Characterizing SLAM Benchmarks and Methods for the Robust Perception Age

Wenkai Ye[1], Yipu Zhao[1], and Patricio A. Vela[1]

***Abstract*— The diversity of SLAM benchmarks affords extensive testing of SLAM algorithms to understand their performance, individually or in relative terms. The *ad-hoc* creation of these benchmarks does not necessarily illuminate the particular weak points of a SLAM algorithm when performance is evaluated. In this paper, we propose to use a decision tree to identify challenging benchmark properties for state-of-the-art SLAM algorithms and important components within the SLAM pipeline regarding their ability to handle these challenges. Establishing what factors of a particular sequence lead to track failure or degradation relative to these characteristics is important if we are to arrive at a strong understanding for the core computational needs of a robust SLAM algorithm. Likewise, we argue that it is important to profile the computational performance of the individual SLAM components for use when benchmarking. In particular, we advocate the use of time-dilation during ROS bag playback, or what we refer to as *slo-mo* playback. Using *slo-mo* to benchmark SLAM instantiations can provide clues to how SLAM implementations should be improved at the computational component level. Three prevalent VO/SLAM algorithms and two low-latency algorithms of our own are tested on selected typical sequences, which are generated from benchmark characterization, to further demonstrate the benefits achieved from computationally efficient components.**

I. Introduction

Simultaneous localization and mapping (SLAM) is a core computational component supporting several application scenarios in augmented/virtual reality and autonomous robotics. As such, benchmarks for assessing the performance of SLAM reflect the diverse deployment profiles of potential application scenarios. They reflect a wide variety of sensors [1]–[3], platforms [4], [5], motion patterns [6]–[8], scene properties [9], [10], and other meaningful characteristics. Given that a large portion of benchmarks are empirical, specialized (to use case) scenarios recorded for evaluation through replay, there can be a lack of control over important configuration variables related to performance. Furthermore, the diversity of software interfaces for the different datasets and algorithms complicates comprehensive evaluation. SLAMBench [11] addresses this last issue through the use of a common API for evaluating algorithms with an emphasis on analysis of different computational platforms and run-time parameters. SLAMBench performance metrics include energy consumption, accuracy, and computational performance. Follow-up work, SLAMBench 2 [12], improves API consistency and includes several SLAM implementations modified for compatibility with the API. The input stream can be synthetic data generated from simulations [9], [13]. Simulation addresses the earlier point through the creation of controlled scenarios that can be systematically perturbed or modified. We hope to advance the practice of benchmarking by providing a meta-analysis or design of experiments inspired analysis of the SLAM benchmarks and algorithms with respect to accuracy and computation. The analysis will characterize existing benchmarks and identify critical components of the SLAM pipeline under different benchmark characteristics.

Our contributions in this direction follows in the subsequent sections. Section II lists existing benchmarks and briefly describes their characteristics according to properties known to impact SLAM accuracy. Analysis of differentiating factors regarding difficulty level is performed to arrive at dominant factors influencing the difficulty annotation. Section III reviews time profiling outcomes of SLAM instantiations in order to determine the time allocation required for (sufficiently) complete execution of the SLAM pipeline prior to receipt of the next frame. Providing sufficient time for the computations enables separating the latency factor from the algorithm factor for establishing the limiting bound of accuracy performance fo SLAM instances relative to existing benchmarks. Section IV applies three prevalent visual SLAM algorithms and two low-latency counterparts to a balanced benchmark set as determined by the analysis of Section II. The aim of the study is to confirm that the qualitative assessment matches the quantitative outcomes with the latter annotations determined by the accuracy results. The outcome distribution will be clustered into four performance classes: *fail*, *low*, *medium*, and *high*, based on clustering the accuracy outcomes into three equal density regions plus adding a fail category. Comparison of the resulting decision trees will establish whether the primary factors impacting performance relative to the distinct performance categories are consistent or if a different prioritization is in order. The described analysis should provide a means to establish where structural weaknesses of published SLAM methods lie and where future research effort should be dedicated to maximize impact. The emphasis will be on monocular SLAM as improvements to monocular systems should translate to the same for stereo and visual-inertial implementation [4], [14]. We anticipate that the findings will support more systematic study of SLAM algorithms in this new era of SLAM research, dubbed the "Robust Perception Age" [15].

