On improving visual odometry through learned pseudo-sensors

by

Valentin Peretroukhin

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto

Abstract

On improving visual odometry through learned pseudo-sensors

Valentin Peretroukhin
Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto
2019

The ability to estimate *egomotion*, that is, to track one's own pose through an unknown environment, is at the heart of safe and reliable mobile autonomy. By inferring pose changes from sequential sensor measurements, egomotion estimation forms the basis of mapping and navigation pipelines, and permits mobile robots to self-localize within environments where external localization sources are intermittent or unavailable. Visual and inertial egomotion estimation, in particular, have become ubiquitous in mobile robotics due to the availability of high-quality, compact, and inexpensive sensors that capture rich representations of the world. To remain computationally tractable, 'classical' visual-inertial pipelines (like visual odometry and visual SLAM) make simplifying assumptions that, while permitting reliable operation in ideal conditions, often lead to systematic error in the final localization output. In this thesis, we present several data-driven models that serve to complement (but not replace) conventional pipelines by inferring latent information from sensor data to improve the final egomotion estimate. Our approach retains a prior based on a pipeline with interpretable components and decades of prior research, while leveraging high-capacity hyper-parametric models to better model sensor uncertainty, or to extract complementary information in the form of a 'pseudo-sensor' whose output can be fused with the original pipeline to improve localization. We validate our methods on several kilometres of sensor data collected in sundry settings such as urban roads, indoor labs, and planetary analogue sites in the high arctic.

Epigraph

A little learning is a dangerous thing;
drink deep, or taste not the Pierian
spring: there shallow draughts
intoxicate the brain, and drinking
largely sobers us again.

Alexander Pope

To all those who encouraged (or, at least, *never discouraged*) my intellectual wanderlust.

Acknowledgements

This document would not have been possible without the generous support and guidance of my supervisor[1], the perennial love of my family and friends[2], and the limitless patience of my lab mates[3]. Thank you all.

[1] as well as all my collaborators and academic mentors
[2] especially the support and encouragement of Elyse
[3] in humouring my insatiable need for debate and banter

Contents

Notation

a	:	Symbols in this font are real scalars.
$\mathbf{a}$	:	Symbols in this font are real column vectors.
$\mathbf{A}$	:	Symbols in this font are real matrices.
$\sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$	:	Normally distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{R}$.
$E[\cdot]$	:	The expectation operator.
$\underrightarrow{\mathcal{F}}_a$	:	A reference frame in three dimensions.
$(\cdot)^\wedge$	:	An operator associated with the Lie algebra for rotations and poses. It produces a matrix from a column vector.
$(\cdot)^\vee$	:	The inverse operation of $(\cdot)^\wedge$
$\mathbf{1}$	:	The identity matrix.
$\mathbf{0}$	:	The zero matrix.
$\mathbf{p}_a^{c,b}$	:	A vector from point b to point c (denoted by the superscript) and expressed in $\underrightarrow{\mathcal{F}}_a$ (denoted by the subscript).
$\mathbf{R}_{a,b}$	:	The 3×3 rotation matrix that transforms vectors from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\mathbf{p}_a^{c,b} = \mathbf{C}_{a,b}\mathbf{p}_b^{c,b}$.
$\mathbf{T}_{a,b}$	:	The 4×4 transformation matrix that transforms homogeneous points from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\mathbf{p}_a^{c,a} = \mathbf{T}_{a,b}\mathbf{p}_b^{c,b}$.

Chapter 1

Introduction

> If you do not know where you come from, then you don't know where you are, and if you don't know where you are, then you don't know where you're going. And if you don't know where you're going, you're probably going wrong.
>
> Terry Pratchett

1.1 Autonomy and humanity through the ages

This thesis deals with the improvement of autonomous systems, which, in some form or another, have been imagined and realized for the bulk of recorded history. In Greek mythology, the god Hephaestus was said to create talking mechanical hand-maidens, while early Hindu and Buddhist texts tell of *yantakara* that lived in Greece and created machines that helped in trade and farming. The secret methods of the *yantakara* (the early 'roboticists') were closely guarded, and mechanical assassins were said to pursue and kill any person who revealed their techniques[1].

Figure 1.1: A robot rebellion from Karel Capek's 1920 play, *Rossum's Universal Robots*.

[1] Please be careful distributing this thesis.

Since the industrial revolution, the idea of an autonomous machine–one that requires no, or very minimal, human intervention or oversight to operate–has been imagined in different ways. Depending on one's perspective, autonomous machines have perennially promised to either usher in a utopia of freedom, or threatened to bring about an age of job loss and social upheaval that worsens socioeconomic divisions. Much like the Luddites of the 19th century, the social critics of the 21st century have continued the dialectic to understand the social ramifications of modern *yantakara* and their newly-created autonomous hand-maidens.

These controversial origins are embedded even within the modern name of for the academic field, *robotics*. The word *robot* comes from an anglicized title of a science fiction play, Rossum's Universal Robots, written by the Czech playwright Karel Capek in 1920 (see Figure 1.2). In naming the play, the word *robot* was derived from the Slavic term for slave, *rab*, and its Czech derivative for serf labour, *rabota*, while the name *Rossum* was inspired by the Czech word for reason, or intellect. Indeed, the concept of enslaved or embodied intelligence is at the heart of modern definitions of the discipline of robotics (Redfield, 2019). Much of the popular culture surrounding robots (e.g., Shelley's *Frankenstein*, Asimov's *I, Robot*, Kubrick's and Clarke's *2001: A Space Odyssey*) also paints a complicated picture of humanity's relationship with such enslaved machines. In this thesis, I focus on improving a specific part of a modern *mobile* autonomy pipeline, while minimizing the use of term *robot* to avoid maelstrom of philosophical and ethical problems that it connotes. I hope my work aids the march of technological progress towards a future which finds some Hegelian synthesis of autonomy and humanity—a future in which human-in-the-loop autonomous systems augment and improve the lot of many people while still negotiating and constantly considering the social costs that come with technological innovation.

1.2 Mobile Autonomy and State Estimation

While the looms and railroads of the industrial revolution were spurred by the discovery of steam engines and electricity, modern *mobile* autonomy was largely born out of the technological arms race of the cold war and the constraints and challenges associated with long-distance flight and extraterrestrial travel (see Grewal and Andrews (2010) for a history of one of the seminal algorithms in mobile autonomy, the Kalman filter). Indeed, much of the work on modern perception algorithms has its origins in the automated compilation of cold-war-era reconnaissance imagery and the design of extraterrestrial rovers like the Mars Exploration Rovers, *Spirit* and *Opportunity* (Scaramuzza and Fraundorfer, 2011). Much of the planning and control algorithms originate in American and Soviet defence-funded research (Nilsson, 1984; Thrun et al., 2006).

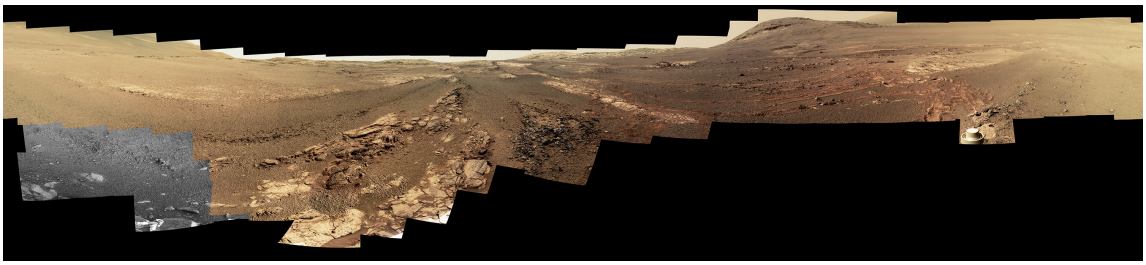
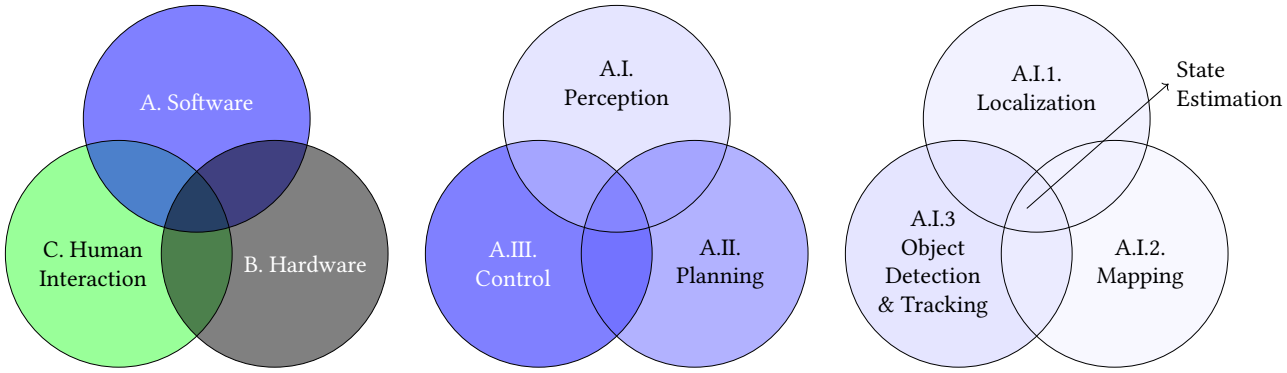


Figure 1.2: The last 360 degree panorama taken by the PanCam apparatus of the Mars Exploration Rover, *Opportunity*, at its final resting place on Mars, the western rim of the Endeavour Crater. Contact with *Opportunity* was lost shortly after this was captured, due to a severe dust storm. (Credit: NASA/JPL-Caltech/Cornell/ASU).

