Overlap Displacement Error: Are Your SLAM Poses Map-Consistent?

Christian Mostegel, Jianbo Ye, Yu Luo and Yang Liu

***Abstract*— Localization is an essential module that supports many intelligent functions of a mobile robot such as transportation or inspection. However, justifying that a localization module is *sufficiently* accurate for supporting all downstream tasks is one of the most difficult questions to answer in practice. To overcome this problem, we move away from the traditional calculation of pose errors and propose a new approach that instead evaluates the potential map inconsistency introduced by those pose errors.**

For this purpose, we propose a new metric, which we call Overlap Displacement Error (ODE). This metric measures the relative displacements between multiple overlapping sensor frustums with respect to the ground truth. All you need to compute this metric are a query trajectory, a ground truth trajectory and the sensor frustum used for mapping. Having the sensor frustum and the map representation as part of the metric, the ODE is customized to the hardware configuration and the mapping strategy. This design allows the analysis of pose accuracy in a space that matters to map creation, and also allows the identification of problems sitting in the interplay between localization and mapping. We demonstrate the potential of this new analysis tool on synthetic and the real-world sequences.

I. Introduction

Autonomous robots are becoming part of our technology-driven world, where they are fulfilling increasingly challenging tasks, such as transportation, environment inspection, and human assistance. These tasks often require autonomous navigation within a partially known world representation, which is typically referred to as *map*.

For creating and updating such a map with new information, there are two kinds of information sources required. The first information source is the obstacle detection module that can detect obstacles within a certain distance of the robot. For high reliability, multiple sensors of different types often contribute to a shared map representation. The second information source is the localization module, which estimates the location of the robot with respect to the current map. The localization module is crucial for associating readings from different obstacle detection sensors in time and space. An accurate localization allows building and updating a consistent map representation.

While it is a well-known fact that the quality of a map-representation strongly depends on the accuracy and robustness of localization, it is hard to answer the following questions in practice: What localization accuracy is "good enough"? Is absolute trajectory error or relative trajectory error more important? How do the angular and linear aspects of the relative trajectory error impact the navigation performance?

All authors are with Amazon (contact: mostegec@amazon.com)

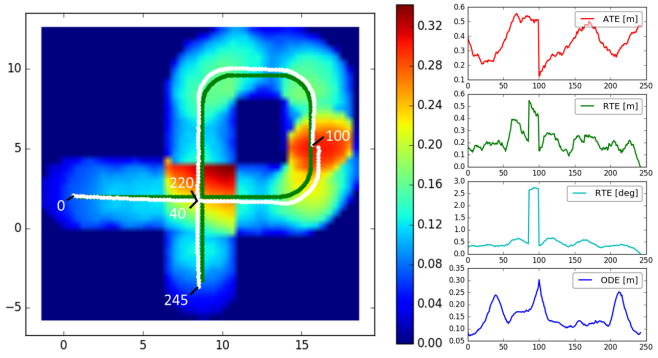

Fig. 1. Our new metric, the Overlap Displacement Error (ODE), measures the map-consistency of localization, which conventional metrics do not provide. On the left, we show a ground truth trajectory (green line), a query trajectory (white line e.g. from a SLAM algorithm) and an error heatmap (in meter) that summarizes the error that localization could potentially induce into the map, if we would use a 360° range sensor with 2m range. On the right, we show different error metric plots for localization. From top to bottom, we show the Absolute Trajectory Error (ATE), the Relative Trajectory Error (RTE - linear [m] and angular [deg]) and the ODE. Note how the ODE focuses on map-consistency and has clear peaks at the discontinuity of poses (100-th frame) and the region where the trajectory imperfectly intersects itself (40-th and 220-th frame), while the other metrics are harder to interpret.

The reason why these questions are so difficult to answer is that they are posed in the space of trajectories, which has no clear meaning to localization consumers, such as path planning and navigation. E.g. If a robot drives 1km in one direction and accumulates 10m absolute trajectory error, it is unclear if such an error would cause any degradation to the system performance. However, if the robot is driving a loop and fails to close the loop, the same magnitude of error may create shifted duplicates in the same area, which significantly increases the risk of a system failure. This motivated us to leave the space of traditional accuracy metrics, and instead work backwards from the perspective of localization consumers.

We consider the path planner — one of the key modules in autonomous robots — that consumes localization and map information to direct the robot motion. In order to succeed, the path planner requires *map-consistency* – i.e. a localization module which delivers poses that result in a consistent map and poses which are themselves consistent with the existing map. With such map consistency, the path planner can then derive a navigation plan that is consistent with its own perception of the world.

This observation motivated us to develop a new metric, which measures the impact of localization errors with respect to the notion of map-consistency. We call this new metric the Overlap Displacement Error (ODE), as it measures the error

that localization induces into the map through displacing overlapping sensor readings as shown in Fig. 1. In our experiments, we demonstrate how this new metric can be used to understand the impact of imperfect localization on the map consistency.

II. Related Work

In the robotics community, there are two standard metrics for the accuracy of localization – the absolute trajectory error (ATE) and the relative trajectory error (RTE). While they are used in slightly different ways in the literature [7], [1], [9], [3], [10], [12], [16], [15], the core ideas of these metrics stay the same. Let us first shortly explain the pros and cons of these metrics (for a deeper review see e.g. [15]).

For the ATE, a query trajectory is aligned to a ground truth trajectory and then the error of each pose of the query trajectory is computed with respect to the corresponding ground truth pose. This approach works well if the error made by localization can be nicely modeled with a Gaussian distribution in a global coordinate frame. However, sudden jumps or outliers in the trajectory can cause severe problems to the alignment procedure (see e.g. the discontinuity in Fig. 1). This metric severely penalizes drift (especially angular drift) over long sequences. For this reason, this metric is adopted for applications that require absolute pose accuracy (e.g. during GPS stabilized flight). For evaluating odometry (where only local accuracy matters), this metric only has limited meaning. E.g. if you have an odometry system that delivers perfect relative pose updates for 99% of the time, but in some edge case makes a severe (angular) mistake, the ATE might blow up disproportionately depending on where the mistake happens in the sequence.

