Optimizing SLAM Evaluation Footprint Through Dynamic Range Coverage Analysis of Datasets

Islam Ali
Department of Computing Science
University of Alberta
Edmonton, Canada
iaali@ualberta.ca

Hong Zhang
Department of Computing Science
University of Alberta
Edmonton, Canada
hzhang@ualberta.ca

Abstract—Simultaneous Localization and Mapping (SLAM) is considered an ever-evolving problem due to its usage in many applications. Evaluation of SLAM is done typically using publicly available datasets which are increasing in number and the level of difficulty. Each dataset provides a certain level of dynamic range coverage that is a key aspect of measuring the robustness and resilience of SLAM. In this paper, we provide a systematic analysis of the dynamic range coverage of datasets based on a number of characterization metrics, and our analysis shows a huge level of redundancy within and between datasets. Subsequently, we propose a dynamic programming (DP) algorithm for eliminating the redundancy in the evaluation process of SLAM by selecting a subset of sequences that matches a single or multiple dynamic range coverage objectives. It is shown that, with the help of dataset characterization and DP selection algorithm, a reduction in the evaluation effort can be achieved while maintaining the same level of coverage. We also compare our proposed algorithm with other available methods and provide evidence of its superiority regardless of the increase in the number of objectives or dataset sequences.

Index Terms—SLAM evaluation, SLAM datasets, dynamic programming, optimization

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the process of determining the location a mobile robot while it is building a map of the environment [1]. On the one hand, the SLAM problem was thought to be a solved problem [2] given the tasks and the controlled environments where such systems are usually deployed. On the other hand, deployment of SLAM in the wild creates an evolving problem with many open questions. For instance, the robustness and resilience of SLAM is a critical requirement for the deployment of SLAM in unknown and unstructured environments. Currently, one main open problem in SLAM is how to quantify robustness, ensure robustness, and design for robustness.

The robustness and resilience of SLAM cannot be determined, guaranteed, or transferred without the quantification of the design and testing conditions first, and then the validation and deployment conditions [3]. The performance of SLAM is often evaluated by subjecting it to a temporal sequence of sensor measurements in the form of a pre-recorded dataset. Thus, this gives rise to the need to provide quantitative characterization of such datasets. Usually, SLAM evaluation begins with the selection of a number of datasets where each dataset consists of a number of sequences. The selection of the datasets and their corresponding sequences are thought to capture the environment conditions and anomalies sufficiently to establish an objective evaluation of the proposed SLAM algorithm. This selection has typically been done qualitatively without an explicit consideration of the dynamic range of a certain environmental parameter. In fact, as this study proposes, there exists a huge level of redundancy and similarity among datasets and among measurement sequences in the same dataset. Thus, the analysis and identification of the coverage of dynamic range achieved by a certain experimental setup are essential in quantifying robustness of a system under test. This identification leads to the optimal selection of validation sequences to achieve the same level of coverage in terms of time and effort. With the aid of the characterization framework proposed in [4], one can identify the level of redundancy present in SLAM datasets, as shown in Figure 1, and can utilize the characterization results for optimizing the evaluation process of SLAM. In this work, we utilize the characterization results in [4] to measure the coverage of dynamic range achieved by a combination of several datasets. Based on the achieved coverage, and the characterization of each measurement sequence, an optimal subset of sequences
The rest of the paper is organized as follows. Section II discusses the motivation of the proposed work as well as a brief discussion of the related work. After that, Section III describes the problem addressed in this work and its mathematical basis. Next, Section IV describes our proposed methodology for solving the problem using a dynamic programming approach. Moreover, Section V discusses the results of the proposed method and compares its outcomes with traditional validation methods. Finally, Section VI presents our conclusions.

II. RELATED WORK

With the frequent introduction of new datasets to evaluate and benchmark SLAM everyday [5], one shall ask if another dataset is needed. In fact, a more fundamental question would be which datasets to use for evaluation given a defined evaluation objective. This work provides a needed linkage between dataset characterization and single and multi-objective SLAM evaluation and benchmarking through the analysis of dynamic range coverage of datasets.

