2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Long-Term Map-Maintenance in Changing Environments using Ray-Bundle-Impact-Factor Estimation

Matthias Breitfuss*, Marcus Geimer*, Christoph Gruber

***Abstract*— Ensuring an accurate and robust localization is one of the most significant problems in the field of mobile robotics. In this context, map-based localization methods utilizing 3D LiDARS for environmental perception are widely used. Even if there exist multiple promising techniques in this field, the majority of approaches can only guarantee an accurate and robust operation if there is no deviation between the map and the real surroundings. Consequently, state of the art localization methods frequently suffer from unreliable results or even complete failure in the case of changing environments. In this paper, we propose an efficient technique for a precise and robust maintenance of localization maps through real-time incorporation of 3D LiDAR scans. Our map update procedure is based on a novel way of estimating the interference between laserbeams and map contents, denoted as Ray-Bundle-Impact-Factor (RBIF). Our technique additionally solves the widespread problem of disruptive hole creation caused by discretization effects. Experiments on real-world as well as synthetic data demonstrate the precision and stability of our method under various challenging conditions and evaluate our approach in comparison to multiple SOTA map maintenance algorithms.**

I. INTRODUCTION

One of the most fundamental capabilities in the field of unmanned ground vehicles (UGVs) is the ability to localize the robot accurately and reliably in its surroundings. A confident localization forms the basis for numerous functionalities executed by a mobile robot, such as collision-free navigation or precise interaction with objects in the surroundings. In the research field of robot localization, various methods have become established achieving accurate and reliable pose estimations. Depending on factors such as sensor availability or characteristics of the operational environment, SOTA approaches use different sensing technologies and algorithmic concepts to solve the localization problem. According to [1], one of the most widespread approaches in this context is the usage of map-based pose estimation incorporating perception systems such as cameras or LiDAR (Light Detection And Ranging).

Even if multiple promising approaches for map-based robot localization have been established, many of the proposed concepts are restricted to static and well-structured environments. As soon as the operational areas are non-static and change over time, these approaches show deficiencies when it comes to long-term localization robustness and accuracy. Depending on the degree of deviation between a changing environment and a static map, these deficiencies range from slight pose shifts compared with the true position of the UGV to ultimate failures of the localization process. Consequently, relying on static maps is fatal in such applications and thus a functionality to update maps according to the state changes of an environment is crucial for robust localization over long periods.

*The authors are with the Institute of Mobile Machines (Mobima), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

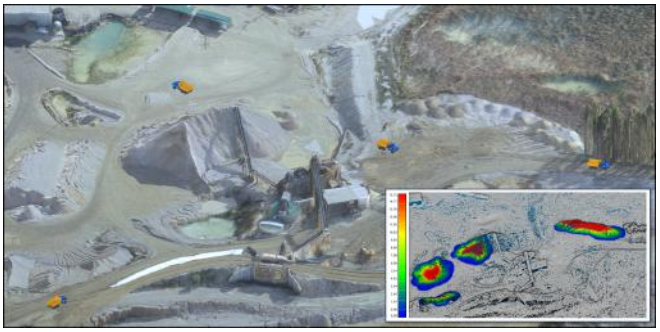


Fig. 1: Virtual environment including high dynamic obstacles (trucks) and low dynamic changes (gravel pile deformations) used during experiment C in table I

Our work contributes to the field of map-based robot localization by proposing a method that supports the integration of environmental changes perceived via 3D LiDARs into digital localization maps. Our emphasis is set on unstructured operational areas of mobile robots which undergo continuous modifications induced by high-dynamic objects (humans, other vehicles) as well as creeping alterations of the surroundings caused by low-dynamic modifications.

Taking these characteristics into account, environments that perfectly comply with the descriptions above are open pit mines or gravel quarries. These application fields contain fast-driving vehicles as well as slowly modified areas such as gravel piles that change shape and dimensions as further material is continuously added via a conveyor belt.

We propose a map maintenance method, which enables mobile robots to keep the deviation between a real operational environment and its digital counterpart at a minimum and consequently guarantee an up-to-date reference for map-based localization algorithms. The centerpiece of our work is the novel Ray-Bundle-Impact-Factor (RBIF), which represents a computationally efficient and accurate model to estimate the interference between a bundle of LiDAR rays and the content of a 3D map. We demonstrate the efficiency of our method and its positive effect on long-term localization robustness in real-world and virtual experiments.

The main contributions of our work are as follows:

- We propose a novel map maintenance method based on the Ray-Bundle-Impact-Factor (RBIF) enabling updates of discretized 3D maps at sub-voxel accuracy
- We prove the ability of our method to properly incorporate environmental changes of different dynamics
- We show that our approach solves the widespread issue of disruptive hole formations
- We evaluate our method in a benchmark with SOTA techniques using virtual and real-world datasets

II. RELATED WORK

In the field of mobile robotics, various types of three-dimensional maps have been established to digitally reconstruct the environment of an UGV. In the following, we give a brief overview of frequently used map representations in the world of UGVs and discuss their conceptual benefits and disadvantages.

The most basic way in which an operational area of a mobile robot can be represented is the usage of an unprocessed point cloud. According to [2], such maps are characterized by conceptual simplicity and have the advantage of capturing the geometry of the environment at a potentially high fidelity and density. Despite these benefits, the usage of unprocessed point cloud maps in robotic applications is often impracticable due to high memory consumption and computational effort. According to [3], these drawbacks prompt many researchers to use alternative map types.

