This paper has been accepted for publication in the Special Issue on Visual SLAM of *IEEE Transactions on Robotics (T-RO)*. The final version will be available via IEEE Xplore.

DOI: 10.1109/TRO.2025.3619051

OKVIS2-X: Open Keyframe-based Visual-Inertial SLAM Configurable with Dense Depth or LiDAR, and GNSS

Simon Boche*, Jaehyung Jung*, Sebastián Barbas Laina*, and Stefan Leutenegger

***Abstract*—To empower mobile robots with usable maps as well as highest state estimation accuracy and robustness, we present OKVIS2-X: a state-of-the-art multi-sensor Simultaneous Localization and Mapping (SLAM) system building dense volumetric occupancy maps, while scalable to large environments and operating in realtime. Our unified SLAM framework seamlessly integrates different sensor modalities: visual, inertial, measured or learned depth, LiDAR and Global Navigation Satellite System (GNSS) measurements. Unlike most state-of-the-art SLAM systems, we advocate using dense volumetric map representations when leveraging depth or range-sensing capabilities. We employ an efficient submapping strategy that allows our system to scale to large environments, showcased in sequences of up to 9 kilometers. OKVIS2-X enhances its accuracy and robustness by tightly-coupling the estimator and submaps through map alignment factors. Our system provides globally consistent maps, directly usable for autonomous navigation. To further improve the accuracy of OKVIS2-X, we also incorporate the option of performing online calibration of camera extrinsics. Our system achieves the highest trajectory accuracy in EuRoC against state-of-the-art alternatives, outperforms all competitors in the Hilti22 VI-only benchmark, while also proving competitive in the LiDAR version, and showcases state of the art accuracy in the diverse and large-scale sequences from the VBR dataset. Code available at: https://github.com/ethz-mrl/OKVIS2-X.**

***Index Terms*—SLAM; Mapping; Localization; Sensor Fusion**

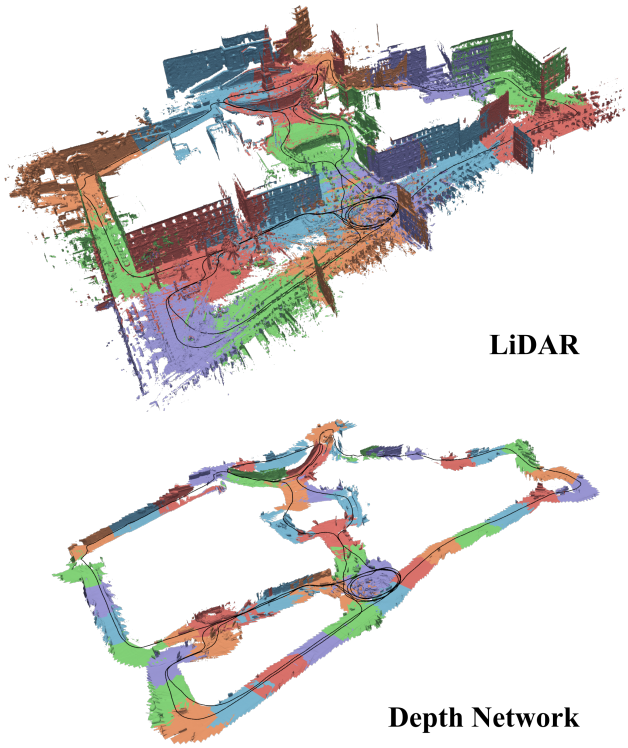

Fig. 1. 3D reconstruction from a run of OKVIS2-X on the `Spagna` sequence of the VBR dataset [1]. Reconstruction with a LiDAR sensor (top) or with a depth network (bottom) to showcase the versatility of the presented system to different sensor modalities. The estimated trajectory is visualized in black. Furthermore, different colors per submap are used.

I. Introduction

While autonomous robots and mobile devices are becoming more ubiquitous, their ability to perceive and spatially relate to their environments remains crucial. To cope with potentially unkown environments, certain such applications demand that the system can simultaneously map the environment while also localizing in it.

Such Simultaneous Localization and Mapping (SLAM) systems typically require heterogeneous sensors for enhanced accuracy and robustness, with visual-inertial configurations being one of the preferred combinations of sensors due to its appealing compromise between cost and accuracy. The setup combines spatial information from the camera with proprioceptive temporal information from the inertial measurement units (IMU), whereby rendering the gravity direction observable when tightly coupled and reducing dependency on brittle visual data associations. For applications that demand a higher mapping and localization accuracy, LiDARs have emerged as a powerful option due to their unmatched depth sensing resolution and accuracy, at the expense of cost. To improve the robustness of SLAM systems, there has been a recent increase of interest in systems that can combine these three sensor modalities, whereby increasing the challenge of obtaining an accurate calibration of the extrinsics between all the different sensors, a fundamental requirement for accurate SLAM systems. Moreover, for systems that require global positioning, GNSS is the obvious choice for outdoor applications, if reliably received – which cannot be guaranteed in all environments.

Most of the visual-inertial SLAM (VI SLAM) systems proposed in the literature [2]–[5] only use a small subset of the camera information to build a sparse map, therefore lacking relevant geometric information required for downstream tasks. To address this limitation, the community has proposed

This work was supported by the EU Horizon Europe program under grant agreement 101070405 (DigiForest) and 101120732 (AUTOASSESS).

Simon Boche, Jaehyung Jung and Sebastián Barbas Laina are with the Mobile Robotics Lab, School of Computation, Information and Technology (CIT) at the Technical University of Munich (TUM), 80333 Munich, Germany, and with the Munich Institute of Robotics and Machine Intelligence (MIRMI). (e-mail: simon.boche@tum.de; jaehyung.jung@tum.de; sebastian.barbas@tum.de)
Stefan Leutenegger is with the Mobile Robotics Lab, ETH Zurich, 8092 Zurich, Switzerland. (e-mail: lestefan@ethz.ch).
* Equal contribution.

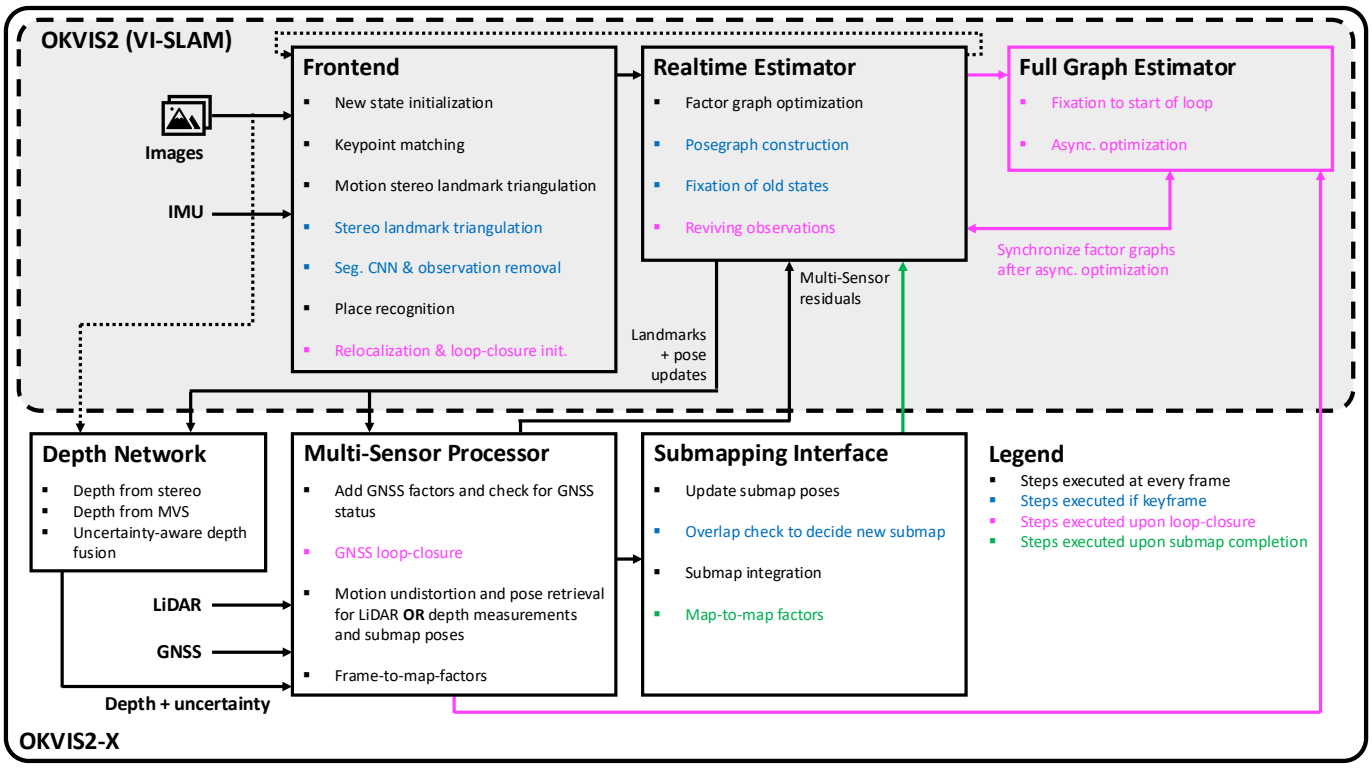

Fig. 2. System architecture of our proposed multi-sensor state-estimator OKVIS2-X. Components with a grey background correspond to original elements from OKVIS2 [2] and components with a white background are the extensions for the multi-sensor setup.

systems that can leverage visual depth information [6]–[8] or LiDAR information [9]–[11] to build denser maps that can then be reused for downstream tasks – thereby naturally increasing memory requirements. The underlying map representation plays a crucial role not only regarding memory, but perhaps more importantly to facilitate downstream tasks. We argue that volumetric occupancy maps present an unmatched characteristic in enabling safe path planning and navigation by explicitly representing free space while the popular point clouds (or other commonly employed representations such as meshes) do not.

To address the aforementioned challenges, we propose a multi-sensor SLAM system called OKVIS2-X, a substantial extension to OKVIS2 [2]. The key idea is to leverage submap-based dense volumetric occupancy maps which are tightly-coupled to the state estimator via submap alignment factors. Examples of the 3D reconstruction obtained from our framework are presented in Fig. 1. Furthermore, we present a unified framework to fuse visual, inertial, LiDAR or depth from neural networks, and the global position from GNSS. Our previous works have augmented OKVIS2 for different sensor modalities: visual-inertial-GNSS [12], visual-inertial-LiDAR [13], and visual-inertial-depth [14]. But to the best of the authors' knowledge, here we present for the first time a unified and configurable multi-sensor SLAM system based on a volumetric map representation, supporting different and novel combinations of sensing modalities (e.g. visual-inertial-LiDAR-GNSS and visual-inertial-depth-GNSS) through a factor graph framework. In a further novel contribution, we seamlessly support online-calibration of camera extrinsics. Fig. 2 describes the system architecture of OKVIS2-X that is configurable with multiple sensor modalities. The resulting system demonstrates unmatched accuracy and robustness in localization and mapping, as well as scalability in building dense volumetric maps from meter to kilometer-level – which we demonstrate in an extensive series of entirely new evaluations. The main contributions of our proposed method are:

- We present a state-of-the-art multi-sensor SLAM system able to fuse visual, inertial, GNSS and dense mapping factors from either LiDAR measurements or a depth network in realtime.
- Our method tightly couples the state estimator with volumetric occupancy mapping, rendering it suitable for large-scale SLAM thanks to our submap representation.
- Our method supports online calibration of the camera-IMU extrinsics in the visual factors including relative pose error terms.
- We perform a thorough evaluation in benchmark datasets including large-scale (up to 9 km) scenarios, where we show state-of-the-art results in VI and VI-LiDAR SLAM.
- We will release our framework as open source to foster research in the field of multi-sensor SLAM.

II. Related Work

We review the most relevant works under the topics of multi-sensor SLAM, dense SLAM, and submap-based SLAM.

A. Visual-Inertial SLAM

VI SLAM systems are generally divided into two main groups: loosely-coupled and tightly-coupled systems. Loosely-coupled estimators typically compute (relative) poses from images alone first before combining with IMU measurements, while the tightly-coupled alternatives consider all measurements simultaneously, typically through IMU kinematics integration with visual observations, e.g. in the form of reprojection errors. The latter approach is known to be more accurate. A second distinction of estimator comes from either recursively estimating a selection of recent states employing filtering, e.g. [15] versus optimizing typically a window of states in a nonlinear least-squares fashion, e.g. [3], [16]. On the other hand, depending on the type of residual, direct methods [17] minimize the photometric error, while the keypoint-based methods such as [3], [16], [18], minimize the reprojection error to estimate robot poses and sparse landmarks.

VINS-Mono [19] proposed tightly-coupled nonlinear optimization-based VI SLAM by fusing IMU preintegration and reprojection errors. OpenVINS [5] is based on the Multi-State Constraint Kalman Filter (MSCKF [15]), and extends this with long-term SLAM landmarks, extrinsic and intrinsic online calibration. ORB-SLAM3 [3] is a tightly-coupled VI-SLAM system that supports multi-session SLAM with monocular, stereo and RGB-D configurations. It records the state-of-the-art accuracy in public SLAM datasets. OKVIS2 [2] is a multi-camera SLAM system with loop closure, where pose-graph factors are formulated from marginalized landmarks. MAVIS [20] is also a multi-camera SLAM system with IMU preintegration on matrix Lie Groups. SVIn2 [21], [22] is a visual-inertial framework that incorporates loop-closures to the factor-graph optimization, while including other sensor modalities for its deployment in underwater environments.

In the context of deep learning-based VI SLAM, DeepVIO [23] is a self-supervised visual-inertial odometry (VIO) network that directly regresses trajectory from the input sensor measurements. DVI-SLAM [24] learns a confidence map of feature-metric and reprojection factors with a dense bundle adjustment (DBA) layer. DBA-Fusion [25] also utilizes the DBA to tightly fuse visual-inertial measurements. The authors demonstrated dense point cloud mapping result in a large-scale environment. However, the DBA layer, which is a main component in end-to-end SLAM systems, often requires high GPU memory usages ($\approx 24\,\text{GB}$).

In contrast to previous works, OKVIS2-X provides a general framework to work with multiple cameras and multi-modalities including depth networks, LiDAR, and GNSS receiver. Furthermore, our method goes beyond the classical sparse landmark map representation since it can build a dense volumetric occupancy representation that scales to large scenes.

B. LiDAR(-Visual-Inertial) SLAM

Due to their inherently accurate depth sensing capabilities, LiDAR sensors have proven to be beneficial for robot localization allowing to reduce the drift significantly, which SLAM systems inevitably suffer from. One of the first systems to successfully demonstrate the applicability of LiDAR sensors to SLAM was LOAM [26]. Its core idea was the extraction of salient geometric features, such as edges and planes, which can be reliably tracked across frames. LOAM has inspired an entire line of research adopting the concept of salient feature extraction [27], [28]. As in VI SLAM, one can further distinguish feature-based and direct methods. Direct LiDAR SLAM methods, such as Kiss-ICP [29] or CT-ICP [30], typically rely on variants of the Iterative Closest Point (ICP) algorithm on raw LiDAR point clouds, omitting the necessity to extract salient features. [31], [32] establish sliding-window bundle adjustment (BA) for LiDAR scans for feature-based and direct formulations.

Fusing complementary IMU measurements enables the deployment on highly dynamic systems. LIO-SAM [33] formulates a tightly-coupled pose graph optimization problem fusing edge and plane error residuals with IMU error residuals. Also filter-based approaches, such as FAST-LIO [34] or FAST-LIO2 [35] have successfully demonstrated high-accuracy localization fusing LiDAR and IMU. In contrast to FAST-LIO, which still uses plane and edge features, FAST-LIO2 is a direct method achieving a significant speed-up by directly operating on the raw points.

Recently, LiDAR-Visual-Inertial SLAM systems have become increasingly popular. A large amount of these approaches still make use of LOAM's idea of geometric feature extraction. LVI-SAM [10] combines a Visual-Inertial (VI) and a LiDAR-Inertial (LI) subsystem to complement each other in challenging scenarios. The LI system builds a factor graph based on IMU preintegration errors and edge and plane residuals. VILENS [11] also builds an optimisation problem consisting of visual, inertial, robot leg odometry, as well as LiDAR-based line and plane residuals. Another line of research uses MSCKF [15]-based approaches for tightly-coupled fusion of visual, inertial and LiDAR measurements, e.g. LIC-Fusion [36] and its successor LIC-Fusion 2.0 [37]. R2LIVE [38] fuses visual features, IMU and LiDAR measurements in an Iterated Kalman Filter. Unlike previously mentioned LVI SLAM systems, the succeeding R3LIVE [39] applies direct methods, increasing its robustness in texture-less or structure-less environments. FAST-LIVO [40] and FAST-LIVO2 [9] also formulate an Iterated Extended Kalman Filter (IEKF) framework. The IEKF update step consists of two sequential state updates: a point-to-plane based LiDAR update and a sparse direct visual update operating on image patches instead of dense pixels. Both updates are computed in a frame-to-map fashion using a local voxel map.

