THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Learning and Optimizing Camera Pose

LUCAS BRYNTE

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2024

Learning and Optimizing Camera Pose

LUCAS BRYNTE
ISBN 978-91-7905-973-6

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

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5439
ISSN 0346-718X

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

Cover:
Generated by DreamStudio by stability.ai (Stable Diffusion v2.1) from the query “*Inference and iterative refinement of camera position and orientation*”.

Printed by Chalmers digitaltryck
Gothenburg, Sweden, January 2024

To my Father, whose mind and memory will be with me always.

Learning and Optimizing Camera Pose

LUCAS BRYNTE
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Plenty of computer vision applications involve assessing the position and orientation, i.e. the *pose*, of one or several cameras, including object pose estimation, visual localization, and structure-from-motion. Traditionally, such problems have often been addressed by detection, extraction, and matching of image keypoints, using handcrafted local image features such as the scale-invariant feature transform (SIFT), followed by robust fitting and / or optimization to determine the unknown camera pose(s). Learning-based models have the advantage that the they can learn from data what cues or patterns are relevant for the task, beyond the imagination of the engineer. However, compared with 2D vision tasks such as image classification and object detection, applying machine learning models to 3D vision tasks such as pose estimation has proven to be more challenging.

In this thesis, I explore pose estimation methods based on machine learning and optimization, from the aspects of quality, robustness, and efficiency. First, an efficient and powerful graph attention network model for learning structure-from-motion is presented, taking image point tracks as input. Generalization capabilities to novel scenes is then demonstrated, without costly fine-tuning of network parameters. Combined with bundle adjustment, accurate reconstructions are acquired, significantly faster than off-the-shelf incremental structure-from-motion pipelines. Second, techniques are presented for improving the equivariance properties of convolutional neural network models carrying out pose estimation, either by intentionally applying radial distortion to images to reduce perspective effects, or via a geometrically sound data augmentation scheme corresponding to camera motion. Next, the power and limitations of semidefinite relaxations of pose optimization problems are explored, notably leading to the conclusion that absolute camera pose estimation is not necessarily solvable using the considered semidefinite relaxations, since while they tend to almost always be tight in practice, counter-examples do indeed exist. Finally, a rendering-based object pose refinement method is presented, robust to partial occlusion due to its implicit nature, followed by a method for long-term visual localization, leveraging on a semantic segmentation model to increase the robustness by promoting semantic consistency of sampled point correspondences.

Keywords: Camera pose estimation, structure-from-motion, machine learning, optimization.

List of Publications

This thesis is based on the following publications:

[A] **Lucas Brynte**, José Pedro Iglesias, Carl Olsson, Fredrik Kahl , "Learning Structure-from-Motion with Graph Attention Networks". Submitted for Review, *arXiv:2308.15984*, 2023.

[B] **Lucas Brynte**[*], Georg Bökman[*], Axel Flinth, Fredrik Kahl , "Rigidity Preserving Image Transformations and Equivariance in Perspective". Scandinavian Conference on Image Analysis, 2023.

[C] **Lucas Brynte**, Viktor Larsson, José Pedro Iglesias, Carl Olsson, Fredrik Kahl , "On the Tightness of Semidefinite Relaxations for Rotation Estimation". Journal of Mathematical Imaging and Vision, 2022.

[D] **Lucas Brynte**, Fredrik Kahl , "Pose Proposal Critic: Robust Pose Refinement by Learning Reprojection Errors". British Machine Vision Conference, 2020.

[E] Carl Toft, Erik Stenborg, Lars Hammarstrand, **Lucas Brynte**, Marc Pollefeys, Torsten Sattler, Fredrik Kahl , "Semantic Match Consistency for Long-Term Visual Localization". European Conference on Computer Vision, 2018.

[*] Equal contribution.

Acknowledgments

First I would like to express my gratitude to my supervisor Fredrik Kahl. Thank you for believing in me from the start, for sharing your knowledge, and for being supportive throughout this journey towards my PhD.

To my co-supervisor Carl Olsson, thank you for introducing me to the beauty of computer vision and projective geometry back in Lund, and for your engagement in our research collaborations.

To my friends and colleagues in the computer vision group, past and present, I am especially grateful for the numereous and well-needed fika interruptions and peculiar discussions: José, Georg, Yara, Rasmus, Ji, Jennifer, Kunal, Axel, Victor, Roman, Sofie, Josef, David, Erik, Alex, Xixi, Dorian, Ida, Christopher, Måns, Carl, Huu, Eskil, Mikaela, Amir, James, and Torsten.

My deepest gratitude goes to my dear family, Mamma, Bodil and Jonathan, and to all my friends, for your constant patience and love. Looking forward, I wish to see much more of all of you. Finally, and from the bottom of my heart, thank you Fazeleh for standing by my side through thick and thin.

Lucas Brynte
Göteborg, January 2024

Acronyms

BA:	Bundle adjustment
CNN:	Convolutional neural network
GNN:	Graph Neural Network
NLP:	Natural language processing
P3P / PnP:	Perspective-3-point / perspective-n-point
POP:	Polynomial optimization problem
QCQP:	Quadratically constrained quadratic program
RANSAC:	Random sample consensus
SDP:	Semidefinite program
SDR:	Semidefinite relaxation
SfM:	Structure-from-motion
SIFT:	Scale-invariant feature transform

Contents

Part I

Overview

CHAPTER 1

Introduction

Key to human perception is vision, that is, the ability of observing and navigating the world around us through rays of light imaged on the retinas of our eyes. Much effort has been put into understanding the processes involved and attempting to replicate them artificially, enabling robots and machines to see as well. The sensory part of the problem, image capture, has to a great extent been resolved by the art of photography, pioneered by Joseph Nicéphore Niépce and his photograph *View from the Window at Le Gras* taken in 1826 (Figure 1.1). Succeeding the sensory problem comes the problem of perception, that is, the ability to understand the contents and meaning of an image by extracting higher-level information from its raw data, which in itself is nothing more than a spatial distribution of light intensity and color. The field of computer vision, sometimes referred to as 'inverse computer graphics', has since the 1960s led researchers to take on the challenges of visual perception, constructing models to infer properties such as semantics and 3D geometry from image data.

Later on came the deep learning era, revolutionizing many computer vision challenges by the introduction of learned models such as convolutional neural networks (CNNs) [2], [3] – most prominently semantic tasks such as image classification. However, learning 3D vision tasks such as camera pose estimation has proven more challenging, and is still actively researched. The pose of a camera can be characterized

Figure 1.1: *View from the Window at Le Gras* (1826), the earliest known photograph to survive until this day. Image source: [1].

as its position and orientation, and estimating it involves geometrical reasoning, which is not always easy to learn with a black box model. One way to handle this is to divide the task into separate modules carrying out sub-tasks of lower complexity, some or all of which may be learned. Examples of such sub-problems are detection, extraction, or matching of sparse or dense local image features. Another example is object detection.

For many years in the last decade, CNN model architectures dominated the field, and were applied almost universally on a plethora of vision tasks. More recently, attention-based architectures such as the Transformer [4] have been widely proposed, beyond their initial success in natural language processing (NLP), and with immpressive results. In fact, computer vision is one of the major applications other than NLP, and attention-based architectures such as the Vision Transformer (ViT) [5], Swin Transformer [6], and Masked Autoencoder (MAE) [7] have shown great promise. In

NLP, transformer-based large language models such as ChatGPT [8] have dramatically shifted the common expectation of what machine learning at scale can achieve, among the public and academia alike. Another recent deep learning research trend – denoising diffusion models – has also shown remarkable results in image generation, not the least when combined with language models to perform tasks such as text-to-image generation [9], [10], which is also how the cover page illustration of this thesis was conceived. In fact, aside from their impressive performance, one of the main advantages of transformer-like architectures is their flexibility. Unlike CNNs, which are inherently associated with feature processing on grids, transformers can operate more seamlessly on data from diverse domains, by means of cross-attention [4]. Nevertheless, the translation equivariant CNN model provides a powerful inductive bias in many contexts, and can be more efficient than transformers, both in terms of memory, computation, and capability of generalization from small amounts of training data. CNNs are often used as a low-level feature extractor for the tokenization of vision transformers.

From a more traditional viewpoint, 3D vision tasks such as various forms of camera pose estimation can often be posed as an optimization problem. These problems are however not convex, one reason being that the constraint set of pose optimization involves restriction to the rotation manifold. While iterative optimization methods can converge to local optima, there is therefore no guarantee that the solution is globally optimal. Convex relaxations can be employed to solve an easier convex optimization problem, called a *relaxation*, which for very special problem settings (e.g. point cloud registration with unit quaternion parameterization) or when the noise level is limited, come with global optimality guarantees. Sometimes, it is feasible to characterize every local optimum, by solving systems of polynomial equations corresponding to the first-order optimality conditions, leading to multiple candidates for the global optimum, all of which one can compare to determine the best one(s), see e.g. [11]–[13].

In this thesis, I focus on camera pose estimation and explore pose estimation methods based on machine learning as well as optimization, with a primary focus on the former. The methods are explored in three different aspects:

1. Quality
2. Robustness
3. Efficiency

1.1 Thesis Outline

The thesis consists of two parts. In Part II, five research papers, which constitute the thesis contributions, are included in reverse chronological order. Part I consists of five chapters, where this introduction is followed by Chapter 2, providing some fundamental background on deep learning and geometry. Chapter 3 proceeds to present the camera pose estimation problems considered in the contributions, while discussing various tools and techniques relevant for solving them. Chapter 4 then presents the thesis contributions, by giving a summary of each of the included articles, before proceeding to the conclusions in Chapter 5.

1.2 Notation

This section describes the mathematical notation used in Part I of the thesis. Plain letters, e.g. x, y, z, λ are denote scalar variables, while boldface letters, e.g. $\mathbf{x}, \mathbf{v}$, are used to denote vectors, or, when capitalized, matrices, e.g. $\mathbf{P}, \mathbf{X}, \mathbf{R}$. Vectors of homogeneous coordinates are denoted with bars, e.g. $\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}}$. In context of matrix-vector multiplications, vectors are in general regarded as single-column matrices. Vector or matrix subscripts are used to extract individual elements, while matrix superscripts, e.g. $\mathbf{P}^3$ or $\mathbf{P}^{1:2}$ are used to extract a row or range of rows, stacked column-wise.

CHAPTER 2

Background

In this chapter, I present some fundamental background in machine learning and geometry of relevance to the thesis, including neural network architectures, their equivariance properties, representations of three-dimensional rotations, and projective geometry preliminaries.