[1]Wenkai Ye, Yipu Zhao, and Patricio A. Vela are with School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. {wye35,yzhao347,pvela}@gatech.edu.

This work was supported in part by the China Scholarship Council (CSC Student No: 201606260089) and the National Science Foundation (Award #1816138).

TABLE I
CHARACTERIZATION OF SELECTED SEQUENCE PROPERTIES

Sequence	**Platform**	**Scene** (x_1)	**Duration** (x_2)	**Motion Dyn.** (x_3)	**Environ. Dyn.** (x_4)	**Revisit Freq.** (x_5)	**Difficulty** (y)
Seq 04 [2]	Car	Outdoor	Short	Low	High	Low	Easy
lr kt0 [9]	Synthesized	Indoor	Short	Low	Low	Low	Easy
f2 desk person [16]	HandHeld	Indoor	Short	Low	Medium	Low	Easy
Conf. hall2	AR Headset	Indoor	Medium	Low	Medium	High	Medium
Seq 02 [2]	Car	Outdoor	Medium	High	Medium	Low	Medium
room3 [7]	HandHeld	Indoor	Short	High	Low	Low	Medium
of kt3 [9]	Synthesized	Indoor	Short	Low	Low	Low	Medium
MH 05 diff [8]	MAV	Indoor	Short	Medium	Low	High	Difficult
V1 03 diff [8]	MAV	Indoor	Short	High	Low	High	Difficult
Corridor	AR Headset	Indoor	Medium	Medium	Medium	High	Difficult
NewCollege [1]	Round Robot	Outdoor	Long	Medium	Medium	High	Difficult
outdoor4 [7]	HandHeld	Outdoor	Long	Medium	Medium	Low	Difficult

II. BENCHMARK PROPERTIES

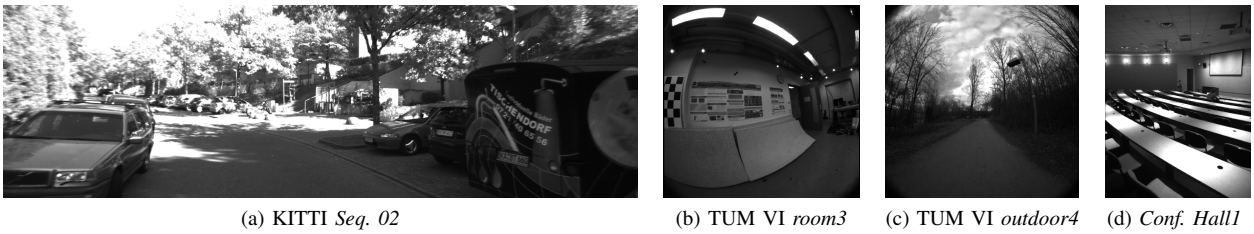
Benchmarking for SLAM varies based on evaluation choices made by different research teams. Some prioritize a select set of benchmark datasets based on anticipated deployment characteristics [5]. Others seek to understand and confirm the general performance properties of a set of methods [17], or to explore the solution landscape associated to parametric variations of a single strategy [11]. Our interest is in understanding the general performance landscape and what subset of available datasets could be used to evaluate general deployment scenarios. If such a subset were to exist, computed averages of the quantitative outcomes could provide a common metric with which to score and compare the impact of algorithmic choices in SLAM implementations.

We performed a literature search for benchmark datasets associated to SLAM algorithms, and any other visual sequence data with ground truth pose information permitting quantitative evaluation of camera pose versus time. Published benchmarks for which the data is no longer available were excluded, such as Rawseeds [18]. In the end, the following corpus of benchmark datasets was identified: NewCollege [1], Alderley [19], Karlsruhe [20], Ford Campus [21], Malaga 2009 [22], CMU-VL [23], TUM RGBD [16], KITTI [2], Malaga Urban [24], ICL NUIM [9], UMich NCLT [25], EuRoC [8], Nordland [26], TUM Mono [27], PennCOSYVIO [28], Zurich Urban MAV [29], RobotCar [10], TUM VI [7], BlackBird [30], and a Hololens benchmark of our own. Altogether, they reflect over 310 sequences with available ground truth signals.