In contrast to previous approaches, the proposed OKVIS2-X can fuse LiDAR measurements omitting the necessity to extract salient geometric features or any other correspondences, usually a computationally expensive step. Instead, based on our previous work [13], residuals can be directly derived from dense occupancy maps which are incrementally constructed and are directly usable for downstream tasks. These residuals are added to a factor graph optimization problem.

C. Multi-Sensor SLAM with GNSS fusion

Early work on fusing global position measurements to reduce the drift in VI SLAM is dominated by filter-based methods, mostly building upon the seminal work in [15] where the authors propose an Extended Kalman Filter (EKF) to tackle realtime visual-inertial navigation. [41] and [42] use an EKF and an Unscented Kalman Filter (UKF), respectively, to fuse different sensor modalities, such as LiDAR and GNSS measurements. Additionally, [43] also estimates the IMU-GNSS spatial extrinsics as well as the sensor time offset online in an MSCKF [15] framework. [44] proposes a robust Iterated Error State Kalman Filter (IESKF) to fuse GNSS, IMU and LiDAR measurements in a tightly-coupled approach. A filter-based system that not only fuses Lidar-Visual-Inertial information, but also wheel odometry and GNSS is MINS [45], a system that extends [5] and demonstrates how multi-sensor information improves the robustness of state-estimation. Filter-based methods are in general computationally efficient. Nevertheless, as investigated in [46], optimization-based approaches potentially deliver superior results, thanks to their ability of re-linearizing old state estimates.

VINS-Fusion [18] and GOMSF [47] are examples for optimization-based methods fusing global position measurements in a loosely-coupled back-end pose-graph optimization. However, a tightly coupled fusion in a unified optimization problem is desirable to fully exploit the available information given by different sensor modalities. [33] offers the integration of GNSS factors into LiDAR-Inertial Odometry as a factor graph optimization problem. [48] proposes a tightly-coupled approach fusing GNSS measurements in the optimization window of a VIO as global factors leveraging IMU preintegration. [49]–[51] have also proposed tightly-coupled optimization-based frameworks which consider not only global position measurements but also pseudo range and Doppler shift errors from raw GNSS measurements. Unlike the previous, pure odometry approaches, [52], [53] successfully demonstrated the fusion of global position measurements into VI SLAM including visual loop closures.

While [48] assumes global position measurements given in the visual-inertial reference frame, in practice, a 4-Degree-of-Freedom (DoF) transformation between a global and the local reference fame has to be estimated. Initialization of this global frame alignment has been addressed using SVD based on correspondences of local and global position measurements [47] or in a tightly-coupled fashion using least-squares minimization [43], [49], [50]. [43] furthermore introduces an heuristic criterion on the observability of the 4-DoF transformation based on the distance traveled. As in [47] and following our previous work [12], we will use a SVD-based initialization of the GNSS-VIO extrinsics. However, it will be estimated in a tightly-coupled way. In contrast to the aforementioned approaches, to simplify the problem, the global reference frame will be fixed in our approach as soon as it becomes observable. Instead of applying a heuristic threshold as in [43], we introduce a fully uncertainty-aware criterion to determine the covariance of the estimated GNSS extrinsics. Furthermore, our framework supports not only visual loop-closures but also provides a loop-closure like optimization approach to effectively tackle the issue of drift during longer periods of GNSS outage as presented in [12]. In new experiments, here, we furthermore present how OKVIS2-X supports Lidar-Visual-Inertial SLAM with intermittently available GNSS.

D. Dense SLAM & Mapping

Most of the previously mentioned SLAM systems achieve high accuracy in localization relying only on sparse map representations, such as sparse 3D landmarks or feature maps. However, understanding dense geometry is indispensable for deploying fully autonomous robots in downstream tasks.

KinectFusion [6] pioneered the area of dense SLAM with Truncated Signed Distance Fields (TSDF) fusion and frame-to-model registration. ElasticFusion [54] models an environment with dense surfels, which are updated non-rigidly after a loop. CNN-SLAM [7] and DeepFusion [55] employed dense depth from a convolutional neural network and fused depths from motion stereo to further improve the depth quality. Kimera [56] and its extension Kimera2 [4] presented the dynamic scene graph built by visual-inertial observations. In their map representation, semantically annotated meshes are deformed by loop-closure constraints. TANDEM [57] integrated a deep Multi-View Stereo (MVS) network in visual odometry, additionally rendering depth from TSDF for image alignment. Simplemapping [58] adopted SimpleRecon [59] and fused depths from the MVS network into a TSDF given poses and sparse landmarks from ORB-SLAM3 [3] but without any feedback from mapping. SigmaFusion [60] predicted dense depth uncertainty based on the information matrix from Droid-SLAM [61] and showed less noisy reconstruction through uncertainty integration. However, their method is memory intensive, and mapping and state estimation are decoupled. A new line of work based on geometric foundation models has recently emerged in the SLAM community, with MASt3R-SLAM [62], a monocular SLAM system, being one of the most notable examples. These systems leverage point maps for both mapping and localization.

In the context of dense neural SLAM, where map geometry and appearance benefits from the neural implicit representation, NICE-SLAM [63] represents the map with hierarchical feature grids that are decoded into a volumetric occupancy map. UncLe-SLAM [64] incorporates pixel-wise depth and color uncertainty in tracking and mapping loss functions in NICE-SLAM to properly account for observation uncertainty. Point-SLAM [65] refrains from a grid-based representation to save memory by assigning neural points only around surfaces that are later used to obtain the map occupancy and color. Loopy-SLAM [66] extends Point-SLAM by incorporating loop-closure constraints in the neural point representation by adopting a submap strategy where neural points are anchored at each submap to build a globally consistent map. Aforementioned dense neural SLAM methods have shown promising appearance rendering in (multi-)room-sized environments. However, it is not clear how those lines of works are scalable to km-level large-scale scenarios, which is the main challenge this paper addresses, given the high dimensionality of neural

features – or how these representations efficiently support mobile robot navigation.

In contrast, our proposed system supports tight-coupling between the state estimator and the volumetric free-space mapping for generic sensor setups, as presented in our previous works for depth cameras [14] and LiDAR sensors [13]. The proposed approach is fully probabilistic, enabling the consideration of uncertainties also in downstream tasks.

E. Submapping Approaches

To minimize 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 [67]. In this context, additional factors can be derived to align submaps and to reduce drift. [11] uses local point cloud submaps and adds ICP odometry measurements into the factor graph. Wildcat [68], a sliding-window optimization-based LiDAR-Inertial Odometry system, achieves peak state-of-the-art robustness and accuracy by building local surfel submaps and aligning the submaps. These methods use point or surface-based 3D representations. While these can evidently be leveraged for high-accuracy localization and 3D reconstruction, the map representation is not suitable for navigation due to the difficulty to distinguish between free and unobserved space.

Submapping on volumetric maps has been addressed in various works. [69] builds occupancy submaps based on OctoMap [70] and aligns them by standard ICP registration. Voxgraph [71] uses VIO to provide poses for integration into TSDF maps. Upon completion, Euclidean Signed Distance Fields (ESDF) are generated and submaps are aligned in pose graph optimization. New submaps are created at a fixed frequency resulting in a comparably large memory usage. [72] instead use occupancy maps as their 3D representation. Using Supereight2 [73], an adaptive-resolution mapping approach, new submaps are spawned based on the distance traveled. Submaps are re-arranged based on updates from the visual-inertial estimator. The follow-up work [74] 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 also adopt the concept of submapping. In contrast to [71], [72], [74], the global alignment of submaps is not decoupled from the estimator but provides direct feedback. We formulate correspondence-free residuals as in [71] without the expensive need to extract ESDFs as we directly use the available occupancy information.

III. Preliminaries

A. Notation and Definitions

The classic VI SLAM problem formulation includes several different coordinate frames. A moving body is tracked with respect to a fixed world reference frame $\underrightarrow{\mathcal{F}}_W$. We will denote the IMU frame by $\underrightarrow{\mathcal{F}}_S$ and the camera coordinate frames by $\underrightarrow{\mathcal{F}}_{C_i}$ for $i = 1 \dots N$ cameras. Fusing LiDAR requires introducing the LiDAR sensor frame $\underrightarrow{\mathcal{F}}_L$ and fusing GNSS measurements requires to introduce a global reference frame $\underrightarrow{\mathcal{F}}_G$ which is a gravity-aligned East-North-UP (ENU) local Cartesian frame located in the global position of the first received GNSS measurement. The submap frame is denoted as $\underrightarrow{\mathcal{F}}_M$ that corresponds to $\underrightarrow{\mathcal{F}}_S$ when the new submap is declared.

The rigid body transformation $\boldsymbol{T}_{AB} \in SE(3)$ transforms homogeneous points between two frames: ${}_A\mathbf{r}_P = \boldsymbol{T}_{AB}\,{}_B\mathbf{r}_P$, where ${}_A\mathbf{r}_P$ is the position of a point P in frame $\underrightarrow{\mathcal{F}}_A$. The rotational part of $\boldsymbol{T}_{AB}$ is expressed by $\mathbf{C}_{AB} \in SO(3)$ and ${}_A\mathbf{r}_B$ denotes the translation component. We also denote the rotation $\mathbf{C}_{AB}$ with its unit quaternion form $\mathbf{q}_{AB}$.

B. State Definition

The state representation in this work is:

$$\mathbf{x} = \left[{}_W\mathbf{r}_S^T, \mathbf{q}_{WS}{}^T, {}_W\mathbf{v}^T, \mathbf{b}_\mathrm{g}^T, \mathbf{b}_\mathrm{a}^T\right]^T, \tag{1}$$

where ${}_W\mathbf{r}_S$, $\mathbf{q}_{WS}$ and ${}_W\mathbf{v}$ denote the position, orientation and velocity of the IMU sensor frame in the fixed world frame. $\mathbf{b}_\mathrm{g}$ and $\mathbf{b}_\mathrm{a}$ stand for gyroscope and accelerometer biases, respectively. To fuse the visual-inertial system and global measurements, we estimate the extrinsic transformation $[{}_G\mathbf{r}_W^T, \mathbf{q}_{GW}{}^T]$ between the world reference frame of the estimator $\underrightarrow{\mathcal{F}}_W$ and the GNSS reference frame $\underrightarrow{\mathcal{F}}_G$. In case of online calibration of the camera-IMU extrinsics, the state is further augmented with the extrinsics $[{}_S\mathbf{r}_{C_i}^T, \mathbf{q}_{SC_i}{}^T]$ for $i = 1 \dots N$ with N being the number of cameras.

IV. Volumetric occupancy mapping

Obtaining an accurate map representation is essential for state-estimation and downstream tasks. Most of the state-of-the-art VI SLAM systems [3], [5], [20], [75] build a sparse map based on image features, which is used for state estimation but cannot be leveraged for most downstream tasks due to the lack of dense geometric details. To address this, OKVIS2-X leverages dense volumetric occupancy submaps based on the multi-resolution mapping framework Supereight2 [73], which are also considered in state estimation and can be used in downstream tasks such as autonomous navigation [76], exploration [77] or place recognition [78].

In OKVIS2-X, we refrain from the concept of monolithic mapping and we leverage a submapping strategy for environment representation, obtaining two major benefits. First, data is allocated into smaller maps, resulting in a shallower data-structure and consequently reducing the computational cost of integrating the depth information. Second, each submap is anchored to individual keyframe states from the state estimator, enabling submap poses to be adjusted during updates.

Upon creation, the submap reference frame $\underrightarrow{\mathcal{F}}_M$ is anchored to a keyframe state $\mathbf{x}^k$, and its body pose $\boldsymbol{T}_{WS_k}$ expresses the relationship between $\underrightarrow{\mathcal{F}}_M$ and $\underrightarrow{\mathcal{F}}_W$. Upon an update of $\mathbf{x}^k$, the new body pose $\boldsymbol{T}_{WS_k}$ updates the rigid transformation between $\underrightarrow{\mathcal{F}}_M$ and $\underrightarrow{\mathcal{F}}_W$, ensuring the local consistency between submaps and improving the overall mapping accuracy. Each submap maintains occupancy log-odds which is updated with either LiDAR point clouds or depth images $\mathbf{D}$. The log-odd of a point in the map, ${}_M\mathbf{p}$ is defined as

$$l\left({}_M\mathbf{p}\right) = \log \frac{P_\mathrm{occ}\left({}_M\mathbf{p} \mid \boldsymbol{T}_{MC}, \mathbf{D}\right)}{1 - P_\mathrm{occ}\left({}_M\mathbf{p} \mid \boldsymbol{T}_{MC}, \mathbf{D}\right)}, \tag{2}$$

where P_{occ} is the occupancy probability. As in Supereight2, we perform additive Bayesian updates of the log-odds occupancies as we integrate new depth information. The mean of log-odd $L(\cdot)$ is recursively updated as

$$L_k\left({}_M\mathbf{p}\right) = \frac{L_{k-1}\left({}_M\mathbf{p}\right) w_{k-1} + l\left({}_M\mathbf{p}\right)}{w_{k-1}+1}, \qquad w_k = \min\{w_{k-1}+1,\ w_{\max}\}, \tag{3}$$

where w_k is the number of observations and saturated in $w_{\max}$. This update alleviates the geometric inconsistencies from non-accurate depth perception, more critical when our system leverages learned depth, while also increasing the robustness to the dynamic entities that are perceived from the scene.

The assumption of our submap strategy is that drift within a submap is negligible, therefore to generate a new submap two criteria have to be met. First, a minimum number of depth measurements has been integrated into the current submap. Second, the overlap between the current depth or LiDAR measurement and the previously completed submap is lower than a threshold or that a maximum of K keyframes have been generated since the creation of the current submap. The keyframe criterion is a proxy for the drift within a submap and the overlap criteria ensures that submap-based alignment constraints can reliably be added to the state estimator.

By leveraging occupancy probabilities as a map representation, the environment can be classified as free, occupied or unobserved, a distinction that becomes essential for safe autonomous navigation. Similar to [76], the path planner only traverses areas considered free in a submap, since it considers unknown areas as non-traversable, and these paths are elastically deformed, according to the submap updates.

Supereight2 [73] uses a piecewise linear inverse sensor model for the log-odds occupancy as a function of the measured depth. Hereby, a log-odds value of zero represents the measured surface position. Moreover, the depth uncertainty σ can also be modeled as a function of the measured depth z_r. In our work, we assume a linearly growing uncertainty for the LiDAR sensor and quadratically for RGB-D cameras.

However, these heuristic uncertainty models are not sufficient for learned depth. For instance, textureless regions or object boundaries yield high depth uncertainty due to the ambiguous correlation volume or viewpoint changes. In response, we will introduce a depth fusion method to predict the pixel-wise uncertainty in the following section.