A well-established alternative that overcomes the disadvantages of unbounded memory consumption is the usage of voxel-based maps. In this context, voxels can be described as cube-shaped elements that discretize space in equally dimensioned volume sectors. One of the predominant areas of application of voxel maps in robotics is the field of localization as shown by [4]. The authors massively reduce the amount of data contained in an environmental map by applying a voxelgrid-based filter on raw point cloud information. According to [4] this voxelization step enables for a high frequency of their robot localization process due to an extensive reduction of the processing effort. Besides the usage of voxel maps in robot localization, another area of application is the field of trajectory planning and collision avoidance ([5], [6]). As an example, [7] use a voxel-map of the environment of an underwater manipulator to determine occupied areas within the system's surroundings to plan collision-free movements of the manipulator.

Even if the usage of voxelmaps remarkably reduces memory consumption and computational complexity compared to maps consisting of raw pointclouds, they still suffer from some memory-related drawbacks. The size of voxel maps in memory mainly depends on two factors, the resolution of the voxels and the overall space the map encovers. As memory consumption increases cubically with the size of the grid, voxelmaps are not feasible for huge environments.

A well-established technique to mitigate these memory issues of voxelmaps is to store information in an octree-based data structure. Octrees, as described thoroughly by [8], depict a memory-efficient and processing-friendly way of storing three-dimensional information about an environment. [9] make use of these advantages to implement a framework for fine-grained mapping in the field of aerial vehicles. The authors use the octree-based map for localization of an autonomous flying device as well as for real-time path planning. [10] and [11] also build upon octrees to implement mapping and localization strategies incorporating environmental perception sensors such as 3D LiDAR and stereo vision.

The aforementioned excerpt of publications in the field of mapping, localization and collision-free trajectory planning and control illustrates the high potential and extensive usage of octrees in robotics. Especially when it comes to representing maps in changing environments, octrees show outstanding benefits. Despite promising results shown by various publications in the context of octree-based map maintenance, the approaches suffer from two major drawbacks.

On the one hand, approaches like the ones of [8] or [12] exclusively maintain the occupancy state of the fixed-size nodes in an octree without caring about the actual geometry of the measured objects. As a consequence, deriving a localization-ready map as a 3D point cloud from such an octree suffers from resolution-dependent deviations when compared to the actual geometry of the surroundings. On the other hand, as mentioned in [13] and [14], octree-based mapping techniques frequently face a discretization-induced issue which is known as *disruptive hole formation*. Due to conceptual deficiencies of the update processes, single nodes or even bigger parts of octrees are undesirably deleted in the course of an ongoing update procedure [12]. These areas of missing data are directly passed on to pointcloud-based localization maps which potentially cause inaccuracies or even failure of the pose estimation process.

We contribute to the field of octree-based map maintenance techniques with a novel method, that enables for the modification of map elements on a sub-voxel scale despite the usage of a discretized data structure. Our approach makes use of the advantageous property octrees concerning low memory usage and fast accessing of map elements while overcoming disadvantages of the discretization by executing map updates at sub-voxel accuracy.

III. METHOD OVERVIEW

In the following section, we go into detail concerning the proposed maintenance method with a primary focus on our novel Ray-Bundle-Impact-Factor (RBIF). At first, we give a general overview of the algorithmic substeps of our map update approach and illustrate the way the module is coupled with a high-level localization system. After that, we describe the data structures we utilize to represent the map and LiDAR scans. Next, we illustrate the Ray-Interference-Classification process, which allocates voxels of the map to one of three interference categories using a technique known as raytracing. This step is followed by the calculation of our novel RBIF which estimates the degree of interference between the rays of a LiDAR scan and the state of map voxels using Kullback-Leibler divergence. Based on the

magnitude of the RBIF, we finally update affected voxels by adjusting their occupancy probability and further node-inherent information.

A. System architecture

Figure 2 illustrates the system architecture behind our proposed approach for continuous map maintenance coupled with a map-based localization algorithm. The latter attempts to solve the problem of estimating the 6DoF pose of an UGV with respect to its surroundings using a state-of-the-art pointcloud registration algorithm [15], which requires three inputs. These are z representing a 3D LiDAR scan, $\hat{x}$ which denotes a prior pose estimation of the UGV as well as an estimated map of the environment $\hat{m}$. Concerning $\hat{x}$ and $\hat{m}$, the localization module is initialized with some starting values, where $\hat{m}_0$ is created during a preceding map-building process using a state-of-the-art algorithm [16].

Assuming there is no map maintenance component available, the localization algorithm would have to permanently rely on the initial map $\hat{m}_0$ for solving the pose estimation problem. As demonstrated in section V-A, the localization error e_x degrades as soon as the environment undergoes changes that lead to an increasing deviation e_m between the static map $\hat{m}_0$ and the real world w. To counteract this problem we propose the integration of a map maintenance algorithm paired with a direct feedback of the map estimation $\hat{m}$ into the localization module. This coupling heavily improves the robustness and accuracy of the pose estimation and thus supports long-term localization in massively changing environments.