For the analysis, we chose five factors to serve as the parameters of interest for characterizing and differentiating the sequences. They were scene, duration, environment dynamics, motion dynamics and revisit frequency. Some factors were merged into these categories. For example scene illumination and image exposure variations were connected to the *scene* attribute in order to have a more manageable review workload. Categorization for each properties is based on heuristic thresholds or qualitative assessment. For example, the *Duration* property is categorized as *Short*, *Long*, or *Medium* if its duration is below 2 mins, over 10 mins, or between the two thresholds, respectively. The *Environ. Dynamics* is categorized as *Low*, *Medium* or *High* if there are no or rarely moving objects in the scene, a few moving objects, or numerous or frequently seen object movements, respectively. Beyond those five properties, we also assigned difficulty labels to the source benchmarks. When available in the source publication, we kept the labeled assigned by the researchers. Otherwise, assignment was determined using reported tracking outcomes or, if needed, through visual review of the sequence. Table I provides sample descriptions of select sequences, with individual frames from them shown in Fig 1. The complete version can be accessed online [1]. After reviewing the full table, we performed a downselection of the benchmark sequences in order to balance the different categories. Removal was determined by overlapping or common configurations or by choosing a characteristic subset from a benchmark with many sequences. The final set consisted of 117 sequences obtained from various platforms (car, train, MAV, ground robot, handheld, head-mounted), scenarios (46% indoor, 47% outdoor, 7% synthesized), duration (37% short, 33% medium and 30% long), and motion patterns (33% smooth, 38% medium, 29% aggressive).

As a first pass at understanding what factors most impact the difficulty annotation, we applied a decision tree classifier to the annotated set of chosen benchmarks. Each sequence is an observation with the five properties as the predictors and *Difficulty* as the response. Cross-validation is adopted in the process to examine the predictive accuracy of the fitted models and meanwhile to protest against overfitting. Specifically, the training pool is partitioned into five disjoint subsets, and the training process is performed on each subset to fit a model, which is trained using four other subsets and

[1] https://github.com/ivalab/Benchmarking_SLAM

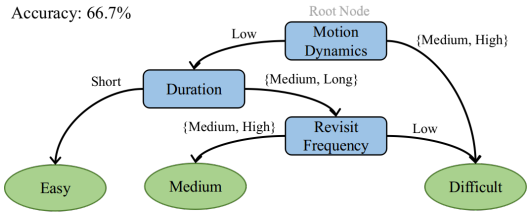


(a) KITTI *Seq. 02* (b) TUM VI *room3* (c) TUM VI *outdoor4* (d) *Conf. Hall1*

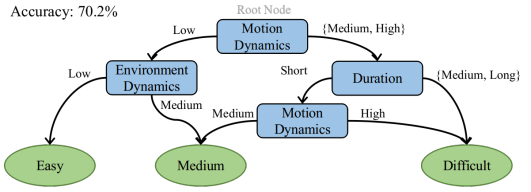
(e) KITTI *Seq. 04* (f) EuRoC *MH 05 difficult* (g) EuRoC *V1 03 difficult*

(h) NewCollege (i) TUM RGBD *desk person* (j) *Corridor* (k) ICL NUIM *lr kt0* (l) ICL NUIM *of kt3*

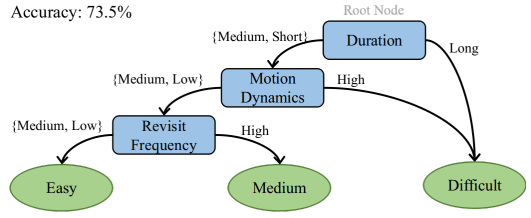
Fig. 1. Characteristic images of selected typical sequences.