A. Uncertainty-aware depth fusion

It is pivotal to obtain a reliable dense depth as well as the associated uncertainty for our downstream tasks. We follow the key ideas presented in [14], our previous work, where depths from static and motion stereo are fused, which are complementary to each other — static stereo provides a reliable small baseline even when stationary, while motion stereo potentially brings a large baseline, depending on the camera motion. Our method does not depend on a specific network, but we found that Unimatch [79] and the MVS network [58] work well in real-world scenes. However, the previous networks only predict disparity or depth without any uncertainties. Therefore, we augment the base architecture with an uncertainty decoder and adopt the Laplacian loss function for (aleatoric) uncertainty learning. Specifically, our stereo network loss function is

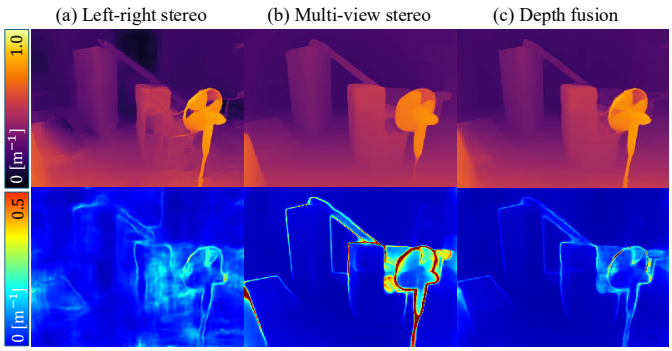

Fig. 3. Predicted inverse depth (top) and its corresponding standard deviation (bottom) of (a) stereo network with the 11 cm baseline, (b) MVS network with the 50 cm maximum baseline among 8 views, and (c) depth fusion in the EuRoC dataset. (Adopted from [14].)

$$\mathcal{L}_{\text{st}}(\boldsymbol{\theta}) = \mathcal{L}_u(\boldsymbol{\theta}) + \mathcal{L}_{\nabla u_x}(\boldsymbol{\theta}) + \mathcal{L}_{\nabla u_y}(\boldsymbol{\theta}), \tag{4}$$

where $\boldsymbol{\theta}$ is the network weights, and u stands for the disparity. The disparity loss $\mathcal{L}_u$, modeled in the Laplacian distribution as in [80], [81], is defined as

$$\mathcal{L}_u(\boldsymbol{\theta}) = \sum_{i\in\mathcal{T}} \frac{\left|u_i - u_{\text{gt}_i}\right|}{\sigma_{u_i}} + \log\sigma_{u_i}, \tag{5}$$

where $\mathcal{T}$ is a training set including pairs of stereo images and the ground-truth disparity. We additionally add the gradient loss $\mathcal{L}_{\nabla u}$ for sharper uncertainty output, which is analogously defined as $\mathcal{L}_u$. The gradient uncertainty along the horizontal and vertical directions is derived from the disparity uncertainty as

$$\sigma_{\nabla u_x} = \sqrt{\sigma_u^2(x+1,y) + \sigma_u^2(x-1,y)}, \qquad \sigma_{\nabla u_y} = \sqrt{\sigma_u^2(x,y+1) + \sigma_u^2(x,y-1)}. \tag{6}$$

We propagate the uncertainty from the disparity to the depth with linearization,

$$\hat{d}_{\text{st}} = \frac{f_c b}{u}, \quad \sigma_{\text{st}} = \frac{f_c b}{u^2}\sigma_u, \tag{7}$$

where f_c is the rectified focal length, b is the stereo baseline.

Likewise, we modify the loss function of the MVS network [58]

$$\mathcal{L}_{\text{mvs}}(\boldsymbol{\phi}) = \sum_{i\in\mathcal{T}} \frac{\left|\log d_i - \log d_{\text{gt}_i}\right|}{\sigma_{l_i}} + \log\sigma_{l_i}, \tag{8}$$

where the network learns log-depth uncertainty. We transform the log-depth to a depth with a linearized model,

$$\hat{d}_{\text{mvs}} = \exp\left(\log d_i\right), \quad \sigma_{\text{mvs}} = \hat{d}_{\text{mvs}}\sigma_l. \tag{9}$$

Given the pixel-wise estimates from the networks $(\hat{d}_{\text{st}}, \sigma_{\text{st}})$, $(\hat{d}_{\text{mvs}}, \sigma_{\text{mvs}})$ and with the assumption that two estimates are independent, we can optimally fuse two depth estimates [82],

$$\hat{d}_{\text{fuse}} = \sigma_{\text{fuse}}^2\left(\sigma_{\text{st}}^{-2}\hat{d}_{\text{st}} + \sigma_{\text{mvs}}^{-2}\hat{d}_{\text{mvs}}\right), \qquad \sigma_{\text{fuse}}^2 = \left(\sigma_{\text{st}}^{-2} + \sigma_{\text{mvs}}^{-2}\right)^{-1}., \tag{10}$$

Our stereo and MVS networks with additional uncertainty decoders are only fine-tuned in a synthetic dataset [83]. Fig. 3 shows an example where the fan stand has more detailed depth in the stereo network, while farther objects are sharper in the MVS network. Fused depth naturally maintains the optimal depth based on uncertainty-aware fusion.

V. Multi-sensor SLAM

A. System Overview

OKVIS2-X proposes a modular multi-sensor SLAM system, specifically designed for large-scale scenarios and robot navigation. It builds on top of the sparse VI SLAM system OKVIS2 [2] and adds capabilities for online camera-IMU extrinsics calibration, fusion of global position measurements and dense map alignment factors derived from LiDAR or depth sensing modalities. Fig. 2 visualizes the overall system architecture and how it extends the original OKVIS2.

The underlying VI SLAM system is split into the visual frontend and a realtime estimator that process images and IMU messages synchronously whenever a new (multi-) frame arrives. To deal with loop-closures, a full factor graph loop optimization is executed asynchronously. As shown in Fig. 2, the frontend deals with state initialization, keypoint matching, stereo triangulation (of successive frames and from stereo images of the same multi-frame), if enabled, running a segmentation CNN (to filter out observations in the sky), as well as place recognition and, if the latter was successful, relocalization and loop-closure initialization. The realtime estimator will then optimize the respective factor graph, and is also responsible for the creation of posegraph edges by marginalizing old observations, as well as for fixation of old states. Upon loop-closure, it turns posegraph edges back into observations, and then triggers the optimization of the full graph around a loop, which runs asynchronously, and which will be synchronized with the realtime factor graph upon completion.

OKVIS2-X extends this system by three additional modules: *Depth Network*, *Multi-Sensor Processor*, and the *Submapping Interface*. *Depth Network* takes stereo images for the stereo network, as well as robot poses and sparse landmarks from the realtime estimator for the MVS network. The sparse landmarks are back-projected to image planes to provide a depth prior for the network. Depending on the use case, the *Multi-Sensor Processor* deals with incoming GNSS measurements and geometric measurements coming either from a LiDAR, a depth camera or the presented depth fusion network. For GNSS measurements, it will formulate global position residuals that are added to the realtime estimator. It further determines the time that has passed since the last received GNSS measurement. In case of long signal outages, it will trigger an asynchronous, loop-closure like optimization of the full graph to compensate for the potentially accumulated drift since the last received measurement. In case of dense submap alignment, it retrieves poses for depth images or applies IMU-based motion undistortion of incoming LiDAR point clouds. For that it keeps an internal representation of the trajectory which is always kept synchronized with the estimator. For every live frame, frame-to-map factors with respect to a previous submap are added to the factor graph.

The *Submapping Interface* manages the collection of all submaps. The pose of each submap is anchored to a keyframe pose from the state estimator, therefore submap poses are always updated with updates from the state estimator using the same internal trajectory representation as the *Multi-Sensor Processor*. Each time a visual keyframe is generated, it checks for the overlap ratio of incoming measurements with respect to the last submap to decide whether a new submap is created. After that decision, depth images or LiDAR measurements are integrated into the current active submap. Upon completion of a submap, dense map-to-map factors are added in the optimization problem to keep consistency of submaps in overlapping regions. In the following sections, we will describe each module in more detail.

B. Visual Frontend

The visual frontend extracts BRISK [84] keypoints and descriptors in every image of a multi-frame, and matches them to the 3D landmarks already in the map; hereby both, descriptor distance and reprojected image distance, are considered. New 3D landmarks are then initialized both from stereo triangulation within all images of a keyframe, as well as from triangulation between live frame and any of the keyframes. The decision of whether a new frame is considered a keyframe is taken depending on the fraction of matched landmarks in the live frame, as well as the fraction of current matches visible in any of the existing keyframes. If the overlap falls below a threshold, we set the live (multi-)frame as a new keyframe.

A finetuned version of the semantic segmentation network Fast-SCNN [85] can be run on keyframe images. For maximum portability and flexibility, inference is carried out asynchronously on the CPU, which is tractable, due to the efficient network as well as the fact that only keyframes are processed. In our implementation, matches that are in the sky are removed. This scheme significantly improves accuracy in presence of slow-moving scene content, particularly clouds, where observations are not already automatically discarded via the Cauchy robustification.

C. Place Recognition, Relocalization, and loop-closure

OKVIS2-X maintains a DBoW2 [86] database. DBoW queries of the current frames return a list of matches as loop-closure candidates. To be considered a valid loop-closure, an additional geometric verification step using 3D-2D RANSAC has to be passed. Then, the posegraph edges connecting the loop-closure state will be "revived" and turned back into landmarks and observations (see Fig. 4 (c)); and observations with the current matching frame will also be created. This may also trigger merging of landmarks, if already existing new landmarks are matched to old landmarks. To optimize the loop inconsistency, the error is equally distributed around the loop using rotation averaging followed by position inconsistency distribution. Then, a background optimization process of the full graph is triggered. After the optimization has finished, a synchronization process imports the optimized states and landmarks into the realtime estimator, and re-aligns new states and landmarks created in the meantime.

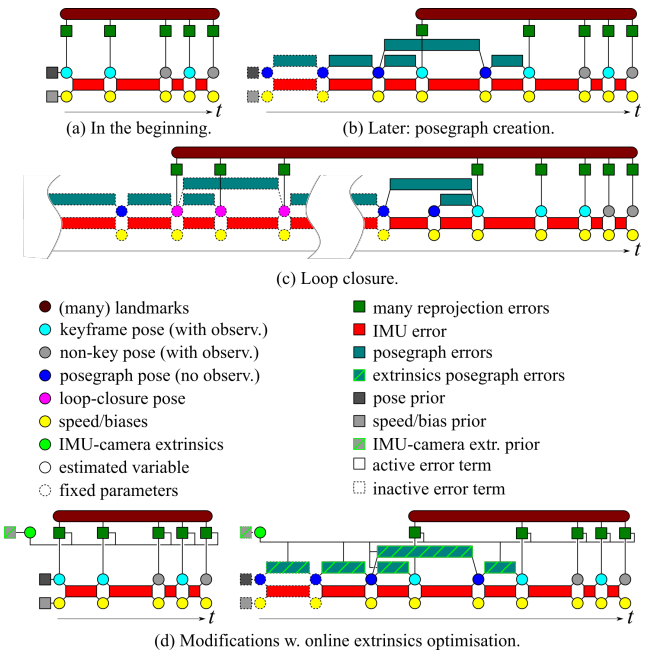

Fig. 4. Initially a full batch VI factor graph (a) is created and optimized. Later (b), frames with least overlap with the live frame and current keyframe are turned into posegraph poses by construction of relative pose errors under marginalization of common observations; also, old poses and speed/bias variables are fixed to keep the problem realtime capable. When a loop-closure occurs (c), respective observations and landmarks are re-activated. The proposed system furthermore supports online calibration of the IMU-camera extrinsics (d). ((a-c) Adopted from [2].)

D. Visual-Inertial SLAM

The VI estimator will be minimizing visual, inertial, and relative pose errors (posegraph edges), which are briefly introduced in what follows. Fig. 4(a)-(c) visually overviews the construction of the underlying factor graph as time progresses.

1) Visual Reprojection Errors: Just like OKVIS2 [2], OKVIS2-X uses 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

$$\mathbf{e}_{\mathrm{r}}^{i,j,k} = \tilde{\mathbf{z}}_{\mathrm{r}}^{i,j,k} - \mathbf{h}\left(\boldsymbol{T}_{SC_i}^{-1}\boldsymbol{T}_{WS_k}^{-1}{}_{W}\mathbf{l}^{j}\right). \tag{11}$$

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

2) IMU Errors: For the formulation of the IMU residuals, we adopt the IMU preintegration approach in [87]. Between time steps k and n, the IMU error is:

$$\mathbf{e}_{\mathrm{s}}^{k} = \tilde{\mathbf{x}}^{n}\left(\mathbf{x}^{k}, \tilde{\mathbf{z}}_{\mathrm{s}}^{k,n}\right) \boxminus \mathbf{x}^{n}, \tag{12}$$

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 [2]).

3) Relative Posegraph Errors: Relative pose errors $\mathbf{e}_{\mathrm{p}}^{r,c}$ between time steps r (the reference) and c are given by

$$\mathbf{e}_{\mathrm{p}}^{r,c} = \mathbf{e}_{\mathrm{p},0}^{r,c} + \begin{bmatrix} {}_{S_r}\mathbf{r}_{S_c} - {}_{S_r}\bar{\mathbf{r}}_{S_c} \\ \mathbf{q}_{S_rS_c} \boxminus \bar{\mathbf{q}}_{S_rS_c}, \end{bmatrix} \tag{13}$$

with ${}_{S_r}\bar{\mathbf{r}}_{S_c}$ and $\bar{\mathbf{q}}_{S_rS_c}$ being nominal relative position and orientations expressed in the IMU frame $\underrightarrow{\mathcal{F}}_{S_r}$.

In short, these posegraph error terms are computed from joint observations into frames r and c by marginalizing out the landmarks. We refer the reader to [2] for a detailed explanation of how a Maximum Spanning Tree (MST) is employed to select posegraph edges to be created. To compute the constant $\mathbf{e}_{\mathrm{p},0}^{r,c}$ as well as the weight matrix $\mathbf{W}_{\mathrm{s}}^{k}$, we transform the landmarks into the reference frame $\underrightarrow{\mathcal{F}}_{S_r}$ and formulate the Gauss-Newton-System of the form $\mathbf{H}\delta\boldsymbol{\chi} = \mathbf{b}$ just from joint observations and respective landmarks, i.e. from standard reprojection errors and respective well-known Jacobians, resulting in the following structure:

$$\begin{bmatrix} \mathbf{H}_{\mathrm{p,p}} & \dots & \mathbf{H}_{\mathrm{p},j} & \dots \\ \vdots & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_{\mathrm{p},j}^{T} & \mathbf{0} & \mathbf{H}_{j,j} & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{0} & \ddots \end{bmatrix} \begin{bmatrix} \delta\mathbf{p} \\ \vdots \\ \delta\mathbf{l}_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\mathrm{p}} \\ \vdots \\ \mathbf{b}_j \\ \vdots \end{bmatrix}. \tag{14}$$

Hereby, only one pose occurs, i.e. the relative pose $\boldsymbol{T}_{S_rS_c}$ referred to by $\delta\mathbf{p}$. The variable ordering follows with all the landmarks $({}_{S^r}\mathbf{l}^{j})$ referred to as $\delta\mathbf{l}_j$. Now, landmarks are marginalized out using the Schur complement:

$$\mathbf{H}^{*} = \mathbf{H}_{\mathrm{p,p}} - \sum_{j} \mathbf{H}_{\mathrm{p},j}\mathbf{H}_{jj}^{+}\mathbf{H}_{\mathrm{p},j}^{T}, \tag{15}$$

$$\mathbf{b}^{*} = \mathbf{b}_{\mathrm{p}} - \sum_{j} \mathbf{H}_{\mathrm{p},j}\mathbf{H}_{jj}^{+}\mathbf{b}_{j}, \tag{16}$$

which yields the reduced Gauss-Newton system

$$\mathbf{H}^{*}\delta\mathbf{p} = \mathbf{b}^{*}, \tag{17}$$

which is now linearized around the current pose $\bar{\mathbf{p}}$, therefore

$$\mathbf{H}^{*}\delta\mathbf{p} = \mathbf{b}^{*} - \mathbf{H}^{*}(\mathbf{p} \boxminus \bar{\mathbf{p}}) \tag{18}$$

The supposedly equivalent Gauss-Newton system of a respective posegraph error term takes the form

$$\mathbf{E}_{\mathrm{p}}^{r,c\,T}\mathbf{W}_{\mathrm{p}}^{r,c}\mathbf{E}_{\mathrm{p}}^{r,c}\delta\mathbf{p} = -\mathbf{E}_{\mathrm{p}}^{r,c\,T}\mathbf{W}_{\mathrm{p}}^{r,c}(\mathbf{e}_{\mathrm{p},0}^{r,c} + \mathbf{p} \boxminus \bar{\mathbf{p}}), \tag{19}$$

where $\mathbf{E}_{\mathrm{p}}^{r,c}$ denotes the Jacobian. For this to be equivalent to (18) at the linearization point (i.e. at $\mathbf{p} = \bar{\mathbf{p}}$ where $\mathbf{E}_{\mathrm{p}}^{r,c} = \mathbf{I}_6$), we now obtain

$$\mathbf{W}_{\mathrm{p}}^{r,c} = \mathbf{H}^{*}, \tag{20}$$

$$\mathbf{e}_{\mathrm{p},0}^{r,c} = -\mathbf{H}^{*+}\mathbf{b}^{*}. \tag{21}$$

Since $\mathbf{H}^{*}$, $\mathbf{b}^{*}$, and $\bar{\mathbf{p}}$ are constants, the Jacobians w.r.t. the poses at steps r and c are fairly straightforward to determine after substituting $\boldsymbol{T}_{S_rS_c} = \boldsymbol{T}_{WS_r}^{-1}\boldsymbol{T}_{WS_c}$.