2.1 Deep Learning

A recurring theme of this thesis is *machine learning*, which regards the development of algorithms to perform tasks without explicit instructions. Unlike unsupervised learning techniques such as clustering algorithms and autoencoders, in this thesis, supervised learning is utilized to learn to estimate camera pose in various ways. In supervised learning, a parameterized model is trained, i.e. guided or tuned, to perform a task based on its performance on an annotated set of training images. In computer vision, examples of such tasks can be image classification, semantic segmentation, object detection, optical flow estimation, or even 3D inference tasks such as object pose estimation, visual localization and structure-from-motion.

In order to train the model, a loss function is defined and minimized w.r.t. the parameters. The loss function may for instance be cross-entropy for classification tasks, or squared errors for regression tasks. If $f_\theta(x)$ is the output of a parameterized model f_θ applied on an input $\mathbf{x}$, let $l(f_\theta(\mathbf{x}), \mathbf{y})$ be the loss function comparing the predicted output with the ground-truth value $\mathbf{y}$. Training the model amounts to minimizing the average loss on an entire training set of input samples $\mathbf{x}_i,\ i = 1, \dots, N$ and corresponding ground-truth outputs a.k.a. target values $\mathbf{y}_i$. Regarded as an optimization problem, this minimization is in general non-trivial, but often works well if using suitable model architectures, e.g. artificial neural networks, and appropriate optimization methods such as stochastic gradient descent (SGD) and its many derivatives. SGD-like optimizers are first order methods which, instead of calculating the full gradient of the entire loss function, takes only one or a few training samples into account at each iteration, replacing the actual gradient with the gradient of the corresponding truncated loss. In addition to accelerating the optimization by speeding up each iteration, stochastic optimization methods are less prone to getting stuck at suboptimal local minima [14].

Unfortunately, even if the loss function has been minimized on the training set, there are still no guarantees that the model acquired will generalize well to novel examples, which, of course, is the ultimate goal of the entire endeavor. The problem is known as overfitting, and for this reason a separate set of annotated samples should always be left out from training and dedicated solely to validating the model's ability to generalize. If one faces too much overfitting, one may need to acquire more training data, apply data augmentation, use models with suitable inductive biases, or apply regularization techniques.

In the rest of this section, selected model architectures of particular importance for the thesis contributions will be presented. They all belong to the deep learning paradigm, that is, they all belong to the broad class of machine learning model architectures known as deep neural networks, i.e. multi-layer artificial neural networks. Artificial neural networks were pioneered many years ago, notably by the biologically inspired Perceptron [15] in the 1960s, but not until the last 10 to 15 years has their remarkable potential been made undisputably evident by the deep learning revolution. The general idea of deep neural networks is to stack so-called 'linear' layers (more precisely, learned affine transformations) and to interleave them with non-linear so-called activation functions, inspired by the synapses of our brains, typically applied in an element-wise fashion. Each layer then consists of a number of features, which we call 'activations'. When a deep architecture is fitted to data, the typical behavior is that early layers capture simple features, while deeper layers allow for increasing levels of abstraction [16]. The neural network architecture holds universal function approximation properties [17], [18].

Several factors have led up to the deep learning revolution, including the rise of general-purpose computing on graphics processing units (GPGPU) and the increasing availability of large amounts of annotated data. Architectural innovations have also been important facilitators for training deep architectures, in particular improved gradient flow via activation functions like the Rectified Linear Unit (ReLU) [19]–[21], and furthermore by residual connections [22] later on.

Convolutional Neural Networks

One of the biggest success stories for machine learning in computer vision has been the convolutional neural network (CNN) [2], [3], [16]. Explained in simple terms, if a regular so-called 'vanilla' feedforward neural network works on *feature vectors* and consists of stacked linear layers (interleaved with non-linear activation functions), a CNN works on *feature maps* and replaces the linear layers with convolutional layers. A feature map is simply a rasterized image, i.e. a grid, typically with multiple channels, and every layer in a CNN holds its activations in such a feature map. Similar to how a linear layer propagates information from one layer to the next by multiplying an input feature vector with a learned weight matrix, a convolutional layer carries out convolution between an input feature map and learned spatial filters[a], often of very limited extent for sake of efficiency. Unlike vanilla neural networks, for CNNs it is the feature maps, not the parameters, that dominate the memory consumption. Thus, in order to limit the memory as well as computational demand, it is common practice to gradually downsample the feature maps.

Group Equivariance

Except for boundary effects, the convolutional layers carried out in CNNs hold the property of translation equivariance. That is, if f is a convolutional layer applied on a feature map $\mathbf{X}$ (where boundary effects are disregarded by letting $\mathbf{X}$ be infinitely defined), then $f(T_{\mathbf{t}}\mathbf{X}) = T_{\mathbf{t}}f(\mathbf{X})$, where the operator $T_{\mathbf{t}}$ carries out translation by $\mathbf{t}$. In other words, convolution commutes with translation, which is a well-known result. What this means for us in practice is that CNN models generalize what they have learnt – beyond the training images themselves also to shifted versions. In

[a] Each channel of the output feature map is the result of convolution with a unique corresponding filter. Furthermore, each of these filters is vector-valued, and, rather than multiplications, carries out scalar products across the input feature map channels before summing up the filter response.

contrast, direct application of a vanilla neural network on an input image requires flattening all pixels to a vector representation, essentially disregarding the spatial structure, typically leading to high sensitivity and poor generalization [16]. If there are symmetries in the data that can be described reasonably well by translations, a CNN will therefore provide a more efficient learner.

While a translation can be intuitively understood as uniformly shifting spatially distributed data, the set of all translations also constitutes a group, and with this point of view the translation equivariance is a particular instance of group equivariance, meaning that the equivariant function commutes with so-called *group action* for any element of the group in question. Without going into too much detail (for more information see e.g. [23]), just as group elements can be multiplied with each other, one may also define the *action* of a group element (e.g. a particular translation) on the domain on which the data is defined, e.g. a grid. The domain is often considered as a vector space or a subset of a vector space, and the group action is represented by a matrix-vector multiplication with invertible matrices, known as the *representation* of the group. Thus, equivariance can be defined for many different groups and domains, e.g. any matrix group acting on a vector space. For instance, by imposing certain restrictions on the matrix structure, convolution as well as translation can be regarded as matrix-vector multiplication (albeit, strictly speaking on a vector space of infinite dimension to avoid boundary effects).

While still a growing research field, there are already concrete examples of group convolutional neural network models beyond standard CNNs for more exotic groups, for instance roto-translations with discrete rotations [24] or even continuous rigid planar motion ($\mathrm{SE}(2)$) using steerable filters [25]. Another group worth mentioning is the symmetric group $\mathcal{S}_n$, as equivariance w.r.t. this group, i.e. permutation equivariance, is a very beneficial property for machine learning models that process unordered sets of points, see e.g. Deep Sets [26].

The Attention Mechanism and Transformers

Another family of permutation equivariant neural network models are *attention* architectures, the most famous example being the Transformer [4]. The attention mechanism is a technique used to define neural network layers which dynamically ‘attend’ to varying degree to different so-called ‘tokens’, which can be thought of as symbols or points, or in particular architectures as nodes in a graph neural network (see the following section). In recent years, not only have attention models such as [5]–[7] challenged the conventional CNN backbone models for image classification and other computer vision tasks, attention models are also the fundamental architectural building block of the large language models of today, which have demonstrated performance beyond anticipation (e.g. [8]).

In an attention layer, information is propagated by the same logic from each input token using shared network parameters, which is what provides the permutation equivariance. While this may sometimes be a very desirable property, for many scenarios it is actually intentionally circumvented by augmenting the features of the tokens with so-called positional encodings. This is common practice for language models as well as vision transformers, since in both cases there are natural spatial relationships between the tokens, not to be ignored. Attention layers deviate from the otherwise common practice of using linear layers interleaved with point-wise non-linear activation functions, as they are neither linear nor merely point-wise non-linear. I believe that this contributes to their power.

Graph Neural Networks

Finally, another family of deep learning architectures of relevance to this thesis are the Graph Neural Networks (GNNs). This term is relatively broad, and thus there are many flavors of GNN model architectures in the literature, see e.g. [23], [27] for examples.

A relatively general formalization of graph neural networks is the Message-Passing Neural Network (MPNN) [28], [29], which is built upon the following feature aggregation layer:

$$\mathbf{h}_u^{t+1} = \phi\left(\mathbf{h}_u^t, \bigoplus_{v\in\mathcal{N}_u} \psi(\mathbf{h}_u^t, \mathbf{h}_v^t)\right), \tag{2.1}$$

where $\bigoplus$ is a permutation-invariant function (e.g. summation), $\mathbf{h}_u$ are the target node features, and $\mathbf{h}_v$ are the features of the source nodes $v \in \mathcal{N}_u$, neighboring u. The functions ϕ and ψ are learned in general, commonly using linear layers or shallow feedforward neural networks. Furthermore, edge features $\mathbf{e}_{uv}$ may also be passed as an additional argument to ψ, and possibly the edge features themselves may be updated similarly to the node features [28]. A message-passing layer is permutation-invariant in the sense that information is aggregated in an identical manner from each of the neighbors.

Another type of powerful GNN layer also incorporates the attention mechanism discussed in the previous section, and can be defined as follows:

$$\mathbf{h}_u^{t+1} = \phi\left(\mathbf{h}_u^t, \bigoplus_{v\in\mathcal{N}_u} \alpha_{uv}\psi(\mathbf{h}_v^t)\right), \tag{2.2}$$

where

$$\alpha_{uv} = \frac{\exp(a(\mathbf{h}_u^t, \mathbf{h}_v^t))}{\sum\limits_{w\in\mathcal{N}_u} \exp(a(\mathbf{h}_u^t, \mathbf{h}_w^t))} \tag{2.3}$$

are softmax-normalized attention weights determined by a learned function $a(\mathbf{h}_u^t, \mathbf{h}_v^t)$.

2.2 Three-Dimensional Rotations

In order to represent camera orientation, we define it as the relative *rotation* between a camera reference frame and a global reference frame. There are, however, a few different ways to represent a 3D rotation. A classic representation is with so-called *Euler angles*, i.e. a sequence of three angles corresponding to a set of three chained rotations about some predetermined axes. Alternatively, a single axis is enough if the axis is not fixed, leading to the so-called *axis-angle* representation, which represents any rotation as an angular displacement α about an axis of revolution $\mathbf{v}, |\mathbf{v}| = 1$. Consequently, the scalar-vector multiplication $\alpha\mathbf{v}$ provides a compact 3-parameter representation of the rotation. Furthermore, the axis of revolution is uniquely determined, save for singularities.

