Bayesian Optimization for Monocular VSLAM under Varied Conditions

Charles Horn
Institute for Aerospace Studies
University of Toronto
Toronto, Canada
charlie.horn@mail.utoronto.ca

***Abstract*— Monocular visual simultaneous localization and mapping (VSLAM) provides a fundamental environment in which to develop best practices for feature detection and matching with direct impact to a relevant robotics application. By identifying guidelines and improvement methods in this simplified framework, we can then transfer these methods to situations involving more sensors and data in different modes. In this paper we perform an analysis of a standard open source monocular VSLAM framework [3]. As a benchmark, we compare localization results across two datasets, KITTI [1] and Tartanair [2]. We then hypothesize about ways to improve results under various conditions. Machine learning is used to optimize this process on a training set, and the results of optimization are evaluated on a test set. The optimization shows a 11.2% improvement over the baseline.**

I. INTRODUCTION

Most SLAM challenges involve multiple sensors, and often muti-modal data [6] [5] [4]. This becomes more prevalent as the cost of sensors, and data processing, decreases. However, analysis of monocular VSLAM can provide best practices to be applied even when more sensors are added. The monocular setting can isolate and identify sources of inaccuracy that might not be directly apparent when there are many components in the system. This simplicity is exactly what we aim to take advantage of in this paper. By reducing the degree of input parameter space, parameters that have the most impact on performance are isolated. On top of that, these parameters are effectively optimized under varied conditions using probabilistic methods.

Another important component of the optimization scheme is the availability of synthetic data. The Tartanair dataset [2] provides simulated environments, with multiple tracks for each. This paper focuses on the tracks that are in the same environment, under different conditions. This provides a reliable comparison when optimizing for these conditions because the underlying scene is equivalent.

The combination of a simple VSLAM framework, and simulated environment data on which to compare localization results allows us to determine optimal sets of feature detection parameters for various environmental conditions.

[1]Charles Horn is a MEng student at University of Toronto Institute for Aerospace Studies, Toronto, Canada. charlie.horn@mail.utoronto.ca

II. RELATED WORK

A. Monocular VSLAM

The first implementation of bundle adjustment (BA) in real time was in the study of visual odometry in [7]. This paper was able to reduce computational complexity by performing fast and local BA instead of global BA.

Building upon this idea of local BA, PTAM [8] created a model that would perform local BA, and global BA if frames are not converged. This BA scheme helped PTAM perform effective keyframe selection. Finally, any features detected in the selected keyframe are added to the global point cloud.

The monocular SLAM developed by ORB-SLAM [9] was the first open-source SLAM framework that used loop closing, relocalization, and map reuse. ORB-SLAM performs tracking, mapping, and loop closure by maintaining a global set of ORB features. Exploiting this global set of features, ORB-SLAM is able to effectively register keyframes by implementing a covisibility graph that allows the algorithm to only compare with keypoints that should be visible from the current estimated state.

While ORB-SLAM was open source, the codebase was not modular or easily extensible. Researchers had a difficult time adapting the program to their specific needs [3]. Open VSLAM [3] created a visual SLAM framework that was modular, extensible, and came with clear documentation. Following the methodology from ORB-SLAM (as well as other SLAM programs), Open VSLAM uses indirect SLAM, which performs local feature mapping on detected keypoints. This approach is more robust to changes in lighting, but often performs worse in textureless settings [3].

Figure 1 shows each of the modules of the Open VSLAM framework. In this paper, the configurations of the tracking module are optimized, and the results of which are then passed into the mapping and global optimization modules. The end results show that optimization in the early stages of this algorithm improves results of the localization, especially under varied environmental conditions.

