}} \sum_{(i,j)\in\mathcal{E}} \left\| \omega\left(\mathtt{R}_j^{(t)\top}\tilde{\mathtt{R}}_{ij}\mathtt{R}_i^{(t)}\right) - (\Delta\boldsymbol{\omega}_j - \Delta\boldsymbol{\omega}_i)\right\|_1, \tag{3.48}$$

which can be done efficiently via convex programming [279].

What about robustifying *anisotropic* residuals? In the anisotropic rotation averaging formulation presented earlier, the assumption is that the estimated Hessians approximate the true distribution sufficiently well. This can be true for the relative rotations that are inliers, but not for outliers. The outlying relative poses with low point correspondence support would likely have high uncertainties and hence would be easy to detect and "ignore" (by downweighting). Consequently, in practice, anisotropic rotation averaging will be more robust *w.r.t.* uncertain outliers than the isotropic counterpart (although it will not completely ignore any of the measurements). However, there might also exist the outlying estimates of relative motions that are more difficult to identify due to, for example, many confident but incorrect point correspondences resulting in "confident" Hessians. This can happen when the scene contains symmetrical objects or other repetitive structures [267], [268]. We are therefore interested in robust anisotropic rotation averaging. This problem has not

been studied much, in fact, the only work that looked at this problem is by Zhang *et al.* [60] that integrates a robust kernel $\rho(\cdot)$ into anisotropic angular distance optimization as following

$$\min_{\{\mathtt{R}_i\}} \sum\nolimits_{(i,j)\in\mathcal{E}} \rho\left(\|\omega(\mathtt{R}_j\mathtt{R}_i^\top\tilde{\mathtt{R}}_{ij}^\top)\|_{\mathtt{H}_{ij}}\right). \tag{3.49}$$

For robust kernel $\rho(\cdot)$, [60] uses MAGSAC [280] (which reduces the dependency on the manually selected robust threshold parameter), and the objective is locally optimized using IRLS. The initialization is obtained via minimum spanning tree. Using this initialization technique risks bringing back the problem of accumulating errors which global SfM is supposed to address (in contrast to incremental SfM). We study initialization and robust refinement further in **Paper C** and later in this section.

Robust anisotropic rotation averaging

In **Paper C**, we propose an anisotropic extension to the standard robust refinement strategy from Chatterjee and Govindu [131]. We naturally incorporate uncertainties into sub-problems (3.46) as following

$$\min_{\{\Delta\boldsymbol{\omega}_i\}} \sum_{(i,j)\in\mathcal{E}} \rho\left(\left\|\omega\left(\mathtt{R}_j^{(t)\top}\tilde{\mathtt{R}}_{ij}\mathtt{R}_i^{(t)}\right) - (\Delta\boldsymbol{\omega}_j - \Delta\boldsymbol{\omega}_i)\right\|_{\mathtt{H}_{ij}}\right). \tag{3.50}$$

We use Geman-McClure [278] kernel and also perform local optimization with IRLS, but we use ACD for initialization. In **Paper C** we show that it gives a better initial estimate of absolute rotations.

Note that robust anisotropic objective formulation in (3.50) uses axis-angle vectors parameterizing rotations. In contrast, anisotropic objective formulation in ACD (3.30) uses rotation matrices directly. We therefore propose a different approach that enables a smooth transition from the non-robust to the robust objective. Specifically, we robustify the "anisotropic chordal distances" directly which leads to a natural extension of the ACD algorithm. Consequently, this approach also alleviates the need for a separate initialization.

To incorporate robust kernel, we first recall from (3.27) that the individual anisotropic chordal distance terms take the following form

$$r_{ij}(\mathtt{R}_i,\mathtt{R}_j) = \mathrm{tr}(\mathtt{M}_{ij}) - \left\langle \mathtt{M}_{ij}\tilde{\mathtt{R}}_{ij}, \mathtt{R}_j\mathtt{R}_i^\top \right\rangle. \tag{3.51}$$

The robustified objective is therefore simply

$$\min_{\{\mathtt{R}_i\}} \sum_{(i,j)\in\mathcal{E}} \rho\left(r_{ij}(\mathtt{R}_i, \mathtt{R}_j)\right). \tag{3.52}$$

Algorithm 2: Robust ACD-GeMM algorithm.
Inputs: $\{\tilde{\mathtt{R}}_{ij}\}, \{\mathtt{M}_{ij}\}, \tau, \left(\mathsf{R}^{(0)}, \mathsf{W}^{(0)}\right)$.

for $t \in \{1, \dots\}$ **do**
  for $k \in$ *shuffle*$(\{1, \dots, n\})$ **do**
    $\mathtt{R}_i^{(t)} \leftarrow \mathtt{R}_i^{(t-1)} \quad (i \neq k)$;
    $\mathtt{R}_k^{(t)} \leftarrow \mathrm{project}_{\mathrm{SO}(3)}\left(\left(\mathsf{N}_k \odot \left(\mathsf{W}_k^{(t-1)} \otimes \mathtt{1}_{3\times 3}\right)\right)^{\top} \mathsf{R}^{(t-1)}\right)$;
    $w_{ik}^{(t)} \leftarrow w_{\mathrm{GeMM}}\left(r_{ik}\left(\mathtt{R}_i^{(t)}, \mathtt{R}_k^{(t)}\right),\ w_{ik}^{(t-1)};\ \tau\right) \quad i:\ (i,k) \in \mathcal{E}$;
  end
end
return R

We propose to optimize this objective using generalized majorization-minimization (GeMM) [61] approach—as discussed earlier, GeMM is an alternative to IRLS that enjoys an improved ability of reaching better local minima (see Section 2.5). We lift the robust kernel by introducing auxilliary variables $\{w_{ij}\}$ called weights as following

$$\min_{\{\mathtt{R}_i\},\{w_{ij}\}} \sum_{(i,j)\in\mathcal{E}} \overline{\rho}\left(r_{ij}(\mathtt{R}_i, \mathtt{R}_j), w_{ij}\right), \tag{3.53}$$

where $\overline{\rho}(x, w) = \frac{1}{2}wx^2 + \kappa(w)$ (see Section 2.5). Denote $x_{ij}^{(t)} = r_{ij}\left(\mathtt{R}_i^{(t)}, \mathtt{R}_j^{(t)}\right)$ to be the residual terms (that are iteratively updated). We initialize the weights as $w_{ij}^{(0)} = 1$ and iteratively update them such that the GeMM condition is satisfied as following

$$\sum_{(i,j)\in\mathcal{E}} \overline{\rho}\left(x_{ij}^{(t)}, w_{ij}^{(t)}\right) \leq \eta \sum_{(i,j)\in\mathcal{E}} \rho\left(x_{ij}^{(t)}\right) + (1-\eta) \sum_{(i,j)\in\mathcal{E}} \overline{\rho}\left(x_{ij}^{(t)}, w_{ij}^{(t-1)}\right), \tag{3.54}$$

There are many solutions that satisfy this condition. We propose to select one that can be obtained efficiently—computing the adjusted individual weights

independently from each other, *i.e.*

$$w_{ij}^{(t)} = v \quad \text{s.t.} \quad \overline{\rho}\left(x_{ij}^{(t)}, v\right) = \eta\rho\left(x_{ij}^{(t)}\right) + (1-\eta)\overline{\rho}\left(x_{ij}^{(t)}, w_{ij}^{(t-1)}\right) \tag{3.55}$$

Let $C(x,w) = \eta\rho(x) + (1-\eta)\overline{\rho}(x,w)$. Then we can define a function $w_{\text{GeMM}}(x,w;\tau)$ that depends on the residual x and the previous state of the weight w, given the robust threshold τ, as following

$$v = w_{\text{GeMM}}(x,w;\tau) \quad \text{s.t.} \quad \tfrac{1}{2}vx^2 + \kappa(v) = C(x,w) \tag{3.56}$$

$$v \in \left[\min\{w, \overline{w}(x;\tau)\}, \max\{w, \overline{w}(x;\tau)\}\right], \tag{3.57}$$

where $\overline{w}(x;\tau)$ is the IRLS weight. At each iteration, the new weights are therefore computed as $w_{ij}^{(t)} = w_{\text{GeMM}}(x_{ij}^{(t)}, w_{ij}^{(t-1)};\tau)$. The new robust algorithm called ACD-GeMM is described in Algorithm (2), where matrix W contains all the weights w_{ij} where they are available and zeros otherwise, and recall that matrix N contains relative rotations $\{\tilde{\mathtt{R}}_{ij}\}$ pre-multiplied with uncertainties $\{\mathtt{M}_{ij}\}$ as in (3.32).

For many robust kernels we obtain a closed-form solution of $w_{\text{GeMM}}(x,w;\tau)$. We provide some examples next. The Geman-McClure [278] kernel is $\rho(x) = \frac{x^2\tau^2}{2(x^2+\tau^2)}$, its corresponding bias function is $\kappa(w) = \frac{\tau^2}{2}\left(\sqrt{w}-1\right)^2$, and the IRLS weight is $\overline{w}(x) = \frac{\tau^4}{(x^2+\tau^2)^2}$. The GeMM weight $v = w_{\text{GeMM}}(x,w;\tau)$ satisfies

$$\frac{1}{2}vx^2 + \frac{\tau^2}{2}\left(\sqrt{v}-1\right)^2 = C(x,w), \tag{3.58}$$

which together with the condition in (3.57) gives

$$w_{\text{GeMM}}(x,w;\tau) = \frac{\left(\tau^2 + \text{sign}\left(w - \overline{w}(x)\right)\sqrt{2C(x,w)\left(x^2+\tau^2\right) - \tau^2x^2}\right)^2}{\left(x^2+\tau^2\right)^2}. \tag{3.59}$$

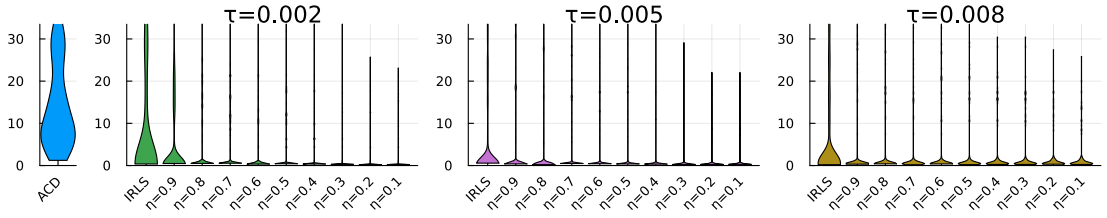
The l_1 kernel generates the following condition

$$\frac{1}{2}vx^2 + 1/v = C(x,w), \tag{3.60}$$

which gives

$$w_{\text{GeMM}}(x,w;\tau) = \frac{C(x,w)}{x^2} \pm \frac{1}{x}\sqrt{\frac{C^2(x,w)}{x^2} - 2}. \tag{3.61}$$

Figure 3.10: Comparison between ACD, ACD-IRLS and ACD-GeMM. The RMS angular error of the absolute rotations found by optimizing (3.52) on 100 synthetic scenes for different parameters τ of the robust kernel, with ACD on the left serving as a baseline. IRLS (leftmost violins) is outperformed by GeMM (remaining violins) with varying adjustment scales denoted as η (setting η to 1 coincides with IRLS).