E. Online Camera-IMU Extrinsics Calibration

We have extended [2] to fully support online calibration of camera-IMU extrinsic poses $\boldsymbol{T}_{SC_i}, i = \{1, \dots, N\}$. While this extension is straightforward and well-known for reprojection errors, the relative posegraph factors from Eqn. (13) now also depend on $\boldsymbol{T}_{SC_i}$. Consequently, before marginalizing co-visible landmarks, the two-view Gauss-Newton system will be augmented to include extrinsics poses – which remain after marginalizing out landmarks. Eqns (15) ff. remain the same,

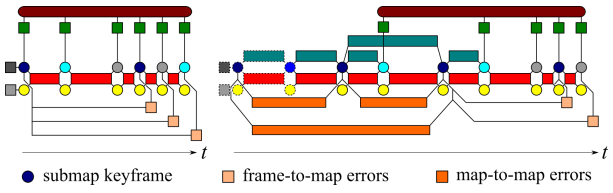

Fig. 5. Factor Graph including dense submap alignment. Left: The realtime estimator connects set of current keyframe and non-keyframe states by IMU errors and visual reprojection errors. For every state in the optimization window, frame-to-map factors are formulated between every live state and the keyframe state associated to the last completed submap. Right: Measurements between frames can be aggregated and map-to-map factors can be added to the factor graph between submap keyframe states if the geometric overlap surpasses a threshold. (Adopted from [13].)

but with a higher-dimensional $\mathbf{H}$ and $\mathbf{b}$. Eqn. (13) is then consequently extended as

$$\mathbf{e}_{\mathrm{po}}^{r,c} = \mathbf{e}_{\mathrm{po},0}^{r,c} + \begin{bmatrix} {}_{S_r}\mathbf{r}_{S_c} - {}_{S_r}\bar{\mathbf{r}}_{S_c} \\ \mathbf{q}_{S_rS_c} \boxminus \bar{\mathbf{q}}_{S_rS_c} \\ {}_{S}\mathbf{r}_{C_1} - {}_{S}\bar{\mathbf{r}}_{C_1} \\ \mathbf{q}_{SC_1} \boxminus \bar{\mathbf{q}}_{SC_1} \\ \vdots \\ {}_{S}\mathbf{r}_{C_N} - {}_{S}\bar{\mathbf{r}}_{C_N} \\ \mathbf{q}_{SC_N} \boxminus \bar{\mathbf{q}}_{SC_N} \end{bmatrix} \in \mathbb{R}^{6+6N}, \tag{22}$$

where ${}_{S}\bar{\mathbf{r}}_{C_i}, \bar{\mathbf{q}}_{SC_i}$ denote the linearization point of the i^{th} extrinsics, i.e. their values upon construction of the (augmented) relative pose factor.

The related factor graph structure can be observed in Fig. 4(d). Furthermore, to regularize the estimation problem, we add a pose prior to all camera extrinsics using the same formulation as with the pose prior of the estimated robot state[1].

F. Submap Alignment

As stated in Sec. V-A and visualized in Fig. 5, there are two types of dense submap alignment residuals. Frame-to-map factors can be formulated between every live frame and the keyframe associated to the last completed submap. Map-to-map residuals are computed between two keyframes that two submaps are anchored to. However, regarding the actual error formulation, they do not differ. Given a completed submap associated to a keyframe pose $\boldsymbol{T}_{WS_a}$ and a point cloud $\mathcal{P}_b$ associated to another state $\boldsymbol{T}_{WS_b}$, we formulate the map-based residual for every ${}_{S_b}\mathbf{p} \in \mathcal{P}_b$ as:

$$e_{\mathrm{m}}^{a,b}\left({}_{S_a}\mathbf{p}\right) = \frac{d}{\sigma} = \frac{L\left({}_{S_a}\mathbf{p}\right)}{\sqrt{\frac{L_{\min}^2}{9} + \sigma_d^2 \left|\nabla L\left({}_{S_a}\mathbf{p}\right)\right|^2}}, \tag{23}$$

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 respective Jacobians can be found in Appendix D. In the case of depth images, points ${}_{S_b}\mathbf{p}$ are obtained by backprojecting pixel depth values. The idea of the map alignment residuals is that every measured point should be on a surface ($L = 0$) 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 near the surface as in the sensor model for mapping. The distance d and the map uncertainty can be derived as:

$$d = \frac{L}{|\nabla L|}, \qquad \sigma_{\mathrm{map}} = \frac{L_{\min}}{3|\nabla L|}. \tag{24}$$

$L_{\min}$ is a configuration parameter and denotes the saturation minimum log-odds occupancy value. With the sensor-specific measurement uncertainty σ_d, we can formulate the weighted residual in Eqn. (23) using the total uncertainty

$$\sigma = \sqrt{\sigma_{\mathrm{map}}^2 + \sigma_d^2}. \tag{25}$$

While we found that for the typically highly precise LiDAR measurements, the simple assumption of an isotropic sensor uncertainty is sufficient, this does not hold for depth from the network. In that case, the depth uncertainty as in Eqn (10), which often includes noisy observations, should be assigned pixel-wise following the approach descriped in Sec. IV-A to properly weight its contribution to the optimization loss.

[1]In case we run a full-batch final BA including extrinsics, this prior is removed beforehand.

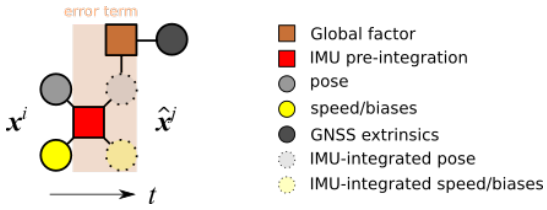

Fig. 6. To evaluate the global position residual, the propagated state $\hat{\mathbf{x}}^j$ at time step j of the measurement is propagated from the state vector $\mathbf{x}^i$ at time step i of the camera frame by leveraging IMU preintegration. This intermediate state is only used for evaluation of the global factor. (Adopted from [12].)

G. GNSS Fusion

In order to fuse global position residuals, such as from GNSS, we augment the state vector to also contain the 4-DoF transformation $\boldsymbol{T}_{GW}$ between the VI reference frame $\underrightarrow{\mathcal{F}}_W$ and a global reference frame $\underrightarrow{\mathcal{F}}_G$ which is a gravity-aligned East-North-Up (ENU) local Cartesian frame at the position of the first measurement.

1) Global Position Errors: The global position residual for a measurement $\mathbf{z}^j$ at a time step j can be formulated as:

$$\mathbf{e}_{\mathrm{g}}^j = \mathbf{z}^j - \left[\mathbf{C}_{GW}\left({}_{W}\hat{\mathbf{r}}_{S_j} + \hat{\mathbf{C}}_{WS_j}\,{}_{S}\mathbf{r}_A\right) + {}_{G}\mathbf{r}_W\right]. \tag{26}$$

Hereby, ${}_{S}\mathbf{r}_A$ considers the position of the GNSS antenna in the IMU sensor frame which is assumed to be known beforehand. To account for asynchronous arrival of GNSS measurements and camera frames, ${}_{W}\hat{\mathbf{r}}_{S_j}$ and $\hat{\mathbf{C}}_{WS_j}$ are predictions of the IMU poses at time step j which can be obtained from a previous state at time step i through IMU preintegration. This is visualized in Fig. 6. Compared to [48], the formulation in (26) extends the global residual by also considering a 4 DoF transformation $\boldsymbol{T}_{GW}$ between the global and a local reference frame. The error Jacobians are derived in Appendix C.

2) Global Reference Frame initialization: A reliable initialization of the global reference frame with respect to the VI reference frame is crucial. It is desirable to define an uncertainty-aware criterion for the observability of the 4-DoF extrinsic transformation between the two reference frames based on the received measurements. In a first step, an initial solution can be obtained using the SVD-based alignment method presented in [88] from correspondences of global measurements and poses in the world reference frame.

Subsequently, the reliability of this initial solution can be evaluated. For N_g received measurements, global position errors $\mathbf{e}_\mathrm{g}^j$ $(j = 1, \ldots, N_\mathrm{g})$ and the corresponding error Jacobians $\mathbf{E}_\mathrm{g}^j$ are computed based on Eqn (26). The covariance matrix $\mathbf{P}$ for the 4 DoF transformation can be estimated as the inverse of the approximate Hessian $\mathbf{H}$:

$$\mathbf{H} = \sum_{j=1}^{N_\mathrm{g}} \left({\mathbf{E}_\mathrm{g}^j}^T \mathbf{W}_\mathrm{g}^j \mathbf{E}_\mathrm{g}^j \right), \tag{27}$$

where $\mathbf{W}_\mathrm{g}^j$ is the inverse of the overall covariance matrix $\mathbf{\Sigma}_\mathrm{g}^j$ for GNSS residuals, which considers GNSS measurement covariances as well as IMU preintegration covariances (see Appendix B for more details). We examine the variance $p_{\theta\theta}$ corresponding to the yaw angle θ in $\mathbf{P}$, and define a decision threshold σ_θ^2 on whether the global reference frame is observable as $p_{\theta\theta} < \sigma_\theta^2$. When the yaw uncertainty falls below this threshold, $\boldsymbol{T}_{GW}$ can be assumed to be known and fixed.

3) GNSS Alignment: This mechanism of initializing and fixing the GNSS extrinsics enables a global alignment strategy to compensate for drift accumulated throughout potential GNSS signal dropouts. OKVIS2-X limits the computational complexity of the optimization problem by fixing states that are far in the past. Whenever GNSS residuals are added to the graph optimization problem, we check the fixation status of the last state that has an associated global position factor. Dropouts in receiving GNSS signals over a long period of time can be identified by whether the last state with a GNSS factor in the graph is already frozen. In that case, the trajectory might have accumulated a significant amount of drift since then. Eliminating this drift can be addressed in a loop-closure like global alignment approach. After detecting a dropout, a re-initialization of the GNSS extrinsics is done as described in the previous paragraph. From this re-initialization process, a new estimate for the extrinsic transformation, $\boldsymbol{T}_{GW_\mathrm{new}}$, is obtained. The discrepancy between the originally estimated $\boldsymbol{T}_{GW}$ and the newly initialized $\boldsymbol{T}_{GW_\mathrm{new}}$ can be calculated as:

$$\boldsymbol{T}_{W_\mathrm{new}W} = \boldsymbol{T}_{GW_\mathrm{new}}^{-1} \boldsymbol{T}_{GW}, \tag{28}$$

which also gives an estimate for the trajectory drift. Using this delta transformation and rotation averaging, the position and orientation error can be distributed across all the states during a GNSS dropout. After this alignment, an optimization of the full graph is triggered.

TABLE I
METHOD DEFINITION BASED ON A SENSOR CONFIGURATIONS AND ESTIMATOR CAUSALITY

	Sensors					Causality		
Methods	*Visual*	*Inertial*	*Depth*	*LiDAR*	*GNSS*	*Causal*	*Non-causal*	*Full-BA*
Ours-vi-c	✓	✓				✓		
Ours-vi-nc	✓	✓					✓	
Ours-vi-ba	✓	✓					(✓)	✓
Ours-v-⋆	✓					⋆	⋆	⋆
Ours-vig-⋆	✓	✓			✓	⋆	⋆	⋆
Ours-vid-⋆	✓	✓	✓			⋆	⋆	⋆
Ours-vidg-⋆	✓	✓	✓		✓	⋆	⋆	⋆
Ours-vil-⋆	✓	✓		✓		⋆	⋆	⋆
Ours-vilg-⋆	✓	✓		✓	✓	⋆	⋆	⋆

H. Factor Graph optimization

All of the aforementioned factors are combined in the overall minimization objective:

$$\begin{aligned} c\left(\mathbf{x}\right) = & \frac{1}{2} \sum_i \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}(i,k)} \rho_\mathrm{c} \left(\mathbf{e}_\mathrm{r}^{i,j,k^T} \mathbf{W}_\mathrm{r} \mathbf{e}_\mathrm{r}^{i,j,k} \right) \\ & + \frac{1}{2} \sum_{k \in \mathcal{P} \cup \mathcal{K} \setminus f} \mathbf{e}_\mathrm{s}^{k^T} \mathbf{W}_\mathrm{s}^k \mathbf{e}_\mathrm{s}^k + \frac{1}{2} \sum_{r \in \mathcal{P}} \sum_{c \in \mathcal{C}(r)} \mathbf{e}_\mathrm{p}^{r,c^T} \mathbf{W}_\mathrm{r}^{r,c} \mathbf{e}_\mathrm{p}^{r,c} \\ & + \frac{1}{2} \sum_{k \in \mathcal{K}} \sum_{\mathbf{p} \in \mathcal{L}_k} \rho_\mathrm{t} \left({e_\mathrm{m}^{C,k}}^2 \right) + \frac{1}{2} \sum_{b \in \mathcal{M}} \sum_{a \in \mathcal{A}_b} \sum_{\mathbf{p} \in \mathcal{L}_b} \rho_\mathrm{t} \left({e_\mathrm{m}^{a,b}}^2 \right) \\ & + \frac{1}{2} \sum_{j \in \mathcal{G}} {\mathbf{e}_\mathrm{g}^j}^T \mathbf{W}_\mathrm{g}^j \mathbf{e}_\mathrm{g}^j. \end{aligned} \tag{29}$$

Here, the set $\mathcal{K}$ contains the most recent frames as well as keyframes with observations of visible landmarks in $\mathcal{J}\left(i,k\right)$. $\mathcal{P}$ contains all posegraph frames and f denotes the most current frame. $\mathcal{C}\left(r\right) \subset \mathcal{P}$ is the set of all posegraph frames connected to a frame r. Furthermore, the set $\mathcal{L}_k$ denotes the set of all map alignment residuals 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. $\mathcal{G}$ is the set of added GNSS residuals. The Cauchy robustifier $\rho_\mathrm{c}\left(\cdot\right)$ and Tukey robustifier $\rho_\mathrm{t}\left(\cdot\right)$ are used for reprojection errors and map alignment errors.

VI. EVALUATION

The objective of this evaluation is to quantify the trajectory accuracy and mapping accuracy and completeness of OKVIS2-X with respect to state-of-the-art methods in small to large-scale scenarios. The large-scale environment (km-level traveling distance) possesses extreme challenges due to huge drift over time and memory usage in dense mapping to cover the large area. To showcase OKVIS2-X under such environments, we selected three public datasets: EuRoC [89], Hilti-Oxford [90], and VBR [1] datasets where the traveled distance spans from tens of meters with a flying drone (EuRoC), to hundreds of meters with a hand-held sensor rig (Hilti-Oxford) to 9 km for a driving car (VBR). For the proposed GNSS fusion,

TABLE II
ROOT MEAN SQUARE OF ABSOLUTE TRAJECTORY ERROR [METER] IN THE EUROC DATASET

		Methods	Sequence											Avg
			MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	
V-SLAM	C	*Ours-v-c*	0.034	0.038	0.048	0.122	0.123	0.040	0.054	0.123	0.072	0.095	1.133[4]	0.075
V-SLAM	NC	ORB-SLAM3 [3][1]	0.029	**0.019**	**0.024**	0.085	**0.052**	**0.035**	0.025	**0.061**	0.041	**0.028**	0.521[2]	**0.040**
V-SLAM	NC	*Ours-v-nc*	**0.021**	0.022	0.032	**0.076**	0.087	0.036	**0.024**	0.085	**0.027**	0.035	1.125[2]	0.045
V-SLAM	NC	*Ours-v-ba*	0.016	0.024	0.029	0.074	0.090	0.035	0.020	0.060	0.025	0.024	1.068[2]	0.040
VIO	Causal (C)	OpenVINS [5][1]	0.183	0.129	0.170	0.172	0.212	0.055	0.044	0.069	0.058	0.045	0.147	0.117
VIO	Causal (C)	Kimera2 [4][1]	0.100	0.100	0.160	0.210	0.150	0.050	**0.040**	0.100	0.060	0.070	0.190	0.112
VIO	Causal (C)	*Ours-vi-c*	**0.050**	**0.048**	**0.085**	**0.144**	**0.125**	**0.045**	0.046	**0.046**	**0.033**	**0.030**	**0.071**	**0.066**
VI-SLAM	Causal	*Ours-vi-c*	0.039	**0.032**	**0.051**	0.102	0.098	**0.036**	**0.030**	**0.031**	0.033	**0.025**	0.051	0.048
VI-SLAM	Causal	*Ours-vid-c*	**0.036**	**0.032**	0.053	**0.093**	**0.086**	0.037	0.032	**0.031**	**0.028**	0.028	**0.051**	**0.046**
VI-SLAM	Non-causal (NC)	ORB-SLAM3 [3][1]	0.036	0.033	0.035	**0.051**	0.082	0.038	**0.014**	0.024	0.032	**0.014**	0.024	0.035
VI-SLAM	Non-causal (NC)	MAVIS-SLAM [20][1]	0.024	0.025	0.032	0.053	0.075	**0.034**	0.016	0.021	0.031	0.021	0.039	0.034
VI-SLAM	Non-causal (NC)	*Ours-vi-nc*	0.023	**0.021**	**0.030**	0.067	**0.056**	0.035	**0.014**	0.020	0.019	0.017	0.022	**0.030**
VI-SLAM	Non-causal (NC)	*Ours-vid-nc*	**0.022**	0.023	**0.030**	0.068	0.058	**0.034**	**0.014**	**0.019**	**0.018**	0.016	**0.020**	**0.030**
VI-SLAM	Non-causal (NC)	*Ours-vi-ba*	0.017	0.024	0.029	0.057	0.055	0.035	0.013	0.020	0.023	0.013	0.020	0.028
VI-SLAM	Non-causal (NC)	*Ours-vid-ba*	0.016	0.024	0.029	0.062	0.053	0.035	0.013	0.019	0.020	0.012	0.022	0.028

All *Ours* report median in 10 runs.
[1] Results taken from respective papers, and OpenVINS from [20].
[2] We consider this as a failed sequence, thus exclude this in the average.

we further added an evaluation in a challenging sequence from the GVINS dataset [50] including indoor-outdoor transitions and cluttered environments. All evaluations were performed in a serial manner, ensuring no frame drops. We account for the time offset between camera and IMU, if known (e.g. in the Hilti-Oxford dataset), as well as time offsets for both, LiDAR and GNSS, with respect to the IMU.

