Authors Copy. This paper has been accepted for publication in RSS 2025.

Certifiably-Correct Mapping for Safe Navigation Despite Odometry Drift

Devansh R. Agrawal, Taekyung Kim, Rajiv Govindjee, Trushant Adeshara,
Jiangbo Yu, Anurekha Ravikumar, and Dimitra Panagou
University of Michigan, Ann Arbor
Correspondence: devansh@umich.edu

***Abstract*—Accurate perception, state estimation and mapping are essential for safe robotic navigation as planners and controllers rely on these components for safety-critical decisions. However, existing mapping approaches often assume perfect pose estimates, an unrealistic assumption that can lead to incorrect obstacle maps and therefore collisions. This paper introduces a framework for certifiably-correct mapping that ensures that the obstacle map correctly classifies obstacle-free regions despite the odometry drift in vision-based localization systems (VIO/SLAM). By deflating the safe region based on the incremental odometry error at each timestep, we ensure that the map remains accurate and reliable locally around the robot, even as the overall odometry error with respect to the inertial frame grows unbounded.**

Our contributions include two approaches to modify popular obstacle mapping paradigms, (I) Safe Flight Corridors, and (II) Signed Distance Fields. We formally prove the correctness of both methods, and describe how they integrate with existing planning and control modules. Simulations using the Replica dataset highlight the efficacy of our methods compared to state-of-the-art techniques. Real-world experiments with a robotic rover show that, while baseline methods result in collisions with previously mapped obstacles, the proposed framework enables the rover to safely stop before potential collisions.

Code[1] and Video[2]

I. Introduction

Accurate state estimation and mapping are essential for safe robotic navigation, as planners and controllers rely on perception outputs to ensure the safety of planned trajectories or control actions. Various methods have been developed to certify that controllers meet predefined safety specifications [1, 2], and when real-time obstacle detection is necessary, it is often intuitive to handle safety constraints in the planner [3, 4, 5]. These methods typically assume perfect perception, a simplification that can lead to safety violations.

A perception module provides a pose estimates and constructs maps of the obstacle geometry, and can take a variety of formats, such as Euclidean Signed Distance Fields (ESDFs) [6, 7], polytopic Safe Flight Corridors (SFCs) [8], occupancy log-odds [9], or NERFs [10]. Although recent advances have achieved significant accuracy improvements [11, 12, 13, 14, 15], formal error analysis is often lacking. Without quantified error bounds, guaranteeing the safety of a closed-loop robotic system remains a challenge.

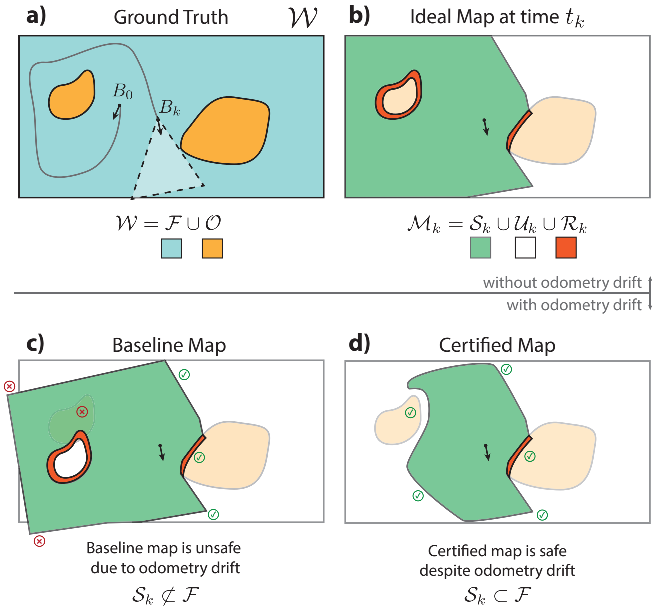
a)
Ground Truth
$\mathcal{W}$
B_0
B_k
$\mathcal{W} = \mathcal{F} \cup \mathcal{O}$
b)
Ideal Map at time t_k
$\mathcal{M}_k = \mathcal{S}_k \cup \mathcal{U}_k \cup \mathcal{R}_k$
without odometry drift ↑
with odometry drift ↓
c)
Baseline Map
Baseline map is unsafe due to odometry drift
$\mathcal{S}_k \not\subset \mathcal{F}$
d)
Certified Map
Certified map is safe despite odometry drift
$\mathcal{S}_k \subset \mathcal{F}$

Fig. 1. Overview of notation and objectives. (a) depicts the operating environment, where the world $\mathcal{W}$ is the union of the free space $\mathcal{F}$ and the obstacles $\mathcal{O}$. The robot does not know $\mathcal{F}$ or $\mathcal{O}$. It starts at B_0, and follows the gray trajectory to B_k building the map as it goes. (b) depicts the ideal mapping output, where at the k-th timestep, the map $\mathcal{M}_k$ is composed of the known safe region $\mathcal{S}_k$, the unknown space $\mathcal{U}_k$ and the known obstacle set $\mathcal{R}_k$. (c) depicts the map produced by current state-of-the-art methods, where due to odometry drift the map is erroneous: notice that the safe region (according to the constructed map) is not a subset of the free space, $\mathcal{S}_k \not\subset \mathcal{F}$. (d) depicts the desired behavior of the certified maps, where although the safe region is smaller, it is certifiably-correct: we can prove that $\mathcal{S}_k \subset \mathcal{F}$.

This paper introduces a framework for "certifiably correct mapping" ensuring that obstacle-free regions of a map remain correct despite odometry drift. The challenge is illustrated in Figure 1. Consider an environment $\mathcal{W} = \mathcal{F} \cup \mathcal{O}$, representing free and obstacle spaces, respectively (Figure 1a). As a robot navigates, at the k-th time step it has created a map $\mathcal{M}_k$, comprising the supposedly safe space $\mathcal{S}_k$, the unknown space $\mathcal{U}_k$ and the recognized obstacles $\mathcal{R}_k$ (Figure 1b). However, due to odometry drift, maps can misclassify obstacles as free space, leading to potential safety violations as indicated in Figure 1c. We address this by deflating safe regions in order to ensure $\mathcal{S}_k \subset \mathcal{F}$ at all times (Figure 1d).

Our main contributions are as follows:

[1] Code: https://github.com/dasc-lab/certifiably-correct-mapping

[2] Video: https://youtu.be/qMlDK7Iou48

- The theoretical framework to construct and deflate the free space in obstacle maps to ensure their correctness despite odometry drift. Assuming the odometry algorithm reports the pose and the covariance of the incremental transform, we propose deflating the supposedly safe region ($\mathcal{S}_{k+1}$ is deflated relative to $\mathcal{S}_k$) to ensure that it remains a subset of the free region $\mathcal{F}$.
- We prove the correctness and applicability of this framework on two popular and state-of-the-art mapping frameworks: the polytopic SFCs of [8] and the ESDFs of [7].
- Beyond providing the theoretical analysis and proofs of correctness, we validate and compare our approach with state-of-the-art baseline methods through extensive simulations on the Replica dataset [16].
- Finally, we demonstrate the approach in a real-world experiment on a robotic rover. A human teleoperates the rover using only the First Person View (FPV) feed and the obstacle map constructed and streamed to the operator in real-time. The rover uses an onboard safety filter to prevent collisions. Unlike baseline methods which result in collisions, our approach prevents crashes by deflating the safe regions appropriately.

It is critical that we deflate $\mathcal{S}_k$ rather than inflate known obstacles $\mathcal{R}_k$. If the obstacles are inflated based on the accumulated odometry error, these obstacles can only grow in size, and might eventually occupy the entire domain $\mathcal{W}$. Instead, by deflating a safe region $\mathcal{S}_k$, the region that is certifiably safe shrinks, eventually becomes an empty set, and is removed from memory (i.e., the region becomes part of $\mathcal{U}_k$). When the region is observed by a sensor again, it can again be added to $\mathcal{S}_k$ again. Computationally, this reduces memory requirements, and mathematically this allows us to treat deflated obstacles as unknown regions and plan paths accordingly. The certified maps can be used together with the uncertified maps for practical applications: the uncertified maps can be used to plan trajectories for example for exploration or for navigating towards a goal location, while the certified map can be used for local obstacle avoidance.

Our paper is organized as follows. After a brief literature review in Section II, in Section III we provide a mathematical background and setup the problem formally. In Section IV and V we introduce the deflation mechanism for both map representations. In Section VI we propose methods to use the certified maps to acheive safe navigation. Finally in Section VII and Section VIII we present the simulation and experimental results.

II. Literature Review

Perception methods have seen significant advancements over the past few decades, driven by improvements in algorithms, sensors, and computational capabilities [17, 18]. The primary goals of these advancements have been to enhance localization and mapping accuracy, improve robustness under diverse environmental conditions, and develop algorithms with lower computational costs. For instance, modern Simultaneous Localization and Mapping (SLAM) systems now report translation error rates below 1% [19, 20], enabling more reliable navigation in real-world scenarios.

With these improvements, robots have been deployed in increasingly complex environments, relying heavily on Visual Inertial Odometry (VIO)/SLAM pose estimates and obstacle maps to navigate safely. As exemplified by the DARPA SubT Challenge, teams have developed perception systems capable of navigating subterranean environments [21, 22, 23]. In these systems, raw measurements are typically processed by a frontend into a more compact representation, while a backend uses nonlinear optimization methods to compute the robot's trajectory and map estimate [21]. Most of these optimization methods are based on factor graphs, which, although effective, become computationally expensive as the map size increases.

A common approach to manage this computational complexity is to use local submaps, connected through a graph of traversable regions or submap connections [21]. These methods reduce odometry drift by optimizing each submap within its own coordinate frame. When a robot revisits a previously mapped region, the submap can be reused, provided that the robot is correctly localized within it. However, even within a submap, odometry drift can still lead to localization errors. Therefore, ensuring safety requires addressing the potential errors within these submaps. The approach proposed in this paper aims to ensure correctness at the submap level, i.e., in the presence of incremental localization errors.

Recent work has explored techniques for ensuring the correctness of perception systems. For example, [24] achieve global optimization in pose graph optimization problems through a convex reformulation, while [25] provide error-bounded localization within 2D convex environments. Additionally, [26, 27] propose certifiably correct point-cloud registration and visual odometry methods. Similarly, [28] showed that bounded attitude errors lead to bounded position errors. In contrast to [27], this paper assumes that the incremental pose estimate is bounded in a Lie-algebraic sense, which allows our methods to be applied to a broader range of odometry algorithms, extending the applicability beyond the methods considered in [27]. In cases where certification of correctness is not feasible, estimating or quantifying the error can still provide valuable insights, for example using the methods in [29, 30] which estimate the error in point-cloud matching.

Other approaches have been proposed to address mapping consistency in the presence of odometry drift. [31] utilize overlapping Truncated Signed Distance Field (TSDF) voxels, which are only fused once the consistency of certain regions has been verified. These ideas share similarities with the work of [32, 33], which also emphasize the importance of ensuring consistency before merging obstacle estimates from different times. These methods propose constructing a manifold map, only merging them when correctness can be guaranteed. In contrast, the method proposed in this paper introduces a different strategy: regions where correctness cannot be assured are ”forgotten,” ensuring that only reliable, consistent parts of the map are used for navigation and decision-making.

III. Preliminaries and Problem Statement

Notation

$\mathbb{N} = \{0, 1, 2, ...\}$ is the set of natural numbers. $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$ denote reals, non-negative reals, and positive reals. $I_n \in \mathbb{R}^{n \times n}$ is the $n \times n$ identity matrix. The subscript is dropped when clear from context. $\mathbb{SO}(n)$ is the n-d special orthogonal group. $\mathbb{SE}(n)$ is the n-d special Euclidean group. $\mathbb{S}_+^n$ is the set of symmetric positive-definite matrices in $\mathbb{R}^{n \times n}$. The matrix square root of positive definite matrix $A \in \mathbb{S}_+^n$ is the matrix $A^{1/2} \in \mathbb{R}^{n \times n}$ such that $A^{1/2} A^{1/2} = A$. For $v \in \mathbb{R}^n$, $\|v\|$ denotes the 2-norm, $\|v\|_p$, ($p \in [1, \infty]$) denotes the p-norm, and $\|v\|_P = \sqrt{v^T P v}$ for $P \in \mathbb{S}_+^n$. All eigenvectors are assumed to be unit-norm. $\lambda(A)$ is the set of eigenvalues of $A \in \mathbb{R}^{n \times n}$, and $\lambda_{\max}(A)$ is the largest eigenvalue of $A \in \mathbb{S}_+^n$. $[p]_\times \in \mathbb{R}^{3 \times 3}$ is the skew-symmetric matrix such that $a \times b = [a]_\times b$ for any $a, b \in \mathbb{R}^3$.

Matrix Lie Groups

Here we present a brief review of Matrix Lie Groups in the context of this paper, with additional equations and details in Appendix A. We refer the reader to the excellent references [34, 35, 36] for a more complete description.

The Lie group $\mathbb{SO}(3)$ defines 3D rotations, and the group $\mathbb{SE}(3)$ defines 3D rigid transformations. Both $\mathbb{SO}(3)$ and $\mathbb{SE}(3)$ are Matrix Lie groups, i.e., group elements are matrices, and composition operator is the standard matrix multiplication operator. In $\mathbb{SE}(3)$ the group action $\cdot : \mathbb{SE}(3) \times \mathbb{R}^3 \to \mathbb{R}^3$ transforms a point p from its representation in frame A to that in frame B. Given $T_A^B = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in \mathbb{SE}(3)$,

$$p|^B = T_A^B \cdot p|^A = Rp|^A + t. \tag{1}$$

The Lie algebra of a group is a vector space of all possible directions a group element can be perturbed locally. The Lie algebras of $\mathbb{SO}(3)$ and $\mathbb{SE}(3)$ are $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ respectively. These vector spaces are isomorphic to $\mathbb{R}^3$ and $\mathbb{R}^6$ respectively. The $\wedge$ operator converts the Euclidean vector to an element of the Lie Algebra, and $\vee$ does the inverse.

Consider a Lie group $\mathbb{G}$ with an associated Lie algebra $\mathfrak{g}$ that is isomorphic to the Euclidean vector space $\mathbb{R}^n$. Given an element $x \in \mathfrak{g}$, we can convert it to the corresponding group element using the exponential map, $\exp : \mathfrak{g} \to \mathbb{G}$. For convenience, we also define the Exp map, which maps from the Euclidean vector space to the group directly, $\operatorname{Exp} : \mathbb{R}^n \to \mathbb{G}$, $\operatorname{Exp}(\xi) = \exp(\xi^\wedge)$. For certain groups including $\mathbb{SE}(3)$, these operations have analytic expressions [34, Appendix].