The Huber [281] kernel generates the following condition

$$\frac{1}{2}vx^2 + \frac{1}{2}\tau^2\left(\frac{1}{v} - 1\right) = C(x, w), \tag{3.62}$$

which gives a similar formulation

$$w_{\text{GeMM}}(x, w; \tau) = \frac{\tau^2 + 2C(x, w)}{2x^2} \pm \frac{1}{x}\sqrt{\frac{(\tau^2 + 2C(x, w))^2}{4x^2} - \tau^2} \tag{3.63}$$

The Tukey's bi-weight kernel leads to

$$\frac{1}{2}vx^2 + \frac{\tau^2}{6}(\sqrt{v} - 1)^2(2\sqrt{v} + 1) = C(x, w), \tag{3.64}$$

which can be re-written as

$$2\left(\sqrt{v}\right)^3 + 3\left(\frac{x^2}{\tau^2} - 1\right)v + 1 = 6\frac{C(x, w)}{\tau^2}, \tag{3.65}$$

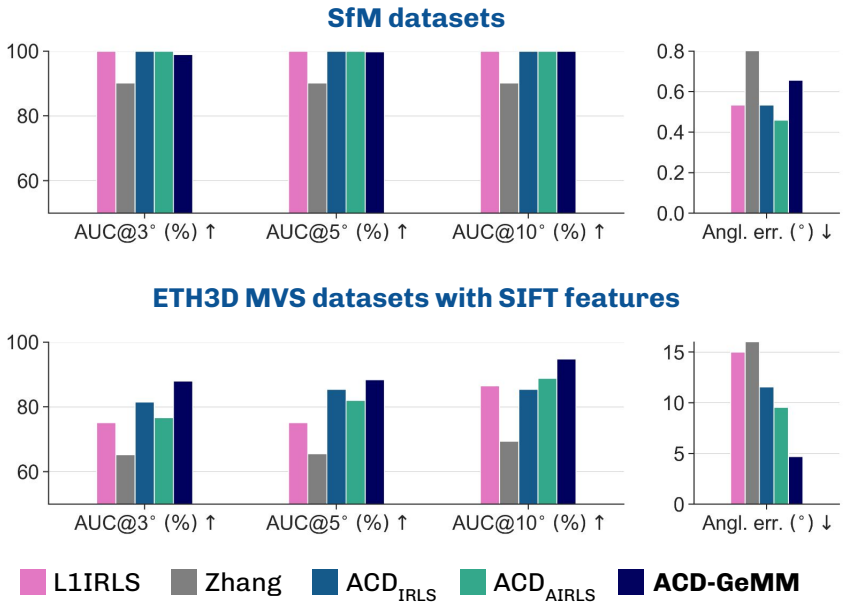
therefore a closed-form solution can be obtained. The truncated quadratic kernel generates

$$\frac{1}{2}vx^2 + \frac{1}{2}\tau^2(1 - v) = C(x, w), \tag{3.66}$$

giving

$$w_{\text{GeMM}}(x, w; \tau) = \frac{2C(x, w) - \tau^2}{x^2 - \tau^2}. \tag{3.67}$$

Figure 3.11: Rotation averaging performance of ACD-GeMM compared to the other robust methods



For some other kernels, the condition does not generally lead to a closed-form solution, such as for Cauchy kernel ($\frac{1}{2}vx^2 + \frac{\tau^2}{2}(v - \log(v) - 1) = C(x, w)$), or smooth truncated quadratic ($\frac{1}{2}vx^2 + \tau^2/p\,(v-1)^p = C(x, w)$).

For the kernels that give closed-form weight solutions, the computational costs of a single iteration are almost unaffected as compared to IRLS, while the improvements in accuracy can be substantial. We choose the Geman-McClure kernel for our experiments and test the proposed approach on the challenging synthetic datasets with varying proportions of outliers. As shown in Figure 3.10, we observe a consistent improvement—robust rotation averaging with higher "exploration" capabilities finds a better minimum more often. It also produces much more accurate reconstructions on challenging real datasets as summarized in Figure 3.11.

3.7 Motion segmentation

Previously we discussed the reconstruction pipeline for static scenes. Let us now imagine a scenario where the scene also contains dynamic objects, such as

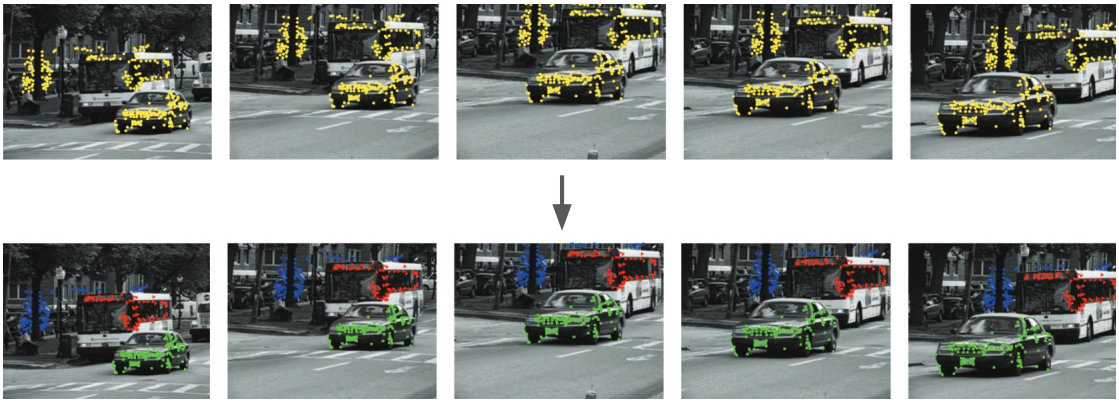
vehicles, people, animals moving in space, which is a very common situation, and we record a video of this scene. The reconstruction pipeline described above will not "work" for the images of such dynamic scenes—it can at most filter out the dynamic part or reconstruct one rigid body only. Knowing which point tracks (or trajectories) correspond to which motions would be very helpful. This is addressed by *motion segmentation*, a fundamental task in dynamic scene understanding and reconstruction.

Motion segmentation deals with two problems simultaneously—given a set of point trajectories observed in the image sequence containing multiple moving (and potentially deforming) objects, (1) estimate the individual motion models and (2) cluster the point trajectories based on these motions. Depending on the context, motions can be either completely independent (or uncorrelated) or partially dependent (or correlated) but modeled separately using simpler models (because it is easier to do so). Estimation of both cluster assignments and motion models is a difficult chicken-and-egg problem. Namely, estimating a single motion model from a set of point trajectories belonging to the same motion is relatively easy (see *e.g.* Section 3.3). Similarly, if we happen to know the motion models, clustering point trajectories becomes the assignment problem where for each trajectory we can choose the model that minimizes the track-to-model error. However, the combination of cluster assignment and model estimation becomes much more complicated, and further complications arise due to occlusions and outliers.

The first step towards addressing dynamic scene reconstruction is moving away from a single rigid motion assumption to *multiple rigid motions* assumption. This is the problem we study in this thesis and is illustrated in Figure 3.12. Traditional methods dealing with rigid motion segmentation can be roughly split into (1) subspace clustering and (2) multi-model fitting.

Subspace clustering *Subspace clustering* is a family of approaches that work with high-dimensional data that lie in a union of low-dimensional subspaces. The goal is to estimate these subspaces and cluster the data based on these models. The first connection to motion segmentation can be established by noting that sequences of tracked 2D point coordinates $(\underline{\mathbf{u}}_{1,j}, \cdots, \underline{\mathbf{u}}_{n,j})$ form

Figure 3.12: Illustration of rigid motion segmentation problem. Given a set of (sparse) point trajectories, the goal is to cluster them based on the underlying rigid motions. Here, images are shown for visualization purposes only.



trajectories that are also points in a high-dimensional space, *i.e.*

$$\begin{bmatrix} \underline{\mathbf{u}}_{1,j} \\ \vdots \\ \underline{\mathbf{u}}_{n,j} \end{bmatrix} \in \mathbb{R}^{2n}. \tag{3.68}$$

The second important connection can be drawn by looking at the seminal works in matrix factorization [65], [282] that show that SfM under affine camera model assumption is essentially a subspace fitting problem (see also Section 3.3). Generalization to multiple rigid motions leads to subspace clustering [283]–[285].

To formulate motion segmentation as subspace clustering, recall first how the $2n \times m$ observation matrix $\mathtt{M}$ is constructed for affine factorization

$$\mathtt{M} = \begin{bmatrix} \underline{\mathbf{u}}_{1,1} & \underline{\mathbf{u}}_{1,2} & \cdots & \underline{\mathbf{u}}_{1,m} \\ \underline{\mathbf{u}}_{2,1} & \underline{\mathbf{u}}_{2,2} & \cdots & \underline{\mathbf{u}}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{\mathbf{u}}_{n,1} & \underline{\mathbf{u}}_{n,2} & \cdots & \underline{\mathbf{u}}_{n,m} \end{bmatrix}. \tag{3.69}$$

Now, with multiple rigid motions, this observation matrix can no longer be factorized as $\mathtt{M} \approx \mathtt{PX}$. Instead, we need to (1) find an $m \times m$ permutation

matrix $\mathtt{P}_\pi$ that orders the trajectories into sub-matrices $\{\mathtt{M}_i\}_{i=1}^c$ corresponding to different motions, and (2) compute these motions $\{\mathtt{B}_i, \mathtt{C}_i\}$. Note that here, we purposefully use a different notation, *i.e.* $\mathtt{B}_i$ and $\mathtt{C}_i$ instead of camera matrices $\mathtt{P}_i$ and 3D points $\mathtt{X}_i$. Each motion is represented by a $2n \times k_i$ basis matrix $\mathtt{B}_i$ whose columns span the subspace and therefore $\mathtt{M}_i = \mathtt{B}_i\mathtt{C}_i^\top$, where $\mathtt{C}_i$ is a $k_i \times m_i$ coefficient matrix. Hence the problem is formulated as

$$\mathtt{M}\mathtt{P}_\pi \approx \begin{bmatrix} \mathtt{B}_1\mathtt{C}_1^\top & \dots & \mathtt{B}_c\mathtt{C}_c^\top \end{bmatrix}. \tag{3.70}$$

Note that $\sum_{i=1}^c m_i = m$, and ranks k_i are assumed to be small. In this work we assume the known ranks $k_1 = \cdots = k_c = 4$ which is a common scenario under multiple rigid motions [286].

