On learning pseudo-sensors to improve egomotion estimation for mobile autonomy

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 egomotion estimation for mobile autonomy

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 this thesis, we present several data-driven learned *pseudo-sensors* that serve to complement conventional pipelines by inferring latent information from the same data stream. Our approach retains much 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
[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.
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$	:	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). This vector can be in homogenous coordinates depending on context.
$\mathbf{C}_{a,b}$	:	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

Predictive Robust Estimation

> Information is the resolution of uncertainty.
>
> — Claude Shannon

1.1 Introduction

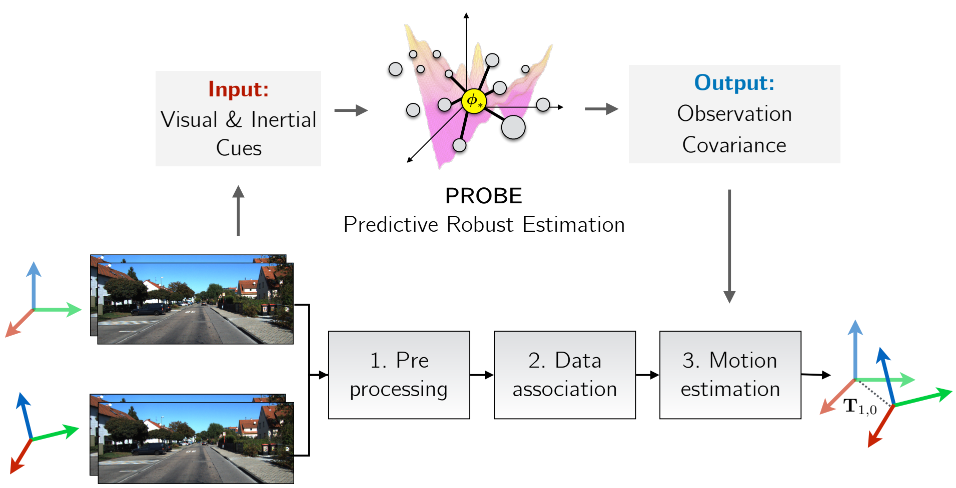

Figure 1.1: PROBE builds a predictive noise model for stereo visual odometry.

The first pseudo-sensor we present is a general technique we call PRedictive ROBust Estimation, or PROBE. This approach uses non-parametric learning to predict observation covariances for a stereo visual odometry pipeline, effectively scaling a least squares objective in a predictive fashion. We present two different methods to learn and incorporate these covariances. First we use a simple k-nearest-neighbours approach to learn isotropic covariances for three dimensional point-cloud matching. Second, we extend this significantly by applying the method of Generalized Kernels to a Bayesian treatment of covariance learning. We show that by assuming a particular covariance prior over re-projection errors, we can derive a robust least squares loss with parameters that are predicted for each error by our approach.

There are three publications associated with this work:

- 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. (2015c). 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., 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.

1.2 Motivation

Robot navigation relies on an accurate quantification of sensor noise or uncertainty in order to produce reliable state estimates. In practice, this uncertainty is often fixed for a given sensor and experiment, whether by automatic calibration or by manual tuning. Although a fixed measure of uncertainty may be reasonable in certain static environments, dynamic scenes frequently exhibit many effects that corrupt a portion of the available observations. For visual sensors, these effects include, for example, self-similar textures, variations in lighting, moving objects, and motion blur. We assert that there may be useful information available in these observations that would normally be rejected by a fixed-threshold outlier rejection scheme. Ideally, we would like to retain some of these observations in our estimator, while still placing more trust in observations that do not suffer from such effects.

1.3 Related Work

There is a large and growing body of work on the problem of deriving accurate, consistent state estimates from visual data. Although our approach to noise modelling is applicable in other domains, for simplicity we focus our attention on the problem of inferring egomotion from features extracted from sequential pairs of stereo images; see Sünderhauf and Protzel (2007) for a survey of techniques. The spectrum of alternative approaches to visual state estimation include monocular techniques, which may be feature-based (Scaramuzza and Fraundorfer, 2011), direct (Irani and Anandan, 2000), or semi-direct (Forster et al., 2014).

Apart from simply rejecting outliers, a number of recent approaches attempt to select the optimal set of features to produce an accurate localization estimate from tracked visual features. For example, Tsotsos et al. (2015) amend Random Sample Consensus (RANSAC) with statistical hypothesis testing to ensure that tracked visual features have normally distributed residuals before including them in the estimator. Unlike our predictive approach, their technique relies on the availability of feature tracks, and requires scene overlap to work continuously. In a different approach, Zhang and Vela (2015) choose an optimally observable feature subset for a monocular SLAM pipeline by selecting features with the highest *informativeness* - a measure calculated based on the observability of the SLAM subsystem. Observability, however, is governed by the 3D location of the features, and therefore cannot predict systematic feature degradation due to environmental or sensor-based effects.

1.4 Scalar k-Nearest Neighbours

Predictive ROBust Estimation is a technique that improves localization accuracy in the presence of such effects by building a model of the uncertainty in the affected visual observations. We learn the model in an offline training procedure and then use it online to predict the uncertainty of incoming observations as a function of their location in a predefined *prediction space*. Our model can be learned in completely unknown environments with frequent or infrequent ground truth data.

The primary contributions of this research are a flexible framework for learning the quality of visual features with respect to navigation estimates, and a straightforward way to incorporate this information into a navigation pipeline. On its own, PROBE can produce more accurate estimates than a binary outlier rejection scheme like Random Sample Consensus (RANSAC) because it can simultaneously reduce the influence of outliers while intelligently weighting inliers. PROBE reduces the need to develop finely-tuned uncertainty models for complex sensors such as cameras, and better accounts for the effects observed in complex, dynamic scenes than typical fixed-uncertainty models. While we present PROBE in the context of visual feature-based navigation, we stress that it is not limited to visual measurements and could also be applied to other sensor modalities.

The aim of PROBE is to learn a model for the quality of visual features, with the goal of

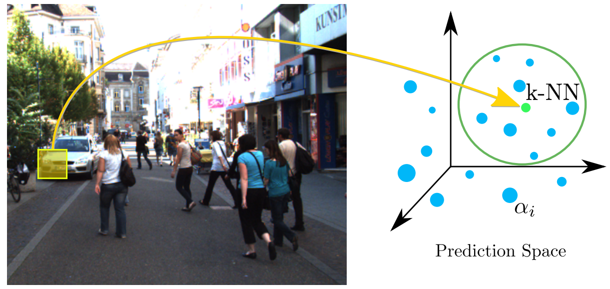

Figure 1.2: PROBE maps image features into a prediction space to predict feature quality (α). Feature quality is a function of the nearest neighbours from training data.

reducing the impact of deleterious visual effects such as moving objects, motion blur, and shadows on navigation estimates. Feature quality is characterized by a scalar weight, β_i, for each visual feature in an environment. To compute β_i we define a prediction space (similar to Vega-Brown et al. (2013)) that consists of a set of visual-inertial predictors computed from the local image region around the feature and the inertial state of the vehicle (Section 1.4.3 details our choice of predictors). We then scale the image covariance of each feature ($\mathbf{R}_a^i$, $\mathbf{R}_b^i$ in **??**) by β_i during the non-linear optimization.

In a similar manner to M-estimation, PROBE achieves robustness by varying the influence of certain measurements. However, in contrast to robust cost functions that weight measurements based purely on estimation error, PROBE weights measurements based on their assessed quality.

To learn the model, we require training data that consists of a traversal through a typical environment with some measure of ground truth for the path, but not for the visual features themselves. Like many machine learning techniques, we assume that the training data is representative of the test environments in which the learned model will be used.

We learn the quality of visual features *indirectly* through their effect on navigation estimates. We define high quality features as those that result in estimates that are close to ground truth. Our framework is flexible enough that we do not require ground truth at every image and we can learn the model based on even a single loop closure error.

