Unsupervised Selection of Optimal Operating Parameters for Visual Place Recognition Algorithms Using Gaussian Mixture Models

James Mount, Ming Xu, Les Dawes, and Michael Milford

Abstract—Visual place recognition (VPR) algorithms are a key part of many autonomous systems, but typically consist of many parameters which require non-trivial optimization for a given deployment environment. Being able to automatically select the optimal operating point for parameters within a VPR algorithm would greatly improve the deployability of autonomous systems in real world scenarios. For example, in an aerial context, platform altitude and camera field of view play a critical role in how much of the environment a downward facing camera can perceive. The sensor coverage and its subsequent processing also has significant computational implications. In this letter, we develop an unsupervised system that can predict the performance of a VPR algorithm, using only a limited number of analogous training images. At the core of our approach is the estimation of a recall proxy using Gaussian mixture models and domain-valid assumptions. We develop a robust, intuitive selection criteria to choose the optimal operating point for a deployment environment to show how our system can facilitate automatic parameter selection. Finally, we show how our system can continuously estimate the performance of a VPR system “on-the-fly”. We evaluate our method’s effectiveness and generality on both aerial and ground-based real-world datasets. We believe these results will assist in the streamlined deployment of visual localization algorithms in real-world situations.

Index Terms—Localization, visual-based navigation.

I. INTRODUCTION

VISUAL place recognition is a critical capability for many robotic and autonomous systems such as self-driving cars, and plays an important role in SLAM and localisation algorithms. The selection of parameters for a VPR system can have a drastic affect on performance [1]. For example, [2] showed that the size of the images used within appearance-based techniques affects performance. Furthermore, parameters selected using a calibration routine can outperform those selected by experts [3], [4]. Selecting suitable parameters for a given application and environment is one of the key challenges in making the transition between research and application [5]–[7]. In this letter we address this challenge within the context of selecting the optimal sensor coverage or image resolution for a specific operational environment and VPR algorithm. We focus on sensor coverage and image resolution because while a naive “more is better” approach using as many pixels as possible can lead to higher recall rates, it also leads to high compute. Hence, if a lower sensor coverage or resolution can be used for effective localisation, significant reductions in compute can be achieved.

Our unsupervised system infers a performance score for a VPR algorithm and its parameters using a limited number of training images analogous to the deployment environment. The core component of our inference system is the parameter estimation of a Gaussian mixture model in combination with domain-valid assumptions and the calculation of a score that is correlated with recall. We also develop a selection criteria for the automatic detection of the optimal operating point that uses a human-interpretable threshold. Finally, we provide a proof-of-concept that demonstrates how our prediction system could be used in an online fashion to constantly infer the performance of a VPR algorithm. These contributions are a significant improvement over our previous work [8], which required supervised training and was only applied to aerial imagery. The improved interpretability also facilitates its use by operators who are not VPR domain experts.

The letter proceeds as follows. In Section II we summarize related works. Sections III and IV describe techniques the experimental evaluation setup. Results are presented in Section V. Finally, Section VI discusses the outcomes of our experiments, the limitations of the systems and potential areas of future research.

II. BACKGROUND

The work presented in [3], [4], [8], [9] is most closely related to the research presented here. In [8], a system used carefully calibrated training sets to select the optimal sensor coverage, but relied on supervised calibration, produced non-interpretable metrics and only applied to surface-based domains. [3], [4], [9] conducted design space exploration on SLAM algorithms to select optimal operating parameters based on the absolute trajectory error and computational load requirements, but only...
on the synthetic ICL-NUIM [10] dataset and with ground truth availability. Lowry et al. and Jacobson et al. created techniques that automatically optimise threshold parameters for place recognition [11]–[13] but operated on the output of the VPR algorithm, rather than modifying its internal parameters.

Active SLAM algorithms [14]–[19] typically attempt to maximise the number of suitable localisation features in order to minimise localisation error. However, this maximisation operates on the input rather than tuning parameters specific to the visual localisation algorithm. Researchers have also explored using the state estimate of the system to reduce the search space in subsequent iterations [20], [21]. Across all this prior work, the primary aim was to maximise recall/accuracy performance without necessarily considering other aspects such as computational efficiency and performance plateaus (beyond which increasing a parameter has no benefit whilst increasing compute).