Characterization of a dataset is the process of extracting descriptive metrics from the dataset systematically. In this work, we utilize the characterization metrics and results presented in [4]. This framework applies several characterization metrics on each data sequence in a dataset, and produces a corresponding characterization vector. The characterization metrics are divided into three groups: general, inertial, and visual characterization metrics, which are summarized in Table I. Each characterization metric measures an environmental property in the environment in which the dataset was recorded, and together those characterization metrics represent an abstract descriptor of the each sequence in a lower dimensional space.

The significance of the characterization metrics stems from the observed correlation between the characterization results and the corresponding \textit{Absolute Trajectory Error (ATE)}. In Figure 2, the non-monotonic correlation coefficient (Kendall-Tau) is measured between different characterization metrics of datasets and ORB-SLAM3 [6] ATE. We can observe medium to high correlation between those characteristics and the SLAM ATE, which suggests SLAM sensitivity to the metrics measured. Thus, coverage of such metrics must be considered in the evaluation process of SLAM.

On the other hand and similar to any single or multi-objective optimization problem [7], the goal of SLAM algorithms is to provide acceptable localization and mapping performance w.r.t. a single or a group of objectives defined by the application in which the system will be deployed. This is manifested in several SLAM algorithms where the objective could be immunity to illumination changes [8][9], ability to handle textureless situations [10][11] or handling environment dynamics [12][13][14], among many others. Due to the lack of datasets characterization results of popular SLAM datasets, researchers tend to collect their own data for evaluation in order to control different environmental parameters.

Not surprisingly, analysis of dataset properties is a crucial topic in a wide range of disciplines in science and engineering especially in learning problems [15] and in pure data analysis problems [16]. The analysis of dataset bias [3] and dataset shift [17] in the aforementioned disciplines led to methods and techniques for correcting the dataset bias [18] particularly in the context of deep neural network models [19]. Although the aforementioned research is directed to other computer vision tasks, the same concepts can be adopted in SLAM research especially in learning-based SLAM [20]. The concept of coverage pattern was introduced in [16], which uses characterization parameters as a feature vector. Moreover, the relationship between different patterns was modelled by edges in a directed graph. The approach proposed is suitable for situations where the characterization metrics are discrete and not continuous. In SLAM, however, the characterization of measurements is often a continuous variable. Thus, to utilize the same concepts mentioned in [16], quantization in the continuous space is required, leading to quantization errors and potential sub-optimal analysis of the coverage.

Consequently, the importance of coverage analysis naturally results from its relation to the proper design of the experimental setup used to evaluate a SLAM algorithm. This analysis was not possible due to the lack of quantitative characterization of SLAM datasets. However, with the availability of a quantitative characterization framework [4], the analysis can now be attempted. This study provides a novel approach to the coverage problem, by abstracting the characterization results of a dataset as continuous random variables to support the utilization of statistical analysis techniques and popular

| TABLE I: Summary of Datasets Characterization Metrics |
|-------------------------------------------------------|
| **Group** | **Characterization Metrics** |
|-------------------------------------------------------|
| **General characterization metrics (G)** | |
| Sampling and rates | - Samples, Total Duration, Sampling time, Sensor timestamps mismatch |
| Higher-order differences | - Samples, Total Duration, Sampling time, Sensor timestamps mismatch |
| Rotation-only motion | - Samples, Total Duration, Sampling time, Sensor timestamps mismatch |
| **Inertial characterization metrics (I)** | |
| Sensor saturation | - Jerk, Snap, Angular acc., Angular jerk |
| Rotation-only motion | - Dynamic range covering and crossing |
| **Visual characterization metrics (V)** | |
| Brightness | - Avg. brightness, Zero-mean avg. brightness derivative, Ratio of Thresholding |
| Exposure | - Trimmed mean, Trimmed skewness, Exposure Zone |
| Contrast | - Contrast Ratio, Weber contrast, Michelson contrast, RMS contrast |
| Blurring | - Blur score, blur percentage/image, blurred images percentage |
| Detectable features | - Avg. # features / sub-image, Avg. spatial dist. ratio, Abs. spatial dist. ratio |
| Disparity | - Avg. & std. dev. of disparity map (StereoBM - StereoSGBM) |
| Similarity | - DBoW2 score, distance to closest match |
set manipulation algorithms benefiting from their implied optimality in selecting optimal subsets of sequences. Unlike previously mentioned studies, this work is directed toward the SLAM problem and highlighting how statistical analysis and dynamic programming can be used to optimize the SLAM evaluation process.