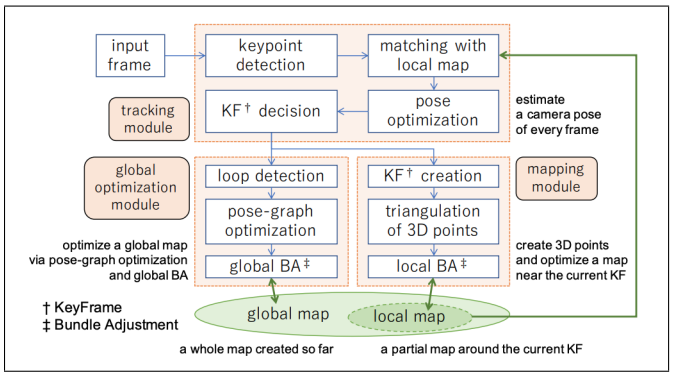

Fig. 1. Relational Diagram of Open VSLAM Modules

B. Feature Detection in Extreme conditions

The impacts to classical feature detection algorithms in extreme lighting conditions has been well documented and compared [15] [16] [17].

One approaches to this problem is to change the sensor type, in order to see beyond the visible spectrum of light [16]. This method can be effective, however the algorithm requires tuning to account for the new data feed, and the lower resolution. This also requires access to a different sensor (eg. infrared camera), or a dataset collected by one.

Pribyl et al. suggest that standard imaging techniques are not as well suited for extreme lighting due to the lower contrast of the image [15]. Instead, they claim that High Dynamic Range (HDR) techniques are required. This technique is logical because digital images can become saturated toward either end of the pixel intensity scale.

Shyam et al. take the approach of image pre-processing and threshold reduction under low-light conditions [17]. The pre-processing is used to enhance image contrast and/or brightness, while the lowering of the threshold will allow for more feature detections. These methods are evaluated across multiple feature detection and matching algorithms on static images.

Due to the availability of RGB images, taken with standard imaging techniques, we do not follow the methods of [15] and [16] in this paper. However, we do explore similar approaches to [17]. Additionally, Machine Learning is used to optimize parameter values. We also consider optimization across multiple parameters, under different environmental conditions, instead of just low-lighting. Finally, we use results from real VSLAM frameworks as a metric, rather than static images.

C. Hyperparameter Optimization

Tuning and optimization is an essential step in any system whose behaviour is dependent on the value of its parameters. In fully observable systems, one can often take a derivative, and then determine a local or global minimum or maximum. However, many systems worth studying do not have the luxury of being fully observable, and are as such referred to as black-box systems. The performance of these black-box systems can still be heavily reliant on the values of its parameters, and therefore they will need to be optimized in some way.

In systems where the dimensions of the parameter space is relatively low, a simple grid search is often sufficient. This process is used by [10], in order to determine optimal parameters for FAST feature detection on static images. The problem with grid search becomes apparent to anyone who tries to implement it in a situation where the parameter space is high dimensional, or iterations of the optimized function can take a long time.

To remove the manual component of parameter tuning, many papers propose automating this process and employing a machine learning approach [11]. The approach applied in this paper is Bayesian Optimization, which is often used to tune hyperparameters of learning models, but can easily be applied to any black-box system.

A requirement for classical Bayesian Optimization (BO) is that the parameters take values from a continuous distribution. For our purposes in this paper, parameters take on discrete values. In order to account for this, [12] proposes a method that optimizes the exploration/exploitation parameters of the underlying acquisition function with the constraint that the output must be an integer. This is an elegant solution that showed good results, but it starts to optimize the optimizer of the parameters of the function, which is quite removed from the initial objective. Alternatively, some papers simply assume a continuous value, and round the result of the acquisition function up or down to the nearest integer. This approach is called Naive Bayesian Optimization. This has been shown to have poor performance in some cases because it will often repeat parameter values and not gain any new information [12]. In this paper we propose to treat input features as continuous variables, and train the optimizer with the result as if parameters were never rounded. Similar to Naive BO, the objective function will not tell the optimizer that the parameter ever lived in a discrete domain, however results will be reported to the BO process as if it were executed with the given floating point value.