The seminal work of [285] derived a shape interaction matrix that, in the noiseless case and after applying permutation, is block-diagonal. However, its quality degrades in the presence of noise. A number of improvements were proposed [287]–[291], in particular, the use of the shape interaction matrix as an affinity matrix in a spectral clustering framework [292]. Many works since then focused on designing a better affinity matrix [293]–[297]. The generalized PCA approach (GPCA) [298]–[300] treats the union-of-subspaces as a zero-level-set of a higher degree polynomial which is fit to the data.

Another branch of methods uses *sparse* subspace clustering formulation [294], [301]. The methods are based on the notion of self-expressiveness, *i.e.* that all data in a d-dimensional subspace are linear combinations of d points from that subspace. This enables an efficient approach that is provably correct in the case of disjoint subspaces (we will follow-up on this assumption in the next section). The LRR approach [293], [302] uses a dictionary basis and enforces low rank patterns among the coefficients. Zhang *et al.* [303] proposes to learn a self-expressive data representation.

The aforementioned subspace clustering techniques are general and therefore do not exploit the temporal dependencies, which, as we advocate in **Paper D**, is an important domain knowledge in motion segmentation. On the contrary, there exist methods that leverage the temporal smoothness by constructing time-dependent bases. *E.g.* Akhter *et al.* [304] note the duality between trajectory space and shape space in matrix factorization for nonrigid SfM and proposed to use the discrete cosine transform (DCT) basis. We take inspiration from this decomposition in **Paper D**.

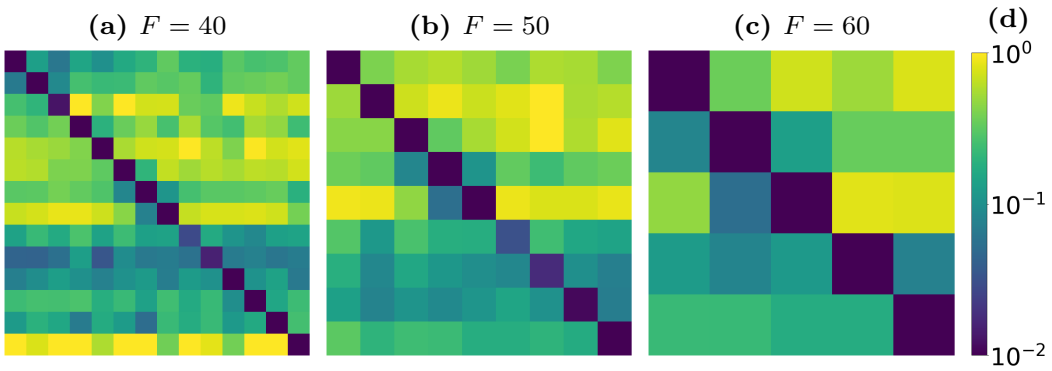
Multi-model fitting A parallel track of works treats motion segmentation as an instance of *multi-model fitting* [305]—a classical problem in computer vision that deals with estimating multiple (unknown number of) models from data. The framework is general, with no assumption on data modality. Besides motion segmentation, other examples include: fitting multiple homographies to points on imaged planes, fitting various geometric primitives to 2D and 3D data (lines, circles, in 3D also planes, spheres, *etc.*), multi-target tracking [306].

Some works propose to adapt robust statistical methods (*i.e.* of RANSAC family) to multi-model fitting [305], [307], [308]. Overall, such methods work well in low-dimensional cases, but otherwise tend to be sub-optimal due to their greedy approach [62]. Joint methods formulated via energy minimization [62], [63], [309]–[311] alleviate this problem. They are however usually computationally expensive. Borrowing a quote from Zhao *et al.* in ParticleSfM: "*Unfortunately, traditional cluster-based trajectory segmentation methods rely on heavy optimization and hand-crafted features, and are hard to scale with dense trajectories*" [312]. We aim to address that in **Paper D** which is discussed in the next section.

Learning trajectory embeddings

Deep neural networks have been shown to perform well on pattern recognition and compression tasks. Motivated by this (and noting that neural networks have not been tried before to solve motion segmentation), we look into learning a low-dimensional feature representation of point trajectories that can be directly used for clustering. This eliminates the need for simultaneous estimation of assignments and subspace models, which is the main complication in this task. Our key empirical observation is that re-using subspaces to explain trajectories in other clusters leads to higher errors (see Figure 3.13). This leads to a disjoint subspace assumption [294]—two subspaces only intersect at the origin, which is a typical assumption for high-dimensional data with low-dimensional subspaces. Based on this assumption, we propose to learn a function that maps a single trajectory to a subspace basis. Therefore, the architectural choices for the proposed feature extractor are: (1) using PointNet [313] style and no global context aggregation (*e.g.* spatial pooling) to leverage the disjoint subspace assumption, and (2) applying 1D convolutions in the temporal domain to leverage the knowledge about temporal dependen-

Figure 3.13: Re-using subspaces to explain trajectories in other clusters leads higher errors. The cell colors are normalized cluster-to-subspace errors in Hopkins155 dataset (subsequences of length $F = 40$, 50, and 60). An (i, j) cell corresponds to the trajectory cluster of the i^{th} motion and the subspace of j^{th} motion.



cies.

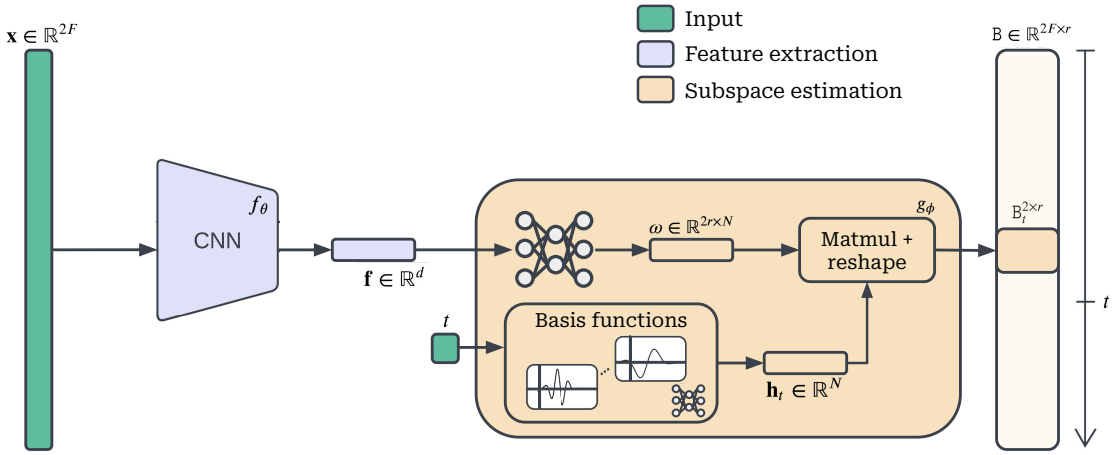
Another goal of the proposed approach is to learn to extract the geometric relations needed to infer the generating motions. For the decoder, we choose to estimate the motion models, *i.e.* subspaces that encode the change of motion over time. It is therefore reasonable to introduce a time-dependent basis. We use a damped version of the cosine basis

$$h_{\psi}^{j}(t) = e^{-(\alpha_j(t-\mu_j))^2} \cos(\beta_j t + \gamma_j), \tag{3.71}$$

where basis coefficients are α_j, μ_j, β_j and γ_j inferred from the features. Overall, we obtain a coordinate MLP-style [175] decoder network. This is a common strategy when learning to reconstruct time-dependent data. *E.g.* HyperTime [314] uses a learning approach with MLPs and sine activations for general time series representations; Ramasinghe and Lucey [315] show that there is a broad class of activations that are suitable for encoding signals with coordinate MLPs; Zheng *et al.* [316] show that depth of a coordinate based network can be traded for complexity of the positional embedding, and further show that this results in methods that are orders of magnitude faster than state-of-the-art. To conclude, we propose a specialized encoder-decoder network that is illustrated in Figure 3.14.

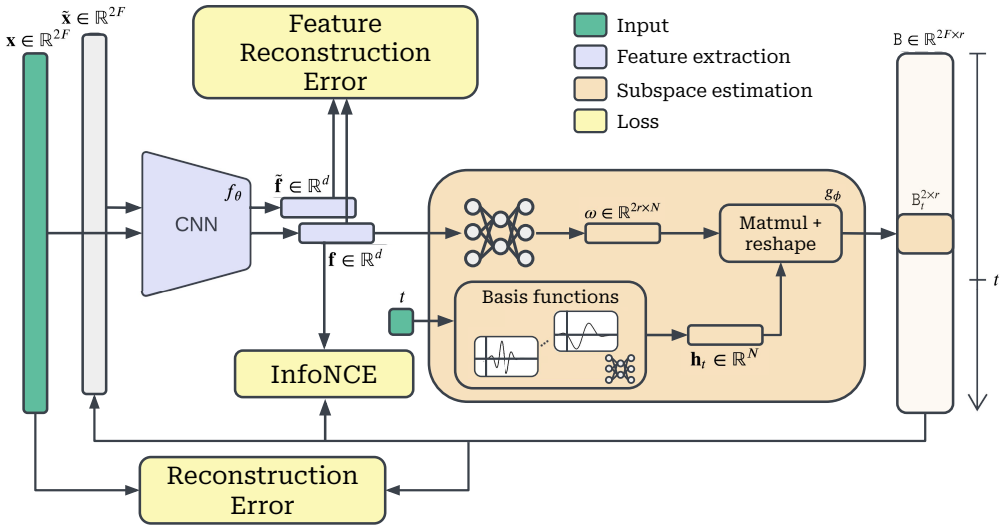
Metric learning, a subfield of (deep) machine learning, deals with learning

Figure 3.14: Neural network-based approach to processing trajectories.



similarity measures, often via projection onto a low-dimensional representation space, that are useful for clustering, classification, retrieval, compression, or other application where low-dimensional representations are of help. Often the similarity measure is (although it does not have to be) obtained as a Euclidean distance in the representation space. Metric learning can be done in a supervised, weakly/semi-supervised, or unsupervised manner. In a typical supervised learning scenario, the supervision is constructed for pairs, triplets, or larger sets of data points as "similar"/"dissimilar" or "belonging to the same group"/"belonging to the different groups", depending on the context. A well-known approach to learn a similarity measure is through *contrastive learning*—by optimizing the objective that simultaneously penalizes dissimilar representations that should be similar and vice versa. In conclusion, the representations are learned by maximizing agreement between related samples (also referred to as *positive pairs*) and minimizing it for unrelated samples (often called *negative pairs*). A notable example of contrastive learning is optimization of the so-called Information Noise Contrastive Estimation (InfoNCE) [317] objective. We borrow this framework to learn useful represen-

Figure 3.15: Training for trajectory clustering involves three losses: contrastive InfoNCE loss and two reconstruction errors (in data space and feature space).



tations for point trajectories. The loss can be formulated as following

$$-\frac{1}{|\mathcal{D}|}\sum_{(i,j,l,k)\in\mathcal{D}}\log\left(\frac{p_{ij}}{p_{ij}+p_{lk}}\right),$$
$$p_{ij}=\exp\left(-\|\mathbf{f}_i-\mathbf{f}_j\|_2^2\right), \tag{3.72}$$

where $\mathbf{f}_i$ are trajectory embeddings, $\mathcal{D}$ is the set of (i,j,l,k) where (i,k) belong to the same cluster and (l,k) are random pairs. In addition, we would like to be able to obtain (or reconstruct) the original trajectories from the network output. Hence we augment InfoNCE loss with the reconstruction loss which is computed both in the data space and feature space. Our training procedure is illustrated in Figure 3.15. We show in **Paper D** that the obtained trajectory embeddings can be used directly for clustering multiple observed motions, which makes the inference much more efficient. The learned embeddings can also be used to recover the underlying subspaces, which is particularly helpful for the completion algorithm described next.

