2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Tightly-Coupled LiDAR-Visual-Inertial SLAM and Large-Scale Volumetric Occupancy Mapping

Simon Boche[1], Sebastián Barbas Laina[1], Stefan Leutenegger[1,2,3]

***Abstract*— Autonomous navigation is one of the key requirements for every potential application of mobile robots in the real-world. Besides high-accuracy state estimation, a suitable and globally consistent representation of the 3D environment is indispensable. We present a fully tightly-coupled LiDAR-Visual-Inertial SLAM system and 3D mapping framework applying local submapping strategies to achieve scalability to large-scale environments. A novel and correspondence-free, inherently probabilistic, formulation of LiDAR residuals is introduced, expressed only in terms of the occupancy fields and its respective gradients. These residuals can be added to a factor graph optimisation problem, either as frame-to-map factors for the live estimates or as map-to-map factors aligning the submaps with respect to one another. Experimental validation demonstrates that the approach achieves state-of-the-art pose accuracy and furthermore produces globally consistent volumetric occupancy submaps which can be directly used in downstream tasks such as navigation or exploration.**

I. Introduction

Robust and accurate state estimation is one of the fundamental components for autonomous navigation of robotic systems. But positioning is only part of the problem. An accurate and especially globally consistent representation of the 3D environment that the robot is operating in, is also indispensable. Simultaneous Localisation and Mapping (SLAM) approaches fusing multiple sensor sources, such as stereo vision, Inertial Measurement Units (IMU) or Light Detection and Ranging (LiDAR) sensors, have proven to achieve accurate performance in localisation. Robustness is gained by fusing complementary sensors to overcome degrading scenarios for each of the individual sensors. Most state-of-the-art LiDAR-Visual-Inertial (LVI) SLAM systems are representing the 3D world in terms of features or point clouds, e.g. [1], [2], [3], [4]. Lately, also surfels have been adopted to LiDAR-based SLAM systems [5]. While these representations may prove suitable for state estimation and surface reconstruction, they cannot be directly used for downstream tasks such as robotic path planning, where an explicit representation of observed free space is desirable.

That is why, lately, there has been research in coupling SLAM and volumetric mapping. To account for potential pose drift, which would lead to degrading map accuracy and consistency, submapping approaches have recently gained interest. The idea behind that is to divide the environment into several local submaps, which will stay accurate and consistent due to locally limited drift. Different strategies on how to create these submaps and to keep global consistency across submaps have been investigated, e.g. in [6] or [7].

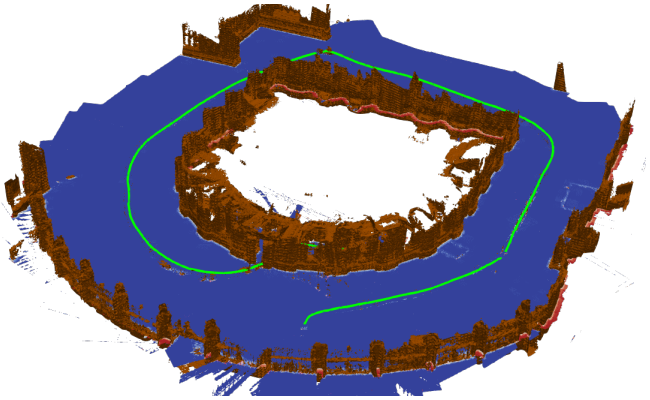

Fig. 1: Horizontal slices of the occupancy fields and 3D reconstructions from Sequence *Exp21* of the HILTI SLAM Challenge [8]. All submaps are overlayed. Meshes of the surfaces (brown) are extracted from the occupancy field as zero-crossings. The blue area denotes free space extracted for gravity-aligned slices through the submaps. The estimated trajectory is shown in green.

Most existing methods decouple the problems of SLAM and volumetric mapping and rearrange the relative position and orientation of all submaps with respect to each other in a loosely-coupled optimisation problem. In contrast, we aim to formulate a whole unified tightly-coupled problem to keep our 3D map representations consistent at all times which makes them suitable for usage in robotic applications. In short, the key contributions of this work are the following.

- We present a tightly-coupled optimisation-based LVI-SLAM and occupancy mapping framework leveraging local submaps to ensure global consistency.
- We introduce a novel residual formulation for LiDAR-based error terms that directly operates on the occupancy values and field gradients. Both can be efficiently queried from the submaps without an expensive data association step as for example in Iterative Closest Point (ICP) approaches. These residuals can be added to the factor graph in two ways, either as factors for every live frame or as factors between submaps.
- We evaluate the approach quantitatively in terms of localisation accuracy and qualitatively regarding map quality. We demonstrate that our system yields globally consistent maps for different LiDAR sensors while achieving state-of-the-art localisation performance.

This work was supported by the Technical University of Munich, the TUM Innovation Network CoConstruct and Leica Geosystems AG.

[1]Smart Robotics Lab, School of Computation, Information and Technology (CIT), Technical University of Munich, Germany. firstname.surname@tum.de

[2] Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich, Germany

[3] Department of Computing, Imperial College London, UK

The remainder of this paper is structured as follows: after a brief summary of related works in Sec. II, Sec. III presents the problem statement and notations. Sec. IV sketches the overall structure of the system before the mapping and estimation approaches are described in Sec. V and VI. An experimental evaluation is given in Sec. VII.