When RTE is used, the aforementioned problem does not occur. The idea of this metric is to only look at smaller parts of the trajectory (i.e. sub-trajectories). Each sub-trajectory is aligned only based on its first pose, then the angular and translational error are measured after a distance d. While this metric allows for meaningful local statistics (e.g. what is the average drift error over 10m), it is not clear how to rank approaches based on this metric. For example, if one approach is better in translation but worse in rotation than the other (or better at $d = 1m$ but worse at $d = 2m$), it is difficult to tell which of the two should be preferred in practice. The reason why this has been so hard is that the calculation of the metric is independent of the localization consuming modules' context. In this paper, we argue that if we look at the problem from the localization consumer's side, the resulting metric becomes significantly easier to interpret.

Additional to the two standard metrics, there are some other metrics found in literature for localization benchmarking (typically in addition to the trajectory errors). Bodin et al. [1] evaluate frame rate, power consumption and reconstruction error after ICP alignment. Shi et al. [9] define a Correct Rate for the number of poses that have an absolute trajectory error and absolute orientation error below a certain threshold. The thresholds are set by hand for each scene differently according to the scene size and the expected drift of the SLAM algorithm. This definition of correctness allows the authors to remove outliers from computing the ATE and RTE. Additionally, they also introduce a measure for the correctness of re-localization. In the domain of avionics, there are standardized localization metrics defined for GPS [6], where accuracy is only one of eleven quality metrics. The other ten metrics (e.g. availability, integrity and reliability) can be seen as additions to the accuracy metric, such as the metric proposed in this paper.

Aside from evaluation metrics, there are a few works on map consistency. Hähnel et al. [4] detect map inconsistency by superimposing a range scan onto a local occupancy grid and measuring positive and negative measurement information. Yue et al. [14] measure the pair-wise inconsistency of sub-maps by modeling a truncated Gaussian distribution from the occupancy iterative closest point distances. Mazuran et al. [5] design a measure for map consistency that analyzes the distance of polygonal chains created from 2D range scans. While all three approaches demonstrate clear advantages for system self-assessment and correction, they all share the same issues with respect to evaluating the pure localization quality. First, the approaches are designed for a binary decision (consistent or not) and the magnitude of inconsistency cannot be reliably evaluated due to sensor noise and data association uncertainty. Second, relying on the range sensor readings, sensor problems (such as calibration issues or wrong measurements due to reflections etc.) are not separable from pure localization problems. In contrast, our proposed metric allows for a clear separation of sensor and localization problems through the capability of simulating an idealized sensor, if this is desired.

III. Overlap Displacement Error

The main idea of the Overlap Displacement Error (ODE) is that we do not measure the pose error directly, but only the effect that this pose error potentially has on map consistency. This means the ODE of a single pose is influenced by all other poses, which have an overlapping sensor frustum with the evaluated pose of interest.

In this work, we evaluate the sensor overlap using a grid map representation and a sensor frustum model (e.g. a cone area for a depth camera as shown in Fig. 2 or a circle area for 360° range sensor in Fig. 1). We will refer to the area that a range sensor can observe at a given time i as the sensor footprint $\mathcal{F}^{(i)}$.

For computing the ODE between two trajectories, we require a sequence of query poses $\{\mathbf{q}_i\}_{i \in T}$ (which comes e.g. from a SLAM algorithm) and their corresponding ground truth poses $\{\mathbf{g}_i\}_{i \in T}$ for a set of discrete time stamps T. For constructing the set T, we use the time stamps of the depth sensor readings. We use the expression *pose* as it is most commonly used in robotics, i.e. as the transform of a coordinate frame on the robot (e.g. `base_link`) to the world coordinate frame (e.g. `query_world`). Note that the ODE can be computed in 2D and 3D, however in this paper, we focus on the 2D case (i.e. $\mathbf{q}_i$ and $\mathbf{g}_i \in \mathrm{SE}(2)$).

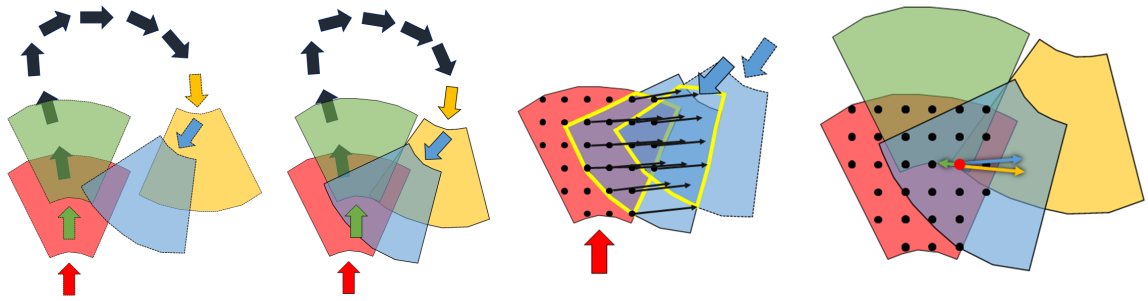

Fig. 2. In this example, we show (a) the ground truth trajectory and (b) a query trajectory (e.g. from a SLAM algorithm). The arrows symbolize poses and the transparent cone shape the sensor footprint (e.g. of a depth camera) at different time stamps. The color links a pose (i.e. arrow) and the corresponding footprints (i.e. cones). For the black poses, we do not show the sensor footprints for simplicity. In this example, the green, yellow and blue time stamps are all neighbors of the red time stamp, because their footprints overlap in the query trajectory. In (c), we show how we compute the displacement of red time stamp with respect to the blue time stamp. For the blue time stamp, we show the ground truth with dashed lines and the query with a solid line. The yellow region marks the overlap in the query trajectory and the black dots symbolize the centers of a regular grid. The black arrows visualize the induced displacement from the query to the ground truth (note that ground truth and query are aligned in the red time stamp). In (d), we show all the displacements that are used for computing the ODE of the grid cell visualized with the red dot.