Research has also focused on calibration routines that identify spatial and temporal transforms between pre-determined sensor configurations [22]–[27]. Investigation of how visual sensors can be employed to overcome kinematic and control model errors is an additional key research field [28]–[30]. However, these research areas have typically had a different focus to the research here, instead examining the relationship between sensors and robotic platforms or between sensors and other non-localisation-based competencies. The automatic selection of hyper-parameters is also related, especially within the field of deep-learning [1], [31]–[34].

III. APPROACH

We contribute two independent processes, Performance Prediction Scoring and Operating Point Selection, collectively referred to here as the calibration procedure. The performance prediction scoring (PPS) algorithm, which is the primary contribution, produces a quality score which is a predictor for final place recognition performance given a specific parameter value. The PPS algorithm is unsupervised and only requires a limited amount of training data analogous to the deployment environment. The operating point selection method uses the output of the PPS system across a variety of calibration parameter values to select the optimal operating point for the deployment environment; we define optimal to be the parameter value which jointly balances place recognition performance and computation time. The performance prediction scoring and operating point selection routines can be found in Algorithms 1 and 2 respectively.

A. Assumptions

We first discuss the assumptions underlying our approach. We assume the following:

- The chosen visual place recognition front-end can compute a comparison score for two images. Formally, we assume the existence of a function \( d : \Omega \times \Omega \rightarrow \mathbb{R} \) which takes two images as input and yields a comparison score between them, where a higher score indicates a higher likelihood two images were captured in the same place.

B. Performance Prediction Scoring Procedure

We introduce the statistical model used to define the performance score and motivate this selection based on observations from benchmark datasets. We conclude by providing implementation details on how to infer this score from a limited training set without any supervision. We use the following notation:
The matrix consisting of pairwise comparison scores between a set of query and reference images outputted by the place recognition front-end is given by $\mathbf{D}$:

- The vector of comparison scores for the $i^{th}$ query image over the entire reference image set is given by $d_i$. This represents a row in matrix $\mathbf{D}$.
- The comparison score between the $i^{th}$ query image and reference image $j$ is given by $d_{ij}$. This represents a single value in the matrix $\mathbf{D}$.

Fig. 1 shows a visual representation of the PPS algorithm.

1) Model Formulation: A key motivation of our model formulation was the observation that for place recognition datasets, where ground truth pose information is available, the distribution over image comparison scores for true positive matches is very distinct from true negative matches. In Figs. 2(b) and 2(d), we show the distribution of positive scores compared to negative scores for two of the datasets in our experiments. The place recognition task can be formulated as an inference problem where, given a set of image comparison scores between a query image and reference image set, we wish to infer whether or not each score is generated by a true-positive or true-negative association. Using this formulation, we assume that our observed image comparison scores are samples from a random variable $D$. In addition, we assume that $D$ depends on a binary valued random variable $Z$, where the value of $Z$ indicates if a score is generated by a positive $P$ or negative $N$ match respectively. This assumption leads to modelling $D$ as a two component Gaussian mixture model (GMM) where $Z$ represents the identity of the mixture component. Formally,

$$
D|Z \sim \mathcal{N}(\mu_Z, \sigma_Z), \quad Z \sim \text{Bernoulli}(p),
$$

(1)

where $Z \in \{0, 1\}$, $\mu_0, \sigma_0, \mu_1, \sigma_1$ are the mixture component parameters for $P$ scores and $N$ scores, respectively and finally $p \in [0, 1]$ is the probability of a score being generated by a positive match. In addition, we denote the full parameter vector by $\theta = \{\mu_0, \sigma_0, \mu_1, \sigma_1, p\}$. Furthermore, the probability density function of this model given an image comparison score $x$ is given by

$$
f(x|\theta) = p \mathcal{N}(x|\mu_0, \sigma_0^2) + (1-p)\mathcal{N}(x|\mu_1, \sigma_1^2)
$$

(2)

The choice of Gaussian mixture components was motivated by observing that the distribution of scores across all of our datasets look approximately Gaussian (see Figs. 2(b) and 2(d)). In addition, parameter estimation for GMM models is well understood and can be easily implemented.