III. PROBLEM DEFINITION

The existence of redundancy within and between datasets leads to inefficient design of experimental setup, where evaluation objectives are not guaranteed to be addressed. This gives rise to the need to select a subset of evaluation sequences from available datasets that reaches similar coverage levels for certain quantifiable objectives. For instance, we seek to answer whether the full KITTI dataset is needed to test for illumination changes or not. If not, which sequences in KITTI dataset are representing of the coverage level achieved by the full dataset. In this section, we formally describe the problem we are solving. Also, we define key parameters needed in the course of this work.

Let $Q$ be a set of sequences of a SLAM dataset where:

$$Q = \{Q_i, \ i = 1, ..., n\}$$  (1)

As such, a sequence $Q_i$ is a vector of $m_i$ measurements (e.g. images/inertial measurements), i.e. $m_i = |Q_i|$, where $i$ corresponds to a sequence.

We characterize each sequence using the method described in [4], and a corresponding characterization vector $q_i$ is generated for one of the characterization metrics mentioned in Table I so that:

$$q_i = \{q_{i,1}, ..., q_{i,m_i}\}$$  (2)

$q_{i,j}$ is a feature representing the result of applying a characterization metric $f(.)$ on sample (e.g. image) $I_{i,j}$ in sequence $i$, and is given by:

$$q_{i,j} = f(I_{i,j})$$  (3)

The characterization vector $q_i$ can be considered a continuous random variable that is represented by the inclusive bounded interval of minimum and maximum characterization value, as such:

$$\delta q_i = [\min_j q_{i,j}, \max_j q_{i,j}]$$  (4)

Thus, the dynamic range coverage for a a given dataset of sequences $Q$ under a specific characterization metric is denoted by the interval $\Delta Q$, and is given by:

$$\Delta Q = \left( \bigcup_{i=1}^{\tilde{n}} \delta q_i \right)$$  (5)

which is a bounded inclusive interval that is given by:

$$[\Delta Q_{\text{min}}, \Delta Q_{\text{max}}] = [\min_{i,j} q_{i,j}, \max_{i,j} q_{i,j}]$$  (6)

where $j$ is an index of the minimum or maximum element in the joint set $\Delta Q$.

Therefore, the outer measure [21] of the interval $\Delta Q$ is given by:

$$|\Delta Q| = |\Delta Q_{\text{max}} - \Delta Q_{\text{min}}| - \varepsilon$$  (7)

where $\varepsilon$ is the numerical discontinuity in the $\Delta Q$.

We seek to find a subset of sequences $\tilde{Q} \subseteq Q$ of size $\tilde{n}$ such that:

$$\tilde{n} \leq n \ & \ \Delta \tilde{Q} = \Delta Q$$  (8)

To find the minimum set $\tilde{Q}$, we define two quantities to compare subsets. The first one is the cost of dynamic range coverage $C(\tilde{Q})$ defined by the number of processed measurements per unit dynamic range coverage:

$$C(\tilde{Q}) = \left( \sum_{i=1}^{\tilde{n}} m_i \right) / |\Delta \tilde{Q}|$$  (9)

The second is the dynamic range coverage percentage $P(\tilde{Q})$, which is the dynamic range coverage of subset $\tilde{Q}$ relative to the full set of sequences $Q$, which is given by:

$$P(\tilde{Q}) = \frac{\Delta \tilde{Q}}{\Delta Q} \%$$  (10)
Finally, we define our problem as one of finding the subset $\tilde{Q}$ that achieves the least number of validation sequences (LS) by solving:

$$\arg\min_{\tilde{n}} \; \tilde{n} = |\tilde{Q}|$$

s.t. $\tilde{n} \leq n$, $\tilde{Q} \subseteq Q$, $\Delta\tilde{Q} = \Delta Q$  

or the lowest possible cost (LC) of dynamic range coverage by solving:

$$\arg\min_{\tilde{Q}} \; C(\tilde{Q}) = \left( \sum_{i=1}^{\tilde{n}} m_i \right) / |\Delta\tilde{Q}|$$

s.t. $\tilde{Q} \subseteq Q$, $\Delta\tilde{Q} = \Delta Q$  

The two objectives aspire reducing the footprint of SLAM evaluation while maintaining the same dynamic range coverage achieved by a pool of evaluation sequences.

