Efficient Automatic Perception System Parameter Tuning On Site without Expert Supervision

Humphrey Hu
Robotics Institute
Carnegie Mellon University
United States
humhu@cmu.edu

George Kantor
Robotics Institute
Carnegie Mellon University
United States
kantor@cmu.edu

Abstract: Many modern perception systems require human engineers to tune parameters in order to adapt to various environments and applications. This incurs a large startup cost when deploying a robotic system by relying on human expertise and ground truth instrumentation. To alleviate this, we propose a technique using empirical trials to automatically tune a perception system's parameters on-site without expert supervision. Our approach extends upon recent work on introspecting perception performance and uses Bayesian optimization to efficiently search the parameter configuration space. We validate our technique by tuning the laser and visual odometry systems of a physical ground robot in a variety of environments, achieving estimation errors competitive with baseline approaches that use ground truth.

Keywords: Parameter tuning, Perception, Bayesian optimization

1 Introduction

All sorts of robots are finding applications in all sorts of places: Airports, hospitals, nuclear reactors, and even homes are just a few of the environments where robotic systems now operate, freeing their human masters from the tedium of manual labor. The cost and effort that goes into deploying a robot, however, is still considerable. One common source of startup costs is a robot's perception systems, which are usually tuned for a particular environment by a human expert aided by ground truth instrumentation. We argue that this manually-laborious step must be streamlined for it to be truly practical to deploy robots *en masse*, so our goal in this work is to develop a system that enables robots to self-tune their perception systems on-site without expert human supervision.

We address two distinct challenges in on-site tuning: First, we assume that ground truth instrumentation is not available as part of our cost-saving goal. Human experts can bypass this limitation to some degree by using their understanding of the system to evaluate performance, though the heuristic process by which experts reason is challenging to capture. Instead, we propose to use recent work on introspecting current performance with empirical trials. Second, the number of perception parameters that need to be tuned is often quite large: A typical stereo visual odometry algorith can easily boast a dozen tunable parameters. Human experts navigate this high-dimensional configuration space with a combination of in-depth knowledge, heuristics, and intuition to minimize the number of empirical trials that must be run. Here we utilize Bayesian optimization to methodically and efficiently search for high-performing configurations.

Our contribution in this work is the proposal and demonstration of an approach for efficient automatic perception parameter tuning on-site through empirical trials without expert human supervision. We empirically validate our approach by tuning the laser odometry and visual odometry systems on a ground robot in a variety of indoor environments. In these experiments, our approach quickly identifies good odometry configurations, and performs competitively compared to baseline approaches that use ground truth.

2 Prior Work

There exists a vast body of literature that falls under the term "parameter tuning," though the nature of a "parameter" in these works and the applications themselves vary considerably. In our setting, we consider parameters as a relatively low-dimensional specification of behavior for a complex system. We cover here relevant works which are similar in concept or application to our work on perception system tuning.

The idea of tuning perception parameters to match the task at hand can be traced back to active perception and vision [1] [2]. These works, particularly in active vision, focus on geometric factors, such as viewpoint selection, as opposed to data capture and processing parameters. Our work employs the "black box" optimization view used by more modern parameter tuning work, making it more general across different systems.

The bulk of modern parameter tuning work can be found in the setting of control and learning for dynamical systems, specifically for tuning parametric gaits to achieve robust and fast motion for bipeds [3], quadrapeds [4] [5] [6], and snakes [7]. The dynamics of a particular gait depends heavily on the local environment, making it challenging to model and predict performance. As such, most of these works rely on empirical trials using the robot itself to measure performance, and then perform a numerical search, *e.g.*, evolutionary algorithms in [4] and [6], hill climbing in [5], and Bayesian optimization in [7].

Also related are works tuning parameters of classical control algorithms. [8] demonstrate learning an inverted mass balancing controller by optimizing LQR weights, which in turn produce a closed-loop controller. Berkenkamp et al. [9] similarly optimize a quadrotor's flight performance by applying a conservative "safe" Bayesian optimization approach to tune controller gains.