To address potential missing data in the trajectories, we derive a simple iterative procedure for trajectory completion. The idea is to pass the trajectories through the encoder-decoder network (only visible parts in the 0^{th}

iteration, and complete tracks in the later iterations) to compute initialization and iterative updates, which is formulated as following

$$\begin{cases} \mathtt{B}_0 \leftarrow B_{\theta,\phi}(\mathbf{x}_{\text{vis}}, \mathbf{t}) \\ \bar{\mathbf{x}}_{i+1} \leftarrow \mathtt{A}(\mathtt{B}_i)\mathbf{x} \\ \mathtt{B}_{i+1} \leftarrow B_{\theta,\phi}(\mathbf{w} \odot \mathbf{x} + \bar{\mathbf{w}} \odot \bar{\mathbf{x}}_{i+1}, \mathbf{t}), \end{cases} \tag{3.73}$$

where $B_{\theta,\phi}(\cdot,\cdot)$ is the entire encoder-decoder model (shown in Figure 3.14). We demonstrate the efficiency of the proposed methodology in **Paper D**. Our learning-based approach to motion segmentation provides top clustering results in a fraction of a second.

CHAPTER 4

Summary of included papers

This chapter provides a summary of the included papers.

4.1 Paper A

Yaroslava Lochman, Kostiantyn Liepieshov, Jianhui Chen, Michal Perdoch, Christopher Zach, James Pritts
BabelCalib: A Universal Approach to Calibrating Central Cameras
Published in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15253-15262.

Summary: This paper proposes a general pipeline for offline calibration of central projection cameras. It particularly addresses the weak point of camera calibration, namely initialization that may cause system failure due to poor model choice and/or parameter guess and therefore may require manual intervention. The developed framework also extends upon the existing frameworks in a few ways: it integrates all most commonly used central projection models, and it supports multiple planar targets. The technical contributions

of the paper are three-fold. First, it proposes a sequence of simple solvers based on the radial fundamental relation between the image points and the target plane points, and the division model parameterization. The division model is a powerful camera model, the paper provides the empirical evidence that that this model can approximate sufficiently well (hence initialize) many central cameras, including wide field-of-view (such as fish eye) cameras. The simplicity of the derived solvers allows to use non-minimal samples of point correspondences that improves the overall initial accuracy. Second, the paper proposes the model-to-model regression step via non-linear least squares and obtains a linear solution for most of the commonly used target projection models. Third, we extensively validate the proposed method on the downstream task of absolute pose estimation, and demonstrate its state-of-the-art performance. As a result, the proposed method, called BabelCalib, can accurately and robustly calibrate pinhole cameras with additive distortion as well as omni-directional cameras and catadioptric rigs. The proposed framework is fully automatic, only requiring the user to choose the target camera projection model. It is robust to inaccurately localized corners, outlying detections and occluded targets.

Contributions: Y.L. contributed to idea generation, implemented the approach, performed all experiments, contributed to the analysis of the results as well as writing and illustrations. K.L. contributed to data processing and evaluation. J.C. contributed to evaluation and writing. M.P. contributed to the analysis of the results. C.Z. and J.P. contributed to idea generation, analysis of the results as well as writing.

4.2 Paper B

Carl Olsson, **Yaroslava Lochman**, Johan Malmport, Christopher Zach
Certifiably Optimal Anisotropic Rotation Averaging
Published in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, pp. 14856-14865

Summary: The methods proposed for rotation averaging mostly consider various *isotropic* error measurements. Since the observed relative rotations are usually obtained from image pairs and therefore are not direct measurements, the corresponding uncertainties in these estimates can be easily ex-

tracted via computing the Hessians $\mathtt{H}$ of the respective two-view optimizations. However, those are typically disregarded. In this paper, we propose a natural *anisotropic* extension to the standard chordal distance that measures the deviation using Frobenius distance [270] and is directly connected to assuming the Langevin distribution on rotations. Using the exponential map parameterization $\mathtt{Q} = e^{[\Delta\boldsymbol{\omega}]_\times}\tilde{\mathtt{R}}$, we obtain a second-order approximation $\frac{1}{2}\Delta\boldsymbol{\omega}^\top \mathtt{H}\Delta\boldsymbol{\omega} \approx \mathrm{tr}(\mathtt{M}) - \langle \mathtt{M}\tilde{\mathtt{R}}, \mathtt{Q}\rangle$ for a specific matrix $\mathtt{M} = \frac{1}{2}\mathrm{tr}(\mathtt{H})\mathtt{I} - \mathtt{H}$ that allows to incorporate two-view uncertainties and obtain a linear cost in $\mathtt{Q}$ that is important for an SDP relaxation. However, the cost matrices $\mathtt{M}$ are often indefinite. Consequently, we show that the standard relaxation ignoring the determinant constraint $\det(\mathtt{Q}) = 1$ [165] is never able to recover a desired rank-3 solution. We instead keep the determinant constraint leading to a new semidefinite relaxation, and obtain a certifiably optimal anisotropic RA framework. The resulting solver is able to recover global minima, and gives more accurate solutions than the isotropic counterparts on synthetic and real datasets.

Contributions: C.O. generated the idea and performed initial testing, contributed to writing and illustrations. C.O., Y.L. and C.Z. jointly contributed to the statistical analysis. J.M. contributed to writing. Y.L. and C.Z. implemented the approach, contributed to writing and illustrations. C.Z. ran intermediate experiments, and Y.L. performed final experiments, with results presented in the paper.

4.3 Paper C

Yaroslava Lochman, Carl Olsson, Christopher Zach

Fast and Robust Rotation Averaging with Anisotropic Coordinate Descent

Published in Proceedings of the British Machine Vision Conference, 2025

Summary: This paper proposes a fast solver for anisotropic rotation averaging where the uncertainties of optimized two-view relative rotations are incorporated into the optimization of absolute rotations. To derive a new solver, we get inspiration from a family of block-coordinate descent methods. We analyze the existing solvers proposed for the isotropic rotation averaging — we show that the recent rotation coordinate descent method [257] re-

tains SO(3)-membership throughout all iterations — and use our findings in constructing a simple algorithm for anisotropic rotation averaging. We obtain notably improved reconstructions for the challenging datasets with many symmetries, where the estimated relative rotations are very noisy and contain outliers, while being orders of magnitude faster than the recently proposed SDP solver for anisotropic rotation averaging.

Contributions: Y.L. generated the idea, implemented the approach, performed all experiments, contributed to the analysis of the results as well as writing and illustrations. C.O. and C.Z. contributed to the analysis of the results and writing.

4.4 Paper D

Yaroslava Lochman, Carl Olsson, Christopher Zach
Learned Trajectory Embedding for Subspace Clustering
Published in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 19092-19102.

Summary: In this paper, we look into ways to inject learning into a classical problem of motion segmentation. We build upon a disjoint subspaces assumption [294], that in the high-dimensional settings with low-dimensional subspaces, two subspaces only intersect at the origin. We therefore propose to learn a mapping from trajectories to embedding vectors that represent the generating motion. The obtained trajectory embeddings are useful for clustering multiple observed motions, but are also trained to contain sufficient information to recover the parameters of the underlying motion by utilizing a geometric loss. We therefore are able to use only weak supervision from given motion segmentation to train this mapping. The entire algorithm consisting of trajectory embedding, clustering and motion parameter estimation is highly efficient. We show state-of-the-art performance of our proposed method on the standard benchmark datasets for trajectory-based motion segmentation on full sequences and its competitiveness on the occluded sequences.

Contributions: Y.L. contributed to the idea generation, implemented the approach, performed all experiments, contributed to the analysis of the results as well as writing and illustrations. C.O. and C.Z. contributed to the idea generation, analysis of the results as well as writing.

CHAPTER 5

Concluding Remarks and Future Work

To conclude, this thesis touches upon several topics that are relevant to the 3D vision research community. With its focus on reliability, robustness, and overall performance, it unveils several limitations of the underlying components of the typical 3D reconstruction pipelines—the issues that could lead to unsatisfactory performance or a complete system failure. To mitigate these issues, the thesis proposes a set of solutions that leverage various techniques from geometry, optimization, and deep machine learning. In particular:

(1) The thesis advocates for and proposes a versatile solution to calibrating central cameras with various lens configurations ranging from pinhole to fish eye and catadioptric lenses. It relies on simple techniques and solvers, and it leverages the assumption of high inlier ratio specific to the problem of camera calibration. With this, it proves to be very stable.

(2) The thesis also proposes a way to theoretically and practically implement uncertainty propagation into the optimization of absolute rotations. The estimated uncertainties of the two-view relative rotations are easily available as Hessians of the corresponding two-view local optimizations and, as the thesis advocates, should not be discarded. The uncertainties provide a crucial information about the reliability of the propagated information which affects

the reconstruction accuracy.

(3) The thesis empirically discovers a useful structure in the high-dimensional point trajectory data that connects to the underlying motion models and can therefore be leveraged for motion clustering, a difficult chicken-and-egg problem. A metric learning approach is proposed that is designed to preserve the geometric problem structure via learning to minimize the metric loss together with the reconstruction error.

5.1 Outlook

This section outlines some potential directions for future research.

Adapting to other modalities and setups

This thesis focuses on the methods that do not rely on the visual information directly and work with either points or transformation matrices. Extending the presented approaches such that they could be used in the reconstruction pipelines for systems with different types of sensors is an interesting direction. It can be more straightforward for some devices—*e.g.* infrared cameras that share similarities in projection with visual cameras although present new difficulties in detection due to lower resolution and higher noise levels—but tricky for the others devices that operate very differently compared to the cameras—*e.g.* LiDARs that emit laser pulses to produce 3D point clouds. Drawing parallels with medical imaging devices could also lead to interesting results.

Below, we analyze each problem relevant to this thesis in terms of potential extensions to other sensor and data modalities.

In camera calibration, the proposed solvers rely on the camera geometry as presented in **Paper A**, therefore it is not obvious if similar techniques could be translated to calibrating sensors with other modalities. However, the approach can be directly integrated into calibrating camera rigs or more complex systems with LiDARs, radars, inertial measurement units (IMUs), ultrasonics *etc.* It would be interesting to apply the rotation averaging framework presented in **Paper B** and **Paper C** to the extrinsic part (*i.e.* aligning trajectories) of calibrating such systems [260], [318].

