On learning pseudo-sensors to improve visual egomotion estimation

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 learning pseudo-sensors to improve visual egomotion estimation

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

The ability to estimate *egomotion* 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 information may be intermittent or unavailable. Visual egomotion estimation, also known as *visual odometry*, has become ubiquitous in mobile robotics due to the availability of high-quality, compact, and inexpensive cameras that capture rich representations of the world. To remain computationally tractable, 'classical' visual odometry pipelines make simplifying assumptions that, while permitting reliable operation in ideal conditions, often lead to systematic error. In this dissertation, we present several data-driven *pseudo-sensors* that serve to augment conventional pipelines by inferring latent information from sensor data. Our approach retains many of the benefits of traditional pipelines, while leveraging high-capacity hyper-parametric models to extract complementary information that can be used to improve uncertainty quantification, correct for systematic bias, and improve robustness to difficult-to-model deleterious effects. We validate our pseudo-sensors on several kilometres of sensor data collected in sundry settings such as urban roads, indoor labs, and planetary analogue sites in the Canadian 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

The universe is no narrow thing and the order within it is not constrained by any latitude in its conception to repeat what exists in one part in any other part. Even in this world more things exist without our knowledge than with it and the order in creation which you see is that which you have put there, like a string in a maze, so that you shall not lose your way. For existence has its own order and that no man's mind can compass, that mind itself being but a fact among others.

Cormac McCarthy

Elephants don't play chess.

Rodney Brooks

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 of my collaborators and academic mentors (special thanks to Lee)
[2] especially the support and encouragement of Elyse
[3] in humouring my insatiable need for debate and banter (special thanks to Lee)

Contents

Notation

a	:	Symbols in this font are real scalars.
$\mathbf{a}$	:	Symbols in this font are real column vectors.
$\boldsymbol{a}$	:	Symbols in this font are real column vectors in homogeneous coordinates.
$\mathbf{A}$	:	Symbols in this font are real matrices.
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$	:	Normally distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{R}$.
$E[\cdot]$	:	The expectation operator.
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	:	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.
$\mathbf{p}_a^{cb}$	:	A vector (resp. homogenous coodrdinates) from point b to point c (denoted by the superscript) and expressed in $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ (denoted by the subscript).
$\mathbf{C}_{ab}$	:	The 3×3 rotation matrix that transforms vectors from $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$ to $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$: $\mathbf{p}_a^{cb} = \mathbf{C}_{ab}\mathbf{p}_b^{cb}$.
$\mathbf{T}_{ba}$	:	The 4×4 transformation matrix that transforms homogeneous points from $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ to $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$: $\boldsymbol{p}_b^{cb} = \mathbf{T}_{ba}\boldsymbol{p}_a^{ca}$.

Chapter 2

Learned Probabilistic Sun Sensor

> He stepped down, avoiding any long look at her as one avoids long looks at the sun, but seeing her as one sees the sun, without looking.
>
> Leo Tolstoy, *Anna Karenina*

2.1 Background

2.1.1 Neural Networks for Parametric Learning

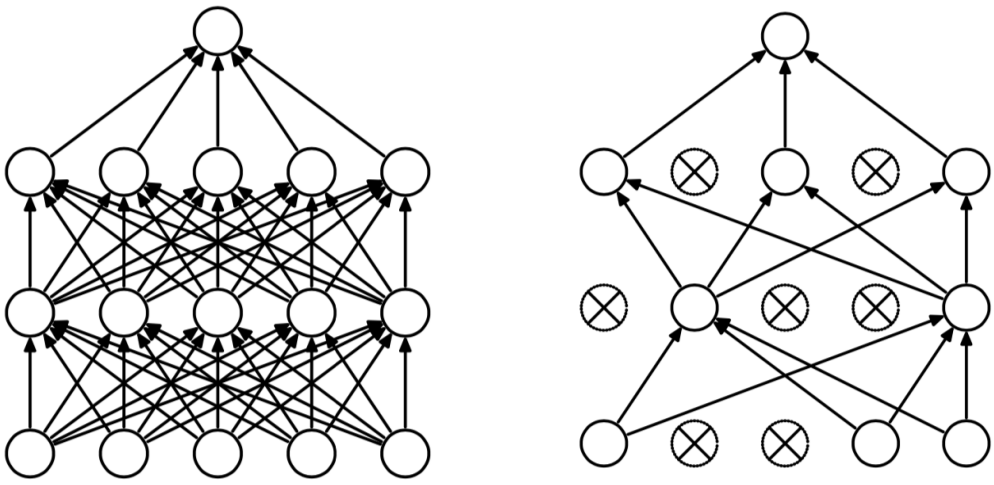
2.1.2 Convolutional Neural Networks

2.1.3 Regularization and Dropout

2.1.4 Dropout as Variational Inference

$$\mathbb{E}\left[\hat{\mathbf{s}}^*\right]_k = \hat{\bar{\mathbf{s}}}_k^* \approx \frac{1}{N}\sum_{n=1}^{N} \hat{\mathbf{s}}_k^*(\mathbf{x}_k^*, \boldsymbol{w}^n) \tag{2.1}$$

$$\begin{aligned}\mathbb{E}\left[\hat{\mathbf{s}}_k^* \; \hat{\mathbf{s}}^{*\,T}_k\right] \approx \tau^{-1}\mathbf{1} + \frac{1}{N}\sum_{n=1}^{N} \hat{\mathbf{s}}_k^*(\mathbf{x}_k^*, \boldsymbol{w}^n) \; \hat{\mathbf{s}}_k^*(\mathbf{x}_k^*, \boldsymbol{w}^n)^T \\ - \hat{\bar{\mathbf{s}}}_k^* \; \hat{\bar{\mathbf{s}}}_k^{*\,T},\end{aligned} \tag{2.2}$$



(a) Standard fully-connected neural network. (b) A neural network with dropout applied.

Figure 2.1: The technique of *dropout* stochastically removes the contribution of certain neurons to regularize learning. Figures from Srivastava et al. (2014).

Chapter 3

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

3.1 Coordinate Frames

Before we can present the main contributions of this dissertation, it will be useful to first outline the notation and mathematical foundations that underly the work. Throughout this dissertation, we largely follow the notation of Barfoot (2017) when dealing with three-dimensional rigid-body kinematics.

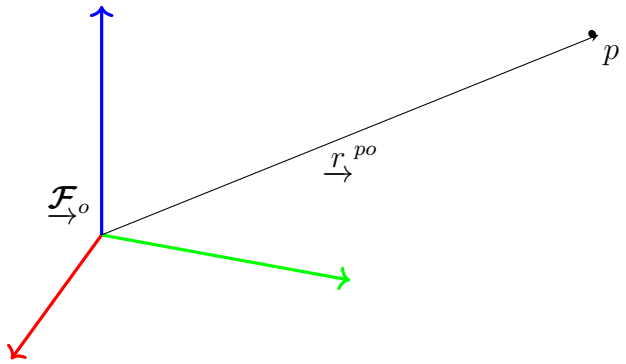


Figure 3.1: A position vector expressed in a coordinate frame.

We refer to a three-dimensional position vector, $\overrightarrow{r}^{po}$, as one that originates at the origin of a coordinate reference frame, $\overrightarrow{\mathcal{F}}_o$, 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.

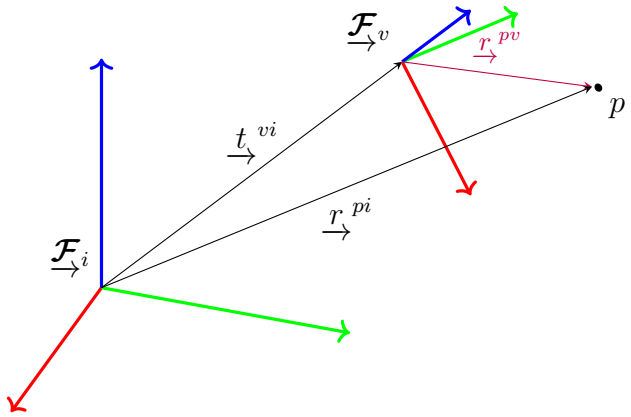


Figure 3.2: Two common references frames used throughout this thesis.

