Introspective Evaluation of Perception Performance for Parameter Tuning without Ground Truth

Humphrey Hu[†] and George Kantor[†]

***Abstract*— Modern perception systems are notoriously complex, featuring dozens of interacting parameters that must be tuned to achieve good performance. Conventional tuning approaches require expensive ground truth, while heuristic methods are difficult to generalize. In this work, we propose an introspective ground-truth-free approach to evaluating the performance of a generic perception system. By using the posterior distribution estimate generated by a Bayesian estimator, we show that the expected performance can be estimated efficiently and without ground truth. Our simulated and physical experiments in a demonstrative indoor ground robot state estimation application show that our approach can order parameters similarly to using a ground-truth system, and is able to accurately identify top-performing parameters in varying contexts. In contrast, baseline approaches that reason only about observation log-likelihood fail in the face of challenging perceptual phenomena.**

I. Introduction

As robots are deployed into an ever-wider variety of tasks, environments, and situations, so will the generality and robustness of their capabilities be tested. In this way, the booming success of robotics is at the same time its greatest challenge as the field struggles with the transition from lab-grade to safety-critical.

Of the many core robot competencies, the one perhaps most challenged by this growing requirement of generality is perception [1]. In some ways this is natural, as perception systems typically provide task-specific interpretations of reality. However, even for a relatively constrained and controlled task like the Amazon Picking Challenge, subtle changes in the environment and situation, which we refer to as the "context", can result in perceptual failure [2].

One source of this brittleness is the highly parametric nature of modern perceptual components. Hardware and software components alike boast numerous parameters that must be specified or "tuned" to achieve good performance in a particular context. For instance, a visual odometry algorithm may require looser outlier thresholds when operating in cluttered outdoor environments and tighter outlier thresholds in clean indoor environments. This affords a great deal of flexibility in adapting components, but also means that an improperly-tuned but otherwise well-designed perception system may perform poorly.

Ideally, perception systems would be tuned upon deployment to a new context, or when the context changes. This is realistic for some applications, for instance factory automation, where the context is unlikely to change and a high initial deployment cost is acceptable. In many applications, however, this "tune on deployment" approach is impractical. As an example, self-driving cars will likely regularly experience contextual changes that affect their perception performance as they drive, and adding a costly tuning maintenance procedure that must be performed often is unlikely to be popular with the consumer market.

The high cost of perception tuning stems primarily from its steep supervisory requirements for evaluating performance, e.g. human-annotated data [3]–[5] or precision instrumentation [6]–[9]. These external sources of feedback which are not available during normal operation are known as "ground truth". There exist heuristic approaches for certain systems which do not rely on ground truth, for instance visual odometry [10], [11], but these are difficult to extend. We desire a way to evaluate perception performance that is generalizable and does not rely on ground-truth.

In this work we present a theoretically-motivated approach to introspectively evaluate the performance of a generic perception system without ground truth. We refer to our approach as "introspective" as it reasons about performance by considering the perception belief state instead of relying on supervision. This has the benefit of being able to take advantage of the posterior distributions produced by many commonly-used Bayesian inference algorithms, for instance Kalman filters, with little to no modifications to an existing system.

We rigorously validate our approach on a state estimation application with simulated and hardware experiments. In our simulations, we test various state and parameter-dependent sensor models, and in our physical experiments we test a laser-based and vision-based odometry system in a variety of environments. Our results show that our proposed introspective approach produces evaluations that are highly-correlated with the ground-truth evaluations, even in the face of challenging perceptual phenomena that cause baseline approaches to fail.

II. Prior Work

Techniques for evaluating perception systems can largely be grouped into a few categories, which we review here.

A. Ground Truth Evaluation

Sources of ground truth vary by application, but can largely be split into human, instrumentation, or by-design. Manual ground truth relies on humans to produce the expected result of an algorithm, for instance, identifying human

[†] Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Email: {humhu, kantor}@cmu.edu

poses in images [3], aligning planar laser scans [4], or choosing image feature correspondences [5].

Instrumentation-based ground truth uses precise measurements for supervision, and is typically used in estimation applications. Motion capture systems can provide excellent pose data, but are generally restricted to limited indoor spaces [6], [7]. Automated survey equipment [8], augmented GPS/INS systems [9] are popular for outdoor applications, and more recently detailed urban maps [12] have been used as well.

Perhaps the least common, but most practical source of ground truth is that which exists automatically, typically achieved through use of simulation or data augmentation. In [13] the authors use a photorealistic simulation to study a visual odometry algorithm. Others have used laser scans collected at the same location but artificially perturbed to test a scan-matching algorithm [14]. Our work differs even from these approaches that do not rely on external ground truth, as we reason over distributions of outcomes instead of point instances.

B. Heuristic Evaluation

Heuristic approaches use contextual information or quantities correlated with performance in place of ground truth. Contextual heuristics use domain information or assumptions to provide supervision, and are commonly applied to classification-type systems. The self-supervised river segmentation approach detailed in [15] assumes that the river lays below the horizon to generate training data during operation. Another example can be found in [16] where scale and ground-plane information are used to find hard-negative pedestrian detection examples.

Estimation systems often use quantities that are correlated with performance as heuristics. A common heuristic in point-based visual odometry systems is the number of matched features, as these algorithms generally rely on having many correspondences. This heuristic is used in [17] and [18] where the authors report the number of correspondences as a substitute for localization performance. Other works studying visual odometry performance in adverse conditions show correlations between image quantities, e.g. brightness, sharpness, blur, with the number of correspondences [10], [11]. Unlike these methods, our approach is theoretical in nature.