2) Implementation: The expectation-maximisation (EM) algorithm is an iterative process that can be used to find the maximum likelihood estimate of the model parameters $\theta$ given a training dataset. In our case, this dataset is the matrix of comparison scores $\mathbf{D}$ unrolled into a single vector. The EM algorithm additionally requires an initial estimate of $\theta$ (e.g., random initialization, or initializing the mean parameters using K-Means). Unfortunately, the EM algorithm can be quite sensitive to the initial value of $\theta$ [35]. Additionally, depending on the sampling proportions and separation of the components found in the training data, learning GMM parameters may result in clusters that do not represent the $P$ and $N$ categories accurately.

To address this challenge, we have identified a parameter initialisation and data augmentation strategy that enables effective learning of GMM parameters in the context of VPR algorithms.

Our parameter initialisation utilises the VPR assumptions established earlier. We rely on the assumption that higher scores in $\mathbf{D}$ will more likely have been generated by the $P$ components, which is again shown in Figs. 2(b) and 2(d). However, a typical $\mathbf{D}$ from a VPR dataset has a disproportionately large number of scores generated from a negative match compared to positive matches. For example, in Fig. 2 a we show the histogram of the comparison score matrix $\mathbf{D}$, for 200 query images. The distribution representing the $P$ scores cannot be identified in this example because for every true-positive score (i.e. query image) there are over 150 000 true-negative scores. Applying the EM algorithm without suitable modifications to the data and parameter initialization will lead to a poor fit without two distinct clusters as desired.

We initialize our mixture component parameters using $\mathbf{D}$ by hypothesising that scores within $r \in (0, 1)$ of the maximum
value of \(d_i\) are likely to be positive scores, while the remaining scores in \(d_i\) belong to the negative category. Formally, for \(n\) query images, we define a partition of \(D\) given by \(d_P\) and \(d_N\) as

\[
\hat{d}_P = \{d_{ij} \mid d_{ij} \geq \tau \cdot \max(d_i), \forall d_{ij} \in D\}
\]

\[
\hat{d}_N = D \setminus \hat{d}_P.
\]

This initial partitioning works under the assumption that there is only a small number of true-positive reference matches for each query, preferably a singular match. However, if this assumption is violated the performance of the system will slowly degrade rather than immediately catastrophically fail. Gaussians are fitted to these partitions, \(\hat{d}_P\) and \(\hat{d}_N\), to form initial estimates for \(\mu_0, \sigma_0, \mu_1, \sigma_1\) respectively. The initial estimate of \(p\) is given by

\[
p_0 = \frac{n}{|d_P| + |d_N|}.
\]

This parameter initialization strategy tends to produce clusters that are initially separated. However, if the separation of the two components is still minimal this parameter initialisation methodology could still result in poor GMM clusters. We empirically found that when estimating the GMM parameters \(\theta\) across a range of VPR parameter values (e.g., sensor coverage), results are improved if it is performed iteratively. Consequently, here we begin with the parameter value that maximises cluster separation and use the resultant \(\theta\) estimate to initialise the EM algorithm for the next parameter value. This improved results, as the separation of the components tends to increase for larger patch sizes and image resolutions, see Figs. 2(e) and 2(f).

Even with a good initialization strategy, the EM algorithm is sensitive to over/under representation of clusters in the training data. To overcome the disproportion of true-negative to true-positive scores samples, we down-sample the estimated true-negative score samples when \(1 - p_0\) is above a user defined proportion threshold \(p_m \in (0, 1)\). Down-sampling is performed by drawing \(m\) samples from a Gaussian fitted to \(\hat{d}_N\), where \(m\) is given by

\[
m = \frac{np_m}{1 - p_m}.
\]

This approach yields a new vector \(\tilde{d}_N\) which is used to update the initial estimate of \(p_0\) by replacing \(\hat{d}_N\) in (5). We draw samples from a fitted Gaussian rather than directly from \(\hat{d}_N\) as we empirically found it improved the consistency of the procedure’s parameters (e.g., \(p_m, r_i\) etc.) across environments, domains and VPR front-ends. By applying both the parameter initialization and down-sampling step before applying the EM algorithm, we learn robust clusters which more adequately reflect the score samples from the true \(P\) and \(N\) distributions.