3.2 Rotations

The rotation matrix $\mathbf{C}$ is a member of the matrix Lie group[2] $\mathrm{SO}(3)$ (the Special Orthogonal group). We can define it as follows:

$$\mathrm{SO}(3) = \{\mathbf{C} \in \mathbb{R}^{3\times 3} | \ \mathbf{C}^T\mathbf{C} = \mathbf{1}, \det \mathbf{C} = 1\}. \tag{3.1}$$

Active and Passive Representations

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 dissertation 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 a thorough treatment of Lie groups for state estimation), we can define a surjective

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

[2]A Lie group is a group that is also a differentiable manifold. See Barfoot (2017) for more details.

exponential map[3] from three axis-angle parameters, $\boldsymbol{\phi} = \phi\mathbf{a}, \quad \phi \in \mathbb{R}, \mathbf{a} \in S^2$, to a rotation matrix, $\mathbf{C}$:

$$\mathbf{C} = \mathrm{Exp}\left(\boldsymbol{\phi}\right) = \exp\left(\boldsymbol{\phi}^{\wedge}\right) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\phi}^{\wedge})^n \tag{3.2}$$

$$= \cos\phi\mathbf{1} + (1 - \cos\phi)\mathbf{a}\mathbf{a}^T + \sin\phi\mathbf{a}^{\wedge}, \tag{3.3}$$

where the wedge operator $(\cdot)^{\wedge}$[4] is defined as

$$\mathbf{a}^{\wedge} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -a_2 & a_1 \\ a_2 & 0 & -a_0 \\ -a_1 & a_0 & 0 \end{bmatrix}. \tag{3.4}$$

Equation (3.3) is often referred to 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 (3.2) is surjective, we can define an inverse map if we restrict its domain to $0 \leq \phi < \pi$:

$$\boldsymbol{\phi} = \mathrm{Log}\left(\mathbf{C}\right) = \log\left(\mathbf{C}\right)^{\vee} = \frac{\phi(\mathbf{C} - \mathbf{C}^T)^{\vee}}{2\sin\phi}, \tag{3.5}$$

where $\phi = \arccos\left(\frac{\mathrm{tr}(\mathbf{C})-1}{2}\right)$ 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 (3.5) is undefined at both $\phi = 0$ and at $\phi = \pi$. In the former case, we can use a small-angle approximation and define

$$\mathrm{Log}\left(\mathbf{C}\right) \approx (\mathbf{C} - \mathbf{1})^{\vee} \quad \text{when} \quad \phi \approx 0. \tag{3.6}$$

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.

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

[4] This operator is sometimes also expressed as $(\cdot)^{\times}$ or $[\cdot]_{\times}$ and is known as the skew-symmetric operator.

3.2.1 Unit Quaternions

We can also represent a rotation with unit quaternion, $\mathbf{q}$. A unit quaternion consists of a scalar value q_ω, and a three-dimensional vector component, $\mathbf{q}_v$:

$$\mathbf{q} = \begin{bmatrix} q_\omega \\ \mathbf{q}_v \end{bmatrix} \in S^3, \quad (\|\mathbf{q}\| = 1). \tag{3.7}$$

Unit quaternions also form a Lie group (Solà et al., 2018) and and lie on a three-dimensional unit sphere within $\mathbb{R}^4$. As with rotation matrices, we can define a surjective map from three parameters to the group itself,

$$\mathbf{q} = \operatorname{Exp}(\boldsymbol{\phi}) = \begin{bmatrix} \cos(\phi/2) \\ \mathbf{a}\sin(\phi/2) \end{bmatrix}. \tag{3.8}$$

By inspection, we can see that both $\mathbf{q}$ and $-\mathbf{q}$ represent the same axis-angle pair, $\{\phi, \mathbf{a}\}$. As a result, unit quaternions represent a *double cover* of $\mathrm{SO}(3)$ and we must be careful to account for this when using them as a rotation parametrization. In particular, when computing the logarithmic map,

$$\boldsymbol{\phi} = \operatorname{Log}(\mathbf{q}) = 2\mathbf{q}_v \frac{\arctan(\|\mathbf{q}_v\|, q_\omega)}{\|\mathbf{q}_v\|}, \tag{3.9}$$

we can must account for the double cover by replacing $\mathbf{q}$ with $-\mathbf{q}$ if q_ω is negative. Also note that as with rotation matrices Equation (3.9) is undefined when $\phi = 0$, but, importantly, we do not face any issues when $\phi = \pi$ due to the half-angle. In the former case, we can again rely on small angle approximations to define:

$$\operatorname{Log}(\mathbf{q}) \approx \frac{\mathbf{q}_v}{q_\omega}\left(1 - \frac{\|\mathbf{q}_v\|^2}{3q_\omega^2}\right) \quad \text{when} \quad \phi \approx 0. \tag{3.10}$$

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

3.2.2 Topology

Topologically, $\mathrm{SO}(3)$ is *diffeomorphic*[5] to the real projective space, $\mathbb{RP}^3$, the space of all lines passing through the origin in $\mathbb{R}^4$ (Hartley et al., 2013). As a result, any global n-parametrization of $\mathrm{SO}(3)$ will incur some cost. If we use rotation matrices ($n = 9$), we need to ensure or-

[5] A diffeomorphism is a smooth invertible function that maps one differentiable manifold to another.

thonormality and that $\det \mathbf{C} = 1$. With unit quaternions ($n = 4$), we need to account for the unit-norm constraint and take note of the double cover. Parametrizations with $n = 3$ (like Euler angles and axis-angle parameters) will be bounded, but unconstrained. However, due to the topological structure of $\mathrm{SO}(3)$ all three-parameter parametrizations will not be invertible for certain rotations. With Euler angles, one has to be wary of *gimbal lock*, wherein two angles become indeterminate from each other. For axis-angle parameters, it is not possible to uniquely represent rotations whose angle is π.

Remark (SO(3) Topology). To see why this is the case, consider that according to Euler's theorem, we can represent any element in $\mathrm{SO}(3)$ by the axis-angle pair $\{\mathbf{a}, \phi\}$, $\mathbf{a} \in S^2$, and $0 \leq \phi \leq \pi$. We can partially represent this space by the closed ball of radius π in $\mathbb{R}^3$ using the combined axis-angle coordinates $\boldsymbol{\phi} = \phi\mathbf{a}$. However, at the boundary ($\phi = \pi$), we must account for the fact that rotations represented by $\{\mathbf{a}, \pi\}$ are identical to those represented by $\{-\mathbf{a}, \pi\}$. In other words, we must *identify* all antipodal points, $\boldsymbol{\phi}$ and $-\boldsymbol{\phi}$ when $\|\boldsymbol{\phi}\| = \pi$. This closed 3-ball with identified antipodal points on its boundary is topologically equivalent to the 3-sphere (S^3) with its antipodal points identified. In turn, this space is equivalent to $\mathbb{RP}^3$ since any line passing through the origin in $\mathbb{R}^4$ can be mapped to two unit normals, $\pm\mathbf{n} \in S^3$. This *identification* makes rotation representation particularly tricky in $\mathbb{R}^3$ and clearly explains why unit quaternions, $\mathbf{q} \in S^3$, are a *double* cover of $\mathrm{SO}(3)$, since we must add the relation $\mathbf{q} = -\mathbf{q}$ to make these two spaces equivalent. We direct the reader to Hartley et al. (2013) who use the *gnonomic* projection to provide further geometric intuition for why $\mathrm{SO}(3)$ is a projective space.

Accordingly, in this dissertation, we parametrize rotations as the constrained quantities, $\mathbf{q}$ or $\mathbf{C}$. When dealing with perturbations about a given rotation (e.g., to compute updates to a state, or to propagate uncertainty), we use *small* rotations, $\delta\mathbf{C}$ or $\delta\mathbf{q}$, parametrized using three unconstrained parameters that we can assume are a one-to-one mapping to elements in $\mathrm{SO}(3)$ (since, for small rotations, $\|\boldsymbol{\phi}\| << \pi$).

3.3 Spatial Transforms

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

$$\mathrm{SE}(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} | \ \mathbf{C} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3\}. \tag{3.11}$$

As a member of a matrix Lie group, it also admits a surjective exponential map,

$$\mathbf{T} = \text{Exp}\left(\boldsymbol{\xi}\right) = \exp\left(\boldsymbol{\xi}^{\wedge}\right) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\xi}^{\wedge})^{n} \tag{3.12}$$

where $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^6$ and the wedge operator is overloaded (following Barfoot (2017)) as follows:

$$\boldsymbol{\xi}^{\wedge} \triangleq \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^{\wedge} = \begin{bmatrix} \boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho} \\ \mathbf{0}^{T} & 0 \end{bmatrix}. \tag{3.13}$$

In practice, we can evaluate the exponential map through the Euler-Rodriguez formula (Equation (3.3)) and by computing the left-Jacobian of $\text{SO}(3)$, $\mathbf{J}$,

$$\mathbf{T} = \text{Exp}\left(\begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{C}(\boldsymbol{\phi}) & \mathbf{J}(\boldsymbol{\phi})\boldsymbol{\rho} \\ \mathbf{0}^{T} & 1 \end{bmatrix}, \tag{3.14}$$

where

$$\mathbf{J}(\boldsymbol{\phi}) = \frac{\sin\phi}{\phi}\mathbf{1} + (1 - \frac{\sin\phi}{\phi})\mathbf{a}\mathbf{a}^{T} + \frac{1-\cos\phi}{\phi}\mathbf{a}^{\wedge}. \tag{3.15}$$

3.3.1 Applying Transforms

Applying our notation for coordinate frames (and referring back to Section 3.1), a transform, $\mathbf{T}_{vi}$ can be expressed as

$$\mathbf{T}_{vi} = \begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_{v}^{iv} \\ \mathbf{0}^{T} & 1 \end{bmatrix}. \tag{3.16}$$

This allows us to use the homogenous point representation for $\mathbf{r}_{i}^{pi}$ and express the following relation:

$$\boldsymbol{r}_{v}^{pv} = \mathbf{T}_{vi}\boldsymbol{r}_{i}^{pi}, \tag{3.17}$$

or

$$\begin{bmatrix} \mathbf{r}_{v}^{pv} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_{v}^{iv} \\ \mathbf{0}^{T} & 1 \end{bmatrix}}_{\mathbf{T}_{vi}} \begin{bmatrix} \mathbf{r}_{i}^{pi} \\ 1 \end{bmatrix}, \tag{3.18}$$

which is numerically equivalent to

$$\mathbf{r}_{v}^{pv} = \mathbf{C}_{vi}\mathbf{r}_{i}^{pi} + \mathbf{t}_{v}^{iv}. \tag{3.19}$$

3.4 Perturbations

When solving optimization problems that involve rotations or rigid-body transforms, it is often useful to consider a small *perturbation* about an operating point. By leveraging a core property of Lie groups (that they are locally 'Euclidian'), we can convert difficult non-linear problems into ones that have local linear approximations.