(a) Indoor + Outdoor



(b) Indoor Only



(c) Outdoor Only

Fig. 2. Trained Decision Tree Factors influencing difficulty level.

validated using its own subset. The best model is adopted and re-assessed using the whole training pool to report an accuracy. Performing the training procedure for the entire benchmark set, the indoor-only subset and the outdoor-only subset leads to three decision trees, all depicted in Fig 2 with prediction accuracy noted above and to the left of each tree.

Common factors for all of the trees are *Motion Dynamics* and *Duration*, with *Motion Dynamics* being fairly consistent regarding the final outcome. *Duration* is also consistent across the trees, however *medium* durations evaluate differently between indoor and outdoor datasets. For indoor sequences *Environment Dynamics* plays a role in differentiating *easy* versus *medium*, whereas for outdoor sequences it does not. It may reflect the different sensor hardware associated to the two use cases (wide vs narrow field-of-view) and the relative size of the moving objects within the image stream. Interestingly the *Revisit Frequency* has an opposing outcome for the full dataset versus the outdoor dataset, suggesting the opposite role of this factor for the indoor dataset though it is not a dominant one. Revisiting for outdoor scenes may reflect the nature of loop closure at intersections. There are four ways to cross an intersection but only one crossing direction can trigger or contribute to loop-closure. For indoor scenes with more freedom of movement, there may actually be less diversity in view direction during revists.

Based on the decision trees, challenging sequences should be those with high motion dynamics or long duration (irrespective of the motion dynamics). To generate a reference set of sequences spanning these different decision variables and reflecting distinct pathways, we reviewed the dominant factors and identified 12 characteristic sequences across the three performance categories. The *easy* sequences are *Seq 04*, *lr kt0*, *f2 desk person*; the *medium* sequences are *Conf. Hall1*, *Seq 02*, *room3*, *of kt3*; and the *difficult* sequences are *MH 05 diff*, *V1 03 diff*, *Corridor*, *NewCollege*, *outdoors4*.

III. Time Profiling and Time Dilation

Time profiling of the computational modules of a SLAM system provides clues to how SLAM implementations should be improved at the computational component level. This is particularly true for feature-based methods, which typically are more costly than direct methods. To understand the time consumption of the modules in a SLAM pipeline, we advocate fine-grained time profiling and the use of time-dilation when evaluating SLAM systems with ROS bag playback, i.e. *slo-mo* playback. The idea is similar to the *process-every-frame* mode in SLAMBench [11], which continues with the next frame after the previous frame is completely processed, The proposed *slo-mo* is straightforward and easy to apply in ROS. We conjecture that *slo-mo* playback will establish performance upper-bounds for evaluated SLAM systems, which serves as a hint on the potential of SLAM system (e.g. running on better hardware in the near future). The time scaling factor for *slo-mo* playback was chosen to be 0.2, providing 5x more time for a single-frame update.

Quantitative eqvaluation on the chosen 12 sequences involved three state-of-the-art VO/SLAM algorithms, i.e. SVO [31], DSO [32] and ORB-SLAM (ORB) [33]. For those using features, the feature quantity parameter was set to use 800 features per frame in *slo-mo*. The testbed is a laptop with a Intel Core i7-6820HQ quadcore 2.70GHz CPU and 32 GB memory. The loop-closure thread in ORB-SLAM was disabled to operate like a visual odometry (VO) system, though the local mapping thread was not disabled (it behaves like a short-term loop closure). Each sequence was tested once for each SLAM algorithm. Evaluation varied based on the available ground truth. For sequences with high-precision 6DoF ground truth (e.g. from Motion Capture system), tracking accuracy is evaluated with RMSE (m) versus the absolute pose references. For sequences with less frequent ground truth signals or with synthesized ground truth (e.g., using SfM), the RMSE of relative pose error (m/s) is used. The time cost of the major computational components of the three VO algorithms was recorded.