There are a variety of aspects to parameter tuning that have been studied in prior works, which while they are not directly relevant in our setting, are still worthy of mention. Bourki et al. [10] consider tuning a Go-learning algorithm over a discretized configuration space, and show that a uniform search approach can take advantage of parallelization, and in some instances, outperform a model-based search. This is similar to many of the aforementioned works on gait tuning, such as that by Kohl and Stone [5], where multiple robots were used to run tests in parallel. We assume in our setting that we have only one robot, but could extend our approach using the techniques described in [11] to parallelize the Bayesian optimization if there were multiple robots available.

In addition to parallelization, Snoek et al. [11] consider evaluation costs that may arise when tuning hyperparameters of machine learning algorithms, *i.e.*, training a small versus large neural network. Kandasamy et al. [12] consider a related setting where the search algorithm can evaluate at varying fidelities, or as described in their machine learning application, train on varying amounts of data. In this work we use only a single empirical test to evaluate the performance of the perception system, though our approach could be extended in the future to incorporate multiple tests of varying length and thoroughness.

3 Approach

We consider a setting where a perception system is deployed into a new environment and must self-tune to minimize a user-specified perceptual loss as quickly as possible. To avoid requiring ground truth, we use an approximation technique to estimate the expected loss from empirical trials. As executing these trials is time-consuming, we use Bayesian optimization to efficiently search over the space of parameters.

3.1 Formulation

Consider a perception system with a set of M parameters $\theta_i \in \Theta_i, \;\; i = 1, \dots, M$, that must be specified or "tuned". We refer to a specification of all parameters as a *configuration* $c = [\theta_1, \dots, \theta_M]$, and the corresponding configuration space as $\mathcal{C} = \prod_{i=1}^{M} \Theta_i$.

The tuning task is defined by an evaluation function $\rho(\cdot)$ that quantifies the performance of a configuration c according to a user-defined perceptual loss. In practice, evaluation is usually performed by replaying recorded sensor data or executing the physical system to observe the perception system behavior. Evaluation is typically time-consuming, so we desire to tune with as few evaluations as possible.

Were it possible, we would like to find the optimal configuration $c^* = \arg\max_{c \in \mathcal{C}} E\left[\rho(c)\right]$ which maximizes the expected performance. However, the complexity of the interaction between perception systems, their parameters, and system performance means that finding the true global optima is impractical. Instead, our goal in tuning is to find a configuration that performs as closely to this optima as possible. In other words, we seek to minimize the simple regret, defined as the performance gap between the global optima c^* and the final returned configuration c_f as $\rho(c^*) - \rho(c_f)$.

3.2 Introspecting Perception Performance

Recently Hu and Kantor [13] introduced the approximate posterior estimate (APE) for introspecting perception performance without ground truth by using the estimator posterior distribution. For mean-squared-error (MSE) loss and assuming a Gaussian latent distribution, the APE is simply the trace of the posterior covariance, making it an intuitive and efficient alternative to ground truth.

3.2.1 Theory

Let ξ represent a set of data and $f_c(\cdot)$ an estimator function that estimates latent quantity $\hat{x} = f_c(\xi)$ from the data using parameters c. Then, given a loss function $\ell(\cdot, \cdot)$ that operates on pairs of true and estimated latents, the expected loss $L(c)$ over data ξ and states x for parameters c can be expressed as:

$$L(c) = E_{x,\xi}\left[\ell(x, \hat{x})\right] = E_{x,\xi}\left[\ell(x, f_c(\xi))\right] \tag{1}$$

In our tuning setting, ξ would be a sequence of sensor measurements, $f_c(\cdot)$ would be the parametric perception pipeline that processes those measurements to produce estimates, and c would be the perception parameters we are interested in tuning. In practice the expression in Eq. 1 is typically estimated empirically with samples:

$$\hat{L}(c) = \frac{1}{N}\sum_{i=1}^{N} r\left(x_i, f_c(\xi_i)\right) \tag{2}$$

where ξ_i and x_i denote data and latents sampled over independent executions on the robot. The difficulty in using Eq. 2 as a maximization objective for tuning is that it requires knowledge of the true latents x. Obtaining this ground truth is often difficult or impractical outside controlled laboratory environments. Alternatively, observe that Eq. 1 can be decomposed into nested expectations, one of which can be exactly evaluated using the posterior distribution $p(x|\xi)$ over the latent:

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

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

By approximating the outer expectation over data with samples and evaluating the inner expectation using the posterior tracked by a Bayesian estimator, *e.g.* a Kalman filter or a particle filter, we arrive at the approximate posterior estimate (APE) of the expected loss:

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

3.2.2 Practical Considerations

In our state estimation experiments we follow [13] and use an adaptive Kalman filter (AKF) to track the posterior $p(x|\xi)$. The AKF estimates the odometry observation covariance online over a short sliding window, as described in [14], allowing it to capture changes in observation noise arising from environmental as well as configuration changes.