II. Related Work

One of the first breakthroughs and still influential works in the field of LiDAR-based SLAM methods was LOAM [9]. Its idea was the extraction of geometric features such as edges and planes, and matching them across frames. This basic concept has been widely adopted in many following works. While LOAM was also able to fuse IMU measurements in a loosely-coupled way, LIO-SAM [10] formulates a tightly-coupled pose graph optimisation fusing edge and plane error residuals with IMU error residuals. Also filter-based approaches, such as FAST-LIO [11] or FAST-LIO2 [12] have successfully demonstrated high-accuracy localisation fusing LiDAR and IMU. While FAST-LIO still uses also plane and edge features, FAST-LIO2 achieves significant speed-up by directly operating on the raw points. [13] establishes sliding-window bundle adjustment (BA) for LiDAR scans.

Recently, LiDAR-Visual-Inertial SLAM systems have become increasingly popular. A large amount of these approaches makes use of LOAM's idea of geometric feature extraction. LVI-SAM [3] combines two subsystems, a Visual-Inertial (VI) and a LiDAR-Inertial (LI), in a tightly-coupled way to complement each other in challenging scenarios. The LI system builds a factor graph based on IMU pre-integration errors and edge and plane residuals. VILENS [1] also builds an optimisation problem consisting of visual, inertial, leg odometry, and LiDAR-based line and plane residuals. Another line of research uses Multi-State Constraint Kalman Filter (MSCKF [14]) based approaches for tightly-coupled, filter-based fusion of visual, inertial and LiDAR measurements. Examples are [15] and [16].

To minimise drift in long-term scenarios, the latest research commonly applies the concept of submapping. This originates from early SLAM research, such as the Atlas framework [17]. In this context, additional factors can be derived to align submaps and to eliminate drift. [1] uses local point-cloud submaps and adds ICP odometry measurements into the factor graph. Wildcat [5], a sliding-window optimisation-based LiDAR-Inertial Odometry system, achieves peak state-of-the-art robustness and accuracy by building local surfel submaps and aligning the submaps in a pose graph optimisation. Most of the aforementioned work uses feature-based or surface-based 3D representations. While they might be sufficient for high-accuracy localisation and 3D reconstruction, they are not suitable for navigation due to their lack of representing observed free space.

Submapping on volumetric maps has been addressed in various works. [18] builds occupancy submaps based on OctoMap [19] and aligns them by standard ICP registration. Voxgraph [6] uses a Visual-Inertial Odometry to provide poses for integration into TSDF maps. Upon completion, ESDF fields are generated and submaps are aligned in a back-end pose graph optimisation using correspondence-free error terms based on ESDF values. New submaps are created at a fixed frequency. Wang *et al.* [20] instead use occupancy maps as their 3D representation. Using Supereight2 [21] as adaptive-resolution mapping approach, new submaps are spawned based on the distance travelled. Submaps are re-arranged based on updates from the visual-inertial estimator. The follow-up work [7] improved the submap creation by evaluating the point cloud overlaps of new scans and alignment of submaps is based on ICP. In this work, we will adopt the concept of submapping. In contrast to [6], [20], [7], the global alignment of submaps is not decoupled from the estimator but provides direct feedback. We formulate correspondence-free residuals as in [6] without the expensive need to extract ESDFs as we directly use the available occupancy information.

III. Preliminaries

A. Problem Statement

In this work, we aim to build an accurate and globally consistent volumetric representation of the environment around a mobile robot, which can be used in downstream tasks, most prominently navigation. We do so by taking a Visual-Inertial SLAM System, OKVIS2 [22], as our starting point, as well as an adaptation of the adaptive-resolution occupancy mapping framework Supereight2 [21]. As VI-SLAM tends to be locally consistent but suffers from the accumulation of larger drift over time, we tackle the problem of degrading map quality by applying submapping strategies. Furthermore, to increase the accuracy of the estimator while simultaneously ensuring global consistency of overlapping maps, we integrate LiDAR constraints based on frame-to-map as well as map-to-map alignment factors into the state estimator in a tightly-coupled way.

B. Notation

Throughout this work, the following notation will be used: coordinate frames are written as $\underrightarrow{\mathcal{F}}_A$ and a vector expressed in this reference frame will be denoted as ${}_A\mathbf{r}$. The rigid body transformation which transforms points from a reference frame $\underrightarrow{\mathcal{F}}_B$ to another reference frame $\underrightarrow{\mathcal{F}}_A$ is given by $\boldsymbol{T}_{AB} \in SE(3)$ and can be decomposed into a rotation matrix $\mathbf{C}_{AB} \in SO(3)$ and the translational component ${}_A\mathbf{r}_B$. We also denote the rotation $\mathbf{C}_{AB}$ with its unit quaternion form $\mathbf{q}_{AB}$. The most important reference frames that will be used are: a fixed world reference frame $\underrightarrow{\mathcal{F}}_W$, the camera coordinate frames $\underrightarrow{\mathcal{F}}_{C_i}$ for $i = 1 \dots N$ cameras, the IMU sensor frame $\underrightarrow{\mathcal{F}}_S$, the LiDAR sensor frame $\underrightarrow{\mathcal{F}}_L$ and a map frame $\underrightarrow{\mathcal{F}}_M$. Furthermore, $[\cdot]^\times$ represents the skew-symmetric matrix of a 3D vector.