Once confined to carefully-controlled factories, autonomous mobile platforms have now began to show great promise in improving the safety of human transport, reducing the burden of repetitive, arduous jobs, and more efficiently leveraging limited resources for environmental monitoring. This newly-realized potential can be attributed to several factors: improvements in the cost and efficiency of computing devices (in terms energy efficiency, processing power, and overall size), the availability of relatively cheap, compact, high-quality sensors and rapid prototyping tools, and the development of open-source hardware, software platforms and datasets (e.g., the Robot Operating System, the KITTI Self-Driving Car dataset (Geiger et al., 2013)).

Despite decades of research, mobile autonomy as a field still has nebulous demarcations between subfields. I have attempted to provide a general overview of the field through a series of Venn diagrams in Figure 1.3. At the highest level, the field can be roughly divided into those researchers who study and develop software, those who study and develop hardware, and those who study and analyze the interaction between autonomous systems (composed of both software and hardware) and humans (Figure 1.3a). There is, of course, a plethora of overlap between all three of these rough categories. Within the software realm, there has historically been a distinction between those who study algorithms that deal with the perception of the interoceptive and exteroceptive data, those who study how to use that data to plan action, and those who study how to use those plans to control a system to execute that action (Figure 1.3b).

Within perception, which is the focus of this thesis, there are three general directions of research: localization, mapping and object detection and tracking. Localization and mapping can refer to self-localization, or egomotion estimation, which deals with the problem of estimating the pose of a moving platform through an unknown world, SLAM, simultaneous localization and mapping, which deals with the former problem and mapping simultaneously and finally, it can refer to localization within a known map given some hitherto unseen observation of that environment. The field of object detection and tracking (whether that be static



(a) Research strands in mobile autonomy.

(b) The components of software research.

(c) The components of perception.

Figure 1.3: Venn diagrams of modern mobile autonomy.

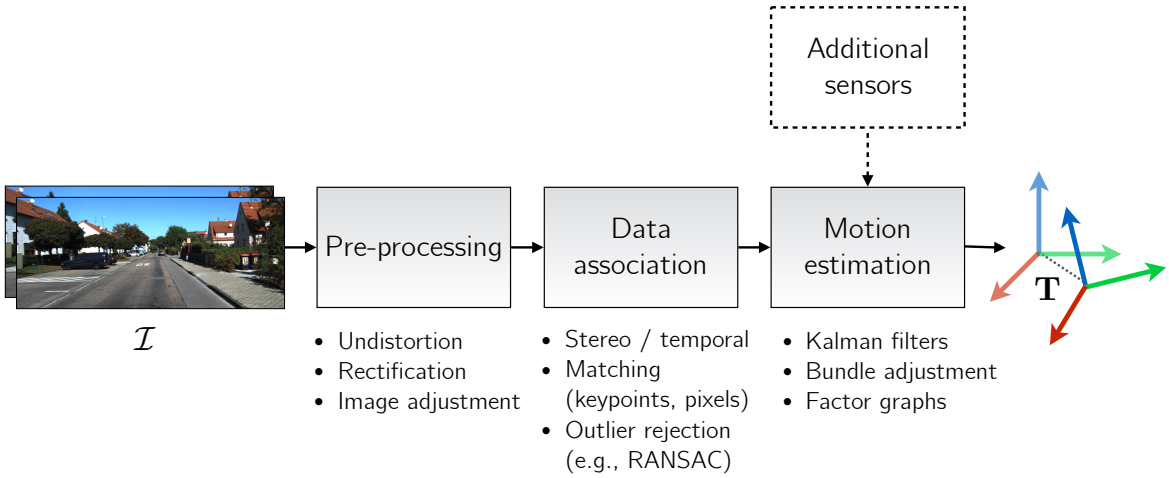


Figure 1.4: A 'classical' visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.

objects like stop signs, or moving objects like humans, animals or vehicles) uses much of the same underlying mathematics as the former two, but has historically been a separate strand of research. Broadly, the overlap of all three of these pursuits within the field of autonomy and robotics is referred to as *state estimation* Barfoot (2017).

1.3 The *State* of State Estimation

Central to *classical* state estimation algorithms (which, in this context, refers to the bulk of state estimation research published prior to 2016) is the idea of a pipeline. A pipeline consists several distinguishable blocks that have interpretable inputs and outputs. By quantifying in-

formation contained within sensor data, pipelines facilitate the construction of complex state estimation architectures that can fuse observations from sensors of varied modality to create rich models of the external world and infer the state of a mobile robot within it. My thesis focuses on egomotion estimation: the problem of accurately and consistently estimating the relative pose of a moving robot. For this task, a variety of different sensors may be useful (e.g., lidar, stereo cameras, or inertial measurement units), and each may allow for various components of a state estimation pipeline. For cameras that process visual data, egomotion estimation is referred to as visual odometry or VO and a typical VO pipeline is illustrated in Figure 1.4.

These types visual egomotion estimation pipelines (Leutenegger et al., 2015; Cvišić and Petrović, 2015; Tsotsos et al., 2015) have achieved impressive localization accuracy on trajectories spanning several kilometres by carefully extracting and tracking sparse visual features (using *hand-crafted* algorithms) across consecutive images. Simultaneously, significant effort has gone to developing localization pipelines that eschew sparse features in favour of *dense* visual data (Alcantarilla and Woodford, 2016; Forster et al., 2014), **ToDo: Add some modern citations here** typically relying on loss functions that use direct pixel intensities.

However, in the last several years, a significant part of the state estimation literature has focused on the idea of replacing classical pipelines with parametric modelling through deep convolutional neural networks (CNNs) and data-driven training. Although initially developed for image classification (LeCun et al., 2015), CNN-based measurement models have been applied to numerous problems in geometric state estimation (e.g., homography estimation (DeTone et al., 2016), single image depth reconstruction (Garg et al., 2016), camera relocalization (Kendall and Cipolla, 2016), place recognition (Sünderhauf et al., 2015)). A number of recent CNN-based approaches have also tackled the problem of egomotion estimation, often purporting to obviate the need for classical visual localization pipelines by learning pose changes *end-to-end*, directly from image data (e.g., Melekhov et al. (2017), Handa et al. (2016), Oliveira et al. (2017)).

Despite this surge of excitement, significant debate has emerged within the robotics and computer vision communities regarding the extent to which deep models should replace existing geometric state estimation algorithms. Owing to their representational power, deep models may move the onerous task of selecting 'good' (i.e., robust to environmental vagaries and sensor motion) visual features from the roboticist to the learned model. By design, deep models also provide a straight-forward formulation for using *dense* data while being flexible in their loss function, and taking full advantage of modern computing architecture to minimize run time. Despite these potential benefits, current deep regression techniques for state estimation often generalize poorly to new environments, come with few analytical guarantees,

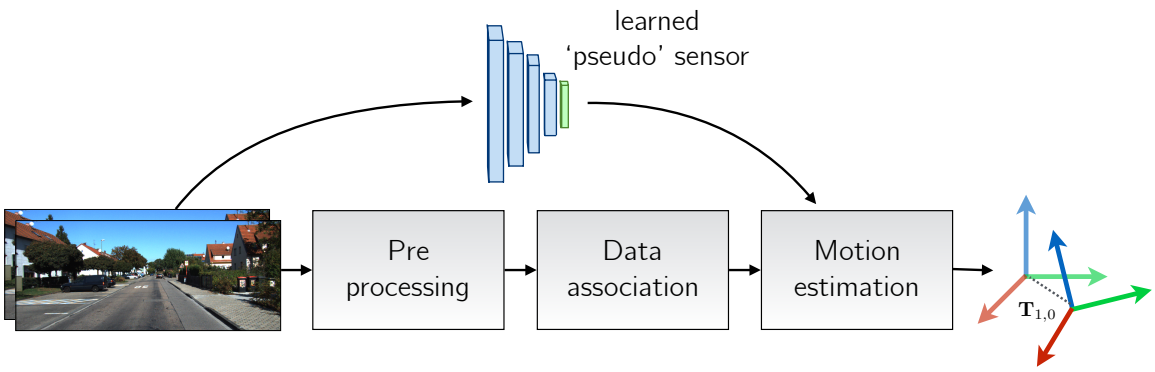


Figure 1.5: A learned *pseudo-sensor* extracts latent information from the same data stream.

and provide only point estimates of latent parameters.