III. Problem Definition

Monocular VSLAM relies on feature detection and matching. This project attempts improve the localization results of the Open VSLAM algorithm under varied conditions by optimizing feature detection and matching parameters. By leveraging Open VSLAM's [3] modular design, and Tartanair's [2] diverse dataset, we present a method for optimizing this process under 4 separate conditions: day time, night time, summer, and winter. Since results will always depend on the specifics of your environment, the individual parameter settings are not the primary results of

this paper. Instead, this paper provides suggestions that can be used as guidelines or starting points when operating under any of the previously mentioned conditions.

IV. Data

A. KITTI

The KITTI dataset is a well established benchmark for many robotics tasks [1]. In this project, the KITTI training set is used to establish baseline results. Open VSLAM [3] also reports baseline results on the KITTI dataset, but only for the stereo case. The KITTI data used is summarized in Table I.

TABLE I
KITTI DATA DETAILS

Attribute	Value
Sequences	22
Average Frames per sequence	2109
Average 'Speed' (metres per frame)	1.04

B. Tartanair

The Tartanair dataset was created in 2020 to provide a challenging set of sequences that include diverse weather and lighting conditions, high variety of motion patterns, and dynamic objects [2]. Using the Unreal Engine [13], they provide 30 photo-realistic environments, in which multiple sequences are gathered. Leveraging the simulated environment, Tartanair is able to provide accurate ground truth data for multimodal data sources, including stereo, depth, segmentation, camera pose, IMU, occupancy map, optical flow, and lidar. Sequences are also categorized into easy and hard classes, based on the aggressiveness and speed of the camera rotations and translations. For this project, the 'easy' sequences from the 16 sample environments are used. The localization results on these tracks were compared to the KITTI baseline to determine that the Tartanair environment posed a greater challenge than the KITTI data (Table III). For the optimization process, we focus on environments that contained opposing conditions ('abandonedfactory' vs 'abandonedfactory_night' and 'seasonsforest' vs 'seasonsforest_winter'). The details of this data can be seen in Table II.

TABLE II
TARTANAIR DATA DETAILS

Attribute	Value
Sequences	43
Average Frames per sequence	1045
Average 'Speed' (metres per frame)	0.15

In order to validate the optimization results, training and test sets were created for each condition. Each training set consisted of 4 individual sequences taken from the same environment under the same condition. The corresponding test set contained two different sequences, which results in a train/test split of $\frac{2}{3}/\frac{1}{3}$.

C. Baseline Results

Table III shows the results of running Open VSLAM's monocular VSLAM with default parameters on 1) all KITTI training sequences and 2) all Tartanair sequences in the train and test set. This table was the first step in motivating this project, because as we can see there is a drop in performance in the Tartanair environment.

TABLE III
BASELINE RPE RESULTS KITTI VS TARTANAIR

Dataset	FAST threshold	Num. Levels	Average RPE (m)
KITTI	20	8	0.061
Tartanair	20	8	**0.98**

V. Methodology

A. Overview

B. Solving Scale for Monocular VSLAM

Monocular VSLAM comes with an inherent issue: it is impossible to recover scale directly from the image feed. There are many ways to rectify this, most of which include additional data sources (IMU, radar, odometry, etc.) in the absence of ground truth data. However, for this project we are not focusing on a practical application of recovering scaled localization results, so we make use of the provided ground truth data to scale and align results. The steps to perform alignment and velocity scaling are seen below, and are provided by [2].

First, we transform translation vectors p_i and quaternion representation q_i to a transformation matrix T_i. Then alignment is done by applying the first transformation matrix to all time steps:

$$T_i = T_1^{-1} T_i \tag{1}$$

Then we convert from transformation matrices back to the translation-quaternion representation. To scale the aligned poses, we calculate the 'velocity' at each time step, measured in $\frac{m}{frame}$. This assumes a relatively consistent frame rate, but the impacts of this assumption are mitigated when we average in the final step.

$$v_{gt_i} = p_{gt_i} - p_{gt_{i-1}},\ v_{est_i} = p_{est_i} - p_{est_{i-1}} \tag{2}$$

$$s_{gt_i} = \sqrt{\sum_{j=1}^{3} v_{gt_i,j}^2},\ s_{est_i} = \sqrt{\sum_{j=1}^{3} v_{est_i,j}^2} \tag{3}$$

$$scale = \frac{1}{N} \sum_{1}^{N} \frac{s_{est_i}}{s_{gt_i}} \tag{4}$$

The results in Figure 2 show that this method is effective, while not requiring us to iteratively minimize a cost function.