3) Inference: After estimating \(\theta\) using the EM algorithm, we use the GMM density (2) to produce a single score that predicts the performance of a VPR algorithm. We found this score is highly correlated to recall, and hence we will refer to it as the Recall Proxy denoted \(r_p\). The recall proxy is calculated by determining the probability of drawing a value from the positive component, that is above the \(n^{th}\) percentile of the negative component. This is given by

\[
r_p = P(D_P > z), \quad z = P(D_N < p_l),
\]

where \(D_P\) and \(D_N\) denote the positive and negative score components \(|D| = 1\) and \(|D| = 0\), respectively and \(P(X > x) = \int_x^{\infty} N(s|\mu, \sigma^2)ds\) is the cumulative distribution function for \(X \sim N(\mu, \sigma^2)\). Finally, \(p_l \in (0, 1)\) is the percentile threshold, a user specified value. We also investigated other inference methods to predict performance; for example, the mean and minimum value of the estimated posterior probabilities of the maximum score for each query belonging to the positive component. We found none of these performed as well as the recall proxy we have presented.

Fig. 3(a) provides an example of the ground truth recall and our recall proxy as the patch size is varied. Notice that the shape of the recall proxy curve behaves similarly to the ground truth recall, particularly the inflexion point. The differences in value, between the two curves, at the smaller patch sizes is because the two components, \(P\) and \(N\), have reduced separation resulting in sub-optimal GMM parameter estimates. However, having a similar shape between the recall proxy and ground truth recall, across the range of values, is the primary aim here rather than having a similar absolute value. This is because we are more concerned with how the performance of the VPR algorithm will be affected by the change in a parameter value, rather than the absolute value. This allows us to select the optimal operating point independent of absolute performance values.

This recall proxy is a major improvement over our previous work [8]. Previously we used the overlap coefficient as a proxy for performance, which had a simple limitation in that it was not overly interpretable by a user. Therefore, by developing a quantity that behaves similarly to recall, we have increased the user-interpretability of the system, further facilitating its use in deployments.

C. Selection of Optimal Operating Point

Our second contribution is a method for selecting the optimal operating point. The selection method uses the rolling standard
deviation of the recall proxy to identify the inflexion point before the plateau region (Fig. 3). We assume that the performance gains of a VPR system decrease monotonically as a parameter is swept across a range of values.

To select the optimal operating point we compute the rolling standard deviation of the recall proxy as the parameter value varies. Formally this is given by

$$\sigma_r(w_r) = \{\sigma_{r,i} \}_{i=1}^N, \quad \sigma_{r,i} = \sigma(\{r_{p,i}-w/2, \ldots, r_{p,i+w/2}\})$$

where \(w\) is the window size, \(r_p\) is a vector of recall proxy scores in increasing patch size order, \(N\) is the number of patch sizes and \(\sigma\) is the sample standard deviation operator. \(\sigma_{r,i}\) denotes the \(i^{th}\) rolling sample standard deviation.

The operating point is identified by the value at which \(\sigma_r(w_r, r_p)\) achieves and stays below a user selected threshold \(r_t\). As the operating point most likely sits between two calibration values we use linear interpolation. If no value within \(\sigma_r(w_r, r_p)\) achieves the desired threshold, the largest calibration value is selected. A visual example of the selection heuristic can be found in Fig. 3(b).

The user selected threshold \(r_t\) is more interpretable than the metric/threshold in our previous work [8]. A user can approximately interpret the threshold \(r_t\) as "I want to select the parameter value that achieves within \(r_t\) of the maximum achievable recall," an improvement in usability over the selection criteria in our previous work.

IV. EXPERIMENTAL SETUP

Experiments were performed on a standard desktop running 64-bit Ubuntu 16.04, or the Queensland University of Technology High Performance Computing system utilizing Python 3. Both local feature-based and two direct image-based VPR algorithms were used to demonstrate that the performance prediction technique is front-end agnostic.

The first technique employed was Normalised Cross Correlation (NCC), which returns a correlation matrix between a template and reference image. The maximum value within the matrix represents the point within the reference image that is most similar to the template. In the surface-based datasets there was a singularly large reference image, which all query templates were correlated against. We used the entire correlation matrix as the comparison score vector, \(d_i\), for a single query image, \(i\). In the forward-facing domain, where the reference set is comprised of multiple images, we used the maximum value of the correlation matrix between each query and each reference as the comparison score. The comparison score vector, \(d_i\), for a single query image, \(i\), was the maximum value of the correlation matrix for that query correlated with each reference image within the dataset.