Using rotations as an example, we can perturb an operating point, $\overline{\mathbf{C}} \triangleq \operatorname{Exp}\left(\overline{\boldsymbol{\phi}}\right)$, in three different ways:

$$\mathbf{C} = \begin{cases} \operatorname{Exp}\left(\delta\boldsymbol{\phi}^{\ell}\right)\overline{\mathbf{C}} & \text{left perturbation,} \\ \operatorname{Exp}\left(\overline{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}^{m}\right) & \text{middle perturbation,} \\ \overline{\mathbf{C}}\operatorname{Exp}\left(\delta\boldsymbol{\phi}^{r}\right) & \text{right perturbation.} \end{cases} \tag{3.20}$$

We can relate all the left and middle perturbations through the left Jacobian of $\mathrm{SO}(3)$ with the following useful identity,

$$\operatorname{Exp}\left(\left(\boldsymbol{\phi} + \delta\boldsymbol{\phi}^{m}\right)\right) \approx \operatorname{Exp}\left(\mathbf{J}(\boldsymbol{\phi})\delta\boldsymbol{\phi}^{m}\right)\operatorname{Exp}\left(\boldsymbol{\phi}\right). \tag{3.21}$$

From this it follows that $\delta\boldsymbol{\phi}^{\ell} \approx \mathbf{J}(\boldsymbol{\phi})\delta\boldsymbol{\phi}^{m}$ and elucidates why $\mathbf{J}$ is called the *left* Jacobian.

In this dissertation, we will use the left and middle perturbations when appropriate. Using small angle approximations, the Euler-Rodriguez formula (Equation (3.3)) yields $\operatorname{Exp}\left(\delta\boldsymbol{\phi}\right) \approx \mathbf{1} + \delta\boldsymbol{\phi}^{\wedge}$, which allows us to write the useful formula for the left perturbation:

$$\mathbf{C} = (\mathbf{1} + (\delta\boldsymbol{\phi}^{\ell})^{\wedge})\overline{\mathbf{C}}. \tag{3.22}$$

Similarly, we can write analogous expressions for a rigid body transform, $\mathbf{T} \in \mathrm{SE}(3)$, as composed of an operating point $\overline{\mathbf{T}} \triangleq \operatorname{Exp}\left(\overline{\boldsymbol{\xi}}\right)$, and a small perturbation about that operating point:

$$\mathbf{T} = \begin{cases} \operatorname{Exp}\left(\delta\boldsymbol{\xi}^{\ell}\right)\overline{\mathbf{T}} & \text{left perturbation,} \\ \operatorname{Exp}\left(\overline{\boldsymbol{\xi}} + \delta\boldsymbol{\xi}^{m}\right) & \text{middle perturbation,} \\ \overline{\mathbf{T}}\operatorname{Exp}\left(\delta\boldsymbol{\xi}^{r}\right) & \text{right perturbation.} \end{cases} \tag{3.23}$$

Now, we can also note a similar identity for $\mathrm{SE}(3)$,

$$\operatorname{Exp}\left(\left(\boldsymbol{\xi} + \delta\boldsymbol{\xi}^{m}\right)\right) \approx \operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}(\boldsymbol{\xi})\delta\boldsymbol{\xi}^{m}\right)\right)\operatorname{Exp}\left(\boldsymbol{\xi}\right), \tag{3.24}$$

where $\mathcal{J}$ is the left Jacobian of $\mathrm{SE}(3)$ and defined as

$$\mathcal{J}(\boldsymbol{\xi}) \triangleq \begin{bmatrix} \mathbf{J}(\boldsymbol{\phi}) & \mathbf{Q}(\boldsymbol{\xi}) \\ \mathbf{0} & \mathbf{J}(\boldsymbol{\phi}) \end{bmatrix}, \tag{3.25}$$

where $\mathbf{Q}(\boldsymbol{\xi})$ can be evaluated analytically (see Barfoot (2017)). This again allows us to write $\delta\boldsymbol{\xi}^{\ell} \approx \mathcal{J}(\boldsymbol{\xi})\delta\boldsymbol{\xi}^{m}$ and form a similar expression,

$$\mathbf{T} = (\mathbf{1} + (\delta\boldsymbol{\xi}^{\ell})^{\wedge})\overline{\mathbf{T}}. \tag{3.26}$$

To derive locally linear systems from sets of rigid-body transforms, or 'poses', we can apply Equation (3.26). To update an operating point, we solve for $\delta\boldsymbol{\xi}^{\ell}$ and then use the constraint-sensitive update $\mathbf{T} \leftarrow \mathrm{Exp}\left(\delta\boldsymbol{\xi}^{\ell}\right)\overline{\mathbf{T}}$.

Finally, we note that we will often drop the perturbation superscripts $(\cdot)^{\ell}$ and $(\cdot)^{m}$ after defining the perturbation scheme.

3.5 Uncertainty through Injection

We can also use perturbation theory to implicitly define uncertainty on constrained manifolds (see Barfoot and Furgale (2014) for a thorough discussion).

Given a concentrated normal density, $\delta\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{6\times6})$, we can *inject* this unconstrained density onto the Lie group through left perturbations about some mean using

$$\mathbf{T} = \mathrm{Exp}\left(\delta\boldsymbol{\xi}\right)\overline{\mathbf{T}}. \tag{3.27}$$

This allows us to keep track of a random variable, $\mathbf{T}$, by keeping its mean in group form, $\overline{\mathbf{T}}$, while its second statistical moment is stored as a standard 6×6 covariance matrix, $\boldsymbol{\Sigma}$.

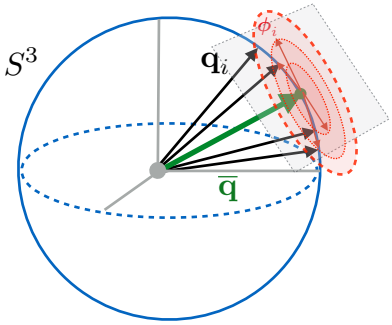


Figure 3.3: We can define uncertainty in the left tangent space of a mean element of a Lie group (here illustrated for unit quaternions).

We can define an analogous density for rotation matrices given normal densities over rotation perturbations $\delta\boldsymbol{\phi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{3\times3})$,

$$\mathbf{C} = \mathrm{Exp}\left(\delta\boldsymbol{\phi}\right)\overline{\mathbf{C}}, \tag{3.28}$$

and also, for unit quaternions,

$$\mathbf{q} = \mathrm{Exp}\left(\delta\boldsymbol{\phi}\right) \otimes \overline{\mathbf{q}} \tag{3.29}$$

where $\otimes$ refers to the standard quaternion product operator Sola (2017).

3.6 Deep Learning

Three of the four pseudo-sensors (Sun-BCNN, DPC, and HydraNet—Chapters 5 to 7) in this dissertation are built using neural networks and the tools of *deep learning* (LeCun et al., 2015). We briefly summarize these concepts here and refer the reader to Goodfellow et al. (2016) for a more thorough treatment.

3.6.1 Feed-forward Neural Networks

The basic premise of deep learning is that certain kinds of data are naturally decomposed into hierarchical structure. For instance, images may have local textures (e.g. edges), which compose basic primitives (e.g., leaves), which then combine to form semantic objects (e.g, a tree). Consequently, for this type of data, latent information may be efficiently extracted through an analogous hierarchical model. Typically, this is done through a neural network composed of multiple *layers* (the term *neural* comes from the loose biological basis for each layer, and *deep* refers to the amount of layers contained in state-of-the-art models). A standard neural layer computes a non-linear transformation from an input $\mathbf{z}_m$ to an output $\mathbf{z}_{m+1}$ as

$$\mathbf{z}_{m+1} = \mathbf{f}_m(\mathbf{z}) = \boldsymbol{\sigma}\left(\mathbf{W}_m\mathbf{z}_m + \mathbf{b}_m\right), \tag{3.30}$$

where $\mathbf{W}_m \in \mathbb{R}^{D_{m_{out}} \times D_{m_{in}}}$ and $\mathbf{b}_m \in \mathbb{R}^{D_{m_{out}}}$ are the parameters of the layer (often referred to as the *weight matrix* and *bias*, respectively) and $\boldsymbol{\sigma}(\cdot)$ is an element-wise *non-linearity*.