C. Statistical Metrics

In the filtering and modeling communities, statistical metrics are sometimes used in addition to ground-truth evaluations. Works on SLAM, for instance, often report the chi-squared test as a measure of solution quality [19] or the optimization objective itself, which is related to the joint observation likelihood [20]. Similarly, observation likelihood is used as an optimization objective in [21] to select Kalman filter parameters. The authors in [22] report the normalized estimate entropy for their unscented Kalman filtering approach, which reflects the state tracking and observation prediction quality. Our work also utilizes a statistical formulation, but offers a clearer motivation and connection to perceptual performance.

D. Introspection

Recently the concept of "introspection" has been introduced in the robotics community to refer to an algorithm's ability to assign an appropriate level of confidence in its output. One work reasons about classification output uncertainty by considering distributions of models [23]. This is similar to the older concept of filtering optimism, a condition wherein a filter becomes overly confident in its estimate [24].

Other introspective approaches do not seek to modify algorithms to produce confidences, but instead to directly predict the algorithm performance. Much of this work has been done on vision systems, for instance predicting segmentation or horizon detection failures [25], or traversability estimation failures [26]. Other work has shown prediction of the heuristic performance of a vision-based navigation system [17], [18], and of a classification system [27].

Our work is similar to [23] in that it uses distributions of hypotheses to introspect performance without ground truth. However, our approach operates on the posterior distribution over latents and observations directly, as opposed to over model hypotheses. In addition, our focus is on evaluating performance for changing parameters, not on the introspective capacity of our inference algorithm.

III. Quantifying Perceptual Performance

We begin by formalizing the notion of performance for a perception system that estimates an unknown quantity, the *latent* $x \in X$, from available data, the *observations* $\xi = \{z_i \in Z\}$. Depending on the application, the latent x may take on different forms. For instance, in state estimation it will typically be a sequence of states, whereas in classification it may be a labeling of a set of inputs.

Now let $f_\theta(\xi, \hat{x}_0) = \hat{x} \in X$ represent the latent estimate $\hat{x}$ generated from observations ξ and prior $\hat{x}_0$ with perception parameters $\theta \in \Theta$. In this work we assume that the prior $\hat{x}_0$ is fixed and write the estimator function as $f_\theta(\xi)$ for notational simplicity. We quantify performance of an estimate compared to ground truth by defining a loss $\ell(\cdot, \cdot) : X \times X \rightarrow \Re$. The loss is typically application-dependent. For instance, square loss is common in estimation, while hinge loss is more common in classification. Similar to most statistical learning work, we define the performance of the system as the expected loss for all possible latents and observations:

$$L = E_{x,\xi}\left[\ell(x, f_\theta(\xi))\right] \tag{1}$$

It is often prohibitively difficult to model the generative distribution $p(x, \xi)$ required to evaluate the expected loss in Eq. 1. As such, we rely on the *empirical loss*, its Monte Carlo approximation:

$$\hat{L} = \frac{1}{N} \sum_{i=1}^{N} \ell\left(x_i, f_\theta(\xi_i)\right) \tag{2}$$

where ground truth x_i and observations ξ_i correspond to independent executions of the system.

A. Evaluating Performance with the Posterior Distribution

To show how we can remove the requirement for ground truth in Eq. 2, we first return to the expected loss in Eq. 1. We can decompose the generative distribution into a product of conditionals, and accordingly, the expectation into nested expectations:

$$p(x, \xi) = p(\xi) p(x|\xi) \tag{3}$$

$$E_{x,\xi}\left[\ell(x, f_\theta(\xi))\right] = E_\xi E_{x|\xi}\left[\ell(x, f_\theta(\xi))\right] \tag{4}$$

The first conditional in Eq. 3 is the true *posterior estimate distribution* $p(x|\xi)$. Bayesian inference techniques such as the ubiquitous Kalman filter provide an estimate of this distribution, which we will denote as $\hat{p}(x|\xi)$. It may be possible to adapt non-Bayesian techniques, for instance SVMs for classification, by introducing introspection as in [23].

Assuming that an estimate of the posterior is available, we propose to use it to compute the inner expectation of Eq. 4 without requiring ground truth. The outer expectation can then be evaluated empirically:

$$\tilde{L} = \frac{1}{N} \sum_{i=1}^{N} \hat{E}_{x|\xi}\left[\ell(x, f_\theta(\xi_i))\right] \tag{5}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \int \hat{p}(x|\xi_i) \ell(x, f_\theta(\xi_i)) dx \tag{6}$$

where $\hat{E}$ denotes an expectation taken with an approximate distribution. We refer to this quantity as the *approximate posterior error*, or APE. It can be interpreted simply as penalizing uncertainty in the latent with regards to the loss function. As such, the APE relies on having an accurate posterior distribution estimate.

B. Application to Kalman filter SSE loss

For certain choices of posterior distribution type and loss function the inner expectation of the APE can be computed in closed form. One common choice of loss function is the sum of squares, or sum square error loss (SSE), defined as:

$$\ell_{SSE}(x, \hat{x}) = (x \ominus \hat{x})^T (x \ominus \hat{x}) \tag{7}$$

where $\ominus$ computes a vector difference between elements of X. Substituting Eq. 7 into the expectation in Eq. 5, we obtain:

$$E_{x|\xi}\left[\ell_{SSE}(x, f_\theta(\xi))\right] = E_{x|\xi}\left[(x \ominus \hat{x})^T (x \ominus \hat{x})\right] \tag{8}$$

$$= E_{x|\xi}\left[\operatorname{tr}\left((x \ominus \hat{x})(x \ominus \hat{x})^T\right)\right] \tag{9}$$

$$= \operatorname{tr}\left(E_{x|\xi}\left[(x \ominus \hat{x})(x \ominus \hat{x})^T\right]\right) \tag{10}$$

which is simply the trace of the $\hat{x}$-centered second moment of the posterior distribution. This quantity is easily computable when using many common estimation algorithms, such as Kalman or particle filters. For the common choice of choosing the posterior mean as the estimate, or $\hat{x} = E_{x|\xi}[x]$, Eq. 10 becomes the trace of the mean-centered second moment, the covariance.

