Drift-free Visual SLAM using Digital Twins

Roxane Merat*, Giovanni Cioffi*, Leonard Bauersfeld, and Davide Scaramuzza

***Abstract*—Globally-consistent localization in urban environments is crucial for autonomous systems such as self-driving vehicles and drones, as well as assistive technologies for visually impaired people. Traditional Visual-Inertial Odometry (VIO) and Visual Simultaneous Localization and Mapping (VSLAM) methods, though adequate for local pose estimation, suffer from drift in the long term due to reliance on local sensor data. While GPS counteracts this drift, it is unavailable indoors and often unreliable in urban areas. An alternative is to localize the camera to an existing 3D map using visual-feature matching. This can provide centimeter-level accurate localization but is limited by the visual similarities between the current view and the map. This paper introduces a novel approach that achieves accurate and globally-consistent localization by aligning the sparse 3D point cloud generated by the VIO/VSLAM system to a digital twin using point-to-plane matching; no visual data association is needed. The proposed method provides a 6-DoF global measurement tightly integrated into the VIO/VSLAM system. Experiments run on a high-fidelity GPS simulator and real-world data collected from a drone demonstrate that our approach outperforms state-of-the-art VIO-GPS systems and offers superior robustness against viewpoint changes compared to the state-of-the-art Visual SLAM systems.**

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

Supplementary Material

Video: https://youtu.be/gmHnhWYfuW0

Code:https://github.com/uzh-rpg/rpg_svo_pro_with_digital_twins

I. Introduction

Accurate and globally-consistent pose estimation in complex environments is a critical requirement for autonomous systems [1], such as self-driving vehicles and drones [2], as well as assistive technologies for visually-impaired people [3]. Camera and inertial measurement units (IMU) are commonly used to solve the pose estimation [4], [5] problem thanks to their low cost, lightweight design, and complementary measurement capabilities—cameras providing rich but low-rate and environment-dependent data, and IMUs offering high-rate, low-dimensional, and environment-independent data. Visual-Inertial Odometry (VIO) and Visual Simultaneous Localization and Mapping (VSLAM) algorithms fuse camera and IMU measurements and output the pose estimate in a local frame. The local frame is usually set equal to the identity pose at the initialization of the algorithm. Due to the fact that camera and IMU provide local measurements, as well as noise and modeling simplifications, these algorithms accumulate drift over time.

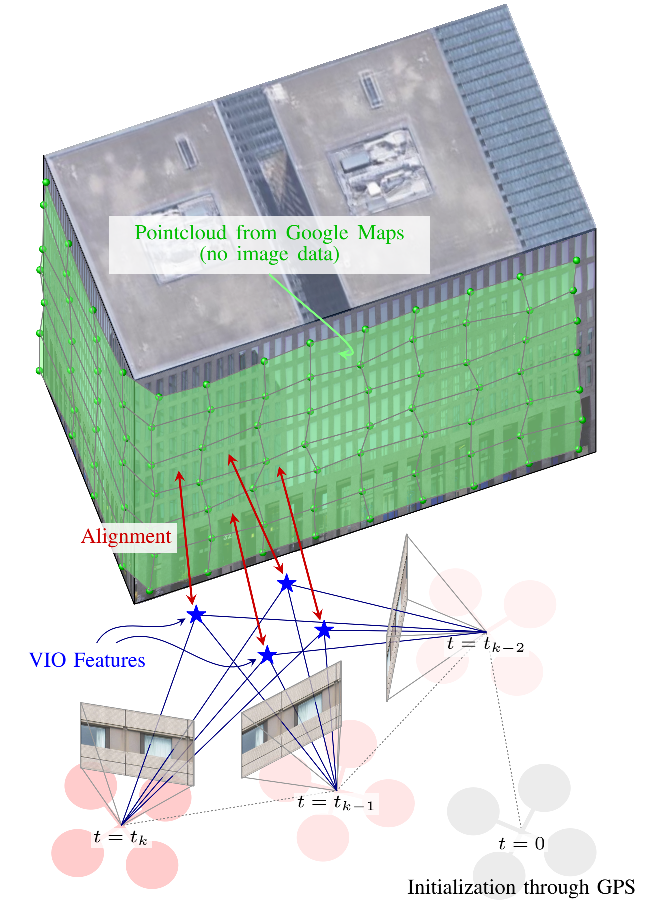

Fig. 1: We propose an approach to achieve drift-free visual SLAM by aligning the local visual SLAM point cloud to a digital twin using point-to-plane matching. The resulting relative transformation between the point cloud and the digital twin provides a global measurement that is then tightly integrated into the SLAM system to obtain global consistency and reduce drift.