1.4.1 Training

Training proceeds by traversing the training path, selecting a subset of visual features at each step, and using them to compute an incremental position estimate. By comparing the estimated position to the ground truth position, we compute the translational Root Mean Squared Error (RMSE), denoted by $\alpha_{l,s}$ for iteration l and step s, and store it at each feature's position in the prediction space (we denote the set of predictors and associated RMSE value by $\Theta_{l,s}$). The full algorithm is summarized in Figure 1.3. Note that $\alpha_{l,s}$ can be computed at each step, at intermittent steps, or for an entire path, depending on the availability of ground truth data.

```
procedure TRAINPROBE()
    for l ← 1, N_iter do
        for s ← 1, N_path do
            f_1, ..., f_J ← visualFeatureSubset(l)
            π_l^1, ..., π_l^J ← predictors(f_1, ..., f_J)
            C̄_ba, r̄_a^ba ← poseChange(f_1, ..., f_J)
            α_{l,s} ← computeRMSE(r̄_a^ba, r_a^ba_GT)
            Θ_{l,s} ← {π_{l,s}^1, ..., π_{l,s}^J, α_{l,s}}
        end for
    end for
    return Θ = {Θ_{l,s}}
end procedure
```

Figure 1.3: The PROBE training procedure.

1.4.2 Evaluation

To use the PROBE model in a test environment, we compute the location of each observed visual feature in our prediction space, and then compute its relative weight β_i as a function of its K nearest neighbours in the training set. For efficiency, the K nearest neighbours are found using a k-d tree. The final scaling factor β_i is a function of the mean of the α values corresponding to the K nearest neighbours, normalized by $\overline{\alpha}$, the mean α value of the entire training set.

```
procedure TESTPROBE(Θ)
    for i ← 1, N_feat do
        π_i ← predictors(f_i)
        α_1, ..., α_K ← findKNN(π_i, K, Θ)
        β_i ← ( (1/(ᾱK)) Σ_{k=1}^{K} α_k )^γ
    end for
    return β = {β_i}
end procedure
```

Figure 1.4: The PROBE evaluation procedure.

The value of K can be determined through cross-validation, and in practice depends on the size of the training set and the environment. The computation of β_i is designed to map small differences in learned α values to scalar weights that span several orders of magnitude. An appropriate value of γ can be found by searching through a set range of candidate values and choosing the value that minimizes the average RMSE (ARMSE) on the training set.

1.4.3 Prediction Space

A crucial component of our technique is the choice of prediction space. In practice, feature tracking quality is often degraded by a variety of effects such as motion blur, moving objects, and textureless or self-similar image regions. The challenge is in determining predictors that account for such effects without requiring excessive computation. In our implementation, we use the following predictors, but stress that the choice of predictors can be tailored to suit particular applications and environments:

- Angular velocity and linear acceleration magnitudes
- Local image entropy
- Blur (quantified by the blur metric of Crete et al. (2007))
- Optical flow variance score
- Image frequency composition

We discuss each of these predictors in turn.

Angular velocity and linear acceleration

While most of the predictors in our system are computed directly from image data, the magnitudes of the angular velocities and linear accelerations reported by the IMU are in themselves good predictors of image degradation (e.g., image blur) and hence poor feature tracking.

Local image entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture in an image or patch. Since the quality of feature detection is strongly influenced by the strength of the texture in the vicinity of the feature point, we expect the entropy of a patch centered on the feature to be a good predictor of its quality. We evaluate the entropy S in an image patch by sorting pixel intensities into N bins and computing

$$S = -\sum_{i=1}^{N} c_i \log_2(c_i), \tag{1.1}$$

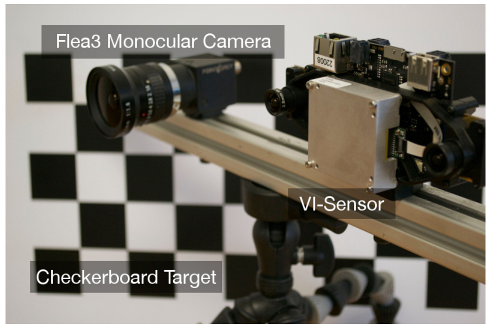

Figure 1.5: The Skybotix VI-Sensor, Point Grey Flea3, and checkerboard target used in our motion blur experiments.

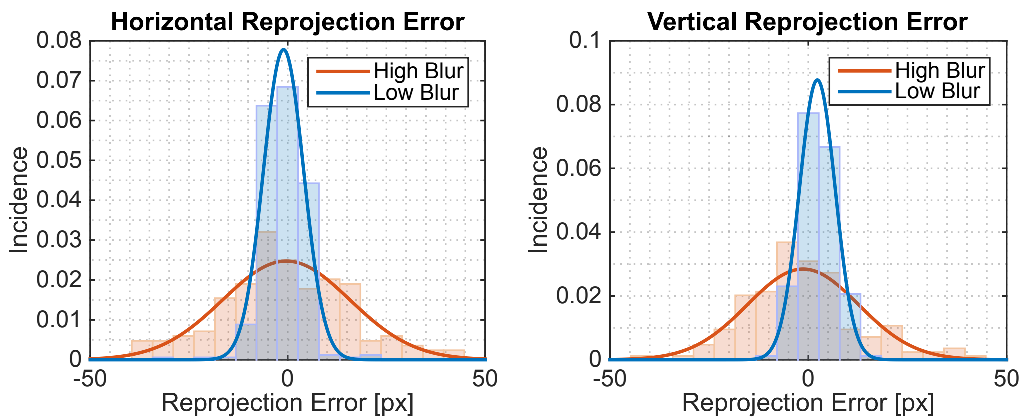

Figure 1.6: Reprojection error of checkerboard corners triangulated from the VI-Sensor and reprojected into the Flea3.

where c_i is the number of pixels counted in the i^{th} bin.

Blur

Blur can arise from a number of sources including motion, dirty lenses, and sensor defects. All of these have deleterious effects on feature tracking quality. To assess the effect of blur in detail, we performed a separate experiment. We recorded images of 32 interior corners of a standard checkerboard calibration target using a low frame-rate (20 FPS) Skybotix VI-Sensor stereo camera and a high frame-rate (125 FPS) Point Grey Flea3 monocular camera rigidly connected by a bar (Figure 1.5). Prior to the experiment, we determined the intrinsic and extrinsic calibration parameters of our rig using the KALIBR[1] package Furgale et al. (2013). The apparatus underwent both slow and fast translational and rotational motion, which induced different levels of motion blur as quantified by the blur metric proposed by Crete et al. (2007).

We detected checkerboard corners in each camera at synchronized time steps, computed

[1] https://github.com/ethz-asl/kalibr

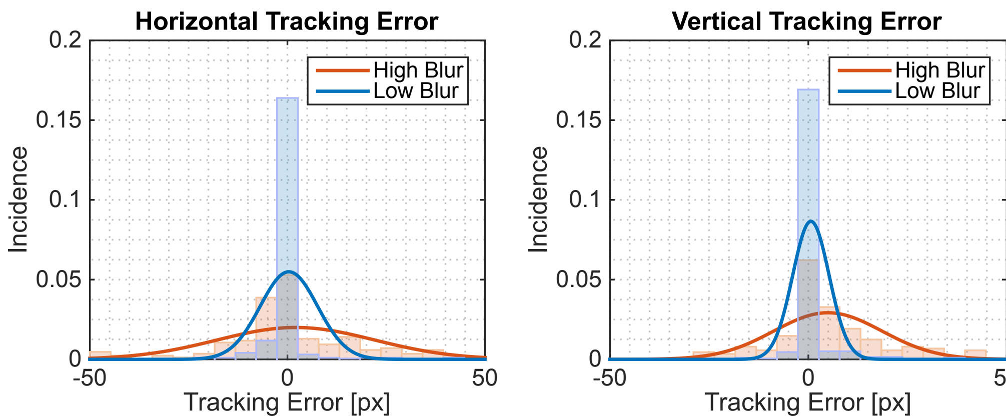

Figure 1.7: Effect of blur on reprojection and tracking error for the slow-then-fast checkerboard dataset. We distinguish between high and low blur by thresholding the blur metric Crete et al. (2007). The variance in both errors increases with blur.

their 3D coordinates in the VI-Sensor frame, then reprojected these 3D coordinates into the Flea3 frame. We then computed the reprojection error as the distance between the reprojected image coordinates and the true image coordinates in the Flea3 frame. Since the Flea3 operated at a much higher frame rate than the VI-Sensor, it was less susceptible to motion blur and so we treated its observations as ground truth. We also computed a tracking error by comparing the image coordinates of checkerboard corners in the left camera of the VI-Sensor computed from both KLT tracking Lucas and Kanade (1981) and re-detection.

Figure 1.7 shows histograms and fitted normal distributions for both reprojection error and tracking error. From these distributions we can see that the errors remain approximately zero-mean, but that their variance increases with blur. This result is compelling evidence that the effect of blur on feature tracking quality can be accounted for by scaling the feature covariance matrix by a function of the blur metric.