Uncertain Poses and Transforms

An uncertain pose or transform $T_A^B \in \mathbb{SE}(3)$ is denoted

$$T_A^B \sim \mathcal{N}(\widehat{T}_A^B, \Sigma_T),$$

where $\widehat{T}_A^B \in \mathbb{SE}(3)$ is the mean estimate, and $\Sigma_T \in \mathbb{S}_+^6$ is a covariance matrix. This indicates T_A^B is the transform

$$T_A^B = \widehat{T}_A^B \operatorname{Exp} \tau, \tag{2}$$

where $\tau \in \mathbb{R}^6$ is a random sample drawn from $\tau \sim \mathcal{N}(0, \Sigma_T)$.

Recall the group action $p|^B = T_A^B \cdot p|^A$. If the transform T_A^B is uncertain, $p|^B$ follows a distribution and, to first order, is a normal distribution [34, 36]:

$$p|^B = \left(T_A^B \cdot p|^A\right) \sim \mathcal{N}(\hat{p}|^B, \Sigma_p) \tag{3}$$

where the mean and covariance are

$$\hat{p}|^B = \widehat{T}_A^B \cdot p|^A \in \mathbb{R}^3, \quad \Sigma_p = J \Sigma_T J^T \in \mathbb{S}_+^3$$

with $J = \begin{bmatrix} R & -R[p|^A]_\times \end{bmatrix} \in \mathbb{R}^{3 \times 6}$.

For the remainder of the paper, we truncate the distribution making the following assumption:

Assumption 1. *Let $T_A^B \sim \mathcal{N}(\widehat{T}_A^B, \Sigma)$, where $\widehat{T}_A^B = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$. Then for any $p|^A \in \mathbb{R}^3$, the point $p|^B \in \mathbb{R}^3$ satisfies*

$$p|^B = T_A^B \cdot p|^A \in \mathcal{E} \tag{4}$$

where $\mathcal{E} \subset \mathbb{R}^3$ is the ellipsoid

$$\mathcal{E} = \left\{ p \in \mathbb{R}^3 : \left\| \Sigma_p^{-1/2} \left(p - \widehat{T}_A^B \cdot p|^A \right) \right\| \leq 1 \right\}, \tag{5}$$

$$\Sigma_p = \kappa J \Sigma J^T \in \mathbb{S}_+^3, \quad J = \begin{bmatrix} R & -R[p|^A]_\times \end{bmatrix} \in \mathbb{R}^{3 \times 6}.$$

for some $\kappa > 0$.[3]

In other words, the assumption is that when a point p is transformed from its representation in frame A to that in frame B, the point $p|^B$ is contained within an ellipsoid $\mathcal{E}$ centered on the estimated point $\widehat{T}_A^B \cdot p|^A$, as defined in (5). The size and principal axes of the ellipsoid are defined by the estimated transform $\widehat{T}_A^B$ and the covariance matrix Σ. This allows us to bound the error of mapping points between frames, and the bound can be made tighter if κ is increased, or if higher order approximations are used, as in [36]. The higher order approximations yield tighter covariance ellipsoids, at the expense of increased computation. Since we focus on rototranslations between successive body frames, the transforms T_A^B should be close to identity where first order approximations work well.

Reference Frames

This paper uses the inertial frame I, a mapping frame M, and the body-fixed frame at the k-th timestep, B_k. Usually, M and I are equivalent, and M is defined such that at $M = B_0$. However, since we are considering odometry drift, M can drift relative to I. We assume that I is the true inertial frame (in which the obstacles are static), and M is the reference frame used to construct the state estimate and the map.

Problem Statement

Let $\mathcal{O}$ represents the obstacle geometry in a static environment $\mathcal{W} \subset \mathbb{R}^3$. Both $\mathcal{O}$ and $\mathcal{F} = \mathcal{W} \backslash \mathcal{O}$ are assumed initially

[3] κ chooses the probability the bound contains the point. For a d-dimensional normal distribution, $x \sim \mathcal{N}(\mu, \Sigma)$, the probability that $\left\|(\kappa\Sigma)^{-1/2}(x - \mu)\right\| \leq 1$ is $p \in [0, 1]$ such that $\kappa = \chi_d^2(p)$, where χ_d^2 is the quantile function of the chi-squared distribution with d degrees of freedom. For 3D points, $\kappa = 2$ corresponds to $p = 97\%$.

unknown. We assume $\mathcal{F}$ does not contain any isolated points, and that $\mathcal{O}$ is closed. As with points, a set can be represented in a frame, i.e., we say that $\mathcal{O}|^{B_k} \subset \mathbb{R}^3$ is the set of all obstacle points represented in frame B_k.

To avoid obstacles, we must build a map of the environment. At the k-th timestep the map is $\mathcal{M}_k$, consisting of the (claimed) free-space $\mathcal{S}_k$, the unknown space $\mathcal{U}_k$, and the (claimed) obstacle space $\mathcal{R}_k$. A map is correct if the claimed free space is a subset of the true free space.[4] More formally,

Definition 1. *A map $\mathcal{M} = \mathcal{S} \cup \mathcal{U} \cup \mathcal{R}$ is the union of the (claimed) safe region $\mathcal{S}$, the unknown region $\mathcal{U}$, and the (claimed) obstacle region $\mathcal{R}$. At the k-th timestep, the map $\mathcal{M}_k$ is* correct *if for all $p|^{B_k} \in \mathbb{R}^3$,*

$$p|^{B_k} \in \mathcal{S}_k|^{B_k} \implies p|^{B_k} \in \mathcal{F}|^{B_k}. \tag{6}$$

In words, $\mathcal{M}_k$ is *correct* if $\mathcal{S}_k$ is a subset of the free space $\mathcal{F}$ *when represented in the k-th body-fixed frame*.

The definition above is intentionally explicit about which reference frame various points and sets are represented in since this is the source of the main problem tackled in this paper. Due to the odometry drift, there are two types of error common in state-of-the-art mapping algorithms:

(A) Errors in constructing the map: In current state-of-the-art implementations, the map is often represented computationally in the mapping frame M. Suppose at some time t_k the robot detects an obstacle (relative to its body-fixed camera) at a position $p|^{B_k}$. It will update the map to remove this point from the claimed free space:

$$\mathcal{S}_{k+1}|^M \subset \mathcal{S}_k|^M \backslash \{\widehat{T}^M_{B_k} \cdot p|^{B_k}\}. \tag{7}$$

However, notice that since the estimated transform $\widehat{T}^M_{B_k}$ is used instead of the true transform $T^M_{B_k}$, the location marked as an obstacle can be wrong. This problem is exacerbated since usually the line connecting the camera origin and the point $p|^{B_k}$ is marked free, and therefore the wrong locations are marked as part of $\mathcal{S}_{k+1}$.

(B) Errors in querying the map: Now suppose the robot wants to navigate the environment. It must therefore (at time t_k) check whether a point $p|^{B_k}$ relative to the body-fixed frame is free. To the best of our knowledge, all implementations will then check whether the corresponding estimated point in the map, $\hat{p}|^M$, is a free point, that is, they check whether

$$\hat{p}|^M = \widehat{T}^M_{B_k} \cdot p|^{B_k} \in \mathcal{S}_k|^M. \tag{8}$$

However notice again, since the estimated transform is used, this can lead to inconsistencies. In particular, owing to the odometry drift, the inconsistency will be worse when the obstacle point was inserted into the map many frames ago.[5]

We overcome both such issues, *by ensuring the map is always correct in the body-fixed frame.* An equivalent perspective is that despite using the estimated transform $\widehat{T}^M_{B_k}$ the map will be constructed and queried correctly.

The problem statement therefore is as follows:

Problem 1. *Consider a robotic system equipped with an RGBD camera operating in a static environment with obstacles $\mathcal{O} \subset \mathbb{R}^3$. Suppose an odometry module provides at each frame k the estimated odometry $\widehat{T}^{B_0}_{B_k} \in \mathbb{SE}(3)$, the relative odometry $\widehat{T}^{B_k}_{B_{k+1}} \in \mathbb{SE}(3)$ and a covariance of the relative odometry $\Sigma^{B_k}_{B_{k+1}} \in \mathbb{S}^6_+$. Suppose a mapping module can construct the best estimate map of the free space in the environment. Design a framework to correct the best-estimate map such that at each timestep, the map $\mathcal{M}_k$ is correct according to Definition 1* despite the odometry drift.

We also assume that if an obstacle point is within the camera's Field of View (FoV), it will be detected as an obstacle. This is a common implicit assumption in the mapping literature. Infrared depth cameras often fail to detect transparent obstacles (e.g., windows and glass doors) or obstacles with minimal texture (where the stereo block-matching algorithm fails). Such issues are beyond the scope of this paper.

In the next two sections, we demonstrate how to construct correct maps by modifying existing baseline mapping algorithms. In particular we extend (A) a mapping algorithm [8] which uses polytopes to represent the map of free space, and (B) the mapping algorithm [7] which uses signed distance fields to represent the free space. See Figure 2.

IV. Approach 1: Certified Safe Flight Corridors

A. Background

In the first approach, the obstacle-free region $\mathcal{S}_k$ at frame k is the union of n polytopes,[6]

$$\mathcal{S}_k|^{B_k} = \bigcup_{l=1}^{n} \mathcal{P}^l_k \tag{9}$$

where each polytope is a compact set of the form

$$\mathcal{P}^l_k = \{p \in \mathbb{R}^3 : A^l_k p \leq b^l_k\}. \tag{10}$$

This is often called the H-representation, since the polytope is defined by a set of half-space constraints [37]. An example of a polytope extracted from a depth image is shown in Figure 2c.

As the robot transitions from frame B_k to frame B_{k+1}, we can map each polytope from the previous frame to the new frame, and maintain the polytopes in the robot's body frame.

In the absence of odometry drift, one can directly compute the new polytopes:

$$\mathcal{P}^l_{k+1} = \{p \in \mathbb{R}^3 : A^l_{k+1} p \leq b^l_{k+1}\}, \tag{11a}$$
$$A^l_{k+1} = A^l_k R^T, \tag{11b}$$
$$b^l_{k+1} = b^l_k + A^l_k R^T t, \tag{11c}$$

[4] Since $\mathcal{O}$ is closed, $\mathcal{F}$ is open. The (claimed) safe region $\mathcal{S}$ can be either an open or closed subset of $\mathcal{F}$. Below, $\mathcal{S}$ will be a closed set.

[5] It will also becomes clear that time is not the only factor - points inserted/queried further from the robot will also be more inaccurate due to the larger moment arm that amplifies rotation errors. This is also why common heuristic algorithms of time- or distance-based forgetting cannot guarantee the correctness of the map. The methods proposed in this paper will directly address such issues.

[6] n can be different at each k.

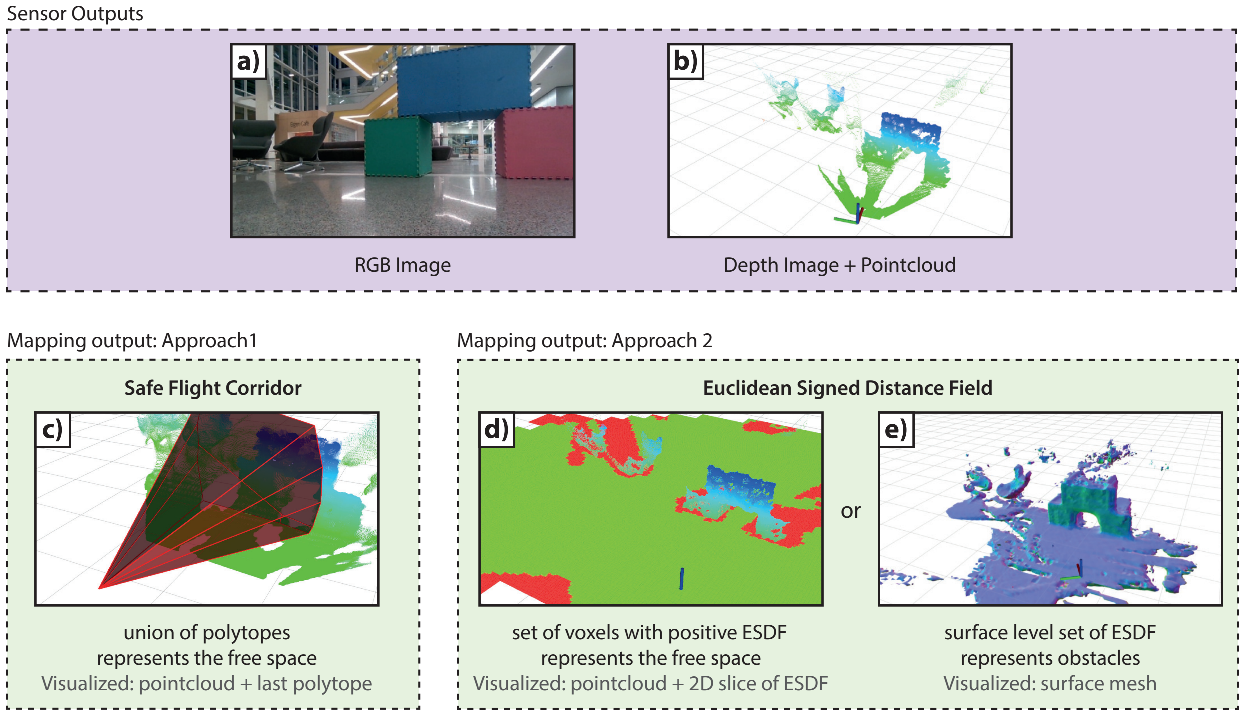
Sensor Outputs
a)
b)
RGB Image
Depth Image + Pointcloud
Mapping output: Approach1
Safe Flight Corridor
c)
union of polytopes
represents the free space
Visualized: pointcloud + last polytope
Mapping output: Approach 2
Euclidean Signed Distance Field
d)
or
e)
set of voxels with positive ESDF
represents the free space
Visualized: pointcloud + 2D slice of ESDF
surface level set of ESDF
represents obstacles
Visualized: surface mesh

Fig. 2. Two approaches to constructing an obstacle map. (Top row) An RGBD camera provides (a) the first person RGB image, and (b) the depth image/pointcloud constructed from stereo images. (Bottom row) The SFC approach represents the free space as a union of polytopes, one of which is depicted in (c). The ESDF approach represents the world using voxels, where each voxel stores the signed distance to the nearest obstacle. From this, both the (d) ESDF at specific voxels or (e) obstacle surface locations can be extracted and used for safe navigation. To aid the reader, in (c) and (d) the raw pointcloud is also visualized, and in (d) the colorscheme is such that voxels are marked green if $d > 0$, and red otherwise. This makes the map look binary, although it contains continuous values. Furthermore, note both methods operate in 3D - the 2D slice is used for visualization.

using the estimated transforms

$$\widehat{T}_{B_k}^{B_{k+1}} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}.$$

In the presence of odometry drift, however, the estimated transform $\widehat{T}_{B_k}^{B_{k+1}}$ is inexact, and this method fails to guarantee $\mathcal{P}_{k+1}^{l} \in \mathcal{F}$. Therefore, $\mathcal{M}_k$ is not guaranteed to be correct.