To illustrate the relation between characterization metrics, and the aforementioned definition, we discuss the following example. Assume we want to evaluate a SLAM system against illumination changes (the objective characterization metric) on the KITTI dataset, denoted as $Q$, that consists of 22 sequences ($n = 22$). After applying the illumination change characterization metric $f(.)$ on each sequence $Q_i$, we obtain a characterization vector of size $m_i$ denoted as $q_i$. We calculate the dynamic range coverage $\Delta Q$, and the cost of coverage $C(\tilde{Q})$ of processing all sequences available in KITTI. We seek to find a minimum set $\tilde{Q}$ with $\tilde{n}$ sequences. This subset $\tilde{Q}$ must achieve a dynamic range coverage for illumination changes that is equivalent to that of the full KITTI dataset by either reducing the number of processed sequences ($\tilde{n} \leq n$) or reducing the cost of dynamic range coverage ($C(\tilde{Q}) \leq C(Q)$).

IV. METHODOLOGY

Given a set of dataset sequences, we seek to find an optimal minimal subset of sequences that achieves a defined evaluation objective, which is achieving a dynamic range coverage that is equivalent to the full input dataset over one of characterization metrics. As shown in Figure 3, the process starts by characterizing all sequences in the dataset w.r.t. the selected characterization metric using the method proposed in [4]. This step produces a characterization vector and a dynamic range coverage for each data sequence. After that, this information is sent to an optimization algorithm which iteratively selects an optimal subset of sequences satisfying the evaluation criteria. At each iteration, the algorithm computes the dynamic range coverage of a candidate and updates its internal state. Once all sequences are considered, the optimal subset is reported.

Computation of the dynamic range coverage is described in Algorithm 1, which depends on integrating sequences boundaries taking into consideration numerical discontinuity regions among intervals that represent sequences in the input dataset. The algorithm yields $\Delta Q$ which is the dynamic range

![Algorithm 1 Dynamic Range Coverage Calculation](image)

**Input:** $\delta q$ : List of characterization vectors  
**Output:** $\Delta Q$ : Dynamic range coverage  

**Initialization:**
1. Sort $\delta q$ list on min. value in characterization vector  
2. for $i$ in range($1$, $n$) do  
3.  $\delta = [\Delta Q_{\min}, \Delta Q_{\max}] \cap [\delta q_{i, \min}, \delta q_{i, \max}]$  
4.  if $\delta == 0$ then  
5.  $\triangleright$ compute discontinuity region $\varepsilon$  
6.  $t_{\max} \leftarrow \max(\Delta Q_{\max}, \delta q_{i, \max})$  
7.  $t_{\min} \leftarrow \min(\Delta Q_{\min}, \delta q_{i, \min})$  
8.  $\varepsilon + = (t_{\max} - t_{\min}) - |\Delta Q| - |\delta q_i|$  
9.  end if  
10.  $\Delta Q_{\max} \leftarrow \max(\Delta Q_{\max}, \delta q_{i, \max})$  
11.  $\Delta Q_{\min} \leftarrow \min(\Delta Q_{\min}, \delta q_{i, \min})$  
12.  end for  
13. Set $\Delta Q = [\Delta Q_{\min}, \Delta Q_{\max}]$  
14. Set $|\Delta Q| = \Delta Q_{\max} - \Delta Q_{\min} - \varepsilon$  
15. Return: $\Delta Q$

![Fig. 3: A block diagram of the system flow illustrating different system parameters](image)
coverage of a given set of sequences $\hat{Q}$.