1.4 Pipelines vs. Parametric Learning

Table 1.1: The benefits and downsides to using learning.

	Classical Pipelines	**Parametric Learning**
Maturity	Decades of literature & domain knowledge	Nascent in autonomy
Interpretability	Good, each component has interpretable input and output	Poor, often with no interpretable intermediate outputs
Uncertainty	Foundational to probabilistic robotics	Some methods (bootstrap, BCNNs)
Robustness	Relatively good	Highly dependant on training data
Flexibility	Limited by components	Limited by training data

There is new evidence that separating different tasks into interpretable components (e.g., optical flow, scene segmentation) works better than end-to-end learning for action (Zhou et al., 2019).

1.5 The Learned Pseudo-Sensor

To retain the benefits of classical state estimation pipelines while leveraging the representational power of modern deep parametric networks, I suggest the paradigm of the *learned pseudo-sensor*. Classical pipelines have shown to be relatively robust across numerous types of environments. Instead of completely replacing them, my thesis presents several ways in which machine learning techniques can be used to

1.6 Original Contributions

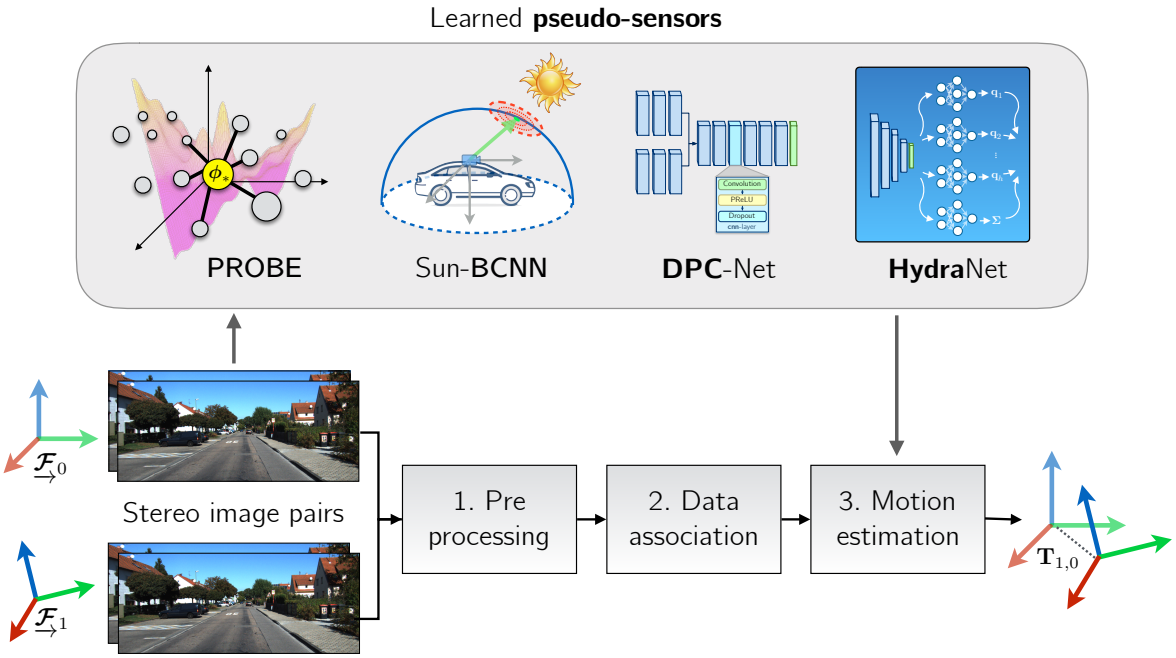


Figure 1.6: My thesis presents four different types of pseudo-sensors to improve visual odometry: PROBE, Sun-BCNN, DPC-Net, and HydraNet.

1.6.1 Publications

1. **Predictive Robust Estimation for Sparse Visual Odometry**
   Predictive Robust Estimation (PROBE) is a technique that uses k-NN regression (original PROBE) or Generalized Kernels (Vega-Brown et al., 2014) (PROBE-GK) to train a predictive model for heteroscedastic measurement covariance to improve estimator accuracy and consistency.

- Peretroukhin, V., Clement, L., Giamou, M., and Kelly, J. (2015). PROBE: Predictive robust estimation for visual-inertial navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'15)*, pages 3668–3675, Hamburg, Germany

- Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 817–824

2. **Virtual Sun Sensor using a Bayesian Convolutional Neural Network**
Sun-BCNN is a technique to infer a probabilistic estimate of the direction of the sun from a single RGB image using a Bayesian Convolutional Neural Networks (BCNN). The method works much like dedicated sun sensors (Lambert et al., 2012), but requires no additional hardware, and can provide mean and covariance estimates that can be readily incorporated into existing visual odometry frameworks. I worked on this project in collaboration with Lee Clement. While he focussed on integrating Sun-BCNN into the visual estimator, I developed the BCNN architecture and focused on uncertainty modelling. Initial exploratory work was published at ISER 2016, and the BCNN improvement was presented at ICRA 2017. An additional journal paper summarizing the work of the prior two papers, adding data from the Canadian High Arctic and Oxford, and investigating the effect of cloud cover and transfer learning was published in the International Journal of Robotics' Research, Special Issue on Experimental Robotics at the end of 2017.

   - Clement, L., Peretroukhin, V., and Kelly, J. (2017). Improving the accuracy of stereo visual odometry using visual illumination estimation. In Kulic, D., Nakamura, Y., Khatib, O., and Venture, G., editors, *2016 International Symposium on Experimental Robotics*, volume 1 of *Springer Proceedings in Advanced Robotics*, pages 409–419. Springer International Publishing, Berlin Heidelberg. Invited to Journal Special Issue

   - Peretroukhin, V., Clement, L., and Kelly, J. (2017). Reducing drift in visual odometry by inferring sun direction using a bayesian convolutional neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'17)*, pages 2035–2042, Singapore

   - Peretroukhin, V., Clement, L., and Kelly, J. (2018). Inferring sun direction to improve visual odometry: A deep learning approach. *International Journal of Robotics Research*

3. **Deep Pose Corrections (DPC-Net)**
   Deep Pose Correction is a novel approach to improving egomotion estimates through pose corrections learned through deep regression. DPC takes as its starting point an efficient, classical localization algorithm that computes high-rate pose estimates. To it, it adds a Deep Pose Correction Network (DPC-Net) that learns low-rate, 'small' *corrections* from training data that are then fused with the original estimates. DPC-Net does not require any modification to an existing localization pipeline, and can learn to correct multi-faceted errors from estimator bias, sensor mis-calibration or environmental effects.

   - Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*

4. **Probabilistic Inference of Elements of SO(3)**

   - Peretroukhin, V., Clement, L., and Kelly, J. (2018). Inferring sun direction to improve visual odometry: A deep learning approach. *International Journal of Robotics Research*

1.6.2 Software Contributions

- DPC-Net
- SO(3) Learning
- `liegroups`
- pyslam

Chapter 2

Mathematical Foundations

> By relieving the brain of all unnecessary work, a good notation sets it free to concentrate on more advanced problems, and, in effect, increases the mental power of the race.
>
> Alfred North Whitehead

2.1 3D Notation

In this thesis, we largely follow the notation of Barfoot (2017) when dealing with three-dimensional rigid-body kinematics. As it is crucial to all four components of the work, we will begin by outlining the basic components.