IV. System Overview

In this work, we build a fully tightly-coupled system combining volumetric mapping and LiDAR-Visual-Inertial (LVI) SLAM. A high-level overview is sketched in Fig. 2. Inputs considered are IMU measurements, stereo camera frames and

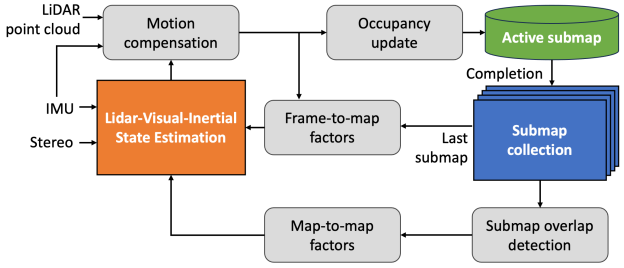

Fig. 2: High-level structure of the proposed tightly-coupled LVI-SLAM and submapping framework. Motion compensation for LiDAR point clouds is performed to formulate live frame-to-map factors and to update the current active submap. Upon submap completion, the most overlapping previous submap is determined and map-to-map factors are fused into the optimisation problem.

potentially unordered LiDAR point clouds. By the use of state estimates and IMU measurements integrated to a point's individual timestamp, motion compensation is performed to account for dynamic movement of the LiDAR sensor. After transforming the measurements to the respective map frame, the occupancy field of the current active submap is updated. Tight coupling between the submapping module and the LiDAR-Visual-Inertial estimator is achieved in two ways. Live frame-to-map factors are formulated for the state corresponding to the most recent stereo frame. Furthermore, upon completion of the active submap, the most overlapping submap is detected and map-to-map factors are added to the LVI state estimator. These factors as well as details regarding the submapping strategies will be given in the following sections.

V. Occupancy Mapping and Submapping

As a volumetric occupancy mapping approach, we adopted the octree-based multi-resolution volumetric mapping framework Supereight2 [21] specifically for the usage with different kinds of LiDAR sensors.

A. Occupancy Mapping for dynamical LiDAR sensors

In contrast to [20], we do not use range image projections of LiDAR point clouds as input to our mapping system. Instead, to account for dynamically moving sensors, we added a ray-based integration interface for Supereight2 that

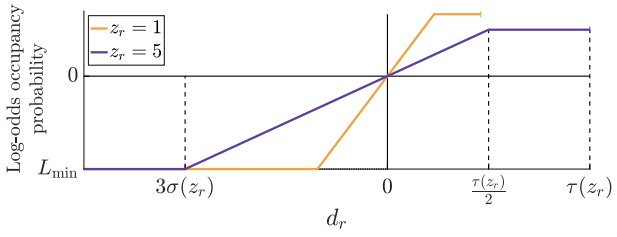

Fig. 3: Inverse sensor model used in Supereight2 [21] for two measurements (1m, 5m). The log-odds occupancy probability along a ray measurement is expressed as a function of the difference d_r between query points along the ray and the measured distance z_r. The occupancy values are clipped in front of the surface at a minimum $L_{\min}$ reached at 3σ where σ is a distance dependent uncertainty value. It grows linearly up to half the surface thickness $\tau(z_r)$. For more details, see [21].

fuses measurements on a per-ray-basis. We use fairly standard occupancy mapping in log-odds space with a simplified inverse sensor model that approximates the probabilistic nature of the sensor in terms of range or depth uncertainty and outlier probabilities. The log-odds occupancy $l^k\left({}_M\mathbf{p}\right)$ of a 3D point ${}_M\mathbf{p}$ in a map frame $\underrightarrow{\mathcal{F}}_M$ at step k is given by

$$l^k\left({}_M\mathbf{p}\right) = \log \frac{P_{\text{occ}}\left({}_M\mathbf{p}|z_r\right)}{1 - P_{\text{occ}}\left({}_M\mathbf{p}|z_r\right)}, \tag{1}$$

where z_r is the measured distance of a single ray and l^k is following a piece-wise linear function along the ray as shown in Fig. 3.

Using clamped weights, we apply the same additive Bayesian updates as in [21]:

$$\begin{aligned} \bar{L}^k\left({}_M\mathbf{p}\right) &= \frac{\bar{L}^{k-1}\left({}_M\mathbf{p}\right) w^{k-1} + l^k\left({}_M\mathbf{p}\right)}{w^{k-1} + 1} \\ w_k &= \min\left\{w_{k-1} + 1, w_{\max}\right\} \end{aligned} \tag{2}$$

The accumulated log-odds can still be preserved as

$$L^k\left({}_M\mathbf{p}\right) = \bar{L}^k\left({}_M\mathbf{p}\right) w_k. \tag{3}$$

To keep the octree representation consistent and the resolution as coarse as possible, similar up-propagation and tree pruning strategies as in [21] are applied.