Optical flow variance score

To detect moving objects, we compute a score for each feature based on the ratio of the variance in optical flow vectors in a small region around the feature to the variance in flow vectors of a larger region. Intuitively, if the flow variance in the small region differs significantly from that in the larger region, we might expect the feature in question to belong to a moving object, and we would therefore like to trust the feature less. Since we consider only the variance in optical flow vectors, we expect this predictor to be reasonably invariant to scene geometry.

We compute this optical flow variance score according to

$$\log\left(\frac{\bar{\sigma}_s^2}{\bar{\sigma}_l^2}\right), \tag{1.2}$$

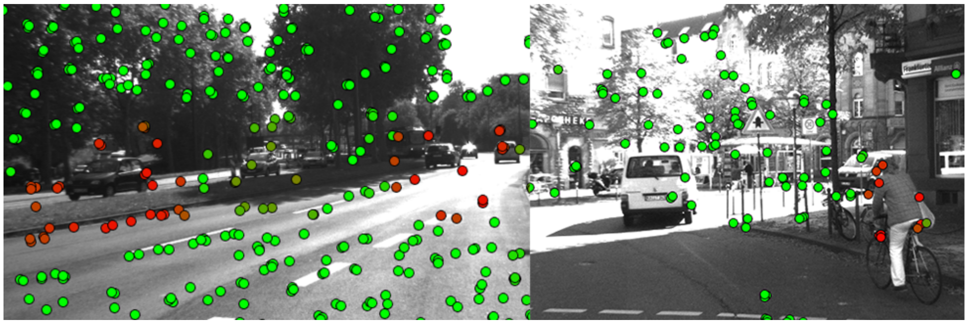

Figure 1.8: The optical flow variance predictor can help in detecting moving objects. Red circles correspond to higher values of the optical flow variance score (i.e., features more likely to belong to a moving object).

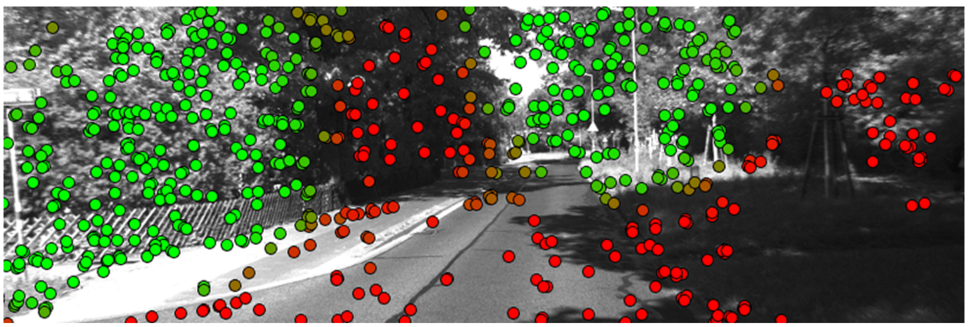

Figure 1.9: A high-frequency predictor can distinguish between regions of high and low texture such as foliage and shadows. Green indicates higher values.

where $\bar{\sigma}_s^2, \bar{\sigma}_l^2$ are the means of the variance of the vertical and horizontal optical flow vector components in the small and large regions respectively. Figure 1.8 shows sample results of this scoring procedure for two images in the KITTI dataset. Our optical flow variance score generally picks out moving objects such as vehicles and cyclists in diverse scenes.

Image frequency composition

Reliable feature tracking is often difficult in textureless or self-similar environments due to low feature counts and false matches. We detect textureless and self-similar image regions by computing the Fast Fourier Transform (FFT) of each image and analyzing its frequency composition. For each feature, we compute a coefficient for the low- and high-frequency regimes of the FFT. Figure 1.9 shows the result of the high-frequency version of this predictor on a sample image from the KITTI dataset. Our high-frequency predictor effectively distinguishes between textureless regions (e.g., shadows and roads) and texture-rich regions (e.g., foliage).

Table 1.1: Comparison of translational Average Root Mean Square Error (ARMSE) and Final Translational Error on the KITTI dataset.

			Nominal RANSAC (99% outlier rejection)		Aggressive RANSAC (99.99% outlier rejection)		PROBE	
Trial	Type	Path Length	ARMSE	Final Error	ARMSE	Final Error	ARMSE	Final Error
26_drive_0051	City [1]	251.1 m	4.84 m	12.6 m	3.30 m	8.62 m	3.48 m	8.07 m
26_drive_0104	City [1]	245.1 m	0.977 m	4.43 m	0.850 m	3.46 m	1.19 m	3.61 m
29_drive_0071	City [1]	234.0 m	5.44 m	30.3 m	5.44 m	30.4 m	3.03 m	12.8 m
26_drive_0117	City [1]	322.5 m	2.29 m	9.07 m	2.29 m	9.07 m	2.76 m	9.08 m
30_drive_0027	Residential [1,†]	667.8 m	4.22 m	12.2 m	4.30 m	10.6 m	3.64 m	4.57 m
26_drive_0022	Residential [2]	515.3 m	2.21 m	3.99 m	2.66 m	6.09 m	3.06 m	4.99 m
26_drive_0023	Residential [2]	410.8 m	1.64 m	8.20 m	1.77 m	8.27 m	1.71 m	8.13 m
26_drive_0027	Road [3]	339.9 m	1.63 m	8.75 m	1.63 m	8.65 m	1.40 m	7.57 m
26_drive_0028	Road [3]	777.5 m	4.31 m	16.9 m	3.72 m	13.1 m	3.92 m	13.2 m
30_drive_0016	Road [3]	405.0 m	4.56 m	19.5 m	3.33 m	14.6 m	2.76 m	13.9 m
UTIAS Outdoor	Snowy parking lot	302.0 m	7.24 m	10.1 m	7.02 m	10.6 m	6.85 m	6.09 m
UTIAS Indoor	Lab interior	32.83 m	–	0.854 m	–	0.738 m	–	0.617 m

[1] Trained using sequence 09_26_drive_0005. [2] Trained using sequence 09_26_drive_0046. [3] Trained using sequence 09_26_drive_0015.
[†] This residential trial was evaluated with a model trained on a sequence from the city category because of several moving vehicles that were better represented in that training dataset.

Figure 1.10: Three types of environments in the KITTI dataset, as well as 2 types of environments at the University of Toronto. We use one trial from each category to train and then evaluate separate trials in the same category.

Figure 1.11: Our four-wheeled skid-steered Clearpath Husky rover equipped with Skybotix VI-Sensor and Ashtech DGPS antenna used to collect the outdoor UTIAS dataset.

1.5 Generalized Kernels

However, not all features are created equal; most feature-based methods rely on random sample consensus algorithms (Fischler and Bolles, 1981) to partition the extracted features into inliers and outliers, and perform estimation based only on inliers. It is common to guard against misclassifying an outlier as an inlier by using robust estimation techniques, such as the Cauchy costs employed in Kerl et al. (2013) or the dynamic covariance scaling devised by Agarwal et al. (2013). These approaches, often grouped under the title of M-estimation, aim to maintain a quadratic influence of small errors, while reducing the contribution of larger errors. The robustness and accuracy of feature-based visual odometry often hinges on the tuning of the parameters of inlier selection and robust estimation. Performance can vary significantly from one environment to the next, and most algorithms require careful tuning to work in a given environment.

In this work, we describe a principled, data-driven way to build a noise model for visual odometry. We combine our previous work (Peretroukhin et al., 2015b) on predictive robust estimation (PROBE) with our work on covariance estimation (Vega-Brown and Roy, 2013) to formulate a predictive robust estimator for a stereo visual odometry pipeline. We frame the traditional non-linear least squares optimization problem as a problem of maximum likelihood estimation with a Gaussian noise model, and infer a distribution over the covariance matrix of the Gaussian noise from a predictive model learned from training data. This results in a Student's t distribution over the noise, and naturally yields a robust nonlinear least-squares optimization problem. In this way, we can predict, in a principled manner, how informative each visual feature is with respect to the final state estimate, which allows our approach to intelligently weight observations to produce more accurate odometry estimates. Our pipeline is outlined in Figure **??**.

1.5.1 Predictive noise models for visual odometry

The process described in the previous section employs a fixed noise covariance $\mathbf{R}$. However, not all landmarks are created equal: differing texture gradients can cause feature localization to degrade in predictable ways, and effects like motion blur can lead to landmarks being less informative. If we had a good estimate of the noise covariance for each landmark, we could simply replace the fixed covariance $\mathbf{R}$ with one that varies for each stereo observation, $\mathbf{R}_{i,t}$. Such a predictive model would allows us to better account for observation errors from a diverse set of noise sources, and incorporate information from landmarks that may otherwise be discarded by a binary outlier rejection scheme.