B. Data structures

One essential input of the map maintenance module in figure 2 is $\hat{z}$, which depicts map-aligned LiDAR scans after successful pointcloud registrations. For reasons of efficiency and real-time capability we convert the aligned LiDAR scans $\hat{z}$ into a set of normal distributions using a restricted voxel-grid filter. The filter condenses scan points sharing a common voxel into a representative multivariate normal distribution $\mathcal{N}_R(\mu_R, \Sigma_R)$ consisting of a mean vector $\mu_R \in \mathbb{R}^3$ and a covariance matrix $\Sigma_R \in \mathbb{R}^{3\times3}$. Henceforward, we denote $\mathcal{N}_R$ as ray bundle distribution which can be interpreted as a geometric approximation of the object situated in a voxel.

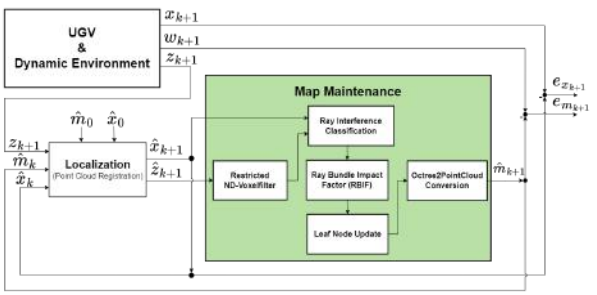


Fig. 2: System architecture of our proposed localization and mapping approach including the interference classifier, RBIF estimation and the sigma point adjustment.

As the coefficients in Σ are remarkably influenced by the number of points being integrated into the matrix, comparisons between different normal distributions are not meaningful. To overcome this issue we normalize Σ to guarantee independence of the size of the original point set by making use of the filter's property, that the maximum distance between any of the original points in a voxel and their corresponding mean is restricted by half of the voxel's space diagonal. This restriction inherently limits the variance of the points to a maximum of $(\sqrt{3}l_V)^2$, with l_V being the predefined resolution of the grid filter.

Besides the aforementioned structure of the input scan, the map $\hat{m}$ builds upon an organized octree, which is a tree-like structure representing a recursive partitioning of space into successively shrinking cubic voxels. The innermost voxels of the octree, the so-called leaf nodes, have a predefined resolution that matches the one of the voxelgrid filter used to calculate $\mathcal{N}_R$ As a result, both data structures are aligned with each other which supports the fusion of information between $\hat{z}$ and $\hat{m}$. Analogous to the preprocessed input scan, the octree-based map uses multivariate normal distributions at leaf node level to approximate terrain and object geometries within occupied voxels ($\mathcal{N}_V$). Coefficients for the mean μ_V and the covariance matrix Σ_V of each distribution in $\mathcal{N}_V$ are initially derived from $\hat{m}_0$ and continuously updated by integrating new information from $\mathcal{N}_R$ using our proposed fusion method described in III-E. Besides the normal distribution, each leaf node additionally includes a variable representing the occupancy probability of the voxel, where values below 0.5 depict unoccupied voxels while values above this threshold illustrate occupied cells.

In the following, we denote the set of ray bundles as $\mathscr{R} = \{\mathcal{N}_{R,i}(\mu_{R,i}, \Sigma_{R,i}) \quad | \quad i \in 1,...,N\}$ and the normal distributions within the occupied voxels of the map as $\mathscr{V} = \{\mathcal{N}_{V,j}(\mu_{V,j}, \Sigma_{V,j}) \quad | \quad j \in 1,...,M\}$, with N and M being the number of distributions in each set. From a geometric point of view, the distributions in $\mathscr{R}$ and $\mathscr{M}$ can be represented by confidence ellipsoids as the ones shown in figure 3. Throughout our work, ellipsoids visualize the three-dimensional confidence interval (CI) for a certainty of 95% with respect to the underlying distribution.

C. Ray Bundle Interference Classification

After calculating $\mathscr{R}$ from $\hat{z}$, the next step in our proposed update method is the ray bundle interference classification which builds upon the process of raytracing as illustrated in figure 3. Under the assumption that a LiDAR ray spreads on a straight line, we identify all leaf nodes in $\hat{m}$ that are traversed by the ray connecting the LiDAR origin and the mean of a ray bundle.

Subsequently, each of the voxels identified to be traversed or finally hit by a ray bundle is assigned to one of three classes. Traversed leaf nodes that are not occupied by any object according to the map (grey voxels in figure 3) are not further considered during the update procedure of the current iteration. This is legitimate, as the unhindered traversal of the

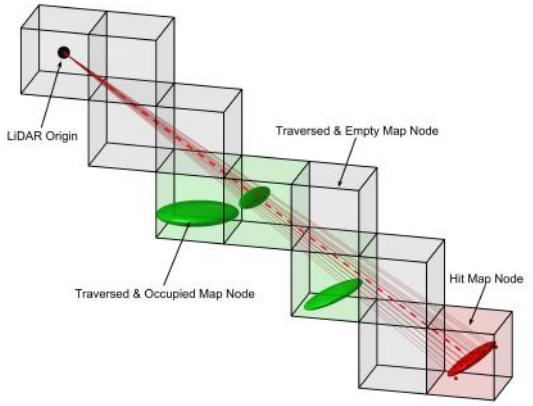


Fig. 3: The raytracing-based classification process assigns traversed leaf nodes to one out of three categories: *traversed & empty*, *traversed & occupied* or *hit*

voxel does implicitly confirm the status of the voxel being empty.