In rotation averaging, the inputs and outputs are rotation matrices that

partially characterize sensor poses. IMUs are the commonly used devices in autonomous navigation that provide information about the same type of data, *i.e.* orientation. This information could potentially be integrated into the proposed rotation averaging pipeline. For example, recall that the key operation of the ACD algorithm proposed in **Paper C** is

$$\text{project}_{\text{SO(3)}}\left(\sum_i \mathtt{N}_{ik}^{\top}\mathtt{R}_i^{(t)}\right). \tag{5.1}$$

Consider the camera-IMU rig that is pre-calibrated. The relative orientations estimated by IMUs (and aligned *w.r.t.* the coordinate system of the rig) can be added as additional measurements $\mathtt{N}_{jk}$ to the corresponding rotations $\mathtt{R}_j$. The IMUs estimate orientations using accelerometer and gyroscope measurements, which are of a different modality than point correspondences. An interesting question is therefore: how to characterize the IMU uncertainties and put them on the same scale as uncertainties obtained from the point correspondence-based two-view reconstructions?

In motion segmentation, a typical input is the 2D point trajectories. Classic approaches are designed to specifically work with this type of input. One benefit of learning-based approaches is their flexibility and ease of adjusting or adding different input modalities. A subfield of multi-modal learning [319]–[321] deals with this type of problem. It would be interesting to see if incorporating visual information, *i.e.* the readily-available image sequences from which the 2D tracks were extracted, would benefit the trajectory learning approach proposed in **Paper D**. On the other hand, since the network proposed in **Paper D** does not need visual information, it would also be interesting to explore its applicability to processing event-based data [322]. An event camera captures changes in intensities (these changes are called *events*), and it does it at a very high speed, in micro-seconds. Recent works [323], [324] show promising results in point tracking from event camera streams.

A-priori certificates for anisotropic rotation averaging

In **Paper B**, we show how anisotropic costs can be incorporated in certifiably optimal rotation averaging. We also demonstrate how existing solvers, designed for isotropic situations, fail in the anisotropic setting. We propose a stronger relaxation and show *empirically* that it is able to recover global

optima. However, there are no *theoretical* guarantees of recovering global optima. The methods used in isotropic rotation averaging [156] to obtain explicit noise bounds do not easily generalize to the anisotropic setting. Deriving a-priori certificates appears to be very difficult, but it could be an important future direction.

Semi-supervised learning of trajectory embeddings

An interesting and potentially important future direction for trajectory representation learning approach presented in **Paper D** is obtaining a general trajectory embedding useful across various datasets and problems. We found that the trajectory datasets with motion cluster annotations are of very limited size. Aiming to address the data generalization capabilities of the motion segmentation network, we discovered various data of unlabelled trajectories that are of much larger scales. A natural idea is to adopt a self- or semi-supervised methodology [325], [326] for learning a general trajectory representation. This could be done in different ways. For example, one could change the sampling strategy and modify the objective to accommodate for the missing labels in some of the inputs. One attractive property of this problem is that the reconstruction can be easily computed without the need to learn the decoder, and supervision though geometric residuals can always be used. To improve the reliability of the prediction, a context-conditioning where the context is represented by a set of trajectories extracted from the same scene can be used. The increasingly popular teacher-student approach [325], [327]–[329] should also be considered.

References

[1] J. Morlana, J. D. Tardós, and J. Montiel, "Colonmapper: Topological mapping and localization for colonoscopy", in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 6329–6336.

[2] J. Morlana, J. D. Tardós, and J. M. Montiel, "Topological slam in colonoscopies leveraging deep features and topological priors", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2024, pp. 733–743.

[3] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer", *International journal of computer assisted radiology and surgery*, vol. 9, no. 2, pp. 283–293, 2014.

[4] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians", *Computerized medical imaging and graphics*, vol. 43, pp. 99–111, 2015.

[5] K. Bera, K. A. Schalper, D. L. Rimm, V. Velcheti, and A. Madabhushi, "Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology", *Nature reviews Clinical oncology*, vol. 16, no. 11, pp. 703–715, 2019.

[6] V. Wåhlstrand Skärström, L. Johansson, J. Alvén, M. Lorentzon, and I. Häggström, "Explainable vertebral fracture analysis with uncertainty

estimation using differentiable rule-based classification", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2024, pp. 318–328.

[7] M. Martinez and R. Stiefelhagen, "Breath rate monitoring during sleep using near-ir imagery and pca", in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, 2012, pp. 3472–3475.

[8] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization", in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 4178–4185.

[9] I. H. Kalfas, "Machine vision navigation in spine surgery", *Frontiers in Surgery*, vol. 8, p. 640 554, 2021.

[10] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "Lookout: Diverse multi-future prediction and planning for self-driving", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 107–16 116.

[11] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies", *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[12] S. Teng, X. Hu, P. Deng, *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives", *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3692–3711, 2023.

[13] K. Hohm, H. M. Hofstede, and H. Tolle, "Robot assisted disassembly of electronic devices", in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, IEEE, vol. 2, 2000, pp. 1273–1278.

[14] F. Yumbla, M. Abeyabas, T. Luong, J.-S. Yi, and H. Moon, "Preliminary connector recognition system based on image processing for wire harness assembly tasks", in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2020, pp. 1146–1150.

[15] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition", *Sensors*, vol. 17, no. 9, p. 2022, 2017.

[16] T. A. Shaikh, T. Rasool, and F. R. Lone, "Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming", *Computers and Electronics in Agriculture*, vol. 198, p. 107 119, 2022.

[17] D. P. McMullen, G. Hotson, K. D. Katyal, *et al.*, "Demonstration of a semi-autonomous hybrid brain–machine interface using human intracranial eeg, eye tracking, and computer vision to control a robotic upper limb prosthetic", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 784–796, 2013.

[18] M. Martinez, A. Roitberg, D. Koester, R. Stiefelhagen, and B. Schauerte, "Using technology developed for autonomous cars to help navigate blind people", in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1424–1432.

[19] X. Chen, X. Huang, Y. Wang, and X. Gao, "Combination of augmented reality based brain-computer interface and computer vision for high-level control of a robotic arm", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 3140–3147, 2020.

[20] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[21] 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*, Springer-Verlag, 1999, pp. 298–372.

[22] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[23] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[24] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[25] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2010.

[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf", in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[28] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections", *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.

[29] R. I. Hartley, "In defense of the eight-point algorithm", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[30] 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.

[31] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[32] D. C. Brown, "The bundle adjustment — progress and prospects", in *Int. Archives Photogrammetry*, vol. 21, 1976, pp. 1–33.

[33] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d", *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.

[34] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system", *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

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

[36] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks", *arXiv*, 2019.

[37] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg, "Roma: Robust dense feature matching", in *Proceedings of the IEEE / CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 790–19 800.

[38] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[39] Y. Cabon, L. Stoffl, L. Antsfeld, *et al.*, "Must3r: Multi-view network for stereo 3d reconstruction", in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1050–1060.

[40] C. M. Bishop, "Pattern recognition and machine learning", *Choice Reviews Online*, vol. 44, no. 09, pp. 44–5091-44-5091, 2007, ISSN: 0009-4978.

[41] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[42] D. A. Roberts, S. Yaida, and B. Hanin, *The principles of deep learning theory*. Cambridge University Press Cambridge, MA, USA, 2022, vol. 46.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, vol. 25, 2012.

[45] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data", *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[46] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[47] O. Siméoni, H. V. Vo, M. Seitzer, *et al.*, "Dinov3", *arXiv preprint arXiv:2508.10104*, 2025.

[48] O. Enqvist, F. Kahl, and C. Olsson, "Non-sequential structure from motion", in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 264–271.

[49] S. Agarwal, Y. Furukawa, N. Snavely, *et al.*, "Building rome in a day", *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[50] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 1689–1696.

[51] A. Rosinol, A. Violette, M. Abate, *et al.*, "Kimera: From slam to spatial perception with 3d dynamic scene graphs", *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

[52] L. Pan, D. Baráth, M. Pollefeys, and J. L. Schönberger, *Global structure-from-motion revisited*, 2024.

[53] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion*.", *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[54] J. Ventura, V. Larsson, and F. Kahl, "Uncalibrated structure from motion on a sphere", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025, pp. 69–78.

[55] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[56] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006, ISSN: 0162-8828.

[57] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras", in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006, pp. 5695–5701.

[58] B. Khomutenko, G. Garcia, and P. Martinet, "An enhanced unified camera model", *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 137–144, 2015.

[59] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model", in *2018 International Conference on 3D Vision, 3DV 2018, Verona, Italy, September 5-8, 2018*, IEEE Computer Society, 2018, pp. 552–560.

[60] G. Zhang, V. Larsson, and D. Barath, "Revisiting rotation averaging: Uncertainties and robust losses", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 215–17 224.

[61] S. N. Parizi, K. He, R. Aghajani, S. Sclaroff, and P. Felzenszwalb, "Generalized majorization-minimization", in *International Conference on Machine Learning*, PMLR, 2019, pp. 5022–5031.

[62] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting", *International journal of computer vision*, vol. 97, no. 2, pp. 123–147, 2012.

[63] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of j-linkage for multi-model fitting", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3954–3961.

[64] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3d reconstruction of a moving point from a series of 2d projections", in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11*, Springer, 2010, pp. 158–171.

[65] 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.

[66] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses", *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[67] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion", in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, IEEE, 2006, pp. 45–45.

[68] C. B. Duane, "Close-range camera calibration", *Photogramm. Eng*, vol. 37, no. 8, pp. 855–866, 1971.

[69] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids", in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, 2007, pp. 3945–3950.

[70] F. Devernay and O. Faugeras, "Straight lines have to be straight", *Machine vision and applications*, vol. 13, no. 1, pp. 14–24, 2001.

[71] S. Urban, J. Leitloff, and S. Hinz, "Improved wide-angle, fisheye and omnidirectional camera calibration", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 72–79, 2015.

[72] V. Larsson, T. Sattler, Z. Kukelova, and M. Pollefeys, "Revisiting radial distortion absolute pose", in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1062–1071.

[73] F. Camposeco, T. Sattler, and M. Pollefeys, "Non-parametric structure-based calibration of radially symmetric cameras", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2192–2200.

[74] L. Pan, M. Pollefeys, and V. Larsson, "Camera pose estimation using implicit distortion models", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 819–12 828.

[75] M. D. Grossberg and S. K. Nayar, "A general imaging model and a method for finding its parameters", in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, IEEE, vol. 2, 2001, pp. 108–115.

[76] P. Sturm and S. Ramalingam, "A generic concept for camera calibration", in *European Conference on Computer Vision*, Springer, 2004, pp. 1–13.

[77] S. Ramalingam and P. Sturm, "A unifying model for camera calibration", *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1309–1319, 2016.

[78] J. Beck and C. Stiller, "Generalized b-spline camera model", in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 2137–2142.