A key assumption of the AKF is that the observation noise is independently drawn. This captures variance-type noise, but not bias-type noise that can arise when exploring the configuration space. As an example, setting too loose of a stopping criteria for a registration algorithm such as ICP results in significant biases; when aligning consecutive point clouds, the algorithm terminates prematurely, resulting in a continuous lagging effect that systematically underestimates the laser displacement. In some instances, biases will be accompanied with increased variance, and so the poor quality of a configuration will still be estimated serendipitiously.

Low variance bias, however, results in an overconfident APE that belies the true poor performance. Thus, finding overconfident configurations is very useful for a perception engineer trying to minimize bias errors in a perception system. Practically speaking, however, we have found that the bulk of bias errors can be eliminated by selecting appropriate search ranges for the parameters. In our experiments, we heuristically set the parameter ranges with a few initial trials by looking for highly overconfident configurations. We found that these search ranges generalized across environments in our experiments.

3.3 Bayesian Optimization

Bayesian Optimization (BO) is a gradient-free sequential optimization technique that selects inputs to query based on past query results. BO and its variants have seen success in applications where querying the objective function is expensive, *i.e.* running an experiment on a robot [9] [7].

In our work we use the Gaussian Process Upper Confidence Bound (GP-UCB) algorithm [15]. Though GP-UCB is known to have good cumulative regret bounds if the true objective function is appropriately representable (has low reproducing kernel Hilbert space norm), recently it has also been shown to have good simple regret bounds for the commonly used squared exponential (SE) and Matérn kernels [16].

The GP-UCB algorithm sequentially queries an objective function, selecting queries that maximize the acquisition function $\phi(\cdot)$ which balances between refinement of configurations with high predicted mean performance, and exploration of configurations with high prediction uncertainty:

$$\phi(c) = \hat{f}(c) + \sqrt{\beta}\sigma(c) \tag{6}$$

Here $\hat{f}(c)$ is the predicted expected loss for configuration c represented by a Gaussian Process (GP), $\sigma(x)$ is the prediction standard deviation at input x, and β is a scheduled exploration weight. Further details on our usage of the algorithm can be found in Sec. 4.3.

4 Experiments and Results

We empirically validate our approach by tuning the odometry systems of a ground robot in a variety of environments. Our goal is to show that we can find configurations which perform similarly to those found by a baseline approach using ground truth.

4.1 Experimental Systems

We use a custom-built indoor ground robot with two onboard odometry systems, shown in Fig. 1 and described below:

Laser Odometry (LO): Two Hokuyo URG-04LX-UG01 planar laser rangefinders mounted on opposite corners of the robot produce scans at 10 Hz each, which are consecutively registered to a keyframe scan with the Point Cloud Library[1] implementation of the Iterative Closest Point (ICP) algorithm.

Visual Odometry (VO): An IDS UI-3140CP USB 3.0 camera mounted inside the robot in a downward-facing configuration with accompanying lighting captures frames at 400 Hz which are registered to a keyframe with the ECC direct registration implementation in OpenCV[2].

These odometry systems produce 2D body velocity observations which are fused by an adaptive Kalman filter (AKF) with angular velocity measurements from an onboard IMU. The AKF sliding window parameters and transition covariance were both tuned heuristically before the experiments to give good filtering performance on hand-tuned configurations for both odometry systems. For a thorough description of the AKF, the reader is referred to [14].

[1] `http://www.pointclouds.org`
[2] `http://www.opencv.org`

Table 1: System parameters and ranges considered for tuning. Units are specified when appropriate.