On the other hand, the problem of finding the optimal subset of intervals to match a target dynamic range was historically solved using greedy-based approaches [22], due to their simplicity, resulting in an acceptable sub-optimal solution in a polynomial time.

A. Greedy algorithms

Greedy algorithm is a methodology for solving optimization problems that depends on selecting the best available options at the time of decision [23]. The algorithm does not allow rolling back a taken decision based on observed better alternatives. Thus, it can yield a sub-optimal or even a non-optimal solution upon termination due to its reliance on achieving local-optimality as illustrated in Algorithm 2.

As the problem definition suggests, one can abstract the problem of finding the optimal set of sub-intervals to match the range of a target interval to the famous knapsack problem [24] where the optimal subset of objects are selected to fill a knapsack with defined capacity. Consequently, dynamic programming solutions can be used for the problem to achieve optimal solution with polynomial time [25].

B. Dynamic Programming (DP)

DP is used for solving an optimization problem by dividing it into smaller and easier sub-problems. The final optimal solution is an incremental compilation of the solution of the sub-problems [23]. Moreover, DP has the ability of providing optimal solutions while maintaining reasonable and linear time complexity when the optimization space is discrete, and can provide near-optimal solution when the space is continuous due to the need to perform quantization. Tabulation-based techniques in DP depend on removing redundant calculations of sub-problems [26] ensuring the calculation of any sub-problem once at most as provided in Algorithm 3. The result of any sub-problem is stored in a table-like data structure and is re-used when needed. Similar to any recursive-based solution, a base case has to be defined and is used as a program entry point. In our case, the dynamic range coverage percentage $P(Q)$ is quantized into 10 regions where each represents 10% coverage of the range. In addition to that, an initial empty state representing 0% coverage have to be defined as a base case. Thus, the DP algorithm defines 11 states, where the first is the base state, and the last is required subset equivalent to $P(Q) = 100\%$. In tabulation-based DP, the optimization objective is embedded in the algorithm by defining the replacement function which is responsible for replacing the current state of a table cell with another. The replacement function behaviour changes based on the optimization objective. For instance, the dynamic range coverage of the potential state solution is computed and is compared to the current state solution. The one with higher dynamic range coverage is

Algorithm 2 Greedy Optimization Algorithm

Input: $Q$: set of sequences to optimize
Input: $\delta Q$: set of characterization vectors
Output: $\hat{Q}$: optimal subset of sequences

Initialization:
1. $P(\hat{Q}) \leftarrow 0$, current coverage percentage
2. $\hat{Q} \leftarrow \{\}$, empty set

BEGIN: Greedy Optimization Approach
1. Sort $\delta Q$ list on min. value in characterization vector
2. $i \leftarrow 1$, current sequence to process
3. while $P(\hat{Q}) < 100\%$ do
4. if $\delta q_{i,max} > \Delta Q_{max}$ then
5. Update $\Delta Q_{max} \leftarrow \delta q_{i,max}$
6. end if
7. if $\delta q_{i,min} < \Delta Q_{min}$ then
8. Update $\Delta Q_{min} \leftarrow \delta q_{i,min}$
9. end if
10. Add $Q_i$ to $\hat{Q}$
11. Update $P(\hat{Q}) \leftarrow (|\Delta \hat{Q}|/|\Delta Q|)\%$
12. Increment $i$
13. end while
14. Return: $\hat{Q}$

Algorithm 3 Dynamic Programming Algorithm

Input: $Q$: set of sequences to optimize
Input: $\delta Q$: set of characterization vectors
Input: $s$: number of states
Output: $\hat{Q}$: optimal subset of sequences