[79] T. Schops, V. Larsson, M. Pollefeys, and T. Sattler, "Why having 10,000 parameters in your camera model is better than twelve", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2535–2544.

[80] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks", *Advances in neural information processing systems*, vol. 31, 2018.

[81] R. Hartley and H. Li, "An efficient hidden variable approach to minimal-case camera motion estimation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 12, pp. 2303–2314, 2012.

[82] A. Karimian and R. Tron, "Essential matrix estimation using convex relaxations in orthogonal space", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 142–17 152.

[83] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting", *Image and vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[84] R. I. Hartley and P. Sturm, "Triangulation", *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[85] P. Lindstrom, "Triangulation made easy", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1554–1561.

[86] P. D. Sampson, "Fitting conic sections to "very scattered" data: An iterative refinement of the bookstein algorithm", *Computer graphics and image processing*, vol. 18, no. 1, pp. 97–108, 1982.

[87] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis", *International journal of computer vision*, vol. 17, no. 1, pp. 43–75, 1996.

[88] F. Rydell, A. Torres, and V. Larsson, "Revisiting sampson approximations for geometric estimation problems", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4990–4998.

[89] M. Pollefeys, F. Verbiest, and L. Van Gool, "Surviving dominant planes in uncalibrated structure and motion recovery", in *European conference on computer vision*, Springer, 2002, pp. 837–851.

[90] P. H. Torr, "An assessment of information criteria for motion model selection", in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 1997, pp. 47–52.

[91] P. H. Torr, A. Zisserman, and S. J. Maybank, "Robust detection of degenerate configurations while estimating the fundamental matrix", *Computer vision and image understanding*, vol. 71, no. 3, pp. 312–333, 1998.

[92] K. Mikolajczyk, T. Tuytelaars, C. Schmid, *et al.*, "A comparison of affine region detectors", *International journal of computer vision*, vol. 65, pp. 43–72, 2005.

[93] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173–5182.

[94] I. Rocco, "Neural architectures for estimating correspondences between images", Ph.D. dissertation, Université Paris sciences et lettres, 2020.

[95] J. Edstedt, "Towards the next generation of 3d reconstruction", Ph.D. dissertation, 2025.

[96] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 337–349, 2018.

[97] D. Mishkin, F. Radenovic, and J. Matas, "Repeatability is not enough: Learning affine regions via discriminability", in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 284–300.

[98] M. Dusmanu, I. Rocco, T. Pajdla, *et al.*, "D2-net: A trainable cnn for joint description and detection of local features", in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[99] J. Edstedt, G. Bökman, M. Wadenbäck, and M. Felsberg, "Dedode: Detect, don't describe–describe, don't detect for local feature matching", *arXiv preprint arXiv:2308.08479*, 2023.

[100] J. Edstedt, G. Bökman, and Z. Zhao, "Dedode v2: Analyzing and improving the dedode keypoint detector", in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 4245–4253.

[101] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 17 627–17 638.

[102] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[103] J. Edstedt, I. Athanasiadis, M. Wadenbäck, and M. Felsberg, "Dkm: Dense kernelized feature matching for geometry estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 765–17 775.

[104] J. Astermark, A. Heyden, and V. Larsson, "Dense match summarization for faster two-view estimation", in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1093–1102.

[105] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1309–1321, 2007.

[106] L. Lucchese and S. K. Mitra, "Using saddle points for subpixel feature detection in camera calibration targets", in *Asia-Pacific Conference on Circuits and Systems*, IEEE, vol. 2, 2002, pp. 191–195.

[107] J. Mallon and P. F. Whelan, "Which pattern? biasing aspects of planar calibration patterns and detection methods", *Pattern recognition letters*, vol. 28, no. 8, pp. 921–930, 2007.

[108] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images", in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 3121–3126.

[109] A. Richardson, J. Strom, and E. Olson, "Aprilcal: Assisted and repeatable camera calibration", in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 1814–1821.

[110] P. Fuersattel, S. Dotenco, S. Placht, M. Balda, A. Maier, and C. Riess, "Ocpad—occluded checkerboard pattern detector", in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2016, pp. 1–9.

[111] M. Fiala, "Artag, a fiducial marker system using digital techniques", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 2, 2005, pp. 590–596.

[112] E. Olson, "Apriltag: A robust and flexible visual fiducial system", in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3400–3407.

[113] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion", *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[114] J. Heikkila, "Geometric camera calibration using circular control points", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1066–1077, 2002.

[115] H. Ha, M. Perdoch, H. Alismail, I. S. Kweon, and Y. Sheikh, "Deltille grids for geometric camera calibration", in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5354–5362.

[116] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[117] K. Levenberg, "A method for the solution of certain non-linear problems in least squares", *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[118] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters", *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[119] J. J. Moré, "The levenberg-marquardt algorithm: Implementation and theory", in *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*, Springer, 2006, pp. 105–116.

[120] K. Konolige and W. Garage, "Sparse sparse bundle adjustment.", in *BMVC*, vol. 10, 2010, pp. 102–1.

[121] 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. 149–170, 1984.

[122] C. Engels, H. Stewénius, and D. Nistér, "Bundle adjustment rules", *Photogrammetric computer vision*, vol. 2, no. 32, 2006.

[123] M. Zollhöfer, M. Nießner, S. Izadi, *et al.*, "Real-time non-rigid reconstruction using an rgb-d camera", *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, pp. 1–12, 2014.

[124] C. Zach, "Robust bundle adjustment revisited", in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 772–787.

[125] C. Zach and G. Bourmaud, "Iterated lifting for robust cost optimization", in *BMVC*, 2017.

[126] C. Zach and G. Bourmaud, "Descending, lifting or smoothing: Secrets of robust cost optimization", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 547–562.

[127] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[128] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares", *Communications in Statistics-theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[129] D. Coleman, P. Holland, N. Kaden, V. Klema, and S. C. Peters, "A system of subroutines for iteratively reweighted least squares computations", *ACM Transactions on Mathematical Software (TOMS)*, vol. 6, no. 3, pp. 327–336, 1980.

[130] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery", *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 63, no. 1, pp. 1–38, 2010.

[131] A. Chatterjee and V. M. Govindu, "Efficient and robust large-scale rotation averaging", in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 521–528.

[132] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock, "On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision", *SIAM Journal on Imaging Sciences*, vol. 8, no. 1, pp. 331–372, 2015.

[133] K. Aftab and R. Hartley, "Convergence of iteratively re-weighted least squares to robust m-estimators", in *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, 2015, pp. 480–487.

[134] S. Bouaziz, A. Tagliasacchi, H. Li, and M. Pauly, "Modern techniques and applications for real-time non-rigid registration", in *SIGGRAPH ASIA 2016 Courses*, ser. SA '16, Association for Computing Machinery, 2016.

[135] L. Peng, C. Kümmerle, and R. Vidal, "On the convergence of irls and its variants in outlier-robust estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 808–17 818.

[136] C. Zach and H. Le, "Truncated inference for latent variable optimization problems: Application to robust estimation and learning", in *European Conference on Computer Vision*, Springer, 2020, pp. 464–480.

[137] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 14, no. 03, pp. 367–383, 1992.

[138] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[139] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky, "A minimal solution for relative pose with unknown focal length", *Image and Vision Computing*, vol. 26, no. 7, pp. 871–877, 2008.

[140] Z. Kukelova, J. Heller, M. Bujnak, and T. Pajdla, "Radial distortion homography", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 639–647.

[141] V. Larsson, K. Astrom, and M. Oskarsson, "Polynomial solvers for saturated ideals", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2288–2297.

[142] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers", in *European Conference on Computer Vision*, Springer, 2008, pp. 302–315.

[143] V. Larsson, K. Astrom, and M. Oskarsson, "Efficient solvers for minimal problems by syzygy-based reduction", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 820–829.

[144] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac", in *Pattern Recognition*, B. Michaelis and G. Krell, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 236–243.

[145] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry", *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[146] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 772–779.

[147] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus", in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 220–226.

[148] F. Kluger, E. Brachmann, H. Ackermann, C. Rother, M. Y. Yang, and B. Rosenhahn, "Consac: Robust multi-model fitting by conditional sample consensus", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4634–4643.

[149] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendialdua, and B. Sierra, "Ransac for robotic applications: A survey", *Sensors*, vol. 23, no. 1, p. 327, 2022.

[150] L. Brynte, V. Larsson, J. P. Iglesias, C. Olsson, and F. Kahl, "On the tightness of semidefinite relaxations for rotation estimation", *J. Math. Imaging Vis.*, vol. 64, no. 1, pp. 57–67, 2022.

[151] 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.

[152] J. Briales and J. Gonzalez-Jimenez, "Convex global 3d registration with lagrangian duality", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4960–4969.

[153] J. P. Iglesias, C. Olsson, and F. Kahl, "Global optimality for point set registration using semidefinite programming", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8287–8295.

[154] G. Schweighofer and A. Pinz, "Globally optimal o (n) solution to the pnp problem for general camera models.", in *BMVC*, 2008, pp. 1–10.

[155] J. Fredriksson and C. Olsson, "Simultaneous multiple rotation averaging using lagrangian duality", in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 245–258.

[156] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin, "Rotation averaging with the chordal distance: Global minimizers and strong duality", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 256–268, 2021.

[157] C. Olsson, Y. Lochman, J. Malmport, and C. Zach, "Certifiably optimal anisotropic rotation averaging", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025, pp. 14 856–14 865.

[158] M. Giamou, Z. Ma, V. Peretroukhin, and J. Kelly, "Certifiably globally optimal extrinsic calibration from per-sensor egomotion", *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 367–374, 2019.

[159] E. Wise, M. Giamou, S. Khoubyarian, A. Grover, and J. Kelly, "Certifiably optimal monocular hand-eye calibration", in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, IEEE, 2020, pp. 271–278.

[160] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2d slam", in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 4589–4596.

[161] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification", *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[162] L. Carlone and G. C. Calafiore, "Convex relaxations for pose graph optimization with outliers", *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1160–1167, 2018.

[163] M. ApS, *The mosek optimization toolbox for matlab manual. version 9.0.* 2019.

[164] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab", in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, IEEE, 2004, pp. 284–289.

[165] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin, "Rotation averaging and strong duality", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 127–135.

[166] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group", *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 95–125, 2019.

[167] A. Nemirovski, "Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems", *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 229–251, 2004.

[168] X.-Y. Zhao, D. Sun, and K.-C. Toh, "A newton-cg augmented lagrangian method for semidefinite programming", *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1737–1765, 2010.

[169] E. Hazan, "Sparse approximate solutions to semidefinite programs", in *Latin American symposium on theoretical informatics*, Springer, 2008, pp. 306–316.

[170] A. Yurtsever, M. Udell, J. Tropp, and V. Cevher, "Sketchy decisions: Convex low-rank matrix optimization with optimal storage", in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1188–1196.