B. Proposed Approach

In the presence of odometry drift, since the transform $T_{B_k}^{B_{k+1}}$ is uncertain, the method in (11) does not work. Extending this approach to uncertain transforms is also not straightforward, since in the H-representation, an uncertain perturbation to a half-space does not result in a new half-space. Here, we propose a novel method that uses the V-representation of the polytope to circumvent this issue. In the V-representation, the polytope is the convex-hull of a set of vertices. Denote the set of vertices by

$$\mathcal{V}_i = \{v_{i,j}\}_{j=1}^{m_i} \subset \mathbb{R}^3, \tag{12}$$

where $v_{i,j} \in \mathbb{R}^3$ is the j-th vertex on the i-th face of a polytope.

We will use the V-representation to compute a new (deflated) polytope $\mathcal{P}_{k+1}$ from $\mathcal{P}_k$. The algorithm is described by the next Lemma and Theorem.

Lemma 1. *Suppose* $T_{B_k}^{B_{k+1}} \sim \mathcal{N}(\widehat{T}_{B_k}^{B_{k+1}}, \Sigma_k)$, *where*

$$\widehat{T}_{B_k}^{B_{k+1}} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}. \tag{13}$$

Consider a polytope $\mathcal{P}_k$ *that is obstacle free,*

$$\mathcal{P}_k = \{p \in \mathbb{R}^3 : A_k p \leq b_k\} \tag{14}$$

where $A_k \in \mathbb{R}^{N \times 3}$, $b_k \in \mathbb{R}^N$. *Denote the* i*-th row as* $a_{k,i} \in \mathbb{R}^3$. *For each vertex* $v_{i,j} \in \mathcal{V}_i(\mathcal{P}_k)$ *on the* i*-th face of the polytope, define*

$$J_{i,j} = \begin{bmatrix} R & -R[v_{i,j}]_\times \end{bmatrix}, \quad \Sigma_{i,j} = \kappa J_{i,j} \Sigma_k J_{i,j}^T, \tag{15}$$

as in Assumption 1. Let each element of $\rho \in \mathbb{R}^N$ *be*

$$\rho_i = \max_{j \in \{1,\dots,m_i\}} \sqrt{a_{k,i}^T \Sigma_{i,j} a_{k,i}} \tag{16}$$

Define a new polytope as

$$\mathcal{P}_{k+1} = \{p \in \mathbb{R}^3 : A_{k+1} p \leq b_{k+1}\}, \tag{17a}$$

$$A_{k+1} = A_k R^T, \tag{17b}$$

$$b_{k+1} = b_k + A_k R^T t - \rho. \tag{17c}$$

Given Assumption 1, $\mathcal{P}_k \in \mathcal{F}|^{B_k} \implies \mathcal{P}_{k+1} \in \mathcal{F}|^{B_{k+1}}$, *i.e., if* $\mathcal{P}_k$ *is obstacle-free, so is* $\mathcal{P}_{k+1}$.

Proof Sketch: [See Appendix B for the full proof.] It suffices to show that any obstacle potentially on the boundary of $\mathcal{P}_k$ will not be in $\mathcal{P}_{k+1}$ after the rigid transform. To do so, we consider a potential obstacle on the i-th face of the polytope, and compute the ellipsoid the obstacle could be in after the transform. We compute the tangent plane of the ellipsoid normal to the i-th hyperplane, and compute the minimum shift necessary such that the shifted hyperplane does not contain the ellipsoid. We use the convexity of the polytope to show that the necessary shift on the i-th hyperplane is ρ_i, the maximum of the shifts necessary at each of the vertices on the i-th hyperplane of the polytope. This deflaion, when applied to each hyerplane of the polytope, guarantees that $\mathcal{P}_{k+1}$ does not contain the obstacle points. ■

Finally, we can construct the main theorem.

Theorem 1. *Suppose the transform between frame is $T_{B_k}^{B_{k+1}} \sim \mathcal{N}(\widehat{T}_{B_k}^{B_{k+1}}, \Sigma_k)$. Given the k-th map is defined as in* (9)*, define the $(k+1)$-th map as*

$$\mathcal{S}_{k+1}|^{B_{k+1}} = \bigcup_{l=1}^{N} \mathcal{P}_{k+1}^{l} \tag{18}$$

where each polytope is defined using Lemma 1. Then, given Assumption 1,

$$\mathcal{S}_k \subset \mathcal{F} \implies \mathcal{S}_{k+1} \subset \mathcal{F}, \tag{19}$$

that is, if $\mathcal{M}_k$ is correct by Definition 1, the updated map $\mathcal{M}_{k+1}$ will also be correct.

Proof: Directly apply Lemma 1 to each polytope in $\mathcal{S}_k$. ■

In words, the theorem shows that when each polytope in the map $\mathcal{M}_k$ is shrunk using Lemma 1, the new safe region $\mathcal{S}_{k+1}$ also remains certifiably obstacle-free. Once a given polytope has shrunk to zero volume, it can be forgotten entirely. Recall that as new camera frames are received, new polytopes can be constructed to define the free space in the operating environment and added to the set $\mathcal{S}_{k+1}$. We empirically study how quickly an environment deflates in Table III and in Section VIII. Naturally, if the odometry covariance is smaller, the deflation rate is smaller Appendix G.

Remark 1. *Compare* (11) *with* (17)*. The two are identical except for the $-\rho$ vector in* (17c)*. Each element $\rho_i \geq 0$, and therefore, this represents a shrinking operation. The net effect is that we transform the polytope by the estimated transform, but then shrink the polytope based on the odometry error covariance. Notice that this shrinking operation is tight: since there could exist an obstacle on the face of the polytope (indeed this is how they are constructed), the shrinking factor is the smallest allowable factor, by construction.*

Remark 2. *In implementation, notice that one needs to compute $\mathcal{V}_i(\mathcal{P}_k)$, the set of vertices, and then update the polyhedron by* (17c)*. Although this operation scales exponentially with the number of faces [38], efficient implementations exist, especially for 3D polytopes [37]. Empirically, we observe each polytope has on the order of 10-20 faces when using [8], and can be handled in real-time.*

V. Approach 2: Certified ESDFs

A. Background

The Euclidean Signed Distance Field (ESDF) is defined as the function $d : \mathbb{R}^3 \to \mathbb{R}$,

$$d(p) = \begin{cases} \operatorname{dist}(p, \partial\mathcal{O}), & \text{if } p \notin \mathcal{O} \\ -\operatorname{dist}(p, \partial\mathcal{O}), & \text{if } p \in \mathcal{O} \end{cases} \tag{20}$$

where $\partial\mathcal{O} \subset \mathbb{R}^3$ is the boundary of the obstacles. The dist measures the minimum distance of a point to a set, i.e., $\operatorname{dist}(p, \partial\mathcal{O}) = \min_{o\in\partial\mathcal{O}} \|p - o\|$. Thus, for any point in free-space,[7] the ESDF is given by

$$d(p) = \min_{o\in\mathcal{O}} \|o - p\|, \tag{21}$$

A 2D slice of the ESDF is depicted in Figure 2d.

To evaluate (21), o and p must be expressed in a common frame, commonly referred to as the mapping frame. Since this is done in the mapping frame, it is denoted as the function $d_M : \mathbb{R}^3 \to \mathbb{R}$. The claimed-safe region $\mathcal{S}_k$ is therefore

$$\mathcal{S}_k = \{p \in \mathbb{R}^3 : d_M(p) \geq 0\} \tag{22}$$

For safety-critical path planning and control, we need the ESDF at points relative to the body-fixed frame. The common approach is to assume the odometry estimate is exact, and determine $d(p|^{B_k})$ by expressing it in the map frame and evaluating d_M:

$$d(p|^{B_k}) \approx d_M(\widehat{T}_{B_k}^{M} \cdot p|^{B_k}) \tag{23}$$

However, since the estimate $\widehat{T}_{B_k}^{M}$ is inexact, this method can lead to over- or under-estimates. Overestimated distances are unsafe since they could lead to collisions.

B. Proposed Approach

The goal is to construct an ESDF that is safe, i.e., underestimates the distance to obstacles. Using Definition 1, a *Certified-ESDF* is defined as

Definition 2. *Let the obstacle set be $\mathcal{O} \subset \mathbb{R}^3$, assumed static in frame I. Let the ESDF of $\mathcal{O}$ be $d : \mathbb{R}^3 \to \mathbb{R}$. A* Certified-ESDF *(C-ESDF) at timestep k is a function $d_M^k : \mathbb{R}^3 \to \mathbb{R}$, such that for all points $p|^{B_k} \in \mathbb{R}^3$,*

$$d(p|^{B_k}) \geq d_M^k(\widehat{T}_{B_k}^{M} \cdot p|^{B_k}) \tag{24}$$

where $\widehat{T}_{B_k}^{M} \in \mathbb{SE}(3)$ is the estimated rototranslation between B_k and M.

Comparing (23) with (24), the goal of certification is to change the $\approx$ into $\geq$. That is, a Certified-ESDF is one where for any body-fixed point $p|^{B_k}$, if the point is expressed in the mapping frame *using the estimated rototranslation*, we have *underestimated* the distance to the nearest obstacle:

$$\underbrace{d(p|^{B_k}) = \min_{o\in\mathcal{O}} \left\|p|^{B_k} - o|^{B_k}\right\|}_{\text{true ESDF}} \geq \underbrace{d_M(\widehat{T}_{B_k}^{M} \cdot p|^{B_k})}_{\text{estimated ESDF}}. \tag{25}$$

[7] We use (21) instead of (20) for the remainder of the section for brevity. The points with $d(p) < 0$ will be removed from memory.

To accomplish this, we propose a strategy of deflating the ESDF. We derive a recursive guarantee to ensure the ESDF remains certified for all k.

Theorem 2. *Suppose at timestep $k \in \mathbb{N}$, the ESDF $d_M^k : \mathbb{R}^3 \to \mathbb{R}$ is a Certified-ESDF. Let the rototranslation between frames be $T_{B_{k+1}}^{B_k} \sim \mathcal{N}(\widehat{T}_{B_{k+1}}^{B_k}, \Sigma_k)$. Let the $d_M^{k+1} : \mathbb{R}^3 \to \mathbb{R}$ be defined by*

$$d_M^{k+1}(p|^M) = d_M^k(p|^M) - \sqrt{\lambda_{\max}(\Sigma_p)} \tag{26}$$

for all $p|^M \in \mathbb{R}^3$, where

$$\widehat{T}_{B_{k+1}}^{B_k} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \tag{27a}$$

$$J = \begin{bmatrix} R & -R[\widehat{T}_M^{B_{k+1}} \cdot p|^M]_\times \end{bmatrix}, \tag{27b}$$

$$\Sigma_p = \kappa J \Sigma_k J^T. \tag{27c}$$

and $\kappa > 0$ is as defined in Assumption 1. Given Assumption 1, d_M^{k+1} is also a Certified-ESDF at timestep $k+1$.

Proof Sketch: [See Appendix C for the full proof.] Consider any point $p|^{B_{k+1}}$ and evaluate the potential positions it could correspond to in frame B_k. This is an ellipsoid as in Assumption 1, and therefore the ESDF at $p|^{B_{k+1}}$ must be the minimum of all of the ESDF values for the corresponding points in the ellipsoid. Since, by definition, the Lipschitz constant of an ESDF is one, this minimum ESDF can be lower bounded by the ESDF at the center minus the radius of the smallest sphere containing the ellipsoid. We use the eigenvalues of the ellipsoid to compute the radius of sphere, arriving at the expression. ■

Remark 3. *Notice that the correction is $-\sqrt{\lambda_{max}(\Sigma_p)}$ in* (26) *(different for each p). As with the certified SFCs, this is a deflation operation that decreases the estimated distance to an obstacle.*

Remark 4. *The implementation of this deflation operation is remarkably simple and easily parallelized on a GPU. In our implementations, we added an additional deflation integrator to the code in [7]. At each frame, when the relative odometry with covariance is received, we can compute the deflation at each voxel in parallel using* (26).

VI. Safe Navigation with Certified Maps

Here we summarize the key ideas presented in this paper, and suggest strategies to achieve safe navigation.

A fundamental principle of our approach is ensuring that maps remain correct with respect to the body-fixed frame. To achieve this, we deflate the safe regions of the map based on the incremental odometry error at each timestep. The required deflation has an analytic expression.

Our implementation is as follows. When the $(k+1)$-th camera frame is received from the sensor, we compute the odometry estimate, and its relative covariance. Next, we apply the deflation step using the proposed algorithms. Finally, we incorporate new safe regions identified by the depth image to assimilate new information while discarding regions that can no longer be certifiably correct.

One can also maintain both the baseline and certified maps in memory simultaneously. While the memory usage increases, the certified maps tend to be smaller than the full map, maintaining both maps offers significant advantages. In particular, our certified mapping methods can integrate naturally with existing safety filtering methods like [5, 4]. These methods generate nominal trajectories to achieve mission objectives, but use a backup trajectory to ensure that the robot can safely stop based on the currently available information. In our framework, one can use the baseline map for nominal trajectory planning, but use the certified map for collision and safety checks. This combination enables agile motion while strictly guaranteeing safety.

VII. Simulations

We present results on the accuracy and correctness of both approaches for certified mapping presented above. As a reminder, the goal is to demonstrate that despite odometry drift, the region reported by our algorithms to be a part of the free space is indeed obstacle-free. First, we evaluate the performance of both the Certified SFCs and the Certified ESDFs methods on the Replica dataset (described below) and compare it to various baselines. Second, we have run hardware experiments with a rover, and show that by considering the certification bound the rover can avoid collisions. Additional results are reported in Appendix F and Appendix G.