Let us define the set of neighbor time stamps N_i for a given time stamp i as the set of time stamps where the other sensor footprints overlap in the query trajectory; i.e.

$$N_i = \left\{ j \in T \,\middle|\, \mathcal{F}_q^{(i)} \cap \mathcal{F}_q^{(j)} \neq \emptyset \text{ and } i \neq j \right\}, \tag{1}$$

where $\mathcal{F}_q^{(i)}$ is the sensor footprint in the coordinate frame of the query poses at time i (see Fig. 2b). In our implementation, we model the sensor foot print using a regular grid, which reduces the overlap evaluation to the look up of grid map values. Note that we use the frame of the query poses to construct N_i, as this represents the self-consistency of the resulting map.

As we model each sensor footprint using a regular grid, we can now compute the displacement of each grid cell center. From a high level perspective, we compute the displacement vector that displaces a cell point c (in the coordinate system of the query poses) from the footprint $\mathcal{F}_q^{(j)}$ to a new position inside the (virtual) footprint $\mathcal{F}_g^{(j)}$ of ground-truth pose $\mathbf{g}'_j$ after aligning the ground-truth trajectory g to the query trajectory q at a given time stamp i as shown in Fig. 2c. However, the necessary displacement transform $D_{i,j}$ can be computed directly without explicitly transforming the ground truth trajectory as:

$$D_{i,j} := \mathbf{q}_i \mathbf{g}_i^{-1} \mathbf{g}_j \mathbf{q}_j^{-1}, \tag{2}$$

where i indicates the time stamp used for aligning the poses, j is a time stamp of query pose $\mathbf{q}_j$ that has footprint overlap with query pose $\mathbf{q}_i$. Using this transform, we define the displacement of a grid cell center c as:

$$\mathbf{d}(c, i, j) := D_{i,j} \circ \mathbf{x}_c - \mathbf{x}_c, \tag{3}$$

where $\mathbf{x}_c$ are the coordinates of the grid cell center c in the query world coordinate frame and the operator $\circ$ transforms $\mathbf{x}_c$ by the displacement transform $D_{i,j}$.

Finally, we define the ODE at time stamp i as the mean absolute displacement required to move a cell point c to its neighbors $N_{c,i}$:

$$\mathrm{ODE}(c, i) = \frac{1}{|N_{c,i}|} \sum_{j \in N_{c,i}} \|\mathbf{d}(c, i, j)\|, \tag{4}$$

with $N_{c,i} := \left\{ j \,\middle|\, c \in \mathcal{F}_q^{(i)} \cap \mathcal{F}_q^{(j)} \right\} \subseteq N_i$. Intuitively, Eq. (4) can be interpreted as the mean energy required to make the grid cell center c at time stamp i consistent with all other measurements that are fused into the same map location.

Theoretical Motivation. Let us consider a grid-based mapping module. For better illustration, we only focus on object measurements and neglect the free space for now. Let us denote a specific object point with c. Assume that this object point c is observed at m different poses $\mathbf{q}_{i_1}, \ldots, \mathbf{q}_{i_m}$ and its relative locations to each device frame are $\mathbf{r}_1, \ldots, \mathbf{r}_m$. Then, let us define $\mathcal{E}_c$ to be the mean map insertion error of the object point c in global coordinate relative to its first mapped location at $\mathbf{x}_c = \mathbf{q}_{i_1} \circ \mathbf{r}_1$ (without loss of generality):

$$\mathcal{E}_c := \frac{1}{m-1} \sum_{1<k\leq m} \|\mathbf{q}_{i_k} \circ \mathbf{r}_k - \mathbf{q}_{i_1} \circ \mathbf{r}_1\| \tag{5}$$

Then let us use the true location of c as $\mathbf{y}_c = \mathbf{g}_{i_k} \circ \mathbf{r}_k, \forall k$ (under the assumption of an error free ground truth) to get:

$$\mathcal{E}_c = \frac{1}{m-1} \sum_{1<k\leq m} \|\mathbf{q}_{i_k} \mathbf{g}_{i_k}^{-1} \circ \mathbf{y}_c - \mathbf{q}_{i_1} \mathbf{g}_{i_1}^{-1} \circ \mathbf{y}_c\| \tag{6}$$

and then we back-substitute $\mathbf{y}_c \equiv \mathbf{g}_{i_1} \mathbf{q}_{i_1}^{-1} \circ \mathbf{x}_c$ to Eq. 6 to obtain:

$$\mathcal{E}_c = \frac{1}{m-1} \sum_{1<k\leq m} \|D_{i_k,i_1} \circ \mathbf{x}_c - \mathbf{x}_c\|. \tag{7}$$

From comparing Eq. (7) and Eq. (4), we can see that $\mathcal{E}_c = \mathrm{ODE}(c, i_1)$, if $N_{c,i_1} = \{i_2, \ldots, i_m\}$. In other words, this means that the ODE of an object point c at time stamp i can be regarded as the mean insertion error of all other measurements of this object point with respect to the first measurement location at time stamp i.

We would like to note that Eq. (4) does not know anything about occupancy states of c: it treats object points and free space the same. In other words, the formulation sees all grid coordinates as equally unique (as if each grid cell had its own class label). On the one hand, this means that there is a remaining gap between the ODE and actual map errors. On the other hand, the formulation retains sufficient flexibility to be adapted to many different practical use cases.