To obtain pose estimates in a global fixed frame (e.g., East-North-Up (ENU) reference frame), and to reduce the drift, global measurements are included in the pose estimation process [6]. The most commonly used global measurement is GPS (Global Positioning System). However, GPS measurements are unavailable indoors and often noisy and unreliable in urban canyons, where signal reflections and blockages are common. As a result, GPS measurements can sometimes be more uncertain than the pose estimates generated by the VIO/VSLAM system itself.

VSLAM systems compensate for the drift by localizing to

Manuscript received: August, 30, 2024; Revised November, 1, 2024; Accepted November, 25, 2024.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the European Union's Horizon-Europe Research and Innovation Programme under grant agreement No. 101120732 (AUTOASSESS) and by the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

*These authors contributed equally. The authors are with the Robotics and Perception Group, Department of Informatics, University of Zurich, and Department of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland (http://rpg.ifi.uzh.ch).

Digital Object Identifier (DOI): see top of this page.

a pre-built map using visual measurements. The localization problem is solved by matching visual features between the current camera view and the map. Standard techniques match feature descriptors [7], [8], [9]. More recent works, instead, replace feature descriptors with neural networks [10], [11], [12], [13]. Regardless of the matching technique, all the visual-based localization methods rely on the similarity between the current camera view and the one in the reference map, limiting their effectiveness in many situations, such as localizing aerial images (e.g., taken by a drone) to a map recorded from a ground vehicle during day and night, or localizing a VIO/VSLAM system based on other spectra cameras (e.g., infrared cameras) to a map recorded with a standard camera.

Despite recent advancements in VIO-GPS algorithms and visual localization techniques, existing methods still struggle to fully address the challenge of achieving globally consistent, low-drift pose estimation in urban environments due to the inherent noise, sparsity, or viewpoint dependency of the global measurements on which they rely on.

In response to these challenges, we propose to use geometric information (e.g., 3D point clouds, meshes) to localize to digital twins. Notably, we introduce a novel approach that localizes the local map (sparse 3D point cloud) generated by the VIO/VSLAM systems to the geometric digital twin. Our method uses the local VIO/VSLAM point cloud to perform point-to-plane matching against the mesh representation of the digital twin, providing a new global measurement that is tightly fused into the VIO/VSLAM system.

Our experiments demonstrate the superiority of the proposed method over the current state-of-the-art VIO-GPS approaches, both in a high-fidelity GPS simulator, which we have developed in the context of this project, and real-world tests conducted with a drone flying in a city. The results show that our approach not only reduces drift more effectively but also offers greater robustness to variations in viewpoint compared to traditional visual localization techniques based on feature descriptors matching. The main contributions of this work are:

- A novel algorithm that localizes the local and sparse VSLAM 3D point cloud to a digital twin via point-to-plane matching, consequently being independent of visual matching.
- Tightly fusing such localization measurement in a VSLAM system to achieve globally consistent, low-drift pose estimates in urban environments.
- We show the superiority of our approach against state-of-the-art VIO-GPS systems in simulation experiments using a novel high-fidelity GPS simulator and real-world data collected onboard a drone flying in a city.
- We show that the proposed local-global map localization is more robust than the state-of-the-art vision localization technique based on image feature descriptors matching.

II. Related Work

Due to drift, long-term pose estimation in large-scale environments requires global measurements, i.e. measurements relative to a static frame.