[171] A. Yurtsever, O. Fercoq, and V. Cevher, "A conditional-gradient-based augmented lagrangian framework", in *International Conference on Machine Learning*, PMLR, 2019, pp. 7272–7281.

[172] A. Majumdar, G. Hall, and A. A. Ahmadi, "Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 331–360, 2020.

[173] A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher, "Scalable semidefinite programming", *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 1, pp. 171–200, 2021.

[174] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[175] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis", *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[176] C. Olsson and O. Enqvist, "Stable structure from motion for unordered image collections", in *Image Analysis*, A. Heyden and F. Kahl, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 524–535.

[177] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering.", *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[178] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[179] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

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

[181] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams", in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, IEEE, vol. 2, 2000, pp. 690–696.

[182] J. H. Hong and C. Zach, "Pose: Pseudo object space error for initialization-free bundle adjustment", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1876–1885.

[183] J. H. Hong, C. Zach, A. Fitzgibbon, and R. Cipolla, "Projective bundle adjustment from arbitrary initialization using the variable projection method", in *European Conference on Computer Vision*, Springer, 2016, pp. 477–493.

[184] K. Kanatani, Y. Sugaya, and H. Niitsuma, "Triangulation from two views revisited: Hartley-sturm vs. optimal correction.", in *BMVC*, 2008, pp. 1–10.

[185] S. H. Lee and J. Civera, "Triangulation: Why optimize?", *arXiv preprint arXiv:1907.11917*, 2019.

[186] S. H. Lee and J. Civera, "Closed-form optimal two-view triangulation based on angular errors", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[187] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp", in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 383–390.

[188] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution", in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2344–2351.

[189] B. Triggs, "Factorization methods for projective structure and motion", in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA: IEEE, 1996, pp. 845–851, ISBN: 978-0-8186-7259-0.

[190] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion", in *Computer Vision — ECCV '96*, B. Buxton and R. Cipolla, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 709–720.

[191] R. Hartley and F. Schaffalitzky, "Powerfactorization: 3d reconstruction with missing or uncertain data", in *Australia-Japan advanced workshop on computer vision*, vol. 74, 2003, pp. 76–85.

[192] Q. Ke and T. Kanade, "Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 739–746.

[193] J. Oliensis and R. Hartley, "Iterative Extensions of the Sturm/Triggs Algorithm: Convergence and Nonconvergence", en, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2217–2233, 2007.

[194] J. P. Iglesias, A. Nilsson, and C. Olsson, "expOSE: Accurate initialization-free projective factorization using exponential regularization", *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, pp. 8959–8968, 2023.

[195] C. Olsson and A. Nilsson, "Towards initialization-free calibrated bundle adjustment", *arXiv preprint arXiv:2506.23808*, 2025.

[196] R. Toldo, R. Gherardi, M. Farenzena, and A. Fusiello, "Hierarchical structure-and-motion recovery from uncalibrated images", *Computer Vision and Image Understanding*, vol. 140, pp. 127–143, 2015.

[197] K. Ni and F. Dellaert, "Hypersfm", in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 2012, pp. 144–151.

[198] Á. P. Bustos, T.-J. Chin, A. Eriksson, and I. Reid, "Visual slam: Why bundle adjust?", in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE Press, 2019, pp. 2385–2391.

[199] Y. Kasten, A. Geifman, M. Galun, and R. Basri, "Gpsfm: Global projective sfm using algebraic constraints on multi-view fundamental matrices", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3259–3267.

[200] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion", in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3248–3255.

[201] K. Wilson and N. Snavely, "Robust global translations with 1dsfm", in *European conference on computer vision*, Springer, 2014, pp. 61–75.

[202] B. Zhuang, L.-F. Cheong, and G. H. Lee, "Baseline desensitizing in translation averaging", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4539–4547.

[203] L. Manam and V. M. Govindu, "Fusing directions and displacements in translation averaging", in *2024 International Conference on 3D Vision (3DV)*, IEEE, 2024, pp. 75–84.

[204] C. Rother and S. Carlsson, "Linear multi view reconstruction with missing data", in *European Conference on Computer Vision*, Springer, 2002, pp. 309–324.

[205] D. Martinec and T. Pajdla, "Robust rotation and translation estimation in multiview reconstruction", in *2007 IEEE conference on computer vision and pattern recognition*, IEEE, 2007, pp. 1–8.

[206] S. H. Lee and J. Civera, "Hara: A hierarchical approach for robust rotation averaging", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 777–15 786.

[207] H. Cui, X. Gao, S. Shen, and Z. Hu, "Hsfm: Hybrid structure-from-motion", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1212–1221.

[208] H. Li, "Multi-view structure computation without explicitly estimating motion", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 2777–2784.

[209] N. Jiang, D. Lin, M. N. Do, and J. Lu, "Direct structure estimation for 3d reconstruction", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2655–2663.

[210] F. Arrigoni, "A taxonomy of structure from motion methods", *arXiv preprint arXiv:2505.15814*, 2025.

[211] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal graphs for efficient structure from motion", in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[212] S. Liu, Y. Gao, T. Zhang, *et al.*, "Robust incremental structure-from-motion with hybrid features", in *European Conference on Computer Vision*, Springer, 2024, pp. 249–269.

[213] L. Brynte, J. P. Iglesias, C. Olsson, and F. Kahl, "Learning structure-from-motion with graph attention networks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4808–4817.

[214] B. Bhowmick, S. Patra, A. Chatterjee, V. M. Govindu, and S. Banerjee, "Divide and conquer: A hierarchical approach to large-scale structure-from-motion", *Computer Vision and Image Understanding*, vol. 157, pp. 190–205, 2017.

[215] L. Manam and V. M. Govindu, "Sensitivity in translation averaging", *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 740–62 763, 2023.

[216] C. Olsson, A. Eriksson, and R. Hartley, "Outlier removal using duality", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1450–1457.

[217] R. Hartley and F. Schaffalitzky, "L_∞ minimization in geometric reconstruction problems", in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 1, 2004, pp. I–I.

[218] Q. Ke and T. Kanade, "Quasiconvex optimization for robust geometric reconstruction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1834–1847, 2007.

[219] C. Olsson, A. P. Eriksson, and F. Kahl, "Efficient optimization for L_∞-problems using pseudoconvexity", in *2007 IEEE 11th International Conference on Computer Vision*, IEEE, 2007, pp. 1–8.

[220] F. Kahl and R. Hartley, "Multiple-view geometry under the L_∞-norm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1603–1617, 2008.

[221] S. Agarwal, N. Snavely, and S. M. Seitz, "Fast algorithms for L_∞ problems in multiview geometry", in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[222] Z. Dai, Y. Wu, F. Zhang, and H. Wang, "A novel fast method for L_∞ problems in multiview geometry", in *European Conference on Computer Vision*, Springer, 2012, pp. 116–129.

[223] A. Eriksson and M. Isaksson, "Pseudoconvex proximal splitting for l-∞ problems in multiview geometry", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4066–4073.

[224] Q. Zhang, T.-J. Chin, and H. M. Le, "A fast resection-intersection method for the known rotation problem", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3012–3021.

[225] M. Dusmanu, J. L. Schönberger, and M. Pollefeys, "Multi-view optimization of local feature geometry", in *European Conference on Computer Vision*, Springer, 2020, pp. 670–686.

[226] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, "Pixel-perfect structure-from-motion with featuremetric refinement", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5987–5997.

[227] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow", in *European conference on computer vision*, Springer, 2020, pp. 402–419.

[228] C. Doersch, A. Gupta, L. Markeeva, *et al.*, "Tap-vid: A benchmark for tracking any point in a video", *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 610–13 626, 2022.

[229] A. W. Harley, Z. Fang, and K. Fragkiadaki, "Particle video revisited: Tracking through occlusions using point trajectories", in *European Conference on Computer Vision*, Springer, 2022, pp. 59–75.

[230] Y. Zheng, A. W. Harley, B. Shen, G. Wetzstein, and L. J. Guibas, "Pointodyssey: A large-scale synthetic dataset for long-term point tracking", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 855–19 865.

[231] C. Doersch, Y. Yang, M. Vecerik, *et al.*, "Tapir: Tracking any point with per-frame initialization and temporal refinement", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 061–10 072.

[232] Q. Wang, Y.-Y. Chang, R. Cai, *et al.*, "Tracking everything everywhere all at once", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 795–19 806.

[233] N. Tumanyan, A. Singer, S. Bagon, and T. Dekel, "Dino-tracker: Taming dino for self-supervised point tracking in a single video", in *European Conference on Computer Vision*, Springer, 2024, pp. 367–385.

[234] H. Wildenauer and A. Hanbury, "Robust camera self-calibration from monocular images of manhattan worlds", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2831–2838.

[235] H. Wildenauer and B. Micusik, "Closed form solution for radial distortion estimation from a single vanishing point.", in *Proceedings of the British Machine Vision Conference*, vol. 1, BMVA Press, 2013, pp. 106.1–106.11.

[236] M. Antunes, J. P. Barreto, D. Aouada, and B. Ottersten, "Unsupervised vanishing point detection and camera calibration from a single manhattan image with radial distortion", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6691–6699.

[237] Y. Lochman, O. Dobosevych, R. Hryniv, and J. Pritts, "Minimal solvers for single-view lens-distorted camera auto-calibration", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2887–2896.

[238] L. Jin, J. Zhang, Y. Hold-Geoffroy, *et al.*, "Perspective fields for single image camera calibration", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 307–17 316.

[239] A. Veicht, P.-E. Sarlin, P. Lindenberger, and M. Pollefeys, "Geocalib: Learning single-image calibration with geometric optimization", in *European Conference on Computer Vision*, Springer, 2024, pp. 1–20.

[240] J. Tirado-Garín and J. Civera, "Anycalib: On-manifold learning for model-agnostic single-view camera calibration", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 8044–8055.

[241] J. Kannala and S. Brandt, "A generic camera calibration method for fish-eye lenses", *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, vol. 1, pp. 10–13, 2004.

[242] J. Heikkila and O. Silvén, "A four-step camera calibration procedure with implicit image correction", in *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, IEEE, 1997, pp. 1106–1112.

[243] 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*, IEEE, vol. 1, 2001, pp. I–I.

[244] P. Sturm and S. Ramalingam, "A generic concept for camera calibration", in *European Conference on Computer Vision*, Springer, 2004, pp. 1–13.

[245] D. Claus and A. W. Fitzgibbon, "A rational function lens distortion model for general cameras", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 213–219.

[246] B. Li, L. Heng, K. Koser, and M. Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern", in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 1301–1307.

[247] S. Ramalingam, P. Sturm, and S. K. Lodha, "Towards complete generic camera calibration", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 1093–1098.

[248] F. Camposeco, T. Sattler, and M. Pollefeys, "Non-parametric structure-based calibration of radially symmetric cameras", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2192–2200.

[249] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 801–808.

[250] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications", in *European conference on computer vision*, Springer, 2000, pp. 445–461.