IV. Using the ODE

The ODE metric can be regarded as a tool that enables researchers to analyze the interplay between localization accuracy and map-consistency. It is designed to answer the questions of the following type: (1) Which localization approach is better suited for maintaining a local map around the robot for a given type of sensor with respect to a specific target mapping accuracy? (2) How can we find a good trade-off between local accuracy and global accuracy, that suits the need of the final application? (3) Suppose that we already have a specific localization module, is it good enough to support a farther sensing range? If not, in what order of magnitude do we have to improve?

All these types of question are very hard to answer with one rigid type of metric. Thus, we designed the ODE as a framework that can be adapted to answer specific questions about the interplay of localization and map-consistency. There are two main interfaces to make the questions more specific, i.e. modeling the sensor footprint and the neighborhood definition.

Choice of sensor footprint. For increasing the expressiveness of the metric, it makes sense to choose the sensor footprint as close to reality as possible. Indeed, it is possible to use the real sensor measurements to constrain the sensor footprint for each time stamp. However, both approaches have their merit: Without sensor readings, one can simulate different sensor setups and sensor settings using the same trajectory and evaluate how well the localization module and the mapping sensor would work together. However, for far-range sensors deployed in tight environments (e.g. Lidar), this approach might be overly pessimistic as it ignores occlusion boundaries. If however, geometric ground truth is available (e.g. from simulation or from a registered map) this problem could be mitigated. With real sensor readings, the sensor footprint can be reduced the actual range measurements, however, sensor problems (such as calibration issues and sensor failure modes) might cause problems in the evaluation. In our experiments, we demonstrate how both approaches can be deployed on real world data.

Choice of neighborhood. Depending on the application, a robotic system might have a local map representation (that aims to be locally consistent for local navigation) or a global map representation (that aims to enable global path planning), further, the map might be generated in real-time (which cannot use measurements from the future) or in an offline procedure (where the acquisition time loses its importance for static objects). For each map type, the ODE neighborhoods can be used to model the mapping process into the ODE computation. In our experiments, we demonstrate this process for a robot-centered local map representation, a real-time global map representation and an offline global map representation.

Interpretation. The basic idea of the ODE is to propagate the localization error into a relative map frame. This relative map frame is always centered on the pose for which one wants to compute the ODE (this is closely related to how the RTE is computed). Through this formulation, the ODE still only represents the localization error, but in a world that is easier to interpret with respect to mapping. Using the ODE, one has to be aware that it does not represent the map error itself, but the magnitude of the spatial map insertion error. This can be interpreted as the "degree of spatial blurriness" if obstacles are close to the assessed grid cell in the sensor footprint. On the other hand, one has to be aware that the ODE might be high although the grid cells are confidently clear of any obstacle, as its current formulation has no interface for the concept of objects and free space.

Computational cost. The computational cost of the ODE metric depends on several factors. In theory, the computational complexity is $\mathcal{O}(t \cdot n \cdot c)$, where t is the number of evaluated sensor time stamps, n the average number of neighbors per grid cell and c the average number of grid cells per sensor footprint. This means that in the worst case (i.e. with $n = t$), the run-time is quadratic in the number of evaluated sensor time stamps. However, in practice n is typically multiple orders smaller than t as the robot typically moves much farther than its maximum sensor range over a full sequence. In most cases (such as a robot moving at constant velocity), n can be bounded by small constant. In this case, the computational cost becomes linear in the number of evaluated sensor time stamps/poses, just like the ATE or the RTE. The average number of grid cells per sensor footprint c is quadratic in the grid cell size. In our experiments, we show that the ODE has a low sensitivity to the grid cell size and the number of evaluation time stamps per second, which means that these parameters can be used to accommodate the available compute budget.

V. Experiments

For our experiments, we implemented three versions of the ODE metric. The reason why we implemented three versions is that each version is tailored to one specific application. The first version is the **Offline-ODE** as defined in the previous section. Offline-ODE is meant to simulate the impact of localization on a global map representation irrespective of the order that measurements were obtained in. This can be seen as the evaluation of an algorithm that stores all measurements and builds a map at the end of the session with all available information.

The second version is the **Online-ODE**. The difference to the Offline-ODE is that the ODE for a time stamp i is only computed with neighbor time stamps of the past (i.e. $j < i, \forall j \in N_i$). This version is meant to evaluate real-time mapping, where the system cannot look into the future