However, estimating these covariances in a principled way is a nontrivial task. Even when we have reasonable heuristic estimates available, it is difficult to guarantee those estimates will be reliable. Instead of relying solely on such heuristics, we propose to learn these image-space noise covariances from data.

We associate with each landmark $\mathbf{y}_{i,t}$ a vector of *predictors*, $\boldsymbol{\phi}_{i,t} \in \mathbb{R}^M$. Each predictor can be computed using both visual and inertial cues, allowing us to model effects like motion blur and self-similar textures. We then compute the covariance as a function of these predictors, so that $\mathbf{R}_{i,t} = \mathbf{R}(\boldsymbol{\phi}_{i,t})$. In order to exploit conjugacy to a Gaussian noise model, we formulate our prior knowledge about this function using an inverse Wishart (IW) distribution over positive definite $d \times d$ matrices (the IW distribution has been used as a prior on covariance matrices in other robotics and computer vision contexts, see for example, (Fitzgibbon et al., 2007)). This distribution is defined by a scale matrix $\boldsymbol{\Psi} \in \mathbb{R}^{d\times d} \succ 0$ and a scalar quantity called the degrees of freedom $\nu \in \mathbb{R} > d-1$:

$$
\begin{aligned}
p\left(\mathbf{R}\right) &= \mathrm{IW}\left(\mathbf{R}; \boldsymbol{\Psi}, \nu\right) \\
&= \frac{|\boldsymbol{\Psi}|^{\nu/2}}{2^{\frac{\nu d}{2}}\Gamma_d(\frac{\nu}{2})} |\mathbf{R}|^{-\frac{\nu+d+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Psi}\mathbf{R}^{-1}\right)\right).
\end{aligned}
\tag{1.3}
$$

We use the scale matrix to encode our prior estimate of the covariance, and the degrees of freedom to encode our confidence in that estimate. Specifically, if we estimate the covariance $\mathbf{R}$ associated with predictor $\boldsymbol{\phi}$ to be $\hat{\mathbf{R}}$ with a confidence equivalent to seeing n independent samples of the error from $\mathcal{N}(\mathbf{0}, \hat{\mathbf{R}})$, we would choose $\nu(\boldsymbol{\phi}) = n$ and $\boldsymbol{\Psi}(\boldsymbol{\phi}) = n\hat{\mathbf{R}}$.

Given a sequence of observations and ground truth transformations,

$$
\mathcal{D} = \{\mathcal{I}_t, \mathbf{T}_t\}, \quad t \in [1, N] \tag{1.4}
$$

where

$$
\mathcal{I}_t = \{\mathbf{y}_{i,t}, \mathbf{y}'_{i,t}, \boldsymbol{\phi}_{i,t}\} \quad i \in [1, N_t], \tag{1.5}
$$

we can use the procedure of generalized kernel estimation (Vega-Brown et al., 2014) to infer a posterior distribution over the covariance matrix $\mathbf{R}_*$ associated with some query predictor vector $\boldsymbol{\phi}_*$:

$$
\begin{aligned}
p(\mathbf{R}_*|\mathcal{D}, \boldsymbol{\phi}_*) \propto \prod_{i,t} &\mathcal{N}(\mathbf{e}_{i,t}|\mathbf{0}, \mathbf{R}_*)^{k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t})} \\
&\times \mathrm{IW}(\mathbf{R}_*; \boldsymbol{\Psi}(\boldsymbol{\phi}_*), \nu(\boldsymbol{\phi}_*)) &(1.6)\\
= \mathrm{IW}(\mathbf{R}_*; &\boldsymbol{\Psi}_*, \nu_*). &(1.7)
\end{aligned}
$$

Here, $\mathbf{e}_{i,t} = \mathbf{y}'_{i,t} - f(\mathbf{T}_t f^{-1}(\mathbf{y}_{i,t}))$ as before. The function $k : \mathbb{R}^M \times \mathbb{R}^M \to [0, 1]$ is a kernel function which measures the similarity of two points in predictor space. Note also that the posterior parameters $\mathbf{\Psi}_*$ and ν_* can be computed in closed form as

$$\mathbf{\Psi}_* = \mathbf{\Psi}(\boldsymbol{\phi}_*) + \sum_{i,t} k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t})\mathbf{e}_{i,t}\,{\mathbf{e}_{i,t}}^T, \tag{1.8}$$

$$\nu_* = \nu(\boldsymbol{\phi}_*) + \sum_{i,t} k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t}). \tag{1.9}$$

If we marginalize over the covariance matrix, we find that the posterior predictive distribution is a multivariate Student's t distribution:

$$p(\mathbf{y}'_{i,t}|\mathbf{T}_t, \mathbf{y}_{i,t}, \mathcal{D}, \boldsymbol{\phi}_{i,t}) \tag{1.10}$$

$$= \int \mathrm{d}\mathbf{R}_{i,t}\mathcal{N}\left(\mathbf{e}_{i,t}; \mathbf{0}, \mathbf{R}_{i,t}\right) \mathrm{IW}(\mathbf{R}_{i,t}; \mathbf{\Psi}_*, \nu_*) \tag{1.11}$$

$$= \mathrm{t}_{\nu_*-d+1}\left(\mathbf{e}_{i,t}; \mathbf{0}, \frac{1}{\nu_* - d + 1}\mathbf{\Psi}_*\right) \tag{1.12}$$

$$= \frac{\Gamma\left(\frac{\nu_*+1}{2}\right)}{\Gamma\left(\frac{\nu_*-d+1}{2}\right)}|\mathbf{\Psi}_*|^{-\frac{1}{2}}\pi^{-\frac{d}{2}}\left(1 + {\mathbf{e}_{i,t}}^T\,{\mathbf{\Psi}_*}^{-1}\mathbf{e}_{i,t}\right)^{-\frac{\nu_*+1}{2}}. \tag{1.13}$$

Given a new landmark and predictor vector, we can infer a noise model by evaluating eqs. (1.8) and (1.9). In order to accelerate this computation, it is helpful to choose a kernel function with finite support: that is, $k(\boldsymbol{\phi}, \boldsymbol{\phi}') = 0$ if $\|\boldsymbol{\phi} - \boldsymbol{\phi}'\|_2 > \rho$. Then, by indexing our training data in a spatial index such as a k-d tree, we can identify the subset of samples relevant to evaluating the sums in eqs. (1.8) and (1.9) in $\mathcal{O}(\log N + \log N_t)$ time. Algorithm 1 describes the procedure for building this model.

Algorithm 1 Build the covariance model given a sequence of observations, $\mathcal{D}$.

function BUILDCOVARIANCEMODEL($\mathcal{D}$)
  Initialize an empty spatial index $\mathcal{M}$
  for all $\mathcal{I}_t, \mathbf{T}_t$ in $\mathcal{D}$ **do**
    for all $\{\mathbf{y}_{i,t}, \mathbf{y}'_{i,t}, \boldsymbol{\phi}_{i,t}\}$ in $\mathcal{I}_t$ **do**
      $\mathbf{e}_{i,t} = \mathbf{y}'_{i,t} - f(\mathbf{T}_t f^{-1}(\mathbf{y}_{i,t}))$
      Insert $\boldsymbol{\phi}_{i,t}$ into $\mathcal{M}$ and store $\mathbf{e}_{i,t}$ at its location
    end for
  end for
  return $\mathcal{M}$
end function

Once we have inferred a noise model for each landmark in a new image pair, the maximum likelihood optimization problem is given by

$$\mathbf{T}_t^* = \underset{\mathbf{T}_t \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} (\nu_{i,t} + 1) \log\left(1 + {\mathbf{e}_{i,t}}^T \mathbf{\Psi}_{i,t}^{-1} \mathbf{e}_{i,t}\right). \tag{1.14}$$

The final optimization problem thus emerges as a nonlinear least squares problem with a rescaled Cauchy-like loss function, with error term ${\mathbf{e}_{i,t}}^T \left(\frac{1}{\nu_{i,t}+1}\mathbf{\Psi}_{i,t}\right)^{-1}\mathbf{e}_{i,t}$ and outlier scale $\nu_{i,t}+1$. This is a common robust loss function which is approximately quadratic in the reprojection error for ${\mathbf{e}_{i,t}}^T \mathbf{\Psi}_{i,t}^{-1}\mathbf{e}_{i,t} \ll \nu_{i,t}+1$, but grows only logarithmically for ${\mathbf{e}_{i,t}}^T \mathbf{\Psi}_{i,t}^{-1}\mathbf{e}_{i,t} \gg \nu_{i,t}+1$. It follows that in the limit of large $\nu_{i,t}$—in regions of predictor space where there are many relevant samples—our optimization problem becomes the original least-squares optimization problem.