A. Evaluation metrics

We evaluate several variations of our method to clearly show the effectiveness of multi-modality depending on the sensor configuration as well as the estimator causality. On the one hand, in Table I, *Visual* includes the reprojection (11) and relative posegraph residuals (13), *Inertial* indicates preintegration residuals (12), *Depth-network* is based on the depth network fusion for the submap alignment residuals (23), *LiDAR* uses LiDAR point clouds for the submap alignment residuals (23), and *GNSS* includes GNSS residuals (26). On the other hand, *Causal* only takes measurements up to the current time, *Non-causal* is the final loop-closed trajectory, and *Full-BA* refines the *Non-causal* trajectory in the entire states by turning relative pose graph edges to reprojection edges. For instance, *Ours-vi-c* means a visual-inertial configuration with the causal evaluation.

To quantify the accuracy of the estimated poses, the estimated trajectory is aligned in $SE(3)$ to the ground-truth before evaluation. We report trajectory accuracy as root mean square error (RMSE) of the absolute trajectory error (ATE) in EuRoC and VBR datasets. In the Hilti-Oxford dataset, we report the predefined score, where the position error $[1, 10]$ cm is scaled to $[0, 100]$ points. To evaluate mapping accuracy, submap meshes are reconstructed using marching cubes [91] and combined using submap poses. Estimated and ground-truth vertices are downsampled with a voxel size of 1cm^3, then we perform a point-to-plane ICP alignment between both sets of vertices. We report accuracy as an average distance from all estimated vertices to the ground-truth within 0.2 m. The completeness is defined as a fraction of ground-truth vertices that are within 0.2 m to the estimated vertices.

B. EuRoC dataset

This dataset provides stereo images and IMU measurements recorded by a drone, the ground-truth trajectory and mm-level accurate point clouds of the Vicon room [89]. We use the stereo pair for our stereo network and left images for our MVS network. Table II summarizes ATE where all competitors also use a stereo or stereo-inertial configuration. We additionally implemented V-SLAM, *Ours-v*, to show versatility of OKVIS2-X where the IMU preintegration term is replaced by a constant velocity model. OKVIS2-X shows competitive trajectory accuracy when compared to ORB-SLAM3. It is worth noting that the V203 sequence is challenging for V-SLAM due to the motion blur, image dropouts, and significantly varying exposure. This is the reason why OKVIS2-X and ORB-SLAM3 present a large trajectory error on that specific sequence.

However, we resolve this large error in a stereo-inertial configuration. We compare the VIO implementation of *Ours-vi* without loop-closure for fair comparison to odometry approaches, OpenVINS [5] and Kimera2 [4]. Our method decreases the localization error by 41% when compared to competitors. In the SLAM implementation with loop-closure, *Ours-vi* outperforms state-of-the-art methods in the non-causal evaluation. It is worth noting that incorporating loop-closures

TABLE III
MESH ACCURACY [M] AND COMPLETENESS [%] WITH 0.2M THRESHOLD IN THE EUROC DATASET

	Simplemapping[1][58]	*Ours-vid (Mono)*	*Ours-vid (Stereo)*	*Ours-vid*	Simplemapping[1][58]	*Ours-vid (Mono)*	*Ours-vid (Stereo)*	*Ours-vid*
	Accuracy				*Completeness*			
V101	0.071	0.050	0.039	**0.032**	40.58	43.98	42.70	**44.77**
V102	0.077	0.043	0.035	**0.031**	55.07	56.45	56.17	**58.38**
V103	0.086	0.062	0.040	**0.039**	47.74	61.65	64.59	**67.77**
V201	0.062	0.066	0.068	**0.047**	41.93	40.51	38.53	**43.82**
V202	0.065	0.047	0.064	**0.041**	56.37	58.16	48.90	**60.01**
V203	0.068	0.062	0.068	**0.044**	**62.58**	60.20	52.72	61.76
Avg	0.072	0.055	0.052	**0.039**	50.71	53.49	50.60	**56.09**

[1] Results obtained ourselves with the same metric.

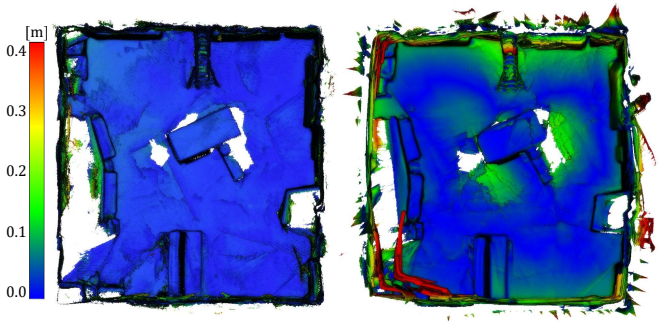

Fig. 7. Reconstructed mesh with mesh-to-point error encoding in V102 sequence. (Left) *Ours-vid* and (Right) Simplemapping [58].

yields more accurate live trajectories, reducing 1.8 cm on average when compared to our VIO variant. Furthermore, our method with the depth networks, *Ours-vid*, improves the baseline, *Ours-vi*, by adding submap alignment factors in the causal evaluation. We obtained the same average accuracy when compared to the baseline in the non-causal evaluation. This indicates that the depth fusion could not achieve 3 cm-level accuracy (0.04% of the trajectory length), which is already quite accurate.

Since our method tightly couples localization and volumetric mapping, we also evaluate the mapping accuracy in Table III. To filter out noisy surfaces, we only mesh surfaces with gradients larger than a threshold and neighbors that have been observed at least three times. To ensure a fair comparison with SimpleMapping [58] in a monocular-inertial setup, we implement *Ours-vid (Mono)*, where only the left camera images are used as input to the estimator and the MVS network. Also, we present *Ours-vid (Stereo)* where only depths from the stereo network are integrated for mapping. *Ours-vid* achieves the highest accuracy and completeness, while *Ours-vid (Mono)* also outperforms the competitor. We can clearly see the depth fusion of temporal and stereo images improves the mapping quality in Table III. Fig. 7 compares the accuracy where *Ours-vid* improves mesh accuracy especially along the edge of the room by properly addressing the uncertainty.

C. Hilti-Oxford dataset

This dataset provides images from 5 cameras, IMU measurements, 32-channel LiDAR point clouds captured by a handheld device, sparse ground-truth positions, and mm-accurate dense point clouds [90]. We use all 5 cameras for the visual-inertial estimator, 2 front cameras for the stereo network, and the front left camera for the MVS network. We calibrate camera-IMU extrinsic parameters online, and the close inspection of the online calibration will be presented in Sec. VI-F.

Table IV summarizes the localization scores in the challenge sequences. Considering the nature of the leaderboard, we compare best scores in 3 runs of our method to competitors. To give insight about the deviation from run-to-run, we also report the median scores in brackets. For the visual-inertial configuration, our non-causal evaluation with the final bundle adjustment significantly improves the position accuracy while outperforming published competitors in the leaderboard. By having additional residuals from submap alignment, *Ours-vid-ba* further improves the accuracy. exp03 was challenging due to a low light condition which leads temporarily to completely black images in the front cameras, as shown in Fig. 8. Given that we only use front cameras for dense mapping, *Ours-vid-ba* records 1.2 m position error, which is relatively higher than in the other sequences, in the control points. However, the score of *Ours-vid-ba* corresponds to 7.0 cm position error on average excluding the challenging sequence, exp03.

Our Visual-Inertial-LiDAR configuration, *ours-vil-⋆*, shows a significant improvement in both the causal and the refined estimates for almost every sequence, especially also robustifying the system in the visually challenging sequence exp03 (Fig. 8). In terms of the actual localization accuracy, it reduces the average position error to 4.1 cm (2.8 cm when excluding exp07 which has an accuracy around 11 cm) which is in the range of the mapping resolution. We also compare our system to the top-ranked published competitors. Ours ranks top amongst the published Visual-Inertial-LiDAR systems, on average outperforming FAST-LIVO2 [9] and VILENS [11] while being outperformed only by Wildcat [68], a purely LiDAR-Inertial SLAM system. The main bottleneck of our

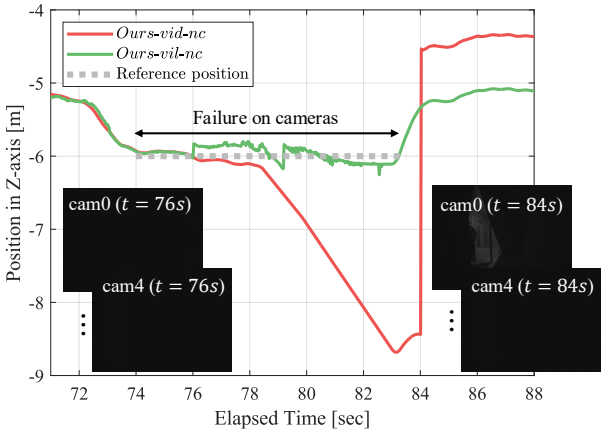

Fig. 8. Failure case in exp03 in the Hilti-Oxford dataset where the LiDAR compensates the vision failure in a dark room.

TABLE IV
LOCALIZATION SCORES IN THE HILTI-OXFORD DATASET

		Methods	Sequence								Total
			exp01	exp02	exp03	exp07	exp09	exp11	exp15	exp21	
Visual-inertial	Causal	OpenVINS [5][1]	0.00	0.00	0.00	1.67	8.75	8.00	4.44	0.00	22.86
		Ours-vi-c	10.77 (7.69)	8.18 (10.00)	0.00 (0.00)	0.00 (0.00)	11.25 (6.25)	10.00 (6.00)	10.00 (11.11)	0.00 (0.00)	50.20 (41.05)
		Ours-vid-c	**56.92** (26.15)	18.18 (4.55)	0.00 (0.00)	0.00 (1.67)	4.38 (1.88)	4.00 (8.00)	7.78 (4.44)	0.00 (0.00)	91.26 (46.69)
	Non-causal	BAMF-SLAM [92][2]	15.38	4.55	8.24	1.67	1.25	2.00	7.78	0.00	40.86
		ORB-SLAM3 [3][1]	3.85	1.82	10.59	0.00	0.00	6.00	0.00	0.00	22.25
		MAVIS SLAM [20][2]	40.00	16.82	**16.47**	11.67	15.00	20.00	26.67	**4.00**	150.62
		Ours-vi-ba	30.00 (36.92)	**40.91** (30.45)	12.35 (0.00)	3.33 (1.67)	16.88 (17.50)	**44.00** (28.00)	20.00 (20.00)	0.00 (0.00)	167.47 (135.54)
		Ours-vid-ba	53.08 (30.00)	34.55 (37.27)	0.00 (0.00)	**13.13** (8.33)	**23.75** (23.75)	22.00 (22.00)	**28.89** (26.67)	2.00 (0.00)	**177.59** (148.02)
(V)-LiDAR	Causal	FAST-LIVO2 [9][2]	**84.62**	70.00	44.71	31.67	7.50	92.00	24.44	48.00	402.93
		Ours-vil-c	63.08 (50.77)	33.64 (37.27)	1.18 (7.06)	0.00 (0.00)	11.88 (7.50)	68.00 (62.00)	30.00 (34.44)	38.00 (34.00)	245.76 (233.05)
	Non-causal	VILENS [11][2]	69.23	49.09	40.00	16.67	5.00	68.00	17.78	60.00	325.77
		Ours-vil-ba	63.85 (72.31)	51.82 (51.36)	49.41 (51.18)	1.67 (0.00)	26.25 (24.38)	**100.00** (84.00)	**61.11** (61.11)	**84.00** (68.00)	438.10 (412.33)
		Wildcat [68][2,3]	**84.62**	**84.55**	**90.59**	**45.00**	**44.38**	84.00	46.67	**84.00**	**563.79**

All *Ours* report the best (median) scores in 3 runs based on the total score.
Due to the nature of the challenge, best (bold) and second (underlined) scores are highlighted per sensor configuration without distinction of causal or non-causal methods.
[1] Results, best scores in 3 runs, obtained from us based on the open-source in stereo-inertial setup.
[2] Results taken from the leaderboard [93].
[3] Wildcat uses LiDAR-Inertial only.

TABLE V
MESH ACCURACY [M] AND COMPLETENESS [%] WITH 0.2M THRESHOLD IN THE HILTI DATASET

	Ours-vid	*Ours-vil*	*Ours-vid*	*Ours-vil*
	Accuracy		*Completeness*	
exp04	0.040	**0.030**	59.15	**77.90**
exp05	0.040	**0.031**	64.04	**76.94**
exp06	0.046	**0.027**	69.63	**80.15**
Avg	0.042	**0.029**	64.27	**78.33**

system is in exp07, a long corridor which is a degenerative case for the LiDAR due to the lack of characteristic geometry in the direction of movement.

In terms of mapping, Table V reports the accuracy and completeness with the network depth and LiDAR in exp04, exp05, exp06 where ground-truth point clouds are available. We obtain fairly accurate mapping (4.2 cm) by the depth fusion which was only fine-tuned in a synthetic dataset [83]. We further improve the accuracy and completeness by replacing depth networks with a LiDAR. The *far plane*, the maximum valid depth to integrate measurements, was set to 3 m and 30 m for the depth fusion and LiDAR, respectively. Fig. 9 visualizes reconstructed meshes in exp06 where geometric structures captured by our depth network are clearly visible, and fusion with a LiDAR further increases the completeness.

D. VBR: A Vision Benchmark in Rome

The VBR dataset [1] presents an extensive sensor suite that allows us to evaluate OKVIS2-X with different configurations. We consider VBR an interesting dataset because it contains long sequences, proving the scalability of OKVIS2-X. The recorded sequences are in real-world environments, posing additional challenges such as: dynamic entities, non-constant illumination conditions, structured and unstructured environments. This dataset contains images from a stereo camera, IMU measurements, LiDAR point clouds and ground-truth poses for quantitative evaluation. We found out that the MVS network outputs noisy depths in the open-sky due to the depth plane parametrisation which is not in inverse depth. In VBR, specifically, we only employ the stereo network. To further improve the stereo disparity accuracy, we finetuned our uncertainty-augmented network in a 2-level image pyramid followed by the additional refinement step [79]. In this dataset, we evaluate the ATE of OKVIS2-X in all its multi-sensor configurations and compare it to the state-of-the-art systems ORB-SLAM3 [3], OpenVINS [5] and FAST-LIVO [40]. To ensure that FAST-LIVO processes all measurements, the rosbags were played at a real-time rate of 0.2.

Table VI presents the trajectory accuracy of the evaluated SLAM systems in the VBR dataset, all of them tested with the same calibration parameters. The table presents the median trajectory error from the three evaluation runs. Since VBR presents very long sequences, up to 9 km, we set the submap resolution to either 10 or 20 cm, depending on the sensor configuration and scale of the sequence. The sensor far plane

TABLE VI
ROOT MEAN SQUARE OF ABSOLUTE TRAJECTORY ERROR [METERS] IN THE VBR DATASET.