C. Adaptive Kalman Filtering

It should be mentioned that in a standard Kalman filter the estimate covariance does not involve the observation themselves, evolving purely as a function of the system model, *i.e.*, the transition and observation models. One way, then, to introspect is to re-identify appropriate system model parameters whenever the system behavior changes, *e.g.*, with an EM procedure [28], whenever the context or perception parameters change. However, such an approach would likely be data-intensive and complex.

If we assume that only the transition and observation covariances change as a function of the context and perception parameters, we can instead opt to use the much simpler adaptive Kalman filter (AKF) to estimate the system parameters online. Let $x_t^{(-)}$ and $x_t^{(+)}$ denote the estimate mean before and after performing an update, respectively, with a similar notation for the estimate covariance $P_t^{(-)}$ and $P_t^{(+)}$. Further define the transition function Jacobian as F_t and the observation function Jacobian as H_t, with the observation at time t written as y_t. Using the approach detailed in [29], the online estimates of the transition covariance Q_t and observation covariance R_t are:

$$Q_{t+1} = \frac{1}{W_Q} \sum_{\tau=t-W_Q+1}^{t} \Delta x_\tau \Delta x_\tau^T \tag{11}$$

$$R_{t+1} = \frac{1}{W_R} \sum_{\tau=t-W_R+1}^{t} \nu_\tau \nu_\tau^T + H_t P_t^+ H_t^T \tag{12}$$

where W_Q and W_R are sliding window lengths, $\Delta x_t = x_t^{(+)} - x_t^{(-)}$) is the *state correction*, and $\nu_t = y_t - \hat{y}_t^{(+)}$ is the post-update measurement prediction error, often referred to as the *residual*. Intuitively, Eqs. 11 and 12 adjust the covariances to match the observation prediction errors during execution: When the state evolves predictably, the state corrections will be small, resulting in a small Q. Similarly, when the observations are well-predicted, the residuals will be small, resulting in a small R.

We note that the AKF estimates rely on an assumption of uncorrelated transition and observation noise to be meaningful. When the noise is systematic or heavily correlated in time, *e.g.*, calibration errors or software bugs, the estimated covariances may not be accurate, resulting in poor introspection.

IV. Experimental Validation

We validate our proposed approach with simulated and hardware experiments in evaluating the performance of an indoor ground robot body velocity estimation system. The AKF is well-suited for introspecting performance in this setting, as the perception parameters and context primarily affect the observation noise magnitude.

Our overall procedure consists of executing a large number of open loop trajectories on the robot with different perception parameters and contexts. We then compare ground-truth evaluations against our proposed evaluation approach and

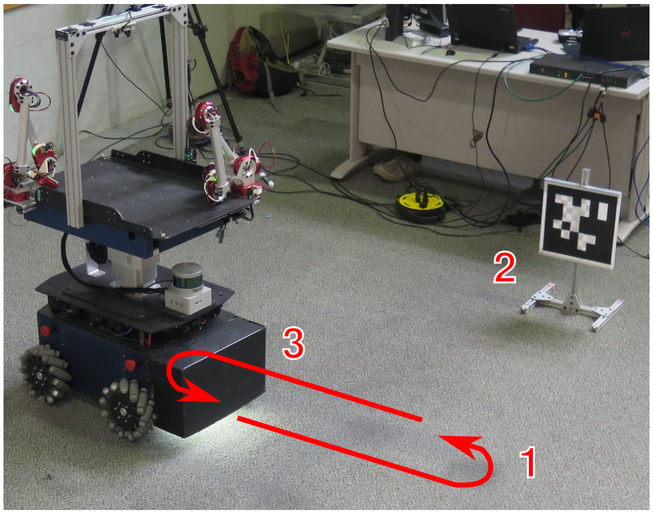


Fig. 1: An illustration of the system execution procedure. 1.) The robot drives an open-loop trajectory consisting of 0.5 meters straight forward, followed by a 180° turn. 2.) The robot resets its pose using a side-facing camera to observe a fiducial. 3.) The robot can then execute its next open-loop trajectory.

other baseline approaches. To maintain a constant level of complexity, we only consider six numerical parameters for each system and normalize them to a domain of $[-1, 1]$.

Both the simulated and physical experimental systems track the robot body velocity with an AKF as described in Sec. III-C. The AKF covariance estimation buffer lengths W_Q and W_R were heuristically tuned to provide good filtering performance.

A. Simulated Robot System

Our simulated system consists of simple point mass second-order dynamics and Gaussian acceleration noise and simulated body velocity observations corrupted by zero-mean Gaussian noise. We use heuristically-motivated sensor models that combine the perception parameters, represented as real-valued vector θ, and the robot speed v with a single "hardness factor" κ:

$$\kappa = \|v\|_2 \cdot \|\theta\|_2^2 \tag{13}$$

This hardness relation reflects that, in our experience, deviating from the optimal parameters θ^* for a particular environment increases the effects of higher speeds on performance, and vice-versa. In our experiments we arbitrarily set $\theta^* = 0$ and test three sensor models:

1) ***DN****: Dependent Noise:* This model captures perception difficulty as sensor noise covariance R that increases exponentially with hardness:

$$R = R_0 + \tilde{R}\left[1 - \exp(k_R \cdot \kappa)\right] \tag{14}$$

where $k_R > 0$ so that R achieves its minimum value R_0 at $\kappa = 0$ and increases with increasing hardness. The sensor rate f is fixed at 200 Hz, and we use constants $R_0 = 1E{-}6I$, $\tilde{R} = 0.25I$, $k_R = 0.25$.