B. Submapping Strategy

Similar to [7], our goal is to leverage the available sensor information as a decision criterion when to spawn a new submap: we evaluate per-frame the ratio between incoming measurements that correspond to already observed space and the total number of measurements. If this ratio falls below a certain threshold λ_{overlap}, a new submap will be created with the next visual keyframe generated as in [22]. Every submap is anchored in the IMU frame $\underrightarrow{\mathcal{F}}_S$ at the time step k of the corresponding keyframe. All submaps are stored together with their respective keyframe poses $\boldsymbol{T}_{WS^k}$.

VI. LiDAR-Visual-Inertial Estimator

In the following, we present the extension of the OKVIS2 [22] factor graph.

A. Factor Graph and Optimisation Problem

The factor graph representation used in this work is shown in Fig. 4. On top of visual, inertial and pose graph factors, that are already used in OKVIS2, two different types of LiDAR factors are added to the graph. For every state in the real-time optimisation window, LiDAR factors with respect to the last completed submap are added. Furthermore, map-to-map LiDAR factors are added to constrain submaps with respect to each other by aggregating measurements between submaps. The state vector that is estimated remains:

$$\mathbf{x} = \left[{}_W\mathbf{r}_S^T, \mathbf{q}_{WS}{}^T, {}_W\mathbf{v}^T, \mathbf{b}_{\text{g}}^T, \mathbf{b}_{\text{a}}^T\right]^T, \tag{4}$$

where ${}_W\mathbf{r}_S$ and $\mathbf{q}_{WS}$ denote the position and orientation of the IMU sensor frame in the fixed world frame, and ${}_W\mathbf{v}$ describes the velocity of the IMU with respect to the world frame. $\mathbf{b}_{\text{g}}$ and $\mathbf{b}_{\text{a}}$ stand for gyroscope and accelerometer biases, respectively.

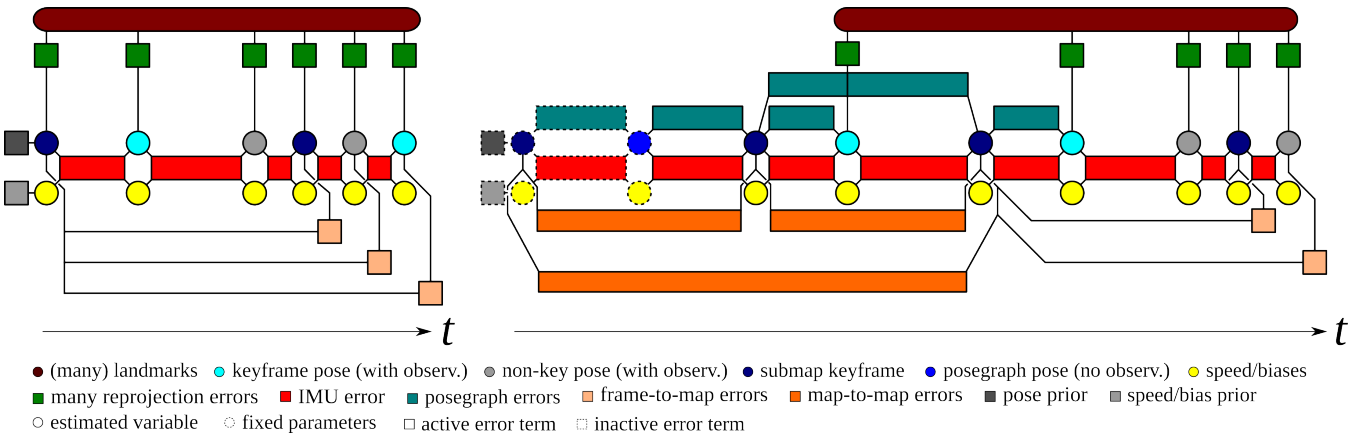

Fig. 4: Optimisation Factor Graph of the LiDAR-Visual-Inertial Estimator. Left: The real-time estimator connects set of current keyframe states and non-keyframe states by IMU errors and visual reprojection errors. It also shows the active submap keyframe and the last completed submap keyframe. For every state in the optimisation window, we can formulate live LiDAR factors between every live frame and the last completed submap. Right: OKVIS2 connects keyframe poses through relative pose errors at a later stage. In addition to the live LiDAR factors, measurements between frames can be aggregated and map-to-map LiDAR factors can be added to the factor graph. Every map will be connected to the previous submap and optionally an older submap if the geometric overlap surpasses a threshold.

B. Error Residuals

Here, for convenience, a summary of the original OKVIS2 factors is given before deriving the new LiDAR factors.

1) Visual Reprojection Errors: OKVIS2 [22] uses 2D reprojection errors $\mathbf{e}_{\mathrm{r}}^{i,j,k}$ of the j-th landmark in the frame of the i-th camera at a timestamp k and its corresponding observation $\tilde{\mathbf{z}}_{\mathrm{r}}^{i,j,k}$ in the image is given by

$$\mathbf{e}_{\mathrm{r}}^{i,j,k} = \tilde{\mathbf{z}}_{\mathrm{r}}^{i,j,k} - \mathbf{h}\left(\boldsymbol{T}_{SC_i}^{-1}\boldsymbol{T}_{S^kW}\mathbf{l}^j\right). \tag{5}$$

Hereby, $\mathbf{h}(\cdot)$ denotes the camera projection.