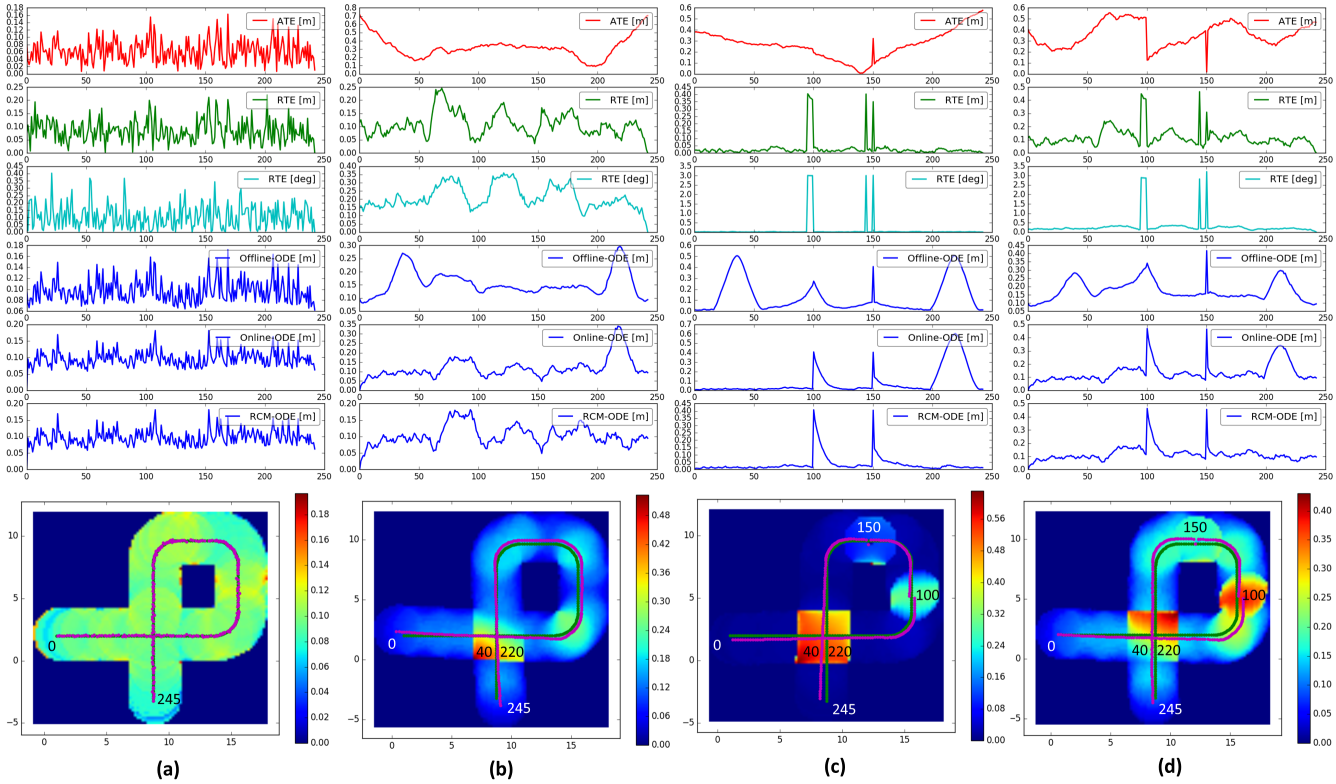

Fig. 3. Comparison of accuracy metrics. From top to bottom, we show the error profile of Absolute Trajectory Error (ATE), the Relative Trajectory Error (RTE) and three versions of the ODE. In the bottom row, we show the ground truth trajectory in green and the aligned query trajectory (with different noise) in purple. The heat map represents the error average Offline-ODE per grid cell in the map (i.e. the average displacement error in the map space).

and can only use the information up to the most recent measurement.

The third version demonstrates how the ODE metric can be adapted to a mapping strategy with a temporal change of the sensor footprints. For this purpose, we have selected the popular robot-centric map (RCM) representation of Frankhauser and Hutter [2]. In this representation, we only maintain a small rolling window around the robot to build dynamic maps. This means that when the robot moves, measurements behind the robot drop off the map and are forgotten, while new measurements are continuously added to the current map. For the ODE computation, we simulate this behavior: Every time the robot moves, all footprints that extend beyond the RCM window are cropped accordingly. Once a footprint completely drops off the RCM, the corresponding time stamp is removed from the set of potential neighbors. We call this approach **RCM-ODE** as it tailored towards the RCM representation.

A. *Comparison of Accuracy Metrics*

In this experiment, we compare our three ODE versions to the Absolute Trajectory Error (ATE) and the Relative Trajectory Error (RTE) with $d = 1m$. For the ODE, we simulate a 360° range sensor with a maximum range of 2m and for the RCM-ODE we set the moving window size to $5m \times 5m$. To high-light the difference between the different metrics, we have chosen a short synthetic trajectory as shown in Fig. 3. We leave the ground truth poses perfect and add different types of artificial noise to the query trajectory.

In Fig. 3a, we add absolute Gaussian noise with a mean and standard deviation of $0 \pm 5cm$ linear and $0 \pm 0.1°$ angular. On this type of noise, all metrics behave very similar. For the ATE, absolute Gaussian noise fits the assumptions made by the least squares trajectory alignment [11]. Hence as an absolute measure, the ATE is lower on average ($6.1cm$) compared to the relative metrics like the RTE ($8.3cm$) or the ODE ($9.7cm$ for all versions).

In Fig. 3b, we add Gaussian drift noise; i.e. starting with the first pose, we construct the query trajectory by adding the relative pose from the ground truth plus a Gaussian noise with the following parameters, $1 \pm 1cm$ linear and $0.03 \pm 0.01°$ angular. For slowly drifting trajectories (such as this example), the ATE does not have a per pose interpretation as the trajectory alignment uses all poses in a joint least squares optimization [11]. Although the drift is relatively small, the ATE already shoots up to $32cm$ mean error and some poses even obtain an error of more than $70cm$. Regarding this type of noise, the RTE behaves very similar to the RCM-ODE (with a mean of $10.8cm$ for RTE and $10.2cm$ for ODE). In the error profile, we can even see very similar peaks (around pose 75 and pose 125 corresponding to the turns in the trajectory). If comparing the error profile of Offline-ODE and Online-ODE, we see additional peaks where the trajectory is intersecting itself. For the Online-ODE, we only see one peak around pose 220 (as it can only look into the past) and for the Offline-ODE two peaks (around 40 and

220). These peaks represent the error that is introduced due to imperfect loop-closure at the path intersection. Looking at the trajectory visualization in the bottom row, we can see that in the intersection close to $50cm$ error (i.e. relative displacement) can accumulate in the map representation. Note that this notion of map consistency is unique to the ODE metric.