Another representation is 3×3 *rotation matrices*, which explicitly define the (positively oriented) orthonormal basis corresponding to the change of reference. While this representation has its benefits, the 9 matrix elements constitute a redundant overparameterization of a rotation, which has only 3 degrees of freedom, thus requiring additional constraints on the matrix elements. A rotation matrix is any matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ which satisfies orthonormality ($\mathbf{R}^T\mathbf{R} = I$) and has positive orientation ($\det \mathbf{R} = 1$). The set of all such matrices, $\mathrm{SO}(3)$, is a three dimensional manifold in the space of $\mathbb{R}^{3\times 3}$. Furthermore, any rotation matrix R is the matrix exponential $\exp A$ for a skew-symmetric matrix

$$A = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix},$$

where $\mathbf{a} = (a_x, a_y, a_z)$ is exactly the axis-angle representation $\alpha\mathbf{v}$ of R. This matrix exponential-logarithm relationship associates elements $R \in \mathrm{SO}(3)$ with elements

$A \in \mathfrak{so}(3)$, where $\mathfrak{so}(3)$ is the so-called Lie algebra of the Lie group $\mathrm{SO}(3)$, the rotation manifold, and also defines its tangent space at the identity element $R = I$.

Finally, another established representation of 3D rotations is by unit-length quaternions. Quaternions are a generalization of complex numbers with one real and three imaginary dimensions, usually referred to as the scalar and the vector part of the quaternion. Considering all 4 dimensions, we may think of quaternions as elements of $\mathbb{R}^4$, and of unit quaternions as elements of the 3D hyper-sphere $\mathcal{S}^3$. Aside from a sign-ambiguity, every rotation can be associated with a unit quaternion by defining the scalar part as $\cos\frac{\alpha}{2}$ and the vector part as $\mathbf{v}\sin\frac{\alpha}{2}$, where $\alpha, \mathbf{v}$ is the axis-angle representation mentioned previously. Furthermore, just as for complex numbers, multiplication is well-defined for quaternions, and the result of multiplying unit quaternions corresponds exactly to chaining rotations.

2.3 Projective Geometry

Projective geometry is a beautifully simple extension of Eucliden geometry which unlocks a surprisingly rich toolbox for theoretical reasoning and computation regarding projective relations. More often than not, however, at first sight the unfamiliar constructions and representations used in projective geometry may seem arbitrary or meaningless, but rest assured, they merely take some leap of faith to get used to. In this section, I will give a brief introduction to projective geometry, focusing on a few core elements of particular interest for the rest of the thesis. For a more complete introduction, see for instance [30].

Point Representation in Projective Space

A (real[b]) projective space $\mathbb{P}^n$ is in many ways similar to the familiar Euclidean space $\mathbb{R}^n$, but differs in two major regards worth emphasizing:

Points at Infinity The projective space is slightly "larger": $\mathbb{R}^n \subsetneq \mathbb{P}^n$. This is due to that in addition to all of the finite (yet infinitely many) Euclidean points, we also consider so-called *ideal points*, a.k.a. points at infinity or simply infinity points. The ideal points are all infinitely far away from the origin, but are distinguished from

[b] I only consider projective spaces over the real numbers in this thesis, and use the terms 'projective space' and 'real projective space' interchangeably.

one another depending on in what direction they are infinitely far away[c]. A famous consequence of this extension is that in the projective plane $\mathbb{P}^2$, unlike $\mathbb{R}^2$, every pair of distinct lines have a unique intersection, even in the case of parallel lines, in which case the intersection is at an ideal point.

Homogeneous Coordinates To be able to represent ideal points as well, every point in projective space is represented by so-called homogeneous coordinates, meaning that 1) they have one more coordinate than the cartesian coordinates we are normally used to and 2) the scale of the representation is arbitrary, in the sense that rescaling a homogeneous vector does not alter which point is represented by it.

In concrete terms, a 2D point $(x, y) \in \mathbb{R}^2$ is in homogeneous coordinates represented by any vector $(\lambda x, \lambda y, \lambda) \in \mathbb{P}^2$, where $\lambda > 0$ is an arbitrary scaling factor. For noting that two homogeneous coordinate vectors $\bar{\mathbf{x}}_1 = (\lambda_1 x, \lambda_1 y, \lambda_1)$ and and $\bar{\mathbf{x}}_2 = (\lambda_2 x, \lambda_2 y, \lambda_2)$ are rescaled versions of one another, and thus represent the very same projective point, we use the notation $\bar{\mathbf{x}}_1 = \bar{\mathbf{x}}_2$. Ideal points are those with homogeneous coordinate representation $(\lambda x, \lambda y, 0)$, i.e. those with final coordinate 0. For a finite point with homogeneous coordinates (a, b, c), $c > 0$, note that conventional cartesian coordinates can easily be extracted by what is known as "perspective division", which simply amounts to normalizing the final coordinate by division of itself[d]: $(a, b, c) \sim (a/c,\ b/c,\ 1)$. We can then identify the point as $(a/c,\ b/c)$ in cartesian coordinates. Analogously, the homogeneous coordinate representation of a finite 3D point $(x, y, z) \in \mathbb{R}^3$ is given by $(\lambda x, \lambda y, \lambda z, \lambda) \in \mathbb{P}^3$, in addition to which there are also ideal points at infinity. Finally, note that a homogeneous coordinate vector is a valid representation of a projective point if and only if not all coordinates are 0.

Point Transformations

Next, we will see what happens when a homogeneous coordinate vector undergoes a matrix multiplication. As it turns out, a whole hierarchy of relevant transformations can be represented in this way, ranging from rotations, translations, and scalings to linear / affine transformations as well as what is known as projective transformations.

Starting with the general case, a *projective transformation* in n dimensions (a.k.a. *projectivity* or *homography*) can be identified (up to scale) with an invertible $(n + 1) \times (n + 1)$ matrix $\mathbf{H}$. Applying the transformation on a point $\bar{\mathbf{x}} \in \mathbb{P}^n$ is as simple

[c] Opposing points are, however, collapsed to and identified as a single ideal point.

[d] There are strong connections between perspective division and perspective projection.

as a matrix multiplication with its homogeneous coordinate vector: $y \sim H\bar{\mathbf{x}}$. Note that $\mathbf{H}$ is a homogeneous representation: $\mathbf{H} \sim \lambda, \quad \forall \lambda \neq 0$. While seemingly linear, the implied perspective division needed to extract cartesian coordinates can result in dramatic perspective effects and heavy distortions. Despite such distorsions, a projective transformation is characterized by always mapping a line to a line, making it a so-called *collineation*. Put in other words, lines are *preserved* by the mapping[e]. Furthermore, for any real projective space of dimension $n \geq 2$, there are in fact no other collineations than the homographies.

As we will see next, by imposing various further restrictions on $\mathbf{H}$ (beyond being invertible), a whole hierarchy of transformations is acquired. The homogeneous matrix representation is indeed very useful for expressing many simple and more familiar transformations such as translations, which could not be expressed by matrix multiplication on cartesian coordinates.

Affine transformations of $\mathbb{P}^n$ can be represented by matrices with the structure

$$\mathbf{H} \sim \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}_{1\times n} & 1 \end{bmatrix}, \quad \mathbf{A} \in \mathbb{R}^{n\times n}, \quad \mathbf{t} \in \mathbb{R}^n, \tag{2.4}$$

and are characterized by preserving parallelity between lines, planes, and subspaces in general. Another case of interest is when $\mathbf{A}$ is a scaled rotation;

$$\mathbf{A} = s\mathbf{R}, \quad s \neq 0, \quad \mathbf{R}^T\mathbf{R} = I, \quad \det \mathbf{R} = 1, \tag{2.5}$$

in which case $\mathbf{H}$ is a similarity transformation, characterized by the preservation of angles. If furthermore $|s| = 1$, $\mathbf{H}$ is a Euclidean transformation, which additionally preserves scale, and if $s = 1$, $\mathbf{H}$ is also orientation-preserving, and known as a (proper) rigid transformation. Each of these types of transformations also corresponds to a group – the projective linear group, the affine group, the isometry group, the Euclidean group, and the special Euclidean group.

Modeling Camera Projection

So far, we have only considered mappings from a projective space to itself, but a central concept in computer vision is naturally that of camera projection. The most widely used camera model is the *projective camera* [30], which can be defined as a mapping from $\mathbb{P}^3 \rightarrow \mathbb{P}^2$, represented by a full rank 3×4 camera matrix $\mathbf{P}$, a homogeneous entity just like the homogeneous representations of points and homographies covered

[e] In fact, not only lines, but any projective subspace is preserved.

already. The camera matrix can be applied directly on a homogeneous 3D point $\bar{\mathbf{X}} \in \mathbb{P}^3$, and by simple matrix-vector multiplication, a projection $\bar{\mathbf{x}} \sim \mathbf{P}\bar{\mathbf{X}} \in \mathbb{P}^2$ is acquired. As an example, the cartesian coordinates $\mathbf{x}$ may be the very pixel coordinates of a digital photography.

A projective camera carries out *central projection*, a.k.a. *rectilinear* or *perspective* projection[f], meaning that a 3D point $\bar{\mathbf{X}}$ is projected onto an image plane along the line connecting $\bar{\mathbf{X}}$ and a special point $\bar{\mathbf{C}}$, called the center of projection, or simply the *camera center*. A key property of central projection is that lines in 3D space are also projected onto lines in the image[g]. The normal vector of the image plane, in the direction not facing the camera center[h], is known as the *principal axis*[i] of the camera. The distance between the camera center and the image plane determines its *focal length*. The simplest projective camera is the canonical camera $\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$, which is centered at the origin and takes the z-axis as its principal axis.

In general, the camera center of a camera $\mathbf{P}$ can be determined in homogeneous coordinates according to $\bar{\mathbf{C}} \sim \mathbf{P}$, i.e. the (one-dimensional) null space of $\mathbf{P}$. A projective camera $\mathbf{P} = \begin{bmatrix}\mathbf{A} & \mathbf{t}\end{bmatrix}$ is finite if and only if $\bar{\mathbf{C}}$ is not an ideal point, which is the case exactly when $\mathbf{A}$ is invertible, and the camera center can be conveniently determined in cartesian coordinates by $\mathbf{C} = -\mathbf{A}^{-1}\mathbf{t}$.