2) IMU Errors: For the formulation of the IMU residuals, OKVIS2 adopts the IMU preintegration approach in [23]. Between time steps k and n, the IMU error is:

$$\mathbf{e}_{\mathrm{s}}^{k} = \tilde{\mathbf{x}}^n\left(\mathbf{x}^k, \tilde{\mathbf{z}}_{\mathrm{s}}^{k,n}\right) \boxminus \mathbf{x}^n, \tag{6}$$

where $\tilde{\mathbf{x}}^n$ is the predicted state at an arbitrary time n as a function of the current state $\mathbf{x}^k$ and IMU measurements $\tilde{\mathbf{z}}_{\mathrm{s}}^{k,n}$. The $\boxminus$ performs regular subtraction except for the quaternion (see [22]).

3) Relative Pose Errors: Additionally, relative pose errors $\mathbf{e}_{\mathrm{p}}^{r,c}$ between time steps r and c are given by

$$\mathbf{e}_{\mathrm{p}}^{r,c} = \mathbf{e}_{\mathrm{p},0}^{r,c} + \begin{bmatrix} {}_{S^r}\mathbf{r}_{S^c} - {}_{S^r}\tilde{\mathbf{r}}_{S^c} \\ \mathbf{q}_{S^rS^c} \boxminus \tilde{\mathbf{q}}_{S^rS^c} \end{bmatrix}. \tag{7}$$

with ${}_{S^r}\tilde{\mathbf{r}}_{S^c}$ and $\tilde{\mathbf{q}}_{S^rS^c}$ being nominal relative position and orientations expressed in the IMU frame $\underrightarrow{\mathcal{F}}_S$. We refer the reader to [22] for a detailed derivation of the constant $\mathbf{e}_{\mathrm{p},0}^{r,c}$ as well as error Jacobians and error weights $\mathbf{W}_{\mathrm{p}}^{r,c}$.

4) Submap-based LiDAR Errors: As stated in the previous section, there are two types of LiDAR residuals. In both cases, the error residuals are formulated based on two states. On the one hand, for live residuals, these two states are given by the current frame and the keyframe associated to the last completed submap. On the other hand, for map-to-map residuals, both states will be submap anchor frames. Regarding the formulation of the error residuals, however, they do not differ.

Given a completed submap anchored at keyframe pose $\boldsymbol{T}_{WS^a}$ and a point cloud $\mathcal{P}_b$ associated to another state $\boldsymbol{T}_{WS^b}$, we formulate the residual for every ${}_{S^b}\mathbf{p} \in \mathcal{P}_b$ as:

$$e_{\mathrm{l}}^{a,b}\left({}_{S^a}\mathbf{p}\right) = \frac{d}{\sigma} = \frac{L\left({}_{S^a}\mathbf{p}\right)}{\sqrt{\frac{L_{\min}^2}{9} + \sigma_z^2\left|\nabla L\left({}_{S^a}\mathbf{p}\right)\right|^2}}, \tag{8}$$

where ${}_{S^a}\mathbf{p} = \boldsymbol{T}_{S^aS^b}\,{}_{S^b}\mathbf{p}$ with $\boldsymbol{T}_{S^aS^b} = \boldsymbol{T}_{WS^a}{}^{-1}\boldsymbol{T}_{WS^b}$. The idea here is that every measured point should be on a surface in the 3D map; and the distance d of the point from the nearest surface can be extrapolated from the occupancy value $L(\cdot)$ and the occupancy gradient $\nabla L(\cdot)$ assuming a linear behavior as in the sensor model for mapping. From the model (Fig. 3), we can derive the distance d and the map uncertainty as:

$$d = \frac{L}{|\nabla L|}, \qquad \sigma_{\mathrm{map}} = \frac{L_{\min}}{3|\nabla L|}. \tag{9}$$

$L_{\min}$ is a configuration parameter and denotes the saturation minimum log-odds occupancy value. With the sensor-specific measurement uncertainty σ_z, we can formulate the weighted residual in Eqn. (8) using the total uncertainty

$$\sigma = \sqrt{\sigma_{\mathrm{map}}^2 + \sigma_z^2}. \tag{10}$$

Furthermore, we want to derive analytic Jacobians, whereby using a perturbation $\delta\boldsymbol{\chi}_T = [\delta\mathbf{r}, \delta\boldsymbol{\alpha}]$ for poses around linearisation points $\bar{\mathbf{r}}$ and $\bar{\mathbf{C}}$:

$$\begin{aligned} \mathbf{r} &= \bar{\mathbf{r}} + \delta\mathbf{r}, \\ \mathbf{C} &= \mathrm{Exp}\left(\delta\boldsymbol{\alpha}\right)\bar{\mathbf{C}}. \end{aligned} \tag{11}$$