In Fig. 3c, we add a discontinuity in the trajectory (i.e. the jump on the right at pose 100 with $40cm$ and $3°$) and a single outlier (i.e. at top of trajectory at pose 150 of $40cm$ and $3°$), together with a very low level of Gaussian drift noise ($0 \pm 0.5cm$ and $0 \pm 0.005°$ angular). Note that the ATE equally distributes the error, which makes the impact of discontinuity completely disappear. For the RTE, on the other hand, we can see that the error shoots up a few poses before the discontinuity and drops back directly after. For the one single outlier pose, we can observe two distinct peaks in the RTE error profile (one when the outlier ends up at end of the sub-trajectory and one when the outlier starts the sub-trajectory). In contrast, we can observe very clear peaks in the error profile of the ODE (exactly at the point where the error is happening). The magnitude of the peaks corresponds to their impact on the map consistency. Depending on the ODE version, there are one or two peaks at the self-intersection of the trajectory.

In Fig. 3d, we have the same discontinuity and outlier as in the previous trajectory, but add a higher level of drift noise (linear $1 \pm 1cm$ and angular $0.03 \pm 0.01°$). Note how the alignment of the ATE is very unstable since the error profile looks completely different from Fig. 3c. In contrast, RTE and ODE are both very stable in the error profile.

B. Real-world experiments

For our real-world experiments, we use the OpenLoris dataset [9]. This datasets has 22 sequences of a robot driving in 5 different in-door environments (including an university office, a cafe, a two-bed room apartment and a supermarket). The robot is equipped with multiple sensors including an Intel RealSense T265 tracking camera, an Intel D435i depth camera and a 2D-Lidar (either Hokuyo UTM-30LX or RoboSense RS-LiDAR-16 - depending on the sequence). For our experiments, we exclude the three market sequences as they do not contain any 2D-Lidar signal, which is required for some experiments. Additional to the official datasets and ground truth, the authors of [9] also kindly provided us with the trajectories of the best four approaches of the IROS 2019 Lifelong Robotic Vision SLAM Challenge [8]. (1) was submitted as "Wheel Odometer-Enhanced VINS with Map-Based Localization" by Segway Robotics, (2) as "Multi-Level Sparse Feature Optical Flow Tracking Based Visual-Inertial SLAM with Fast Relocalization" by Xie and Song, (3) as "Customized VINS-Mono with unsupervised-based deep loop closure" by Song and Wang and (4) as "Modified ORB-SLAM with learning-based and odometry-aided relocalization" by Wang et al. (more details on [8]). Additionally, we also denote the raw wheel odometry as the fifth approach (5).

Qualification Procedure. In our ODE evaluation, we use a qualification step to maintain the evaluation integrity (i.e. to avoid cases where no poses are reported or poses are randomly placed in space without any overlap). A trajectory is disqualified if either of the three following conditions is not met: (1) the RTE[m] with $d = 1m$ must not exceed $1m$ (this only happens if the poses are erratically jumping around), (2) the bounding box over all query poses has a similar size compared to the bounding box over all ground truth poses; i.e. the bounding box diagonal ratio must stay below a factor of three (this only occurs if an approach diverges and runs off into infinity), (3) the query trajectory must at least contain valid poses for 50% of the time stamps (this only occurs if an approach completely fails).

Result Reporting. In the following experiments, we report the performance of each approach on the Offline-ODE metric in an accumulative error histogram plot (similar to [13]). The plot is constructed in the following way. First, we compute the mean ODE over all cells in a query footprint $\mathcal{F}_q^{(i)}$ for all available time stamps $i \in T$. If a time stamp is missing in the query (i.e. there is no pose reported for longer than 200ms), then the corresponding ODE value is set to infinity (this is also the case for all time stamps of a disqualified trajectory). All ODE values are then collected in a histogram ranging from 0 to 1m in 200 steps (additionally we maintain one infinity bucket for all values larger than 1m). In the accumulative error histogram, we start with the bucket of the lowest error and subsequently add the normalized values of the higher buckets, which results in an monotonically increasing error curve. For any given error threshold (e.g. the 0.5m border between yellow and orange background in Fig.4a), the intersection between the accumulative curve and the threshold gives the percentage of time stamps that have an ODE below this threshold. For example, in Fig.4a the red curve has about 80% of the time stamps with an ODE below 0.5m. In this graph, *better* means reaching higher percent values at lower ODE thresholds.

Experiment 1: Frequency Analysis. In this experiment, we analyze the sensitivity of the evaluation on the time stamp frequency. For this purpose, we model the sensor footprint of the laser scan as a half circle with 15m range. As a base frequency, we use the original laser time stamps from the data bag, which means that we sample the evaluation time stamps at approximately 40Hz. Using this base frequency, we then sub-sample the time stamps for evaluation taking only every n^{th} element with $n = [1, 2, 4, 8]$ resulting in an evaluation frequency of $[40, 20, 10, 5]$Hz. We show the results in Fig.4a. In this experiment, there is barely any change between the different evaluation frequencies. Only for approach 3 and 4, we can see a minor optimism for the lowest frequency (darkest color). The reason for this optimism is that some outliers are already missed at this frequency. Overall, we can conclude that the sensitivity to the evaluation frequency is very low.

Experiment 2: Grid Cell Size Analysis. To analyze the sensitivity to the grid cell size, we run all five approaches with varying grid cell size (i.e. $[0.2, 0.4, 0.8, 1.6]$m). If look