In contrast to that, when LiDAR rays hit an object (red voxel in figure 3), we need to take these new measurements into account and adjust the underlying voxel state. We are referring to this case as an *additive interference* between the LiDAR ray bundle and the leaf node.

The third class describes the situation of an unobstructedly traversed leaf node that is marked as occupied according to the map (green voxels in figure 3). This type of ray bundle interference is the most challenging case, as the unhindered node traversal can be justified by two different effects. On the one hand, a ray bundle can indeed traverse an occupied leaf node without any interference with the object, whenever the latter occupies the voxel only partly and the ray bundle traverses the volume element in the remaining section. On the other hand, though, the object in the leaf node might also have changed its position within the borders of the voxel or have fully disappeared since the last update. In this case, the traversal of the ray bundle contradicts the current occupancy status of the voxel, as the unhindered traversal argues for an empty leaf node. We denote this type of relationship as *subtractive interference*.

D. Calculation of Ray-Bundle-Impact-Factor

After determining the affiliation of each traversed leaf node to one of the aforementioned classes, we estimate the degree of interference between the ray bundles and the affected map voxels. As given by equation 1, the expansion of a ray bundle is modeled as a linear scaling of the normal distribution based on the normalized position t along the ray between the LiDAR origin $o \in \mathbb{R}^3$ and the ray bundle mean.

$$\begin{aligned} \mathcal{N}_R(t) &= \mathcal{N}_R(\mu_R(t), \Sigma_R(t)) \\ &= \mathcal{N}_R(o + t(\mu_R - o), t \cdot \Sigma_R), \text{with } 0 < t \leq 1 \end{aligned} \tag{1}$$

Using this parametric form, we search for the value of t that corresponds to the point of maximum interference between $\mathcal{N}_R(t)$ and a considered map node distribution $\mathcal{N}_V$. To determine this specific value we formulate an optimization problem based on the Kullback-Leibler Divergence (KLD) [17] by incorporating the parametric normal distribution depicted in equation 1.

$$\begin{aligned} &D_{KL}(\mathcal{N}_R(t) || \mathcal{N}_V) = \\ &\frac{1}{2}\Big(tr\left(\Sigma_V^{-1}\Sigma_R(t)\right) - k \\ &+ (\mu_V - \mu_R(t))^T \Sigma_V^{-1} (\mu_V - \mu_R(t)) + \ln\left(\frac{\det(\Sigma_V)}{\det(\Sigma_R(t))}\right)\Big) \end{aligned} \tag{2}$$

where $k = 3$ is the dimensionality of the problem.

By deriving $D_{KL}(\mathcal{N}_R(t) || \mathcal{N}_V)$ with respect to the aforementioned scaling factor t, we find a closed form solution that enables to precisely locate the point of minimum KLD and thus the maximum ray bundle impact (eq. 3).

$$t_{D_{KL,min}} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{3}$$

$$\begin{aligned} a &= 2 \cdot (o - \mu_R)^T \Sigma_V^{-1} (o - \mu_R) \\ b &= tr\left(\cdot \Sigma_V^{-1}\Sigma_R\right) + (\mu_V - o)^T \Sigma_V^{-1} (o - \mu_R) \\ &\quad + (o - \mu_R)^T \Sigma_V^{-1} (\mu_V - o) \\ c &= -k \end{aligned} \tag{4}$$

Using $t_{D_{KL,min}}$ determined by equation 3, we can describe the position and stamping of the ray bundle distribution that has the highest similarity to the distribution within the voxel. As shown by the yellow confidence ellipsoids in figure 4, equation 3 results in meaningful estimations of the maximum impact positions in consideration of our approach to model the beam expansion as a scaled multivariate normal distribution.

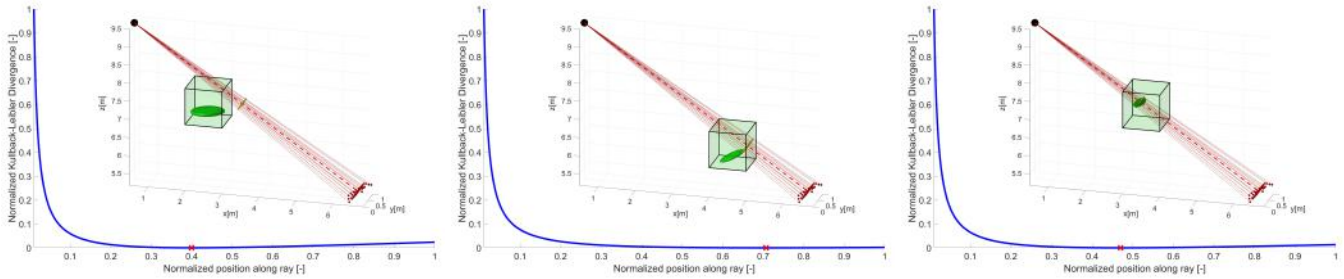
After calculating the maximum impact constellation of a specific interference situation, we evaluate the proposed Ray-Bundle-Impact-Factor using the following case distinction.