We further define the error state as $\delta\boldsymbol{\chi} = [\delta\boldsymbol{\chi}_{T_{WS}}, \delta\boldsymbol{\chi}_{\mathrm{sb}}]$, with $\delta\boldsymbol{\chi}_{\mathrm{sb}}$ denoting an additive perturbation of the speed and biases. With the LiDAR residuals only depending on pose states, but not on speed and biases, we can compute Jacobians with respect to the poses of frames $\underrightarrow{\mathcal{F}}_{S^a}$ and $\underrightarrow{\mathcal{F}}_{S^b}$ leveraging the chain rule:

$$\frac{\partial e_{\mathrm{l}}^{a,b}}{\delta\boldsymbol{\chi}_{T_{WS^m}}} = \frac{\partial e_{\mathrm{l}}^{a,b}}{\partial_{S^a}\mathbf{p}}\frac{\partial_{S^a}\mathbf{p}}{\delta\boldsymbol{\chi}_{T_{WS^m}}}, \tag{12}$$

with $m \in \{a, b\}$. The individual Jacobians in Eq. (12) are:

$$\begin{aligned}
\frac{\partial e_{\mathrm{l}}^{a,b}}{\partial_{S^a}\mathbf{p}} &= \frac{\nabla L}{\sqrt{\frac{L_{\min}^2}{9} + |\nabla L|^2 \sigma_z^2}} \\
\frac{\partial_{S^a}\mathbf{p}}{\delta \boldsymbol{\chi}_{T_{WS^a}}} &= \mathbf{C}_{S^aW} \begin{bmatrix} -\mathbf{I}_3 & \left[\mathbf{C}_{WS^b}{}_{S^b}\mathbf{p} + {}_W\mathbf{r}_{S^b} - {}_W\mathbf{r}_{S^a}\right]^\times \end{bmatrix} \\
\frac{\partial_{S^a}\mathbf{p}}{\delta \boldsymbol{\chi}_{T_{WS^b}}} &= \mathbf{C}_{S^aW} \begin{bmatrix} \mathbf{I}_3 & -\left[\mathbf{C}_{WS^b}{}_{S^b}\mathbf{p}\right]^\times \end{bmatrix}. \qquad (13)
\end{aligned}$$

Note that the overall idea of this optimisation resembles a point-to-plane ICP. In contrast to the usual point-to-plane ICP, the expensive step of data association and normal computation can be omitted as occupancy values and gradients can be directly retrieved from the occupancy field. Furthermore, outliers are implicitly covered as they will most likely lie in either free or unobserved space with zero or invalid gradients.

5) Optimisation Problem: All of the aforementioned factors are combined in the overall minimisation objective:

$$\begin{aligned}
c(\mathbf{x}) = &\frac{1}{2}\sum_i \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}(i,k)} \rho\left(\mathbf{e}_{\mathrm{r}}^{i,j,k\,T}\mathbf{W}_{\mathrm{r}}\mathbf{e}_{\mathrm{r}}^{i,j,k}\right) \\
&+ \frac{1}{2}\sum_{k\in\mathcal{P}\cup\mathcal{K}\setminus f} \mathbf{e}_{\mathrm{s}}^{k\,T}\mathbf{W}_{\mathrm{s}}^{k}\mathbf{e}_{\mathrm{s}}^{k} + \frac{1}{2}\sum_{r\in\mathcal{P}}\sum_{c\in\mathcal{C}(r)} \mathbf{e}_{\mathrm{p}}^{r,c\,T}\mathbf{W}_{\mathrm{r}}^{r,c}\mathbf{e}_{\mathrm{p}}^{r,c} \\
&+ \frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{\mathbf{p}\in\mathcal{L}_k} e_{\mathrm{l}}^{C,k\,2} + \frac{1}{2}\sum_{b\in\mathcal{M}}\sum_{a\in\mathcal{A}_b}\sum_{\mathbf{p}\in\mathcal{L}_b} e_{\mathrm{l}}^{a,b\,2}. \qquad (14)
\end{aligned}$$

Here, the set $\mathcal{K}$ contains the most recent frames as well as keyframes with observations of visible landmarks in $\mathcal{J}(i,k)$. $\mathcal{P}$ contains all pose graph frames and f denotes the most current frame. $\mathcal{C}(r) \subset \mathcal{P}$ is the set of all pose graph frames connected to a frame r. Furthermore, the set $\mathcal{L}_k$ denotes the set of all LiDAR measurements associated to a frame k. $\mathcal{M}$ is the set of all past submaps, and $\mathcal{A}_b$ the set of all submap frames connected to a submap frame $\boldsymbol{T}_{WS^b}$ via map-to-map residuals. C denotes the last completed submap frame.

VII. Experimental Results

We will quantitatively evaluate the accuracy of the proposed tightly-coupled LiDAR-Visual-Inertial SLAM system in Section VII-B. In section VII-C, we will qualitatively demonstrate that the proposed system yields globally consistent submaps that can be used in robotic applications. Finally, in Section VII-D, we will briefly analyse the computational performance of the proposed system.

A. Implementation Details

All experiments were performed on an Intel i7-11700K CPU with 3.6 GHz and 64 GB RAM. Google's non-linear least squares optimisation framework Ceres [25] has been used with an implementation of analytical Jacobians. As a threshold λ_{overlap} for new submap creation, a value of 0.4 was chosen. For every live state, we added 100 frame-to-map residuals. Upon completion of submaps, 1000 map-to-map residuals were added. For that, points were randomly sampled from the aggregated point clouds between the two corresponding frames. Submaps have a maximum dimension of 15.36 m and a finest resolution of 3 cm.