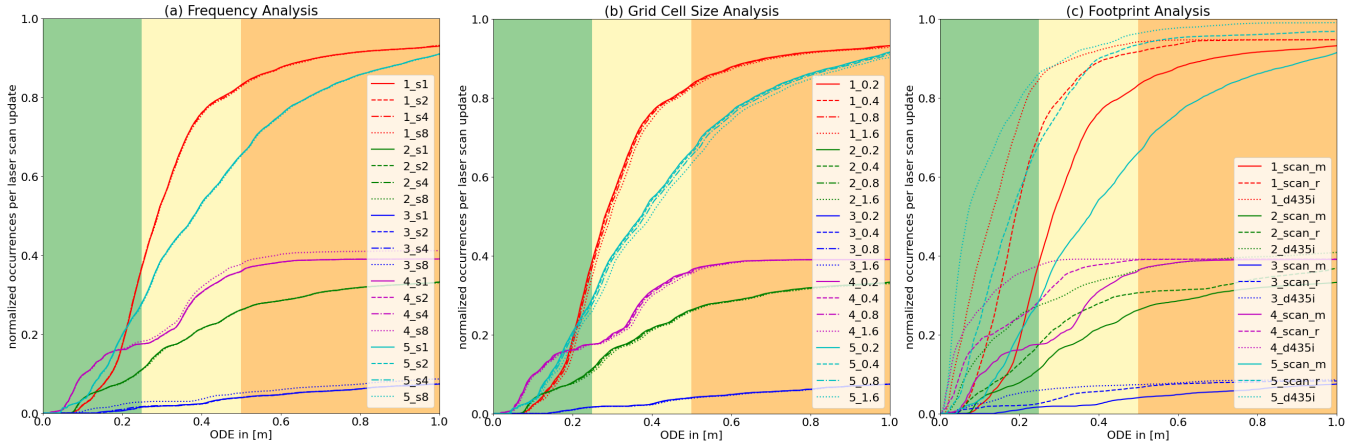

Fig. 4. Accumulative error histograms for the analysis on the sensitivity of the ODE metric to the (a) time stamp frequency, (b) the grid cell size and (c) the footprint model. Each approach (1-5) is plotted in its own color and each line style denotes a different parameter setting. For (a), finer dashes mean lower frequency, for (b), finer dashes mean larger grid cell size, and for (c) finer dashes mean a smaller footprint size. For (a) and (b), the footprint model was set to *scan_m*. Note that the ODE has a low sensitivity to the time stamp frequency and the grid cell size, which are hyper-parameters without real-world meaning, but is very sensitive to the model of the sensor footprint, which reflects the fact that mapping with long-range sensors requires a higher degree of self-consistency compared to short-range sensors.

at the impact of the grid cell size on the ODE result (Fig.4b), we can see that the relative performance gap between the different approaches stays fairly constant. The remaining difference stems from quantization effects; i.e. for larger grid cell sizes, it is more likely that two time stamps are considered neighbors. For larger grid cell sizes, this then leads to a slightly lower mean ODE. This effect, however, is very similar for each trajectory and does not significantly change relative performance between the different approaches, which speaks for a low sensitivity to this evaluation parameter.

Experiment 3: Footprint Analysis. In this experiment, we evaluate the impact of the footprint model on the evaluation outcome. We compare the simple laser scan model from the previous experiments (denoted as *scan_m*), a model where the footprint is created by ray casting using the real scan measurements (denoted as *scan_r*) and a simple cone shaped model for the Intel RealSense d435i depth camera with a maximum range of 2.5m (at which point the depth uncertainty exceeds 5cm) and a field of view of 69.4° (denoted as *d435i*). In Figure 5, we show two trajectories of Approach 1 evaluated with each of the three footprint models.

Looking at the results in Fig.4c, we can see that for all approaches the choice of the mapping sensor has a huge impact on the ODE metric. At this point, let us recall that the whole purpose of the ODE metric is to evaluate the impact of a localization error on the map consistency. Depending on the range of the sensor, this implicitly means that each mapping sensor comes with a different requirement for the localization consistency. The ODE visualizes this implicit requirements in the mapping space. Thus, a far range sensor (e.g. *scan_m*) requires a significantly higher consistency than a very short range sensor (*d435i*). Consequently, the choice of the localization algorithm has to be adapted to the used mapping sensors. E.g. if you only use the d435i for mapping, then even simple wheel odometry (Approach 5) could be sufficient for simple scenarios. However, when you have a far-range sensor (*scan_m*), the wheel odometry is not enough and more complex approaches (such as Approach 1) might be a better choice.

Summary. In our experiments, we have shown how the ODE metric can link a localization error to a potential map inconsistency/error, which allows us analyze the location and magnitude of the potential map inconsistency based on sensor frustum. In contrast to the ATE and RTE, the ODE computes peak values at the time when map inconsistencies are introduced due to a localization error. We note that this makes this a potential loss function for machine learning based approach (either for generating training samples or direct optimization). Calculating the ODE has a higher computational complexity than ATE/RTE. On real world experiments, we have shown that the ODE metric shows a low sensitivity to the choice of the evaluation frequency and also the grid cell size, which means that one can adjust these two parameters to accommodate the available compute budget. Finally, we have shown that not all sensors and mapping strategies have the same prerequisites for localization consistency, which reflects in very different error statistics for different types of sensors. This data suggests that there is not one single metric that can cover all SLAM applications, but that more consumer-driven localization metrics could help to understand the individual strengths and weaknesses of different SLAM approaches better.

VI. Conclusion

In this paper, we introduce a new way for evaluating robotic localization. In contrast to traditional evaluation strategies that treat localization as a stand-alone module, our evaluation technique considers localization as part of a bigger autonomous system which requires consistency in its own view of the world. In this work, we focused on the transformation of the localization error into the mapping space. While we think that more research in this direction is needed to fully understand the relationship between localization error and map consistency, this work already led to many