		Methods	Sequence								Avg
			Colosseum	Diag	Pincio	Spagna	Campus0	Campus1	Ciampino0	Ciampino1	
Visual-inertial	Causal	OpenVINS [5]	20.827	24.628	6.928	10.023	18.343	4.719	109.777	34.921	28.770
		Ours-vi-c	1.922	**1.156**	**3.368**	3.618	**3.065**	2.631	**3.721**	**2.971**	**2.807**
		Ours-vid-c	**1.771**	1.173	3.673	**3.569**	3.084	**2.552**	3.728	3.002	2.819
	Non-causal (NC)	ORB-SLAM3 [3][1]	7.713	1.834	**1.594**	3.108	8.023[2]	7.289	-[2]	9.443[2]	5.572[2]
		Ours-vi-nc	1.541	1.026	3.301	**0.605**	**3.026**	2.499	3.490	**2.847**	2.292
		Ours-vid-nc	**1.450**	**0.452**	3.150	0.744	3.037	**2.482**	**3.472**	2.889	**2.210**
		Ours-vi-ba	1.613	0.931	3.278	0.662	3.017	2.471	3.358	2.920	2.281
		Ours-vid-ba	1.441	0.346	3.199	0.807	3.016	2.451	3.391	2.995	2.206
VI-LiDAR	Causal	FAST-LIVO [40][1]	6.459	1.013	**2.042**	3.963	**1.259**	**0.434**	3.080	**1.548**	2.475
		Ours-vil-c	**1.517**	**0.900**	2.861	**0.691**	1.955	1.634	**1.716**	2.892	**1.771**
	NC	*Ours-vil-nc*	0.275	0.906	1.613	0.326	1.529	0.868	0.529	0.582	0.829
		Ours-vil-ba	0.337	0.891	1.745	0.345	1.495	0.875	0.502	0.606	0.850
Trajectory length [km]			1.451	1.037	1.277	1.562	2.735	2.925	9.008	5.200	3.149

All methods report median in 3 runs.
[1] Results obtained from us based on the open-source implementation.
[2] System failure: Reported numbers are obtained from less than 3 successful runs (out of min. 5 runs) if any available.

is limited to 30 m for the LiDAR and 10 or 15 m for the stereo network in the handheld or driving sequences.

In the visual-inertial sensor configuration OKVIS2-X demonstrates a superior performance in most sequences, with the version that leverages learned depth being the most accurate in the non-causal evaluation. Since OpenVINS does not support loop-closures, its accuracy naturally degrades in large-scale scenarios with loops. In the VI-LIDAR sensor configuration, OKVIS2-X demonstrates an overall superior performance over FAST-LIVO already in the causal evaluation, with an average error of 1.771 m (0.06% of the trajectory). Our system's support for loop-closures leads to another significant improvement. Figures 1 and 9 show examples of globally consistent submap meshes.

OKVIS2-X demonstrates a superior performance over the compared systems. Furthermore, OKVIS2-X showcases its robustness in the presence of dynamic entities, especially pronounced in `Colosseum` or `Spagna`, or in case of corrupted data, as we found that the driving sequences `Campus*` and `Ciampino*` contain episodes of missing IMU measurements throughout the sequence. In these cases, the other systems suffered from degrading accuracy or system failure.

E. Dropout-Tolerant Fusion of Global Position Measurements

To demonstrate the versatility and modularity of OKVIS2-X for different sensor setups, and to demonstrate the capability to fuse global position measurements, we conducted two more studies. The first one is on the VBR sequence `Campus1`, one of the driving sequences. As the dataset does not provide geodetic GNSS measurements, we simulated RTK-GNSS measurements by applying Gaussian noise of 1 cm in horizontal and 2 cm in vertical direction to the ground-truth trajectory. Furthermore, we simulated a GNSS dropout of 75 seconds and a corresponding trajectory length of 450 m.

TABLE VII
DROPOUT-TOLERANT GNSS FUSION: ATE RMSE[M] IN VBR CAMPUS1.

	Ours-vi	*Ours-vig*	*Ours-vid*	*Ours-vidg*	*Ours-vil*	*Ours-vilg*
Causal	2.631	1.119	2.552	1.099	1.634	**0.328**
(during Dropout)	-	(2.766)	-	(2.737)	-	(0.701)
Non-Causal	2.499	0.751	2.482	0.750	0.868	**0.177**
(during Dropout)	-	(1.974)	-	(1.972)	-	(0.423)
Final BA	2.471	0.661	2.451	0.636	0.875	**0.169**
(during Dropout)	-	(1.719)	-	(1.655)	-	(0.386)

All methods report median in 3 runs.

To mimic a realistic scenario, the dropout was chosen in an area of the sequence that has more narrow streets and higher buildings than on average. The main target of this evaluation is to give a proof-of-concept for the fusion of global position measurements and especially to showcase the effectiveness of the global, loop-closure-like, alignment strategy to handle potential signal outages over longer periods of time, presented in Sec. V-G3. Table VII reports the ATE for all possible sensor setups (*vig*, *vidg* and *vilg*) for the whole trajectory as well as during the GNSS signal outage only. Obviously, the overall trajectory error is significantly reduced with the fusion of GNSS measurements in all configurations. Furthermore, it can be seen that the accuracy of our Visual-Inertial-LiDAR configuration helps to minimize the drift during the GNSS dropout. Loop-closures and a final BA enable us to estimate a final trajectory with an RMSE ATE of only 16.9 cm for a trajectory of almost 3 km. Qualitative results on the reconstruction and trajectory accuracy are visualized in Fig. 10.

Even though the system was originally designed to fuse intermittent high-accuracy (RTK-) GNSS measurements, it can still also benefit from lower-grade GNSS measurements.

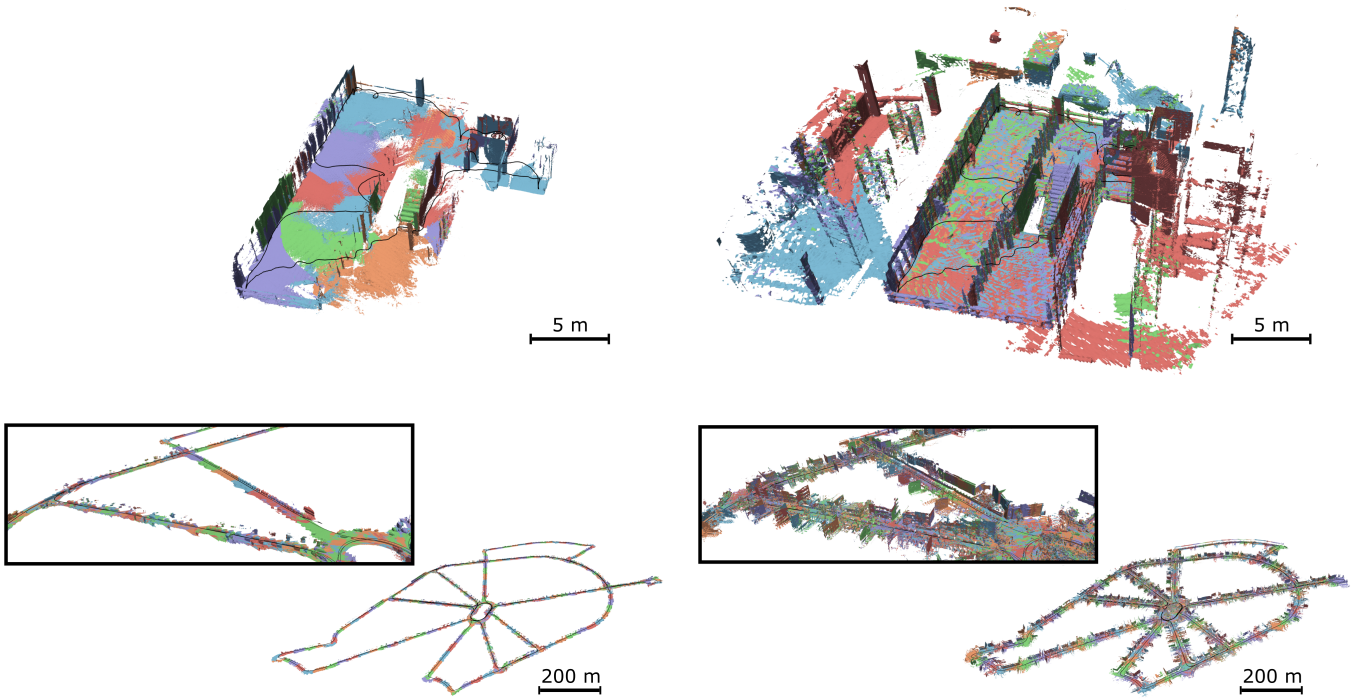

Fig. 9. 3D reconstructions obtained from the volumetric submaps of OKVIS2-X. The left column showcases reconstructions when leveraging learned-depth while the ones on the right use a LiDAR sensor. The top row is obtained from the Hilti22 `exp06` sequence, the bottom row is extracted from VBR's `Ciampino0` sequence. A line with a distance indicates the different scales of the sequences.

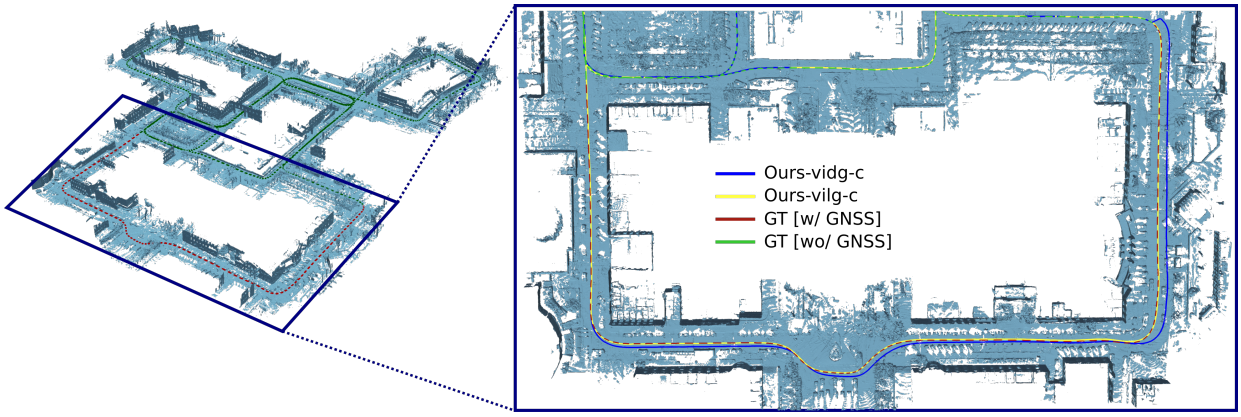

Fig. 10. Reconstruction of the `Campus1` (VBR) is visualized as overlay of all submaps. Causal Trajectory Estimates for our Visual-Inertial-Depth (blue) and Visual-Inertial-LiDAR (yellow) configurations with GNSS fusion are drawn together with ground-truth measurements in GNSS-accessible (dashed green) and GNSS-denied regions (dashed red). While the LiDAR-based setup aligns accurately with the ground-truth even in GNSS-denied areas, the vision-based configuration drifts significantly. The visible jump in the trajectory at the end of the GNSS dropout highlights the global alignment strategy of our approach.

To demonstrate that, we evaluated OKVIS2-X on one of the challenging dataset sequences provided by GVINS [50]. This dataset consists of a stereo camera, an IMU and a ZED-F9P GNSS sensor. The authors provide RTK solutions as well as raw GNSS measurements (e.g. pseudoranges, Doppler measurements, etc.). We chose to evaluate on the nearly 3 km `complex_environment` sequence due to its numerous challenges for different sensor modalities: featureless images in very bright and dark areas as well as high corruption or complete unavailability of GNSS signals in cluttered environments or indoors (see Fig. 11). Using RTKLIB [94], we computed the Single Point Positioning (SPP) solution from raw measurements and fused it into the factor graph optimization. We evaluate the RMSE ATE after 6DoF alignment with the RTK solution as a reference in areas with a valid RTK fix. Even with the highly imprecise SPP solution (RMSE ATE of 19.99 m), the proposed GNSS fusion significantly reduced the error of the visual-inertial baseline from 23.72 m to 12.51 (causal), 8.47 (non-causal) and 5.12 m (final BA). Note that in this evaluation, visual loop closures have been disabled for a fairer comparison with GVINS [50], which achieves an RMSE ATE of 4.45 m. This shows that, despite the simpler fusion strategy of GNSS measurements with less precise data (SPP only, and no Doppler velocities), the overall accuracy is still

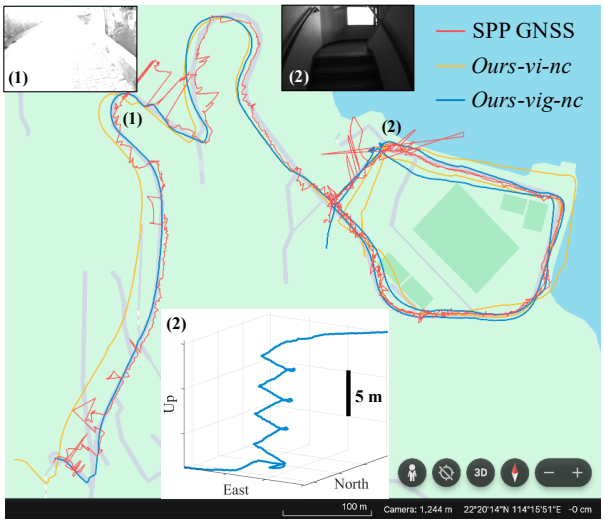

Fig. 11. SPP Solution of GNSS and OKVIS2-X estimated trajectories overlaid in Google Maps in the `complex_environment` in the GVINS dataset.

TABLE VIII
ABLATION STUDY IN ONLINE EXTRINSIC CALIBRATION: ATE RMSE [M] IN THE HILTI-OXFORD DATASET

	Online-calibration	*Post-calibration*	`exp04`	`exp05`	`exp06`	Avg
Ours-vi-c	X	X	0.198	0.192	0.085	0.158
	✓	X	0.070	0.058	0.067	0.065
	X	✓	**0.058**	**0.044**	**0.048**	**0.050**
Ours-vi-nc	X	X	0.157	0.169	0.090	0.139
	✓	X	0.036	0.049	**0.041**	0.042
	X	✓	**0.033**	**0.026**	0.042	**0.034**
Ours-vi-ba	X	X	0.096	0.088	0.061	0.082
	✓	X	0.023	**0.022**	0.050	**0.032**
	X	✓	**0.020**	0.028	**0.048**	**0.032**

All methods report median in 3 runs.

competitive. Fig. 11 shows the estimated trajectories overlaid with the SPP solution. It is clearly visible that, compared to the visual-inertial baseline, the fused estimate aligns well with a road layout in areas of good coverage but stays consistent even in areas of highly corrupted GNSS signals.

F. Ablation study: Online extrinsic calibration

To show effectiveness of the IMU-to-camera extrinsic online calibration, Table VIII reports an ablation study of how online calibration affects the estimated trajectory accuracy where *Online-calibration* indicates an online optimization of the extrinsic parameters, and *Post-calibration* means that the extrinsic parameters are initialized with our calibrated extrinsics and kept constant. We chose `exp04`, `exp05`, and `exp06` sequences of the Hilti-Oxford dataset where the dense ground-truth trajectory is available. In the live trajectory, we set 1mm standard deviation for the prior translation and $0.29°$ for the prior orientation for all five cameras. In the full bundle adjustment, where well-initialized poses and landmarks are available, we apply weaker constraints on the standard deviation to give more freedom in optimizing extrinsics: 3mm and $0.92°$. It can be clearly seen that the online calibration improves the trajectory accuracy in all sequences in Table VIII when compared to the case without calibration. To show the validity of the optimized extrinsic parameters, we evaluate additional runs with our extrinsic parameters without online calibration. Interestingly, our *Post-calibration* improves the online calibration results. This indicates that our estimated extrinsics are close to the ground-truth extrinsics. Furthermore, to show the robustness to the initial error, we perturbed the initial extrinsic parameters from our extrinsics after the full-BA calibration. with an uniform distribution of $\pm 1\,\mathrm{cm}$ and $\pm 0.3°$ in `exp06`. Considering 10 runs, the extrinsics record a standard deviation of only $0.08\,\mathrm{cm}$ and $0.02°$ at the end of the sequence. This indicates successful convergence of the perturbed extrinsics to the reference in our online calibration.

TABLE IX
MAXIMUM MEMORY [GB] REQUIRED FOR DIFFERENT BENCHMARKED METHODS AND DATASETS

	Ours-vi	*Ours-vid*	*Ours-vil*	*FAST-LIVO*
`HILTI-Oxford: exp06`	4.86	11.62	11.51	3.97
`VBR: Campus1`	4.50	10.8	25.5	15.0

G. Configurable parameter study

We thoroughly investigate trajectory errors and real-time graph optimization timings of vid and vil modes with respect to the number of keyframes (kf), imu frames (if), submap factors (ns), and the voxel resolution (res), as shown in Fig. 13. For the submap-based alignment factors, we set the number of map-to-map factors to a multiple of the number of frame-to-map factors (i.e. $5ns$).