The timing outcomes for the tested algorithms are shown in Fig 3, where the estimated time cost for each component is computed by averaging over all tracked frames in all selected sequences. The methods with direct pose estimation components, SVO and DSO, did not consume significantly more time. Interestingly, DSO ran faster in normal speed, which could due to the improved inverse-depth estimation provided by back-end. With these minor changes, weak performance points in these algorithms should be attributed to algorithm performance limits. ORB-SLAM consumed more time in *slo-mo* versus normal time for many of the early components, but less time for the pose optimization step. Faster convergence of the pose optimization implies better conditioning of the optimization, better predicted poses, or improved feature selection or coordinate estimation. For total time cost, ORB-SLAM take the most time processing each frame, primarily due to the feature extraction and matching.

Pose tracking performance results for *slo-mo* are listed

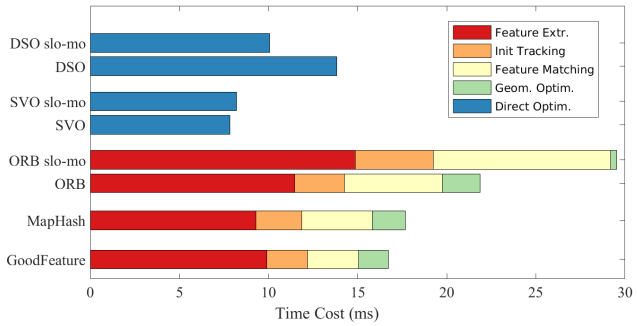


Fig. 3. Time profiling of modules in three state-of-the-art VO algorithms and two low-latency algorithms, running under normal speed and *slo-mo*.

on the left side of Table II. On each sequence, the method with the lowest RMSE/RPE is underlined and the failure cases with tracking loss over one third of the entire seuqence are discarded (marked as dash). Considering first track loss only, DSO is the only algorithm to successfully track all sequences. Furthermore, it has good tracking accuracy (second to ORB-SLAM). This strong performance suggests that improvements to DSO will most likely involve additional components or modifications outside of the core DSO components. In terms of available tracking accuracy, ORB-SLAM achieves the best performance with average RMSE of 0.16m and average RPE of 0.12m/s. However its timing does not match that of DSO, thus modifications should prioritize enhancing ORB-SLAM's timing properties. Though SVO has excellent timing, it has the lowest performance with regards to track loss and pose tracking accuracy (average RMSE of 1.5m and average RPE of 2.65m/s). These outcomes indicate that, the *slo-mo* can help understand performance properties of SLAM systems.

IV. Dataset Properties Influencing Performance

To explore the performance limits in actual operational conditions, the *slo-mo* results can be compared with the ones generated at normal speed. Performance differences may point to potential source of improvement by establishing modifications that nullify them. We run these three algorithm at normal speed five times on each sequence. We also applied two additional ORB-SLAM modifications that aim to lower the compute time of the front-end computations [34], [35] (time improvements can be seen in Fig. 3). The results are summarized on the right side of Table II. To communicate tracking accuracy and the number of tracking failures, we compute the average tracking error only over successful cases, but mark the error in different gray levels according to failure quantity.

Two of the three algorithms experience performance degradation to different degrees when operating with time limit. One, SVO, did not significantly change. Though one additional sequence (*outdoor4*)was tracked for one out of five runs, it did experience more failure than success. Thus, we consider the change in track success rate to be negligible. The tracking accuracy was within 2% of the *slo-mo*