There is much to be said regarding different camera models, and I will not discuss them all, but focus on what is more relevant for this thesis. In particular, I will from now on exclude cameras at infinity. In practice, it is often more useful to factorize the camera matrix $\mathbf{P}$ into ‘extrinsic’ camera parameters, defining the physical position and orientation, and another set of ‘intrinsic’ camera parameters. The factorization reads

$$\mathbf{P} \sim \mathbf{K}\tilde{\mathbf{P}}, \quad \mathbf{K} = \begin{bmatrix}\alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1\end{bmatrix}, \quad \tilde{\mathbf{P}} = \begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}. \tag{2.6}$$

The extrinsic parameters consist of a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, bundled together in a so-called *calibrated* (a.k.a. normalized) camera matrix $\tilde{\mathbf{P}}$, and define a proper rigid transformation from global coordinates to the camera coordinate system. That is, given a finite 3D point $\mathbf{X} = (X, Y, Z)^T$ expressed in a homogeneous coordinate vector $\bar{\mathbf{X}} = (X, Y, Z, 1)^T$ (normalized with final coordinate 1), the simple multiplication $\tilde{\mathbf{X}} = \tilde{\mathbf{P}}\bar{\mathbf{X}}$ provides the cartesian coordinates

[f] Central projection is also strongly related to the gnomonic map projection, its restriction on the sphere.

[g] With the exception of lines that intersect the center of projection, which instead collapse into points.

[h] For the particular case of cameras at infinity, the sign of this direction is undefined.

[i] Also known as principal ray or optical axis.

$\tilde{\mathbf{X}} = (\tilde{X}, \tilde{Y}, \tilde{Z})^T$ of the very same 3D point expressed in the camera coordinate system. Now, to carry out central projection on the image plane $\tilde{Z} = 1$, we may simply interpret these coordinates as homogeneous coordinates of a 2D point $\bar{\bar{\mathbf{x}}} \sim (\tilde{X}, \tilde{Y}, \tilde{Z})^T \sim (\tilde{X}/\tilde{Z},\ \tilde{Y}/\tilde{Z},\ 1)^T$, i.e. the perspective division carries out the central projection. We refer to the extrinsic parameters $\tilde{\mathbf{P}}$ as the *pose* of the camera – the center of attention for this entire thesis.

The intrinsic parameters consist of an upper triangular matrix $\mathbf{K}$, a.k.a. the calibration matrix, and allows us to change the frame of reference within the image plane, e.g. using pixel coordinates rather than metric units. It is almost as general as an affine transformation, but ensures the direction of the x-axis to be preserved. It is also typically the case that $\alpha_x, \alpha_y > 0$, and so no reflection occurs. Other common restrictions on the intrinsics are that the skew parameter $s = 0$ and that there is a fixed focal length $f = \alpha_x = \alpha_y$[j]. The principal point (x_0, y_0) is the projection of the optical axis in pixel coordinates, and is usually roughly centered in the image.

Note how the relationship $\bar{\mathbf{x}} \sim \mathbf{P}\bar{\mathbf{X}}$ of the central projection camera model, also known as the *camera equation*, is elegantly linear in homogeneous coordinates (although there is a hidden non-linearity in the perspective division). It should however be noted that this linear model has its limitations in that it does not account for lens distortion. In reality, camera lenses are at best approximately rectilinear[k], and so-called fish-eye lenses are even by design not rectilinear, for the benefit of a wider field of view. The most important lens distortion models are radial distortion models. These are applied on the calibrated image points $\bar{\bar{\mathbf{x}}} \sim \mathbf{K}^{-1}\bar{\mathbf{x}} \sim \tilde{\mathbf{P}}\bar{\mathbf{X}}$ and amount to a radial rescaling, meaning they can be expressed as (non-linear) functions mapping distorted radial distances to undistorted distances[l].

[j] If $\alpha_x \neq \alpha_y$ the focal length is ambiguous, but if $f = \alpha_x = \alpha_y$, one can think of f as the distance between the camera center and the image plane. Equally well, and more generally, α_x and α_y can be thought of as scaling factors from metric to pixel units on the canonical image plane at depth 1.

[k] A rectilinear lens is a lens that makes straight lines in 3D space project onto straight lines in the image.

[l] Radial distance refers to the distance of an image point from the origin (the normalized principal point).

CHAPTER 3

Pose Estimation

The purpose of this chapter is to present the pose estimation problems considered in this thesis, while discussing various tools and techniques commonly used to solve them, providing relevant context to more novel approaches like the learning and optimization methods proposed or investigated in the contributed research articles. The pose estimation problems I have considered can be broadly categorized as 1) absolute camera pose estimation and 2) structure-from-motion, and I will present the problems in that order. The methods and techniques commonly used to solve these problems are to a large extent overlapping, and many of these will therefore first be presented in the context of absolute pose estimation, and then referred back to when discussing structure-from-motion.

Before proceeding, I would like to clarify that within the scope of this thesis and all included articles, the term ‘pose’ explicitly refers to rigid pose, meaning a position and an orientation in 3D space. Typically, we are interested in the pose of one or many cameras, described precisely by their extrinsic camera parameters. In other contexts, the term ‘pose’ may have a slightly different meaning. For instance, in the computer vision tasks of hand pose or human pose estimation, ‘pose’ refers to the configuration of joints and limbs, and in the context of capsule neural networks it has an abstract meaning as a set of properties of a vector-valued neuron [31].

The chapter begins to present the absolute camera pose estimation problem and its applications, followed by an introduction to correspondence-based pose estimation and robust fitting using random sample consensus (RANSAC). Next, the relative camera pose estimation and structure-from-motion (SfM) problems are presented, before concluding with one section about pose optimization and another one about pose regression.

3.1 Absolute Camera Pose Estimation

One of the fundamental problems in computer vision is that of *absolute camera pose estimation*, a.k.a. calibrated *camera resectioning*, meaning that given a photograph, the pose of a camera, i.e. its location and orientation in 3D space, is to be determined relative to a 3D reference model. This is in contrast to *relative camera pose estimation*, where the problem is instead to estimate relative poses between cameras without any reference model. In the most typical formulation of absolute pose estimation, a reference coordinate system is assigned to the 3D model, and one is establishing a set of point correspondences between 3D keypoints residing on the model, and their respective projections observed in the camera view, i.e. 2D image points[a]. In many cases, the intrinsic camera parameters are known, and the image points can be converted ('normalized') from pixel coordinates to meaningful physical units, such that each normalized image point can be interpreted as a viewing ray with a precise direction in 3D space. With as few as three of these 2D-3D point correspondences, one can eliminate all camera poses but a finite set of at most 4 solutions which can explain the projective relationship. If the camera intrinsics are not known, they can be solved for as well in addition to the camera pose, in which case the problem is known as (uncalibrated) camera resectioning, which requires at least 5.5 point correspondences. Even if there are enough measurements for uncalibrated camera resectioning, it is always preferable to use any existing knowledge of the camera intrinsics if the measurements are subject to noise.

An essential aspect of absolute pose estimation is that apart from observing an image projection, one has additional knowledge of a 3D reference model of some sort. The representation for encoding this knowledge, however, need not necessarily be in the particular form of 2D-3D point correspondences, but other representations, explicit or implicit, may be utilized, e.g. a set of reference images with annotated

[a] For relative camera pose estimation, one would instead establish 2D-2D correspondences, between image points in one view and another.

poses. Finally, note that there are also extensions of the problem to multiple cameras, e.g. a stereo camera system, calibrated such that the relative camera poses are known a priori.

There are two applications considered in this thesis which fall into the absolute pose estimation category: 1) Visual localization and 2) Object pose estimation. Each of these will be elaborated on in the following, after which a few approaches relevant to absolute pose estimation are presented. Another important application of absolute pose estimation, which can be seen as an instance of object pose estimation, is to calibrate the extrinsic parameters of a camera relative to a scene, via partial knowledge of the 3D geometry of the scene, or by placing a calibration object with known geometry into the scene.

Application: Visual Localization

In visual localization, the problem is to determine the position of a camera relative to a reference map, given one or many photographs. A typical use case is navigation of an autonomous robot or vehicle on which a camera has been mounted. There are many variations to the problem, possibly involving tracking over time or the presence of other sensors than the camera itself, e.g. global navigation satellite system (GNSS) transceivers, inertial measurement units (IMUs), light detection and ranging (LIDAR) units, or odometers, but a relatively common problem formulation is to determine the camera pose (position and orientation) given a single image and a reference scene representation.

One common representation of the scene is a sparse 3D point cloud, where each point is stored along with an associated image feature descriptor (e.g. SIFT features [32]). Another common representation of the reference model is slightly more implicit: A collection of reference images with known associated camera poses, possibly together with a 3D point cloud as well as annotations of which 3D points are visible in which images. Note that the image feature descriptors in the first representation are conceptually straight-forward to extract also in this case. Common approaches to determine both reference poses and a 3D reference model together are via SfM (see Section 3.3), or, in indoor environments, via RGB-D cameras based on structured light together with corresponding 3D mapping algorithms. The visual localization problem is central in Paper E, and also considered in Paper B.

Application: Object Pose Estimation

Another instance of absolute pose estimation is the task of estimating the pose of a camera relative to an object observed in its image, known as (rigid) object pose estimation, or often simply as 6D pose estimation. Conceptually, this problem is the same as visual localization, and differs mostly in the characteristics of the 3D reference model: Instead of a large reference map out of which typically only a small part is visible from a particular view, the reference model is now an object of limited spatial extent, often entirely within the field-of-view. This does not imply, however, that estimating pose is less challenging in this scenario than for localization. In particular when the object is small, it gets more challenging to maintain high precision, especially regarding the relative position estimate in the depth direction.

A well-known use-case of object pose estimation is for industrial robots to precisely detect the position and orientation of an object such as an assembly component, allowing it to be grasped and manipulated, without the need to spatially organize items in grids or similar structures. The problem is considered in Papers B and D.

3.2 Correspondence-Based Pose Estimation

As touched upon already, a common approach to absolute pose estimation is to establish a set of 2D-3D point correspondences, and search for a pose which makes the 3D keypoints project as close as possible to the corresponding measured 2D points. This all works very well, assuming one can indeed acquire reliable point correspondences in the first place. More commonly than not, however, a robust pose estimator capable of rejecting outlier correspondences is needed. In this section, I will briefly discuss the main components of a typical robust correspondence-based pose estimator, with emphasis on absolute camera pose estimation.

Feature Extraction and Matching

For absolute pose estimation, we want to establish 2D-3D point correspondences, but let's first briefly consider the case of relative pose estimation, where instead 2D-2D point correspondences are to be established. This is done by matching local image features, which traditionally has been done sequentially, through the steps of 1) keypoint detection, 2) feature description, and 3) feature matching. Much research efforts have been put into addressing the matter, but perhaps it is especially worth mentioning the handcrafted SIFT features [32], proposed by David Lowe over two decades ago, used in countless research works and practical applications ever since. The SIFT descriptor is based on histograms over directions of local image intensity gradients, and is designed for rotation- and scale-invariance. While this type of handcrafted feature descriptor is quite powerful, it can suffer from stability issues e.g. under heavy changes in lighting. Due to its limitations, the approach based on handcrafted features has in recent years been challenged by deep learning methods, by *learned feature descriptors* [33]–[36], often carrying out detection and description in parallel, in either a joint or decoupled manner, as well as *learned matchers* [35], [37], often using attention-based models and / or graph neural networks (see Section 2.1).