Evaluation Method

We evaluated the performance of our implementations on the Replica dataset [16], with ground-truth trajectories generated as in [39]. From the ground-truth trajectory the RGBD image sequence was generated. We perturbed the trajectory to generate the estimated trajectory from a simulated odometry system as follows:

$$\widehat{T}_{B_{k+1}}^{B_k} = T_{B_{k+1}}^{B_k} \operatorname{Exp}(\tau), \quad \tau \sim \mathcal{N}(0, \Sigma) \tag{28}$$

where $T_{B_{k+1}}^{B_k} \in \mathbb{SE}(3)$ is the transform between subsequent frames of the ground-truth trajectory of the camera and $\widehat{T}_{B_{k+1}}^{B_k} \in \mathbb{SE}(3)$ is the estimated transform between subsequent frames used in the mapping algorithms. We used $\Sigma \in \{10^{-5}I, 10^{-6}I\}$. Evaluating the Absolute Translation Error (ATE) as in [40], the generated trajectories had between $1-3\%$ ATE, inline with the performance of state-of-the-art VIO methods. Each trajectory has 2000 frames at 30 FPS.

Baselines

We compared our proposed certified approaches to the following mapping methodologies:

(A) *Baseline SFC* - At each camera frame, the depth map is used to construct a pointcloud of obstacles within the current field of view. From this a convex polyhedron is extracted, and appended to a list of safe polyhedrons. The union of these polyhedrons is considered the safe

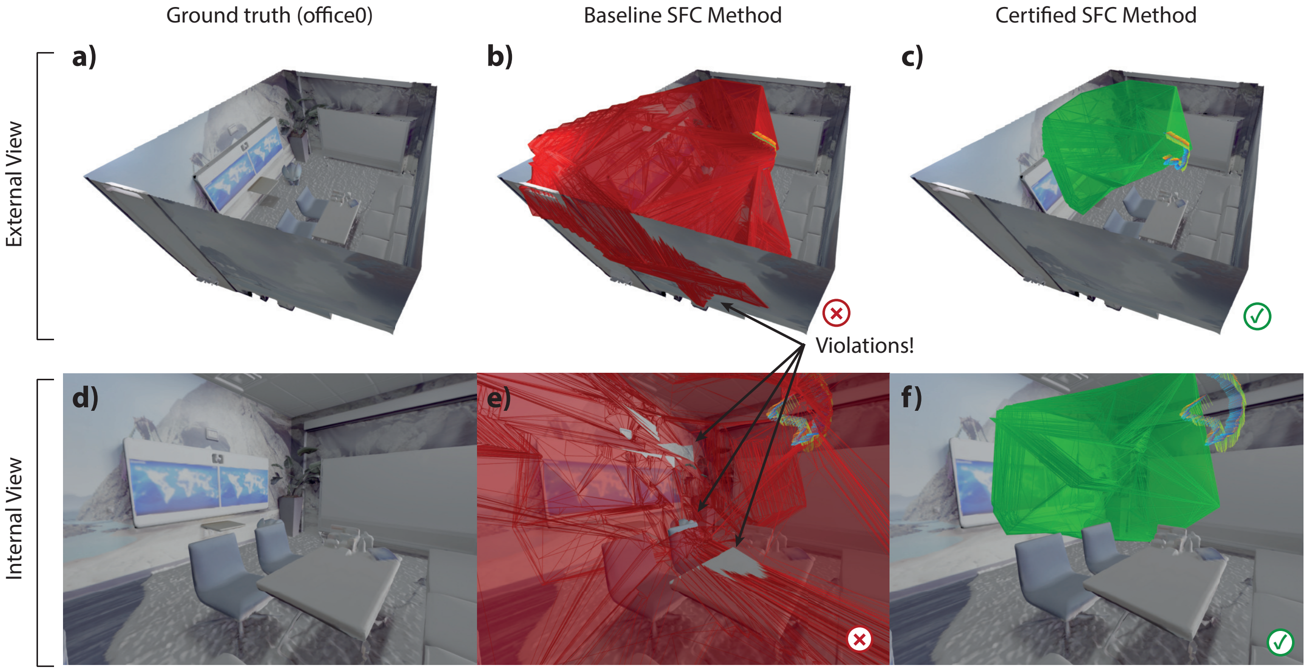

Fig. 3. Visualization of a snapshot of the `office0` environment mapped using the baseline and certified SFC methods. (a, d) shows the `office0` environment, while (b, e) and (c, f) show the respective $\mathcal{S}$ sets at the 500-th timestep from an external and an internal view. The baseline map claims a larger volume to be safe compared to the certified method (red volume is larger than green volume). However, we can also see numerous regions where the red region intersects with the ground truth mesh, indicating that the claimed safe region contains obstacle points. In the certified method, we see no violations.

flight region. We used the library [8] to perform the convex decomposition.

(B) *Heuristic SFC* - This is the same algorithm as in (A), except that a time-based forgetting mechanism is introduced, as is common in robotic mapping implementations. In particular, we only keep the last 60 frames (2 seconds) of polyhedrons when constructing the safe flight region.

(C) *Baseline ESDF* - At each camera frame, the depth map is used to update the TSDF of the environment. At regular intervals a wave propagation algorithm constructs/updates the ESDF of the environment. Regions with positive ESDF are considered part of the safe flight region. We used the library [7] to construct the TSDF and ESDF.

(D) *Heuristic ESDF* - This is the same algorithm as in (C), except that a distance-based forgetting mechanism is introduced. In particular, we forget any TSDF and ESDF voxels that are more than 3 m away from the camera.

These are compared to the proposed certified methods:

(E) *Certified SFC* - This is the same algorithm as in (A), except that at each frame, each polytope is deflated as described in Section IV.

(D) *Certified ESDF* - This is the same algorithm as in (C), except that at each frame, the ESDF is deflated as described in Section V.

Metrics

To evaluate the performance, we consider three metrics:

(I) *Violation Rate:* The violation rate measures the percentage of ground-truth mesh points that (incorrectly) lie within the claimed free space. The violation rate should be close to 0%.

(II) *Maximum Violation Distance:* For any violating point we measure the maximum distance of the violation, i.e., how far into the claimed free space is an obstacle point. The violation distance should be close to 0 mm. If there are no violating points, the violating distance is 0 mm.

(III) *Free-Space Volume:* This measures the total volume of the space that is claimed to be free. The free-space volume should be as large as possible.

Results

Tables I, II, and III summarize the results from the simulations. Figure 3 and Figure 4 visualize the results and qualitatively show the behavior of the proposed methods.

Figure 3 visualizes one of the runs from the `office0` environment. Figures (a, d) shows the ground-truth mesh of the environment from two different views. In (b, e) we see the safe flight polytopes in the baseline method visualized as the red region. One can see that the red region clearly intersects with the ground-truth mesh, and each intersection represents a violation. The violations are particularly noticeable for regions that were mapped further in the past, and from non-convex and thin obstacles like the chair or table surfaces. In contrast, in (c, f) we see the safe flight polytope from the proposed certified algorithms, drawn as the green region. We can see that the green region is smaller than the red polytope, but it also

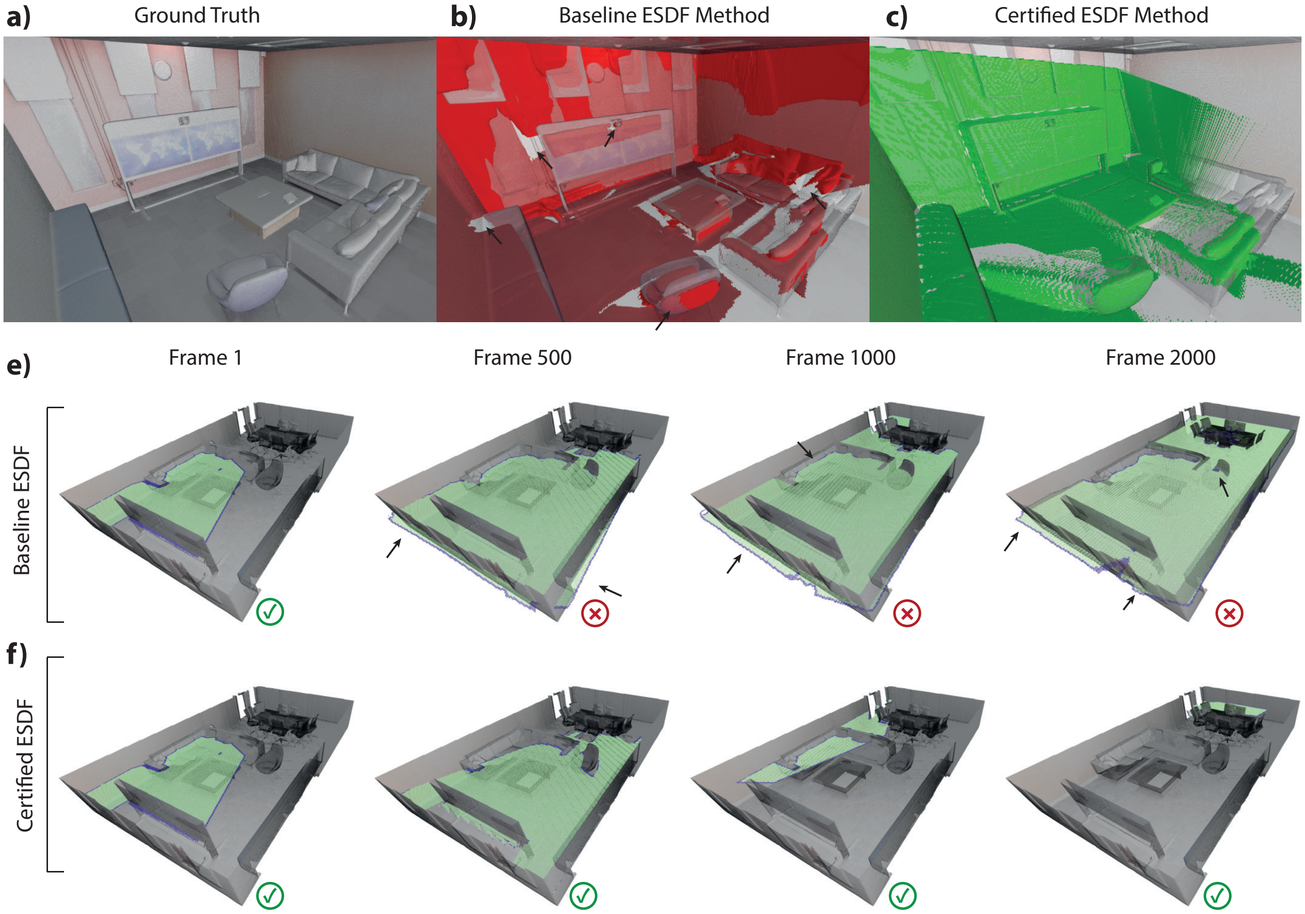

Fig. 4. Visualization of the maps generated using the baseline and certified ESDF methods on the `office3` environment. In (a) we see the ground-truth mesh. In (b) and (c) we can see the internal view after 500 timesteps. As in Figure 3, although the baseline method maps a larger volume (red mesh is larger than green mesh), it also contains many violations. In (e) and (f) we see a slice of the ESDF over time. The green region indicates the $\mathcal{S}$ set at the respective times. The small black arrows point to various violations in the baseline method, while in the certified methods we see no violations.

contains no violating points (see also Table II and Table III). Effectively, we can see that due to the odometry drift, the algorithm cannot be confident about the exact location of, for example, the chair and the desk, and therefore these regions were removed from the map. Although the volume of free space is smaller, the map is guaranteed to be correct.

From Table I we can observe that both certification methods significantly reduce the number of violations. In the baseline methods, the violation rates are between 6 and 60%, while in the certified methods, the violation rates are between 0-3%. Note, we cannot expect the certified methods to have exactly zero violations, since we are using the truncated noise model for odometry. Nonetheless, empirical performance of the certified methods still shows that the proposed methods can effectively avoid classifying obstacle regions as free.

Furthermore, we can see that although the heuristic forgetting methods can also reduce the number of violations, the level of reduction is hard to control. Since the forgetting factor is tuned heuristically and independently of the true noise level in the system, it can sometimes lead to good rejection of obstacles (as in the SFC method) or poor rejection of obstacles (as in the ESDF).

From Table II we observe that the maximum distance a violating point intersects the map is also reduced using the certified methods. We see that the maximum violation is sub-millimeter for the SFC methods, demonstrating a reduction of 2 orders of magnitude compared to the baseline. In the ESDF approaches, we still see a significant reduction in the maximum violation distance (about an order of magnitude reduction), although there are some violations on the order of 100 mm. This seems to be a limitation of the ESDF approach, since the ESDFs are represented using discrete voxels computationally. We chose a voxel size of 20 mm, and therefore the violations are on the order of 1-5 voxels of error.[8]

The source of this larger error is likely the dataset itself. We have checked which voxels are causing these large errors, and it seems to be the voxels that are close to non-manifold surfaces in the Replica dataset, for instance near the leaves of

[8] Finer grid resolution can help, but will increase the computational and memory requirements. As a sense of scale, each environment is on the order of $6 \times 6 \times 3$ m, and therefore has approximately $300 \times 300 \times 150$ voxels. See Table IV for additional details.

TABLE I

Violation Rates. This table summarizes the fraction of violating ground-truth obstacle points for each environment and algorithm. This table shows results with $\Sigma =$1e-6I.

Algorithm	Violation Rates (%)							
	`office0`	`office1`	`office2`	`office3`	`office4`	`room0`	`room1`	`room2`
Baseline SFC	18.60 %	12.76 %	10.13 %	12.74 %	14.44 %	10.74 %	19.17 %	6.85 %
Heuristic SFC	0.11 %	0.57 %	0.09 %	0.10 %	0.27 %	0.02 %	0.39 %	0.92 %
Certified SFC	0.0002%	0.0047%	0.0008%	0.0005%	0.0014%	0.0002%	0.0009%	0.0012%
Baseline ESDF	48.15 %	35.31 %	51.51 %	54.66 %	48.35 %	62.03 %	48.15 %	47.49 %
Heuristic ESDF	31.55 %	34.39 %	7.63 %	4.66 %	10.08 %	9.25 %	20.88 %	16.32 %
Certified ESDF	0.5443%	0.0610%	0.0809%	0.0227%	0.0538%	2.4259%	0.0149%	0.0519%

TABLE II

Maximum Violation Distance. This table summarizes the distance by which violating ground-truth obstacle points penetrate the estimated free space for each environment and algorithm. This table shows results with $\Sigma =$1e-6I.