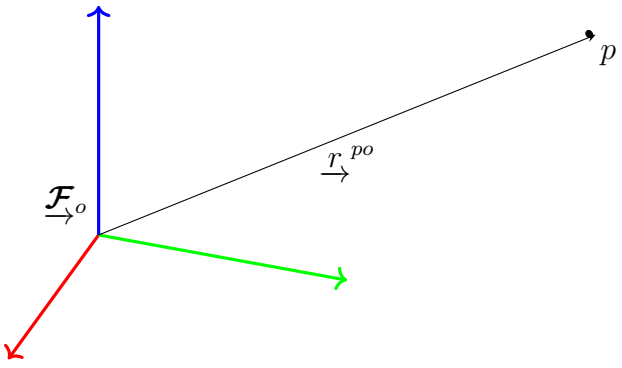


Figure 2.1: A position vector expressed in a coordinate frame.

We refer to a three-dimensional position vector, $\underrightarrow{r}^{po}$, as one that originates at the origin of a coordinate reference frame, $\underrightarrow{\mathcal{F}}_o$, and terminates at the point p. This geometric quantity has

the numerical coordinates $\mathbf{r}_o^{po}$ when expressed in $\underrightarrow{\mathcal{F}}_o$. Often, we will refer to two reference frames such as a world or *inertial* frame, $\underrightarrow{\mathcal{F}}_i$, and a vehicle frame, $\underrightarrow{\mathcal{F}}_v$. Rotation matrices or rigid-body transformations that convert coordinates from $\underrightarrow{\mathcal{F}}_i$ to $\underrightarrow{\mathcal{F}}_v$ will be represented as $\mathbf{T}_{vi}$, and $\mathbf{C}_{vi}$[1], respectively.

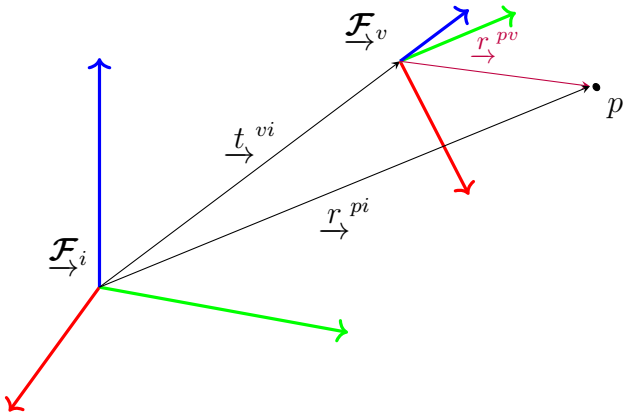


Figure 2.2: Two common references frames used throughout this thesis.

2.2 Rotations

The rotation matrix $\mathbf{C}$ is a member of the matrix Lie group $\mathrm{SO}(3)$ and can be defined as a matrix as follows:

$$\mathrm{SO}(3) = \{\mathbf{C} \in \mathbb{R}^{3\times 3} | \ \mathbf{C}^T\mathbf{C} = \mathbf{1}, \det \mathbf{C} = 1\}. \tag{2.1}$$

Active vs. Passive

An active (or alibi) rotation changes the coordinates of a position directly while implicitly assuming that the reference frame is fixed. A passive (or alias) rotation rotates the reference frame. Following Barfoot (2017), all rotation matrices in this thesis are passive unless otherwise noted.

Exponential and Logarithmic Maps

Since rotations form a matrix Lie group (we refer the reader to Solà et al. (2018) and Barfoot (2017) for more thorough presentation of Lie groups), we can define a surjective exponential

[1]We use $\mathbf{C}$ and not $\mathbf{R}$ for rotation matrices to avoid confusion with common notation for measurement model covariance.

map[2] from three axis-angle parameters, $\boldsymbol{\phi} = \phi\mathbf{a}, \;\; \phi \in \mathbb{R}, \mathbf{a} \in S^2$, to a rotation matrix, $\mathbf{C}$:

$$\mathbf{C} = \mathrm{Exp}\left(\boldsymbol{\phi}\right) = \exp\left(\boldsymbol{\phi}^{\wedge}\right) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\phi}^{\wedge})^n \tag{2.2}$$

$$= \cos\phi\mathbf{1} + (1 - \cos\phi)\mathbf{a}\mathbf{a}^T + \sin\phi\mathbf{a}^{\wedge}, \tag{2.3}$$

where the wedge operator $(\cdot)^{\wedge}$[3] is defined as

$$\mathbf{a}^{\wedge} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -a_2 & a_1 \\ a_2 & 0 & -a_0 \\ -a_1 & a_0 & 0 \end{bmatrix}. \tag{2.4}$$

Equation (2.3) is known as the Euler-Rodriguez formula and it can also be derived geometrically, starting from Euler's theorem that any rotation can be expressed as an axis of rotation and an angle of rotation about that axis. Although the map in Equation (2.2) is surjective, we can define an inverse map if we restrict its domain to $0 \leq \phi < \pi$:

$$\boldsymbol{\phi} = \mathrm{Log}\left(\mathbf{C}\right) = \log\left(\mathbf{C}\right)^{\vee} = \frac{\phi(\mathbf{C} - \mathbf{C}^T)^{\vee}}{2\sin\phi}, \tag{2.5}$$

where $\phi = \arccos\frac{\mathrm{tr}(\mathbf{C})-1}{2}$ and the *vee* operator, $(\cdot)^{\vee} : \mathbb{R}^{3\times3} \to \mathbb{R}^3$, is defined as the unique inverse of the wedge operator $(\cdot)^{\wedge}$. Note Equation (2.5) is undefined at both $\phi = 0$ and at $\phi = \pi$. In the former case, we can use a small-angle approximation and define

$$\mathrm{Log}\left(\mathbf{C}\right) \approx \left(\mathbf{C} - \mathbf{1}\right)^{\vee} \;\; \text{when} \;\; \phi \approx 0. \tag{2.6}$$

The latter case, (when $\phi = \pi$), defines the *cut locus* of the space where $\mathrm{Exp}\left(\cdot\right)$ is not a covering map and both $+\boldsymbol{\phi}$ and $-\boldsymbol{\phi}$ map to the same $\mathbf{C}$. This *cut locus* is related to the idea that any three parameterization of $\mathrm{SO}(3)$ will have singularities associated with it.

2.2.1 Unit Quaternions

Another way (and historically, the original way) to represent a general rotation is to use a unit quaternion, $\mathbf{q}$. A unit quaternion has four parameters, a scalar q_ω and a three-dimensional vector component, $\mathbf{q}_v$:

[2] We follow Solà et al. (2018) and also define *capitalized* map for notational clarity.

[3] This operator is often expressed as $(\cdot)^{\times}$ and is known as the skew-symmetric operator.

$$\mathbf{q} = \begin{bmatrix} q_\omega \\ \mathbf{q}_v \end{bmatrix} \in S^3, \quad (\|\mathbf{q}\| = 1). \tag{2.7}$$

Unit quaternions also form a Lie group (Solà et al., 2018) and and lie on a three-dimensional unit sphere within $\mathbb{R}^4$. This manifold represents a double cover of $\mathrm{SO}(3)$ (since both $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation). As with rotation matrices, we can define a surjective map from three parameters to the group itself,

$$\mathbf{q} = \mathrm{Exp}\left(\boldsymbol{\phi}\right) = \begin{bmatrix} \cos \phi/2 \\ \mathbf{a} \sin \phi/2 \end{bmatrix}. \tag{2.8}$$

Similarly, we can also define a logarithmic map,

$$\boldsymbol{\phi} = \mathrm{Log}\left(\mathbf{q}\right) = 2\mathbf{q}_v \frac{\arctan\left(\|\mathbf{q}_v\|, q_\omega\right)}{\|\mathbf{q}_v\|}. \tag{2.9}$$

To avoid issues with the double cover, we replace $\mathbf{q}$ with $-\mathbf{q}$ if q_ω is negative before evaluating Equation (2.9). Also note again that Equation (2.9) is undefined when $\phi = 0$, but, importantly, we do not face any issues when $\phi = \pi$ due to the half angle. As with rotation matrices, we can use small angle approximations to define:

$$\mathrm{Log}\left(\mathbf{q}\right) \approx \frac{\mathbf{q}_v}{q_\omega}\left(1 - \frac{\|\mathbf{q}_v\|^2}{3q_\omega^2}\right) \quad \text{when} \quad \phi \approx 0. \tag{2.10}$$

A fantastic summary of the history of rotation parameterizations, unit quaternions and the story of Hamilton and Rodriguez can be found in Altmann (1989).