TABLE II

RMSE (M) / RPE (M/S) OF 3 VO ALGORITHMS IN SLO-MO/NORMAL SPEED ON SELECTED SEQUENCES

		one run in *slo-mo*			five runs in normal speed (#failures are highlighted by -/4/3/2/1/0)				
	Seq.	SVO [31]	DSO [32]	ORB [33]	SVO [31]	DSO [32]	ORB [33]	GF-ORB [34]	MH-ORB [35]
RMSE	*f2 desk person*	1.26e0	1.25e-1	4.76e-2	1.53e0	5.36e-1	5.94e-2	3.51e-2	2.88e-2
	lr kt0	4.97e-1	2.21e-1	2.10e-1	3.07e-1	2.61e-1	-	-	-
	of kt3	6.43e-1	3.82e-2	5.58e-2	5.60e-1	3.87e-2	2.57e-1	2.76e-1	6.69e-2
	room3	2.03e0	2.86-1	2.09e-1	2.02e0	-	1.80e-1	-	-
	outdoors4	-	2.21e-2	8.39e-2	1.74e0	-	-	-	-
	MH 05 diff	4.26e0	1.38e-1	3.03e-1	1.44e0	1.08e-1	1.18e0	1.43e-1	2.29e-1
	V1 01 diff	5.53e-1	1.08e0	2.32e-1	6.09e-1	1.34e0	1.25e0	9.23e-1	4.61e-1
	Average	1.54	0.50	0.16	1.17	0.46	0.59	0.34	0.20
RPE	*KITTI Seq 04*	2.06e0	7.88e-2	9.18e-2	1.82e0	8.09e-2	9.74e-2	1.00e-1	9.92e-2
	KITTI Seq 02	6.91e0	1.31e-1	1.38e-1	7.17e0	1.52e-1	2.08e-1	-	1.41e-1
	conf. hall1	4.35e-1	4.33e-1	-	4.25e-1	4.57e-1	1.50e-1	1.64e-1	2.17e-1
	corridor	1.20e0	6.50e-1	2.34e-1	1.38e0	5.31e-1	1.54e0	6.22e-1	1.05e0
	NewCollege	-	1.92e-2	1.65e-2	-	1.93e-2	1.88e-2	1.95e-2	1.92e-2
	Average	2.65	0.26	0.12	2.70	0.25	0.40	0.23	0.31

version. DSO exhibits track loss for some sequences (*room3*, *outdoors4*, *MH 05 diff*) relative to *slo-mo* which might also point to degradation of the back-end processing due to the time constraints. Further analysis would be necessary to understand the source of these differences. ORB degrades the most when returning back to normal speed, both in terms of increased track failure and higher pose error. The higher time cost of ORB-SLAM impacts performance, as ORB-SLAM has to skip frames to complete the process initiated from an earlier one but not yet completed. To examine the impact of lower latency two addtional low-latency algorithms, GoodFeature [14] and MapHash [35] are evaluated for comparison. Both were implemented in the ORB-SLAM framework and achieve low tracking latency through different strategies. The former employs active matching and the second employs more efficient local map subset selection. By lowering the pose tracking latency, the two algorithms improve somewhat tracking success. A bigger improvement is seen for the pose accuracy.

While it is important to understand how well given visual SLAM algorithm work in terms of relative standings, a better understanding or characterization of performance would illuminate where additional effort should be spent improving a particular SLAM algorithm. Here, we replicate the exploration of benchmark properties of Section II but use the quantitative outcomes from the selected sequences. In particular, we re-annotate the *Difficulty* label based on the track loss rate and the tracking accuracy. Each run with each algorithm on each sequence is taken as an observation, thus there will be 300 two-dimensional observations in total for training. To prevent biasing, we saturate RPE at 2 m/s and normalize the values. These two factors yield four candidate categories. An algorithm performance is considered as *high* if it can track poses with low loss and low RPE. If it tracks the entire sequence but with poor accuracy, we consider performance to be *medium*. Moreover, we mark the performance as *difficult* if it fails to track sometime in the middle

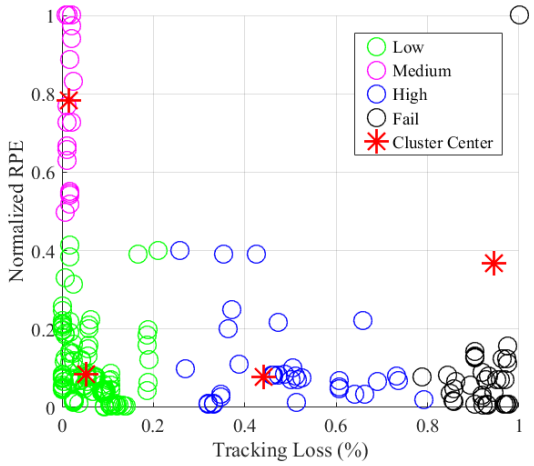


Fig. 4. Kmeans++ clustering on pose tracking errors.

of the sequence, and mark them as *fail* if it is lost in the begining, no matter how accurately it tracks. K-means++ [36] is applied to cluster observations into these four categories. The distribution of observations and the clustered centroids are shown in Fig 4.