Parameter	**Range**	**Parameter**	**Range**
Laser Odometry		*Visual Odometry*	
Log voxel filter size (log m)	$[-2, 0]$	Camera gain	$[0, 100]$
ICP max iterations	$[10, 100]$	Camera exposure time (ms)	$[0, 3]$
ICP max corresp. dist. (m)	$[0, 1]$	Image downsample scale	$[0.25, 1.0]$
ICP max solution error	$[0.01, 1.0]$	ECC pyramid depth	$[0, 2]$
ICP log objective epsilon	$[-6, -3]$	ECC max iterations	$[10, 100]$
ICP min inlier ratio	$[0.5, 0.95]$	ECC log objective epsilon	$[-4, -2]$
RANSAC iterations	$[0, 100]$	ECC log min correlation	$[-3, -1.875]$
RANSAC inlier dist. (m)	$[0.1, 1.0]$	Max keyframe movement	$[0.05, 0.25]$

(a) The *lab* environment. (b) The *tunnel* environment. (c) The *office* environment.

Figure 1: The ground robot and three test environments used in our experiments

In our experiments we run and tune each system separately. We heuristically selected 8 significant numerical parameters for each system and normalize them to the ranges listed in Table 1. All odometry software was run on an onboard Intel NUC computer with a Core i5 processor and can be found on our Github[3].

Our validation ground truth system is a four camera Vicon Bonita motion capture setup[4]. The system provides pose data at 100 Hz, which we differentiate to produce ground truth body velocities at 10 Hz. The ground truth velocities are then used with the AKF mean to compute the mean squared error (MSE) ground truth reward signal, and as in [13] the AKF covariance trace is used to compute an APE reward signal. For both MSE and APE we use unity weights to combine linear and angular terms. In our experiments we use the log MSE and log APE as losses, allowing us to better differentiate between ever smaller expected errors.

4.2 Test Environments

We tested in three different environments, as shown in Fig. 1. The **lab** environment has medium-gloss painted concrete floors, desks, and test equipment tripods. The **tunnel** environment has bare concrete floors, and pipes and sandbags along the walls. The **office** environment has glossy linoleum tile floors and multiple chairs and tables.

We mounted our ground truth system cameras along the upper walls and ceiling in the tunnel and office environments so as to not alter the local geometry and affect the laser odometry. In the lab environment, the camera tripods serve as part of the local environment.

4.3 Approaches Tested

We consider four tuning approaches in our experiments:

[3] `https://github.com/Humhu/argus`
[4] `http://www.vicon.com`

Uniform Random on MSE (MSE-UR): As its names suggests, UR samples configurations uniformly randomly and evaluates a configuration based on the ground truth MSE. This approach tests the general "hardness" for the tuning task and tests the benefit of using BO.

Uniform Random on APE (APE-UR): The same uniform random search as UR-MSE, but evaluates configurations based on the APE.

Bayesian Optimization on MSE (MSE-BO): A Bayesian optimization approach using the ground truth MSE as the optimization objective. This approach provides an upper bound on achievable performance.

Bayesian Optimization on APE (APE-BO): A Bayesian optimization approach using the introspected APE as the optimization objective.

Both BO approaches use the GPy[5] implementation of a Gaussian Process model with an Matern kernel ($\nu = 1.5$) with automatic relevance detection (ARD). Each optimization was initialized with 10 uniformly randomly sampled configurations, and model hyperparameters were fitted every 5 samples. We use the exploration rate schedule form given in [15], but as a function of the optimization runtime t as $\beta = \alpha d \log(\gamma t)$, where d is the configuration space dimensionality. In our experiments we used $\alpha = 0.5$ and $\gamma = 0.2$ with t measured in seconds.

4.4 Evaluation Procedure

Our evaluation function $\rho(\cdot)$ executes a short open loop trajectory on the robot and returns the time-averaged loss over the trajectory. More specifically, first the perception system parameters are set to the values to be evaluated. We then wait one second to allow the AKF estimators to converge, after which the robot executes the open-loop evaluation trajectory. Once the trajectory is complete, the time-averaged loss over the trajectory is returned. Finally, to prepare for the next evaluation, the robot uses a side-facing camera to servo to a starting pose relative to a vision fiducial.

In our experiments we used an open loop trajectory consisting of a forward and backward motion, followed by left and right point turns. This trajectory succinctly tests both the linear and angular tracking performance of the odometry systems, taking on average eight seconds per evaluation.