GPS measurements are widely used in VIO/VSLAM systems to achieve low drift pose estimates in the long term. Early efforts in this direction include the work of Qin et al. [14] and Mascaro et al. [15], who loosely fuse GPS data with a visual-inertial odometry system. In subsequent work, Cioffi et al. [6] show that the accuracy of the pose estimates can be improved by tightly coupling visual, inertial, and GPS measurements. Boche et al. [16] extend the tightly-coupled approach to make it robust to short-term GPS data dropouts. All these methods rely on the availability of GPS data and consequently work well in open-sky scenarios but suffer in urban environments. Different methods have been proposed to deal with urban environments. Cao et al. [17] propose a raw GPS pseudorange measurement model, which improves the system's resilience against corrupted GPS signals. Yin et al. [18] add filtering of these measurements based on an upward-facing camera, keeping only the pseudoranges from line-of-sight satellites. Lee et al. [19] propose a method that is resilient to the loss of GPS signals by gradually adding GPS measurements when they become available again. Although these methods help to limit the error introduced by uncertain GPS measurements in dense areas, they suffer when GPS measurements are corrupted for long periods.

Alternative methods obtain global measurements by localizing to a map of the environment. In VSLAM, the localization process is based on visual data association between the current camera view and the map. Classical loop closure techniques rely on matching feature descriptors, e.g., the ORB descriptor [7]. Majdik et al. [20] have developed a method to match images captured by unmanned aerial vehicles (UAVs) with Google Street View images by back-projecting their textures onto CAD models of cities to infer global positions. In a follow-up work [21], they improved their approach against viewpoint differences. While this method can compensate for moderate viewpoint differences, UAVs can navigate at altitudes where the camera field of view does not overlap with the Google Street View images. Bao et al. [22] propose a similar approach as [20], [21] where they first back-project the VIO point cloud onto a previous VSLAM-generated map, then use visual feature matching to verify these projections and derive a localization constraint. Recently, many methods relying on deep learning have been proposed [10], [11], [12]. These methods usually train a neural network to predict the pose of the camera given an image. The training data usually consists of images contained in the map to which one is interested to localize to. We refer the reader to the survey paper in [23] where a broad overview of deep learning techniques for visual localization is provided. Satellite images have also been used to localize the current camera view in a fixed global frame by Sadli et al. [24] where they propose a technique to match vehicle images with satellite images by aligning road features. Visual localization techniques provide cm-accurate localization when visual data association can be accurately performed. However, they are very susceptible to viewpoint differences and environmental conditions (e.g., lighting). For example, methods that rely on Google Street View images, which are taken from ground viewpoints, fail to localize images taken onboard UAVs flying high above the ground.

In this work, we are inspired by point cloud registration techniques, which are at the core of LiDAR-based SLAM sys-

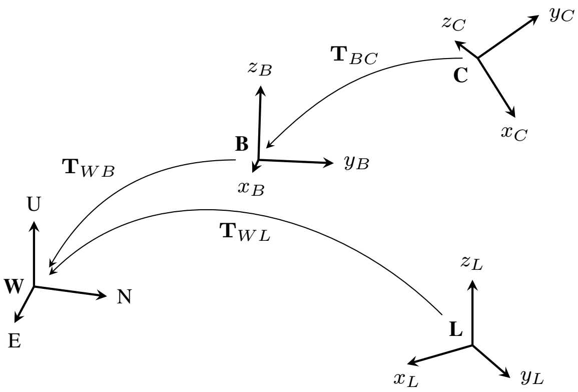

Fig. 2: Reference frames used in this work.

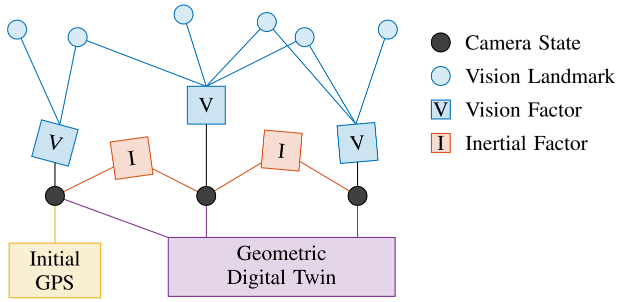

Fig. 3: Factor graph representation of the proposed visual SLAM system with visual, inertial, and city digital twin registration factors. The system relies on GPS data to initialize.

tems. Point cloud registration aims to find the transformation that aligns two point clouds. Commonly, this problem is solved by applying the iterative closest point (ICP) algorithm [25]. Indeed, we propose to localize the local point cloud generated by a VIO/VSLAM system to a mesh representation of a city digital twin.