2) ***DNR****: Dependent Noise and Rate:* This model builds upon the DN model by additionally having the sensor rate exponentially decrease with hardness:

$$f = f_0 + \tilde{f}\exp(k_f \cdot \kappa) \tag{15}$$

where $k_f < 0$ so that f achieves its maximum value $f_0 + \tilde{f}$ at $\kappa = 0$ and decreases with increasing hardness to a minimum of f_0. We use the same constants as the Dn model with additionally $f_0 = 20$, $\tilde{f} = 180$, and $k_f = -1.0$, corresponding to a maximum rate of 200Hz.

3) ***DNR+C****: Dependent Noise and Rate with Cutout:* This model modifies the DNR model by modeling a phenomena where the sensor will fail or "cut out" above a certain hardness κ_{max} and not produce any observations:

$$f = \begin{cases} 0, & \kappa > \kappa_{max} \\ f_0 + \tilde{f}\exp(k_f \cdot \kappa), & o/w \end{cases} \tag{16}$$

We use the same constants as the DNR model with additionally $\kappa_{max} = 1.5$.

B. Physical Robot System

The physical robot, shown in Fig. 1, is an indoor ground robot with an omnidirectional "mecanum" wheel drivetrain, a visual odometry system using a downward-facing camera, and a laser odometry system using two planar lidars. Both odometry systems output body velocity observations and are intended to run in parallel, but we use one at a time in our experiments, giving us two test systems. All perception software ran on an embedded Intel NUC box with an Intel Core i5-4250U and 16GB of RAM. A Vicon motion capture system provided ground truth poses for the physical robot at 200 Hz, which we downsampled to 10 Hz and numerically differentiated to compute ground truth body velocities.

The visual odometry (VO) system uses a single downward-facing IDS UI-3140CP USB 3.0 camera capturing 400×400 resolution frames at 200 frames per second. We employ high-intensity lighting under the robot with low camera exposure times to minimize the effect of motion blur. We perform Lucas-Kanade tracking on a regularly-spaced set of points in the images to find correspondences. We then estimate the 2D rigid displacement of the camera assuming all points are on a plane parallel to the camera and differentiate it to estimate the body velocity. Our software uses the OpenCV library's implementation of Lucas-Kanade with pyramids and rigid transformation estimation[1]. The six VO parameters we tune in our experiments are described in Tab. I.

The laser odometry (LO) system uses two Hokuyo URG-04LX-UG01 planar lidars. Each lidar scans 240° at 10 Hz with a maximum range of 5.6m. Sequential scans coming from a lidar are first preprocessed by a voxel filter and then registered to one another, with the resulting transformation differentiated to produce a body velocity estimate. We use the Point Cloud Library (PCL) implementation of Iterative Closest Point (ICP)[2], which features a large number of

[1] http://opencv.org/

[2] http://pointclouds.org/

TABLE I: Physical system parameters and ranges considered for tuning.

Parameter	Type	Values
Laser Odometry Parameters		
Voxel filter width	float	$\in [0.01, 0.2]$
ICP max iterations	int	$\in [5, 100]$
ICP max corresp. distance	int	$\in [0, 1]$
ICP max solution error	int	$\in [0.01, 1.0]$
RANSAC iterations	float	$\in [5, 100]$
RANSAC inlier threshold	float	$\in [0.0, 1.0]$
Visual Odometry Parameters		
Point grid dimension	int	$\in [5, 30]$
LK min solver improvement	float	$\in [10^{-6}, 1.0]$
LK search window	int	$\in [10, 40]$
LK pyramid level	int	$\in [0, 5]$
LK max solution error threshold	float	$\in [0, 7.5]$
RANSAC max error	float	$\in [0.0, 0.05]$

numerical parameters. The six LO parameters we tune in our experiments are described in Table I.

We performed experiments in two environments for each of the two physical odometry systems, for a total of four different physical contexts as shown in Fig. 2:

1) **clear:** LO in a cleared indoor area
2) **clutter:** LO in an area cluttered with traffic cones
3) **carpet:** VO on low-pile speckled carpeting
4) **concrete:** VO on medium-gloss painted concrete floor

C. Evaluation Approaches Tested

We tested our proposed approach against a standard approach that uses ground truth, and two baseline approaches that do not use ground truth:

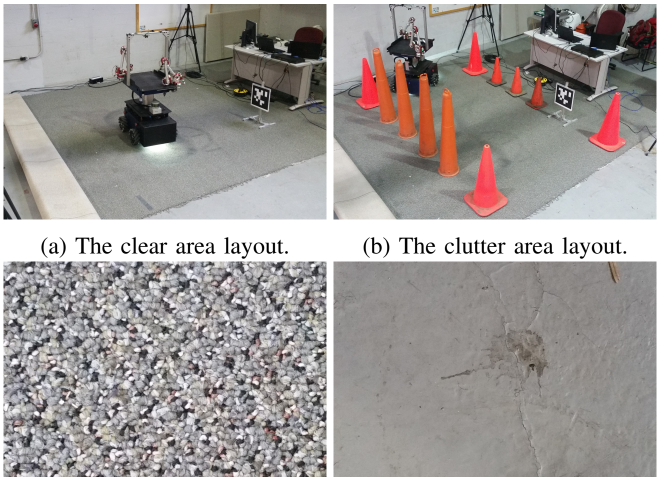
*1) **SSE**: Sum of Squares Error:* The conventional baseline that uses ground truth to compute Eq. 7 for the body velocity error. We numerically integrate the SSE computed at each estimator update to compute the time-averaged SSE over a trajectory. Low SSE corresponds to better performance.

*2) **APE**: Approximate Posterior Error:* Our proposed approach, computed as described in Eq. 10 as the trace of the state estimate covariance. Like the SSE, the APE is computed over a trajectory with numerical integration. Low APE corresponds to better performance.

*3) **SOL**: Sum Observation Log-likelihood:* A baseline that computes the sum of log-likelihoods for all observations received in the trajectory, and thus does not use ground truth. This is equivalent to computing the joint probability of all received observations assuming they are independent. High SOL corresponds to better performance.