Initialization:
1. $T_{2,s} \leftarrow \{\}$, empty set
2. $T_1 \leftarrow \{\}$

BEGIN: DP Optimization Approach
1. for $i$ in range(1, $s$) do
2. for $j$ in $n$ do
3. $L \leftarrow \{\}$, temp list to current subset candidate
4. if $T_i$ not null then
5. $L \leftarrow T_i \cup q_j$
6. $P(L) \leftarrow (\Delta(L))/|\Delta Q|\%$
7. $loc \leftarrow |P(L)|/10 + 1$
8. if $loc < 11$ then
9. if $T_{loc}$ not null then
10. $T_{loc} \leftarrow L$
11. else
12. if $|\Delta(T_{loc})| < |\Delta(L)|$ then
13. $T_{loc} \leftarrow L$
14. end if
15. if $|\Delta(T_{loc})| = |\Delta(L)|$ then
16. Execute: Replacement Fn.
17. end if
18. end if
19. end if
20. end if
21. end for
22. end for
23. Set $\hat{Q} \leftarrow T_s$
24. Return: $\hat{Q}$
selected. Otherwise, either the one with the least number of sequences or the least coverage cost is selected.

Support of multiple objectives is also conducted to allow SLAM researcher to optimize for a number of objectives at once when needed. Formerly, DP states are defined by the dynamic range coverage percentage \( P(\tilde{Q}) \). For multiple objectives, DP states are defined by the average dynamic range coverage percentage \( \bar{P}(\tilde{Q}) \), which is given by:

\[
\bar{P}(\tilde{Q}) = \frac{1}{k} \sum_{i=1}^{k} P(\tilde{Q})_i
\]

(13)

where \( k \) represents the number of parameters determine the performance and level of optimality the DP algorithm can achieve. These parameters are the granularity of continuous space quantization, the choice of the state replacement function, and the choice of the state aggregation method. For instance, the definition of the DP states has a huge influence on the performance specially when the space we want to cover is continuous.

In order to enable multiple objective optimization to happen in polynomial time, two extensions to the original DP algorithm are introduced:

1) Adaptive state quantization: Since the dynamic range coverage is a continuous random variable, the need to perform quantization is inevitable. This introduces some shortcomings which are improper choice of granularity of quantization levels and can lead to local minima, which will result in the inability of the DP algorithm to achieve a solution. For that reason, we enhanced the traditional DP structure to adaptively selecting the quantization level when a solution cannot be reached as such, the algorithm refines the quantization granularity gradually by increasing the number of states represents the total dynamic coverage range when until a solution is reached.

2) State aggregation: Aggregation in dynamic programming is a well-known method used in order to limit the number of states to maintain the polynomial time complexity of the algorithm [27]. With the introduction of multiple objectives optimization, representing all possible combinations of objectives is not feasible specially when dealing with an increasing number of objectives. For that reason, state aggregation is performed as such, the average dynamic range coverage percentage \( \bar{P}(\tilde{Q}) \) is used to represent the state space on which quantization is performed.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present our experimental setup, results, and discussion. Three datasets, KITTI [28], EuroC-MAV [29], and TUM-VI [30], and their characterizations were used to demonstrate the performance of the proposed method. When combined, they provide a total of 61 sequences. The corresponding average combined dynamic range is used as the target coverage dynamic range (100% coverage). Characterization metrics were divided into four groups. The first three groups are general, visual, and inertial characterization metrics, while the fourth one represents these three groups combined. To examine the performance of our algorithm against multiple objective (i.e. characterization metrics summarized in Table I) optimization, we vary the number of objectives to cover \( \epsilon \) in the range \( \epsilon \in [1, k] \), where \( k \) is the number of objectives in a given group. For each step \( \epsilon \), we apply the proposed method on 100 randomly selected subsets objectives each of size \( \epsilon \). The process is conducted twice to cover two optimization goals, which are: least number of sequences (LE) and least possible cost (LC).

A. Optimal subset selection given an evaluation objective

A greedy algorithm is defined to be the baseline to which we compare our results in terms of the subset size and cost of the produced evaluation subset. For that to happen, a heuristic must be selected to govern the operation of the greedy approach, which was selected to be the average dynamic range coverage percentage when optimizing for least experiments goal, and the average cost when optimizing for least cost goal. Figure 4 shows the performance of the DP algorithm compared to the greedy approach on the four characterization metrics group while varying the number of objectives to cover for the two aforementioned goals. We can observe the superiority of the dynamic programming algorithm in selecting the optimal subset compared to the greedy approach regardless of the objective group, number of objectives to optimize for, and optimization goal. The superior performance is achieved in polynomial time and is a result of the adaptive setting of quantization states. As mentioned in Sec IV, the DP algorithm adjusts the number of states adaptively to a finer granularity when a certain quantization level is not able achieve a solution, and stops when a solution is acquired. Figure 5 provides an histogram of different quantization levels and their success rate. As shown, the adaptive setting of quantization steps is very useful when objectives are originating from different groups, allowing the algorithm to be more general and suitable for multi-sensor SLAM systems.