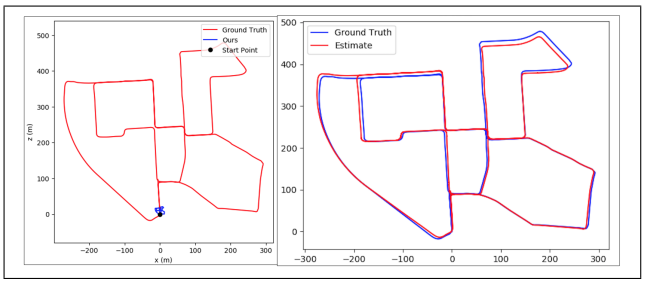

Fig. 2. Before and after Alignment and Velocity Scaling

C. Assumptions

An underlying assumption of Bayesian Optimization (BO) is that the objective function behaves as a Gaussian Process. Therefore we assume that the RPE score of a SLAM sequence can be treated as a random variable sampled from a standard normal distribution. This property allows the joint distribution for all inputs in the domain X to be Gaussian, and therefore the objective function can be treated as a RV with a Gaussian distribution over the space of functions with a continuous domain.

We also assume no prior about performance as a function of these parameters. This is an assumption that motivated this choice of optimization scheme, because BO assumes a weak or non-existent prior.

In order to narrow our scope, we assume that the parameters with the most impact on performance will be the pixel difference threshold h, and the number of scaling levels l. These will become the subject of our optimization.

Finally, we assume independence between these two parameters as they relate to the performance of the SLAM algorithm. This assumption was made to reduce computational cost. This is a reasonable assumption given that h is a parameter of feature *detection* while l is a parameter of feature *matching*.

D. Parameter Optimization

Now that we have an open-source VSLAM framework, a dynamic dataset, a method to correct for scale, and an evaluation metric, we can begin optimization. In order to tune the parameters of the feature detection and matching process, we use Bayesian Optimization.

1) Acquisition Function: BO is a machine learning algorithm that provides the next best parameter value to test in order to maximize the objective function. This next best value is provided by the *acquisition function*. In our case, the objective function is the monocular VSLAM process, and the maximized result is $1/RPE$. We use Expected Improvement (EI) as our acquisition function. EI returns the value x^* in the range of parameter values that maximizes Equation 5.

$$x^* = argmax_x \ [max(0, f(x) - f^+)|D] \qquad (5)$$

Where f^+ is the current best performance, and D is the data we have observed already. With the assumption of a Gaussian distribution over the space of functions, equation 5 becomes:

$$x^* = argmax_x(\mu(x) - f^+)\phi(Z) + \sigma(x)\phi(Z) \qquad (6)$$

Where $Z = \frac{\mu(x) - f^+}{\sigma(x)}$ and $\phi(Z)$ is the standard Normal PDF. This function is now broken into two terms, one of which relies on the mean $\mu(x)$, and the other relies on the covariance $\sigma(x)$. Weighting these two terms can bias the algorithm's preference for exploration or exploitation.

EI is the most commonly used acquisition function because it applies to most scenarios, it performs well, and is simple to implement [14].

E. Hypotheses

1) FAST Feature detection pixel difference threshold: Our first hypothesis is that, under challenging lighting conditions, FAST feature detection will need to lower the pixel difference threshold value when detecting corners. As discussed in [15], this is due to the fact that, as the scene gets darker, pixels intensities will approach 0 at a decreasing rate. This will eventually cause lower contrast in the surrounding area of relevant features, which could lead to them not being detected.