Algorithm	Maximum Violation Distance (mm)							
	`office0`	`office1`	`office2`	`office3`	`office4`	`room0`	`room1`	`room2`
Baseline SFC	102.7	95.3	159.7	177.6	125.5	117.1	191.4	85.0
Heuristic SFC	22.1	14.5	18.4	11.6	8.9	11.0	14.2	12.8
Certified SFC	0.0	0.9	0.4	0.9	1.7	0.9	0.7	0.7
Baseline ESDF	604.3	406.9	520.0	671.1	636.9	990.8	604.6	594.0
Heuristic ESDF	563.6	379.5	311.8	429.4	366.6	428.5	384.7	435.4
Certified ESDF	109.5	82.5	141.4	100.0	66.3	120.0	100.0	82.5

TABLE III

Estimated Free Space Volume. This table summarizes the volume of the estimated free space at the end of the simulation for each environment and algorithm. This table shows results with $\Sigma =$1e-6I.

Algorithm	Estimated Free Space Volume (m^3)							
	`office0`	`office1`	`office2`	`office3`	`office4`	`room0`	`room1`	`room2`
Baseline SFC	34.8	17.6	40.8	56.6	63.3	53.0	38.7	29.4
Heuristic SFC	6.7	3.6	4.3	4.6	15.7	12.3	6.9	7.5
Certified SFC	5.7	2.6	3.6	3.0	12.5	9.1	5.8	4.4
Baseline ESDF	46.1	23.2	77.5	110.9	99.7	105.4	53.8	63.6
Heuristic ESDF	39.5	23.0	31.3	42.0	51.5	28.6	34.5	38.7
Certified ESDF	10.7	3.8	6.2	5.0	14.3	31.5	6.6	4.5

plants, or around table/chair legs, which are thin and long. Near these surfaces, the raw data is inconsistent, and we suspect that it leads to higher error rates than expected.

Finally, we can see that due to the certification the volume of the estimated free space is lower for the certified methods than it is for the heuristic or baseline methods (Table III). However, since the violation rate of the uncertified methods is significant, the free space cannot be trusted for path planning around obstacles. Despite the smaller volume of free space, the certified methods allow the full region to be trusted when used in planning (Section VIII).

Comparing the SFC and ESDF methods, in the results presented the SFC methods seem superior, since they have fewer violations, and the violating points violate the free space by a smaller distance. However this does come at the expense of expressiveness and computational cost. The SFC methods require the use of unions of convex polytopes to represent the free space, and in cluttered environments can sometimes lead to very small volumes of free space. The ESDF implementations are also more mature, with implementations like [7] allowing for efficient use of a GPU, which allows the ESDF to be computed more efficiently than the SFC.

VIII. Rover Experiments

In this section we demonstrate the utility of the proposed certified mapping frameworks in ensuring a robot can safely navigate an environment. We demonstrate that when a rover is tasked to navigate through an environment, and in particular reverse blindly into a region it previously mapped, the accumulated odometry error can lead to the rover colliding with previous mapped obstacles. Instead, by using the proposed methods, the rover will avoid traversing into regions that it can no longer certify are obstacle-free.

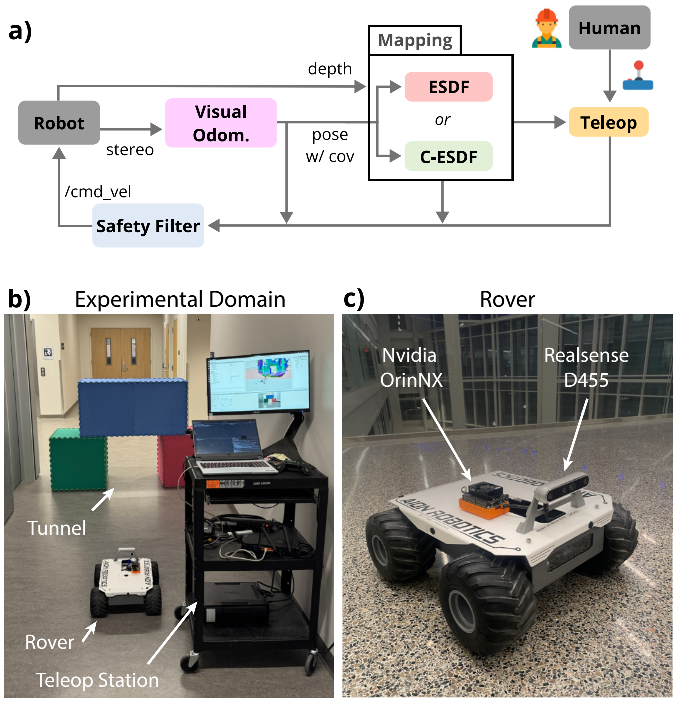

Fig. 5. Rover Experimental Setup. (a) Block diagram. The human is teleoperating the rover using only the FPV feed and the reconstructed obstacle map computed and streamed in real-time. The map is also used onboard the robot to stop the robot if it violates safety constraints. The safety filter can either use the baseline ESDF or the Certified ESDF. (b) Picture of the testing environment. The robot drives through the tunnel, mapping it as it passes through. After exploring the corridors, the rover tries to return through the tunnel in reverse, without remapping the tunnel. (c) shows the rover in more detail. The AION R1 UGV has been modified, with all sensing on Intel Realsense D455, and all compute on the Nvidia OrinNX 16GB.

Experimental Setup

A block diagram of the experimental setup is shown in Figure 5a). We use a ground rover, the AION R1 UGV equipped with an Intel Realsense D455 camera. All perception, planning, and control is executed on the onboard computer, an Nvidia Orin NX 16GB. The Realsense camera sends stereo infrared images to the Orin NX at 30FPS. A state-of-the-art visual slam algorithm (Nvidia IsaacROS Visual SLAM) is used to compute the odometry estimate. The Realsense camera also produces a depth image, which is sent to the obstacle mapping library (an adapted version of Nvidia IsaacRos NvBlox) which constructs an ESDF of the environment in real-time. All parameters and code is available at [redacted].

A human operator uses a joystick to send desired linear and angular velocities to the robot. Using the constructed ESDF, a safety filter forward propagates the robot's state under a desired command a short (0.5 s) horizon into the future and checks whether the trajectory lies strictly within $\mathcal{S}_k$. If so, the command is sent to the robots' motor controllers. If not, the safety filter zeros the linear command, and sends a reduced angular speed command. This allows the robot to continue to spin to acquire new information about the environment, without physically moving and potentially colliding with the obstacles. The safety filter was tuned offline to ensure that in the absence of odometry drift, the robot stops within 15 cm of the obstacle both when driving forwards or backwards.

To compute the certified-correct map, we use the techniques of Section V to compute the certified ESDF representing the local geometry. To correctly deflate the ESDF, we require the odometry estimate, and the covariance of the incremental transform between successive camera frames, i.e., of $\widehat{T}^{B_k}_{B_{k-1}}$.

To the best of the author's knowledge however, this information is not reported by any state-of-the-art odometry/pose estimation algorithms. Most algorithms (including Nvidia's vSLAM) only report the covariance of the odometry estimate between the initial frame and the current frame, i.e., of $\widehat{T}^{B_k}_{B_0}$. In [35] the authors computed the covariance of relative poses after solving a pose-graph optimization problem by using the Jacobian of the local solution (see [35, Section IX.B] for details). However this only allows one to find the covariance of relative transforms between keyframes, and does not allow one to find the relative transform between successive camera frames. Note, [29] reports the error covariance for frame-to-frame pointcloud matching, and could be integrated into the experiments below. However the accuracy of the pointcloud reported by the RGBD camera must also be considered [41].

Here, we use the following method to estimate the covariance between relative frames. VSLAM reports the odometry estimates $\widehat{T}^{B_k}_{B_0}$, $\widehat{T}^{B_{k+1}}_{B_0}$, and the associated covariances $\Sigma^{B_k}_{B_0}$, $\Sigma^{B_{k+1}}_{B_0}$. Assuming $T^{B_k}_{B_0}$ and $T^{B_{k+1}}_{B_0}$ are highly correlated since they are successive frames, we can define a correlation coefficient $\rho \in [-1, 1]$ (we use $\rho = 0.99$) between these camera frames. We can then estimate the covariance of the relative transform $\Sigma^{B_{k+1}}_{B_k}$ along the lines of [35]. The analysis is presented in Appendix D.

Experimental Results

Figure 6 summarizes the results of the rover experiments, with additional trials available in the supplementary video, all demonstrating similar outcomes.

The human operator's task was to navigate the rover without line-of-sight through a narrow tunnel, explore and map the environment, and return to the starting location by reversing through the tunnel. The rover was intentionally reversed through the tunnel to avoid re-mapping the obstacle geometry, forcing it to rely on its previously constructed maps.

Snapshots in Figure 6a show the baseline method. Initially, the tunnel and the surrounding corridors are mapped accurately. As the operator tries to reverse through the tunnel the final snapshot suggests that the rover is well aligned with the tunnel and is within the green region $\mathcal{S}$. However, despite this seemingly safe alignment, the rover collided with an obstacle Figure 6c, a failure in the baseline mapping approach.

In contrast, our proposed method deflates the safe regions in response to the odometry drift. In Figure 6b, the map initially classifies a large region as safe (green). However, as rover reverses to the tunnel, the deflation has caused parts of the map to turn red, indicating that these areas can no longer be certified to be obstacle free. Indeed, when the rover reaches

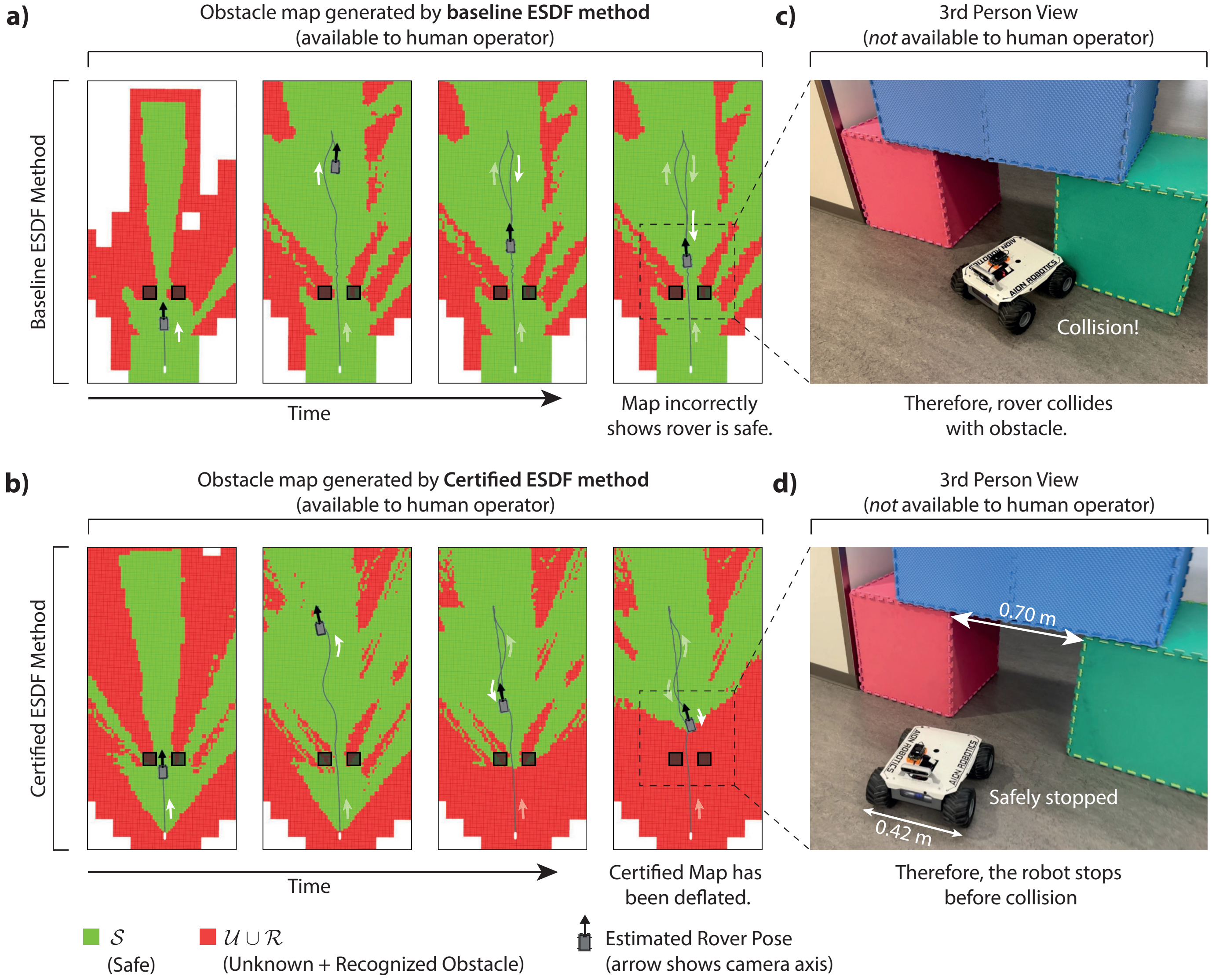
a)
Obstacle map generated by **baseline ESDF method**
(available to human operator)
Baseline ESDF Method
Time
Map incorrectly
shows rover is safe.
c)
3rd Person View
(*not* available to human operator)
Collision!
Therefore, rover collides
with obstacle.
b)
Obstacle map generated by **Certified ESDF method**
(available to human operator)
Certified ESDF Method
Time
Certified Map has
been deflated.
d)
3rd Person View
(*not* available to human operator)
0.70 m
Safely stopped
0.42 m
Therefore, the robot stops
before collision
🟩 $\mathcal{S}$
(Safe)
🟥 $\mathcal{U} \cup \mathcal{R}$
(Unknown + Recognized Obstacle)
Estimated Rover Pose
(arrow shows camera axis)

Fig. 6. Rover Experimental Results. (a, b) shows snapshots of the reconstructed obstacle map and the estimated rover pose with (a) the baseline method and (b) the certified method. This is the view presented to the human teleoperating the robot. Note, two small black boxes are drawn in each frame (in post) to indicate to the reader the location of the red and green boxes during the experiment. These were not visible to the human operator during the experiments. (c, d) show the final state of the robots at the end of the trajectory. In (c), the baseline method the robot has crashed with the green obstacle, although looking at the last panel of (a), we can see that the robot thinks it is in the middle of the tunnel in the free space. In (d), we see the robot stopped 15 cm before crashing with the red obstacle, and this is because the map has been deflated sufficiently that the safety filter prevents the robot from continuing backwards. Notice between the second, third and fourth frames in (b) the green regions near the bottom change into red regions, indicating the Certified ESDF cannot certify that the red region is obstacle-free.

the boundary between red and green regions, the safety filter prevents further motion, successfully preventing collision. The same behavior was consistently observed across multiple trials.