*4) **AOL**: Average Observation Log-likelihood:* The second ground-truth free approach that normalizes the SOL by the number of observations received. High AOL corresponds to better performance.

Though SOL and AOL may appear to be scaled versions of each other, they are distinct since the number of observations changes as a function of the perception parameters. Thus, SOL captures the "quantity" of observations while AOL captures the "quality" of observations received. In contrast, the APE captures the estimation "certainty" or "risk", which the SSE measures directly.



(a) The clear area layout.

(b) The clutter area layout.

(c) The carpet floor texture.

(d) The concrete floor texture.

Fig. 2: The test environments for the physical robot experiments.

D. Experimental Procedure

Our experiments consist of executing a large number of trajectories with different perception parameters and contexts. We assume that the executions are independent of each other and generated by a stationary distribution. Thus, we can execute multiple trajectories using the same perception parameters and context, and average their performances to predict the expected performance as in Eqs. 2 and 5.

Each system execution is performed with the following procedure. We note that the robot moves in an open-loop fashion to avoid affecting the perception performance through the closed-loop perception-control dynamics.

1) The perception parameters are set to their test values. Trial data recording begins.
2) The robot remains stationary for one second to initialize the AKF.
3) The robot drives forward at 0.5 m/s for 1.0 m.
4) The robot turns in place at 1.0 rad/s for a half-rotation (π radians).

This motion tests the odometry system performance to both translational and rotational motion and is illustrated in Fig. 1. In the physical experiments, the robot resets its pose every two evaluations using a side-facing camera and a fiducial in the test area to avoid drifting over time, while in simulation the robot is simply allowed to drift.

We collected a dataset for each of the simulated sensor models and physical context described in Secs. IV-A and IV-B. Each dataset was collected by first executing trials for 100 uniformly-randomly sampled parameters from the full normalized parameter range $[-1, 1]$. Each trial consists of 10 system executions, and any trials with executions that did not receive any observations are dropped from final analysis, as we do not have a reasonable way to compute the AOL or SOL for these trials. We then executed trials for 30 parameters uniformly randomly sampled from a range of ± 0.1 centered on the best parameters from the first 100

TABLE II: Summaries of datasets, with lowest trial SSE and corresponding RMS, APE, AOL, and number of observations.

Dataset	Trials Run	Trials Dropped	Best RMS	Corr. APE	Corr. AOL	Corr. Num Obs
Laser Odometry						
clear	100	0	0.127	0.177	3.00	417
clutter	30	0	0.101	0.170	3.18	391
Visual Odometry						
carpet	30	3	0.086	0.029	7.20	3064
concrete	100	31	0.065	0.059	6.47	3077
Simulated						
DN	30	0	0.078	0.017	2.61	4145
DNR	30	0	0.080	0.023	2.59	4137
DNR+C	30	0	0.078	0.020	2.63	4134

TABLE III: Best-performing parameters across contexts.

Laser Odometry Parameter	**clear**	**clutter**
Voxel filter width	0.034	0.01
ICP max iterations	238	111
ICP max corresp. distance	0.198	0.149
ICP max solution error	0.076	0.014
RANSAC iterations	93	72
RANSAC inlier threshold	0.71	0.52
Visual Odometry Parameter	**carpet**	**concrete**
Point grid dimension	20	13
LK min solver improvement	0.283	0.312
LK search window	21	37
LK pyramid level	4	3
LK max solution error threshold	5.87	4.33
RANSAC inlier threshold	1.28×10^{-3}	4.67×10^{-4}

trials. This adds many more near-optimal parameters to the datasets, which are of interest for parameter tuning.

Overall we collected 4 physical datasets for a total of 520 trials and 5200 executions, as well as 3 simulated datasets for another 390 trials and 3900 executions. Statistics summarizing the datasets are presented in Table II. The best parameters for each physical context are shown in Table III.

E. Analysis Performed

We assert that an accurate performance evaluation will order trials similarly to the expected loss in Eq. 1. Since the true expected loss is not known, we approximate it with the empirical loss over the trial's executions, which in our case is the mean SSE. This provides us with a ground-truth ordering of the trials against which we can compare other evaluation approaches.

We perform two types of analysis for each simulated sensor model, each physical system across both contexts, and all physical experiments. This allows us to analyze performance not just over different parameters and contexts, but across physical systems as well.

1) Comparing Rankings with Kendall-τ: Our first analysis uses the Kendall-τ coefficient to quantify the similarity between the ground truth ordering and an ordering generated by another evaluation approach. The Kendall-τ coefficient can be interpreted as the number of bubble-sort insertions required to transform one ordering into the other. As such, making multiple small ordering mistakes produces higher τ values than a single severe ordering mistake.

In these results, we use the τ-b variant implementation in SciPy[3] which normalizes to the number of data and accounts for ties. Since this test is not often used in the robotics literature, we review it briefly here:

Consider a set of joint observations $\{(x_i, y_i)\}$. In our setting, x_i is the trial mean SSE and y_i is a performance evaluation. A pair of observations (x_i, y_i) and (x_j, y_j) with $i \neq j$ is *concordant* if the ordering of observations match, *i.e.*, $x_i > x_j$ and $y_i > y_j$, and *discordant* if the ordering of observations is reversed, *i.e.*, $x_i > x_j$ and $y_i < y_j$. The τ-b coefficient is computed as:

$$\tau = \frac{n_C - n_D}{\sqrt{(n_C + n_D + T_x)(n_C + n_D + T_y)}} \tag{17}$$

where n_C is the number of concordant pairs, n_D is the number of condordant pairs, T_x is the number of ties in the x_i observations, and T_y is the number of ties in the y_i observations. A $\tau = -1$ corresponds to exactly opposite rankings and a $\tau = 1$ corresponds to exactly identical rankings. Thus, in our analysis, approaches that produce τ values near 1 can be understood as accurately replicating the ground-truth ordering across all trials.