The second and third techniques utilised the percentage of inlier ORB features and the Mean of Absolute Differences (MAD) and were only applied on the forward-facing datasets. The ORB method involved extracting features and determining the percentage of inlier features between the two images with a value of 1 representing a 100% feature match. The scores from the MAD method were negated (i.e. \(1 - MAD(Q, R)\) where \(Q\) and \(R\) represent query and reference image respectively) so that similar images had higher scores. The comparison score vector \(d_i\) for query image \(i\) was the percent of inlier features or the negated MAD score between query image \(i\) and every reference image within the dataset.

A. Image Datasets

The proposed techniques were evaluated on both surface-based and forward-facing datasets, varying in both operational domain and environment type, which ranged from urban to forested locations (see Fig. 4 and Table I).

1) Surface Datasets: Surface datasets were sourced from aerial photography provider Nearmap. We curated the datasets to contain a diverse range of imagery, including forests, grassland, rural and suburban environments, as well as images at various altitudes and differing levels of appearance variation. Each Nearmap dataset consists of two pairs of pixel aligned images (this alignment is only used to aid evaluation), a reference and query image with a calibration and validation pair. The reference and query image in each pair are from different dates with consequent appearance variation. We randomly generate 200 sub-images from the query image in each pair to use within our experiments. The reference image in each pair remains as a single image and each query sub-image is compared against it.

2) Forward Datasets: We also used two benchmark forward-facing camera datasets to evaluate our calibration procedure. The first comprises the Summer and Fall traverses of the Nordland dataset, a 729 km traversal through Norway. We extract the images at 1FPS and use approximately the first 11 000 frames in our experiments. The second dataset uses the 2015-02-10-11-58-05 and 2015-02-13-09-16-26 routes from the Oxford RobotCar data as the reference and query traverse respectively. We find the nearest-neighbour in \(SE(3)\) space between the traverses.
Lw is the difference between the max-normalised recall

\[ \text{Shape Similarity Vector} \]

\[ r_s \]

\[ R_p X \]

PERIMENTS AND

\[ r \]

\[ r \]

\[ (8) \]

\[ s = \frac{r_p}{\max(r_p)} - \frac{r}{\max(r)} \]

where \( r_p \) and \( r \) are the calibration recall proxy and validation recall vectors respectively.

A. Surface Datasets

The performance of the calibration routine and its predictive properties on the six Nearmap datasets using NCC are shown in Figs. 5(a)–(c). The shape of the calibration recall proxy curves (Fig. 5(a)) reflect the shape of the validation recall curves (Fig. 5(b)) especially near and past the inflexion point as demonstrated by the shape similarity vector curves in Fig. 5(c) decaying to zero. These results demonstrate the success of the system in predicting the performance of a VPR algorithm even with a limited amount of training data. The Nearmap 6 recall proxy does not reflect the validation recall because there is a limited amount of unique data within the calibration pair (i.e., localization is nearly impossible in this dataset), while there is some unique information in the validation pair.

The selected patch size from the sensor coverage selection routine is indicated by the black circle in Fig. 5(a). The patch size selected by the calibration data achieves a validation recall that occurs at, or nearby, the inflexion point (Fig. 5(b)). This inflexion point is the optimal patch size in terms of maximising performance while minimising computation overhead. This demonstrates the improved selection criteria is effective in selecting the optimal sensor coverage.

B. Forward Datasets

The performance of the calibration procedure in optimising the sensor coverage and image size can be seen in Figs. 5(d)–(e) and 5(g)–(i). The recall proxy curves from the calibration data reflect the shape of the validation recall curves, as is demonstrated by Figs. 5(f) and 5(i). This similarity in shape (rather than absolute values) is key here when determining the optimal parameter value. The results show that the performance prediction system is applicable to multiple parameters and domains even with limited training data. The recall proxy curves reasonably predict the VPR performance, but with reduced similarity due to the number of reference images drastically increasing in the validation set, from 100 to 1000+. The 100 image calibration set is less representative of the 1000+ validation set, which for example has increased perceptual aliasing.

The selected patch and image size from the calibration routine is indicated by the black circle in Figs. 5(d) and 5(g) respectively. As can be seen in Fig. 5(h) the selected patch size achieves a validation recall near the inflexion point in 2 of the 3 cases. In Fig. 5(g) the selected image resolution also achieves a validation recall near the inflexion point. The optimal operating points are correctly selected despite the difference in the absolute

to create a dataset with one-to-one frame correspondence with approximately 7000 frames at a spacing of approximately 3 m. Datasets were split into calibration (first 800 frames) and validation sets (remaining frames).