Larger Scale Experiments

In this section, we show qualitatively and quantitatively the volume of free space usable by a robotic system. The rover was operated in a room approximately 40×20 m large drawn in Figure 7. Starting in the middle, the robot was teleoperated to explore and map the room. The robot has a horizontal field of view of 75°, and a maximum depth integration distance of 8 m. This means that from the depth image, the maximum distance that NvBlox will mark as free or safe is 8 m from the camera origin. Thus, in these experiments, the heuristic method also uses a forgetting radius of 8 m.

A quantitative comparison of the algorithms is presented in Figure 8a, b. In (a) we can see the area of the claimed safe region by each of the three methods. Although the claimed free region is largest for the baseline method, the map is erroneous. The certified and heuristic methods have similar free area, although the heuristic method is also often incorrect.

In Figure 8b, we show the distance to the furthermost safe point from the robot position. This gives an indication of extent of the map that would be free if it were not for the obstacles

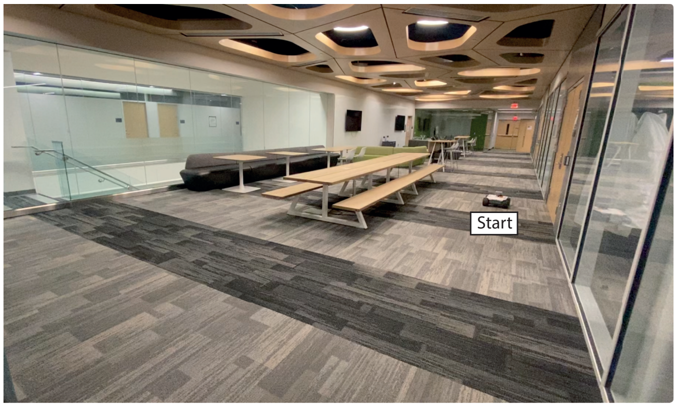

Fig. 7. Experimental domain used in Figure 8.

in the environment. Here, we can see that compared to the maximum integration distance of 8 m, the certified method has its furthermost safe voxel approximately 12 m away, and upto 18 m away. In contrast, the heuristic method is clipped at 8 m. The evolution of the maps in time is clearer in the accompanying video, where the FPV and third person view of the robot are also drawn.

Slices of the ESDF and the Certified ESDF are shown in Figure 8c, d. The robot's trajectory is also drawn. Compare Figure 8c1-c4. We can see that the map drifts significantly - in (c1) we use a gray dashed line to highlight the end of the corridor as mapped at that time. In (c4), we draw the corridor mapped in (c1) as well as the newly mapped corridor, and we can see a significant shift in the map. In (d1-d4) we can see the certified ESDF region marked in green, and even as the robot moves around a significant part of the area around the robot remains part of the safe region.

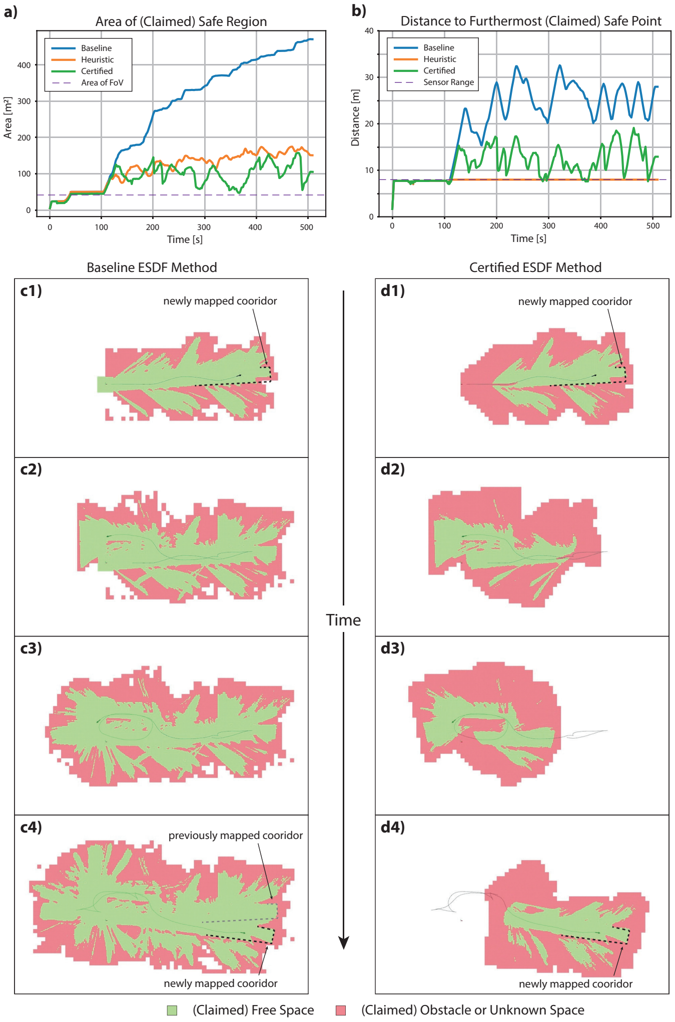

Fig. 8. Quantitative and qualitative analysis of the effect of the deflation on the volume of the certified free space. (a) Compares the area of the claimed safe region on a 2D slice of the ESDF extracted at the robot height. As a reference, the area of the FoV of the camera is also drawn (black dashed line). (b) Compares the distance of the furthermost (claimed) free voxel from the robot position. As a reference, the maximum depth of the depth sensor (8 m) is indicated (black dashed line). In (c1-c4) we see snapshots of the map generated by the Baseline ESDF method, and in (d1-d4) we see the corresponding snapshots from the Certified ESDF method. The supplementary video animates the map slices.

IX. Conclusions

Limitations and Future Directions

While the proposed methods are provably correct, they rely on key assumptions, particularly Assumption 1, which truncates the normal distribution of pose perturbations to bound the effects of a rototranslation on an obstacle point. Although this simplification facilitates our framework, it may not hold in practice. Methods such as those in [36, 35] could improve these approximations and warrant further exploration.

Additionally, we assumed that incremental odometry perturbations follow a normal distribution in the Lie algebra of $\mathbb{SE}(3)$. However, this may not hold in practice, especially with outliers (see e.g. [26]). A valuable direction for future work is to rigorously characterize the error distribution of odometry systems, both analytically and empirically.

We also highlight the need for modern perception algorithms to report the uncertainty of incremental pose transforms (e.g., as in [29]), rather than overall pose error/covariance, which grow unbounded without successful loop closures. Metrics such as relative translation and rotation errors [40] or the correlation between pose uncertainties (as in [35]) should be computed and reported. In lieu of this, our experiments estimated incremental pose error covariances using the method described in Appendix D. For certifiability guarantees, going forward we will need odometry algorithms capable of directly reporting the incremental pose error covariance.

Our algorithm intentionally deflates the map, and this reduces the navigable volume for the robot. It is challenging to estimate how much the volume reduces prior to a mission, since the deflation depends on the exact obstacle geometry, features used by the odometry algorithm, and the speed of the robot (which affects how quickly new parts of the environment are observed). Empirically, we have shown that as the odometry covariance decreases, the volume of the free space increases, and approaches the volume of baseline methods in the error-free case (Appendix G). We also operated our rover in a larger room, and in Section VIII we show empirically that the certified methods can yield similar or larger volumes of free space than the heuristic method. Further analysis into this warranted.

While this paper focused on deflating the map to ensure correctness, future work can consider methods to reinflate deflated regions when the correctness can be guaranteed again. For example, when a loop closure is detected, the odometry drift is reduced, and therefore uncertified regions can perhaps be marked as certifiably free again. To achieve this however we will require further analysis into the correctness guarantees of loop closures (e.g. [24]), as well as efficient algorithms and map representations to handle the inflation and deflation steps.

Beyond odometry drift, there are other sources of error that can invalidate the correctness of the map - the operating environment and each subsystem can introduce errors that are hard to correct or even detect. For instance, depth estimation algorithms (e.g., block-matching methods) can fail under conditions like glass surfaces or featureless walls. Similarly, communication/computational latencies can introduce errors that are hard to characterize with the current framework.

Summary

As robots increasingly operate in unstructured environments, the importance of tightly integrated perception, planning, and control systems becomes evident. Our experiments demonstrate that even over short distances, perception inaccuracies due to odometry drift can lead to unsafe behaviors, including collisions.

This paper presents a step toward building perception modules that not only generate accurate state estimates and obstacle maps but also provide correctness guarantees. Specifically, if the incremental odometry error per frame can be bounded, our framework modifies (or deflates) obstacle-free regions in a map such that it remains correct at all times with respect to the robot's body frame.

We proposed two methods for implementing these corrections based on different map representations: (I) Certified SFCs, and (II) Certified ESDFs. By constructively proving the correctness of these methods, we developed algorithms that guarantee safe map modifications. Extensive simulations using high-quality datasets, along with real-world experiments on a robotic rover, validate the effectiveness of our approach in creating certifiably-correct maps.

A key insight from our rover experiments is the demonstration of failure modes in state-of-the-art mapping methods. Unlike typical demonstrations, where robots map regions within the camera's field of view or use 360° sensors (e.g., LIDAR), we intentionally operated the robot in its blind spot to highlight the challenges posed by accumulated odometry drift. Our proposed methods successfully mitigated these issues, preventing collisions and ensuring safe navigation.

Acknowledgments

The authors would like to acknowledge the support of the National Science Foundation (NSF) under grant no. 1942907.

References

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE TAC*, vol. 62, no. 8, pp. 3861–3876, 2016.

[2] K. Garg, J. Usevitch, J. Breeden, M. Black, D. Agrawal, H. Parwana, and D. Panagou, "Advances in the theory of control barrier functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems," *arXiv preprint arXiv:2312.16719*, 2023.

[3] B. T. Lopez and J. P. How, "Aggressive 3-D collision avoidance for high-speed navigation.," in *IEEE ICRA*, pp. 5759–5765, 2017.

[4] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *IEEE IROS*, pp. 1934–1940, IEEE, 2019.

[5] D. R. Agrawal, R. Chen, and D. Panagou, "gatekeeper: Online safety verification and control for nonlinear systems in dynamic environments," *IEEE Transactions on Robotics*, 2024.

[6] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE IROS*, pp. 1366–1373, IEEE, 2017.

[7] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvblox: Gpu-accelerated incremental signed distance field mapping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2698–2705, IEEE, 2024.

[8] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[10] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," in *IEEE ICRA*, pp. 3437–3444, IEEE, 2023.

[11] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. & Automat. Mag.*, vol. 18, no. 4, pp. 80–92, 2011.

[12] Z. Yu, L. Zhu, and G. Lu, "Vins-motion: tightly-coupled fusion of vins and motion constraint," in *IEEE ICRA*, pp. 7672–7678, IEEE, 2021.

[13] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE TRO*, vol. 38, no. 4, 2022.

[14] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *IEEE ICRA*, pp. 3983–3989, IEEE,

2023.

[15] R. Merat, G. Cioffi, L. Bauersfeld, and D. Scaramuzza, "Drift-free visual slam using digital twins," *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1633–1640, 2025.

[16] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

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

[18] 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.

[19] Nvidia-Isaac, "cuVSLAM." https://nvidia-isaac-ros.github.io/concepts/visual_slam/cuvslam/index.html, 2024.

[20] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-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.

[21] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, *et al.*, "Present and future of slam in extreme environments: The DARPA SubT challenge," *IEEE Transactions on Robotics*, vol. 40, pp. 936–959, 2023.

[22] T. H. Chung, V. Orekhov, and A. Maio, "Into the robotic depths: Analysis and insights from the darpa subterranean challenge," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, no. 1, pp. 477–502, 2023.

[23] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[24] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: a certifiably correct algorithm for synchronization over the special euclidean group," *IEEE IJRR*, vol. 38, no. 2-3, pp. 95–125, 2019.

[25] M. Marchi, J. Bunton, B. Gharesifard, and P. Tabuada, "LiDAR point cloud registration with formal guarantees," in *IEEE CDC*, pp. 3462–3467, 2022.

[26] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE TRO.*, vol. 37, no. 2, pp. 314–333, 2020.

[27] D. R. Agrawal, R. Govindjee, J. Yu, A. Ravikumar, and D. Panagou, "Online and certifiably correct visual odometry and mapping," *arXiv preprint arXiv:2402.05254*, 2024.

[28] J. Zhang and S. Singh, "Ins assisted monocular visual odometry for aerial vehicles," in *Field and Service Robotics: Results of the 9th International Conference*, pp. 183–197, Springer, 2015.

[29] F. A. Maken, F. Ramos, and L. Ott, "Stein icp for uncertainty estimation in point cloud matching," *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 1063–1070, 2021.

[30] J. Laconte, D. Lisus, and T. D. Barfoot, "Toward certifying maps for safe registration-based localization under adverse conditions," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1572–1579, 2023.

[31] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena, "C-blox: A scalable and consistent tsdf-based dense mapping approach," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 995–1002, IEEE, 2018.

[32] A. Howard, G. S. Sukhatme, and M. J. Mataric, "Multirobot simultaneous localization and mapping using manifold representations," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1360–1369, 2006.

[33] T. Cieslewski, A. Ziegler, and D. Scaramuzza, "Exploration without global consistency using local volume consolidation," in *The International Symposium of Robotics Research*, pp. 559–574, Springer, 2019.

[34] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[35] J. G. Mangelson, M. Ghaffari, R. Vasudevan, and R. M. Eustice, "Characterizing the uncertainty of jointly distributed poses in the lie algebra," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1371–1388, 2020.

[36] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[37] B. Legat, "Polyhedral computation," in *JuliaCon*, July 2023.

[38] K. Fukuda and A. Prodon, "Double description method revisited," in *Franco-Japanese and Franco-Chinese conference on combinatorics and computer science*, pp. 91–111, Springer, 1995.

[39] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[40] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7244–7251, IEEE, 2018.