When it comes to absolute pose estimation and establishing 2D-3D point correspondences, the matching problem is, however, no longer symmetric. It is also more open-ended due to the varying representations that may be used for the 3D reference model. As a special case, the reference model may be implicitly defined by a set of reference images with annotated poses, in which case similar matching strategies may be used. In general, however, matching local image features w.r.t. a 3D model may involve multiple modalities, and the optimal strategy is not evident. That being said, I do believe that studying attention-based learned local feature matching strategies similar to [35], [37] may be an interesting research avenue, also when it comes to matching images to 3D models. The following are a few approaches that have been proposed for establishing 2D-3D correspondences:

- Scene / object coordinate regression: Training a machine learning model to, given the image contents, predict the corresponding 3D point for any image point on the silhouette of the projected 3D model. This approach is suitable to both localization and object pose estimation. Examples of this approach are [38], [39].

- Learned recognition of projected keypoints: Given a set of sparse 3D keypoints, manually or automatically selected, train a machine learning model to recognize their projections in the image. This approach is mostly relevant for object pose, since not a lot of keypoints are required. Examples of this approach are [40]–[42].

- Reconstructing a 3D point cloud from a bunch of reference images, while annotating the 3D keypoints with the corresponding image features. Later on, the points are to be detected in novel query images, and matched with the 3D point cloud descriptors. This approach is common to localization. The image features may either be traditional hand-crafted features such as SIFT [32], or learned features such as SuperPoint [34]. See [43] and the references therein for examples of such methods.

Robust Fitting

Now, if we have finally managed to establish our point correspondences, may we proceed to estimate the pose? Yes and no. Section 3.4 presents various methods used to solve this exact problem, a.k.a. the perspective-n-point (PnP) problem. The only issue is – how do we know that we can trust the correspondences? In practice, not only will our correspondences be subject to measurement noise; some of them, possibly a significant majority, may be utterly wrong. We call such flawed matches *outlier* correspondences, and they need to be identified and filtered out. Random sample consensus (RANSAC) [44] methods aim to do exactly this, by letting correspondences vote for pose hypotheses, until a hypothesis is found that many correspondences agree with.

The main idea regards to try and compare many different models, each fitted to a small random subset of the available data measurements. The motivation for using a small subset is to reduce the risk of outliers being included. Ideally, the subset is *minimal*, i.e. exactly as large as needed for uniquely determining a model which fits the subset perfectly. As an example, consider fitting a line to outlier-corrupted data points in two dimensions. In this case, RANSAC repeatedly draws pairs of points, since each pair exactly defines a line and thus constitutes a minimal subset. There are many different variations of RANSAC to improve the efficiency of the algorithm, but the main idea is simple: Select the model hypothesis (in this case, a line) which best fits the data in the sense of leading to the highest number of probable inliers. The inlier / outlier classification is done by thresholding an error function, e.g. point-to-line distance in this example.

Beyond line fitting, other problems for which RANSAC can be applied involve plane fitting, homography estimation, and essential matrix estimation, not to mention the PnP problem itself. In the case of PnP, a minimal solver requires 3 point correspondences, in which case at most 4 geometrically valid camera poses can be identified, and simply be tested one by one in the RANSAC scheme. In the minimal setting, the problem is known specifically as the perspective-3-point (P3P) problem, and there are many methods to solve it [44]–[47], usually amounting to solving a system of polynomial equations.

3.3 Relative Pose / Structure-from-Motion

So far, we have focused mainly on absolute pose estimation. For relative pose estimation, we no longer have any 3D reference model of the object or scene, and thus no fixed global coordinate system to relate to, and so the task is instead to determine the camera poses relative to each other, or relative to some arbitrarily assigned global frame. In addition, if the task is carried out by establishing 2D-2D point correspondences, as mentioned upon in Section 3.2, and relative camera poses have successfully been established, it is relatively straightforward to also reconstruct a 3D model by triangulating a point cloud from the correspondences. For each set of corresponding 2D points, this amounts to determining a 3D point which projects as close as possible to the 2D points.

Two Views: Epipolar Geometry

In the simplest scenario, there are only two camera views. As it turns out, in this case there is a quite simple relationship between the projections in one image and the other, captured by what is called *epipolar geometry*. An *epipole* refers to the the center of one camera projected into the other. Central to epipolar geometry are also the *epipolar lines*, all of which intersect at the epipole. The epipolar lines in one camera are in correspondence with the image points of the other camera, and without going into further detail, this relationship is captured by a 3×3 matrix called the essential matrix. Estimating the essential matrix from 2D-2D correspondences is relatively straightforward, often by using RANSAC together with a minimal solver. After having estimated the essential matrix, the relative pose of the two cameras can be established (almost!). The following section will elaborate on what cannot be determined.

Reconstruction Ambiguity

First and foremost, in a relative pose setting, we can clearly not recover absolute camera poses, due to the arbitrary choice of global cartesian coordinate frame. In addition to this, the relative pose estimates suffer from a scale ambiguity, since when there is no 3D reference model, it is unfortunately not possible to measure absolute distances[b]. That is, the scale of the scene can not be inferred from projections

[b] Knowledge of a 3D reference model is not a fundamental requisite for measuring distance using computer vision. Another alternative is stereo vision, where one instead relies on the relative pose between a pair

alone. Strictly speaking there is actually an ambiguity w.r.t. reflection as well, but mirrored solutions can be identified and eliminated, since the 3D points would end up behind the cameras in which they were observed. The result is a so-called *Euclidean reconstruction*, and if in addition the scale is determined, it is called a *metric reconstruction*.

Furthermore, recall from Section 3.1 that, provided the 3D reference model is known, not only can the absolute camera pose be determined from the (normalized) image point correspondences, but it is even possible to simultaneously determine the intrinsic camera parameters by solving the uncalibrated resectioning problem given the raw pixel coordinates of the corresponding image points[c]. In contrast, for relative pose estimation it is absolutely crucial to estimate and eliminate the camera intrinsics, or we will face a phenomenon known as projective ambiguity, meaning that the estimated solution, 3D points and cameras together, is distorted by an arbitrary projective transformation of $\mathbb{P}^3$, impossible to identify without calibrating the cameras. In general, camera calibration requires further knowledge of the scene, e.g. by capturing images of a known calibration object, followed by camera resectioning. If we have access to multiple images taken from the same camera device but from multiple viewpoints, the camera may however be determined by a technique known as auto-calibration, without the need for any calibration object.

Many Views: Structure-from-Motion

Considering the general case with an arbitrary number of views, m, relative pose estimation becomes less straightforward. In addition to there being many views, every 3D point may not be visible in every view, e.g. due to being out of view, due to occlusion, or simply due to unsuccessful detection and matching. The 2D-2D image point correspondences, which may now involve many images, are referred to as point tracks. In general, and especially for large scenes, some views will be more related than others, and sometimes a so-called view graph or pose graph is used to represent the view-to-view relationships / relative pose estimates. The problem of jointly estimating multiple relative camera poses as well as the structure of a scene, e.g. in the form of a 3D point cloud, is known as structure-from-motion (SfM), which can broadly be categorized into incremental SfM and global SfM.

Incremental SfM pipelines such as the popular COLMAP library [48]–[50], set

of cameras to be pre-established, similar to how humans are able to measure depth.

[c] If the camera calibration is already known, it is however preferable to eliminate it and solve for the camera pose only, as the noise-resilience increases and fewer observations are required.

out to reconstruct the entire scene by incorporating one additional view at a time. One way to initialize the reconstruction is from two views, using epipolar geometry and triangulation. For each additional view added, one typically detects and matches image keypoints with existing point tracks, and estimates, e.g. via RANSAC, its absolute pose w.r.t. the 3D point cloud (triangulated from the point tracks). This is followed by triangulation of newly matched point tracks, and typically an iterative refinement of camera poses and 3D points, known as bundle adjustment (BA, see Section 3.4), before moving on to the next view. It should be noted that incremental SfM suffers from the risk of drift to occur in the solution. In case of the camera trajectory being a loop, detecting the loop closure and running bundle adjustment may to some extent compensate for the drift, but convergence is not guaranteed for large drifts. Very related to incremental SfM methods are simultaneous localization and mapping (SLAM) methods, which essentially solve the same problem, but assume causal observations, focus more on real-time performance, and often involve fusion of multiple sensors via sequential filtering of measurements.

In global SfM, all camera poses are instead estimated simultaneously. The typical pipeline starts from pairwise relative pose estimates, between whatever views they can be estimated. This is followed by an optimization problem on the entire pose graph, known as pose graph optimization or pose averaging, possibly eliminating the translations and solving a separate rotation averaging problem instead. In any case, the problem amounts to determining absolute camera pose estimates for which the given relative poses are as consistent as possible. As a final step, one triangulates the 3D points and carries out bundle adjustment.

There are also alternative global SfM approaches, such as projective factorization and related methods (see e.g. [51] for an overview), or emerging learning-based methods such as [52], or Paper A in this thesis.

3.4 Pose Optimization

In Section 3.2, we have already seen how RANSAC can solve the PnP problem in the presence of outliers, through the combination of minimal solvers and voting. RANSAC has highly favorable robustness properties, but it is a rather blunt instrument. First, it should be noted that RANSAC can be rather costly. In addition to that, while the resulting solution provided by the minimal solver is ideally tolerated by many measurements, it is still suboptimal, as it does not care to minimize residual errors beyond being within the acceptance threshold. If we are fortunate enough to face an outlier-free scenario, we may instead use more sophisticated optimization techniques. A particular such scenario is to carry out a final fitting w.r.t. all inlier correspondences as determined by RANSAC. Another outlier-free scenario could be camera calibration given highly reliable point correspondences, e.g. manually annotated or determined by visual markers. A third scenario could be relative pose estimation given optical flow correspondences[d]. On a final note, in addition to correspondence-based pose estimation, other pose estimation problems such as rotation averaging may also be solved using optimization methods.