Remark (Non-linearities). The optimal choice of non-linearity is an area of active research. Some common examples include,

$$\sigma_i(x) = \begin{cases} \frac{1}{1+e^{-x}} & \text{Logistic,} \\ \frac{e^x - e^{-x}}{e^x + e^{-x}} & \text{Hyperbolic tangent, } \textit{tanh}, \\ \max(x, 0) & \text{Rectified Linear Unit, } \textit{ReLU}, \\ \begin{cases} x & \text{if } x \geq 0 \\ -\alpha x & \text{if } x < 0 \end{cases} & \text{Parametric ReLU, } \textit{PReLU}. \end{cases} \tag{3.31}$$

To produce a *feed-forward neural network*, M layers are composed in a hierarchy to create a single parametric function,

$$NN(\mathbf{x}) = \mathbf{f}_M \circ \mathbf{f}_{M-1} \circ \cdots \circ \mathbf{f}_1(\mathbf{x}), \tag{3.32}$$

where $f \circ g$ refers to function composition, $f(g(\cdot))$.

3.6.2 Convolutional Neural Networks

Convolutional Neural Networks (LeCun et al., 1989), or CNNs, are a particular form of neural network where at least one of the layers uses *convolution* in place of matrix multiplication. Due to their efficiency in processing high-dimensional image data, they have become ubiquitous in computer vision applications (LeCun et al., 2015). Briefly, a convolution is an operation that transforms a continuous input signal $x(t)$ into a new signal $y(t)$ as

$$y(t) = \int x(s)k(t-s)ds, \tag{3.33}$$

where k is often called the *kernel* of the convolution. For two-dimensional discrete input signals, convolution is defined as

$$y(i,j) = \sum_p \sum_q I(i-p, j-q)K(p,q). \tag{3.34}$$

Remark (Convolution vs. Cross-correlation). Most deep learning libraries implement convolution as the cross-correlation operation

$$y(i,j) = \sum_p \sum_q I(i+p, j+q)K(p,q) \tag{3.35}$$

which is convolution with the kernel 'flipped.' Unlike convolution, cross-correlation is not commutative with respect to the input and kernel (but this is typically not important in deep learning contexts). We note that kernels learned with cross-correlation will be flipped relative to those learned with true convolution (Goodfellow et al., 2016).

In this discrete case, a kernel can be represented by a kernel matrix, $\mathbf{K}$, which can have different *size* (e.g., $\mathbf{K} \in \mathbb{R}^{3\times3}$), and we may compute Equation (3.34) at different spatial intervals

called *strides*. To ensure that the convolution is defined at the boundaries of the signal, one can also use different *padding* strategies. Please refer to Goodfellow et al. (2016) for further information about these hyper-parameters.

By limiting the size of a kernel, we can use convolutional layers (parametrized by $\mathbf{K}$) to efficiently process high-dimensional signals. For images, a standard feed-forward layer would require a scalar weight for every pixel. In contrast, a convolutional layer can rely on a single kernel (with significantly less parameters than a weight matrix) to process an entire image with *shared weights*. This reduces the number of parameters of our network, but limits the type of spatial correlations we can model (since kernels can only capture local relations). To increase the potential modelling capacity of our network, we can use several independent kernels to process and output multiple *channels* (Figure 3.4). Finally, it is important to note that the convolution operator is particularly suited to visual data, as kernels can act as *filters* that pick out salient visual features like corners and edges.

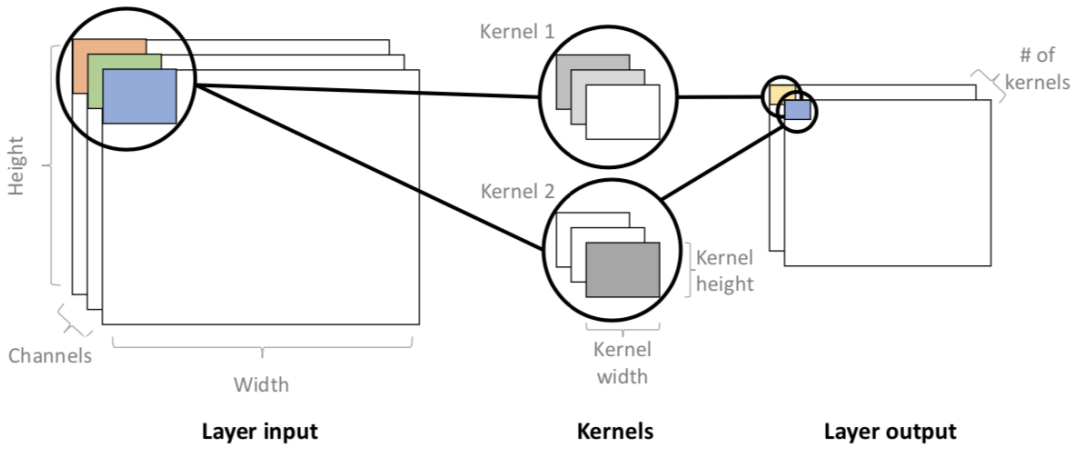


Figure 3.4: A convolutional layer with two kernels that operate over different channels of a two-dimensional input. Figure from Gal (2016).

3.6.3 Supervised Training

Given a (convolutional) neural network with a set of parameters $\boldsymbol{\Theta}$ (where $\boldsymbol{\Theta}$ may include weight and bias parameters, $\mathbf{W}_m, \mathbf{b}_m$, as well as convolutional kernel parameters, $\mathbf{K}_{km}$) we can use *supervised training* to obtain an optimal parameter set $\boldsymbol{\Theta}^*$. Supervised training requires set of training pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$ where $\mathbf{x} \in \mathbb{R}^{D_x}$ is an *input* and $\mathbf{y} \in \mathbb{R}^{D_y}$ is a *target* that corresponds to the desired output of our parametric model. To *train* the network,

we find the parameters which minimize a *loss function* over this dataset,

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \mathcal{L}\left(\{NN(\mathbf{x}_i; \boldsymbol{\Theta}), \mathbf{y}_i\}_{i=1}^N\right). \tag{3.36}$$

For example, a common loss function for regression problems is *mean squared error*,

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \frac{1}{2N} \sum_{i=1}^{N} \left\| NN(\mathbf{x}_i; \boldsymbol{\Theta}) - \mathbf{y}_i \right\|_2^2. \tag{3.37}$$

. In this dissertation, we will use supervised training in Chapters 5 to 7 and we will explore different forms of loss functions for geometric quantities that do not allow for a simple Euclidian norms.

3.6.4 Practical Considerations

Optimization

To train a deep network—that is, to solve Equation (3.36) for optimal parameters—modern approaches rely on stochastic gradient descent (SGD) through back-propagation (LeCun et al., 2015). SGD avoids the computationally prohibitive task of computing a gradient over an entire dataset by approximating the gradient using a randomly selecting subset of training data called a *mini-batch*. In the majority of this dissertation, we rely on Adam, a modern SGD-based approach that also maintains an estimate of second-order curvature (Kingma and Ba, 2017).

Regularization

In order to prevent *overfitting* to a dataset (i.e., to obtain a model that generalizes to unseen data), the literature provides a number of methods. An important approach that is used in this dissertation is *dropout* (Srivastava et al., 2014). Dropout stochastically 'zeros-out' inputs of a particular layer with probability p,

$$\mathbf{z}_{m+1} = \mathbf{f}_m(\mathbf{z}) = \boldsymbol{\sigma}\left(\mathbf{W}_m \tilde{\mathbf{z}}_m + \mathbf{b}_m\right), \tag{3.38}$$

where $\tilde{\mathbf{z}}_m = \mathbf{b} \circledcirc \mathbf{z}_m$, $b_i \sim \text{Bernoulli}(p)$ and $\circledcirc$ refers to element-wise multiplication. The intuition behind dropout is that it effectively creates an ensemble of smaller-parameter networks that will generally be less prone to overfitting than a commensurately-sized monolithic network. Importantly, dropout has a fundamental connection variational inference, which we exploit in Chapter 5.

Pooling and Spatial Invariance

A common technique in convolutional neural networks is *pooling* (Goodfellow et al., 2016). Pooling is a non-parametric operation that summarizes the output of a kernel in a particular region. For example, *max pooling* selects the maximum response of a kernel in demarcated regions of an input channel, thereby downsampling the resulting output. Pooling operations are designed to make convolution operations invariant to the spatial location of a particular kernel response. This is important in many classification tasks (e.g., a tree classifier should be invariant to the location of a particular leaf) but is a detriment to regression tasks where we would like to preserve spatial information. We explore building convolutional neural networks without pooling in Chapter 6.