III. Methodology

In this work, we propose a VSLAM system that achieves low-drift and global consistent pose estimates in urban environments by leveraging the scene geometry to localize to a digital twin. The localization problem is formulated as a point cloud registration problem that can be efficiently solved by the point-to-plane ICP algorithm. Specifically, we obtain a global measurement by registering the local sparse 3D point cloud generated by the VSLAM system to the city digital twin. This global measurement is then tightly coupled with the current camera and IMU measurements in the VSLAM pose estimation process. Additionally, we propose an algorithm that adaptively finds the weights that are used to fuse the global measurement in the VSLAM. This adaptive weighting solution allows us to account for uncertain solutions to the registration problem, which derive from degenerate cases and depend on the scene geometry. We describe in this section our methodology in detail. A diagram of our methodology is in Fig. 3

A. Notation

The coordinates frames used in this work are illustrated in Fig. 2. We transform the GPS measurements, typically represented in the geodetic coordinate system (latitude, longitude, altitude), in Cartesian coordinates (x, y, z) expressed in the ENU frame, denoted by W. L represents the VSLAM local frame. This frame is initialized with the identity at the first VSLAM pose estimate. B is the body frame and corresponds to the IMU frame. The camera frame is denoted by C. We assume that the IMU-camera calibration is known. To simplify the mathematical derivations in this section, we assume that the GPS and camera-IMU sensor platform are time synchronized and that the GPS antenna is located in the same place as the IMU. These quantities can be obtained in practice via offline calibration [26], [27]. We use $\boldsymbol{p}_{WB}^k$ and $\boldsymbol{R}_{WB}^k$ to represent the position and orientation (in the form of a rotation matrix as an element of $SO(3)$) of the frame B with respect to W at time t_k. The corresponding homogeneous transformation matrix is written as $\boldsymbol{T}_{WB}^k = [\boldsymbol{R}_{WB}^k | \boldsymbol{p}_{WB}^k]$. The same notation applies to the other reference frames. We use the symbol $\hat{\cdot}$ to represent the estimated quantities, for example, $\hat{\boldsymbol{p}}_{WB}^k$ is the position estimated by the VSLAM at time t_k.

B. Problem Formulation

Our pose estimation problem follows the formulation of the sliding-window optimization-based VSLAM algorithm [5]. The camera poses, as well as the local sparse 3D point cloud, and the IMU biases, are obtained by minimizing the following cost function:

$$\begin{aligned}\min_{\mathcal{X}} \mathcal{J} = \min_{\mathcal{X}} \sum_{k=0}^{K-1} \sum_{j\in\mathcal{J}_k} \left\| \boldsymbol{e}_v^{j,k} \right\|^2_{W_v^{j,k}} + \sum_{k=0}^{K-1} \left\| \boldsymbol{e}_i^k \right\|^2_{W_i^k} \\ + \sum_{k=0}^{K-1} \left\| \boldsymbol{e}_m^k \right\|^2_{W_m^k} + \left\| \boldsymbol{e}_p \right\|^2\end{aligned} \tag{1}$$

where $\mathcal{X}$ is the vector that contains all the optimization variables.

The term $\boldsymbol{e}_v^{j,k} = \boldsymbol{z}^{j,k} - h(\boldsymbol{l}_j^l)$ represents the visual factor. Given $\boldsymbol{l}_j^l \in \mathcal{J}_k$ a 3D point visible from the keyframe k, the visual factor is defined by reprojecting $\boldsymbol{l}_j^l$ to the image frame via the function $h(\cdot)$ and computing the error against the corresponding image point $\boldsymbol{z}^{j,k}$. The term $\boldsymbol{e}_i$ represents the inertial factor. To compute this factor, we use the IMU preintegration algorithm proposed by Forster et al. [28]. The term $\boldsymbol{e}_m$ represents the map registration factor. Its derivation is in Sec. III-C. The term $\boldsymbol{e}_p$ represents the marginalization factor. This marginalization factor encodes information related to old quantities that fall outside the current sliding window. To compute this factor, we use the strategy proposed by Leutenegger et al. [28]. This strategy distinguishes between variables to marginalize (included in the derivation of the term $\boldsymbol{e}_p$) and variables to drop. The variables to drop are the ones that are not connected to keyframes, for example, 3D points that are visible from the frame that falls outside the sliding window when this frame is not a keyframe. Dropping, instead of marginalizing, such variables allows to keep the sparsity of the Jacobian of Eq. 1.