4.5 Data Collected

In our experiments we allocated a budget of 30 minutes of evaluations for each Bayesian optimization trial, or approximately 200 evaluations with a small amount of variance between trials due to servoing and other variations. We ran 5 trials of APE-BO and 5 trials of MSE-BO for each system in each environment, with the APE-BO and MSE-BO trials interleaved to minimize systematic effects. Overall this gives us 30 BO trials amounting to $6,000$ evaluations.

In addition we collected 200 uniformly randomly selected configurations for each environment and system to use for both the MSE-UR and APE-UR baselines, for a total of another $1,200$ evaluations. In total our data consists of $7,200$ evaluations, or 21 hours of robot runtime.

4.6 Metrics

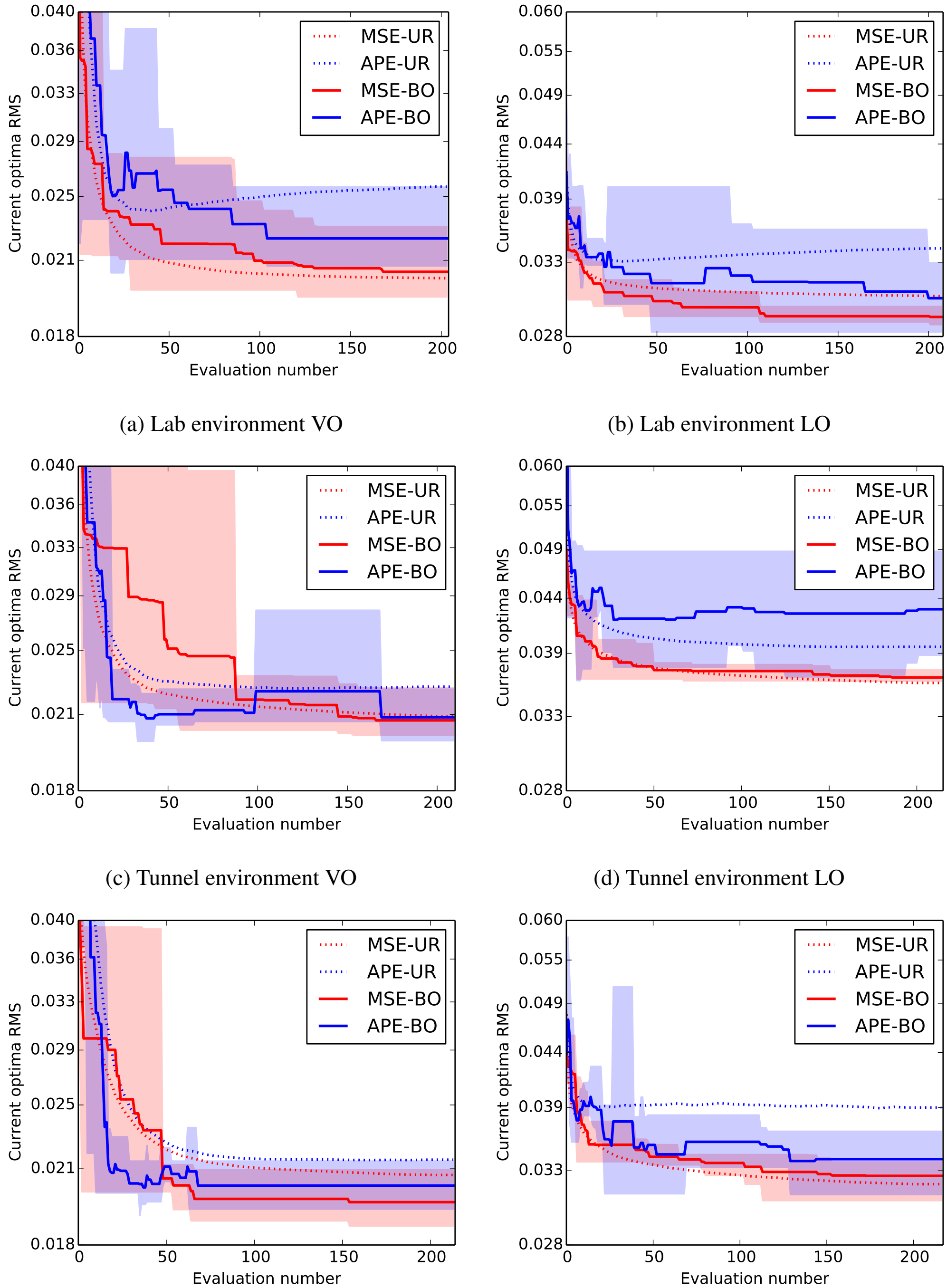
We show the loss of the best configuration found so far according to each approach for each environment in Fig. 2. That is, for MSE-BO we show the best seen MSE for any configuration so far, while for APE-BO we show the MSE of the best seen APE. Instead of running multiple UR trials, we use bootstrapped samplings of the full set of uniform random data to generate both the MSE-UR and APE-UR results.