The Kendall-τ coefficient allows us to measure the accuracy of a single ordering. Accordingly, we can measure the effect of stochasticity in the executions and the number of executions N per trial on ordering stability. Ideally we would collect a large number of independent datasets, but this is rather impractical, so we instead bootstrap to generate synthetic datasets by sampling executions from each trial. We present results using 100 synthetic datasets with $N = 1$ through $N = 9$, shown in 3.

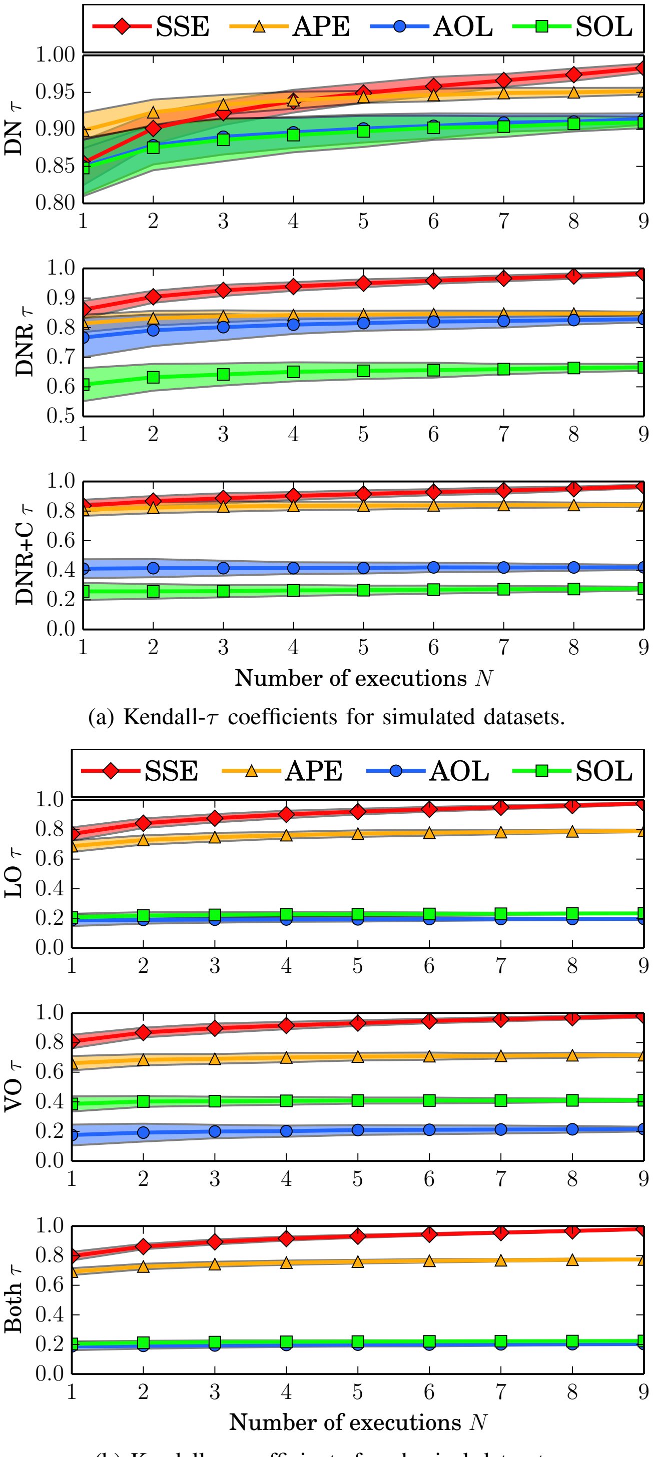
2) Comparing Performance with Dataset Partitions: In parameter tuning, our objective is not to accurately order parameters, but instead to identify parameters that perform well. Our second test captures this desire by looking at the ground-truth performance of what each approach considers to be the top performing parameters. Specifically, we compute the mean SSE over the top $1-r \in [0, 1]$ portion of trials in a dataset, sorted according to a particular evaluation approach. For instance, the APE $r = 0.9$ partition is the average SSE of the top 10% of trials, sorted according to the mean APE. The r-partition results for all approaches are shown in Fig. 4. An approach whose top performers all have low SSE can be considered useful for parameter tuning.

V. Discussion

A. Simulated Results

We observe in Figs. 4a and 3a that both APE and the baselines perform well on the simplest DN sensor model, with only slight differences in the τ performance. However, the baseline performances suffer as the model complexity increases.