When considering seasons, a winter landscape will have some areas of high contrast covered with snow. We expect that snowy scenes will cause low texture regions, which could also lead to fewer feature detections.

Fewer detected features will mean fewer matched features, which will lead to less robust localization and higher error. Therefore, a reduced threshold h will be more effective. This hypothesis is studied in [17], however it is restricted to the matching results between two stereo images. In this case we are applying the theory to a monocular VSLAM framework that is matching features across frames in order to perform localization.

2) ORB Feature matching scaling levels: The second hypothesis relates to the more complex movements of the robot, and the impact that this has on the matching of features. We expect that fast and dynamic movements of the Tartanair dataset [2] will cause issues with scaling feature descriptions for the ORB matching process. If feature matches are less resistant to scale, they could be unmatched between frames which would lead to poor localization. Because of this, we expect that under all conditions (day, night, summer, and winter), Open VSLAM will perform better with more scaling levels.

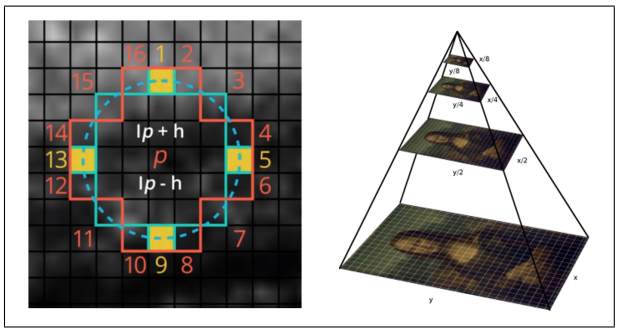

Fig. 3. Parameters to optimize. Left: FAST threshold h, Right: ORB scaling levels l

VI. Results

A. Hypothesis #1

The results of optimization of the threshold value h show that night conditions prefer a lower value than day conditions. An example of the BO iterations on the night vs day sequences can be seen in Figure 4. Therefore, the first hypothesis is upheld on the training set under day/night conditions.

When comparing summer vs winter conditions, both scenarios are maximized for the same threshold value. This is not what was expected, and this implies that feature detection in textureless regions/scenes is not affected by the threshold value in the same way as low-lighting conditions. Perhaps a slight snow covering actually provides high contrast in some areas where the edges of the snow stops.

The resulting optimal values, and the resulting RPE can be seen in Table V.

B. Hypothesis #2

When optimizing the scaling levels parameter l, we see some unexpected results. While it was hypothesized that all conditions would perform better with a higher number of scaling levels, it appears that the opposite is true. While some deviation in the data could be explained by noise in the system, it is significant that all four conditions, each trained on 4 separate sequences, showed a similar trend. An example of this trend for summer vs winter conditions can be seen in Figure 5. We can see that results are fairly consistent in the range of [4-13], with a sharp drop-off as we increase past this range.

Initially we expected that more scaling levels would improve performance at the price of computation cost. However, the results indicate that increased scaling levels introduces confusion in the matching process. Provided enough scaling levels, perhaps the algorithm could find a significant amount of faulty matches that were able to meet the requirements of the outlier rejection process. This could have caused features in one frame to be attributed to a scaled version of another feature in the previous frame.

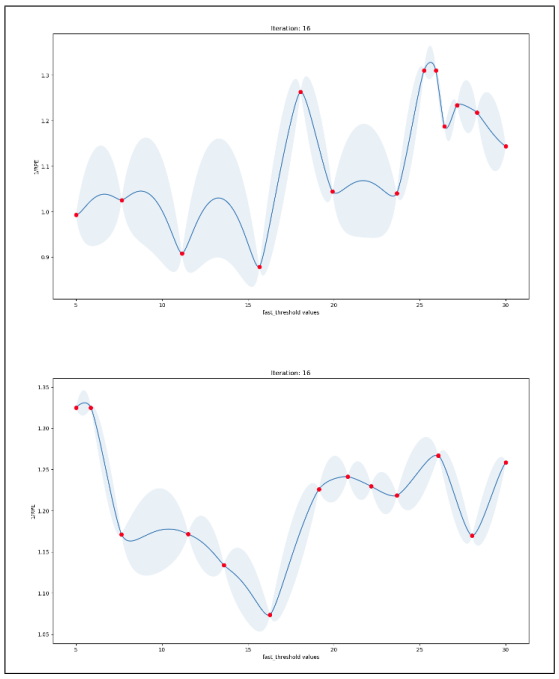

Fig. 4. Optimization Results for FAST Threshold values. Top: Abandoned Factory Day, Bottom: Abandoned Factory Night