Local Optimization and Pose Refinement

Given a measured image point $\mathbf{x}$, a corresponding 3D point $\mathbf{X}$, and a camera pose $\mathbf{P}$, we may define the so-called reprojection error r by measuring the deviation, a.k.a. the residual, of the measurement $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2)$ from the 3D point projection $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) \sim \mathbf{PX}$ (expressed in homogeneous coordinates):

$$\begin{aligned} r(\mathbf{x}, \mathbf{P}, \mathbf{X}) &= \|\mathbf{r}\| \\ &= \|(\hat{x}_1, \hat{x}_2) - (\bar{x}_1/\bar{x}_3,\ \bar{x}_2/\bar{x}_3)\| \\ &= \left\| \hat{\mathbf{x}} - \frac{\mathbf{P}^{1:2^T}\mathbf{X}}{\mathbf{P}^{3^T}\mathbf{X}} \right\|, \end{aligned} \tag{3.1}$$

where $\mathbf{P}^i$ denotes the i:th row of $\mathbf{P}$, as a column vector, and $\mathbf{P}^{i:j}$ denotes a range of rows, stacked column-wise. Squaring and summing up the reprojection errors of all image point measurements, one acquires the objective function

$$E = \sum_{i=1}^{m} \sum_{j \in \mathcal{V}_i} r(\mathbf{x}_{ij}, \mathbf{P}_i, \mathbf{X}_j)^2, \tag{3.2}$$

where $\mathcal{V}_i$ denotes the set of indices j of the 3D points $\mathbf{X}_j$ visible in view $\mathbf{P}_i$, and $\mathbf{x}_{ij}$ is the measured projection of 3D point j in view i. For absolute pose estimation, $\mathbf{X}$ would be known and m may be 1 while for relative pose estimation / SfM, $\mathbf{X}$ would be unknown, but measured in multiple views. In any case, (3.2) can be minimized locally using iterative optimization. When all parameters are free (i.e. SfM), this is known as Bundle Adjustment (BA). Quite commonly, an optimization algorithm known as Levenberg-Marquardt is utilized, which is a dampened version of Gauss-Newton. The

[d] Optical flow correspondences may be relatively reliable due to the limited visual change between consecutive video frames.

Gauss-Newton family of optimization methods are specific to non-linear least-squares problems like (3.2), and provide a compromise between the number of iterations versus the computational cost of each iteration. Gauss-Newton is similar to Newton's method, but approximates the Hessian of E by $\mathbf{H} \approx 2\mathbf{J}_{\mathbf{r}}^T \mathbf{J}_{\mathbf{r}}$, where $\mathbf{J}_{\mathbf{r}}$ is the Jacobian of $\mathbf{r}$ w.r.t. the parameters, i.e. the first-order derivatives. The approximation stems from an assumption that $\mathbf{r}$ is small, such that additional terms of H involving $\mathbf{r}$ can be ignored. These terms also involve the second-order derivatives of $\mathbf{r}$, which are costly to compute. Far away from the optimal solution, the assumption is less reasonable, and can lead to complications. The dampened Levenberg-Marquardt algorithm is then a viable alternative, as the unreliable approximate Hessian is blended with some amount of the identity matrix, resulting in a blend between the efficient Gauss-Newton and reliable gradient descent. It is often desirable to use a relatively strong dampening initially, and reduce it while approaching the optimum and $\mathbf{r}$ becomes smaller.

There is, however, quite a dilemma – (3.2) is not convex and has many local minima that are globally suboptimal. Thus, starting from a random initialization, we would end up with a globally optimal solution only if we are lucky enough that the initialization resides in the basin of convergence of the algorithm. There are ways of widening the basin of convergence, e.g. by using a neural network parameterization of the SfM solution as done in [52]. Another strategy to achieve this for SfM is to consider alternative loss functions related to projective factorization such as pOSE [53] and its derivatives (which are applied in the uncalibrated setting), see [51] for an overview. In general, however, local pose optimization / bundle adjustment is used primarily as a refinement step to improve the precision of the solution.

Global Pose Optimization

Estimating camera pose by solving an optimization problem is an appealing approach, and while the sum of squared reprojection errors (3.2) is in general only feasible to minimize locally, there are other formulations of pose optimization for which it may be possible to determine a globally optimal solution. These formulations usually involve a polynomial objective function (no perspective division), often allowing for elimination of the translation components of the camera pose parameters. Examples of common error functions used as alternatives to the squared reprojection error

$$\left\| \hat{\mathbf{x}} - \frac{\mathbf{P}^{1:2^T}\mathbf{X}}{\mathbf{P}^{3^T}\mathbf{X}} \right\|^2 \tag{3.3}$$

are the squared back-projection error

$$\left\|\left(\mathbf{P}^{3^T}\mathbf{X}\right)\cdot\hat{\mathbf{x}}-\mathbf{P}^{1:2^T}\mathbf{X}\right\|^2 \tag{3.4}$$

and the squared point-to-ray distance

$$\left\|\left(I-\mathbf{v}\mathbf{v}^T\right)\mathbf{P}\mathbf{X}\right\|^2, \quad \mathbf{v}=\frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|}, \tag{3.5}$$

where again $\hat{\mathbf{x}}$ denotes a measured image point, $\mathbf{P}=\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$ denotes a calibrated camera matrix, and superscripts denote a row or range of rows, stacked columnwise. The errors (3.4) and (3.5) both disregard the projection to *image space*, and are instead calculated in *object space*, thus being referred to as object space errors (OSE). In both cases, the squared Euclidean distance from 3D point to its projection on a measured viewing ray is considered, where the projection is either parallel to the principal plane of the camera, or orthogonal to the ray. Object space error functions are useful components for absolute pose estimation in particular, occurring in various formulations of the PnP problem [13], [54]–[58]. Theoretically, PnP can actually be solved by globally minimizing the reprojection error (3.3) itself, e.g. using Branch-and-Bound [59], but the efficiency of this approach is unfortunately not adequate for practical purposes. There are also PnP methods that do not minimize geometrically meaningful errors, e.g. [11], [12].

Global pose optimization can be applied to SfM as well, by jointly optimizing the entire set of absolute camera poses $\mathbf{P}_i,\ i=1,\dots,m$ to comply with a set of relative pose measurements. This is known as pose-graph optimization (PGO), and one can also eliminate the translation components, and solve a rotation averaging problem instead. It should, however, be noted that global SfM methods based on pose optimization are often less scalable in terms of the number of views as compared to incremental SfM. As objective function for rotation averaging, one may use a rotation-to-rotation metric to quantify the inevitable inconsistency between the estimated solution and the measurements. A natural choice of error between two rotation matrices $\mathbf{S}$ and $\mathbf{R}$ is the angular distance $d_\angle(\mathbf{S}^T\mathbf{R})$ associated with the relative rotation, but easier to optimize is the squared Frobenius norm $\|\mathbf{S}-\mathbf{R}\|_F^2$ of the residual, a.k.a. the squared chordal distance of the embedding in $\mathbb{R}^{3\times 3}=\mathbb{R}^9$. The two metrics are however more related than it may seem at first glance, since $\|\mathbf{S}-\mathbf{R}\|_F = 2\sqrt{2}\sin\left(\frac{1}{2}d_\angle(\mathbf{S}^T\mathbf{R})\right)$ [60]. There is also a similar relation with the quaternion distance, see [60].

While these polynomial objectives are simpler to optimize than reprojection errors, a major challenge remains in respecting the rotational constraints when optimizing the rotation parameters. If parameterizing a rotation in terms of its 9 rotation matrix elements, feasibility can be formalized with the quadratic equality constraints $\mathbf{R}^T\mathbf{R} = I$ on orthonormality, together with an additional constraint ensuring $\det \mathbf{R} = 1$, e.g. the quadratic cross product constraint $\mathbf{R}^3 = \mathbf{R}^1 \times \mathbf{R}^2$ on the rows of $\mathbf{R}$. Using a quaternion parameterization $\mathbf{q} \in \mathbb{R}^4$, the unit-norm constraint $||\mathbf{q}||^2 = 1$, also a quadratic equality constraint, is enough to ensure a valid rotation. For none of the parameterizations the corresponding feasible region is a convex set, and thus, pose optimization problems are not completely straightforward to solve. The unit-quaternion constraint $||\mathbf{q}||^2 = 1$ is in general the easier of the two, but a quaternion formulation may come at the price of the objective function becoming a polynomial of higher degree compared to using the rotation matrix parameterization.

For PnP as well as rotation averaging, we are facing a non-convex equality-constrained polynomial optimization problem (POP). For PnP, which is the smaller problem, a common strategy regards identifying every stationary point of the optimization problem, and ranking them to determine the global optimum / optima. This amounts to solving a system of polynomial equations, and many variations to it have been studied, e.g. [11]–[13], [55]. As a side note, in the minimal case (P3P), rather than determining the stationary points of the POP, one is looking for solutions where the projections of the three 3D points align perfectly with the corresponding image point measurements, but this can also be done by solving a system of polynomial equations.

Another approach to global pose optimization is to minimize a semidefinite relaxation of the POP. In this case, the optimization problem is lifted to a larger number of variables, corresponding to monomials of higher degree, in which the objective function as well as the original constraints can be expressed as linear functions. For example, consider the quadratically constrained quadratic program (QCQP)

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{n+1}} \quad & \mathbf{x}^T\mathbf{M}\mathbf{x} \\ \text{subject to} \quad & \mathbf{x}^T\mathbf{A}_i\mathbf{x} = 0, \quad i = 1,\dots,l \\ & x_{n+1} = 1, \end{aligned} \tag{3.6}$$

where $M \succeq 0$. By lifting the variables $\mathbf{x} \in \mathbb{R}^{n+1}$ to a matrix variable $\mathbf{X} \in \mathbb{R}^{(n+1)\times(n+1)}$ of second-degree monomials in the elements of $\mathbf{x}$, (3.6) can be relaxed to (3.7):

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{R}^{(n+1)\times(n+1)}} \quad & \text{tr}\left(\mathbf{M}\mathbf{X}\right) \\ \text{subject to} \quad & \text{tr}\left(\mathbf{A}_i\mathbf{X}\right) = 0, \quad i = 1,\dots,l \\ & \text{tr}\left(\mathbf{e}\mathbf{e}^T\mathbf{X}\right) = 1 \\ & \mathbf{X} \succeq 0, \end{aligned} \tag{3.7}$$

where $\mathbf{e} = (0,\dots,0,1)^T$. If we would also add the constraint rank $(\mathbf{X}) = 1$ to (3.7), the problems would be identical, and neither easy to solve, but by simply disregarding this rank constraint we have acquired a semidefinite relaxation (SDR) of (3.6), so-called because it is an instance of a semidefinite program (SDP). SDPs are convex, and thus much easier to solve. Now, if we are lucky, the solution to (3.7) will also fulfill the rank constraint, in which case we may extract a globally optimal solution to (3.6). In that case, we say that the relaxation is tight, since there is no *gap* between the optimal value η^* of (3.6) and γ^* of (3.7), for which in general $\eta^* \geq \gamma^*$.

Fortunately, in practice the relaxations are tight more often than one may expect. In particular, if the problem stems from noise-free measurements the optimal objective value is zero, and the relaxation is tight. Similarly, if the noise level is very low, the probability is higher that the relaxation is tight [61]. For PnP, the relaxation is almost always tight, and only rare counter-examples have been reported, see [62] and Paper C in this thesis. For rotation averaging, tightness is also guaranteed within a relatively liberal range of noise [63].