We use only 100 images for each calibration set to evaluate the ability of the system to accurately predict performance using a limited number of training images. This corresponds to using only 50% and 12.5% of the calibration set for the surface and forward datasets, respectively.

B. Parameter Values

Key parameter values were determined empirically from a number of smaller test sets not included here (Table II). Parameters values remain constant within their domain, despite the variation in environments (forest, urban etc.), although the percentile threshold, \( p_t \), parameter does appear to be dependent on the VPR algorithm. Many parameters are constant across the two domains, demonstrating some generality.

The rolling standard deviation window size, \( w \), is set to 2 for the selection of optimal operating point. An equivalent selection method could utilise the difference in consecutive elements in \( r_p \), since \( w \) equals 2. However, we consider the rolling standard deviation approach to have a greater flexibility and be more applicable to future work.

V. EXPERIMENTS AND RESULTS

Four experiments were performed. The first two investigate the predictive capabilities of the PPS algorithm on both surface-based and forward-facing datasets, within the context of selecting the optimal sensor coverage for an analogous deployment environment. The third is a proof-of-concept to demonstrate that the calibration procedure can be extended to alternative parameters, in this case image resolution. The fourth demonstrates how such a system could be used in an online fashion to predict VPR performance and adjust the sensor coverage. The forward-facing datasets were pre-processed using all calibration parameter values: we have rounded the selected optimal parameter value to the nearest pre-processed parameter value in the validation recall curves.

| Parameter | Surface | Forward | Description |
|-----------|---------|---------|-------------|
| \( p_m \) | 0.99 | 0.9 | Maximum True-Neg. Sample Proportion |
| \( p_t \) | 0.9999 | 0.95 (NCC) | Recall Proxy Percentile Threshold |
| \( p_t \) | 0.95 (MAD) | 0.99 (ORB) | |
| \( n \) | 100 | | Number of Calibration Images |
| \( r \) | 0.99 (NCC, ORB) | 0.8 (MAD) | Estimated True-Pos. Max Value Percentage |
| \( w \) | 2 | | Rolling Standard Deviation Window Size |
| \( t_t \) | 0.02 | | Operating Point Recall Threshold |
| \( k \) | 2 pixels | 2 images | True Match Distance Threshold |

To quantify the shape similarity between the calibration recall proxy and the validation recall curves, we utilise a metric referred to here as the Shape Similarity Vector. The shape similarity vector \( s \) is the difference between the max-normalised recall proxy and recall vectors at each calibration point. Formally,
value between the calibration and validation recall curves. This additionally demonstrates that the improved selection criteria using the rolling standard deviation is robust across domains. In the RobotCar NCC patch size optimisation case, the rolling standard deviation of the recall proxy does not stay below the user selected threshold. This is caused by the larger patch sizes reducing the performance of the VPR algorithm, which violates the assumption that performance gains are monotonically decreasing. A more sophisticated selection criteria could be of help here. This result also reveals that specific sub regions of an image may be more informative than using the entire image, as observed in other studies [36].

The results additionally demonstrate that such a system could be used in VPR systems like SeqSLAM [2] to optimise the image resolution for the deployment environment. Additionally, the recall proxy could be used to choose an appropriate sequence length.

C. Online Coverage Calibration

The PPS algorithm can be used online to continuously estimate the behaviour of a VPR without supervision or ground truth access. We provide a proof-of-concept by running the calibration procedure in an iterative manner using the previous 100 frames to estimate the optimal sensor coverage for the current frame and compare this against using the entire image. The first 4000 frames of the Nordland and RobotCar validation sets are used with NCC as the VPR method. Results are not directly comparable to the offline results due to differences in the dataset subsets used.

The Nordland dataset achieved recalls of 58.5% and 60.1% for the online sensor coverage calibration and entire image VPR systems respectively. Equivalent figures for the RobotCar dataset were 64.8% and 72.8%. Although online recall is slightly lowered, the results in Fig. 6 show the system dynamically varies the sensor coverage and at times using a patch size significantly smaller than the full-image.