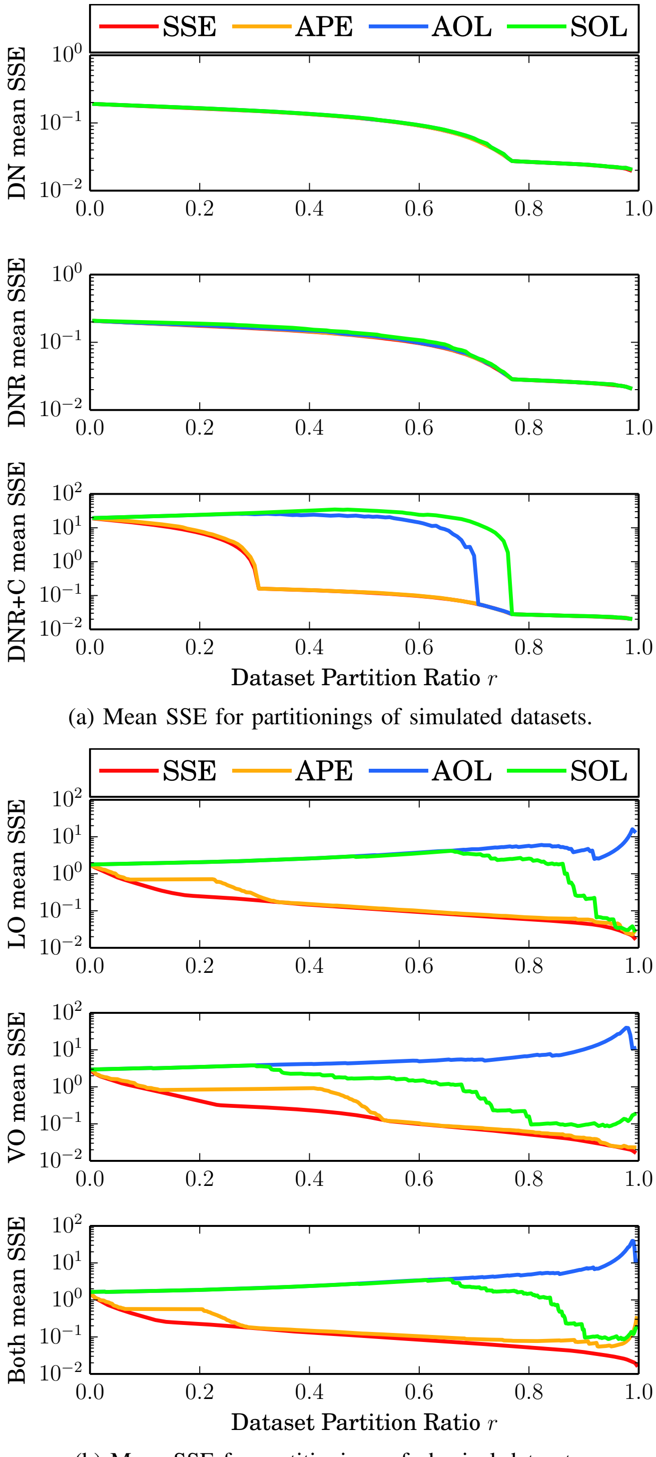
[3] https://www.scipy.org/



(a) Kendall-τ coefficients for simulated datasets.

(b) Kendall-τ coefficients for physical datasets.

Fig. 3: Mean τ coefficient for each approach over 100 bootstrapped dataset subsamplings. Shading denotes ±3 standard deviations. A τ closer to 1 denotes similarity in ordering to using the mean SSE.



(a) Mean SSE for partitionings of simulated datasets.

(b) Mean SSE for partitionings of physical datasets.

Fig. 4: Mean SSE for top-performing r portion of trials, according to each approach. Lower SSE for larger r indicate that an approach is able to identify high-performing parameters.

SOL overall ranking accuracy degrades from varying sensor rate in DNR, and since AOL performs similary on DN and DNR, the drop in SOL's τ is likely due to the number of observations changing across trials. Further, as SOL exhibits good partition performance for DNR, we conclude that SOL's reduced τ is due to a large number of minor ordering errors. This suggests that the effect of variable sensor rate alone on SOL is not catastrophic. APE reasons about the latent directly, and is thus unaffected by changing number of observations, while AOL explicitly normalizes out this effect.

Incorporating sensor cutout results in degraded partition and τ performance for both baseline approaches. This is because cutout removes the highest-noise observations from trials with highly suboptimal parameters, which would have otherwise contributed to decreasing the SOL and AOL. However, trials with near-optimal parameters are less affected, as reflected in the good partition performance for the baselines at $r > 0.7$. In contrast, APE captures the effect of cutout on the latent estimate quality as the AKF covariance grows rapidly in the absence of observations, resulting in good performance across all trials.

Finally we observe that all approaches perform well for large r, accurately identifying the top-performing parameters. This is significantly different from the physical results seen in Fig. 4b, suggesting that our heuristic "hardness"-based models do not capture all of the real-world phenomena that make fine-tuning difficult. Nevertheless, these simulations are useful in highlighting how varying sensor rates and cutout cause the baselines to fail.

B. Physical Results

APE significantly outperforms the baselines in overall ranking for all physical datasets, as seen in Fig. 3b. APE also performs quite well on the partition test for the LO and VO datasets, closely matching the SSE curve after roughly $r > 0.5$. These results show that APE is able to both order parameters overall and accurately identify top-performing parameters across varying contexts.

SOL also achieves good partition performance on LO for $r \approx 1$, and slightly outperforms APE on the combined LO and VO dataset. This may be due to the heuristic nature of how the AKF was tuned compounding the approximately $5\times$ difference in sensor rates, resulting in subpar orderings across systems. However, SOL performs poorly for $r \approx 1$ on VO, possibly due to the camera's high sensing rate resulting in significant variance in the number of observations among the top-performing parameters. In addition, SOL performs poorly at ordering suboptimal parameters, as seen in its low τ and poor performance for $r < 0.8$. We speculate that this may make APE a better choice for use in an online search algorithm, such as Bayesian Optimization, which can use information from suboptimal parameters to guide future search.

In contrast, AOL performs very poorly on all tests for the physical datasets, with τ values near 0.2 and extremely high mean SSE in its top-performing parameters. One likely explanation is extreme cutout occurring from selecting high error threshold parameters: When the robot is stationary, the odometry system outputs very refined, high-likelihood observations, but as the robot begins moving, the odometry systems stop outputting observations due to the high error thresholds. This results in a very high AOL that is difficult to exceed. As such, though AOL performs slightly better than SOL in simulation, it is unsuited to parameter tuning on physical systems where parameters can dramatically change system behavior.

C. Effect of Number of Executions

Figs. 3a and 3b suggest that running more executions per trial improves overall ranking and reduces variance, but does not have a significant effect on the overall performance of each approach past $N = 4$. This is likely due to the highly repetitive nature of our executions. If the executions were carried over disparate parts of a building, at different speeds, or extracted from longer executions, for example, we would expect to see a much stronger effect from N.