$$RBIF = \begin{cases} 0 & \text{, if } d_M(x_{max}; \mathcal{N}_V)^2 > \chi^2_\alpha \\ 1 & \text{, if } d_M(\mu_V; \mathcal{N}_{R,t_{D_{KL,min}}})^2 \leq \chi^2_\alpha \\ \frac{||\mu_V - x_{max}||}{2\sqrt{\lambda_{V,max}}} & \text{, otherwise} \end{cases} \tag{5}$$

$$\begin{aligned} &d_M(x; \mathcal{N}(\mu, \Sigma))^2 = (x - \mu)^T \Sigma^{-1} (x - \mu) \\ &x_{max} = \max_x \left(\log f(x; \mathcal{N}_{R,t_{D_{KL,min}}}) + \log f(x; \mathcal{N}_V)\right) \end{aligned} \tag{6}$$

$d_M(x; \mathcal{N}(\mu, \Sigma))^2$ is the squared mahalanobis distance of a point x with respect to a multivariate normal distribution. χ^2_α is the critical value from the chi-square distribution table for a significance level α, which we set to 95%. x_{max} represents the point that maximizes the joint likelihood of the ray bundle distribution at the maximum impact position and the voxel distribution. $\lambda_{V,max}$ depicts the largest eigenvalue of the covariance matrix Σ_V.

As illustrated in figure 4 a) and c), there exist two edge cases that allow immediately setting the RBIF without the need for further computations. These exceptional situations are the interference-free traversal of an occupied voxel and the case of maximum ray bundle interference.



(a) (b) (c)

Fig. 4: Normalized Kullback-Leibler Divergence for different interference situations. Positions of maximum ray bundle impact (red crosses/yellow ellipsoids) are reliably detected for the cases of a) no interference, b) partial interference and c) maximum interference.

Concerning case a), we define a ray bundle traversal to be interference-less whenever the point maximizing the joint likelihood is located outside of the 95% confidence interval of the voxel distribution.

In contrast to the aforementioned interference-less case, we immediately set the RBIF to its maximum value of 1, whenever the mean of the voxel distribution is located within the 95% confidence interval of the ray bundle distribution. This corresponds to situation c) where the ray bundle undeniably penetrates the object distribution in the voxel, which highly contradicts the current belief that the voxel is occupied.

The third case listed in equation 5 covers the situation of partial overlaps between the distributions, which we evaluate with respect to the probabilistic interference by the denoted approximation. We assume that the distance between the point of maximum joint likelihood and the voxel mean is indirectly proportional to the impact the ray bundle penetration has on the occupancy state of the voxel. In this context, the maximum distance possible is half the length of the major axis of the confidence ellipsoid. We utilize this limit not only to normalize the impact factor to the desired range of $[0,1]$, but also to make the RBIF independent of the shape of the covariance matrix.

E. Final leaf node update

In the final step of the update, we use the information gathered during the previous calculations of interference class and RBIF to adjust the voxel state with respect to occupancy probability and normal distribution parameterization.

Regarding the case of *subtractive interference*, the RBIF derived from equation 5 inherently represents the necessity to change the state of a node from occupied to empty. In this context we consciously skip each node that has a RBIF equal to 0, as the traversal of the ray bundle does not carry any kind of evidence concerning a potential change of the node state. The contrary is on hand, when the RBIF is 1 and thus supports the assumption that the object in the leaf node has disappeared. In this specific case, we reduce the occupancy probability of the node by a factor of $p_{sub} = 0.2$, which corresponds to a quite aggressive update. In other words, the occupancy probability of a node falls below the threshold of 0.5 after just a few measurements causing an almost instantaneous deletion of the node. For any other value of the RBIF, we use the proposed interference description as a scaling factor for p_{sub}. High RBIFs lead to faster deletions of nodes, while RBIFs close to 0 require multiple confirmations that the leaf node indeed is empty before deleting the voxel from the tree. In the obverse case of an *additive interference*, we simply increase the occupancy by multiplying with a constant factor of $p_{add} = 0.9$ regardless of the RBIF, assuming that the LiDAR measurements are highly accurate.

Besides the update of the occupancy, we additionally modify the coefficients of the normal distributions in an occupied voxel to consider new measurements. Depending on whether an update is caused by an *additive* or *subtractive interference*, we adjust the normal distribution in the leaf node to either represent an approximated union ($\mathcal{N}_V \cup \mathcal{N}_R$) or difference ($\mathcal{N}_V \setminus \mathcal{N}_R$) between the distributions.

IV. EXPERIMENTS

The following section illustrates the results of a benchmark between the map maintenance methods Octomap [8] and Ufomap [12] and our proposed approach. As listed in table I we selected three scenarios that come along with different challenges for the map maintenance algorithms using real-world as well as simulated datasets.

TABLE I: Map Maintenance Scenarios

Exp.	Dataset	Initial Map	Changes	Robot State
A	real-world	outdated	no (static)	driving
B	real-world	up-to-date	high dynamic	standstill
C	simulation	up-to-date	low dynamic	driving

In case of the real-world experiments data is recorded using a mobile platform carrying an Ouster OS1 LiDAR sensor generating pointclouds of dimension 1024x64 at a rate of 10 Hz and a maximum range of 200 m. For the virtual experiment, a simulated LiDAR with identical characteristics is utilized. We run the experiments on an Intel i7-1360P@2.2 GHz with 32 GB RAM.