VI. DISCUSSION AND FUTURE WORK

We developed a procedure that can predict the performance of a VPR algorithm. Our system introduces additional parameters in order to facilitate the unsupervised calibration of the normal system parameters. The key practical advantage of this approach is that these new parameters require far less consideration or management, given they can be mostly held constant across different domains and front-ends, as shown here. However, an exhaustive analysis identifying the robustness of the parameters...
across domains and VPR front-ends would be valuable. The system is significantly improved over past work, with new human-interpretable metrics and the ability to operate online and in a completely unsupervised manner, and should be applicable to other parameters, such as the number of features, as long as the assumptions hold.

Future improvements will address the sensitivity of the EM algorithm to the initial parameter estimates, especially when the mixture proportions are heavily skewed or the component separation is limited. The raw estimate (RE) true-negative distribution closely resembles the ground truth distribution (Figs. 2(e) and 2(f)) - using a Bayesian approach that uses strong and weak priors on the initial true-negative and true-positive estimates could improve the reliability of the system. Generality could be improved by implementing an automated curation system that moderates the availability of query images per spatial location. Future work could also investigate methods for relaxing the assumption that performance gains monotonically diminish as patch size increases, improving applicability in scenarios where using specific sub-regions can outperform using the entire image.

The selection of optimal VPR parameter values typically relies on in-depth knowledge and/or in-environment supervised testing and evaluation. We hope that the automated, unsupervised methods developed here could mitigate these challenges and help facilitate further deployment of autonomous systems across a range of environments.

REFERENCES

[1] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2011, pp. 2546–2554.

[2] M. J. Milford and G. F. Wyeth, “Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights,” in Proc. IEEE Int. Conf. Robot. Autom., 2012, pp. 1643–1649.

[3] B. Bodin et al., “Integrating Algorithmic Parameters into Benchmarking and Design Space Exploration in 3D Scene Understanding,” in Proc. Parallel Architectures Compilation Techn. - Conf. Proc., no. 16 Institute of Electrical and Electronics Engineers Inc., 2016, pp. 57–69.

[4] B. Bodin, L. Nardi, H. Wagstaff, P. H. Kelly, and M. O’Boyle, “Algorithmic Performance-Accuracy Trade-off in 3D Vision Applications,” in Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw., 2018, pp. 123–124.

[5] A. Jacobson, F. Zeng, D. Smith, N. Boswell, T. Peynot, and M. Milford, “Semi-supervised slam: Leveraging low-cost sensors on underground autonomous vehicles for position tracking,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2018, pp. 3970–3977.

[6] F. Zeng, A. Jacobson, D. Smith, N. Boswell, T. Peynot, and M. Milford, “i2-s2: Intra-image-seqslam for more accurate vision-based localisation in underground mines,” in Proc. Australas. Conf. Robot. Autom., 2018, pp. 1–10.

[7] F. Zeng, A. Jacobson, D. Smith, N. Boswell, T. Peynot, and M. Milford, “Enhancing underground visual place recognition with Shannon entropy saliency,” in Proc. Australas. Conf. Robot. Autom., 2017, pp. 223–232.

[8] J. Mount, L. Dawes, and M. Milford, “Automatic Coverage Selection for Surface-Based Visual Localization,” IEEE Robot. and Autom. Lett., vol. 4, no. 4, pp. 3900–3907, Oct. 2019.

[9] S. Saeedi, L. Nardi, E. Johns, B. Bodin, P. H. Kelly, and A. J. Davison, “Application-oriented design space exploration for SLAM algorithms,” in Proc. - IEEE Int. Conf. Robot. Autom., 2017, pp. 5716–5723.

[10] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM,” in Proc. - IEEE Int. Conf. Robot. Autom., 2014, pp. 1524–1531.

[11] G. H. Lee and M. Pollefeys, “Unsupervised learning of threshold for geometric verification in visual-based loop-closure,” in Proc. - IEEE Int. Conf. Robot. Autom., 2014, pp. 1510–1516.

[12] S. Lowry and M. J. Milford, “Building beliefs: Unsupervised generation of observation likelihoods for probabilistic localization in changing environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2015, pp. 3071–3078.

[13] A. Jacobson, Z. Chen, and M. Milford, “Autonomous multisensor calibration and closed-loop fusion for slam,” J. Field Robot., vol. 32, no. 1, pp. 85–122, 2015.