If (3.7) is not tight, a higher level relaxation involving monomials of higher degree may be considered. The so-called moment-SOS / Lasserre hierarchy of relaxations [64], [65] formalize an infinite sequence of relaxations with monotonically decreasing gap to η^*. Monomials of higher degree are also required in case the POP is not a QCQP, but involves polynomials of higher degree in the objective function and / or constraints. A relevant example of this is the PnP problem, which leads to a quartic objective function if formulated using quaternions as in [54], [57], [58]. A benefit of the semidefinite relaxation approach is that the precision of the solutions tends to be high.

Finally, a novel approach to global optimization of the PnP problem is proposed in [56]. Instead of quaternions, rotation matrices are used to represent the rotation variable. While this constraint set is usually more difficult to handle, they relax the problem to consider all 3 × 3 matrices with Frobenius norm $\sqrt{3}$, including rotation matrices. This relaxed problem is relatively easy to solve, and all stationary points can be identified by solving an eigenvalue problem and extracting the corresponding eigenvectors. Each of these eigenvectors can then be projected to actual rotation matrices, and the authors prove that iterative local optimization will converge to a global minimizer of the original problem for at least one of these projections.

3.5 Pose Regression

Another strategy to estimating camera pose is to learn to predict the pose based on some observation, e.g. one or several images or a set of measured image point correspondences, fed as inputs to a machine learning model. The learned model then outputs a parameterization of the predicted camera pose. Perhaps the most common rotation parameterization in this context is unit-length quaternions, which by a straight-forward normalization enjoy the property of automatically satisfying the rotational constraints. Parameterizations with some redundancy such as truncated rotation matrices to their first two columns have however also been proposed [66].

Learning-based methods have great potential in several regards, especially in their ability to automatically extract meaningful features from complex or otherwise noisy, corrupted, or greatly varying data, such as outlier point correspondences or image observations subject to varying lighting conditions or seasonal variation, provided enough training examples including such effects can be acquired, either real or artifically generated / augmented. Another benefit is that learned methods can often perform inference blazingly fast compared to iterative approaches. Due to these benefits in efficiency and learned robustness, pose regression is indeed a quite appealing approach. The main issues with pose regression approaches as of today regards in part domain shifts, and in part precision. While high robustness can be achieved to greatly varying conditions within the domain in which training data has been acquired, generalization can be an issue, and it is amplified by the fact that the availability of training data is often limited due to the cost of 3D annotation. As for the precision, it is not uncommon that a learned method is paired with iterative refinement methods, in which the case the latter unfortunately dominates the computational burden during inference.

CHAPTER 4

Thesis Contributions

This chapter provides a summary of the papers which form the thesis contributions.

4.1 Paper A

Lucas Brynte, José Pedro Iglesias, Carl Olsson, Fredrik Kahl
Learning Structure-from-Motion with Graph Attention Networks
Submitted for Review, arXiv:2308.15984 (2023). .

In this paper, an efficient and powerful model architecture based on graph attention networks is presented for learning the complex problem of structure-from-motion. Given a set of observed image point tracks, the model is trained to regress camera poses and scene point coordinates. Although the training set consists of as little as 12 scenes of varying size, the model is demonstrated to generalize to novel test scenes, well enough for bundle adjustment to converge to an optimal or in some case nearly optimal solution. Therefore, unlike a preceding related method [52], costly fine-tuning of the model parameters on the novel test scenes is not required, resulting in accurate reconstructions to be recovered about $5 - 10\times$ faster than the incremental SfM pipeline COLMAP [48]–[50]. In addition to the increased efficiency compared to COLMAP, and the increased reconstruction quality compared to [52], regarding robustness it is demonstrated how the proposed method can conveniently be trained to disregard outlier measurements, by injecting artificial outliers to the training data.

4.2 Paper B

Lucas Brynte[*], Georg Bökman[*], Axel Flinth, Fredrik Kahl
Rigidity Preserving Image Transformations and Equivariance in Perspective
Scandinavian Conference on Image Analysis (2023) .

The inspiration to this paper comes from the great success of convolutional neural networks, which is often attributed to their translation equivariance. In general, the equivariance constraints imposed on a model, such as translation equivariance for CNNs, can improve learning by exploiting data symmetries. A requisite for this, however, is that the symmetry group w.r.t. which equivariance is imposed, reflects actual symmetries present in the data. This paper points out that for CNNs applied on 3D inference tasks, this is only approximately the case. That is, nearby the principal point, an image translation appears similar to a rigid motion of the camera, namely rotation about its center, which probably explains why the equivariance properties of CNNs are beneficial in the first place. Far away from the principal point, this is however no longer the case, and furthermore, the only rigid camera motions which can be associated with corresponding image transformations, are indeed the rotations of a camera about its center, which correspond to a subset of homographies that we call 'rotational homographies', as their 3×3 matrix representations are, up to scale, members of $\mathrm{SO}(3)$. Given this observation, two alternative techniques are advocated for improving the equivariance properties of CNN models trained on 3D inference tasks. The first technique is to radially distort every image to a warped version with reduced perspective effects, namely the azimuthal equidistant map projection, on which the translation symmetries learned by a CNN more closely approximate the effect of camera rotations. The second technique is to learn the equivariance w.r.t. perspective effects, by augmenting the training data with random rotational homographies. Both techniques demonstrate improvements on object pose estimation, using an off-the-shelf state-of-the-art CNN model trained to regress the pose. The results are improved both in terms of pose estimation quality, as well as sample efficiency, with data augmentation performing the best.

[*] Equal contribution.

4.3 Paper C

Lucas Brynte, Viktor Larsson, José Pedro Iglesias, Carl Olsson, Fredrik Kahl
On the Tightness of Semidefinite Relaxations for Rotation Estimation
Journal of Mathematical Imaging and Vision **64**, 57–67 (2022). .

This work differs from the others in that it focuses solely on pose optimization and not on learning. More specifically, it regards semidefinite relaxations of optimization problems with quadratic objective functions and rotational constraints, i.e. a family of quadratically constrained quadratic programs (QCQPs) including the applications of registration, hand-eye calibration, absolute camera pose estimation and rotation averaging. While in general semidefinite relaxations are not guaranteed to be tight, for surprisingly many problem instances (particularly when involving few rotations) they are. Motivated by this observation, a theoretical framework based on tools from algebraic geometry is introduced for analyzing the power and limitations of such relaxations. It is shown that despite being empirically rare, there are still plenty of failure cases for which the relaxation is not tight, even in the case of a single rotation. In particular, for absolute camera pose estimation (referred to as resectioning in the paper), we were at the time of submission not aware of a single non-tight problem instance having been reported in the literature. Despite this, we managed to find such problems, unfortunately proving that they do exist. Concurrently with our paper, Alfassi et al. [62] also proved the existence of non-tight semidefinite relaxations of instances of camera resectioning. They used a unit quaternion parameterization, resulting in a fourth degree polynomial objective for resectioning, which does not fit into our framework, but considering both works one can conclude that resectioning is not necessarily tight, neither for the unit quaternion parameterization, nor for the parameterization in terms of $\mathrm{SO}(3)$ matrix elements that we used. In general, the tightness of the semidefinite relaxation is related to the noise level[61] (indeed, the relaxation of a least squares problem is always tight in the noise-free case), and it has e.g. been observed that the rotation averaging problem is always tight if the relative pose measurements are not subject to too high noise levels [63]. The bound depends on the specific pose graph and in particular its connectivity, but as an example, if the pose graph is fully connected, noise levels as high as $42.9°$ are tolerated. As for the approach of estimating absolute camera pose by solving an SDP, it may be that the method only breaks down for quite significant noise levels, and in that sense is very robust, but from the existence of counterexamples we can conclude that the applicability of the method is not independent of noise.

4.4 Paper D

Lucas Brynte, Fredrik Kahl
Pose Proposal Critic: Robust Pose Refinement by Learning Reprojection Errors
British Machine Vision Conference (2020) .

This paper focuses on pose estimation of partially occluded objects given a single image and a CAD model reference of the object, i.e. an instance of absolute pose estimation. Occlusion poses a significant challenge for object pose estimation, and it is not uncommon that a method which performs well under ideal conditions breaks down in case of occlusion. One strategy which can be used to improve object pose estimation results in general, but has proven particularly useful for coping with occlusion, is rendering-based pose refinement, where a model learns to determine a mismatch in object pose between the observed image and a rendered image of the object in a “best-guess” pose. Unlike previous methods [67]–[69] which learn to estimate all degrees of freedom of the relative pose mismatch, this paper opts to simplify the learning objective, learning to estimate the average reprojection error associated with the relative pose error, rather than the pose itself. The result is increased pose estimation quality and robustness to partial occlusion, at the expense of efficiency. Inference takes 33 seconds on average, and involves iterative local minimization of the predicted reprojection error w.r.t. the pose hypothesis. The backbone of the model is a pretrained optical flow network, fine-tuned on the task at hand.

4.5 Paper E

Carl Toft, Erik Stenborg, Lars Hammarstrand, **Lucas Brynte**, Marc Pollefeys, Torsten Sattler, Fredrik Kahl
Semantic Match Consistency for Long-Term Visual Localization
European Conference on Computer Vision (2018) .

This paper focuses on long-term visual localization, meaning that the camera is to be localized despite a long period of time having passed from when the map was created. Within the context of a standard correspondence-based localization pipeline, a learned semantic segmentation model is leveraged on to rank feature correspondence quality, used to bias the sampling of correspondences within RANSAC. The efficiency of the sampling scheme is thus increased which, given a limited computational budget, leads to higher quality pose estimates, robust to seasonal variations.

CHAPTER 5

Concluding Remarks and Future Work

In this thesis I have explored various camera pose estimation methods based on machine learning and optimization, with primary focus on learning, in three different regards: Quality, Robustness, and Efficiency. Absolute pose applications such as visual localization and object pose estimation have been considered, as well as structure-from-motion.

The most recent article, Paper A, deals with the most complex of the applications, structure-from-motion. An efficient and powerful graph attention network model is designed and trained to regress camera poses and scene point coordinates, and is demonstrated to generalize to test scenes. When combined with bundle adjustment, the quality of the reconstruction is superior to previous work [52], and almost on par with COLMAP, while the efficiency in terms of execution time is orders of magnitude better than [52], and $5 - 10\times$ faster than COLMAP [48]–[50]. Furthermore, it is demonstrated how robustness w.r.t. outlier observations can be achieved by artificial outlier injection to the training data.