Experiment A focuses on the ability of the algorithms to minimize the deviation between an outdated initial map and

a current map of an environment while driving through a static real-world scene. In comparison, experiments B and C are characterized by the presence of dynamic changes in the environment. In experiment B the robot is at a standstill while fast objects (cyclist and vehicle) move through the lidar's field of view (figure 7). This scenario demands the algorithms to quickly react to non-persistent changes and recover to the initial state of the map after the disappearance of the objects. Experiment C represents the most difficult case as the algorithms have to adapt to a combination of slow but large-scale changes (i.e. shape change of huge gravel piles) as well as high dynamic objects while maneuvering through a virtual environment (figure 1).

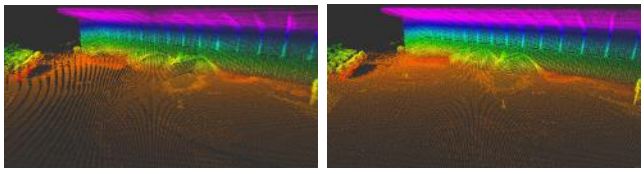
V. Evaluation of Map Adaptability

In this section, we investigate the capability of different map update algorithms to properly correct deviations between an environment and its digital counterpart. As mentioned in the previous section deviations arise from outdated initial maps (Exp. A), highly dynamic objects (Exp. B) and slow changes of the environments (Exp. C). In terms of fairness, we feed recorded lidar scans of each experiment one by one into the algorithms regardless of the different method's processing times. This way each algorithm gets the opportunity to integrate the same information into their maps.

For evaluation purposes, we assess the degree of compliance between the maps maintained by the algorithms and corresponding ground truth maps using the well-known comparison metrics *Surface Coverage* (COV), *Reconstruction Accuracy* (ACC) and *Average Hausdorff Distance* (AHD) [18]. We take the center point of each occupied voxel as reference when calculating the AHD for octomap and ufomap compared to the mean of the normal distribution in the case of our proposed method. Even though this difference seems to skew the results, it emphasizes the distinctive advantage of our approach to describing the geometry of the environment at a sub-voxel accuracy rather than being restricted to discrete volume elements.

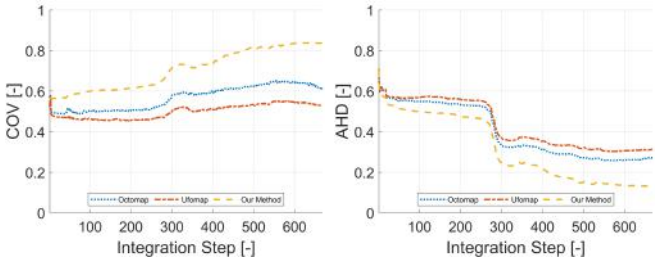
Exp. A - Correction of initial map deviation

Figure 6 illustrates the results of experiment A in terms of the selected metrics COV and AHD. As each of the benchmarked methods is initialized with the same outdated map, the COV (figure 6a) shows the identically low degree of surface coverage at the beginning of the experiment. As



(a) Ufomap (b) Our Approach

Fig. 5: Disruptive hole formations in case of Ufomap (a) vs. dense mapping using our approach (b)



(a) (b)

Fig. 6: COV (a) and AHD (b) for benchmarked update methods during experiment A

lidar scans are consecutively integrated by the algorithms, the deviation compared to the ground truth map is reduced resulting in an improvement of the COV.

The degree of this expected improvement differs between the benchmarked methods. While the surface coverage in case of our proposed method rises continuously, Octomap and Ufomap both show an almost stagnating trend of the COV. This can be explained by the critical discretization effect mentioned in section I that leads to a massive creation of disruptive holes in the map (figure 5). As our method is capable of avoiding such erroneous deletions of voxels the COV reaches an impressive peak of 83.5% at the end of the experiment (figure 6a).

We additionally outperform the other algorithms in the benchmark when it comes to maintaining a high degree of geometric compliance within the occupied voxels measured by the AHD. By fusing lidar measurements within a voxel into a multivariate normal distribution our algorithm approximates the ground truth geometry of the environment much better than the other methods that rely on discrete volume elements. This advantage is clearly illustrated by the increasing gap between the AHD in figure 6b.

About coupling the map update methods with a real-time localization process as described in section III-A, another relevant aspect alongside the map quality is the processing time required for LiDAR scan integration. Low processing times enable to integrate new environmental information at high frequencies and thus react to changes in the surroundings almost instantaneously.

As listed in table II, Octomap performs best in a map resolution range from 1.0 m down to 0.4 m. For these coarses resolutions, our approach is slower than Octomap

TABLE II: Processing times [s] at different map resolutions for an integration distance of 200 m during experiment A

Res. [m]	Octomap	UFOMap	Ours
1.0	0.016 ± 0.001	0.137 ± 0.003	0.031 ± 0.004
0.8	0.021 ± 0.002	0.169 ± 0.006	0.045 ± 0.004
0.6	0.034 ± 0.003	0.228 ± 0.008	0.068 ± 0.006
0.4	0.074 ± 0.008	0.384 ± 0.014	0.128 ± 0.013
0.2	0.387 ± 0.032	1.361 ± 0.032	0.338 ± 0.026
0.1	1.561 ± 0.051	5.336 ± 0.341	0.784 ± 0.038

although the difference narrows the finer the resolution gets. Considering a LiDAR scan rate of 10 Hz, our approach and Octomap are capable of integrating new scans in real-time down to a map resolution of 0.6 m and 0.4 m respectively, whereas Ufomap is far behind. It is also worth mentioning, that our method shows the highest efficiency at resolutions below 0.4 m, as it takes more and more advantage of the benefits induced by the data condensation process described in section III-B.