C. Localization to 3D Digital Twins

1) Residual: At the initialization of our system, a GPS measurement prior is used to access the part of the city digital twin around the current pose. This also allows to represent the city map in the reference frame W. During deployment, we keep track of the camera pose in both the local frame L and global frame W. This allows to run the map registration algorithm between the sparse local VSLAM point cloud and the global semi-dense city digital twin. We use the ICP point-to-plane algorithm to solve the registration problem. Our ICP algorithm minimizes the function:

$$\min_{\delta\boldsymbol{R},\delta\boldsymbol{p}} \mathcal{J} = \sum_{j\in\mathcal{J}_k} \left([\delta\boldsymbol{R}\cdot\boldsymbol{a}_j + \delta\boldsymbol{p} - \boldsymbol{b}_j]\cdot\boldsymbol{n}_j\right)^2 \tag{2}$$

where $\delta\boldsymbol{R}, \delta\boldsymbol{p}$ are the rotation and position that align the source point cloud (the local VSLAM point cloud) to the target point cloud (the city digital twin point cloud). The point $\boldsymbol{a}_j$ is the point in the source point cloud that is associated to the point $\boldsymbol{b}_j$ in the target point cloud. The quantity $\boldsymbol{n}_j$ is the normal vector at the point $\boldsymbol{b}_j$.

We define the solution of Eq. 2 in the form of a homogeneous transformation matrix a $\delta\boldsymbol{T} = [\delta\boldsymbol{R}|\delta\boldsymbol{p}]$. Taken a point in the target point cloud $\boldsymbol{m}_j^w$ and its position derivated from the current pose estimate $\hat{\boldsymbol{m}}_j^w$ (namely, $\boldsymbol{m}_j^c = \hat{\boldsymbol{m}}_j^c$), we derive the global measurement as:

$$\delta\boldsymbol{T}\cdot\hat{\boldsymbol{m}}_j^w = \boldsymbol{m}_j^w \tag{3}$$

$$\delta\boldsymbol{T}\cdot\hat{\boldsymbol{T}}_{WL}\cdot\hat{\boldsymbol{T}}_{LB}\cdot\boldsymbol{T}_{BC}\cdot\boldsymbol{m}_j^c = \hat{\boldsymbol{T}}_{WL}\cdot\boldsymbol{T}_{LB}\cdot\boldsymbol{m}_j^w \tag{4}$$

$$\delta\boldsymbol{T}\cdot\hat{\boldsymbol{T}}_{WL}\cdot\hat{\boldsymbol{T}}_{LB} = \hat{\boldsymbol{T}}_{WL}\cdot\boldsymbol{T}_{LB} \tag{5}$$

$$\hat{\boldsymbol{T}}_{LW}\cdot\delta\boldsymbol{T}\cdot\hat{\boldsymbol{T}}_{WL}\cdot\hat{\boldsymbol{T}}_{LB} = \boldsymbol{T}_{LB}, \tag{6}$$

where $\boldsymbol{T}_{LB} = [\boldsymbol{R}_{LB}|\boldsymbol{p}_{LB}]$.

The map registration residual term that is added to the optimization problem in Eq. 1 is defined as:

$$\boldsymbol{e}_m^k = \begin{bmatrix} \hat{\boldsymbol{p}}_{LB}^k - \boldsymbol{p}_{LB}^k \\ \mathrm{Log}((\hat{\boldsymbol{R}}_{LB}^k)^t\cdot\hat{\boldsymbol{R}}_{LB}^k), \end{bmatrix} \tag{7}$$

where $\mathrm{Log}(\cdot)$ is the logarithm map that maps an element of the Lie group, *SO*(3), to an element of the Lie algebra.