2.3 Spatial Transforms

The rigid body transform $\mathbf{T}_{vi}$ is a also a member of the matrix Lie group, $\mathrm{SE}(3)$ and can be defined as a 4×4 matrix as follows:

$$\mathrm{SE}(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} | \ \mathbf{C} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3\}. \tag{2.11}$$

As a member of a matrix Lie group

$$\mathbf{T} = \exp\left(\boldsymbol{\xi}^\wedge\right) = \sum_{n=0}^{\infty} \frac{1}{n!} (\boldsymbol{\xi}^\wedge)^n \tag{2.12}$$

and the map $\mathbb{R}^6 \to \mathfrak{se}(3)$,

$$\boldsymbol{\xi}^\wedge \triangleq \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}. \tag{2.13}$$

The transform $\mathbf{T}_{vi}$ can be expressed as a 4×4 transformation matrix as

$$\mathbf{T}_{vi} = \begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_v^{iv} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{2.14}$$

where $\mathbf{C} \in \mathrm{SO}(3)$ is a rotation matrix. This allows us to use the homogenous point representation for $\mathbf{r}_i^{pi}$ and express the following relation:

$$\begin{bmatrix} \mathbf{r}_v^{pi} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_v^{iv} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{T}_{vi}} \begin{bmatrix} \mathbf{r}_i^{pi} \\ 1 \end{bmatrix} \tag{2.15}$$

This is equivalent to

$$\mathbf{r}_v^{pi} = \mathbf{C}_{vi}\mathbf{r}_i^{pi} + \mathbf{t}_v^{iv} \tag{2.16}$$

2.4 Perturbations

ToDo: Left,middle,right ToDo: Look at Seans thesis

2.5 Uncertainty

ToDo: Injection onto the manifold

Chapter 3

Classical Visual State Estimation

3.1 Visual Odometry Pipeline

The learned pseudo-sensors in this thesis are applied to a standard visual odometry pipeline largely based on the work of Furgale (2011). We briefly summarize the pipeline here, and outline any unique design choices we made.

3.1.1 Pre processing

In the preprocessing stage, stereo images are undistorted and rectified such that their principal axes are parallel, see Figure 3.2. We assume the stereo camera intrinsic and extrinsic parameters as well as the lens properties are known a priori (or computed through a calibration process like that detailed in Kelly (2011)).

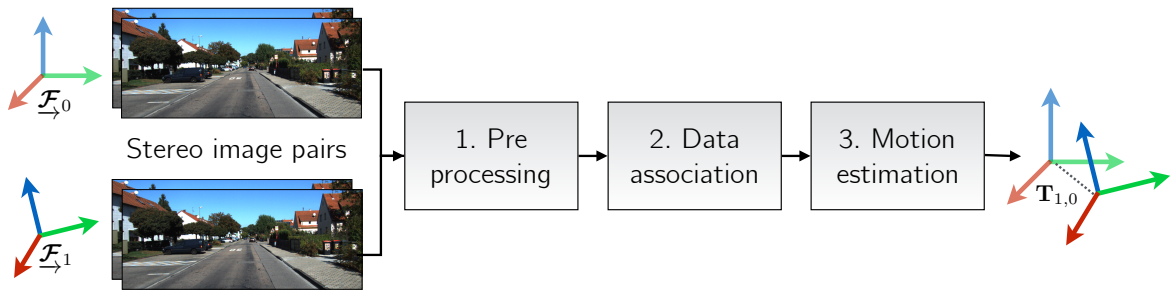


Figure 3.1: A 'classical' stereo visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.

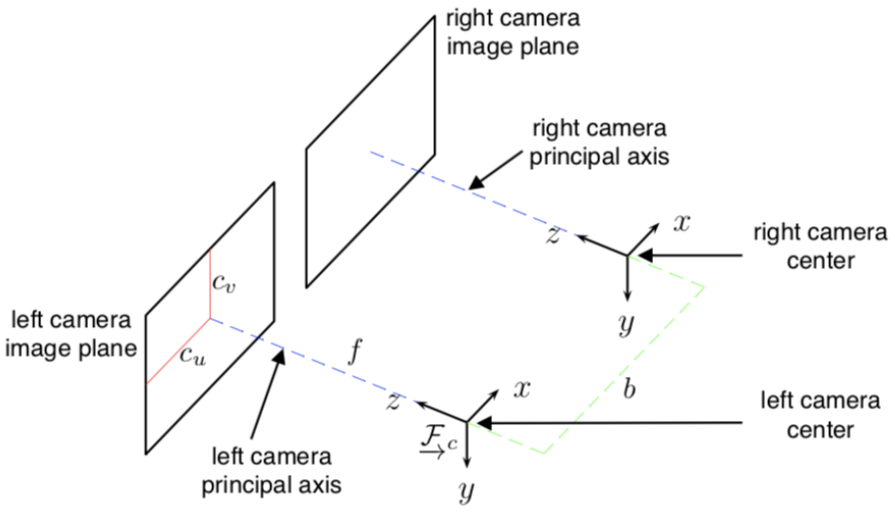


Figure 3.2: A rectified stereo camera with equal focal lengths. Taken Furgale (2011). **ToDo: reproduce myself**.

3.1.2 Data association

Feature extraction and matching

During data association, some form of image features must be matched. In this thesis, we focus on sparse stereo visual odometry which uses the `libviso2` image feature extraction and matching mechanism.

Each landmark corresponds to a point in space, expressed in homogeneous coordinates in the camera frame as $\boldsymbol{p}_{i,t} := \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix}^T \in \mathbb{P}^3$. The stereo-camera model, f, projects a landmark expressed in homogeneous coordinates into image space, so that $\mathbf{y}_{i,t}$, the stereo pixel coordinates of landmark i in the first camera pose at time t, is given by

$$\mathbf{y}_{i,t} = \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} = f(\boldsymbol{p}_{i,t}) = \mathbf{M}\frac{1}{p_3}\boldsymbol{p}_{i,t}, \tag{3.1}$$

where

$$\mathbf{M} = \begin{bmatrix} f_u & 0 & c_u & f_u \frac{b}{2} \\ 0 & f_v & c_v & 0 \\ f_u & 0 & c_u & -f_u \frac{b}{2} \\ 0 & f_b & c_v & 0 \end{bmatrix}. \tag{3.2}$$

Here, $\{c_u, c_v\}$, $\{f_u, f_v\}$, and b are the principal points, focal lengths and baseline of the stereo camera respectively. Note that in this formulation, the stereo camera frame is centered between the two individual lenses.

Outlier rejection

Brief description of RANSAC and Horn's method.

3.1.3 Motion solution

In our frame-to-frame sparse stereo odometry pipeline, the objective is to find $\mathbf{T}_t \in \mathrm{SE}(3)$, the rigid transform between two subsequent stereo camera poses (note that the temporal index t refers to the set of two stereo camera poses). We begin by rectifying, then stereo and temporally matching the set of 4 images to generate the corresponding locations of a set of N_t visual landmarks in each stereo pair.

We triangulate landmarks in the first camera frame, $\mathbf{y}_{i,t}$, and re-project them into the second frame, $\mathbf{y}'_{i,t}$. We model errors due to sensor noise and quantization as a Gaussian distribution in image space with a known covariance $\mathbf{R}$,

$$p(\mathbf{y}'_{i,t}|\mathbf{y}_{i,t}, \mathbf{T}_t, \mathbf{R}) = \mathcal{N}\left(\mathbf{e}_{i,t}(\mathbf{T}_t); \mathbf{0}, \mathbf{R}\right), \tag{3.3}$$

where

$$\mathbf{e}_{i,t} = \mathbf{y}'_{i,t} - f(\mathbf{T}_t f^{-1}(\mathbf{y}_{i,t})). \tag{3.4}$$

The maximum likelihood transform, $\mathbf{T}^*_t$, is then given by

$$\mathbf{T}^*_t = \underset{\mathbf{T}_t \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} {\mathbf{e}_{i,t}}^T \mathbf{R}^{-1} \mathbf{e}_{i,t}. \tag{3.5}$$

This is a nonlinear least squares problem, and can be solved iteratively using standard techniques. During iteration n, we represent the transform as the product of an estimate $\mathbf{T}^{(n)} \in \mathrm{SE}(3)$ and a perturbation $\delta\boldsymbol{\xi} \in \mathbb{R}^6$ represented in exponential coordinates:

$$\mathbf{T}_t = \exp\left(\delta\boldsymbol{\xi}^\wedge\right)\mathbf{T}^{(n)}_t. \tag{3.6}$$

Linearizing the transform for small perturbations $\delta\boldsymbol{\xi}$ yields a linear least-squares problem:

$$\mathcal{L}(\delta\boldsymbol{\xi}) = \frac{1}{2}\sum_{i=1}^{N_t}\left(\mathbf{e}_{i,t}^{(n)} - \mathbf{J}_{i,t}^{(n)}\delta\boldsymbol{\xi}\right)^T \mathbf{R}^{-1}\left(\mathbf{e}_{i,t}^{(n)} - \mathbf{J}_{i,t}^{(n)}\delta\boldsymbol{\xi}\right) \tag{3.7}$$

Here, $\mathbf{J}_{i,t}^{(n)}$ is the Jacobian matrix of the reprojection error. The explicit form of the Jacobian matrix is omitted for brevity but can be found in our supplemental materials.