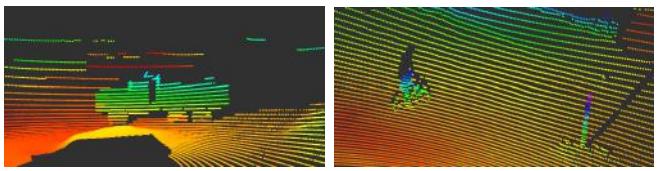
Exp. B - Responsiveness to highly dynamic obstacles

In addition to the ability of map maintenance algorithms to adapt to large-scale environmental changes, it is crucial that the methods properly react to local appearances of high dynamic obstacles. In the course of the present experiment, a human, a cyclist and a vehicle maneuver through the environment (figure 7). As these objects should induce an immediate integration in the maintained maps, the ACC is expected to drop abruptly as the moving objects cause a deviation from the obstacle-free ground truth map.

Figure 8 confirms this expectation as all three methods show prompt reactions to the appearances of the objects. The diagram illustrates a similar responsiveness of all algorithms when it comes to adding the object into the map. For instance, the sudden appearance of the vehicle (integration step 300 in figure 8) causes a massive decrease in the ACC. Each algorithm is capable of recovering from this drop after the vehicle is out of the LiDAR's field of view. Despite these positive similarities, our approach is the only one able to almost perfectly reconstruct the ground truth map in the long-term, as we recover to an ACC of almost 100% after the disappearance of all dynamic objects. The significantly worse trends for Ufomap and Octomap are again explainable by the disruptive hole problem.

Exp. C - Robustness to multi-dynamic environment changes

Analogous to experiment A, COV and AHD are used to determine the quality of the maintained maps in comparison to the ground truth map under the influence of low but large-scale changes and highly dynamic obstacles. When investigating the results in figure 9, it is conspicuous that none of the algorithms is capable of perfectly reconstructing the altering environment. This is understandable though, as some changes of the environment are never perceived by the lidars due to occlusion or sensor restrictions (height, max



(a) Vehicle at iteration 400 (b) Cyclist at iteration 670

Fig. 7: Exemplary scans of vehicle (a) and cyclist (b) appearing as high dynamic objects in experiment B.

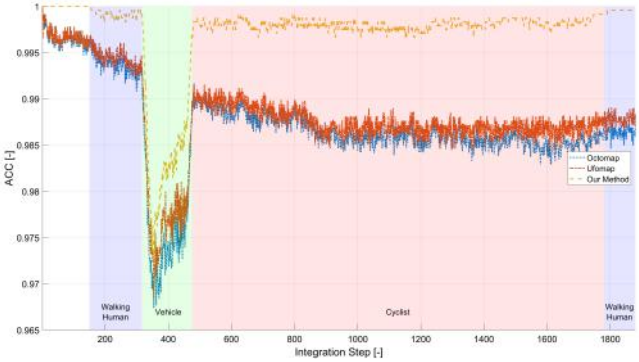


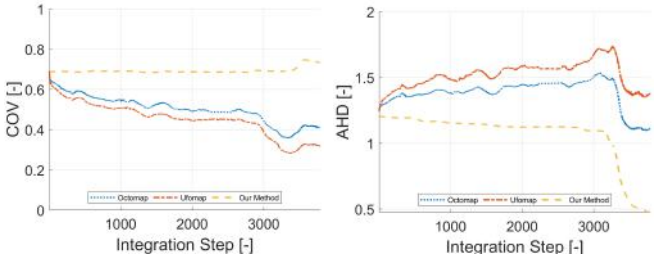
Fig. 8: ACC for benchmarked update methods during experiment B including dynamic objects (colored sections).

range o.s.). Consequently, a certain amount of environmental changes cannot be integrated in the maps resulting in persistent and uncorrectable deviations.

Nevertheless, according to the metrics, our method still shows the best compliance among the tested algorithms. Especially the AHD demonstrates once more the advantage of the proposed approach in retaining a high congruence compared to the ground truth map. We reuse the data from experiment C, but this time the position of the UGV and thus the knowledge of the perfect scan alignment is not given a priori. Rather, it is the task of the localization module to find the ideal transformation for a scan and pass this data to the map maintenance algorithms. Vice versa, the update methods feedback improved versions of the environmental map to provide an up-to-date reference of the surroundings for pose estimation. To investigate the influence of this coupling, we evaluate the accuracy of the pose estimations during the experiment using the *Absolute Trajectory Error* (ATE) [19] between the estimated trajectory of the robot and the ground truth. It is to be expected that the deviation from the ground truth stays low as long as the update methods properly incorporate the environmental changes.