Solving nonlinear optimization problems with the form of Equation (1.14) is a well-studied and well-understood task, and software packages to perform this computation are readily available. Algorithm 2 describes the procedure for computing the transform between a new image pair, treating the optimization of Equation (1.14) as a subroutine.

Algorithm 2 Compute the transform between two images, given a set, $\mathcal{I}_t$, of landmarks and predictors extracted from an image pair and a covariance model $\mathcal{M}$.

---

function COMPUTETRANSFORM($\mathcal{I}_t$, $\mathcal{M}$)
  for all $\{\mathbf{y}_{i,t}, \mathbf{y}'_{i,t}, \boldsymbol{\phi}_{i,t}\}$ in $\mathcal{I}_t$ **do**
    $\mathbf{\Psi}, \nu \leftarrow$ INFERNOISEMODEL($\mathcal{M}$, $\boldsymbol{\phi}_{i,t}$)
    $g(\mathbf{T}) = \mathbf{y}_{i,t} - f(\mathbf{T} f^{-1}(\mathbf{y}'_{i,t}))$
    $\mathcal{L} \leftarrow \mathcal{L} + (\nu+1)\log\left(1 + g(\mathbf{T})^T \mathbf{\Psi}^{-1} g(\mathbf{T})\right)$
  end for
  return $\operatorname{argmin}_{\mathbf{T}\in\mathrm{SE}(3)} \mathcal{L}(\mathbf{T})$
end function
function INFERNOISEMODEL($\mathcal{M}$, $\boldsymbol{\phi}_*$)
  NEIGHBORS $\leftarrow$ GETNEIGHBORS($\mathcal{M}$, $\boldsymbol{\phi}_*$, ρ)
                  ▷ ρ is the radius of the support of the kernel k
  $\mathbf{\Psi}_* \leftarrow \mathbf{\Psi}(\boldsymbol{\phi}_*)$
  $\nu_* \leftarrow \nu(\boldsymbol{\phi}_*)$
  for $(\boldsymbol{\phi}_{i,t}, \mathbf{e}_{i,t})$ in NEIGHBORS **do**
    $\mathbf{\Psi}_* \leftarrow \mathbf{\Psi}_* + k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t})\mathbf{e}_{i,t}\,{\mathbf{e}_{i,t}}^T$
    $\nu_* \leftarrow \nu_* + k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t})$
  end for
  return $\mathbf{\Psi}_*, \nu_*$
end function

We observe that Algorithm 2 is predictively robust, in the sense that it uses past experiences not just to predict the reliability of a given image landmark, but also to introspect and estimate its own knowledge of that reliability. Landmarks which are not known to be reliable are trusted less than landmarks which look like those which have been observed previously, where "looks like" is defined by our prediction space and choice of kernel.

1.5.2 Inference without ground truth

Algorithm 1 requires access to the true transform between training image pairs. In practice, such ground truth data may be difficult to obtain. In these cases, we can instead formulate a likelihood model $p(\mathcal{D}'|\mathbf{T}_1, \ldots, \mathbf{T}_t)$, where $\mathcal{D}' = \{\mathcal{I}_t\}$ is a dataset consisting only of landmarks and predictors for each training image pair. We can construct a model for future queries by inferring the most likely sequence of transforms for our training images. The likelihood has the following factorized form:

$$p(\mathcal{D}'|\mathbf{T}_{1:T}) \propto \int \prod_{i,t} \mathrm{d}\mathbf{R}_{i,t}\, p(\mathbf{y}'_{i,t}|\mathbf{y}_{i,t}, \mathbf{T}_t, \mathbf{R}_{i,t}) p(\mathbf{R}_{i,t}|\boldsymbol{\phi}_{i,t}, \mathcal{D}, \mathbf{T}_{1:T}). \tag{1.15}$$

We cannot easily maximize this likelihood, since marginalizing over the noise covariances removes the independence of the transforms between each image pair. To render the optimization tractable, we follow previous work (Vega-Brown and Roy, 2013) and formulate an iterative expectation-maximization (EM) procedure. Given an estimate $\mathbf{T}_t^{(n)}$ of the transforms, we can compute the expected log-likelihood conditioned on our current estimate:

$$Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)}) = \int \left(\prod_{i,t} \mathrm{d}\mathbf{R}_{i,t}\, p(\mathbf{R}_{i,t}|\mathcal{D}_{\setminus i,t}, \mathbf{T}_{1:T}^{(n)}) \right) \log \prod_{i,t} p(\mathbf{y}'_{i,t}|\mathbf{y}_{i,t}, \mathbf{T}_t, \mathbf{R}_{i,t}). \tag{1.16}$$

This has the effect of rendering the likelihood of each transform to be estimated independently. Moreover, the expected log-likelihood can be evaluated in closed form:

$$Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)}) \cong -\frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{N_t} \mathbf{e}_{i,t}{}^T \left(\frac{1}{\nu_{i,t}^{(n)}} \boldsymbol{\Psi}_{i,t}^{(n)} \right)^{-1} \mathbf{e}_{i,t}. \tag{1.17}$$

The symbol $\cong$ is used to indicate equality up to an additive constant. A derivation of this observation can be found in our supplemental material.

We can iteratively refine our estimate by maximizing the expected log-likelihood

$$\mathbf{T}_{1:T}^{(n+1)} = \underset{\mathbf{T}_{1:T} \in \mathrm{SE}(3)^T}{\operatorname{argmax}}\, Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)}). \tag{1.18}$$

Due to the additive structure of $Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)})$, this takes the form of T separate nonlinear least-squares optimizations:

$$\mathbf{T}_t^{(n+1)} = \underset{\mathbf{T}_t \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} \mathbf{e}_{i,t}{}^T \left(\frac{1}{\nu_{i,t}^{(n)}} \boldsymbol{\Psi}_{i,t}^{(n)} \right)^{-1} \mathbf{e}_{i,t}. \tag{1.19}$$

Algorithm 3 describes the process of training a model without ground truth. We refer to this process as PROBE-GK-EM, and distinguish it from PROBE-GK-GT (Ground Truth). We note that the sequence of estimated transforms, $\mathbf{T}_{1:T}^{(n)}$, is guaranteed to converge to a local maxima of the likelihood function (Dempster et al., 1977). It is also possible to use a robust loss function (Equation (1.14)) in place of Equation (1.19) during EM training. Although not formally motivated by the derivation above, this approach often leads to lower test errors in practice. Characterizing when and why this robust learning process outperforms its non-robust alternative is part of ongoing work.

Algorithm 3 Build the covariance model without ground truth given a sequence of observations, $\mathcal{D}'$, and an initial odometry estimate $\mathbf{T}_{1:T}^{(0)}$.

---

function BUILDCOVARIANCEMODEL($\mathcal{D}'$, $\mathbf{T}_{1:T}^{(0)}$)
    Initialize an empty spatial index $\mathcal{M}$
    for all $\mathcal{I}_t$ in $\mathcal{D}'$ **do**
        for all $\{\mathbf{y}_{i,t}, \mathbf{y}'_{i,t}, \boldsymbol{\phi}_{i,t}\}$ in $\mathcal{I}_t$ **do**
            $\mathbf{e}_{i,t} = \mathbf{y}_{i,t} - f(\mathbf{T}_t^{(0)} f^{-1}(\mathbf{y}'_{i,t}))$
            Insert $\boldsymbol{\phi}_{i,t}$ into $\mathcal{M}$ and store $\mathbf{e}_{i,t}$ at its location
        end for
    end for
    repeat
        for all $\mathcal{I}_t$ in $\mathcal{D}'$ **do**
            for all $\{\mathbf{y}_{i,t}, \mathbf{y}'_{i,t}, \boldsymbol{\phi}_{i,t}\}$ in $\mathcal{I}_t$ **do**
                $\boldsymbol{\Psi}, \nu \leftarrow$ INFERNOISEMODEL($\mathcal{M}$, $\boldsymbol{\phi}_{i,t}$)
                $g(\mathbf{T}) = \mathbf{y}_{i,t} - f(\mathbf{T} f^{-1}(\mathbf{y}'_{i,t}))$
                $\mathcal{L} \leftarrow \mathcal{L} + g(\mathbf{T})^T \left(\frac{1}{\nu}\boldsymbol{\Psi}\right)^{-1} g(\mathbf{T})$
            end for
            $\mathbf{T}_t \leftarrow \operatorname{argmin}_{\mathbf{T} \in \mathrm{SE}(3)} \mathcal{L}(\mathbf{T})$
            $\mathbf{e}_{i,t} = \mathbf{y}_{i,t} - f(\mathbf{T}_t^{(0)} f^{-1}(\mathbf{y}'_{i,t}))$
            Update the error stored at $\boldsymbol{\phi}_{i,t}$ in $\mathcal{M}$ to $\mathbf{e}_{i,t}$
        end for
    until converged
    return $\mathcal{M}$
end function