[14] A. J. Davison and D. W. Murray, “Simultaneous localization and map-building using active vision,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 865–880, Jul. 2002.

[15] S. Frintrop and P. Jensfelt, “Attentional landmarks and active gaze control for visual SLAM,” IEEE Trans. Robot., vol. 24, no. 5, pp. 1045–1065, Oct. 2008.

[16] A. Kim and R. M. Eustice, “Perception-driven navigation: Active visual SLAM for robotic area coverage,” in Proc.- IEEE Int. Conf. Robot. Autom., 2013, pp. 3196–3203.

[17] S. M. Chaves, A. Kim, and R. M. Eustice, “ Opportunistic sampling-based planning for active visual SLAM,” in Proc. IEEE Int. Conf. Intell. Robots Syst., 2014, pp. 3073–3080.

[18] A. Kim and R. M. Eustice, “Active visual SLAM for robotic area coverage: Theory and experiment,” Int. J. Robot. Res., vol. 34, no. 4/5, pp. 457–475, Apr. 2015.

[19] S. M. Chaves, A. Kim, E. Galceran, and R. M. Eustice, “Opportunistic sampling-based active visual SLAM for underwater inspection,” Auton. Robots, vol. 40, no. 7, pp. 1245–1265, Oct. 2016.

[20] N. Nourani-Vatani and P. K. Borges, “Correlation-based visual odometry for ground vehicles,” J. Field Robot., vol. 28, no. 5, pp. 742–768, 2011.

[21] M. O. Agel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, “Adaptivesearch template matching technique based on vehicle acceleration for monocular visual odometry system,” IEEJ Trans. Elect. Electron. Eng., vol. 11, no. 6, pp. 739–752, 2016.

[22] W. Maddern, A.arrison, and P. Newman, “Lost in translation (and rotation): Rapid extrinsic calibration for 2d and 3d lidars,” in Proc. IEEE Int. Conf. Robot. Autom., 2012, pp. 3096–3102.

[23] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2013, pp. 1280–1286.

[24] J. Kelly and G. S. Sukhatme, “Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration,” Int. J. Robot. Res., vol. 30, no. 1, pp. 56–79, 2011.

[25] D. Scaramuzza, A. Harati, and R. Siegwart, “Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2007, pp. 4164–4169.

[26] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, “Automatic extrinsic calibration of vision and lidar by maximizing mutual information,” J. Field Robot., vol. 32, no. 5, pp. 696–722, 2015.

[27] S. Weiss, M. W. Achtelik, S. Lynen, M. Chi, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments,” in Proc. IEEE Int. Conf. Robot. Autom., 2012, pp. 957–964.

[28] Y. Meng and H. Zhuang, “Autonomous robot calibration using vision technology,” Robot. Comput.-Integr. Manuf., vol. 23, no. 4, pp. 436–446, 2007.

[29] M. Švaco, B. Škeroranja, F. Šulígou, and J. Beriči, “Calibration of an industrial robot using a stereo vision system,” Procedia Eng., vol. 69, pp. 459–463, 2014.

[30] G. Du and P. Zhang, “Online robot calibration based on vision measurement,” Robot. Comput.-Integr. Manuf., vol. 29, no. 6, pp. 484–492, 2013.

[31] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” J. Mach. Learn. Res., vol. 13, no. Feb, pp. 281–305, 2012.

[32] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Auto-weka: Combined selection and hyperparameter optimization of classification algorithms,” in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2013, pp. 847–855.

[33] R. Bardenet, M. Brendel, B. Kégl, and M. Sebag, “Collaborative hyperparameter tuning,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 199–207.

[34] C. Gold, A. Holub, and P. Sollich, “Bayesian approach to feature selection and parameter tuning for support vector machine classifiers,” Neural Netw., vol. 18, no. 5/6, pp. 693–701, 2005.

[35] E. Shireman, D. Steinley, and M. J. Brusco, “Examining the effect of initialization strategies on the performance of Gaussian mixture modeling,” Behav. Res. Methods, vol. 49, no. 1, pp. 282–293, Feb. 2017.

[36] M. Milford, E. Vig, W. Scheirer, and D. Cox, “Vision-based simultaneous localization and mapping in changing outdoor environments,” J. Field Robot., vol. 31, no. 5, pp. 780–802, 2014.