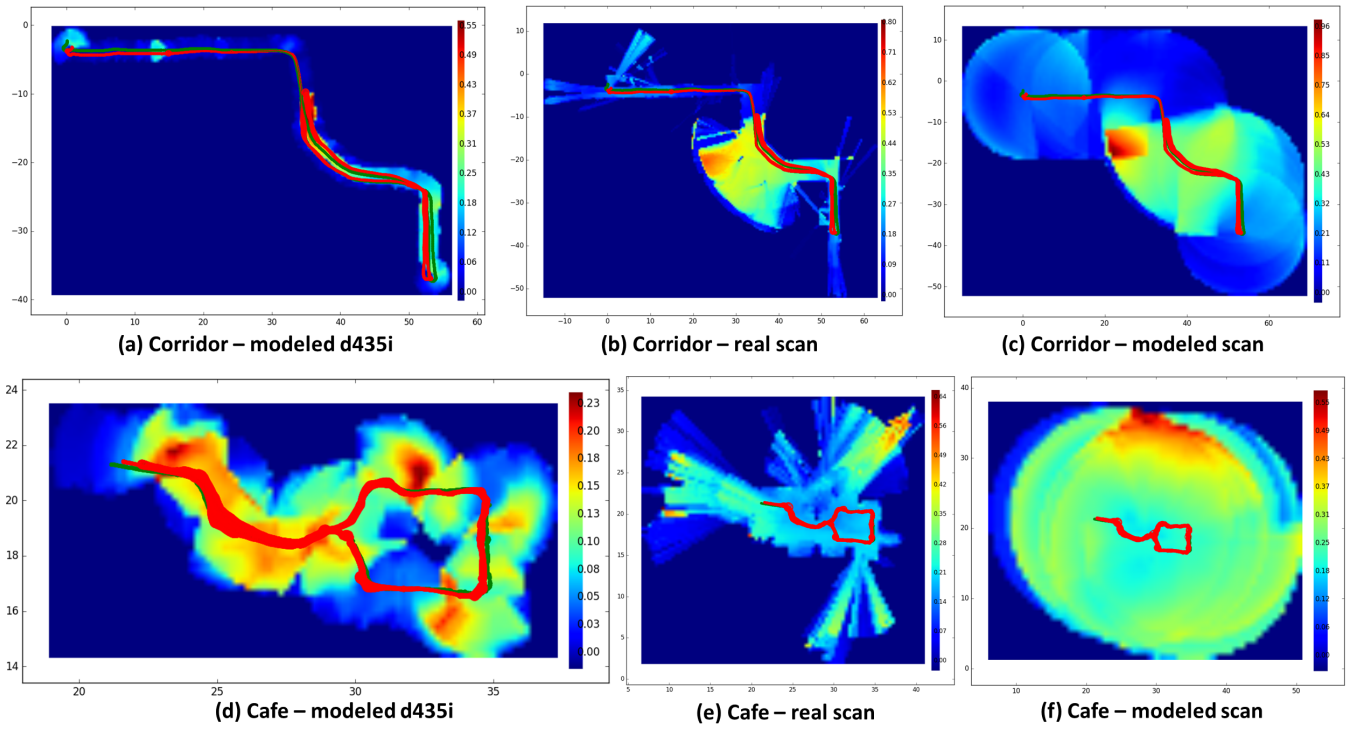

Fig. 5. Footprint Comparison. We show the trajectory of Approach 1 on two datasets (top: corridor1-5, bottom: cafe1-2), which is evaluated using three different footprint models. (a+d) use a model of the Intel RealSense d435i depth camera, (b+e) use the real lidar scan and (c+f) a model of the lidar footprint. The small green line shows the ground truth and the red line the query trajectory. The thickness of the red line corresponds to the ODE for this time stamp. The heat map in the background shows the average ODE per grid cell. This value can be interpreted as the potential map blurriness at this location only due to localization imperfection.

interesting observations and helped us to see localization in a new light. We see the main contribution of this paper as providing a tool to further and deepen the understanding of the interplay between localization and mapping and to highlight the need for more research in localization metrics such that localization algorithms can be better compared in a way that matters to the final application.

References

[1] Bruno Bodin, Harry Wagstaff, Sajad Saecdi, Luigi Nardi, Emanuele Vespa, John Mawer, Andy Nisbet, Mikel Luján, Steve Furber, Andrew J Davison, et al. Slambench2: Multi-objective head-to-head benchmarking for visual slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[2] Péter Fankhauser and Marco Hutter. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. In Anis Koubaa, editor, *Robot Operating System (ROS) The Complete Reference (Volume 1)*, chapter 5. Springer, 2016.

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[4] Dirk Hähnel, Sebastian Thrun, Ben Wegbreit, and Wolfram Burgard. Towards lazy data association in slam. In *Robotics Research. The Eleventh International Symposium*, pages 421–431. Springer, 2005.

[5] Mladen Mazuran, Gian Diego Tipaldi, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. A statistical measure for map consistency in slam. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 3650–3655. IEEE, 2014.

[6] ME Peters, RM Gates, and M Chertoff. Federal radionavigation plan. *Department of Transportation USA*, 2008.

[7] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. IEEE, 2018.

[8] Qi She, Xuesong Shi, Yimin Zhang, Fei Qiao, and Rosa Chan. IROS 2019 Lifelong Robotic Vision Challenge. https://lifelong-robotic-vision.github.io/competition. Accessed: 2020-07-13.

[9] Xuesong Shi, Dongjiang Li, Pengpeng Zhao, Qinbin Tian, Yuxin Tian, Qiwei Long, Chunhao Zhu, Jingwei Song, Fei Qiao, Le Song, et al. Are we ready for service robots? the openloris-scene datasets for lifelong slam. *arXiv preprint arXiv:1911.05603*, 2019.

[10] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.

[11] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):376–380, 1991.

[12] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems, 2015.

[13] Nan Yang, Rui Wang, Xiang Gao, and Daniel Cremers. Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*, 3(4):2878–2885, 2018.

[14] Yufeng Yue, Danwei Wang, PGCN Senarathne, and Chule Yang. Robust submap-based probabilistic inconsistency detection for multi-robot mapping. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2017.

[15] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[16] Yong Zhao, Shibiao Xu, Shuhui Bu, Hongkai Jiang, and Pengcheng Han. Gslam: A general slam framework and benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1110–1120, 2019.