For this reason, the second hypothesis was incorrect, and there was not a general agreement on the number of scaling levels across all conditions.

C. Overall Results

Running Bayesian Optimization for the parameters h and l on the training data was able to efficiently produce optimal parameter values for each condition. These optimal values, and their resulting RPE on the training set, can be seen in Table IV.

TABLE IV
Optimized RPE results on Tartanair training data

Condition	FAST threshold	Num. Levels	Average RPE (m)
Day	25	11	0.99
Night	5	12	0.76
Summer	11	13	0.44
Winter	11	9	0.42

In order to test the results of the optimization process, we run the SLAM process with both the optimal parameter values and the default values (provided by [3]) on the *test* sequences. We then compare the resulting average RPE and record the results in Table V.

The optimized parameters showed improvement in the day and night conditions. This aligns with what the training process suggested under these conditions, and supports

TABLE V

RESULTS OF DEFAULT AND OPTIMIZED PARAMETERS ON TARTANAIR TESTING DATA

Conditions	Parameter Set	Threshold	Num. Levels	Average RPE (m)
Day	Default	20	8	1.57
Day	Optimized	25	11	**1.49**
Night	Default	20	8	1.0
Night	Optimized	5	12	**0.71**
Summer	Default	20	8	0.54
Summer	Optimized	11	13	**0.54**
Winter	Default	20	8	0.38
Winter	Optimized	25	9	**0.36**

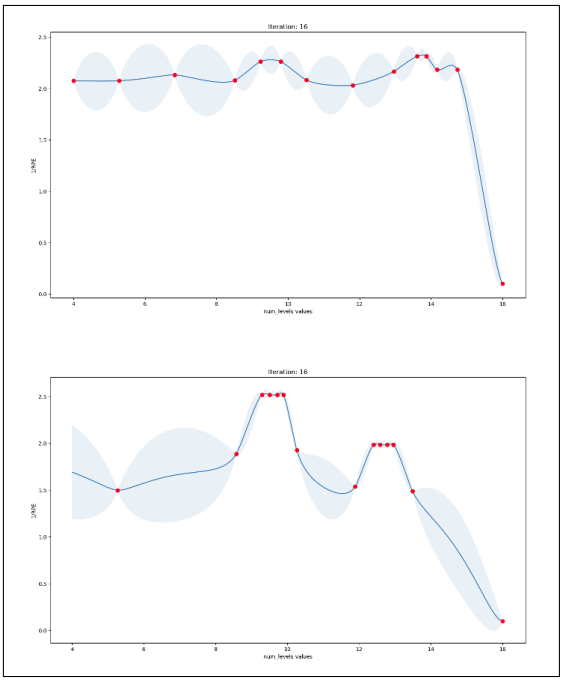

Fig. 5. Optimization Results for Scaling Levels values. Top: Seasons Forest Summer, Bottom: Seasons Forest Winter

our first hypothesis. Under reduced lighting, decreasing the feature detection pixel intensity difference threshold h provided better localization results for Monocular VSLAM.

However, we can see from the results that the optimized parameters did not always improve results on the test set. Under summer and winter conditions, the results were very similar. This indicates that the chosen parameters, h and l, did not influence localization results under seasonal variations of the Tartanair dataset.

VII. CONCLUSION

In order to determine the most impactful parameters when comparing VSLAM results across seasons, we recommend extending the parameter space to R^n. By extending the parameter space, the optimization process will be able to explore multi-dimensional combinations of values, and find the optimal set of values for each season. Comparing these optimal sets will then indicate the parameters that require attention when deploying VSLAM system in different seasons.