5 Discussion

5.1 Effective Difficulty of Parameter Tuning

Since we do not know the true optima, we cannot compute the true simple regret in our experiments. For the purposes of self-tuning in a new environment, however, and taking into account the quality

[5] `https://github.com/SheffieldML/GPy`



(a) Lab environment VO
(b) Lab environment LO
(c) Tunnel environment VO
(d) Tunnel environment LO
(e) Office environment VO
(f) Office environment LO

Figure 2: Velocity estimate RMS loss of best-seen configuration for each search strategy versus evaluation number. 200 evaluations is approximately 30 minutes of runtime. Max and min loss across trials are shown for MSE-BO and APE-BO methods as shaded areas.

of our sensors, we believe that the final RMS, on the order of 0.02 to 0.04, achieved in most of the trials is acceptable. This is surprising considering that we only budgeted 200 samples to search the 8-dimensional configuration space.

It is also surprising to note that on Lab-VO, Tunnel-VO, Tunnel-LO, and Office-LO, MSE-UR achieves within 2% of the final RMS compared to MSE-BO, and within 5% and 7% on Lab-LO and Office-VO. This tells us that large volumes of the configuration space provide good performance, possibly due to dependencies between parameters or certain parameters having minimal impact on performance. For instance, the number of ICP RANSAC iterations may have little effect in the tunnel environment, which consists of mostly flat concrete walls. It may be possible to discover and take advantage of this reduced dimensionality with techniques such as [17] or [18].

5.2 Importance of Refinement to APE-based Tuning

Our experiments show two different results with respect to using the APE versus the MSE for tuning. APE-UR performs poorly, returning a configuration with 26% greater RMS than MSE-BO on Lab-VO in the worst case, or 7% greater RMS on Tunnel-LO in the best case.

In contrast, APE-BO performs well in many trials, achieving within 4% final RMS compared to MSE-BO on Lab-LO, Office-VO, Office-LO, and Tunnel-VO. On Lab-VO, APE-BO achieves within 9% RMS, but on Tunnel-LO it performs poorly, achieving within 17% RMS. This suggests that while the APE is only fair at evaluating a wide spread of configurations, it performs better at evaluating good configurations which are actively sought out by Bayesian optimization. An instance of this can be seen in Tunnel-VO, where the APE-BO loss increases at 100 evaluations in one trial due to an evaluation with poor introspection becoming the current optima, but eventually is displaced by a better configuration.

We note also that APE-BO converges significantly faster than MSE-BO on Tunnel-VO and Office-VO. This may be due to MSE being unable to distinguish between configurations which "get lucky" and achieve low loss through a few good observations, and those which produce a more consistent stream of quality observations. This results in the search wasting time searching near these lucky, but otherwise poor configurations. In contrast, the APE captures the high estimate uncertainty produced by these configurations, giving a high estimated loss.

5.3 Failure Modes of APE and Bayesian Optimization

As previously mentioned, APE-BO performs comparatively poorly on Lab-VO and Tunnel-LO, but the causes of these two failures may be different: On Lab-VO, APE-UR also performs poorly, giving 24% greater RMS than MSE-BO. This suggests that the AKF is overconfident in the Lab-VO setting, resulting in low APE losses for configurations with high MSE. Possible causes of this are mentioned in Sec. 3.2.2, and indicate that the filter or parameter ranges may need tuning.