Chapter 4

Classical Visual Odometry

> Eventually, my eyes were opened, and I really understood nature.
>
> — Claude Monet

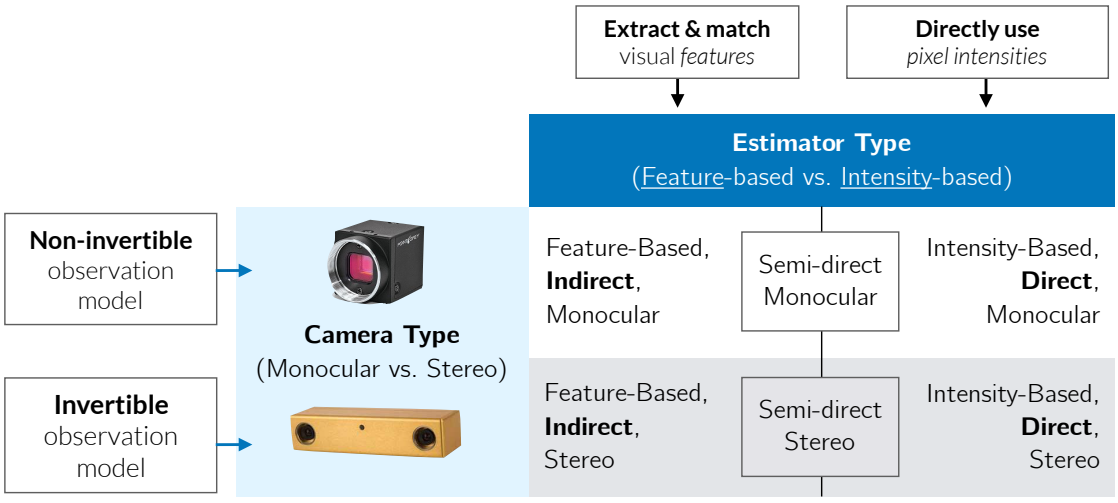


Figure 4.1: A taxonomy of different types of visual odometry.

Visual odometry (VO) has a rich history in mobile robotics and computer vision. As this dissertation largely deals with the improvement of a baseline visual odometry pipeline, we first outline the components of what we have chosen to be a canonical VO system. For a seminal tutorial on visual odometry and its more general cousin, visual SLAM, we refer the reader to two seminal papers: Scaramuzza and Fraundorfer (2011) and Cadena et al. (2016).

4.1 A Taxonomy of VO

VO can be largely divided along two dimensions (Figure 4.1): (1) the type of camera used to capture images (monocular vs. stereo) and (2) the type of data association used to compute motion estimates (indirect, or feature-based vs. direct, or pixel intensity-based).

Monocular vs. Stereo: Monocular VO methods use a single camera to infer motion and can use a single compact, low-power vision sensor. They do not require any extrinsic calibration but must rely on known visual cues or external information (e.g., wheel odometry, inertial measurements) to provide metric egomotion estimates. Conversely, stereo VO methods use a stereo camera to triangulate objects with metric scale. This allows stereo VO to provide metrically-acurate egomotion estimates. However, stereo methods rely on accurate extrinsic calibration, and their ability to resolve depth is limited by the baseline distance between the stereo pair and by the quality of stereo matches (which can be degraded by self-similar textures, occlusions, and foreshortening effects).

Direct vs. Indirect: The second distinction is based on the type of data association used to match sequential images and infer motion. Direct methods make the assumption of brightness constancy, and attempt to find the egomotion estimate that *directly* maximizes the similarity of pixel intensities between images. Indirect methods, conversely, rely on image features detectors to extract a set of salient landmarks or features, and then match these landmarks across images (typically by relying on a view-invariant descriptor).

4.2 Canonical VO Pipeline

In this thesis, we apply our learned pseudo-sensors to a baseline stereo, indirect visual odometry pipeline (Figure 4.2) largely based on the work of Furgale (2011). We choose this baseline system for its computational efficiency and robustness. We briefly summarize the main components of the pipeline here.

4.2.1 Preprocessing

During preprocessing, we use a lens model (assumed to be known apriori) to undistort each stereo image. Next, using the camera extrinsic parameters (also assumed to be known), we *rectify* the stereo pair such that the images can be assumed to come from two cameras whose principal axes are parallel (Figure 4.3). Finally, we also assume that the stereo camera intrinsics are known a priori or compute them through a calibration process (Furgale et al., 2013).

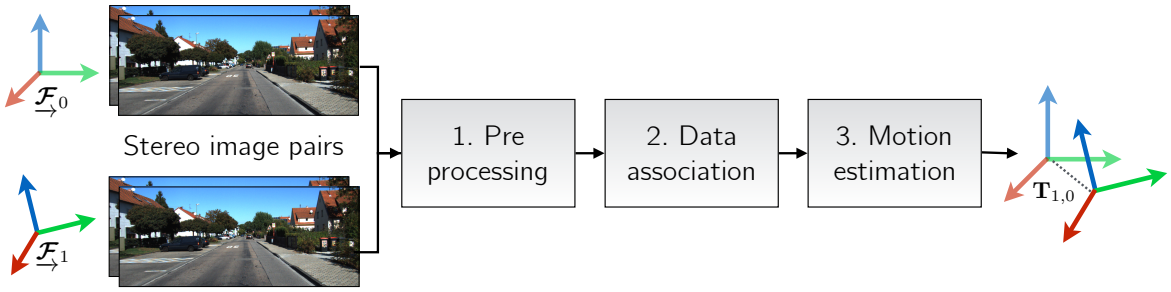
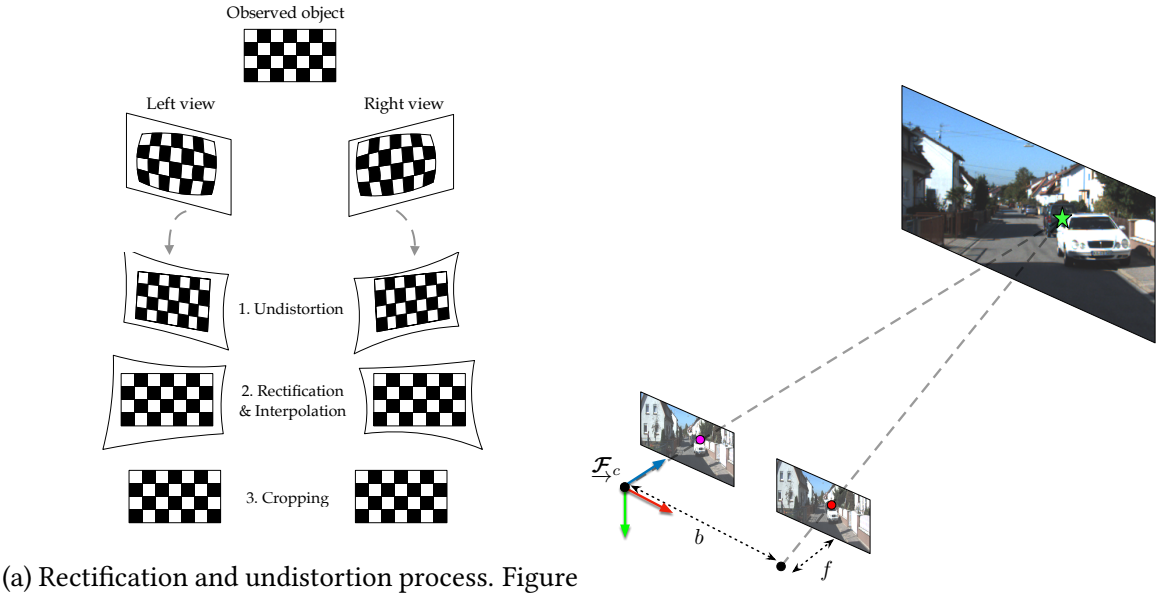


Figure 4.2: A 'classical' stereo visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.



(a) Rectification and undistortion process. Figure adapted from Florez (2010).

(b) Ideal stereo camera.

Figure 4.3: We pre-process stereo images (left) to simulate an ideal stereo camera (right).

4.2.2 Data Association

Feature Extraction and Matching

In this thesis, we focus on indirect stereo visual odometry for its computational efficiency. Although a number of different types of indirect feature extraction and matching methods can be used towards this end, we choose to use the `viso2` (Geiger et al., 2011) image feature extraction and matching algorithm as it is especially designed for sequential feature matching. In `viso2`, features are extracted using blob and corner masks with non-minimum and non-maximum suppression. Unlike other features detectors that do not assume a particular camera motion, `viso2` assumes a smooth camera trajectory that permits fast matching through a

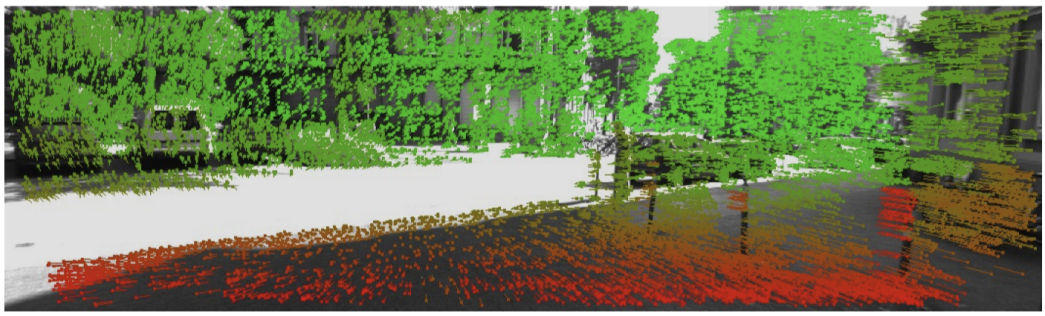


Figure 4.4: Feature tracking using `libviso2`, taken from Geiger et al. (2011). Colours correspond to depth.

simple sum-of-absolute-difference error metric based on Sobel filter responses. Features are matched across a stereo-pair and forward in time to ensure that a single feature exists across two consecutive stereo camera poses.