A. Evaluation of Localization Accuracy

This last section focuses on the usability of the algorithms in the context of real-time map localization. Due to our method guaranteeing the highest degree of congruence between map and environment over time, it represents the



(a) (b)

Fig. 9: COV (a) and AHD (b) for benchmarked update methods during experiment C

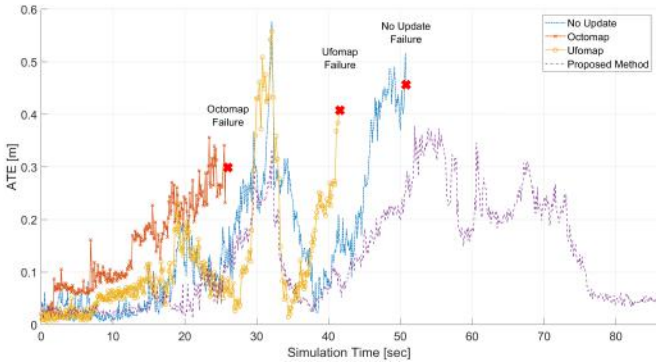


Fig. 10: Localization accuracy under increasing changes of the operational environment using different update methods.

best method among the benchmarked algorithms concerning long-term navigation map maintenance.

As depicted by figure 10, the localization module can properly find solutions for the pose estimation problem when continuously substituting the navigation map with updated versions from our algorithm. The contrary is the case when investigating the ATE for Octomap and Ufomap. As these methods cannot integrate the progressive modification of the environment properly into the maps, the localization process is fed with increasingly worse input data. Consequently, the pose estimations degrade which again has an impact on map maintenance as the to-be-integrated scans suffer from bad map alignment. Figure 10 illustrates this effect in terms of successive increases concerning the difference in ATE between the methods. Especially the ultimate loss of localization in case of Ufomap and Octomap compared to the permanently stable pose estimation when using our approach underlines the outstanding performance of our method.

VI. CONCLUSION AND FUTURE WORK

We presented a novel method for updating three-dimensional maps of an environment for the specific case of dynamically changing surroundings. Our method builds upon the estimation of the degree of interference between a bundle of rays originating from a 3D LiDAR and the information stored in a digital map. The core of this estimation procedure is our proposed Ray-Bundle-Impact-Factor which essentially evaluates the ray-map interference by minimizing a parametric form of the Kullback-Leibler-Divergence. Based on the interference estimate we update the map by either adding new voxels to the map, deleting superfluous nodes in case they are not occupied anymore or adjusting the geometric information stored in the map cells.

To confirm the usability and quality of our proposed algorithm in the field of long-term map maintenance, we evaluated our method on real-world as well as virtual data sets. We compared the performance of our approach with two SOTA mapping techniques showing, that our proposed approach outperforms the other algorithms in terms of maintaining high map compliance with the real surroundings and stabilizing the pose estimation process when coupled with a localization system.

REFERENCES

[1] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review," *Applied Sciences*, vol. 13, no. 17, p. 9877, 2023.

[2] B. Fei, W. Yang, W. Chen, Z. Li, Y. Li, T. Ma, X. Hu, and L. Ma, "Comprehensive review of deep learning-based 3d point cloud completion processing and analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22 862–22 883, 2022.

[3] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, 2018.

[4] T. Suzuki, M. Kitamura, Y. Amano, and T. Hashizume, "6-dof localization for a mobile robot using outdoor 3d voxel maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5737–5743.

[5] M. Pantic, I. Meijer, R. Girod, N. Alatur, O. Andersson, C. Cadena, R. Siegwart, and L. Ott, "Obstacle avoidance using raycasting and riemannian motion policies at khz rates for mavs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1666–1672.

[6] I. Babataev, A. Fedoseev, N. Weerakkodi, E. Nazarova, and D. Tsetserukou, "Hyperguider: Virtual reality framework for interactive path planning of quadruped robot in cluttered and multi-terrain environments," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 2037–2042.

[7] M. Rossi, S. Sivcev, P. Trslic, G. Dooly, E. Omerdic, and D. Toal, "Voxel map based collision detection for underwater manipulators," in *OCEANS 2018 MTS/IEEE Charleston*, 2018, pp. 1–6.

[8] K. M. Wurm, H. Armin, B. Maren, S. Cyrill, and B. Wolfram, "Octomap: A probabilistic, flexible and compact 3d map representation for robotic systems," *Proc. of the ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[9] E. Vespa, N. Nikolay Nikolov, M. Marius Grimm, L. Luigi Nardi, P. H. Paul H J Kelly, and S. Stefan Leutenegger, "Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1144–1151, 2018.

[10] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Mapping the static parts of dynamic scenes from 3d lidar point clouds exploiting ground segmentation," in *European Conference on Mobile Robots (ECMR)*, 2021, pp. 1–6.

[11] A. Pfrunder, P. V. K. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3d lidar," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[12] D. Duberg and P. Jensfelt, "Ufomap: An efficient probabilistic 3d mapping framework that embraces the unknown," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6411–6418, 2020.

[13] K. Schauwecker and A. Zell, "Robust and efficient volumetric occupancy mapping with an application to stereo vision," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6102–6107.

[14] V. Reijgwart, C. Cadena, R. Siegwart, and L. Ott, "Efficient volumetric mapping of multi-scale environments using wavelet-based compression," *Robotics: Science and Systems*, 2023.

[15] K. Koide, "small_gicp: Efficient and parallel algorithms for point cloud registration," *Journal of Open Source Software*, vol. 9, no. 100, p. 6948, 2024.

[16] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

[17] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[18] S. Aravecchia, M. Clausel, and C. Pradalier, "Comparing metrics for evaluating 3d map quality in natural environments," *Robotics and Autonomous Systems*, vol. 173, p. 104617, 2024.

[19] 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.