VI. Conclusion

We have presented a theoretically-grounded and generalizable approach for evaluating a perception system's performance without relying on ground truth. Our extensive physical and simulated experiments with indoor ground robot velocity estimation highlight the efficacy of our introspective APE approach in ordering parameters closely to their ground truth performance. This suggests that our approach can be practically used to select perception parameters when ground truth is unavailable. In contrast, baseline methods that reason only about observation likelihood perform poorly in our tests due to perception parameters and context affecting the quantity and distribution of data received.

Removing the need for ground truth for perception evaluation enables a number of capabilities. We can consider perception systems that self-tune upon deployment or during operation, allowing them to generalize across multiple contexts and removing the need for expert human supervision. In addition, perception performance information can inform other robot capabilities, such as path planning, allowing robots to avoid situations where their sensors are likely to fail.

To achieve these goals, however, a number of shortcomings must be addressed. First, though we have shown that the AKF's introspective power is sufficient for a simple velocity estimation task, more work must be done to understand how this approach generalizes to different systems and settings. We have also observed instances where introspection fails due to systematic errors violating AKF assumptions. We must be able to detect these conditions and compensate for them, or develop a more powerful introspection approach that can work under a wider range of conditions. Finally, automatic approaches for tuning the AKF parameters should be developed to replace the current heuristic step, removing the final reliance on a human expert.

REFERENCES

[1] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, PP(99):1–17, 2016.

[2] Clemens Eppner, Sebastian Höfer, Rico Jonschkowski, Roberto Martín-Martín, Arne Sieverling, Vincent Wall, and Oliver Brock. Lessons from the amazon picking challenge: Four aspects of building robotic systems. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[3] Nataraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and C. V. Jawahar. *Has My Algorithm Succeeded? An Evaluator for Human Pose Estimators*, pages 114–128. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[4] John G. Rogers, Alexander J. B. Trevor, Carlos Nieto-Granda, Alex Cunningham, Manohar Paluri, Nathan Michael, Frank Dellaert, Henrik I. Christensen, and Vijay Kumar. *Effects of Sensory Precision on Mobile Robot Localization and Mapping*, pages 433–446. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[5] Javier Civera, Oscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel. 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, 2010.

[6] J. Rwekmper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3158–3164, Oct 2012.

[7] Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar. *Initialization-Free Monocular Visual-Inertial State Estimation with Application to Autonomous MAVs*, pages 211–227. Springer International Publishing, Cham, 2016.

[8] Philipp Krsi, Bastian Bcheler, Franois Pomerleau, Ulrich Schwesinger, Roland Siegwart, and Paul Furgale. Lighting-invariant adaptive route following using iterative closest point matching. *Journal of Field Robotics*, 32(4):534–564, 2015.

[9] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[10] Christopher Brunner, Thierry Peynot, and Teresa Vidal-Calleja. Visual metrics for the evaluation of sensor data quality in outdoor perception. *International Journal of Intelligent Control and Systems*, 16(2):142–159, June 2011. Special Edition: Quantifying the Performance of Intelligent Systems.

[11] Christopher Brunner and Thierry Peynot. *Perception Quality Evaluation with Visual and Infrared Cameras in Challenging Environmental Conditions*, pages 711–725. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[12] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo. Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3377–3392, Dec 2015.

[13] Ankur Handa, Richard A. Newcombe, Adrien Angeli, and Andrew J. Davison. *Real-Time Camera Tracking: When is High Frame-Rate Best?*, pages 222–235. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[14] L. Montesano, J. Minguez, and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3499–3504, Aug 2005.

[15] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh. Self-supervised segmentation of river scenes. In *2011 IEEE International Conference on Robotics and Automation*, pages 6227–6232, May 2011.

[16] Jeffrey Hawke, Corina Gurău, Chi Hay Tong, and Ingmar Posner. *Wrong Today, Right Tomorrow: Experience-Based Classification for Robot Perception*, pages 173–186. Springer International Publishing, Cham, 2016.

[17] W. Churchill, Chi Hay Tong, C. Guru, I. Posner, and P. Newman. Know your limits: Embedding localiser performance models in teach and repeat maps. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4238–4244, May 2015.

[18] J. Dequaire, C. H. Tong, W. Churchill, and I. Posner. Off the beaten track: Predicting localisation performance in visual teach and repeat. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 795–800, May 2016.

[19] N. Snderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *2012 IEEE International Conference on Robotics and Automation*, pages 1254–1261, May 2012.

[20] G. Grisetti, R. Kmmerle, and K. Ni. Robust optimization of factor graphs by using condensed measurements. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 581–588, Oct 2012.

[21] Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Y. Ng, and Sebastian Thrun. Discriminative training of kalman filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[22] J. Dunik, M. Simandl, and O. Straka. Unscented kalman filter: Aspects and adaptive setting of scaling parameter. *IEEE Transactions on Automatic Control*, 57(9):2411–2416, Sept 2012.

[23] Hugo Grimmett, Rudolph Triebel, Rohan Paul, and Ingmar Posner. Introspective classification for robot perception. *The International Journal of Robotics Research*, 35(7):743–762, 2016.

[24] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the ekf-slam algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, Oct 2006.

[25] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh. Predicting failures of vision systems. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3566–3573, June 2014.

[26] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert. Introspective perception: Learning to predict failures in vision systems. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1743–1750, Oct 2016.

[27] Chi Hay Tong Corina Gurau and Ingmar Posner. Fit for purpose? predicting perception performance based on past experience. In *International Symposium on Experimental Robotics*, 2016. *To Appear*. UOXF.

[28] Zoubin Ghahramani and Sam T. Roweis. Learning nonlinear dynamical systems using an em algorithm. In *Proceedings of the 11th International Conference on Neural Information Processing Systems*, NIPS'98, pages 431–437, Cambridge, MA, USA, 1998. MIT Press.

[29] A. H. Mohamed and K. P. Schwarz. Adaptive Kalman Filtering for INS/GPS. *Journal of Geodesy*, 73(4):193–203, 1999.