The ith feature corresponds to a point in space, expressed in homogeneous coordinates in the camera frame as $\boldsymbol{p}_{i,c} \in \mathbb{P}^3$. Given our intrinsics and extrinsic calibration parameters, our idealized stereo-camera model, $\mathbf{f}$, projects a landmark expressed in homogeneous coordinates into image space, so that $\mathbf{y}_{i,c}$, the stereo pixel coordinates of landmark i in the camera frame, are given by

$$\mathbf{y}_{i,c} = \begin{bmatrix} u_l \\ v_l \\ d \end{bmatrix} = \mathbf{f}(\boldsymbol{p}_{i,c}) = \mathbf{f}\left(\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\right) = \mathbf{M}\frac{1}{z}\boldsymbol{p}_{i,c}, \tag{4.1}$$

where

$$\mathbf{M} = \begin{bmatrix} f & 0 & c_u & 0 \\ 0 & f & c_v & 0 \\ 0 & 0 & 0 & fb \end{bmatrix}. \tag{4.2}$$

Here, $\{c_u, c_v\}$, f, and b are the principal points, focal length and baseline of the stereo camera respectively (computed through intrinsic calibration) and $d \triangleq u_l - u_r$ is the *disparity* of the feature. Note that in this formulation, the stereo camera frame is in the left optical centre. We can also define the inverse operation, $\mathbf{f}^{-1}$ (triangulation) as:

$$\boldsymbol{p}_{i,c} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{f}^{-1}\left(\begin{bmatrix} u_l \\ v_l \\ d \end{bmatrix}\right) = \begin{bmatrix} \frac{b}{d}(u_l - c_u) \\ \frac{b}{d}(v_l - c_v) \\ \frac{b}{d}f \\ 1 \end{bmatrix}. \tag{4.3}$$

Outlier Rejection

To filter out any stereo tracks that are *outliers*, we use a three-point random sample consensus algorithm (RANSAC, Fischler and Bolles (1981)) based on an analytic solution to the six degree-of-freedom motion (Umeyama, 1991) (refer to Appendix B for more details).

4.2.3 Maximum Likelihood Motion Solution

Finally, we compute the rigid-body transform between to stereo camera frames using maximum likelihood estimation. We define the rigid-body transform, $\mathbf{T}_t \in \mathrm{SE}(3)$, to be the rigid-body transform between two subsequent stereo camera poses, $\underrightarrow{\boldsymbol{\mathcal{F}}}_{c_0}$ and $\underrightarrow{\boldsymbol{\mathcal{F}}}_{c_1}$,

$$\mathbf{T}_t = \mathbf{T}_{c_1 w} \mathbf{T}_{c_0 w}^{-1}, \tag{4.4}$$

where $\underrightarrow{\boldsymbol{\mathcal{F}}}_{w}$ is a predefined world frame. After data association, we assume we have a set of N_t matches, $\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}\}_{i=1}^{N_t}$, between visual landmarks in the subsequent camera frames. For each match, we define an error function, $\mathbf{e}_i(\mathbf{T}_t)$, that relates the rigid transform to these stereo feature matches. Throughout this dissertation, we assume that these errors are corrupted by zero-mean independent Gaussian noise with the (potentially heteroscedastic) covariance, $\boldsymbol{\Sigma}_{i,t}$;

$$\mathbf{e}_i(\mathbf{T}_t) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{i,t}\right). \tag{4.5}$$

Under this noise model, the maximum likelihood transform, $\mathbf{T}_t^*$, is given by

$$\mathbf{T}_t^* = \underset{\mathbf{T} \in \mathrm{SE}(3)}{\operatorname{argmax}} \prod_{i=1}^{N_t} p(\mathbf{e}_i(\mathbf{T}_t)) = \underset{\mathbf{T} \in \mathrm{SE}(3)}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{N_t} \mathbf{e}_i(\mathbf{T}_t)^T \boldsymbol{\Sigma}_{i,t}^{-1} \mathbf{e}_i(\mathbf{T}_t). \tag{4.6}$$

We will define the error function in two different ways.

Point Cloud Error

First, we can follow classical approach (Maimone et al., 2007) and define $\mathbf{e}_i(\mathbf{T}_t)$ based on a three-dimensional point cloud error. To do this, we invert our stereo camera model to triangulate pairs of points in each frame, $\boldsymbol{p}_{i,c_0} = \mathbf{f}^{-1}(\mathbf{y}_{i,c_0})$ and $\boldsymbol{p}_{i,c_1} = \mathbf{f}^{-1}(\mathbf{y}_{i,c_1})$, and then define a three-dimensional error,

$$\mathbf{e}_i(\mathbf{T}_t) = \mathbf{D}(\boldsymbol{p}_{i,c_1} - \mathbf{T}_t \boldsymbol{p}_{i,c_0}) \in \mathbb{R}^3, \tag{4.7}$$

where $\mathbf{D} = \begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3\times 4}$ converts homogenous coordinates into Euclidian coordinates.

We can then follow Maimone et al. (2007) and assume each stereo projection is corrupted by additive Gaussian noise,

$$\mathbf{y}_{i,c} \sim \mathcal{N}\left(\bar{\mathbf{y}}_{i,c}, \mathbf{R}_{i,c}\right), \tag{4.8}$$

and compute a density on the error function itself through first order noise propagation. This gives the density

$$\mathbf{e}_i(\mathbf{T}_t) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{i,t}\right), \tag{4.9}$$

where

$$\mathbf{\Sigma}_{i,t} = \mathbf{D}\mathbf{G}_{i,c_1}\mathbf{R}_{i,c_1}\mathbf{G}_{i,c_1}^T\mathbf{D}^T + \mathbf{D}\mathbf{T}_t\mathbf{G}_{i,c_0}\mathbf{R}_{i,c_0}\mathbf{G}_{i,c_0}^T\mathbf{T}_t^T\mathbf{D}^T, \tag{4.10}$$

with $\mathbf{G}_{i,c} = \left.\dfrac{\partial \mathbf{f}^{-1}}{\partial \mathbf{y}}\right|_{\mathbf{y}_{i,c}}$.

Reprojection Error

Alternatively, we can represent reprojection errors in the second frame directly as

$$\mathbf{e}_i(\mathbf{T}_t) = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}_t\mathbf{f}^{-1}(\mathbf{y}_{i,c_0})), \tag{4.11}$$

and assume the following simple noise model

$$\mathbf{e}_i(\mathbf{T}_t) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{i,t}\right) = \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{i,t}\right), \tag{4.12}$$

where we abuse notation (slightly) and replace the index for the camera frames c_0 or c_1 with t to indicate that this covariance refers to the reprojection error that involves both sets of features.

Importantly, Sibley et al. (2007) show that using reprojection error (as compared to 3D point cloud error) results in less biased estimates for long-range stereo triangulation. Consequently, we favour this latter formulation in the large majority of our work (the one exception being the initial work on isotropic PROBE described in Appendix A).

Solution via Gauss-Newton Optimization

In either case, we have now defined a weighted nonlinear least squares problem which can be solved iteratively using standard techniques. For our purposes, we opt to use Gauss-Newton optimization and follow Barfoot (2017) to optimize constrained poses.