2) Adaptive Weighting: The solution of Eq. 2 is not guaranteed to be unique. The presence of many collinear or coplanar points can lead to unobservable directions. This situation can be encountered in urban environments, for example when the camera moves in a straight line facing a building facade. To tackle this problem, we propose an adaptive weighting strategy that assigns a low weight to the detected unobservable direction while still allowing high weights for well-observed directions. Our solution, inspired by [29], assigns the weight to the map registration residual, W_m^k, in Eq. 1 based on the covariance matrix of the cost function in 2. The cost function in 2 can be linearized as:

$$\min_{\boldsymbol{x}} \sum_{j\in\mathcal{J}_k} \left|\left|\boldsymbol{y_j} - \boldsymbol{A_j x}\right|\right|^2, \tag{8}$$

where:

$$\boldsymbol{y}_j = \boldsymbol{n}_j^T(\boldsymbol{a}_j - \boldsymbol{b}_j), \quad \boldsymbol{A}_j = \begin{bmatrix} -(\boldsymbol{a}_j\times\boldsymbol{n}_j)^T & -\boldsymbol{n}_j^T \end{bmatrix}. \tag{9}$$

The covariance is then equivalent to the inverse of the Hessian, $\boldsymbol{H} = \sum_j(\boldsymbol{A_j^T A_j})$. The weight of the map registration residual is defined as:

$$\boldsymbol{W}_m^k = \frac{\beta}{\sigma^2}\cdot\exp\left(-\frac{\gamma^2}{2}\right)\cdot\boldsymbol{H}, \tag{10}$$

where: γ is the root-mean-square-error of the inliers, $\sigma^2 = \sum_k(\boldsymbol{H}_{kk})$ is the sum of the diagonal elements of $\boldsymbol{H}$, and β is a scaling factor that makes $\boldsymbol{W}_m^k$ of the same order of magnitude as the weights of the other residuals in Eq. 1. We found β empirically and kept it constant in all our experiments.

D. Initial Frame alignment

The relative transformation (position, and heading, i.e., the rotation around the gravity-aligned z-axis), $\boldsymbol{T}_{WL} = [\boldsymbol{R}_{WL}^z|\boldsymbol{p}_{WL}]$ is estimated as new position estimates from the VSLAM, $\boldsymbol{p}_{LB}^k$, and global measurements, $\boldsymbol{p}_{WB}^k$, are available. Our frame alignment solution is found by optimizing the cost function:

$$\arg\min_{\boldsymbol{R}_{WL},\boldsymbol{t}_{WL}} \sum_n ||\boldsymbol{p}_{WB}^n - \boldsymbol{R}_{WL}^z\cdot\boldsymbol{p}_{LB} - \boldsymbol{p}_{WL}||^2 \tag{11}$$

We apply the Umeyama method [30] to find the solution to Eq. 11. Initially (before map registrations can be successfully found), we solve the problem in Eq. 11 using the VSLAM pose estimates and the GPS measurements. Due to the inaccuracy of GPS measurements in urban environments, this initial alignment can be inaccurate. We found empirically on real data that the heading error resulting from the frame alignment based on GPS measurements cannot be neglected. Once the map registration can be successfully performed, we use the position obtained from the map registration, c.f. Eq. 2, instead of the GPS data. The frame alignment Eq. 11 is run until a convergence criterion is met. We adopt the convergence criteria proposed by Boche et al. [16], which checks that the heading covariance is below a threshold.

E. Implementation Details

We integrated the proposed geometry-based map registration strategy in SVO [1]. The sliding-window optimization is based on the approach presented in [31]. We limited the number of keyframes in the sliding-window to 5. The visual frontend is based on SVO [32]. We run the ICP registration on a local portion, of constant size, of the global map. Consequently, the runtime and memory requirements of the map registration process do not depend on the dimensions of the global map. In all our experiments, the local portion of the global map has horizontal dimensions of 150 x 150 m around the current camera position. We take all the available points along the vertical dimension. We name our system SVO-Digital Twin. All the experiments ran on a laptop equipped with a 2.60GHz Intel Core i7 CPU. Our system runs in real time, namely, it can complete the processing of one image before the next one is available (the maximum camera frame rate in all the datasets used in the experiments is 30 Hz).