Given the cluster results, we categorize the pose tracking performance and build a decision tree for the three main SLAM algorithms tested. Since the tracking results are generated only from three VO algorithms, the property *Revisit Frequency* is removed as a factor. The tree strucuture, from top to bottom, can indicate the significance of sequence properties to each algorithm. According to Fig 5, these algorithms act differently in terms of the selected characteristic sequences. For SVO, *Scene* is the most important factor for decent operation, and *Motion Dynamics* and *Duration* come in the second place. No *Difficult* leaf node exists, meaning that SVO usually tracked all the way if it is successfully started. The *Medium* leaf node is connected with two *Dura-*

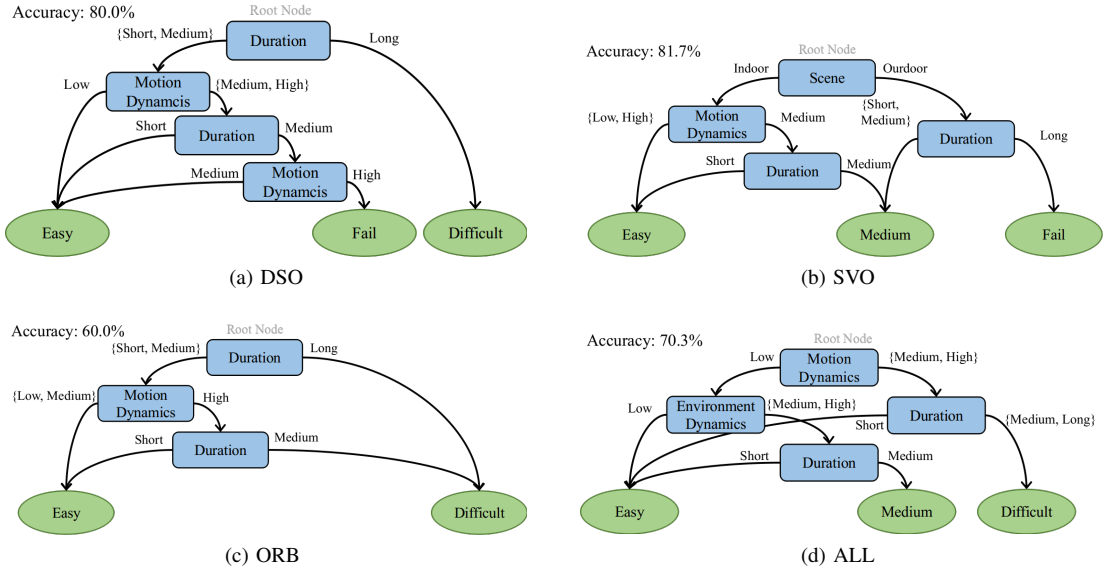


Fig. 5. Trained decision trees using pose tracking errors.

tion middle nodes, indicating its tracking accuracy depends to a great extent on the sequence length. An interesting insight for DSO and ORB is their similarity. Their upper structures are basically the same, from *Duration*, *Motion Dynamics* to *Duration*, which might partially explain the reason they can obtain competitive tracking accuracy on selected sequences. The difference lies in the last judgement for *Motion Dynamics*, where DSO can handle *Medium* dynamics but will fail when it is *High*. In contrast, ORB is not affected by *Motion Dynamics* at this point, and no related failure will be caused. In summary, all algorithms as sensitive to *Duration*, which is related to map maintenance, environment changes, and drift correction if a loop closure module is available.