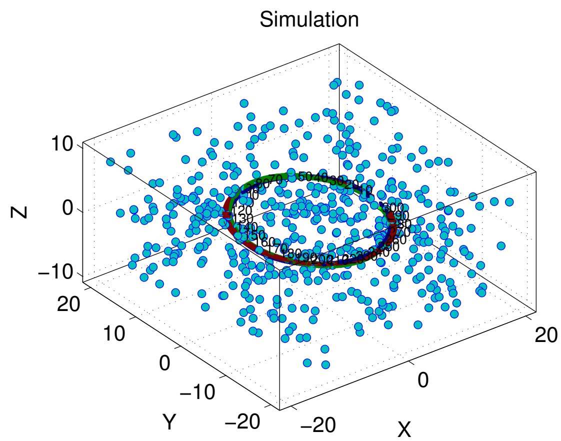

Figure 1.12: Our synthetic world. A stereo camera rig moves through a world with 2000 point features.

1.5.3 Experiments

Synthetic

Next, we formulated a synthetic dataset wherein a stereo camera traverses a circular path observing 2000 randomly distributed point features. We added Gaussian noise to each of the ideal projected pixel co-ordinates for visible landmarks at every step. We varied the noise variance as a function of the vertical pixel coordinate of the feature in image space. In addition, a small subset of the landmarks received an error term drawn from a uniform distribution to simulate the presence of outliers. The prediction space was composed of the vertical and horizontal pixel locations in each of the stereo cameras.

We simulated independent training and test traversals, where the camera moved for 30 and 60 seconds respectively (at a forward speed of 3 metres per second for final path lengths of 90 and 180 meters). Figure 1.13 and Table 1.2 document the qualitative and quantitative comparisons of PROBE-GK (trained with and without ground-truth) against two baseline stereo odometry frameworks. Both baseline estimators were implemented based on **??**. The first utilized fixed covariances for all reprojection errors, while the second used a modified robust cost (i.e. M-estimation) based on Student's t weighting, with $\nu = 5$ (as suggested in Kerl et al. (2013)). These benchmarks served as baseline estimators (with and without robust costs) that used fixed covariance matrices and did not include a predictive component.

Using PROBE-GK with ground truth data for training, we significantly reduced both the translation and rotational Average Root Mean Squared Error (ARMSE) by approximately 50%. In our synthetic data, the Expectation Maximization approach was able to achieve nearly identical results to the ground-truth-aided model within 5 iterations.

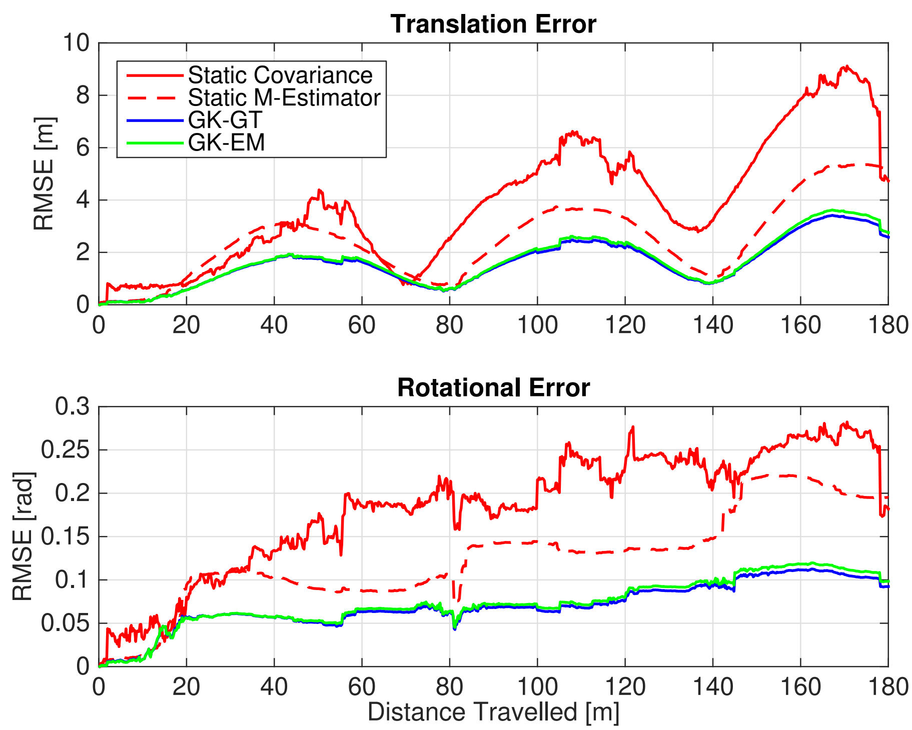

Figure 1.13: A comparison of translational and rotational Root Mean Square Error on simulated data (RMSE) for four different stereo-visual odometry pipelines: two baseline bundle adjustment procedures with and without a robust Student's t cost with a fixed and hand-tuned covariance and degrees of freedom (M-Estimation), a robust bundle adjustment with covariances learned from ground truth with algorithm 1 (GK-GT), and a robust bundle adjustment using covariances learned without ground truth using expectation maximization, with algorithm 3 (GK-EM). Note in this experiment, the RMSE curves for GK-GT and GK-EM very nearly overlap. The overall translational and rotational ARMSE values are shown in Table 1.2.

Figure 1.14: The KITTI dataset contains three different environments. We validate PROBE-GK by training on each type and testing against a baseline stereo visual odometry pipeline.

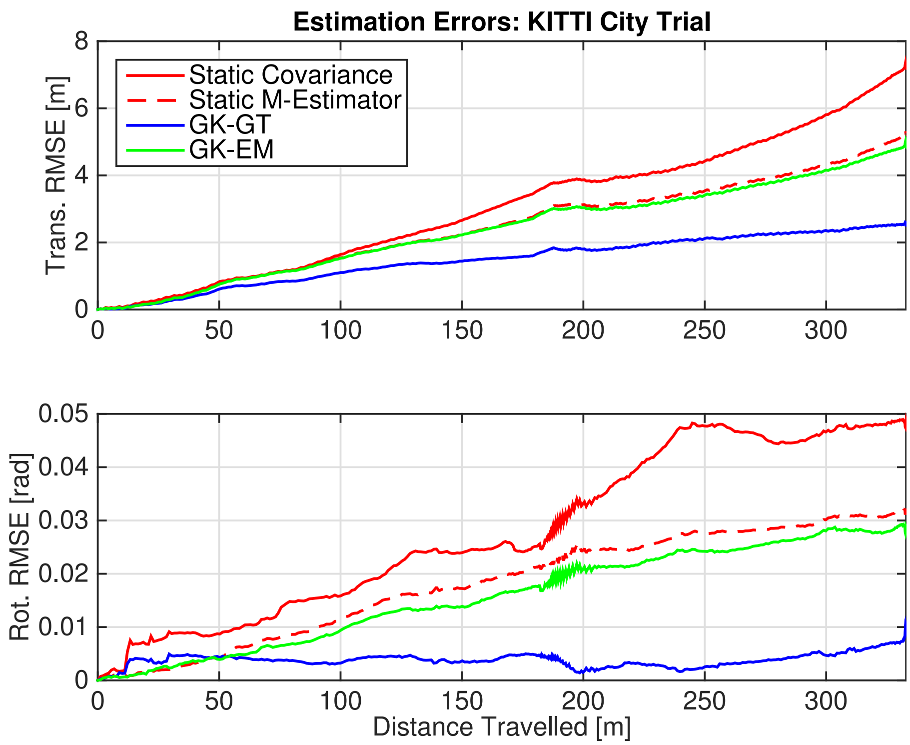

Figure 1.15: RMSE comparison of stereo odometry estimators evaluated on data from the city category in the KITTI dataset. See Table 1.2 for a quantitative summary.