For the vid configuration in Fig. 13(a), increasing the number of keyframes and IMU frames results in longer processing times, while the accuracy is saturated at the proposed parameter configuration (gray bars). Similarly, adding more submap factors and using finer voxel resolutions increases computation time but does not necessarily improve accuracy, as the estimated trajectory already exceeds the accuracy of the network depth. In contrast, in the vil configuration, Fig. 13(b), a higher number of submap factors enhances the accuracy. However, finer voxel resolutions do not always yield better trajectory estimates due to noise in the occupancy fields and gradients. Additionally, more keyframes and IMU frames do not significantly improve accuracy, as submap factors are dominant in vil mode. Overall, our current parameter configuration shows a good balance between accuracy and computation time in both, depth and LiDAR, setups.

H. Timing and memory

In this section we demonstrate that our system can run in realtime and that the memory efficiency of the overall architecture and its modules make this system suitable to perform large-scale SLAM. We first present in Fig. 12 the running times of the main components of our SLAM system: visual frontend, realtime graph optimization, submapping interface, and the depth network. All timings were measured in

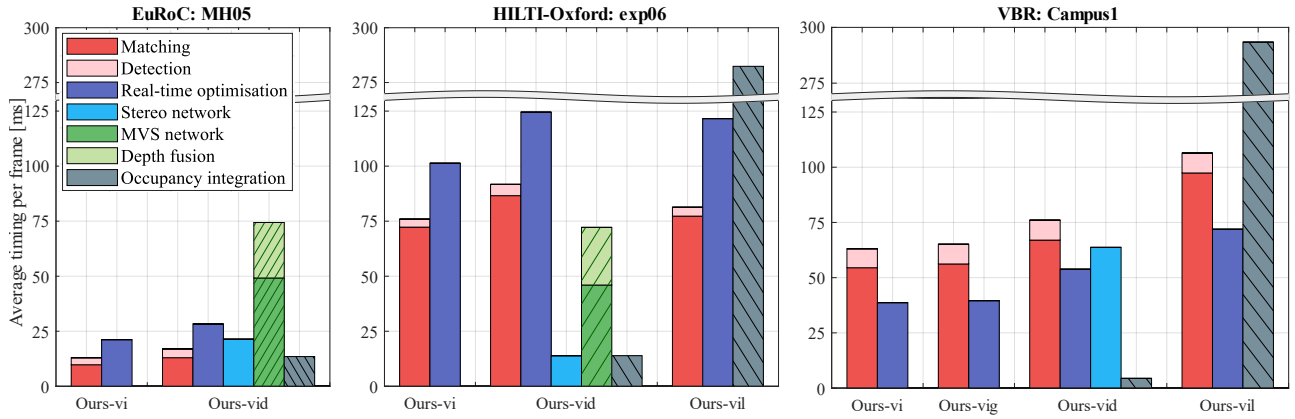

Fig. 12. Timing breakdown of OKVIS2-X with different configurations in ms per frame. Each bar represents a thread, and multiple threads run in parallel, therefore, the total timing is not the sum of each bar. The networks, depth fusion, and map integration do not necessarily have to be performed for every frame or LiDAR scan. Also, HILTI-Oxford uses five cameras on the rig, and VBR employs high-resolution cameras amenable to higher keypoint numbers.

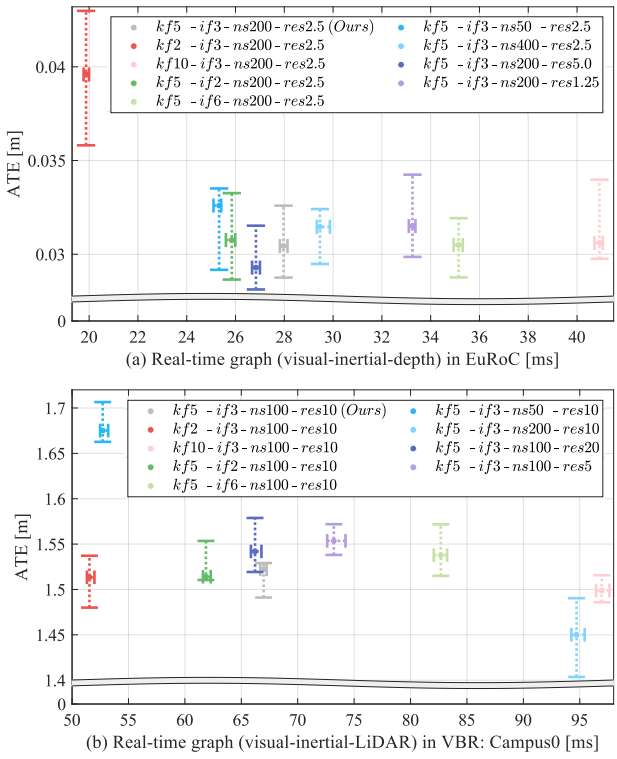

Fig. 13. Parameter study on OKVIS2-X (a) across all sequences in EuRoC and (b) `Campus0` in the VBR dataset with the number of keyframes kf, imu frames if, submap factors ns, and the voxel resolution res in cm. Each vertical bar represents 10, 50, 90 precentiles in non-causal ATE, while the horizontal bar means 10, 50, 90 percentiles in the real-time optimization timings per frame. The statistics comes from 10 independent runs.

a Desktop with i7-13700 CPU and RTX-3080 10 GB GPU. Thanks to our multi-threaded approach and by leveraging internal queues, we computationally decouple our submapping interface from the state estimator. Therefore, not all running times are sequential and a direct summation of these does not correspond to the overall time needed to process a multi-frame.

In `MH05` of EuRoC, our visual-inertial configuration, *Ours-vi*, runs up to 47 Hz, and with the depth network configuration, the MVS thread can process 8-view images at 13 Hz. Note that the stereo network and depth fusion with the MVS network run on a GPU. Furthermore, the networks and map integration do not necessarily have to process every frame. In `exp06` of the Hilti-Oxford dataset, *Ours-vi* can process all five incoming images at 10 Hz, and *Ours-vid* processes at 8 Hz given that the bottleneck is the realtime optimization graph. The realtime estimator of *Ours-vil* runs at 8 Hz, while the submapping interface runs at 3.5 Hz due to a high volume of LiDAR points. In `Campus1`, *Ours-vi* runs at 16 Hz, and the global position fusion from GNSS only adds negligible overhead. Note the use of high-resolution cameras in VBR leads to higher numbers of extracted keypoints. As mentioned in Sec. VI-D, we adopted additional refinement steps [79] in the stereo network which increases the inference time. Overall, our uncertainty-aware depth networks are capable of realtime processing at a minimum of 13 Hz across all datasets. However, we emphasize that many parameters offer a trade-off to exchange a little accuracy for faster runtime, which allows us to run the system including networks, depth or LiDAR integration also on a drone featuring an NVIDIA Orin NX (see supplementary video).

To benchmark the timings, we measured total elapsed time (wall time) and divided it by the total number of processed frames for both *Ours-vi* and ORB-SLAM3. Please note that due to the different algorithmic structures of OKVIS2-X and ORB-SLAM3, a direct timing comparison of individual modules is not possible, therefore we regard a wall time comparison as the fairest metric. Our method outperforms in efficiency on `MH05` (38.1 vs. 64.7 ms) and performs on par on `exp06` (112.3 vs. 116.1 ms). In `Campus1`, although our method shows a higher runtime (107.3 vs. 69.1 ms), it achieves significantly better accuracy (2.499 m) compared to ORB-SLAM3 (7.289 m), making the trade-off worthwhile.

In Table IX, we present the maximum memory required of OKVIS2-X with different sensor configurations and in different datasets. As expected, leveraging our dense volumetric map increases the memory requirements over landmark-based VI SLAM. When compared with FAST-LIVO [40], our

approach demands more memory, since our maps store free and occupied regions volumetrically, while FAST-LIVO stores downsampled point clouds only. However, our map representation can be directly used for downstream tasks, notably safe autonomous navigation of guaranteed free-space. Nonetheless, the memory requirements of the proposed approach can be fulfilled with consumer grade computers, even on-board our drone. In `exp06`, *Ours-vid* and *Ours-vil* require a similar memory usage since *Ours-vid* generates more submaps, due to our overlap strategy, even though the volume mapped is larger with the LiDAR sensor. In terms of GPU memory usages, *Ours-vid* only takes 3.51 GB for the MVS and stereo networks.

VII. Conclusion

In this paper, we have presented a multi-sensor SLAM system, OKVIS2-X, a substantial extension to the sparse landmark-based OKVIS2 [2]. To achieve highly robust and accurate localization and mapping in large-scale environments, we introduced volumetric occupancy submapping that is tightly-coupled with the state estimator. Furthermore, we presented a unified framework to embrace multi-modality including visual, inertial, depth from neural networks, point clouds from a LiDAR, and global position measurements from a GNSS receiver. We have thoroughly evaluated OKVIS2-X in multiple benchmark datasets in terms of trajectory and mapping accuracy, memory usage, and timings where we have created a new baseline over the state-of-the-art SLAM systems. Notably, we could map dense volumetric occupancy while having sub-meter trajectory accuracy in a 9 km driving sequence with a visual-inertial-LiDAR configuration.

In the future, we would like to push boundaries further by incorporating online calibration of sensor time offsets, LiDAR and GNSS antenna extrinsics, multi-session relocalization, open-vocabulary scene understanding, dynamic objects, and change detection over time in complex and large-scale environments. Furthermore, adopting an on-demand local online map saving and loading scheme would offer the potential for even finer occupancy resolutions and practically unlimited map scalability.

Appendix

A. Error State Definition

We generally use the perturbation $\delta\boldsymbol{\chi}_T = [\delta\mathbf{r}, \delta\boldsymbol{\alpha}]$ for poses around linearisation points for translation $\bar{\mathbf{r}}$ and orientation $\overline{\mathbf{C}}$:

$$\mathbf{r} = \bar{\mathbf{r}} + \delta\mathbf{r}, \qquad \mathbf{C} = \mathrm{Exp}\left(\delta\boldsymbol{\alpha}\right)\overline{\mathbf{C}},$$

where an additional subscript is used to indicate the respective transformation. With this, 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.

B. GNSS Covariances

The overall covariance matrix $\boldsymbol{\Sigma}_{\mathrm{g}}^j$ for the GNSS residual in Eqn. (26) is composed of the GNSS measurement covariance matrix $\boldsymbol{\Sigma}_{\mathrm{g}_0}^j$ and IMU pre-integration covariance denoted by $\boldsymbol{\Sigma}_x^j$. Thus, $\boldsymbol{\Sigma}_{\mathrm{g}}^j$ can be derived by:

$$\boldsymbol{\Sigma}_{\mathrm{g}}^j = \boldsymbol{\Sigma}_{\mathrm{g}_0}^j + \mathbf{J}\boldsymbol{\Sigma}_{\mathbf{x}}^j\mathbf{J}^T.$$

Here, $\mathbf{J}$ denotes the Jacobian of $\mathbf{e}_{\mathrm{g}}^j$ with respect to $\hat{\boldsymbol{\chi}}_{T_{WS_j}}$.

C. GNSS Jacobians

The Jacobians of the global position residuals in Eqn. (26) with respect to the pose error states of the predicted pose $\hat{\boldsymbol{T}}_{WS}$ based on the IMU measurements and the relative pose between the GNSS and world frame $\boldsymbol{T}_{GW}$ are given by:

$$\frac{\partial\mathbf{e}_{\mathrm{g}}^j}{\partial\delta\hat{\boldsymbol{\chi}}_{T_{WS_j}}} = \begin{bmatrix}-\overline{\mathbf{C}}_{GW} & \overline{\mathbf{C}}_{GW}\left[\overline{\mathbf{C}}_{WS}\mathbf{r}_A\right]^\times\end{bmatrix},$$
$$\frac{\partial\mathbf{e}_{\mathrm{g}}^j}{\partial\delta\boldsymbol{\chi}_{T_{GW}}} = \begin{bmatrix}-\mathbf{I}_{3\times3} & \left[\overline{\mathbf{C}}_{GW}\left({}_W\bar{\mathbf{r}}_S + \overline{\mathbf{C}}_{WS}\mathbf{r}_A\right)\right]^\times\end{bmatrix}.$$

The predicted poses are not estimated states in the posegraph optimization. Since the predicted measurement is a function of the estimated states at time step i, i.e. $\hat{\mathbf{x}}^j = f(\mathbf{x}^i)$, we can actually optimize $\mathbf{x}^i$ in the estimated states. The Jacobian with respect to $\mathbf{x}^i$ can be computed using the chain rule and standard IMU pre-integration Jacobians, see [87].

D. Submap Alignment Jacobian

With the dense alignment residuals (Eqn. (23)) 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:

$$\frac{\partial e_{\mathrm{m}}^{a,b}}{\delta\boldsymbol{\chi}_{T_{WS_k}}} = \frac{\partial e_{\mathrm{m}}^{a,b}}{\partial_{S_a}\mathbf{p}}\frac{\partial_{S_a}\mathbf{p}}{\delta\boldsymbol{\chi}_{T_{WS_k}}},$$

with $k \in \{a, b\}$. The individual Jacobians are:

$$\frac{\partial e_{\mathrm{m}}^{a,b}}{\partial_{S_a}\mathbf{p}} = \frac{\nabla L}{\sqrt{\frac{L_{\min}^2}{9} + |\nabla L|^2\,\sigma_d^2}}$$
$$\frac{\partial_{S_a}\mathbf{p}}{\delta\boldsymbol{\chi}_{T_{WS_a}}} = \overline{\mathbf{C}}_{S_aW}\begin{bmatrix}-\mathbf{I}_3 & \left[\overline{\mathbf{C}}_{WS_bS_b}\mathbf{p} + {}_W\bar{\mathbf{r}}_{S_b} - {}_W\bar{\mathbf{r}}_{S_a}\right]^\times\end{bmatrix}$$
$$\frac{\partial_{S_a}\mathbf{p}}{\delta\boldsymbol{\chi}_{T_{WS_b}}} = \overline{\mathbf{C}}_{S_aW}\begin{bmatrix}\mathbf{I}_3 & -\left[\overline{\mathbf{C}}_{WS_bS_b}\mathbf{p}\right]^\times\end{bmatrix}.$$

References

[1] L. Brizi, E. Giacomini, L. D. Giammarino, S. Ferrari, O. Salem, L. D. Rebotti, and G. Grisetti, "VBR: A Vision Benchmark in Rome," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[2] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial SLAM with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[3] 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, 2021.

[4] M. Abate, Y. Chang, N. Hughes, and L. Carlone, "Kimera2: Robust and Accurate Metric-Semantic SLAM in the Real World," in *International Symposium on Experimental Robotics*. Springer, 2023, pp. 81–95.

[5] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[6] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011.

[7] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[8] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, 2017.

[9] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu *et al.*, "FAST-LIVO2: Fast, direct lidar-inertial-visual odometry," *IEEE Transactions on Robotics*, 2024.

[10] 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.

[11] 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, 2022.

[12] S. Boche, X. Zuo, S. Schaefer, and S. Leutenegger, "Visual-inertial SLAM with tightly-coupled dropout-tolerant GPS fusion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[13] S. Boche, S. B. Laina, and S. Leutenegger, "Tightly-Coupled LiDAR-Visual-Inertial SLAM and Large-Scale Volumetric Occupancy Mapping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[14] J. Jung, S. Boche, S. B. Laina, and S. Leutenegger, "Uncertainty-aware visual-inertial slam with volumetric occupancy mapping," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[15] 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.

[16] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, 2015.

[17] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, 2017.

[18] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.

[19] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on robotics*, vol. 34, no. 4, 2018.

[20] Y. Wang, Y. Ng, I. Sa, A. Parra, C. Rodriguez-Opazo, T. Lin, and H. Li, "MAVIS: Multi-Camera Augmented Visual-Inertial SLAM using SE 2 (3) Based Exact IMU Pre-integration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[21] S. Rahman, A. Q. Li, and I. Rekleitis, "SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[22] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: A multi-sensor fusion-based underwater SLAM system," *The International Journal of Robotics Research*, vol. 41, no. 11-12, 2022.

[23] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[24] X. Peng, Z. Liu, W. Li, P. Tan, S. Y. Cho, and Q. Wang, "DVI-SLAM: A dual visual inertial slam network," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[25] Y. Zhou, X. Li, S. Li, X. Wang, S. Feng, and Y. Tan, "DBA-Fusion: Tightly Integrating Deep Dense Visual Bundle Adjustment with Multiple Sensors for Large-Scale Localization and Mapping," *IEEE Robotics and Automation Letters*, 2024.