The method described in [17] only compares feature matching results on static stereo images, and then suggests a direct correlation to SLAM results. In the method described in this paper, we apply feature detection and matching optimization directly to a SLAM process, and evaluate the results. These results are valuable because they confirm the suggestions of [17]. However, it became clear from the data that using a VSLAM process as the objective function for Bayesian Optimization posed some challenges. Mainly, the system contains significant sources of noise. These noisy results are difficult to model because they depend on many factors of the dataset. Instead, we suggest that this optimization process be applied to a more controlled environment as in [17], or be run on more sequences in order to acheive more consistent results in both training and testing.

This paper presents the effects of changing certain parameters of monocular visual SLAM on the localization results. We make use of a modular and extensible SLAM framework [3] in order to change parameters and collect results. By applying a probabilistic optimization method, we provide guidelines around parameter value choices. Finally, by exploiting a diverse and complex simulated dataset [2], we are able to specify guidelines for various weather and lighting conditions.

REFERENCES

[1] Andreas Geiger, Philip Lenz, Christoph Stiller and Raquel Urtasun, Vision meets Robotics: The KITTI Dataset, in International Journal of Robotics Research (IJRR), 2013.

[2] Wang, Wenshan and Zhu, Delong and Wang, Xiangwei and Hu, Yaoyu and Qiu, Yuheng and Wang, Chen and Hu, Yafei and Kapoor, Ashish and Scherer, Sebastian, TartanAir: A Dataset to Push the Limits of Visual SLAM. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.

[3] Shinya Sumikura, Mikiya Shibuya, Ken Sakurada, OpenVSLAM: A Versatile Visual SLAM Framework, 2020.

[4] Lifelong Robotic Vision Competition, https://lifelong-robotic-vision.github.io/competition/

[5] TartanAir Visual SLAM - Stereo Track, https://www.aicrowd.com/challenges/tartanair-visual-slam-stereo-track

[6] The KITTI Vision Benchmark Suite, http://www.cvlibs.net/datasets/kitti/eval$_o dometry.php$

[7] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Real Time Localization and 3D Reconstruction, LASMEA UMR 6602, Universite Blaise Pascal/CNRS, 63177 Aubiere Cedex, France

[8] Georg Klein and David Murray, Parallel Tracking and Mapping for Small AR Workspaces.

[9] J. Wang, ORB-SLAM: a Versatile and Accurate Monocular SLAM System, Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos, IEEE TRANSACTIONS ON ROBOTICS.

[10] Manyi Wu, Research on optimization of image fast feature point matching algorithm, EURASIP Journal on Image and Video Processing volume, 2018.

[11] P. Koch and B. Wujek and Oleg Golovidov and S. Gardner, Automated Hyperparameter Tuning for Effective Machine Learning, Koch2017AutomatedHT, 2017.

[12] Phuc Luong(B), Sunil Gupta, Dang Nguyen, Santu Rana, and Svetha Venkatesh, Bayesian Optimization with Discrete Variables, Advances in Artificial Intelligence, 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019.

[13] Unreal Engine, https://www.unrealengine.com/

[14] Peter I. Frazier, A Tutorial on Bayesian Optimization, arXiv, July 8, 2018.

[15] Bronislav Pribyl, Alan Chalmers, Pavel Zemcik, Feature Point Detection under Extreme Lighting Conditions, Proceedings of the 28th Spring Conference on Computer Graphics, May 2012.

[16] Tarek Mouats, Nabil Aouf, David Nam1, Stephen Vidas, Performance Evaluation of Feature Detectors and Descriptors Beyond the Visible, Journal of Intelligent & Robotic Systems, June 2017.

[17] Tarek Mouats, Nabil Aouf, David Nam1, Stephen Vidas, Retaining Image Feature Matching Performance Under Low Light Conditions, International Conference on Control, Automation and Systems, October 2020.