[41] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3d reconstruction and tracking," in *Intl. Conf. 3D imaging, modeling, processing, visualization & transmission*, pp. 524–530, IEEE, 2012.

Appendix

A. *Review of Matrix Lie Groups*

Here we review the fundamentals of representing a pose and its uncertainty through the language of Lie groups and Lie algebras. We refer to readers to [34, 35, 36] and references therein for a more complete description.

The Lie group $\mathbb{SO}(3)$ is the set of valid 3D rotation matrices, and the group $\mathbb{SE}(3)$ is the set of rigid transformations in 3D:

$$\begin{aligned}\mathbb{SO}(3) &= \left\{R \in \mathbb{R}^{3\times 3} : RR^T = I_3, \det R = 1\right\},\\ \mathbb{SE}(3) &= \left\{T = \begin{bmatrix} R & t \\ 0 & 1\end{bmatrix} \in \mathbb{R}^{4\times 4} : R \in \mathbb{SO}(3), t \in \mathbb{R}^3\right\}.\end{aligned}$$

Both $\mathbb{SO}(3)$ and $\mathbb{SE}(3)$ are matrix Lie groups, i.e., the group composition operation is the standard matrix multiplication operation.

The group action for $\mathbb{SE}(3)$ is $\cdot : \mathbb{SE}(3) \times \mathbb{R}^3 \to \mathbb{R}^3$, which transforms a point p from its representation in frame A to that in frame B. Given $T_A^B = \begin{bmatrix} R & t \\ 0 & 1\end{bmatrix} \in \mathbb{SE}(3)$,

$$p|^B = T_A^B \cdot p|^A = Rp|^A + t. \tag{29}$$

The tangent space centered at identity is called the Lie algebra of a Lie group. The Lie algebra is a vector space of all possible directions an element of the group can be perturbed locally. The Lie algebras of $\mathbb{SO}(3)$ and $\mathbb{SE}(3)$ are denoted $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ respectively:

$$\begin{aligned}\mathfrak{so}(3) &= \left\{\omega \in \mathbb{R}^{3\times 3} : \omega^T = -\omega\right\},\\ \mathfrak{se}(3) &= \left\{\begin{bmatrix} \omega & \rho \\ 0 & 0\end{bmatrix} \in \mathbb{R}^{4\times 4} : \omega \in \mathfrak{so}(3), \rho \in \mathbb{R}^3\right\}.\end{aligned}$$

These vector spaces are isomorphic to the Euclidean vector space $\mathbb{R}^3$ and $\mathbb{R}^6$ respectively. The $\wedge$ operator converts the Euclidean vector to an element of the Lie Algebra. For $\mathbb{SO}(3)$, $\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$:

$$\phi^\wedge = \begin{bmatrix}\phi_1 \\ \phi_2 \\ \phi_3\end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0\end{bmatrix} \tag{30}$$

while for $\mathbb{SE}(3)$, $\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$:

$$\xi^\wedge = \begin{bmatrix}\rho \\ \phi\end{bmatrix}^\wedge = \begin{bmatrix}\phi^\wedge & \rho \\ 0 & 0\end{bmatrix}. \tag{31}$$

The $\vee$ operator performs the inverse of $\wedge$.

Given an element of the Lie algebra, we can convert it to the corresponding element of the group using the exponential map. For $\mathbb{SE}(3)$, the exponential map is $\exp : \mathfrak{se}(3) \to \mathbb{SE}(3)$,

$$\exp(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!} = I + X + \frac{X^2}{2} + \cdots \tag{32}$$

For convenience, we also define the Exp map, which maps from the Euclidean representation directly to the group element, $\mathrm{Exp} : \mathbb{R}^6 \to \mathbb{SE}(3)$,

$$\mathrm{Exp}(\xi) = \exp(\xi^\wedge). \tag{33}$$

Analytic expressions for this are provided in [34, Appendix]. The corresponding inverse operations are $\log$ and Log.

The adjoint matrix of $\mathbb{SE}(3)$ at $T \in \mathbb{SE}(3)$ is the unique matrix $\mathrm{Ad}_T \in \mathbb{R}^{6\times 6}$ such that

$$T\,\mathrm{Exp}(\xi) = \mathrm{Exp}(\mathrm{Ad}_T\, \xi)T \tag{34}$$

for all $\xi \in \mathbb{R}^6$. Again, the analytic expression is available in [34, Appendix].

B. *Proof of Lemma 1*

Before we prove Lemma 1, we derive a separating hyperplane result, Lemma 2. It defines the hyperplane that separates potential obstacle points from the free space after an uncertain rigid transformation.

Lemma 2. *Let the transform between two frames be $T_A^B \sim \mathcal{N}(\widehat{T}_A^B, \Sigma)$. Consider a point $p|^A \in \mathbb{R}^3$. Given Assumption 1, for any non-zero vector $a \in \mathbb{R}^3$,*

$$p|^B = T_A^B \cdot p|^A \in \mathcal{H} \tag{35}$$

where

$$\mathcal{H} = \{p \in \mathbb{R}^3 : a^T p \geq r\} \tag{36a}$$

$$r = a^T(\widehat{T}_A^B \cdot p|^A) - \sqrt{a^T \Sigma_p a} \tag{36b}$$

and $\Sigma_p \in \mathbb{S}_+^3$ is as defined by Assumption 1.

Proof of Lemma 2: By Assumption 1, the transformed point satisfies

$$p|^B \in \mathcal{E} = \left\{p \in \mathbb{R}^3 : \left\| \Sigma_p^{-1/2}(p - \hat{p}) \right\| \leq 1 \right\}$$

where $\hat{p} = \widehat{T}_A^B \cdot p|^A$, and $\Sigma_p \in \mathbb{S}_+^3$ is defined in Assumption 1. Next, we define

$$p^\perp = \hat{p} - \frac{\Sigma_p a}{\sqrt{a^T \Sigma_p a}}$$

such that $p^\perp \in \mathbb{R}^3$ is on the surface of the ellipsoid and has a surface normal $-a$. Therefore, the set of points $\mathcal{H} = \{p \in \mathbb{R}^3 : a^T(p - p^\perp) \geq 0\}$ contains the ellipsoid, i.e., $\mathcal{E} \subset \mathcal{H}$,

$$r = a^T p^\perp = a^T \hat{p} - \frac{a^T \Sigma_p a}{\sqrt{a^T \Sigma_p a}} = a^T \hat{p} - \sqrt{a^T \Sigma_p a}$$

which completes the proof. ■

We can now prove Lemma 1.

Proof of Lemma 1: It suffices to show that any obstacle potentially on the boundary of $\mathcal{P}_k$ will not be in $\mathcal{P}_{k+1}$. Consider an obstacle point $o|^{B_k} = p|^{B_k} + \epsilon a_k$, where $\epsilon > 0$ and $p|^{B_k}$ is a point on the surface of $\mathcal{P}_k$. Then for some $i \in \{1, ..., N\}$,

$$a_{k,i}^T p|^{B_k} = b_{k,i}.$$

After the rigid transformation, by Lemma 2, $o|^{B_{k+1}} \in \mathcal{E} \subset \{p : a_{k+1,i}^T p \geq r\}$ where

$$\begin{aligned} r &= a_{k+1,i}^T(\widehat{T}_{B_k}^{B_{k+1}} \cdot o|^{B_k}) - \sqrt{a_{k+1,i}^T \Sigma_p a_{k+1,i}} \\ &= a_{k+1,i}^T(R(p|^{B_k} + \epsilon a_{k,i}) + t) - \sqrt{a_{k+1,i}^T \Sigma_p a_{k+1,i}} \\ &= a_{k,i}^T(p|^{B_k} + \epsilon a_{k,i}) + a_{k,i}^T R^T t - \sqrt{a_{k+1,i}^T \Sigma_p a_{k+1,i}} \\ &= b_{k,i} + \epsilon \left\|a_{k,i}\right\|^2 + a_{k,i}^T R^T t - \sqrt{a_{k+1,i}^T \Sigma_p a_{k+1,i}} \\ &= b_{k+1,i} + \epsilon \left\|a_{k,i}\right\|^2 + \rho_i - \sqrt{a_{k+1,i}^T \Sigma_p a_{k+1,i}} \end{aligned}$$

Now consider the last term:

$$\begin{aligned}
\sqrt{a_{k,i+1}^T \Sigma_p a_{k+1,i}} &= \left\| \Sigma_p^{1/2} a_{k+1,i} \right\| \\
&= \left\| \sqrt{\kappa} \Sigma_k^{1/2} J^T a_{k+1,i} \right\| \\
&= \left\| \sqrt{\kappa} \Sigma_k^{1/2} \begin{bmatrix} R^T \\ -(R[o|^{B_k}]_\times)^T \end{bmatrix} a_{k+1,i} \right\| \\
&= \left\| \sqrt{\kappa} \Sigma_k^{1/2} \begin{bmatrix} a_{k,i} \\ [o|^{B_k}]_\times a_{k,i} \end{bmatrix} \right\| \\
&= \left\| \sqrt{\kappa} \Sigma_k^{1/2} \begin{bmatrix} a_{k,i} \\ -[a_{k,i}]_\times o|^{B_k} \end{bmatrix} \right\| \\
&= \left\| \sqrt{\kappa} \Sigma_k^{1/2} \begin{bmatrix} a_{k,i} \\ -[a_{k,i}]_\times p|^{B_k} \end{bmatrix} \right\|
\end{aligned}$$

where in the last line, we used $[a_{k,i}]_\times (\epsilon a_{k,i}) = 0$.

Finally, since Σ_k is positive definite, this expression is convex wrt $p|^{B_k}$. Considering $p|^{B_k}$ must be some convex combination of the vertices on the i-th face,

$$\begin{aligned}
\left\| \Sigma_p^{1/2} a_{k+1,i} \right\| &\le \max_{j \in \{1,\ldots,m_i\}} \left\| \sqrt{\kappa} \Sigma_k^{1/2} \begin{bmatrix} a_{k,i} \\ -[a_{k,i}]_\times v_{i,j}|^{B_k} \end{bmatrix} \right\| \\
&= \rho_i
\end{aligned}$$

where $v_{i,j}|^{B_k}$ is the j-th vertex on the i-th face of $\mathcal{P}_k$.

Therefore, we have

$$\begin{aligned}
r &= b_{k+1,i} + \epsilon \left\| a_{k,i} \right\|^2 + \rho_i - \left\| \Sigma_p^{1/2} a_{k+1,i} \right\| \\
&\ge b_{k+1,i} + \epsilon \left\| a_{k,i} \right\|^2 > b_{k+1,i},
\end{aligned}$$

that is,

$$\begin{aligned}
& o|^{B_{k+1}} \in \mathcal{E} \subset \{p : a_{k+1,i}^T p \ge r\}, \\
& \implies o|^{B_{k+1}} \notin \{p : a_{k+1,i}^T p \le b_{k+1,i}\}
\end{aligned}$$

which completes the proof. ■

C. *Proof of Theorem 2*

Proof: Consider any point $p|^{B_{k+1}}$. When represented in frame B_k, it could correspond to a set of points within the ellipsoid

$$p|^{B_k} \in \mathcal{E} = \{p \in \mathbb{R}^3 : \left\| \Sigma_p^{-1/2}(p - \hat{p}) \right\| \leq 1\}$$

where $\hat{p} = \widehat{T}^{B_k}_{B_{k+1}} \cdot p|^{B_{k+1}}$, and $\Sigma_p \in \mathbb{S}^3_+$ is as defined by Assumption 1. Therefore,

$$\begin{aligned} d(p|^{B_{k+1}}) &\overset{(1)}{\geq} \min_{p|^{B_k} \in \mathcal{E}} d(p|^{B_k}) \\ &\overset{(2)}{\geq} \min_{p|^{B_k} \in \mathcal{E}} d^k_M(\widehat{T}^M_{B_k} \cdot p|^{B_k}) \\ &\overset{(3)}{\geq} d^k_M(\widehat{T}^M_{B_k} \cdot \hat{p}) - \operatorname{diam}(\mathcal{E})/2 \\ &\overset{(4)}{=} d^k_M(\widehat{T}^M_{B_k} \widehat{T}^{B_k}_{B_{k+1}} \cdot p|^{B_{k+1}}) - \sqrt{\lambda_{\max}(\Sigma_p)} \\ &\overset{(5)}{=} d^k_M(\widehat{T}^M_{B_{k+1}} \cdot p|^{B_{k+1}}) - \sqrt{\lambda_{\max}(\Sigma_p)} \\ &\overset{(6)}{=} d^{k+1}_M(\widehat{T}^M_{B_{k+1}} \cdot p|^{B_{k+1}}) \end{aligned}$$

where $\operatorname{diam}(\mathcal{E})$ is the diameter of $\mathcal{E}$. (1) is true by defition, (2) uses the fact that d^k_M is a certified-ESDF. (3) is true because ESDFs have unit gradient everywhere, (4) uses the eigenvalue of Σ_p to bound the ellipsoid with a sphere, and (5), and (6) are basic simplifications. Therefore, d^{k+1}_M is also a certified-ESDF. ■

D. Extracting Covariance of Relative Transforms from Odometry with Covariance

To the best of the author's knowledge, all Visual Odometry (VO)/VIO/SLAM algorithms report the mean odometry estimate and the covariance with respect to the initial frame: at the k-th frame, the following quantities are available:

$$\widehat{T}^{B_0}_{B_k} \in \mathbb{SE}(3), \quad \Sigma^{B_0}_{B_k} \in \mathbb{S}^6_+ \tag{37}$$

i.e., the pose of the k-th body frame with respect to the initial frame, and the covariance of the estimate.

However, to use the frameworks proposed in this paper, the relative transform and its covariance are required:

$$\widehat{T}^{B_k}_{B_{k+1}} \in \mathbb{SE}(3), \quad \Sigma^{B_k}_{B_{k+1}} \in \mathbb{S}^6_+. \tag{38}$$

Here we detail a method to obtain these quantities.

Consider the following result adapted from [35, Section VIII] to match the convention used in this paper.