B. Time and memory complexity analysis

One of the main advantage of dynamic programming approaches is the ability to reach an optimal solution in polynomial time compared to brute force solutions. However, this comes with the cost of using more memory. To explain the trade-off, we compare the time and memory complexity of our proposed DP approach with the greedy approach (baseline), exhausted search approach (brute force), and random search approach. As shown in Table II, one can observe the ability of the DP algorithm to achieve a near optimal solution in polynomial time while maintaining an acceptable memory complexity. On the other hand, other solutions are not par in either their optimality level (e.g. random search and greedy approaches) or in their time complexity (e.g. brute force exhausted search).

C. DP optimization limitations

A number of parameters determine the performance and level of optimality the DP algorithm can achieve. These parameters are the granularity of continuous space quantization, the choice of the state replacement function, and the choice of the state aggregation method. For instance, the definition of the DP states has a huge influence on the performance specially when the space we want to cover is continuous.
Fig. 4: Evaluation subsets resulting from both greedy and DP approaches while optimizing for least experiments (first row) and least cost (second row). Each column represents one objectives group (all metrics, general, inertial, and visual respectively). Each figure represents the average evaluation subset size and the standard deviation of running the algorithm on 100 randomly selected set of objectives.

Fig. 5: Quantization step size and their success rate for each optimization group.

TABLE II: Time and memory complexity of different approaches compared to their optimality level

| Method              | Time     | Memory | Optimality |
|---------------------|----------|--------|------------|
| Brute force         | $O(n \times 2^n)$ | $O(n)$ | Optimal    |
| Random search       | N/A*     | $O(n)$ | Not optimal|
| Greedy approach     | $O(n \log(n))$ | $O(n)$ | Not optimal|
| DP algorithm (ours) | $O(n \times m)$ | $O(m)$ | Sub optimal|

* time complexity depends on the no. of iterations tried

$n$ : no. of sequences

$m$ : no. of characterization metrics

In nature and quantization is required. Therefore, the granularity of quantization has a direct impact on the optimality and time/memory complexity. Moreover, the definition of the replacement function in case of collision also has an impact on the performance as it can lead to local minima. Finally, the choice of the state aggregation method can also add to the algorithm sub-optimality if improper aggregation is conducted. Despite the mentioned shortcomings of the DP algorithm, it can still outperform the greedy approach.