In addition to the algorithm-specific decision trees, we aggregated all of the data to generate a decision tree. Similar pre-process, clustering and training steps were conducted with the entire set, but with clustering into three categories. The generated decision tree is displayed on the lower right of Fig 5. This tree is a quantitative vesion of the tree from Section II, based on actual outcomes as opposed to subjectively determined labels. By comparison we can find that, both trees take the *Motion Dynamics* as the first important factor, then comes *Duration* and other factors. Our knowledge about what SLAM can do and what scenarios it can complete is consistent with the quantative truth. A more comprehensive understanding can be obtained if breaking the sequence properties into finer scale, but this comparion presents at least two promising fields that SLAM research can focus on in the near future: 1) the robustness under aggressive motion patterns and 2) the ability to handle long-term operation. The former is usually handled by visual-inertial SLAM methods, to which the same analysis can be applied. The latter will require developing a quantitative analysis methodology to better establish how to improve long-term operation.

V. Conclusion

This paper characterizes state-of-the-art SLAM benchmarks and methods, with special attention on challenging benchmark properties and crucial components within the SLAM pipeline. A decision tree is proposed to identify these properties and components. By comparing the performance efficiency of SLAM systems on both normal speed and slo-mo playback, we are able to identify how SLAM implementations should be improved at the computational component level, and suggest where future research effort should be dedicated to maximize impact.

References

[1] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, May 2009.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[3] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580.

[4] Y. Zhao, J. S. Smith, and P. A. Vela, "Robust, low-latency, feature-based visual-inertial SLAM for improved closed-loop navigation," *submitted to IEEE Conference on Decision and Control*, 2019.

[5] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," *IEEE International Conference on Robotics and Automation*, vol. 10, p. 20, 2018.

[6] S. Urban and B. Jutzi, "LaFiDaa laserscanner multi-fisheye camera dataset," *Journal of Imaging*, vol. 3, no. 1, p. 5, 2017.

[7] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *IEEE/RJS International Conference on Intelligent Robot Systems*, 2018.

[8] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[9] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014.

[10] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[11] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. Kelly, A. J. Davison, M. Luján, M. F. O'Boyle, G. Riley *et al.*, "Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5783–5790.

[12] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Luján, S. Furber, A. J. Davison *et al.*, "SLAMBench2: Multi-objective head-to-head benchmarking for visual SLAM," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1–8.

[13] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," in *British Machine Vision Conference*, 2018.

[14] Y. Zhao and P. A. Vela, "Good feature matching: Towards accurate, robust VO/VSLAM with low latency," *submitted to IEEE Transactions on Robotics*, 2019.

[15] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[16] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RJS International Conference on Intelligent Robot Systems*, 2012.

[17] A. Q. Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. OKane *et al.*, "Experimental comparison of open source vision-based state estimation algorithms," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 775–786.

[18] G. Fontana, M. Matteucci, and D. G. Sorrenti, "Rawseeds: building a benchmarking toolkit for autonomous robotics," in *Methods and Experimental Techniques in Computer Engineering*. Springer, 2014, pp. 55–68.

[19] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1643–1649.

[20] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision*, 2010.

[21] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

[22] J.-L. Blanco, F.-A. Moreno, and J. Gonzalez, "A collection of outdoor robotic datasets with centimeter-accuracy ground truth," *Autonomous Robots*, vol. 27, no. 4, p. 327, 2009.

[23] H. Badino, D. Huber, and T. Kanade, "Visual topometric localization," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 794–799.

[24] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.

[25] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[26] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons," in *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation*, 2013.

[27] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," in *arXiv:1607.02555*, July 2016.

[28] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, "Penncosyvio: A challenging visual inertial odometry benchmark," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3847–3854.

[29] A. L. Majdik, C. Till, and D. Scaramuzza, "The zurich urban micro aerial vehicle dataset," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 269–273, 2017.

[30] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The blackbird dataset: A large-scale dataset for UAV perception in aggressive flight," in *International Symposium on Experimental Robotics*, 2018.

[31] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[32] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, 2017.

[33] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[34] Y. Zhao and P. Vela, "Good feature selection for least squares pose optimization in VO/VSLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 3569–3574.

[35] Y. Zhao, W. Ye, and P. Vela, "Low-latency visual SLAM with appearance-enhanced local map building," in *IEEE International Conference on Robotics and Automation*, 2019.

[36] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.