Lemma 3. *Let $T_{ij}, T_{ik}, T_{jk} \in \mathbb{SE}(3)$ represent the poses between coordinate frames $(i,j), (i,k)$, and (j,k) respectively. Let $\hat{T}_\cdot$ be the corresponding estimated transform. Let*

$$T_{ij} = \hat{T}_{ij} \operatorname{Exp}(\xi_{ij}) \tag{39}$$

and similar for $(ik), (jk)$. Suppose

$$\begin{bmatrix} \xi_{ij} \\ \xi_{ik} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_{ij} & \Sigma_{ij,jk} \\ \Sigma_{ij,ik}^T & \Sigma_{ik} \end{bmatrix} \right). \tag{40}$$

Then, the estimated relative transform is

$$\hat{T}_{jk} = \hat{T}_{ij}^{-1} \hat{T}_{ik} \tag{41}$$

and the associated covariance is (to first order)

$$\Sigma_{jk} = A \Sigma_{ij} A^T + \Sigma_{ik} - A \Sigma_{ij,ik} - \Sigma_{ij,jk}^T A^T, \tag{42}$$

where $A = \operatorname{Ad}_{\hat{T}_{jk}^{-1}} \in \mathbb{R}^{6\times6}$ is the adjoint matrix of $\mathbb{SE}(3)$ at $\hat{T}_{jk}^{-1}$.

Notice that the negative signs on the cross terms implies that a non-zero $\Sigma_{ij,jk}$ decreases the covariance of the relative pose.

Proof: Since $T_{jk} = T_{ij}^{-1} T_{ik}$, the following must hold:

$$\begin{aligned} \hat{T}_{jk} \operatorname{Exp}(\xi_{jk}) &= \left(\hat{T}_{ij} \operatorname{Exp}(\xi_{ij}) \right)^{-1} \left(\hat{T}_{ik} \operatorname{Exp}(\xi_{ik}) \right) \\ &= \operatorname{Exp}(-\xi_{ij}) \hat{T}_{ij}^{-1} \hat{T}_{ik} \operatorname{Exp}(\xi_{ik}) \\ &= \operatorname{Exp}(-\xi_{ij}) \hat{T}_{jk} \operatorname{Exp}(\xi_{ik}) \\ &= \hat{T}_{jk} \operatorname{Exp}(-\operatorname{Ad}_{\hat{T}_{jk}^{-1}} \xi_{ij}) \operatorname{Exp}(\xi_{ik}) \end{aligned}$$

where in the last equality we used the following property of the adjoint matrix: $\operatorname{Exp}(\xi) T = T \operatorname{Exp}(\operatorname{Ad}_{T^{-1}} \xi)$ for any $T \in \mathbb{SE}(3)$ and $\xi \in \mathbb{R}^6$.

Defining $\xi'_{ij} = -\operatorname{Ad}_{\hat{T}_{jk}^{-1}} \xi_{ij}$, we have

$$\operatorname{Exp}(\xi_{jk}) = \operatorname{Exp}(\xi'_{ij}) \operatorname{Exp}(\xi_{ik})$$

and therefore using the Baker-Campbell-Hausdorff (BCH) formula (see [35]), the first order estimated covariance is

$$\begin{aligned} E[\xi_{jk} \xi_{jk}^T] &\approx \underbrace{E[\xi'_{ij} {\xi'_{ij}}^T] + E[\xi_{ik} \xi_{ik}^T]}_{\text{2nd order diag. terms}} \\ &\quad + \underbrace{E[\xi'_{ij} \xi_{ik}^T] + E[\xi_{ik} {\xi'_{ik}}^T]}_{\text{2nd order cross terms}} \\ &= A \Sigma_{ij} A^T + \Sigma_{ik} - A \Sigma_{ij,ik} - \Sigma_{ij,jk}^T A^T \end{aligned}$$

where $A = \operatorname{Ad}_{\hat{T}_{jk}^{-1}}$. This completes the proof. ■

We can now apply this lemma to estimate the relative transforms between successive frames. Recall the odometry algorithm defines the covariances as

$$T^{B_0}_{B_k} = \widehat{T}^{B_0}_{B_k} \operatorname{Exp}(\xi_{k,0}), \quad \xi_{k,0} \sim \mathcal{N}(0, \Sigma_{k,0}) \tag{43}$$

and similar for $k+1$. The perturbations ξ are assumed to be correlated,

$$\begin{bmatrix} \xi_{k,0} \\ \xi_{k+1,0} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_{k,0} & \Sigma_{k,0;k+1,0} \\ * & \Sigma_{k+1,0} \end{bmatrix} \right) \tag{44}$$

where the $*$ indicates to the symmetric element.

We assume that the two poses are highly correlated, with a correlation coefficient $\rho \in [-1, 1]$, (we chose $\rho = 0.99$). Then,

$$\Sigma_{k,0;k+1,0} = \rho \left(\Sigma_{k,0} \Sigma_{k+1,0}^T \right)^{1/2} \tag{45}$$

Then, using Lemma 3, the estimated relative transform is

$$\widehat{T}_{B_{k+1}}^{B_k} = (\widehat{T}_{B_k}^{B_0})^{-1} \widehat{T}_{B_{k+1}}^{B_0} \tag{46}$$

and the estimated relative covariance is

$$\Sigma_{B_{k+1}}^{B_k} = A \Sigma_{B_k}^{B_0} A^T + \Sigma_{B_{k+1}}^{B_0} - A \Sigma_\times - \Sigma_\times^T A^T \tag{47}$$

where

$$\Sigma_\times = \rho \left(\Sigma_{B_k}^{B_0} (\Sigma_{B_{k+1}}^{B_0})^T \right)^{1/2}, \quad A = \mathrm{Ad}_{(\widehat{T}_{B_{k+1}}^{B_k})^{-1}} .$$

Note, the adjoint matrix for $T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in \mathbb{SE}(3)$ is

$$\mathrm{Ad}_T = \begin{bmatrix} R & [t]_\times R \\ 0 & R \end{bmatrix}$$

and $\mathrm{Ad}_{T^{-1}} = (\mathrm{Ad}_T)^{-1}$ [34].

E. Replica Dataset Environment Details

Table IV shows the size and volume of the bounding box for each environment used in the simulation studies. It also shows the number of mesh points in the environment.

TABLE IV
SIZE AND VOLUME OF EACH ENVIRONMENT USED.

Env.	Length X (m)	Length Y (m)	Length Z (m)	Bounding Box Volume (m^3)	Number of Mesh Points
`office0`	4.40	5.01	2.99	65.95	589 517
`office1`	4.81	4.11	2.80	55.24	423 007
`office2`	6.47	8.14	2.77	145.89	858 623
`office3`	8.64	9.20	3.10	246.85	1 187 140
`office4`	6.55	6.51	2.82	119.96	993 008
`room0`	7.76	4.70	2.81	102.43	954 492
`room1`	6.65	5.73	2.75	104.81	645 512
`room2`	6.77	4.95	3.59	120.34	722 496

F. Additional Simulation Results

Table V and Table VI show additional results of the performance of the SFC and ESDF methods on the Replica dataset. Here we show the results from a trajectory perturbed by $\Sigma =$1e-5I and $\Sigma =$1e-6I.

TABLE V

Results of the three Safe Flight Corridor (SFC) methods on the Replica dataset. Each environment was run with $\Sigma = \sigma^2 I$ for two different σ^2 values, 1e-5 and 1e-6.

		Violation Rate (%)			Max Violation (mm)			SFC Volume (m^3)		
Env	σ^2	Baseline	Heuristic	Certified	Baseline	Heuristic	Certified	Baseline	Heuristic	Certified
office0	1e-6	18.6%	0.1%	0.0%	102.74	22.05	0.03	34.8	6.7	5.7
	1e-5	32.5%	0.6%	0.0%	397.89	33.83	0.03	38.9	6.8	4.8
office1	1e-6	12.8%	0.6%	0.0%	95.30	14.48	0.86	17.6	3.6	2.6
	1e-5	12.9%	0.1%	0.0%	373.39	24.65	0.86	17.7	3.7	2.0
office2	1e-6	10.1%	0.1%	0.0%	159.66	18.42	0.39	40.8	4.3	3.6
	1e-5	21.3%	0.9%	0.0%	299.11	21.93	0.39	44.9	4.3	3.0
office3	1e-6	12.7%	0.1%	0.0%	177.65	11.61	0.88	56.6	4.6	3.0
	1e-5	16.5%	0.0%	0.0%	460.25	7.38	0.94	57.9	4.6	0.9
office4	1e-6	14.4%	0.3%	0.0%	125.48	8.91	1.69	63.3	15.7	12.5
	1e-5	24.6%	4.7%	0.0%	262.23	82.75	1.69	66.5	16.1	10.6
room0	1e-6	10.7%	0.0%	0.0%	117.12	11.02	0.95	53.0	12.3	9.1
	1e-5	20.1%	0.5%	0.0%	396.74	47.97	0.95	55.8	12.3	8.0
room1	1e-6	19.2%	0.4%	0.0%	191.43	14.20	0.71	38.7	6.9	5.8
	1e-5	25.7%	1.1%	0.0%	377.01	23.68	0.71	39.5	6.7	5.3
room2	1e-6	6.8%	0.9%	0.0%	85.02	12.85	0.65	29.4	7.5	4.4
	1e-5	11.1%	1.5%	0.0%	322.36	25.63	0.65	30.1	7.5	1.8

TABLE VI

Results of the three Euclidean Signed Distance Field (ESDF) methods on the Replica dataset.

		Violation Rate (%)			Max Violation (mm)			ESDF Volume (m^3)		
Env	σ^2	Baseline	Heuristic	Certified	Baseline	Heuristic	Certified	Baseline	Heuristic	Certified
office0	1e-6	48.1%	31.6%	0.5%	604.3	563.6	109.5	46.1	39.5	10.7
	1e-5	21.2%	11.8%	0.5%	384.2	322.5	107.7	42.3	38.1	10.9
office1	1e-6	35.3%	34.4%	0.1%	406.9	379.5	82.5	23.2	23.0	3.8
	1e-5	11.1%	10.6%	0.3%	172.0	172.0	93.8	21.9	21.8	4.2
office2	1e-6	51.5%	7.6%	0.1%	520.0	311.8	141.4	77.5	31.3	6.2
	1e-5	23.8%	2.0%	0.1%	212.6	253.8	100.0	68.7	31.1	6.2
office3	1e-6	54.7%	4.7%	0.0%	671.1	429.4	100.0	110.9	42.0	5.0
	1e-5	28.2%	1.5%	0.0%	330.5	226.3	72.1	96.9	41.4	6.0
office4	1e-6	48.3%	10.1%	0.1%	636.9	366.6	66.3	99.7	51.5	14.3
	1e-5	21.0%	3.9%	0.1%	260.0	215.4	69.3	90.9	50.9	14.4
room0	1e-6	62.0%	9.2%	2.4%	990.8	428.5	120.0	105.4	28.6	31.5
	1e-5	34.4%	3.2%	3.2%	335.3	244.1	164.9	90.9	27.6	32.9
room1	1e-6	48.1%	20.9%	0.0%	604.6	384.7	100.0	53.8	34.5	6.6
	1e-5	17.5%	8.8%	0.0%	240.0	169.7	72.1	47.6	33.1	6.9
room2	1e-6	47.5%	16.3%	0.1%	594.0	435.4	82.5	63.6	38.7	4.5
	1e-5	21.9%	5.1%	0.0%	291.9	200.0	66.3	56.8	37.8	9.5

G. *Effect of Odometry Covariance*

TABLE VII
PERFORMANCE OF THE THREE EUCLIDEAN SIGNED DISTANCE FIELD (ESDF) METHODS IN THE OFFICE0 ENVIRONMENT UNDER VARYING ODOMETRY COVARIANCE AT $\kappa = 3$.

	Violation Rate (%)			Max Violation (mm)			ESDF Volume (m^3)		
σ^2	Baseline	Heuristic	Certified	Baseline	Heuristic	Certified	Baseline	Heuristic	Certified
1e-04	61.09%	43.46%	0.19%	1.24	1.24	0.17	63.87	45.37	10.68
1e-05	48.15%	31.55%	0.50%	0.60	0.56	0.11	46.14	39.46	10.74
1e-06	21.16%	11.79%	0.49%	0.38	0.32	0.09	42.33	38.11	10.94
1e-07	5.17%	2.75%	0.49%	0.22	0.19	0.10	41.79	37.93	11.59
1e-08	2.04%	1.72%	0.54%	0.18	0.13	0.13	41.70	37.86	16.54
1e-09	1.94%	1.54%	0.62%	0.18	0.12	0.16	41.69	37.85	31.52
1e-10	1.91%	1.51%	0.77%	0.18	0.11	0.16	41.69	37.85	37.18
1e-11	1.93%	1.53%	1.21%	0.18	0.11	0.16	41.69	37.86	41.20
1e-12	1.93%	1.53%	1.35%	0.18	0.11	0.16	41.69	37.86	41.59

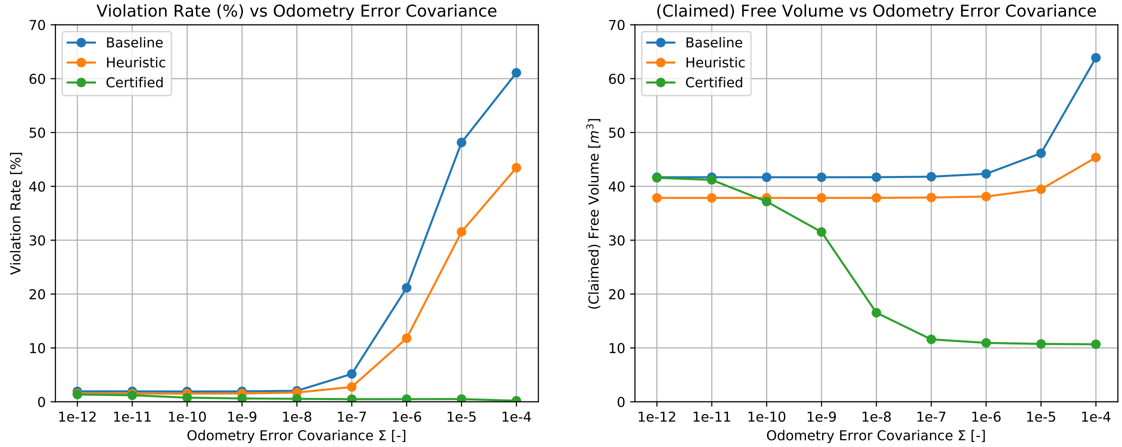

Fig. 9. (Left) Effect of the odometry covariance on the mapping violation rate. (Right) Effect of the odometry covariance on the claimed free volume. Notice that the true free volume is approximately 42 m^3, and in the uncertified methods the volume of claimed free space incorrectly increases beyond 42 m^3. In contrast, in the certified methods the volume decreases to reflect the increased uncertainty.