In Paper B, the equivariance properties of CNN models for 3D inference tasks such as object pose estimation are scrutinized. More specifically, it is noted that image translations can not result from rigid camera motion, and therefore that the symmetries being learned are not in perfect alignment with the actual symmetries in the data. Two techniques are proposed for improving this alignment, either what can be seen as an intentional radial distortion of the images, or a data augmentation scheme. It is shown that both techniques can lead to an improvement on the task of object pose estimation, both in terms of pose estimation quality and training data efficiency.

In Paper C, the power and limitations of semidefinite relaxations are analyzed for QCQPs with rotational constraints, using toolss from algebraic geometry. In particular, together with the empirical observation that a surprising amount of such problem instances result in tight relaxations, rare non-tight counterexamples are presented, even for absolute camera pose estimation. Together with the related concurrent work [62], it is now proven that non-tight semidefinite relaxations of instances of absolute camera pose estimation do exist, both for the parameterization in terms of matrix elements used by us, and for unit quaternions used by them, and in that sense the method is not robust to noise to the extent of its applicability being independent of it.

Paper D focuses on object pose estimation, in particular pose refinement, using a combination of learning and optimization. A rendering-based refinement method is proposed, with the particular edge of being robust to partial occlusions, leveraging on a simplified learning task, where a CNN is trained to estimate the reprojection error between an observed and a rendered image. At the expense of efficiency, the method results in increased pose estimation quality and robustness to partial occlusion.

Finally, Paper E presents a pioneering work on combining geometry and semantics for the application of long-term visual localization. A standard correspondence-based localization pipeline is adapted to leverage on a learned semantic segmentation model, used to rank feature correspondence quality when sampling minimal sets in RANSAC. The result is a more efficient sampling scheme, higher quality pose estimates, and robustness to seasonal variations.

5.1 Future Work

Regarding learned structure-from-motion and Paper A, I think improving the quality of the direct inference reconstructions should be of priority since the execution time, while still quite a lot faster than COLMAP, is currently completely dominated by bundle adjustment. One way to achieve this with relatively little effort could be to simply consider larger training datasets with greater scene variability. Another avenue of research could be to explore similar architectures for predicting pairwise relative poses, or other techniques for better handling the reconstruction ambiguity.

Regarding exploitation of data symmetries when learning to estimate camera pose from image contents, I think considering the rigidity preserving image transformations analyzed in Paper B makes perfect sense, but while the proposed radial distortion technique results in image translations to more accurately resemble camera rotation, there is still a discrepancy between the two transformations, especially in the periphery of an image, and there may be alternative formulations for which this discrepancy is limited further, especially if considering more novel architectures than CNNs. The transfer learning benefits that come with using off-the-shelf pretrained backbone models should, however, not be overlooked.

When it comes to the semidefinite relaxations of pose estimation problems analyzed in Paper C, and especially the application to absolute camera pose estimation, the jury is, to my knowledge, still out regarding exactly how rare the the problem instances with non-tight relaxations are. In particular, there is reason to suspect that tightness could be guaranteed up to quite high levels of noise, since reports of non-tight problem instances are few, and establishing such bounds would make an appreciated contribution.

While the implicit approach to rendering-based pose refinement presented in Paper D demonstrated increased robustness w.r.t. occlusion, it came at a relatively high price in terms of efficiency. First, there are probably more efficient ways to acquire smooth derivatives than to take finite differences with large step sizes, such as making the error function smoother. This could perhaps be achieved by an appropriate choice of regularization, or by replacing the ReLU activations with a smoother version such as ELU [70], or even with the periodic activation functions proposed in [71].

Finally, I see potential for learned attention-based feature matchers similar to [35], [37] for absolute pose estimation. For estimating object pose, a common use case is within manufacturing, where CAD models are often available for the objects / items of interest. Learning 2D-3D matches is a challenging multimodal task, especially for textureless objects, but I think the cross-attention mechanism is a promising way of handling the multimodality. By training a matcher on a few objects with associated pose-annotated reference images, the goal would be to generalize to unseen objects given only their CAD models, without any reference images with annotated pose. For visual localization, learned multimodal matching between images and point clouds may also be a promising direction, but the architecture has to be carefully designed to be able to handle large scenes. The learned approach to feature matching may also be useful for handling seasonal variations or other changing conditions.

References

[1] *The Niépce heliograph*, https://www.hrc.utexas.edu/niepce-heliograph/, Accessed: 2023-12-29.

[2] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[3] 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, 1998.

[4] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[6] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 10 012–10 022.

[7] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 16 000–16 009.

[8] OpenAI, *Gpt-4 technical report*, 2023.

[9] A. Ramesh, M. Pavlov, G. Goh, *et al.*, “Zero-shot text-to-image generation,” in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, Jul. 2021, pp. 8821–8831.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, 2021.

[11] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o(n) solution to the pnp problem,” *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, Feb. 2009, ISSN: 1573-1405.

[12] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, “Revisiting the PnP problem: A fast, general and optimal solution,” in *ICCV*, 2013.

[13] L. Kneip, H. Li, and Y. Seo, “Upnp: An optimal o(n) solution to the absolute pose problem with universal applicability,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 127–142, ISBN: 978-3-319-10590-1.

[14] L. Bottou *et al.*, “Stochastic gradient learning in neural networks,” *Proceedings of Neuro-Nımes*, vol. 91, no. 8, p. 12, 1991.

[15] F. Rosenblatt, *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan books Washington, DC, 1962, vol. 55.

[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[17] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080.

[18] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, *The expressive power of neural networks: A view from the width*, 2017.

[19] K. Fukushima, “Cognitron: A self-organizing multilayered neural network,” *Biological cybernetics*, vol. 20, no. 3-4, pp. 121–136, 1975.

[20] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[21] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, Eds., ser. Proceedings of Machine Learning Research, vol. 15, Fort Lauderdale, FL, USA: PMLR, Nov. 2011, pp. 315–323.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[23] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric deep learning: Grids, groups, graphs, geodesics, and gauges*, 2021.

[24] T. Cohen and M. Welling, “Group equivariant convolutional networks,” in *Int. Conf. on Machine Learning*, 2016.

[25] M. Weiler and G. Cesa, “General e (2)-equivariant steerable cnns,” *Advances in neural information processing systems*, vol. 32, 2019.

[26] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, “Deep sets,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[27] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[28] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *International conference on machine learning*, PMLR, 2017, pp. 1263–1272.

[29] P. W. Battaglia, J. B. Hamrick, V. Bapst, *et al.*, *Relational inductive biases, deep learning, and graph networks*, 2018.

[30] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[31] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[32] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[33] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, “LIFT: Learned Invariant Feature Transform,” in *ECCV*, 2016.

[34] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018.

[35] 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 (CVPR)*, Jun. 2021, pp. 8922–8931.

[36] J. Edstedt, G. Bökman, M. Wadenbäck, and M. Felsberg, “DeDoDe: Detect, Don’t Describe — Describe, Don’t Detect for Local Feature Matching,” in *2024 International Conference on 3D Vision (3DV)*, IEEE, 2024.

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

[38] E. Brachmann and C. Rother, “Learning Less is More - 6D Camera Localization via 3D Surface Regression,” in *CVPR*, 2018.

[39] E. Brachmann and C. Rother, “Visual camera re-localization from RGB and RGB-D images using DSAC,” *TPAMI*, 2021.

[40] M. Oberweger, M. Rad, and V. Lepetit, “Making deep heatmaps robust to partial occlusions for 3D object pose estimation,” in *The European Conference on Computer Vision (ECCV)*, Sep. 2018.

[41] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, “PVNet: Pixel-wise voting network for 6DoF pose estimation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[42] C. Song, J. Song, and Q. Huang, “HybridPose: 6D object pose estimation under hybrid representations,” in *CVPR*, 2020.

[43] C. Toft, “Towards robust visual localization in challenging conditions,” Ph.D. dissertation, Chalmers University of Technology, 2020.

[44] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography,” *Commun. Assoc. Comp. Mach.*, vol. 24, pp. 381–395, 1981.

[45] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[46] L. Kneip, D. Scaramuzza, and R. Siegwart, "A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation," in *CVPR*, 2011.

[47] M. Persson and K. Nordberg, "Lambda twist: An accurate fast robust perspective three point (p3p) solver," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.

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

[49] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[50] J. L. Schönberger, *Colmap code*, `https://colmap.github.io/`.

[51] J. P. Iglesias, "Towards reliable and accurate global structure-from-motion," Ph.D. dissertation, Chalmers University of Technology, 2023.

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

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

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

[55] 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.

[56] G. Terzakis and M. Lourakis, "A consistently fast and globally optimal solution to the perspective-n-point problem," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, Springer, 2020, pp. 478–494.

[57] I. Jubran, F. Fares, Y. Alfassi, F. Ayoub, and D. Feldman, “Newton-pnp: Real-time visual navigation for autonomous toy-drones,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 363–13 370.

[58] D. Keren, M. Osadchy, and A. Shahar, “A fast and reliable solution to pnp, using polynomial homogeneity and a theorem of hilbert,” *Sensors*, vol. 23, no. 12, 2023, ISSN: 1424-8220.

[59] C. Olsson, F. Kahl, and M. Oskarsson, “Branch-and-bound methods for Euclidean registration problems,” *IEEE TPAMI*, vol. 31, no. 5, pp. 783–794, 2009.

[60] R. Hartley, J. Trumpf, Y. Dai, and H. Li, “Rotation averaging,” *International journal of computer vision*, vol. 103, pp. 267–305, 2013.

[61] D. Cifuentes, S. Agarwal, P. A. Parrilo, and R. R. Thomas, “On the local stability of semidefinite relaxations,” *arXiv preprint arXiv:1710.04287*, 2017.

[62] Y. Alfassi, D. Keren, and B. Reznick, “The non-tightness of a convex relaxation to rotation recovery,” *Sensors*, vol. 21, no. 21, 2021, ISSN: 1424-8220.

[63] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin, “Rotation averaging and strong duality,” in *CVPR*, 2018.

[64] J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” *SIAM J. on Opt.*, vol. 11, no. 3, pp. 796–817, 2001.

[65] D. Henrion, M. Korda, and J. B. Lasserre, *The Moment-SOS Hierarchy*. World Scientific, 2020.

[66] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[67] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, “Deep model-based 6D pose refinement in RGB,” in *The European Conference on Computer Vision (ECCV)*, Sep. 2018.

[68] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, “DeepIM: Deep iterative matching for 6d pose estimation,” *International Journal of Computer Vision*, vol. 128, no. 3, pp. 657–678, Nov. 2019.

[69] S. Zakharov, I. Shugurov, and S. Ilic, “DPOD: 6D pose object detector and refiner,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019.

[70] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv:1511.07289*, 2016.

[71] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 7462–7473.