Namely, at a given iteration n, we linearize the error function $\mathbf{e}_i(\mathbf{T}_t)$, about an operating point $\mathbf{T}_t^{(n)} \in \mathrm{SE}(3)$, which results in a quadratic approximation to Equation (A.3). To

linearize, we consider the left perturbations $\delta\boldsymbol{\xi} \in \mathbb{R}^6$ represented in exponential coordinates:

$$\mathbf{T}_t = \operatorname{Exp}(\delta\boldsymbol{\xi})\,\mathbf{T}_t^{(n)} \approx (\mathbf{1} + \delta\boldsymbol{\xi}^{\wedge})\mathbf{T}_t^{(n)}. \tag{4.13}$$

This allows us to transform Equation (A.3) into a linear least squares objective in $\delta\boldsymbol{\xi}$:

$$\mathcal{L}(\delta\boldsymbol{\xi}) = \frac{1}{2}\sum_{i=1}^{N_t} (\mathbf{e}_i - \mathbf{J}_i\delta\boldsymbol{\xi})^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{e}_i - \mathbf{J}_i\delta\boldsymbol{\xi}) \tag{4.14}$$

where $\mathbf{J}_i = \left.\dfrac{\partial \mathbf{e}_i}{\partial \delta\boldsymbol{\xi}}\right|_{\mathbf{T}_t^{(n)}}$, $\mathbf{e}_i = \mathbf{e}_i(\mathbf{T}_t^{(n)})$, and $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i,t}(\mathbf{T}_t^{(n)})$. The minimum to this objective can be solved for analytically by solving the normal equations. This results in the optimal parameters,

$$\delta\boldsymbol{\xi}^{\star} = \left(\sum_{i=1}^{N_t} \mathbf{J}_i^T \boldsymbol{\Sigma}_i^{-1}\mathbf{J}_i\right)^{-1} \sum_{i=1}^{N_t} \mathbf{J}_i^T \boldsymbol{\Sigma}_i^{-1}\mathbf{e}_i. \tag{4.15}$$

Given $\delta\boldsymbol{\xi}^{\star}$, we can update the operating point using the constraint-sensitive update

$$\mathbf{T}^{(n+1)} = \operatorname{Exp}(\delta\boldsymbol{\xi}^{\star})\,\mathbf{T}^{(n)}, \tag{4.16}$$

and iterate until convergence. See Appendix B for more details and an analytic expression for $\mathbf{J}_i$. There are many reasonable choices for both the initial transform $\mathbf{T}^{(0)}$ and for the conditions under which we terminate iteration. For most visual odometry applications, it suffices to initialize the estimated transform to identity, and iteratively perform the update given by eq. (4.16) until we see a relative change in the squared error of less than one percent after an update.

4.3 Robust Estimation

Since Equation (4.14) 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 quadratic loss function with one that is less sensitive to large measurement errors (MacTavish and Barfoot, 2015). These robust cost functions are collectively known as M-estimators[1], and many variants exist. Each uses a re-weighting function, $\rho(\cdot)$, to define a new optimization problem,

[1]M, for *maximum-likelihood-type* since they generalize the basic maximum likelihood solution (Barfoot, 2017).

$$\mathbf{T}^*_{\text{RLS}} = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}} \sum_{i=1}^{N} \rho\left(\sqrt{\mathbf{e}_i{}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{e}_i}\right) = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}} \sum_{i=1}^{N} \rho(\epsilon_i) = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}}\, \mathcal{L}_{\text{RLS}}(\mathbf{T}), \tag{4.17}$$

where we have defined $\epsilon_i \triangleq \sqrt{\mathbf{e}_i{}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{e}_i}$ (and dropped the t subscript for clarity). The basic idea with M-estimation is to use a $\rho(\cdot)$ that reduces the influence of large ϵ below that of the quadratic $\rho(\epsilon) = \frac{1}{2}\epsilon^2$. There are several examples of such functions, including,

$$\rho(\epsilon) = \begin{cases} \frac{c^2}{2}\log\left(1 + \frac{\epsilon^2}{c^2}\right) & \text{Cauchy,} \\ \frac{1}{2}\frac{\epsilon^2}{c^2+\epsilon^2} & \text{Geman-McClure (Geman et al., 1992),} \\ \begin{cases} \frac{\epsilon^2}{2} & \text{if } \epsilon < c \\ c\epsilon - \frac{c^2}{2} & \text{if } \epsilon \geq c \end{cases} & \text{Huber (Huber, 1964).} \end{cases} \tag{4.18}$$

where the constant c can be set with reference to asymptotic efficiency relative to a unit Gaussian (Holland and Welsch, 1977). To solve Equation (4.17), it is common in the literature to apply the technique of *iteratively reweighted least squares* (IRLS) (Holland and Welsch, 1977). To do this, we define a new non-linear least squares minimization problem,

$$\mathbf{T}^*_{\text{IRLS}} = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}} \frac{1}{2} \sum_{i=1}^{N} \mathbf{e}_i{}^T \mathbf{M}_i \mathbf{e}_i = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}}\, \mathcal{L}_{\text{IRLS}}(\mathbf{T}) \tag{4.19}$$

where we define these new weights, $\mathbf{M}_i$, based on an *influence function*, $\psi(\cdot)$ as

$$\mathbf{M}_i = \frac{1}{\epsilon_i} \underbrace{\left.\frac{\partial \rho}{\partial \epsilon}\right|_{\epsilon_i}}_{\psi(\cdot)} \boldsymbol{\Sigma}_i^{-1}, \tag{4.20}$$

and solve it using the Gauss-Newton approach presented in Section 4.2.3. We claim that upon convergence, $\mathbf{T}^*_{\text{IRLS}}$ will also minimize Equation (4.17). To see why, consider that

$$\frac{\partial \mathcal{L}_{\text{RLS}}}{\partial \delta\boldsymbol{\xi}} = \sum_i^N \frac{\partial \rho}{\partial \epsilon_i} \frac{\partial \epsilon_i}{\partial \mathbf{e}_i} \frac{\partial \mathbf{e}_i}{\partial \delta\boldsymbol{\xi}} = \sum_i^N \frac{1}{\epsilon_i} \frac{\partial \rho}{\partial \epsilon_i} \mathbf{e}_i{}^T \boldsymbol{\Sigma}_i^{-1} \frac{\partial \mathbf{e}_i}{\partial \delta\boldsymbol{\xi}}, \tag{4.21}$$

where he have used the fact that $\frac{\partial \epsilon_i}{\partial \mathbf{e}_i} = \frac{1}{\epsilon_i} \mathbf{e}_i{}^T \boldsymbol{\Sigma}_i^{-1}$. Now using our definition of $\mathbf{M}_i$, we can write,

$$\frac{\partial \mathcal{L}_{\text{RLS}}}{\partial \delta \boldsymbol{\xi}} = \sum_i^N \mathbf{e}_i{}^T \underbrace{\frac{1}{\epsilon_i} \frac{\partial \rho}{\partial \epsilon_i} \boldsymbol{\Sigma}_i^{-1}}_{\mathbf{M}_i(\mathbf{T})} \frac{\partial \mathbf{e}_i}{\partial \delta \boldsymbol{\xi}} = \sum_i^N \mathbf{e}_i{}^T \mathbf{M}_i(\mathbf{T}) \frac{\partial \mathbf{e}_i}{\partial \delta \boldsymbol{\xi}}, \tag{4.22}$$

where we have made the dependence on $\mathbf{T}$ explicit. We could potentially proceed to set this gradient to $\mathbf{0}$ and attempt to solve for an optimal update $\delta \boldsymbol{\xi}$. However, due to $\mathbf{M}_i(\mathbf{T})$, this may be difficult in general. Instead, we note that if we evaluate $\mathbf{M}_i(\mathbf{T})$ at the current operating point, $\mathbf{T}^{(n)}$, (i.e., we *re-weight* the loss) we are then left with the equivalent normal equations that solve $\frac{\partial \mathcal{L}_{\text{IRLS}}}{\partial \delta \boldsymbol{\xi}} = \mathbf{0}$.

Furthermore, upon convergence, our solution to the iteratively re-weighted problem $\mathbf{T}^{(n)} = \mathbf{T}^*_{\text{IRLS}}$ will also minimize the robust objective Equation (4.17), since we must have that,

$$\left. \frac{\partial \mathcal{L}_{\text{IRLS}}}{\partial \delta \boldsymbol{\xi}} \right|_{\mathbf{T}^*_{\text{IRLS}}} = \left. \frac{\partial \mathcal{L}_{\text{RLS}}}{\partial \delta \boldsymbol{\xi}} \right|_{\mathbf{T}^*_{\text{IRLS}}} = \mathbf{0}. \tag{4.23}$$

4.4 Outstanding Issues

Finally, we summarize three high-level limitations of such a canonical visual odometry pipeline that we will address with learned pseudo-sensors: efficiency, systematic bias and homoscedastic uncertainnty.

Table 4.1: **Data efficiency vs. computational efficiency**

Synopsis	**Addressed by**
Classical VO pipelines face a difficult-to-optimize trade-off between using all of the information contained within image and while still remaining computationally tractable.	PROBE, DPC-Net, SunBCNN, HydraNet

Table 4.2: **Systematic bias**

Synopsis	**Addressed by**
Stereo visual odometry can incur systematic bias through poor extrinsic or intrinsic calibration, stereo triangulation errors, poor feature *spread* (i.e., concentration of features on one side of an image), and poor data association due self-similar textures.	DPC-Net

Table 4.3: **Homoscedastic uncertainty**

Synopsis	**Addressed by**
Stationary, homoscedastic noise in observation models can often reduce the consistency and accuracy of state estimates. This is especially true for complex, inferred measurement models.	PROBE, Sun-BCNN, HydraNet

Appendices

Bibliography

Agarwal, S., Mierle, K., et al. (2016). Ceres solver.

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.

Barfoot, T. D. and Furgale, P. T. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Rob.*, 30(3):679–693.

Brachmann, E. and Rother, C. (2018). Learning less is more-6d camera localization via 3d surface regression. In *Proc. CVPR*, volume 8.

Byravan, A. and Fox, D. (2017). SE3-nets: Learning rigid body motion using deep neural networks. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 173–180.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the Robust-Perception age. *IEEE Trans. Rob.*, 32(6):1309–1332.