VI. Conclusions

In this paper, the problem of measuring the dynamic range coverage of SLAM was discussed. The study started with a brief review of the topic in closely related disciplines. After that, we introduced an approach for optimizing the selection of the validation set with two different optimization objectives: minimization of the number of validation sequences, and
minimization of the total cost of dynamic range coverage. The results of each optimization objective were presented and discussed. After that, the results for each characterization category were presented in detail. It was shown that the DP algorithm was able to provide a superior performance in selecting the evaluation dataset compared to greedy algorithm, while maintaining polynomial time complexity. Utilizing the DP-based approach provided a reduced mix of sequences that achieves the same coverage objectives of the whole evaluation pool of multiple datasets. The reduction achieved highlights the redundancy present in SLAM datasets, and provides a systematic approach to design SLAM experiments given defined criteria. This work directs the attention of SLAM evaluation from quantity to quality and provide a framework for objective evaluation of SLAM. This shall open doors to proper measurement of robustness and resilience of SLAM solutions in a quantitative manner.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” IEEE robotics & automation magazine, vol. 13, no. 2, pp. 99–110, 2006.
[2] U. Frese, “Interview: Is slam solved?” KI-Künstliche Intelligenz, vol. 24, no. 3, pp. 255–257, 2010.
[3] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in CVPR 2011. IEEE, 2011, pp. 1521–1528.
[4] I. Ali and H. Zhang, “Are we ready for robust and resilient slam? a framework for quantitative characterization of slam datasets,” in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 2810–2816.
[5] Y. Liu, Y. Fu, F. Chen, B. Goossens, W. Tao, and H. Zhao, “Simultaneous localization and mapping related datasets: A comprehensive survey,” arXiv preprint arXiv:2102.04036, 2021.
[6] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Obh-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,” IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, 2021.
[7] K. Deb, “Multi-objective optimization,” in Search methodologies. Springer, 2014, pp. 403–449.
[8] S. Park, T. Schöps, and M. Pollefeys, “Illumination change robustness in direct visual slam,” in 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017, pp. 4523–4530.
[9] A. Ranganathan, S. Matsumoto, and D. Istroum, “Towards illumination invariance for visual localization,” in 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 3791–3798.
[10] Y. Dong, S. Wang, J. Yue, C. Chen, S. He, H. Wang, and B. He, “A novel texture-less object oriented visual slam system,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 36–49, 2019.
[11] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, “Struckslam: Visual slam with building structure lines,” IEEE Transactions on Vehicular Technology, vol. 64, no. 4, pp. 1364–1375, 2015.
[12] L. Cui and C. Ma, “Sof-slam: A semantic visual slam for dynamic environments,” IEEE access, vol. 7, pp. 166 528–166 539, 2019.
[13] D. Zou and P. Tan, “Coslam: Collaborative visual slam in dynamic environments,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 2, pp. 354–366, 2012.
[14] N. Brasch, A. Bozic, J. Lallemant, and F. Tombai, “Semantic monocular slam for highly dynamic environments,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 393–400.
[15] S. Mani, A. Sankaran, S. Tamilselvam, and A. Sethi, “Coverage testing of deep learning models using dataset characterization,” arXiv preprint arXiv:1911.07209, 2019.
[16] A. Asudeh, Z. Jin, and H. Jagadish, “Assessing and remedying coverage for a given dataset,” in 2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, 2019, pp. 554–565.
[17] T. Turhan, “On the dataset shift problem in software engineering prediction models,” Empirical Software Engineering, vol. 17, no. 1, pp. 62–74, 2012.
[18] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, “Undoing the damage of dataset bias,” in European Conference on Computer Vision. Springer, 2012, pp. 158–171.
[19] A. Kortylewski, B. Egger, A. Morel-Forster, A. Schneider, T. Gerig, C. Blumer, C. Reyneke, and T. Vetter, “Can synthetic faces undo the damage of dataset bias to face recognition and facial landmark detection?” arXiv preprint arXiv:1811.08565, 2018.
[20] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, “A comprehensive survey of visual slam algorithms,” Robotics, vol. 11, no. 1, p. 24, 2022.
[21] W. Rudin et al., Principles of mathematical analysis. McGraw-Hill New York, 1976, vol. 3.
[22] A. Prolubnikov, “The set-cover problem with interval weights and the greedy algorithm for its solution,” Comput. Technologies, vol. 20, no. 6, pp. 70–84, 2015.
[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to algorithms. MIT press, 2022.
[24] H. M. Salkin and C. A. De Kloeyer, “The knapsack problem: a survey,” Naval Research Logistics Quarterly, vol. 22, no. 1, pp. 127–144, 1975.
[25] P. Toth, “Dynamic programming algorithms for the zero-one knapsack problem,” Computing, vol. 25, no. 1, pp. 29–45, 1980.
[26] R. S. Bird, “Tabulation techniques for recursive programs,” ACM Computing Surveys (CSUR), vol. 12, no. 4, pp. 403–417, 1980.
[27] J. C. Bean, J. R. Birge, and R. L. Smith, “Aggregation in dynamic programming,” Operations Research, vol. 35, no. 2, pp. 215–220, 1987.
[28] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012, pp. 3354–3361.
[29] 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.
[30] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Sturm, and D. Cremers, “The tum vi benchmark for evaluating visual-inertial odometry,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1680–1687.