1.5.4 KITTI

To evaluate PROBE-GK on real environments, we trained and tested several models on the KITTI Vision Benchmark suite (Geiger et al., 2012, 2013), a series of datasets collected by a car outfitted with a number of sensors driven around different parts of Karlsruhe, Germany. Within the dataset, ground truth pose information is provided by a high grade inertial navigation unit which also fuses measurements from differential GPS. Raw data is available for different types of environments through which the car was driving; for our work, we focused on the city, residential and road categories (Figure 1.14). From each category, we chose two separate trials for training and testing.

Our prediction space consisted of inertial magnitudes, high and low image frequency coefficients, image entropy, pixel location, and estimated transform parameters. The choice of predictors is motivated by the types of effects we wish to capture (in this case: grassy self-

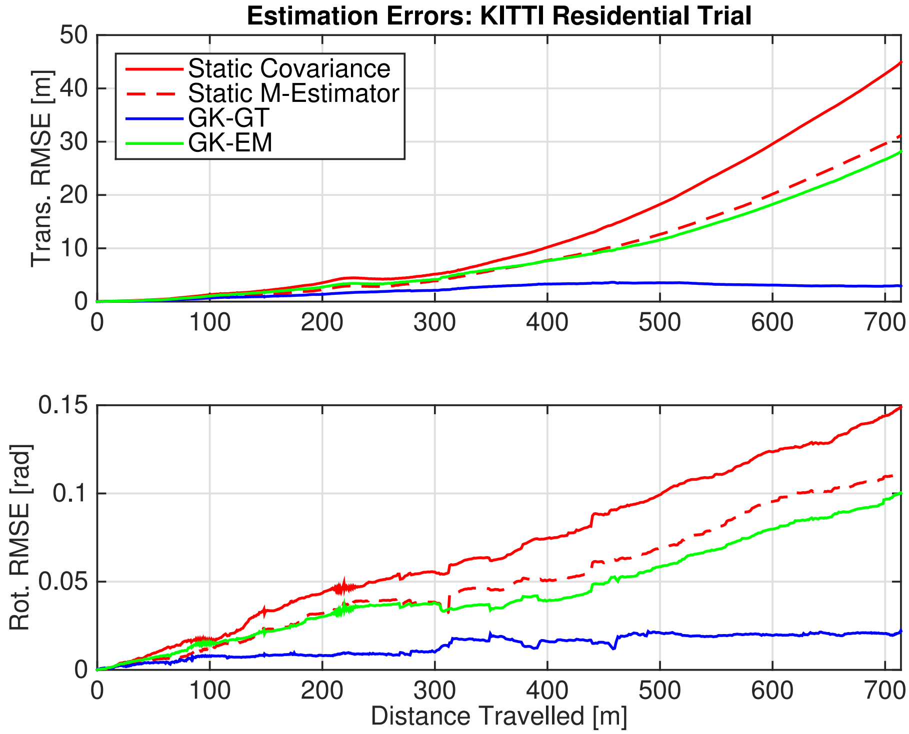

Figure 1.16: RMSE comparison of stereo odometry estimators evaluated on data from the residential category in the KITTI dataset.

similar textures, as well as shadows, and motion blur). For a more detailed explanation of our choice of prediction space, see our previous work (Peretroukhin et al., 2015b).

Figures 1.15 to 1.17 show typical results; **??** presents a quantitative comparison. PROBE GK-GT produced significant reductions in ARMSE, reducing translational ARMSE by as much as 80%. In contrast, GK-EM showed more modest improvements; this is unlike our synthetic experiments, where both GK-EM and GK-GT achieved similar performance. We are still actively exploring why this is the case; we note that although our simulated data is drawn from a mixture of Gaussian distributions, the underlying noise distribution for real data may be far more complex. With no ground truth, EM has to jointly optimize the camera poses and sensor uncertainty. It is unclear whether this is feasible in the general case with no ground truth information.

Further, we observe that the performance of PROBE-GK depends on the similarity of the training data to the final test trials. A characteristic training dataset was important for consistent improvements on test trials.

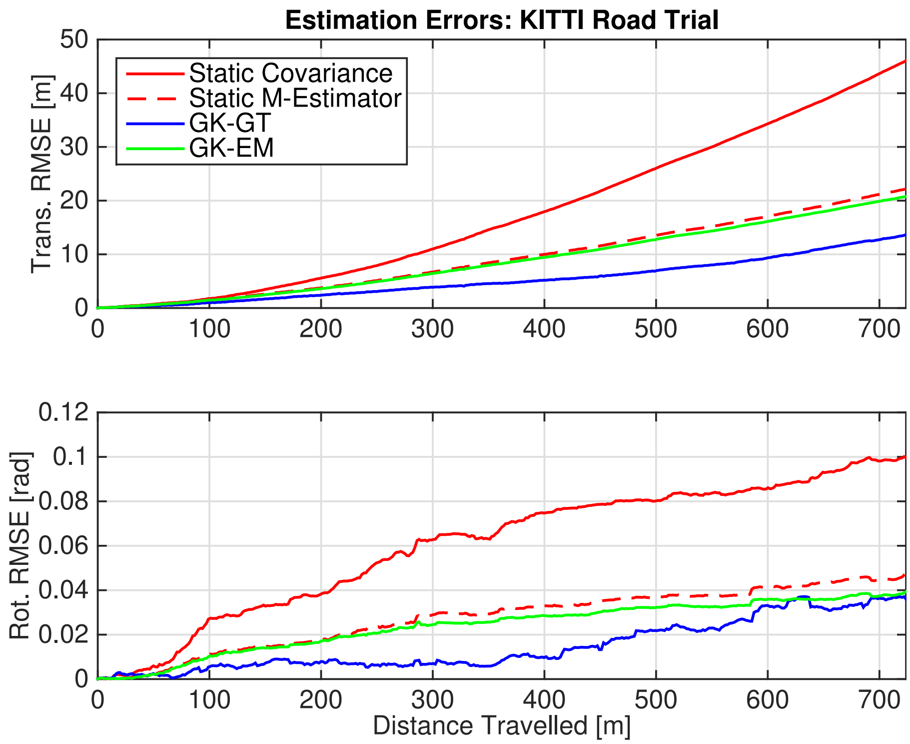

Figure 1.17: RMSE comparison of stereo odometry estimators evaluated on data from the road category in the KITTI dataset.

Table 1.2: Comparison of average root mean squared errors (ARMSE) for rotational and translational components. Each trial is trained and tested from a particular category of raw data from the synthetic and KITTI datasets.

		Trans. ARMSE [m]				Rot. ARMSE [rad]			
	Length [m]	Fixed Covar.	Static M-Estimator	GK-GT	GK-EM	Fixed Covar.	Static M-Estimator	GK-GT	GK-EM
Synthetic	180	3.87	2.49	1.59	1.66	0.18	0.13	0.070	0.073
City	332.9	3.84	2.99	1.69	2.87	0.032	0.021	0.0046	0.018
Residential	714.1	13.48	9.37	1.97	8.80	0.068	0.050	0.013	0.044
Road	723.8	17.69	9.38	5.24	8.87	0.060	0.027	0.015	0.024

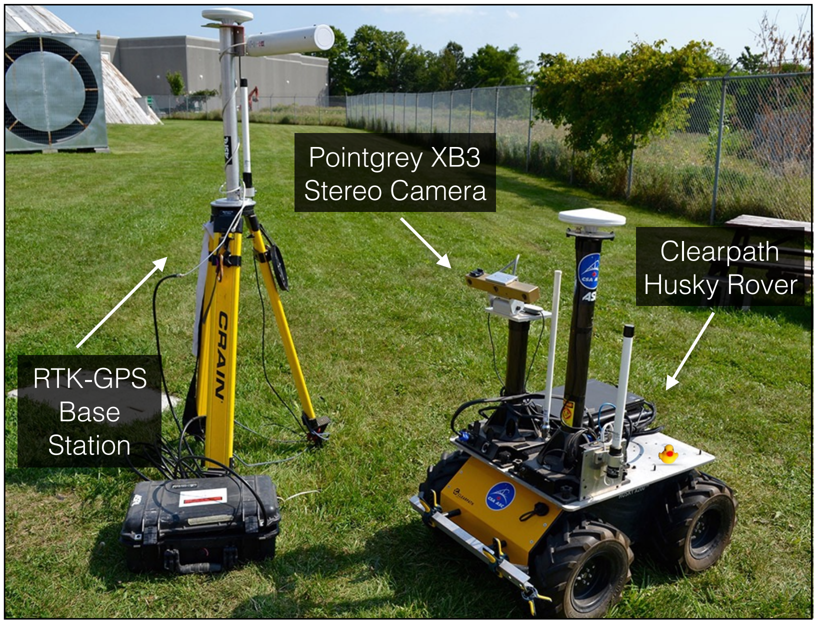

Figure 1.18: Our experimental apparatus: a Clearpath Husky rover outfitted with a PointGrey XB3 stereo camera and a differential GPS receiver and base station.