B. HILTI Slam Challenge

We evaluated our approach on the HILTI 2022 SLAM Challenge benchmark [8]. The Hilti-Oxford dataset was collected using a handheld sensor device consisting of an Alphasense five-camera module, an IMU and a 32 beam Hesai PandarXT-32 LiDAR sensor. With the available calibration datasets, we ran our own sensor calibration of the multi-camera system and IMU-camera extrinsics using Kalibr [26]. As we experienced that the performance on the different sequences dependent significantly on the calibration parameters that were used, we also performed online calibration of the IMU-camera extrinsics. The challenge includes 8 different sequences with varying levels of difficulty. Difficulty levels are based on challenging environments (e.g. bad lighting) or aggressive motions of the sensor suite. Out of the 8 sequences, 3 were taken on a construction site (*Exp01*, *Exp02* and *Exp03*), 1 was taken in a long corridor of an office building (*Exp07*) and the remaining 4 sequences were recorded in and around the Oxford Sheldonian Theatre (*Exp09*, *Exp11*, *Exp15* and *Exp21*). After the challenge, 3 additional sequences from the construction site (*Exp04*, *Exp05* and *Exp06*) were added to the automatic evaluation system. For evaluation, sparse ground-truth with millimeter accuracy is provided. An automatic evaluation system is available that uses a score based on the Absolute Trajectory Error (ATE) as an evaluation metric. First, the estimated trajectories are aligned with the sparse ground-truth using $SE(3)$ Umeyama alignment. For every ground-truth control point, the corresponding estimate is retrieved and awarded a score ranging from 0 to 10 (10 for errors below 1 cm, 0 for errors above 10 cm, for more details see [8]). The total score S_j for a dataset j with N control points is computed as

$$S_j = \left(\frac{1}{10N}\sum_{i=0}^{N} s_i\right) \times 100. \qquad (15)$$

A full score of 100 would correspond to a localisation error below 1 cm across the whole trajectory.

Table I shows the final scores of our approach on all challenge sequences with identical parameters. We report the causal as well as non-causal estimates including a final Bundle Adjustment. We compare it to the Visual-Inertial SLAM (including loop closures) performance of OKVIS2 [22] as a baseline. The VI baseline used all 5 cameras and the same visual frontend parameters as in the LVI case. Furthermore, we compare it to other state-of-the-art methods that have been published an evaluated on the benchmark including the – at the time of the challenge – winning algorithm Wildcat [5].

It can be seen that adding LiDAR factors increases the accuracy of OKVIS2 significantly, in the causal as well as the final optimised evaluation. Furthermore, the proposed approach achieves state-of-the-art localisation accuracy. At the moment, a final score of 427.65 achieves rank 7 out of

Approach	C	Sensors L	V	I	Sequence exp01(e)	exp02(m)	exp03(h)	exp04(e)	exp05(e)	exp06(m)	exp07(m)	exp09(h)	exp11(m)	exp15(h)	exp21(e)	Score
OKVIS2 [22]	✓		✓	✓	8.46	8.18	0.00	17.14	23.33	14.29	0.00	0.00	0.00	0.00	0.00	16.64
OKVIS2 [22]			✓	✓	30.77	20.00	0.00	47.14	56.67	34.28	31.67	13.75	28.00	12.22	0.00	136.41
VILENS [1]		✓	✓	✓	69.23	49.09	40.00	-	-	-	16.67	5.00	68.00	17.78	60.00	325.77
HKU*	✓	✓	✓	✓	**84.62**	70.00	44.71	-	-	-	31.67	7.50	**92.00**	24.44	48.00	402.93
Wildcat [5]		✓		✓	**84.62**	**84.55**	**90.59**	-	-	-	**45.00**	**44.38**	84.00	46.67	**84.00**	**563.79**
Ours	✓	✓	✓	✓	43.08	17.27	32.35	57.14	61.67	17.14	0.00	16.25	32.00	51.11	62.00	254.06
Ours		✓	✓	✓	62.31	39.09	42.35	**68.57**	**86.67**	**47.14**	13.33	38.12	**92.00**	**64.44**	76.00	427.65

TABLE I: Evaluation scores on HILTI22 SLAM Challenge. Bold: best score, underlined: second best; C: causal evaluation (definition in [22]); L, V, I: LiDAR, Visual and Inertial measurements are considered; (e/m/h) classifies the difficulty level of the sequence: easy, medium or hard; *based on [24]. Sequences exp04, exp05 and exp06 greyed out as they are not included in the final score.

53 submissions on the challenge leaderboard. Note that also VILENS and Wildcat performed some form of offline batch or posegraph optimisation for the reported scores. For a better understanding, it can be stated that the reported scores here denote a mean ATE of 1 cm (or even below) to 3 cm in the final estimates. Only for *exp07* where a long corridor with a lack of characteristic visual features or geometric features does not achieve the same performance and results in a mean ATE of approximately 7 cm.

C. Mapping

1) HILTI Dataset: Fig. 1 shows an overlay of all submaps created on *Exp21*. Meshes of the surfaces and a slice through the occupancy fields at a height of 0.2 m are shown. It qualitatively demonstrates that we are able to provide accurate 3D reconstructions and sensible occupancy fields immediately usable for robot navigation without discernible inconsistencies between submaps.