Rearranging, we see the minimizing perturbation is the solution to a linear system of equations:

$$\delta\boldsymbol{\xi}^{(n)} = \left(\sum_{i=1}^{N_t}\mathbf{J}_{i,t}^T\mathbf{R}_{,}^{-1}\mathbf{J}_{i,t}\right)^{-1}\sum_{i=1}^{N_t}\mathbf{J}_{i,t}^T\mathbf{R}^{-1}\mathbf{e}_{i,t}^{(n)}. \tag{3.8}$$

We then update the estimated transform and proceed to the next iteration.

$$\mathbf{T}_t^{(n+1)} = \exp\left(\delta\boldsymbol{\xi}^{(n)\wedge}\right)\mathbf{T}_t^{(n)}. \tag{3.9}$$

There are many reasonable choices for both the initial transform $\mathbf{T}_t^{(0)}$ and for the conditions under which we terminate iteration. We initialize the estimated transform to identity, and iteratively perform the update given by eq. (3.9) until we see a relative change in the squared error of less than one percent after an update.

3.2 Pose Graph Relaxation

we fused our probabilistic rotation regression with classical stereo visual odometry using pose graph relaxation implemented with the help of a Python-based factor graph library which we will publicize after the review process. Using that framework, we solved

$$\mathbf{T}_{1,w}^*, \mathbf{T}_{2,w}^* = \underset{\mathbf{T}_{1,w},\mathbf{T}_{2,w}\in\mathrm{SE}(3)}{\operatorname{argmin}} \mathcal{L}(\hat{\mathbf{T}}_{2,1}, \hat{\mathbf{C}}_{2,1}) \tag{3.10}$$

$$= \underset{\mathbf{T}_{1,w},\mathbf{T}_{2,w}\in\mathrm{SE}(3)}{\operatorname{argmin}} \boldsymbol{\xi}_{1,2}^T\boldsymbol{\Sigma}_{\text{vo}}^{-1}\boldsymbol{\xi}_{1,2} + \boldsymbol{\phi}_{1,2}^T\boldsymbol{\Sigma}_{\text{hn}}^{-1}\boldsymbol{\phi}_{1,2} \tag{3.11}$$

where

$$\boldsymbol{\xi}_{1,2} = \operatorname{Log}\left(\left(\mathbf{T}_{2,w}\mathbf{T}_{1,w}^{-1}\right)\hat{\mathbf{T}}_{2,1}^{-1}\right), \tag{3.12}$$

$$\boldsymbol{\phi}_{1,2} = \operatorname{Log}\left(\left(\mathbf{C}_{2,w}\mathbf{C}_{1,w}^T\right)\hat{\mathbf{C}}_{2,1}^T\right), \tag{3.13}$$

and $\hat{\mathbf{T}}_{2,1}$, $\boldsymbol{\Sigma}_{\text{vo}}$ and $\hat{\mathbf{C}}_{2,1}$, $\boldsymbol{\Sigma}_{\text{hn}}$ were provided by our classical estimator and the HydraNet network respectively. Note that $\boldsymbol{\Sigma}_{\text{hn}} \in \mathbb{R}^{3\times3} \geq 0$ while $\boldsymbol{\Sigma}_{\text{vo}} \in \mathbb{R}^{6\times6} \geq 0$. We also overload the

logarithm function, $\mathrm{Log}\,(\cdot)$ to represent both $\mathrm{SE}(3)$ and $\mathrm{SO}(3)$ logarithmic maps as necessary. To account for gauge freedom, we fixed the first transformation to identity, $\mathbf{T}_{1,w} = \mathbf{1}$, and initialized $\mathbf{T}_{2,w}$ to $\hat{\mathbf{T}}_{2,1}$. After convergence, we composed the final frame-to-frame estimate as $\mathbf{T}^*_{2,1} = \mathbf{T}^*_{2,w}\left(\mathbf{T}^*_{1,w}\right)^{-1} = \mathbf{T}^*_{2,w}$.

3.3 Robust Estimation

3.3.1 Removing outliers

3.3.2 Robust M-Estimation

Since the standard L_2 cost function (**??**) assigns cost values that grow quadratically with measurement error, it is very sensitive to outlier measurements. A common solution to this problem is to replace the L_2 cost function with one that is less sensitive to large measurement errors. These robust cost functions are collectively known as M-estimators, and many variants exist. Here, we consider three M-estimators: the Huber (**??**), Tukey (**??**), and "Fair" (**??**) cost functions.

Optimization using robust cost functions such as these typically proceeds using an Iteratively Reweighted Least-Squares (IRLS) procedure in which we define a weight for the i^{th} measurement as

$$w_i = \frac{1}{e_i}\frac{\partial \rho(e_i)}{\partial e_i} \tag{3.14}$$

and repeatedly minimize the weighted least-squares objective

$$\mathcal{O} = \sum_i w_i e_i^2, \tag{3.15}$$

updating the weights at each iteration until convergence.

Appendices

Appendix A

Left and Middle Perturbations

A.1 Identities

$$\operatorname{Exp}\left(\left(\boldsymbol{\xi}+\delta\boldsymbol{\xi}\right)\right) \approx \operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}\delta\boldsymbol{\xi}\right)\right)\operatorname{Exp}\left(\boldsymbol{\xi}\right), \tag{A.1}$$

$$\begin{aligned}\log\left(\mathbf{T}_1\mathbf{T}_2\right)^{\vee} &= \log\left(\operatorname{Exp}\left(\boldsymbol{\xi}_1\right)\operatorname{Exp}\left(\boldsymbol{\xi}_2\right)\right)^{\vee} \\ &\approx \begin{cases} \boldsymbol{\mathcal{J}}(\boldsymbol{\xi}_2)^{-1}\boldsymbol{\xi}_1+\boldsymbol{\xi}_2 & \text{if } \boldsymbol{\xi}_1 \text{ small} \\ \boldsymbol{\xi}_1+\boldsymbol{\mathcal{J}}(-\boldsymbol{\xi}_1)^{-1}\boldsymbol{\xi}_2 & \text{if } \boldsymbol{\xi}_2 \text{ small.} \end{cases}\end{aligned} \tag{A.2}$$

A.2 Perturbing $\mathrm{SE}(3)$

Consider the quantity,

$$\mathbf{T}_{ba} = \mathbf{T}_{bi}{\mathbf{T}_{ai}}^{-1} \tag{A.3}$$

where $\underrightarrow{\mathcal{F}}_i$ is some inertial frame.

A.2.1 Left Perturbation

Separating $\mathbf{T}_{ba}$ into a mean component, $\overline{\mathbf{T}}_{ba}$, and a small left perturbation,

$$\mathbf{T}_{ba} = \operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)\overline{\mathbf{T}}_{ba} = \operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba}\right) \tag{A.4}$$

Applying a logarithm to both sides,

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} = \log\left(\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba}\right)\right)^{\vee} \tag{A.5}$$

Using Equation (A.2),

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} \approx \overline{\boldsymbol{\xi}}_{ba} + \boldsymbol{\mathcal{J}}_{ba}^{-1}\delta\boldsymbol{\xi}_{ba}^{l} \tag{A.6}$$

where $\boldsymbol{\mathcal{J}}_{ba} \triangleq \boldsymbol{\mathcal{J}}(\overline{\boldsymbol{\xi}}_{ba})$. This is exactly Equation 6.104 in Sean Anderson's thesis.

A.2.2 Middle Perturbation

Now consider the middle perturbation,

$$\mathbf{T}_{ba} = \operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba} + \delta\boldsymbol{\xi}_{ba}^{m}\right) \tag{A.7}$$

Immediately, we can take the logarithm of both sides and see that,

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} = \overline{\boldsymbol{\xi}}_{ba} + \delta\boldsymbol{\xi}_{ba}^{m}, \tag{A.8}$$

where we now observe that $\delta\boldsymbol{\xi}_{ba}^{l} \approx \boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}$.