[251] J. P. Barreto and H. Araujo, "Issues on the geometry of central catadioptric image formation", in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, vol. 2, 2001, pp. II–II.

[252] G. Nakano, "A simple direct solution to the perspective-three-point problem.", in *BMVC*, 2019, p. 26.

[253] V. M. Govindu, "Lie-algebraic averaging for globally consistent motion estimation", in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. I-684–I-691.

[254] R. Tron and R. Vidal, "Distributed 3-d localization of camera sensor networks from 2-d image measurements", *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3325–3340, 2014.

[255] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri, "Global motion estimation from point matches", *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 81–88, 2012.

[256] F. Dellaert, D. M. Rosen, J. Wu, R. E. Mahony, and L. Carlone, "Shonan rotation averaging: Global optimality by surfing $\mathrm{so(p)}^{\mathrm{n}}$", *CoRR*, vol. abs/2008.02737, 2020.

[257] A. Parra, S.-F. Chng, T.-J. Chin, A. Eriksson, and I. Reid, "Rotation coordinate descent for fast globally optimal rotation averaging", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4298–4307.

[258] A. Chiuso, G. Picci, and S. Soatto, "Wide-sense estimation on the special orthogonal group.", *Commun. Inf. Syst.*, vol. 8, no. 3, pp. 185–200, 2008.

[259] V. M. Govindu, "Combining two-view constraints for motion estimation", in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, IEEE, vol. 2, 2001, pp. II-218–II-225.

[260] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging", *International Journal of Computer Vision*, vol. 103, no. 3, pp. 267–305, 2013, ISSN: 09205691.

[261] 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)*, IEEE, 2015, pp. 4597–4604.

[262] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1997, vol. 28.

[263] R. Sanyal, F. Sottile, and B. Sturmfels, "Orbitopes", *Mathematika*, vol. 57, no. 2, pp. 275–314, 2011.

[264] J. Saunderson, P. A. Parrilo, and A. S. Willsky, "Semidefinite descriptions of the convex hull of rotation matrices", *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1314–1343, 2015.

[265] T. Schops, J. L. Schonberger, S. Galliani, *et al.*, "A multi-view stereo benchmark with high-resolution images and multi-camera videos", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3260–3269.

[266] V. M. Govindu, "Robustness in motion averaging", in *Asian conference on computer vision*, Springer, 2006, pp. 457–466.

[267] C. Zach, M. Klopschitz, and M. Pollefeys, "Disambiguating visual relations using loop constraints", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1426–1433.

[268] R. Cai, J. Tung, Q. Wang, H. Averbuch-Elor, B. Hariharan, and N. Snavely, "Doppelgangers: Learning to disambiguate images of similar structures", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 34–44.

[269] Y. Xiangli, R. Cai, H. Chen, J. Byrne, and N. Snavely, "Doppelgangers++: Improved visual disambiguation with geometric 3d features", in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 166–27 175.

[270] R. Hartley, K. Aftab, and J. Trumpf, "L1 rotation averaging using the weiszfeld algorithm", in *CVPR 2011*, IEEE, 2011, pp. 3041–3048.

[271] H. Yang and L. Carlone, "One ring to rule them all: Certifiably robust geometric perception with outliers", *Advances in neural information processing systems*, vol. 33, pp. 18 846–18 859, 2020.

[272] H. Yang and L. Carlone, "Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization", *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 2816–2834, 2022.

[273] L. Wang and A. Singer, "Exact and stable recovery of rotations for robust synchronization", *Information and Inference: A Journal of the IMA*, vol. 2, no. 2, pp. 145–193, 2013.

[274] F. Arrigoni, L. Magri, B. Rossi, P. Fragneto, and A. Fusiello, "Robust absolute rotation estimation via low-rank and sparse matrix decomposition", in *2014 2nd International Conference on 3D Vision*, IEEE, vol. 1, 2014, pp. 491–498.

[275] F. Arrigoni, B. Rossi, P. Fragneto, and A. Fusiello, "Robust synchronization in so (3) and se (3) via low-rank and sparse matrix decomposition", *Computer Vision and Image Understanding*, vol. 174, pp. 95–113, 2018.

[276] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case", in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.

[277] A. Chatterjee and V. M. Govindu, "Robust relative rotation averaging", *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 958–972, 2017.

[278] D. Geman and S. Geman, "Bayesian image analysis", in *Disordered systems and biological organization*, Springer, 1986, pp. 301–319.

[279] E. Candes, J. Romberg, *et al.*, "L1-magic: Recovery of sparse signals via convex programming", vol. 4, no. 14, p. 16, 2005.

[280] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "Magsac++, a fast, reliable and accurate robust estimator", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1304–1312.

[281] P. J. Huber, "Robust estimation of a location parameter", in *Breakthroughs in statistics: Methodology and distribution*, Springer, 1992, pp. 492–518.

[282] C. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206–218, 1997.

[283] T. Boult and L. Gottesfeld Brown, "Factorization-based segmentation of motions", in *Proceedings of the IEEE Workshop on Visual Motion*, 1991, pp. 179–186.

[284] C. Gear, "Feature grouping in moving objects", in *Proceedings of 1994 IEEE Workshop on Motion of Non-rigid and Articulated Objects*, 1994, pp. 214–219.

[285] J. Costeira and T. Kanade, "A multi-body factorization method for motion analysis", in *Proceedings of IEEE International Conference on Computer Vision*, 1995, pp. 1071–1076.

[286] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms", in *2007 IEEE conference on computer vision and pattern recognition*, IEEE, 2007, pp. 1–8.

[287] C. W. Gear, "Multibody grouping from motion images", *International Journal of Computer Vision*, vol. 29, pp. 133–150, 1998.

[288] Y. Wu, Z. Zhang, T. S. Huang, and J. Y. Lin, "Multibody grouping via orthogonal subspace decomposition", in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, vol. 2, 2001, pp. II–II.

[289] K.-i. Kanatani, "Motion segmentation by subspace separation and model selection", in *Proceedings Eighth IEEE International Conference on computer Vision. ICCV 2001*, IEEE, vol. 2, 2001, pp. 586–591.

[290] K. Kanatani, "Evaluation and selection of models for motion segmentation", in *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III 7*, Springer, 2002, pp. 335–349.

[291] P. Ji, M. Salzmann, and H. Li, "Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data", in *Proceedings of the IEEE International Conference on computer Vision*, 2015, pp. 4687–4695.

[292] J. Park, H. Zha, and R. Kasturi, "Spectral clustering for robust motion segmentation", in *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV 8*, Springer, 2004, pp. 390–401.

[293] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[294] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, 2013.

[295] P. Ji, M. Salzmann, and H. Li, "Efficient dense subspace clustering", in *IEEE Winter conference on applications of computer vision*, IEEE, 2014, pp. 461–468.

[296] X. Xu, L. F. Cheong, and Z. Li, "Motion segmentation by exploiting complementary geometric models", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2859–2867.

[297] Y. Huang and J. Zelek, "Motion segmentation from a moving monocular camera", *arXiv preprint arXiv:2309.13772*, 2023.

[298] R. Vidal and R. Hartley, "Motion segmentation with missing data using powerfactorization and gpca", in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 2, 2004, pp. II–II.

[299] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)", *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[300] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using powerfactorization and gpca", *International Journal of Computer Vision*, vol. 79, pp. 85–105, 2008.

[301] L. Bai and J. Liang, "Sparse subspace clustering with entropy-norm", in *International conference on machine learning*, PMLR, 2020, pp. 561–568.

[302] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction", in *2011 International Conference on Computer Vision*, 2011, pp. 1615–1622.

[303] S. Zhang, C. You, R. Vidal, and C.-G. Li, "Learning a self-expressive network for subspace clustering", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 388–12 398.

[304] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1442–1456, 2010.

[305] D. Barath, D. Rozumnyi, I. Eichhardt, L. Hajder, and J. Matas, "Finding geometric models by clustering in the consensus space", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5414–5424.

[306] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 1926–1933.

[307] P. H. Torr, "Geometric motion segmentation and model selection", *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 356, no. 1740, pp. 1321–1340, 1998.

[308] M. Zuliani, C. S. Kenney, and B. Manjunath, "The multiransac algorithm and its application to detect planar homographies", in *IEEE International Conference on Image Processing 2005*, IEEE, vol. 3, 2005, pp. III–153.

[309] R. Toldo and A. Fusiello, "Robust multiple structures estimation with j-linkage", in *Computer Vision – ECCV 2008*, Springer Berlin Heidelberg, 2008, pp. 537–547.

[310] L. Magri and F. Andrea, "Robust multiple model fitting with preference analysis and low-rank approximation", in *Procedings of the British Machine Vision Conference 2015*, 2015, pp. 20–1.

[311] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?", *J. ACM*, vol. 58, no. 3, 2011, ISSN: 0004-5411.

[312] W. Zhao, S. Liu, H. Guo, W. Wang, and Y.-J. Liu, "Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild", in *European Conference on Computer Vision*, Springer, 2022, pp. 523–542.

[313] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[314] E. Fons, A. Sztrajman, Y. El-Laham, A. Iosifidis, and S. Vyetrenko, "Hypertime: Implicit neural representation for time series", *arXiv preprint arXiv:2208.05836*, 2022.

[315] S. Ramasinghe and S. Lucey, "Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps", in *European Conference on Computer Vision*, Springer, 2022, pp. 142–158.

[316] J. Zheng, S. Ramasinghe, X. Li, and S. Lucey, "Trading positional complexity vs deepness in coordinate networks", in *European Conference on Computer Vision*, Springer, 2022, pp. 144–160.

[317] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding", *arXiv preprint arXiv:1807.03748*, 2018.

[318] Y. Dai, J. Trumpf, H. Li, N. Barnes, and R. Hartley, "Rotation averaging with application to camera-rig calibration", in *Asian Conference on Computer Vision*, Springer, 2009, pp. 335–346.

[319] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, *et al.*, "Multimodal deep learning.", in *ICML*, vol. 11, 2011, pp. 689–696.

[320] Q. Wang, V. Ye, H. Gao, *et al.*, "Shape of motion: 4d reconstruction from a single video", in *International Conference on Computer Vision (ICCV)*, 2025.

[321] A. Sahoo, V. Tibrewal, and G. Gkioxari, "Aligning text, images, and 3d structure token-by-token", *arXiv preprint arXiv:2506.08002*, 2025.

[322] Y. Zhou, G. Gallego, X. Lu, S. Liu, and S. Shen, "Event-based motion segmentation with spatio-temporal graph cuts", *IEEE transactions on neural networks and learning systems*, vol. 34, no. 8, pp. 4868–4880, 2021.

[323] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection", in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[324] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera", in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 3769–3776.

[325] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning", *arXiv preprint arXiv:1610.02242*, 2016.

[326] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks", *Advances in neural information processing systems*, vol. 28, 2015.

[327] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", *Advances in neural information processing systems*, vol. 30, 2017.

[328] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning", *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[329] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6728–6736.