2) Real-World Example: To qualitatively demonstrate the approach's versatility to a variety of different LiDAR sensor types, we additionally processed a dataset recorded on a Leica BLK2Fly drone. This dataset has been recorded in the outdoor area around an office building. The drone is equipped with five cameras and also five IMUs, out of which we only use one for state estimation. The LiDAR sensor is a single-beam dual-axis spinning LiDAR sensor. Our mapping approach avoids projection into very sparse range images. Fig. 5 shows a reconstruction as well as the trajectory and a slice through the occupancy field.

D. Timings

We further show timings for map integration and graph optimisation of *Exp15* in Table II. LVI Setup 1 here denotes the original setup as it was used for the evaluation of all sequences in I. In Setup 2, we reduced the number of LiDAR residuals to 50 for live frame-to-map and 500 for map-to-map factors. Furthermore, the input point clouds were downsampled with a factor of 3. The results show that the rate of the optimisation can be increased from approximately 10 Hz to almost 15 Hz. Also the map integration can be sped up significantly by downsampling input point clouds for mapping. Instead of running at 10 Hz, we can run integration at 25 Hz. Note, that this speed-up only leads to a very minor degradation in terms of localisation accuracy.

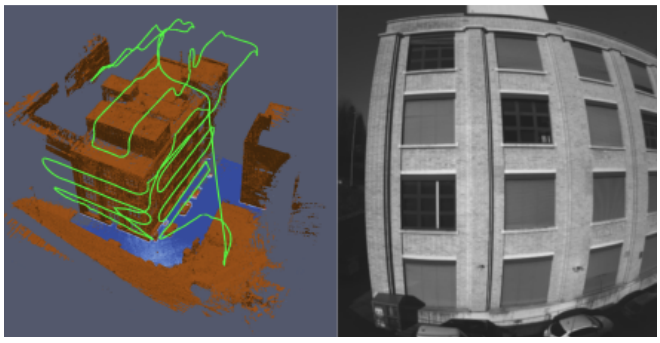

Fig. 5: Outdoor Example: Reconstruction (brown), horizontal slice of the free space (blue) and estimated trajectory (green).

Function	VI only	LVI Setup 1	LVI Setup 2
Optimisation	49.0 ± 16.6	98.9 ± 30.9	79.3 ± 25.0
Batch Integration	-	98.4 ± 83.0	38.2 ± 34.5
ATE (Causal) [cm]	23.5	2.5	3.8

TABLE II: Per-frame average timings in ms and standard deviations for optimisation and map integration on *Exp15*.

VIII. Conclusion

Autonomous navigation in the real-world requires high-accuracy localisation, but also accurate and more importantly a globally consistent 3D representation of the environment. In this paper, we propose a fully tightly-coupled LiDAR-Visual-Inertial system that addresses both of these tasks simultaneously. Results from the state estimator are used to integrate incoming LiDAR measurements into local submaps. A novel, correspondence-free residual formulation has been introduced that only uses occupancy field values and gradients which can be efficiently queried. In a tightly-coupled approach, these LiDAR factors can be added to the optimisation problem either as live frame-to-map constraints or as map-to-map constraints across larger time spans. The proposed system has proven to achieve state-of-the-art performance in localisation while yielding consistent occupancy submaps. In future work, we are planning to improve on several aspects. These include amongst others a more realistic uncertainty model for the LiDAR sensor or an informed way of downsampling incoming LiDAR scans leveraging e.g. geometric characteristics. Another challenge, that we will address in the future is the robustness to challenging scenarios in which the visual frontend is prone to tracking failures. Finally, we are also planning to extend the current work to a full exploration framework navigating through submaps in the real-world.

REFERENCES

[1] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.

[2] J. Lin and F. Zhang, "R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 672–10 678.

[3] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 5692–5698.

[4] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5580–5586.

[5] M. Ramezani, K. Khosoussi, G. Catt, P. Moghadam, J. Williams, P. Borges, F. Pauling, and N. Kottege, "Wildcat: Online continuous-time 3d lidar-inertial slam," *arXiv preprint arXiv:2205.12595*, 2022.

[6] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 227–234, 2019.

[7] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon, "Strategies for large scale elastic and semantic lidar reconstruction," *Robotics and Autonomous Systems*, vol. 155, p. 104185, 2022.

[8] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon, "Hilti-oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 408–415, 2022.

[9] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[10] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[11] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[12] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[13] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[14] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3565–3572.

[15] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidar-inertial-camera odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5848–5854.

[16] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys, "Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5112–5119.

[17] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1899–1906.

[18] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph slam and planning in 3d environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2175–2182.

[19] 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.

[20] Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popović, M. Camurri, S. Leutenegger, and M. Fallon, "Elastic and efficient lidar reconstruction for large-scale exploration tasks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5035–5041.

[21] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger, "Multi-resolution 3d mapping with explicit free space representation for fast and accurate mobile robot motion planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3553–3560, 2021.

[22] S. Leutenegger, "Okvis2: Realtime scalable visual-inertial slam with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[23] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[24] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry," 2022.

[25] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[26] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 473–480.