A.3 DPC SE(3) Loss

Using the notation in this document, the DPC derivation requires an expression for $\delta\boldsymbol{\xi}_b$, and assumes that $\boldsymbol{\xi}_a$ is constant.

A.3.1 Middle Perturbation

Consider,

$$\mathbf{T}_{ba} = \mathbf{T}_b\mathbf{T}_a{}^{-1} \tag{A.9}$$

where we have dropped the i frame for clarity. Middle perturbing $\mathbf{T}_{ba}$ and $\mathbf{T}_b$ and keeping $\mathbf{T}_a$ fixed (i.e., $\mathbf{T}_a = \overline{\mathbf{T}}_a$).

$$\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba} + \delta\boldsymbol{\xi}_{ba}^{m}\right) = \operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{b} + \delta\boldsymbol{\xi}_{b}^{m}\right)\overline{\mathbf{T}}_a{}^{-1} \tag{A.10}$$

Using Equation (A.1) twice,

$$\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}\right)\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba}\right) = \operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{b}\right)\overline{\mathbf{T}}_a{}^{-1} \tag{A.11}$$

Collecting terms, we have

$$\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}\right)\right)\overline{\mathbf{T}}_{ba}=\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right)\overline{\mathbf{T}}_{b}\overline{\mathbf{T}}_{a}{}^{-1} \tag{A.12}$$

Right multiplying by $\overline{\mathbf{T}}_{ba}^{-1}$, we are left with

$$\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}\right)\right)=\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right) \tag{A.13}$$

or

$$\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}=\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}. \tag{A.14}$$

Solving for $\delta\boldsymbol{\xi}_{ba}^{m}$,

$$\delta\boldsymbol{\xi}_{ba}^{m}=\boldsymbol{\mathcal{J}}_{ba}^{-1}\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}. \tag{A.15}$$

Now inserting Equation (A.15) into Equation (A.8),

$$\log\left(\mathbf{T}_{ba}\right)^{\vee}\approx\overline{\boldsymbol{\xi}}_{ba}+\boldsymbol{\mathcal{J}}_{ba}^{-1}\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m} \tag{A.16}$$

This is exactly Equation 13 in the DPC-Net paper:

$$g(\boldsymbol{\xi}+\delta\boldsymbol{\xi})\approx\boldsymbol{\mathcal{J}}(g(\boldsymbol{\xi}))^{-1}\boldsymbol{\mathcal{J}}(\boldsymbol{\xi})\delta\boldsymbol{\xi}+g(\boldsymbol{\xi}). \tag{A.17}$$

A.3.2 Left Perturbation

Using the left perturbation, we can repeat the procedure of relating $\delta\boldsymbol{\xi}_{ba}^{l}$ and $\delta\boldsymbol{\xi}_{b}^{l}$ (by perturbing $\mathbf{T}_{ba}$ and $\mathbf{T}_{b}$ and keeping $\mathbf{T}_{a}$ fixed (i.e., $\mathbf{T}_{a}=\overline{\mathbf{T}}_{a}$).

$$\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)\overline{\mathbf{T}}_{ba}=\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)\overline{\mathbf{T}}_{b}\overline{\mathbf{T}}_{a}{}^{-1} \tag{A.18}$$

from which we see immediately that $\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)=\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)$ and therefore,

$$\delta\boldsymbol{\xi}_{ba}^{l}=\delta\boldsymbol{\xi}_{b}^{l} \tag{A.19}$$

Now using Equation (A.6), we have

$$\log\left(\mathbf{T}_{ba}\right)^{\vee}\approx\overline{\boldsymbol{\xi}}_{ba}+\boldsymbol{\mathcal{J}}_{ba}^{-1}\delta\boldsymbol{\xi}_{b}^{l} \tag{A.20}$$

A.3.3 Summary

Using the left perturbation, we have

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} \approx \overline{\boldsymbol{\xi}}_{ba} + \boldsymbol{\mathcal{J}}_{ba}^{-1}\delta\boldsymbol{\xi}_{b}^{l} \tag{A.21}$$

Using the centre/middle perturbation, we have

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} \approx \overline{\boldsymbol{\xi}}_{ba} + \boldsymbol{\mathcal{J}}_{ba}^{-1}\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m} \tag{A.22}$$

And we see the same earlier expression relating left and middle perturbations,

$$\delta\boldsymbol{\xi}^{l} \approx \boldsymbol{\mathcal{J}}\delta\boldsymbol{\xi}^{m} \tag{A.23}$$

A.3.4 Reconciliation

Consider the two update rules:

$$\mathbf{T}_{b} \leftarrow \mathrm{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)\overline{\mathbf{T}}_{b} \tag{A.24}$$

$$\mathbf{T}_{b} \leftarrow \mathrm{Exp}\left(\left(\overline{\boldsymbol{\xi}}_{b} + \delta\boldsymbol{\xi}_{b}^{m}\right)\right) \tag{A.25}$$

But using Equation (A.1), the middle update becomes,

$$\mathrm{Exp}\left(\left(\overline{\boldsymbol{\xi}}_{b} + \delta\boldsymbol{\xi}_{b}^{m}\right)\right) \approx \mathrm{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right)\mathrm{Exp}\left(\overline{\boldsymbol{\xi}}_{b}\right) = \mathrm{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right)\overline{\mathbf{T}}_{b} = \mathrm{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)\overline{\mathbf{T}}_{b} \tag{A.26}$$

So the middle perturbation does not require us to keep the mean in the group (as long as we avoid any degeneracies).

Appendix B

More Notes on Rotation

B.1 Metrics on $\mathrm{SO}(3)$

The three different rotation metrics can be related to the angular (or geodesic) metric, d_{ang}, as follows,

$$
\begin{aligned}
d_{\mathrm{ang}}(\mathbf{R}_a, \mathbf{R}_b) &= \left\| \mathrm{Log}\left(\mathbf{R}_a \mathbf{R}_b^T\right) \right\|_2 && \text{(B.1)} \\
&= \theta, && \text{(B.2)} \\
d_{\mathrm{quat}}(\mathbf{q}_a, \mathbf{q}_b) &= \min\left(\left\| \mathbf{q}_a - \mathbf{q}_b \right\|_2, \left\| \mathbf{q}_a + \mathbf{q}_b \right\|_2 \right) && \text{(B.3)} \\
&= 2 \sin \frac{\theta}{4}, && \text{(B.4)} \\
d_{\mathrm{ang}}(\mathbf{R}_a, \mathbf{R}_b) &= \left\| \mathbf{R}_a - \mathbf{R}_b \right\|_{\mathrm{Frob}} && \text{(B.5)} \\
&= 2\sqrt{2} \sin \frac{\theta}{2}. && \text{(B.6)}
\end{aligned}
$$

Given a set of rotations parametrized by unit quaternions $\{\mathbf{q}_i\}_{i=1}^n$,

$$
\overline{\mathbf{q}} = \frac{\sum_{i=1}^n \mathbf{q}_i}{\left\| \sum_{i=1}^n \mathbf{q}_i \right\|}, \tag{B.7}
$$

solves

$$
\mathbf{q} = \underset{\mathbf{R}(\mathbf{q}) \in \mathrm{SO}(3)}{\mathrm{argmin}} \sum_{i=1}^n d_{\mathrm{quat}}(\mathbf{q}_i, \mathbf{q})^2, \tag{B.8}
$$

so long as $d_{\mathrm{ang}}(\mathbf{R}(\overline{\mathbf{q}}), \mathbf{R}(\mathbf{q}_i)) < \pi/2$. See **?** for more details.

B.2 Topology

The exponential map for SO(3) is not a covering map, it cannot represent rotations by π. These form the 'cut locus'.

A covering map is not a 'surjective map' but also requires 'each point in X has a neighbourhood that is the same after the mapping'.

To get $\mathrm{SO}(3)$, you take the ball given by $\|\phi\| \leq \pi$ and then 'glue together' antipodal points at the border $\|\phi\| = \pi$.

$\mathrm{SO}(3)$ is diffeomorphic to RP(3) 'real projective space' which can be made by 'identifying' antipodal points in S^3 (unit sphere in $\mathbb{R}^4$). The unit quaternion is simply S^3, which is why it is a double cover (since we have not identified $\mathbf{q}$ and $-\mathbf{q}$).

B.3 Antipodal Rotations

What is the geodesic distance of two 'antipodal' SO(3) elements $\mathbf{C}_1$, $\mathbf{C}_2$ (i.e., $\mathbf{C}_1 = \mathbf{C}(\pi)\mathbf{C}_2$)?