On Tunnel-LO, however, APE-UR achieves 8% greater RMS than MSE-BO, which is considerably less than the 17% of APE-BO in this setting. This suggests that incorrect introspection is not the main cause of failure here, but may instead be a BO failure where the algorithm does not explore aggressively enough. This can be mitigated by tuning the exploration rate schedule parameters or using a larger evaluation budget.

6 Conclusion

We have empirically demonstrated an approach to automatically tune the parameters of a robotic perception system. By using recent work on introspecting perceptual performance, our approach does away with conventional demands on expert human supervision and ground truth instrumentation, allowing automated tuning to be performed on-site instead of only in a lab or factory. This allows highly local specialization instead of needing to rely on "one-size-fits-all" configurations.

More work is needed to propel automated perception tuning into the realm of full practicality. Primarily, the APE relies on the introspective power of the AKF to identify good configurations but, bias-type errors that are not accompanied by variance can result in degraded tuning performance. Detecting these errors or more powerful methods of introspection are needed to address this weakness.

References

[1] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, Aug 1988. ISSN 0018-9219. doi:10.1109/5.5968.

[2] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356. ISSN 1573-1405. doi:10.1007/BF00133571.

[3] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth. An experimental comparison of bayesian optimization for bipedal locomotion. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1951–1958, May 2014. doi:10.1109/ICRA.2014.6907117.

[4] S. Chernova and M. Veloso. An evolutionary approach to gait learning for four-legged robots. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2562–2567 vol.3, Sept 2004. doi:10.1109/IROS.2004.1389794.

[5] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, pages 2619–2624 Vol.3, April 2004. doi:10.1109/ROBOT.2004.1307456.

[6] T. Röfer. *Evolutionary Gait-Optimization Using a Fitness Function Based on Proprioception*, pages 310–322. 2005. doi:10.1007/978-3-540-32256-6_25.

[7] M. Tesch, J. Schneider, and H. Choset. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1069–1074, Sept 2011. doi:10.1109/IROS.2011.6095076.

[8] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe. Automatic lqr tuning based on gaussian process global optimization. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 270–277, May 2016. doi:10.1109/ICRA.2016.7487144.

[9] F. Berkenkamp, A. P. Schoellig, and A. Krause. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 491–496, May 2016. doi:10.1109/ICRA.2016.7487170.

[10] A. Bourki, M. Coulm, P. Rolet, O. Teytaud, and P. Vayssière. Parameter tuning by simple regret algorithms and multiple simultaneous hypothesis testing. In *ICINCO2010*, page 10, funchal madeira, Portugal, 2010. URL `https://hal.inria.fr/inria-00467796`.

[11] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959. 2012. URL `http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf`.

[12] K. Kandasamy, G. Dasarathy, J. B. Oliva, J. Schneider, and B. Poczos. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems 29*, pages 992–1000. 2016. URL `http://papers.nips.cc/paper/6118-gaussian-process-bandit-optimisation-with-multi-fidelity-evaluations.pdf`.

[13] H. Hu and G. Kantor. Introspective evaluation of perception performance for parameter tuning without ground truth. In *Proceedings of Robotics: Science and Systems*, Boston, USA, July 2017. URL `http://www.roboticsconference.org/program/papers/44/`.

[14] A. H. Mohamed and K. P. Schwarz. Adaptive kalman filtering for ins/gps. *Journal of Geodesy*, 73(4):193–203, 1999. ISSN 0949-7714. doi:10.1007/s001900050236.

[15] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 1015–1022, USA, 2010. URL `http://dl.acm.org/citation.cfm?id=3104322.3104451`.

[16] J. Scarlett, I. Bogunovic, and V. Cevher. Lower bounds on regret for noisy Gaussian process bandit optimization. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1723–1742, Amsterdam, Netherlands, 07–10 Jul 2017. URL http://proceedings.mlr.press/v65/scarlett17a.html.

[17] J. Djolonga, A. Krause, and V. Cevher. High-dimensional gaussian process bandits. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1025–1033. 2013. URL http://papers.nips.cc/paper/5152-high-dimensional-gaussian-process-bandits.pdf.

[18] K. Kandasamy, J. Schneider, and B. Póczos. High dimensional bayesian optimisation and bandits via additive models. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 295–304, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045151.