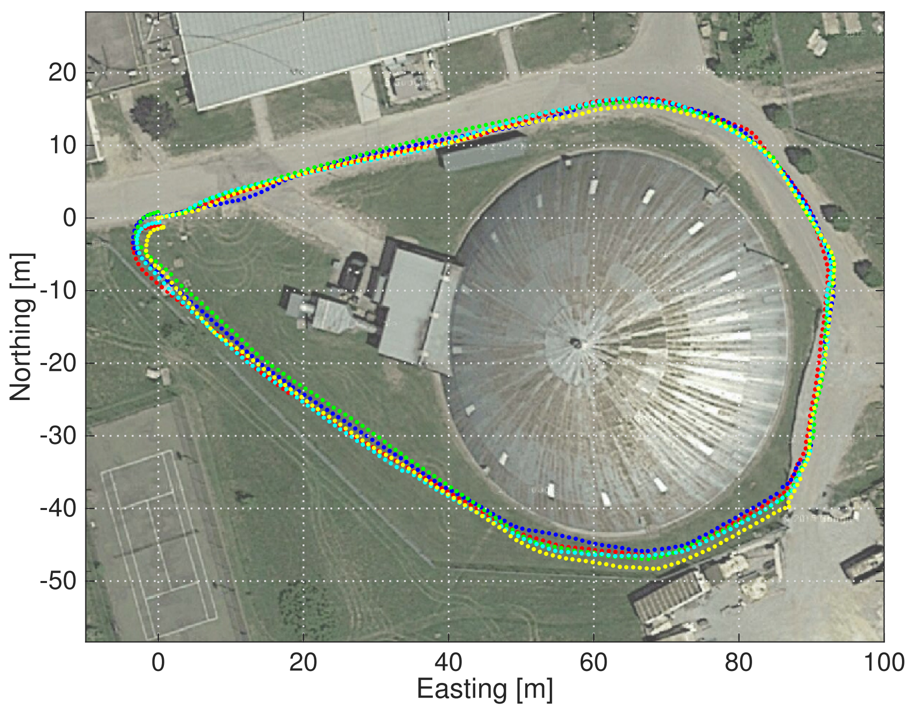

Figure 1.19: GPS ground truth for 5 experimental trials collected near the UTIAS Mars Dome. Each trial is approximately 250 m long.

Table 1.3: Comparison of loop closure errors for 4 different experimental trials with and without a learned PROBE-GK-EM model.

		Loop Closure Error [m]	
Trial	Path Length [m]	PROBE-GK-EM	Static M-Estimator
2	250.3	3.88	8.07
3	250.5	3.07	6.64
4	205.4	2.81	7.57
5	249.9	2.34	7.75

UTIAS

To further investigate the capability of our EM approach, we evaluated PROBE-GK on experimental data collected at the University of Toronto Institute for Aerospace Studies (UTIAS). For this experiment, we drove a Clearpath Husky rover outfitted with an Ashtech DG14 Differential GPS, and a PointGrey XB3 stereo camera around the MarsDome (an indoor Mars analog testing environment) at UTIAS (Figure 1.18) for five trials of a similar path. Each trial was approximately 250 m in length and we made an effort to align the start and end points of each loop. We used the wide baseline (25 cm) of the XB3 stereo camera to record the stereo images. The approximate trajectory for all 5 trials, as recorded by GPS, is shown in Figure 1.19. Note that the GPS data was not used during training, and only recorded for reference.

For the prediction space in our experiments, we mimicked the KITTI experiments, omitting inertial magnitudes as no inertial data was available. We trained PROBE-GK without ground truth, using the Expectation Maximization approach. Figure 1.20 shows the likelihood and loop closure error as a function of EM iteration.

The EM approach indeed produced significant error reductions on the training dataset after just a few iterations. Although it was trained with no ground truth information, our PROBE-GK model was used to produce significant reductions in the loop closure errors of the remaining 4 test trials. This reinforced our earlier hypothesis: the EM method works well when the training trajectory more closely resembles the test trials (as was the case in this experiment). Table 1.3 lists the statistics for each test.

1.6 Summary

Predictive Robust Estimation applied two different techniques (scalar weighted covariances and the method of generalized kernel estimation) to improve on the uncorrelated and static Gaussian error models typically employed in stereo odometry. PROBE and its follow up PROBE-GK, contributed

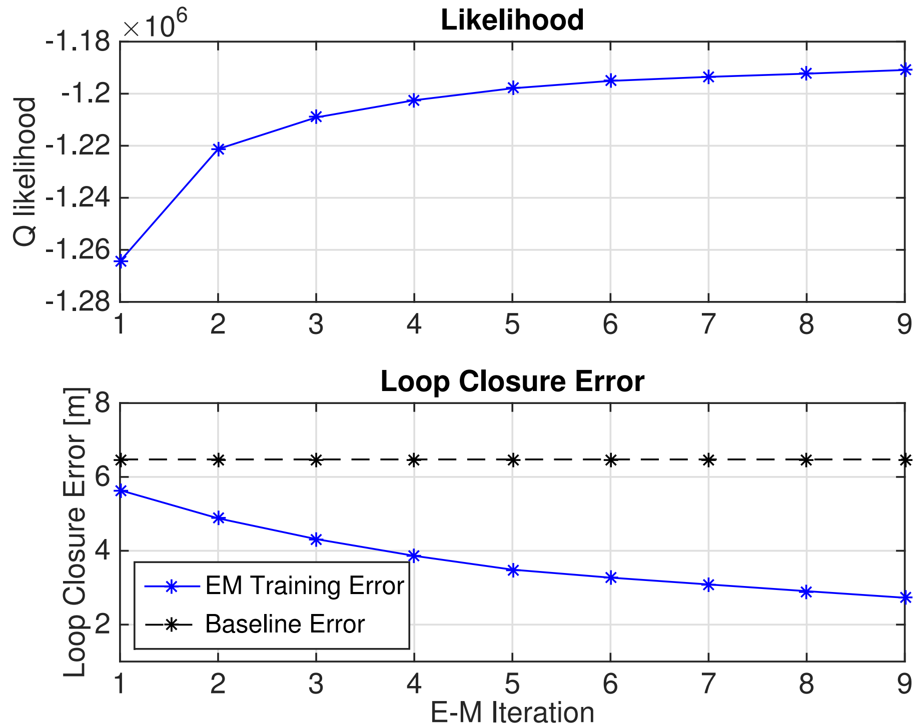

Figure 1.20: Training without ground truth using PROBE-GK-EM on a 250.2m path around the Mars Dome at UTIAS. The likelihood of the data increases with each iteration, and the loop closure error decreases, improving significantly from a baseline static M-estimator.

1. a probabilistic model for indirect stereo visual odometry, leading to a predictive robust algorithm for inference on that model,

2. two different approaches to constructing the robust algorithm: one based on k-nearest neighbours, and one based on Generalized Kernel (GK) estimation,

3. a procedure for training our model using pairs of stereo images with known relative transforms, and

4. an iterative, expectation-maximization approach to train our GK model when the relative ground truth egomotion was unavailable.

Appendices

Bibliography

Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W. (2013). Robust map optimization using dynamic covariance scaling. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 62–69.

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.

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.

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.

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.

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., 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.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *Int. Journal Robot. Research (IJRR)*.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Irani, M. and Anandan, P. (2000). About direct methods. In *Vision Algorithms: Theory and Practice*, pages 267–277. Springer.

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.

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.

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., Giamou, M., and Kelly, J. (2015b). PROBE: Predictive robust estimation for visual-inertial navigation. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pages 3668–3675.

Peretroukhin, V., Clement, L., and Kelly, J. (2015c). 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., 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.

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

Sünderhauf, N. and Protzel, P. (2007). Stereo odometry: a review of approaches. *Chemnitz University of Technology Technical Report.*

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., Bachrach, A., Bry, A., Kelly, J., and Roy, N. (2013). CELLO: A fast algorithm for covariance estimation. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3160–3167.

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 *Advances in Neural Information Processing Systems 27*, pages 2546–2554.

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