Carlone, L., Rosen, D. M., Calafiore, G., Leonard, J. J., and Dellaert, F. (2015a). Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 125–132.

Carlone, L., Tron, R., Daniilidis, K., and Dellaert, F. (2015b). Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4597–4604.

Cheng, Y., Maimone, M. W., and Matthies, L. (2006). Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging. *IEEE Robot. Automat. Mag.*, 13(2):54–62.

Clark, R., Wang, S., Wen, H., Markham, A., and Trigoni, N. (2017). Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem.

Clement, L. and Kelly, J. (2018). How to train a CAT: learning canonical appearance transformations for direct visual localization under illumination change. *IEEE Robotics and Automation Letters*, 3(3):2447–2454.

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.

Costante, G., Mancini, M., Valigi, P., and Ciarfuglia, T. A. (2016). Exploring representation learning with CNNs for Frame-to-Frame Ego-Motion estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25.

Crete, F., Dolmiere, T., Ladret, P., and Nicolas, M. (2007). The blur effect: perception and estimation with a new no-reference perceptual blur metric. In *Human vision and electronic imaging XII*, volume 6492, page 64920I. International Society for Optics and Photonics.

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.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*, pages 248–255.

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

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *Proc. Int. Conf. on Machine Learning*, ICML'16, pages 1329–1338.

Eisenman, A. R., Liebe, C. C., and Perez, R. (2002). Sun sensing on the mars exploration rovers. In *Aerosp. Conf. Proc.*, volume 5, pages 5–2249–5–2262 vol.5. IEEE.

Engel, J., Stuckler, J., and Cremers, D. (2015). Large-scale direct SLAM with stereo cameras. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, pages 1935–1942.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395.

Fisher, R. (1953). Dispersion on a sphere. In *Proc. Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 217, pages 295–305. The Royal Society.

Fitzgibbon, A. W., Robertson, D. P., Criminisi, A., Ramalingam, S., and Blake, A. (2007). Learning priors for calibrating families of stereo cameras. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 1–8.

Florez, S. A. R. (2010). *Contributions by vision systems to multi-sensor object localization and tracking for intelligent vehicles*. PhD thesis.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2015). IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori 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. IEEE.

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

Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *J. Field Robot.*, 27(5):534–560.

Furgale, P., Carle, P., Enright, J., and Barfoot, T. D. (2012). The devon island rover navigation dataset. *Int. J. Rob. Res.*, 31(6):707–713.

Furgale, P., Enright, J., and Barfoot, T. (2011). Sun sensor navigation for planetary rovers: Theory and field testing. *IEEE Trans. Aerosp. Electron. Syst.*, 47(3):1631–1647.

Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.

Gal, Y. and Ghahramani, Z. (2016a). Bayesian convolutional neural networks with Bernoulli approximate variational inference. In *Proc. Int. Conf. Learning Representations (ICLR), Workshop Track*.

Gal, Y. and Ghahramani, Z. (2016b). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int. Conf. Mach. Learning (ICML)*, pages 1050–1059.

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.

Geiger, A., Ziegler, J., and Stiller, C. (2011). StereoScan: Dense 3D reconstruction in real-time. In *Proc. IEEE Intelligent Vehicles Symp. (IV)*, pages 963–968.

Geman, S., McClure, D. E., and Geman, D. (1992). A nonlinear filter for film restoration and other problems in image processing. *CVGIP: Graphical models and image processing*, 54(4):281–289.

Glocker, B., Izadi, S., Shotton, J., and Criminisi, A. (2013). Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

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.

Haarnoja, T., Ajay, A., Levine, S., and Abbeel, P. (2016). Backprop KF: Learning discriminative deterministic state estimators. In *Proc. Advances in Neural Inform. Process. Syst. (NIPS)*.

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.

Hartley, R., Trumpf, J., Dai, Y., and Li, H. (2013). Rotation averaging. *Int. J. Comput. Vis.*, 103(3):267–305.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Holland, P. W. and Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827.

Hu, H. and Kantor, G. (2015). Parametric covariance prediction for heteroscedastic noise. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, pages 3052–3057.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM Int. Conf. Multimedia (MM)*, pages 675–678.

Kelly, J., Saripalli, S., and Sukhatme, G. S. (2008). Combined visual and inertial navigation for an unmanned aerial vehicle. In *Proc. Field and Service Robot. (FSR)*, pages 255–264.

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.

Kendall, A. and Cipolla, R. (2017). Geometric loss functions for camera pose regression with deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564.

Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A convolutional network for Real-Time 6-DOF camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946.

Kerl, C., Sturm, J., and Cremers, D. (2013). Robust odometry estimation for RGB-D cameras. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3748–3754.

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. arXiv: 1412.6980.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc.

Lalonde, J.-F., Efros, A. A., and Narasimhan, S. G. (2011). Estimating the natural illumination conditions from a single outdoor image. *Int. J. Comput. Vis.*, 98(2):123–145.

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.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. (2015). Why M heads are better than one: Training a diverse ensemble of deep networks.

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.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*

Li, Q., Qian, J., Zhu, Z., Bao, X., Helwa, M. K., and Schoellig, A. P. (2017a). Deep neural networks for improved, impromptu trajectory tracking of quadrotors. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5183–5189.

Li, R., Wang, S., Long, Z., and Gu, D. (2017b). UnDeepVO: Monocular visual odometry through unsupervised deep learning.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ma, W.-C., Wang, S., Brubaker, M. A., Fidler, S., and Urtasun, R. (2016). Find your way by observing the sun and other semantic cues.

MacTavish, K. and Barfoot, T. D. (2015). At all costs: A comparison of robust cost functions for camera correspondence outliers. In *Proc. Conf. on Comp. and Robot Vision (CRV)*, pages 62–69.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2016). 1 year, 1000 km: The oxford RobotCar dataset. *Int. J. Rob. Res.*

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *J. Field Robot.*, 24(3):169–186.

Mayor, A. (2019). *Gods and Robots*. Princeton University Press.

McManus, C., Upcroft, B., and Newman, P. (2014). Scene signatures: Localised and point-less features for localisation. In *Proc. Robotics: Science and Systems X*.

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.

Oliveira, G. L., Radwan, N., Burgard, W., and Brox, T. (2017). Topometric localization with deep learning. *arXiv preprint arXiv:1706.08775*.

Olson, C. F., Matthies, L. H., Schoppers, M., and Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robot. Auton. Syst.*, 43(4):215–229.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped DQN. In *Proc. Advances in Neural Inform. Process. Syst. (NIPS)*, pages 4026–4034.

Peretroukhin, V., Clement, L., Giamou, M., and Kelly, J. (2015a). 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. (2015b). Get to the point: Active covariance scaling for feature tracking through motion blur. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshop on Scaling Up Active Perception*, Seattle, Washington, USA.

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*, 37(9):996–1016.

Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*, 3(3):2424–2431.

Peretroukhin, V., Kelly, J., and Barfoot, T. D. (2014). Optimizing camera perspective for stereo visual odometry. In *Canadian Conference on Comp. and Robot Vision*, pages 1–7.

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.

Peretroukhin, V., Wagstaff, B., and Kelly, J. (2019). Deep probabilistic regression of elements of SO(3) using quaternion averaging and uncertainty injection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19) Workshop on Uncertainty and Robustness in Deep Visual Learning*, pages 83–86, Long Beach, California, USA.

Punjani, A. and Abbeel, P. (2015). Deep learning helicopter dynamics models. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3223–3230.

Rosen, D. M., Carlone, L., Bandeira, A. S., and Leonard, J. J. (2019). SE-Sync: A certifiably correct algorithm for synchronization over the special euclidean group. *Int. J. Rob. Res.*, 38(2-3):95–125.

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

Sibley, G., Matthies, L., and Sukhatme, G. (2007). Bias reduction and filter convergence for long range stereo. In *Robotics Research*, pages 285–294. Springer Berlin Heidelberg.

Sola, J. (2017). Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*.

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

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

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.

Sunderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., and Milford, M. (2015). Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. In *Proc. Robotics: Science and Systems XII*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*, pages 1–9.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle Adjustment — A Modern Synthesis. In Goos, G., Hartmanis, J., van Leeuwen, J., Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*, volume 1883, pages 298–372. Springer Berlin Heidelberg, Berlin, Heidelberg.

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.

Umeyama, S. (1991). Least-Squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380.

Vega-Brown, W. and Roy, N. (2013). CELLO-EM: Adaptive sensor models without ground truth. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pages 1907–1914.

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.

Wang, S., Clark, R., Wen, H., and Trigoni, N. (2017). DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050.

Yang, F., Choi, W., and Lin, Y. (2016). Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proc. IEEE Int. Conf. Comp. Vision and Pattern Recognition (CVPR)*, pages 2129–2137.

Yang, N., Wang, R., Stueckler, J., and Cremers, D. (2018). Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision (ECCV)*. accepted as oral presentation, arXiv 1807.02570.

Zhang, G. and Vela, P. (2015). Optimally observable and minimal cardinality monocular SLAM. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5211–5218.

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

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in Neural Inform. Process. Syst. (NIPS)*, pages 487–495.

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and Ego-Motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619.