[26] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9, 2014.

[27] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized Lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[28] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020.

[29] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In defense of point-to-point ICP–simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, 2023.

[30] 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.

[31] Z. Liu and F. Zhang, "BALM: Bundle adjustment for Lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021.

[32] L. Wiesmann, E. Marks, S. Gupta, T. Guadagnino, J. Behley, and C. Stachniss, "Efficient LiDAR bundle adjustment for multi-scan alignment utilizing continuous-time trajectories," *arXiv preprint arXiv:2412.11760*, 2024.

[33] 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.

[34] 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, 2021.

[35] 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, 2022.

[36] 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.

[37] 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.

[38] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021.

[39] J. Lin and F. Zhang, "R3LIVE: 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.

[40] 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," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[41] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013.

[42] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.

[43] W. Lee, K. Eckenhoff, P. Geneva, and G. Huang, "Intermittent GPS-aided VIO: Online initialization and calibration," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[44] J. Gao, J. Sha, H. Li, and Y. Wang, "A Robust and Fast GNSS-Inertial-LiDAR Odometry with INS-Centric Multiple Modalities by IESKF," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[45] W. Lee, P. Geneva, C. Chen, and G. Huang, "MINS: Efficient and Robust Multisensor-Aided Inertial Navigation System," *Journal of Field Robotics*, 2023.

[46] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010.

[47] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "GOMSF: Graph-optimization based multi-sensor fusion for robust uav pose estimation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[48] G. Cioffi and D. Scaramuzza, "Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[49] J. Liu, W. Gao, and Z. Hu, "Optimization-based visual-inertial SLAM tightly coupled with raw GNSS measurements," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[50] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly coupled GNSS–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[51] H. Zhang, C.-C. Chen, H. Vallery, and T. D. Barfoot, "GNSS/Multi-Sensor Fusion Using Continuous-Time Factor Graph Optimization for Robust Localization," *IEEE Transactions on Robotics*, 2024.

[52] J. Cremona, J. Civera, E. Kofman, and T. Pire, "GNSS-stereo-inertial SLAM for arable farming," *Journal of field robotics*, vol. 41, no. 7, 2024.

[53] X. Li, L. Zhang, X. Zhang, and Z. Jiao, "LC-GVINS: A Robust, Accurate, Monocular Visual-Inertial-GNSS SLAM with Loop Closure," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2023.

[54] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, 2016.

[55] T. Laidlow, J. Czarnowski, and S. Leutenegger, "DeepFusion: Real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[56] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12-14, 2021.

[57] L. Koestler, N. Yang, N. Zeller, and D. Cremers, "Tandem: Tracking and dense mapping in real-time using deep multi-view stereo," in *Conference on Robot Learning*. PMLR, 2022.

[58] Y. Xin, X. Zuo, D. Lu, and S. Leutenegger, "SimpleMapping: Real-time visual-inertial dense mapping with deep multi-view stereo," in *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2023.

[59] M. Sayed, J. Gibson, J. Watson, V. Prisacariu, M. Firman, and C. Godard, "SimpleRecon: 3D reconstruction without 3D convolutions," in *European Conference on Computer Vision*. Springer, 2022.

[60] A. Rosinol, J. J. Leonard, and L. Carlone, "Probabilistic volumetric fusion for dense monocular SLAM," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[61] Z. Teed and J. Deng, "DROID-SLAM: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, 2021.

[62] R. Murai, E. Dexheimer, and A. J. Davison, "MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[63] 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*, 2022.

[64] E. Sandström, K. Ta, L. Van Gool, and M. R. Oswald, "Uncle-SLAM: Uncertainty learning for dense neural SLAM," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[65] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-SLAM: Dense neural point cloud-based SLAM," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[66] L. Liso, E. Sandström, V. Yugay, L. Van Gool, and M. R. Oswald, "Loopy-SLAM: Dense neural SLAM with loop closures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[67] 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*, vol. 2. IEEE, 2003.

[68] 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.

[69] 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.

[70] 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.

[71] 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, 2019.

[72] 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.

[73] 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, 2021.

[74] 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.

[75] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2019.

[76] S. Barbas Laina, S. Boche, S. Papatheodorou, D. Tzoumanikas, S. Schaefer, H. Chen, and S. Leutenegger, "Scalable autonomous drone flight in the forest with visual-inertial SLAM and dense submaps built without LiDAR," *arXiv preprint arXiv:2403.09596*, 2024.

[77] S. Papatheodorou, S. Boche, S. B. Laina, and S. Leutenegger, "Efficient submap-based autonomous MAV exploration using visual-inertial slam configurable for LiDARS or depth cameras," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[78] J. Knights, S. B. Laina, P. Moghadam, and S. Leutenegger, "Solvr: Submap oriented lidar-visual re-localisation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[79] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, 2023.

[80] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[81] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[82] N. A. Carlson, "Federated square root filter for decentralized parallel processors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 3, 1990.

[83] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual SLAM," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[84] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*. Ieee, 2011.

[85] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-SCNN: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019.

[86] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, 2012.

[87] 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, 2016.

[88] 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)*. IEEE, 2018.

[89] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[90] 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, 2022.

[91] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, 1998, pp. 347–353.

[92] W. Zhang, S. Wang, X. Dong, R. Guo, and N. Haala, "BAMF-SLAM: Bundle adjusted multi-fisheye visual-inertial slam using recurrent field transforms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[93] "HILTI SLAM Challenge 2022 Leaderboard," 1 2025. [Online]. Available: https://hilti-challenge.com/leader-board-2022.html

[94] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB," in *International Symposium on GPS/GNSS*, vol. 1. International Convention Center Jeju Korea Seogwipo-si, Republic of Korea, 2009.

Simon Boche is currently pursuing a Ph.D. degree at the Mobile Robotics Lab (MRL) at the Technical University of Munich (TUM), Germany, under the supervision of Prof. Dr. Stefan Leutenegger. He received the M.Sc. degree in Robotics, Cognition, Intelligence and the B.Sc. degree in Engineering Science, both from TUM. His research interests include multi-sensor fusion, localization and mapping, and autonomous navigation for mobile robots.

Jaehyung Jung is currently a Postdoctoral Researcher at the Mobile Robotics Lab (MRL) at the Technical University of Munich (TUM). He received the Ph.D. and M.Sc. degrees in Aerospace Engineering from Seoul National University, and the B.Sc. degree in Aerospace Engineering from Pusan National University. His research interests include state estimation and robot perception.

Sebastián Barbas Laina is currently pursuing a Ph.D. degree at the Mobile Robotics Lab (MRL) at the Technical University of Munich (TUM), under the supervision of Prof. Dr. Stefan Leutenegger. He received the M.Sc. degree in Systems, Control and Robotics from KTH Royal Institute of Technology, and the B.Sc. degree in Industrial Engineering with a focus on Electrical Engineering from Universidad Politécnica de Madrid (UPM). His research interests include localization and mapping, multi-sensor fusion, and autonomous navigation for mobile robots.

Stefan Leutenegger has been an Associate Professor of Mobile Robotics in the Department of Mechanical and Process Engineering at ETH Zurich since early 2025. Prior to that, in 2021, he joined the Technical University of Munich (TUM) as a Tenure-Track Assistant Professor. Between 2014 and 2020, he was a (Senior) Lecturer at Imperial College London, where he continues to hold an honorary Reader position.

SUPPLEMENTARY MATERIAL

In the supplementary material, we provide additional details that were omitted from the main paper due to space limitations. These primarily concern the ablation study on online extrinsic calibration (Sec. VI-F) and the demonstration of GNSS fusion using real-world GNSS data (Sec. VI-E).

Online extrinsic calibration - Details

Robustness to the initial extrinsic error is crucial in practice. To show the robustness of OKVIS2-X to the initial perturbation of the extrinsic prior, we conducted an additional experiment which was only briefly summarized at the end of Sec. VI-F and is detailed further here.

The contribution in multi-camera to IMU extrinsic calibrations is shown on sequence `exp06` in the Hilti-Oxford dataset which contains 5 cameras. Each axis of 5 cameras $(i = 1, 2, \cdots, 5)$ was perturbed with a uniform distribution,

$$\overline{\boldsymbol{T}}_{SC_i} = \boldsymbol{T}_{SC_i} \begin{bmatrix} \mathrm{Exp}(\delta\alpha_i) & \delta r_i \\ 0_{1\times 3} & 1 \end{bmatrix},$$
$$\text{where} \quad \delta\alpha_i \sim \mathcal{U}(-0.3^\circ, 0.3^\circ),$$
$$\delta r_i \sim \mathcal{U}(-1\,\mathrm{cm}, 1\,\mathrm{cm}).$$

We use the same notations as defined in the main paper. For example, $\underrightarrow{\mathcal{F}}_S$ and $\underrightarrow{\mathcal{F}}_{C_i}$ are IMU and i^{th} camera coordinate frames, respectively. Since we do not know the true extrinsics, the perturbed extrinsics $\overline{\boldsymbol{T}}_{SC_i}$ is perturbed from our post-calibrated extrinsics $\boldsymbol{T}_{SC_i}$ which was obtained by full bundle adjustment with all available measurements. Figures 14a-14e visualize the convergence behavior of all five camera-to-IMU extrinsics in 10 runs. The sudden correction observed after roughly 190 seconds was due to a loop closure that as such provides additional information leading to a sudden change in extrinsics. Considering 10 runs, online estimated extrinsics

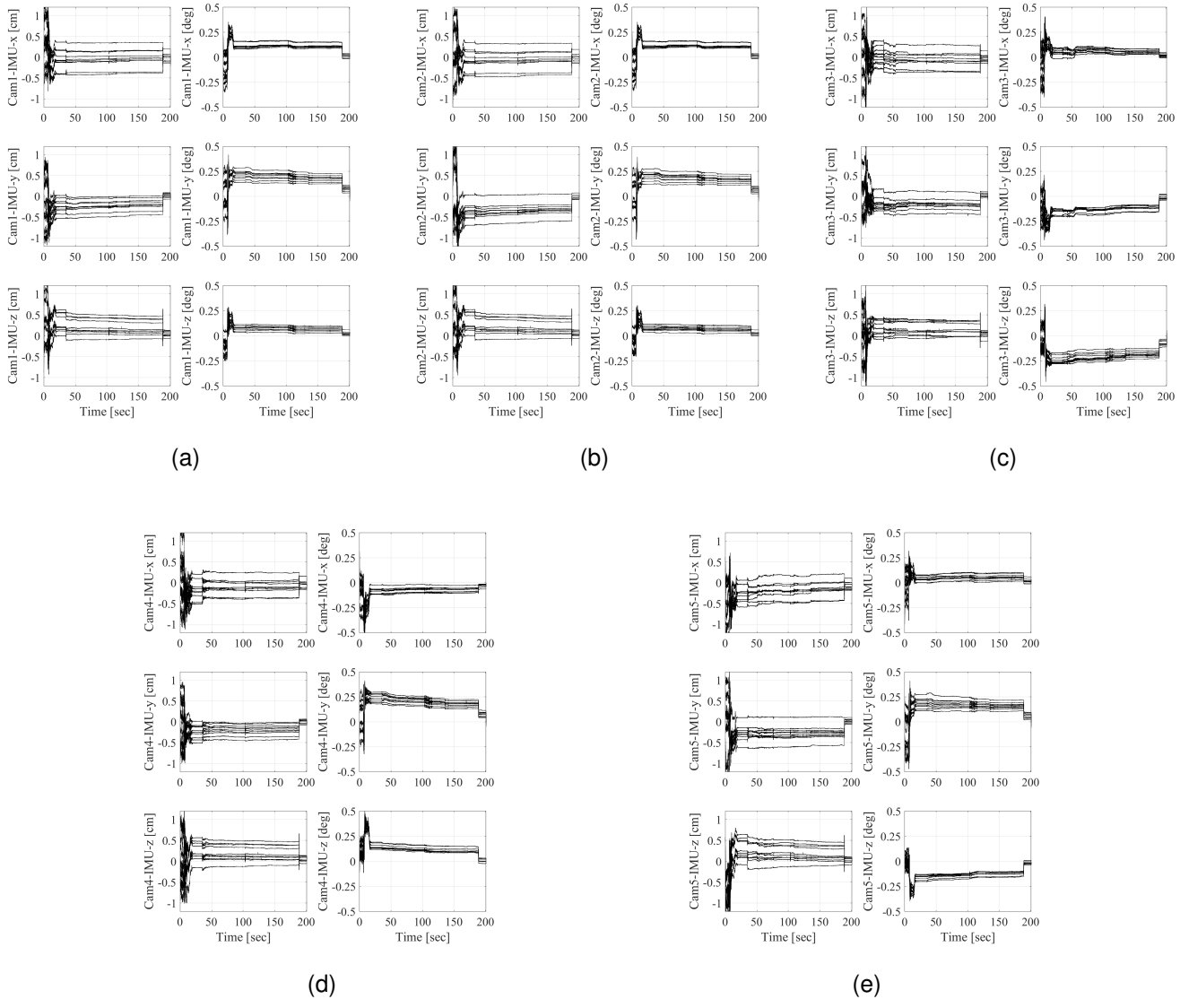

Fig. 14. Camera-IMU extrinsic perturbation study in `exp06`, Hilti-Oxford dataset for 5 cameras. Each line represents the difference between online calibration and the post-calibrated extrinsics in 10 independent runs.

in OKVIS2-X show a standard deviation of $0.08\,\mathrm{cm}$ and 0.02° at the end of the sequence. This study demonstrates the robustness of online calibration to the initial error.

GNSS Fusion - Details on Experiment

As a first note, we want to elaborate on the definition of *tightly-coupled* fusion of GNSS measurements used in this paper. We found that in the literature the definition of *tightly-coupled* fusion is not 100% clear. In our definition, we follow works like [48], in which the term *tightly-coupled* refers to a tight coupling of heterogeneous sensor modalities, such as IMU and vision, on the level of the non-linear optimization. This means, that residuals from different sensor sources are all minimized in one unified optimization problem as opposed to e.g., [18], where global fusion is performed as a separate, loosely-coupled, parallel posegraph optimization.

Furthermore, we want to give some more insights into the experiment in Sec. VI-E, where the capability of OKVIS2-X to handle real-world, and potentially erroneous, GNSS data has been demonstrated. The experiment was conducted on the dataset published by the authors of GVINS [50] alongside their original paper (https://github.com/HKUST-Aerial-Robotics/GVINS-Dataset). This dataset consists of a stereo camera, an IMU and a ZED-F9P GNSS sensor. The authors provide RTK solutions as well as raw GNSS measurements (e.g. pseudoranges, Doppler measurements, etc.). As stated in our paper, we chose to evaluate on the nearly 3 km `complex_environment` sequence due to its numerous challenges for different sensor modalities: featureless images in very bright or dark areas as well as high corruption or complete unavailability of GNSS signals in cluttered environments or indoors (see Fig. 11). Using RTKLIB [94], we computed the Single Point Positioning (SPP) solution as a measurement that would be available from a low-grade consumer-level GNSS device, and used it as an input for the proposed GNSS fusion approach. It is worth mentioning that we could not achieve the same level of SPP accuracy as originally stated in the paper. The RMSE ATE of the SPP solution with respect to the RTK solution in areas with a valid RTK fix is 19.99m as opposed to 6.036m as stated in [50]. Even after filtering out outliers with an error of more than 40 meters, the RMSE ATE of the SPP solution would still be 9.01m. Table X summarizes the resulting accuracy of OKVIS2-X (using the unfiltered SPP solutions), GVINS and SPP. The RMSE ATE has been computed after a full 6DoF alignment of the trajectories. Note that for a fairer comparison with GVINS, OKVIS2-X has also been evaluated without visual loop closures. This evaluation shows that, even though the system was originally designed to fuse intermittent high-accuracy (RTK-) GNSS measurements, it can still also benefit from lower-grade GNSS measurements.

However, we want to emphasize that the proposed system does not claim superiority for the fusion of GNSS in general. We recognize the impressive work of GVINS and think that adopting their formulations for error terms based on the raw measurements, such as pseudoranges and Doppler measurements could improve our system significantly.

TABLE X
RMSE ATE [M] OF THE DIFFERENT APPROACHES IN THE COMPLEX_ENVIRONMENT SEQUENCE.

	Causal	Non-Causal	Final BA
RTKLIB (SPP)	19.99[1]	-	-
GVINS	4.45	-	-
Ours-vi	23.72	23.72	23.74
Ours-vig	12.51	8.47	5.12

[1] RMSE ATE is evaluated for the part of the sequence that is not indoors.