Consider $\mathrm{Log}\left(\mathbf{C}_1\mathbf{C}_2^T\right) = \mathrm{Log}\left(\mathbf{C}(\pi)\mathbf{C}_2\mathbf{C}_2^T\right) = \mathrm{Log}\left(\mathbf{C}(\pi)\right) =$ undefined? ($\pi\hat{\mathbf{n}}$ or $-\pi\hat{\mathbf{n}}$?)

We might be tempted to use the following logic:

$\mathbf{C}(\pi) = \mathbf{C}(-\pi) = \mathbf{C}(\pi)^T$, so $\mathbf{C}(\pi)\mathbf{C}(\pi) = \mathbf{1}$. Given this,

$$\begin{aligned}
\mathbf{0} &= \mathrm{Log}\left(\mathbf{1}\right) && \text{(B.9)}\\
&= \mathrm{Log}\left(\mathbf{C}(\pi)\mathbf{C}(\pi)\right) && \text{(B.10)}\\
&\text{(since } \mathbf{C}(\pi) \text{ commutes with itself)} && \text{(B.11)}\\
&= \mathrm{Log}\left(\mathbf{C}(\pi)\right) + \mathrm{Log}\left(\mathbf{C}(\pi)\right) && \text{(B.12)}\\
&= 2\mathrm{Log}\left(\mathbf{C}(\pi)\right) && \text{(B.13)}
\end{aligned}$$

. So $\mathrm{Log}\left(\mathbf{C}(\pi)\right) = \mathbf{0}$! But this doesn't make sense. Clearly $\mathbf{C}_1$, $\mathbf{C}_2$ are not the same elements - how can their geodesic distance (given by $\left\|\mathrm{Log}\left(\mathbf{C}_1\mathbf{C}_2^T\right)\right\|$) be 0?

I believe this is resolved by realizing that $\mathrm{Log}\left(\mathbf{C}(\pi)\right)$ cannot be defined in $\mathbb{R}^3$ since its magnitude is not 0, but $\mathrm{Log}\left(\mathbf{C}(\pi)\right) + \mathrm{Log}\left(\mathbf{C}(\pi)\right) = \mathbf{0}$. Hand waving, if we define $\mathrm{Log}\left(\mathbf{C}(\pi)\right)$ up to a sign ambiguity, we have:

$$\mathrm{Log}\left(\mathbf{C}(\pi)\right) + \mathrm{Log}\left(\mathbf{C}(\pi)\right) = \pm\pi\hat{\mathbf{n}} \mp \pi\hat{\mathbf{n}} = \mathbf{0} \tag{B.14}$$

$$\left\|\mathrm{Log}\left(\mathbf{C}_1\mathbf{C}_2^T\right)\right\| = \|\mathrm{Log}\left(\mathbf{C}(\pi)\right)\| \tag{B.15}$$

$$= \|\pm\pi\hat{\mathbf{n}}\| \tag{B.16}$$

$$= \pi! \tag{B.17}$$

Without loss of generality, consider $\mathbf{C}_2 = \mathbf{1}$, $\mathbf{C}_1 = \mathbf{C}(\pi)$.
Then using the Barfoot noise injection scheme,

$$\mathbf{C} = \mathrm{Exp}\left(\boldsymbol{\phi}\right)\overline{\mathbf{C}}, \qquad \boldsymbol{\phi} \sim \mathcal{N}\mathbf{0}\boldsymbol{\Sigma} \tag{B.18}$$

If we set $\mathbf{C}_2$ to be the mean, we can still produce $\mathbf{C}_1$ as a sample since,

$$\mathbf{C}_1 = \mathrm{Exp}\left(\pm\pi\hat{\mathbf{n}}\right)\mathbf{C}_2 = \mathrm{Exp}\left(\pm\pi\hat{\mathbf{n}}\right) \tag{B.19}$$

However, to use this in a loss function, we need to compute the likelihood of $\mathbf{C}_1$ given the mean $\mathbf{C}_2$. If $\boldsymbol{\Sigma} = \mathbf{1}$ then the negative log likelihood is equal to π^2 since , but in general this involves the Mahalanobis distance. Can we compute the Mahalonobis distance without explicitly evaluating the components of the log map (since they involve sign ambiguities)?

Appendix C

Representations of $\mathrm{SE}(3)$

ToDo: Discuss the banana uncertainty (cannot represent uncertainty over SE(3) with certain position but uncertain rotation)
ToDo: Discuss SO(3) x R3 vs SE(3) vs (SO(3), R3)

Bibliography

Alcantarilla, P. F. and Woodford, O. J. (2016). Noise models in feature-based stereo visual odometry.

Altmann, S. L. (1989). Hamilton, rodrigues, and the quaternion scandal. *Math. Mag.*, 62(5):291–308.

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press.

Clement, L., Peretroukhin, V., and Kelly, J. (2017). Improving the accuracy of stereo visual odometry using visual illumination estimation. In Kulic, D., Nakamura, Y., Khatib, O., and Venture, G., editors, *2016 International Symposium on Experimental Robotics*, volume 1 of *Springer Proceedings in Advanced Robotics*, pages 409–419. Springer International Publishing, Berlin Heidelberg. Invited to Journal Special Issue.

Cvišić, I. and Petrović, I. (2015). Stereo odometry based on careful feature selection and tracking. In *Proc. European Conf. on Mobile Robots (ECMR)*, pages 1–6.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2016). Deep image homography estimation.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 15–22.

Furgale, P. (2011). *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. PhD thesis.

Garg, R., Carneiro, G., and Reid, I. (2016). Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conf. on Comp. Vision*, pages 740–756. Springer.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.*, 32(11):1231–1237.

Grewal, M. S. and Andrews, A. P. (2010). Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Syst. Mag.*, 30(3):69–78.

Handa, A., Bloesch, M., Pătrăucean, V., Stent, S., McCormac, J., and Davison, A. (2016). gvnn: Neural network library for geometric computer vision. In *Computer Vision – ECCV 2016 Workshops*, pages 67–82. Springer, Cham.

Kelly, J. (2011). *On Temporal and Spatial Calibration for High Accuracy Visual-Inertial Motion Estimation*. PhD thesis, University of Southern California, Los Angeles, California, USA.

Kendall, A. and Cipolla, R. (2016). Modelling uncertainty in deep learning for camera relocalization. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 4762–4769.

Lambert, A., Furgale, P., Barfoot, T. D., and Enright, J. (2012). Field testing of visual odometry aided by a sun sensor and inclinometer. *J. Field Robot.*, 29(3):426–444.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *Int. J. Rob. Res.*, 34(3):314–334.

Melekhov, I., Ylioinas, J., Kannala, J., and Rahtu, E. (2017). Relative camera pose estimation using convolutional neural networks. In *Proc. Int. Conf. on Advanced Concepts for Intel. Vision Syst.*, pages 675–687. Springer.

Nilsson, N. J. (1984). Shakey the robot. Technical report, SRI INTERNATIONAL MENLO PARK CA.

Oliveira, G. L., Radwan, N., Burgard, W., and Brox, T. (2017). Topometric localization with deep learning.

Peretroukhin, V., Clement, L., Giamou, M., and Kelly, J. (2015). PROBE: Predictive robust estimation for visual-inertial navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'15)*, pages 3668–3675, Hamburg, Germany.

Peretroukhin, V., Clement, L., and Kelly, J. (2017). Reducing drift in visual odometry by inferring sun direction using a bayesian convolutional neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'17)*, pages 2035–2042, Singapore.

Peretroukhin, V., Clement, L., and Kelly, J. (2018). Inferring sun direction to improve visual odometry: A deep learning approach. *International Journal of Robotics Research*.

Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*.

Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 817–824.

Redfield, S. (2019). A definition for robotics as an academic discipline. *Nature Machine Intelligence*, 1(6):263–264.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry [tutorial]. *IEEE Robot. Autom. Mag.*, 18(4):80–92.

Solà, J., Deray, J., and Atchuthan, D. (2018). A micro lie theory for state estimation in robotics.

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M. (2015). On the performance of ConvNet features for place recognition. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, pages 4297–4304.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the DARPA grand challenge. *J. Field Robotics*, 23(9):661–692.

Tsotsos, K., Chiuso, A., and Soatto, S. (2015). Robust inference for visual-inertial sensor fusion. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5203–5210.

Vega-Brown, W. R., Doniec, M., and Roy, N. G. (2014). Nonparametric Bayesian inference on multivariate exponential families. In *Proc. Advances in Neural Information Proc. Syst. (NIPS) 27*, pages 2546–2554.

Zhou, B., Krähenbühl, P., and Koltun, V. (2019). Does computer vision matter for action?