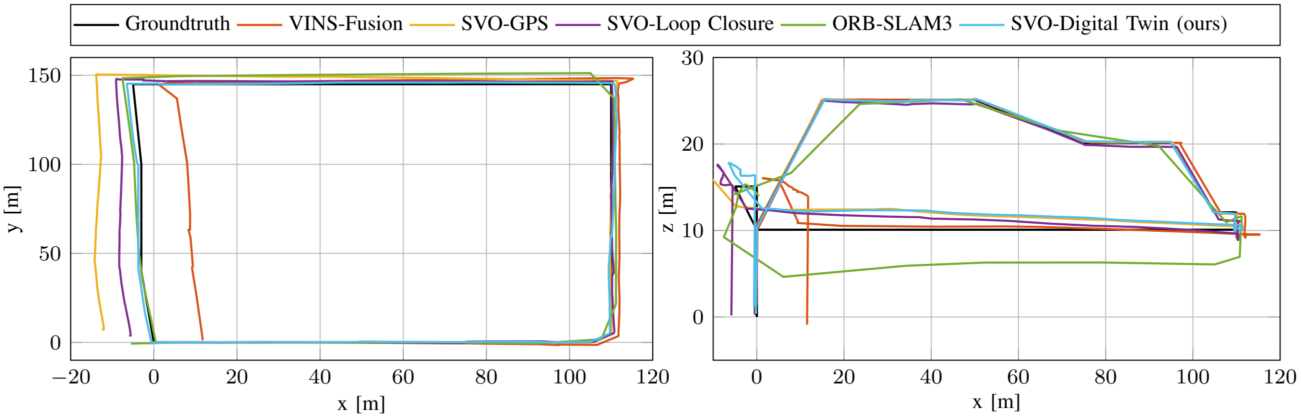

Fig. 4: Simulated data. Trajectory estimated by SVO-Digital Twin and the baselines together with the ground truth trajectory. Left: Top-down view, right: Side view.

IV. Experiments

To evaluate the effectiveness of our method, we conducted experiments in both simulated and real-world environments. We compare our method against four baselines. To compare our method against VIO systems that integrate GPS data, we select VINS-Fusion [14] and SVO-GPS [6] as baselines. VINS-Fusion loosely couples a VIO [33] with GPS measurements. Instead, SVO-GPS tightly couples the GPS measurements in the VIO sliding-window optimization. To compare our method against SLAM systems that close loops based on visual feature matching, we select SVO-Loop Closure [32] and ORB-SLAM3 [8]. SVO-Loop Closure uses the same VIO pipeline as our system. ORB-SLAM3 is a Visual SLAM system that performs loop closure by matching ORB features.

In all our experiments, we use the metrics:

- Positional absolute trajectory error (ATE_P) [m] : $\sqrt{\frac{1}{N}\sum_{k=0}^{N-1}\left\|\boldsymbol{p}_{WB}^k - \hat{\boldsymbol{p}}_{WB}^k\right\|^2}$.
- Rotational absolute trajectory error (ATE_R) [deg] : $\sqrt{\frac{1}{N}\sum_{k=0}^{N-1}\left\|\text{Log}((\boldsymbol{R}_{WB}^k)^t \cdot \hat{\boldsymbol{R}}_{WB}^k)\right\|^2}$.

We use the trajectory evaluation toolbox proposed by Zhang et al. [34] to compute these evaluation metrics. We use the transformation $\boldsymbol{T}_{WL}$, c.f. Eq 11, to align the pose estimates to the world frame and then compute the evaluation metrics. This transformation is computed offline, as proposed in [34], for the systems that do not use global measurements.

A. Simulation results

1) Setup: We use the trajectory simulator in [35] to simulate trajectories and IMU data, and, the Flightmare simulator [36] to render images from a virtual city model.

Differently from the related works [6], [14], [16], where GPS measurements are generated by sampling a zero-mean Gaussian distribution with a pre-selected standard deviation, we train a data-driven measurement generator on real GPS data. We collected training data by randomly sampling poses in our simulated city. We assume that the simulated city is located at the same coordinates as Zurich, Switzerland. We use ray tracing [37] for each data sample to compute the received GPS signals. In this way, we create two datasets, one containing the number of satellites seen as a function of the height, and the other containing the multipath error of a single satellite as a function of the height. This dataset generation strategy is inspired by [37]. We trained a Gaussian Process (GP) regressor to predict the number of visible satellites and a Gaussian Mixture Model (GMM) to predict the multipath error as a function of the height. To generate data for the VIO/SLAM algorithms, we simulate a drone flying a trajectory long 500 meters. During the flight, the GP regressor is used to predict the number of visible satellites, and the GMM is used to generate a multipath error for each satellite. The GPS measurement is then generated by performing trilateration using the measurements from all the visible satellites. GPS measurements are generated at 5 Hz. Our virtual city is shown in Fig. 5. The simulated GPS data is shown in Fig. 6.

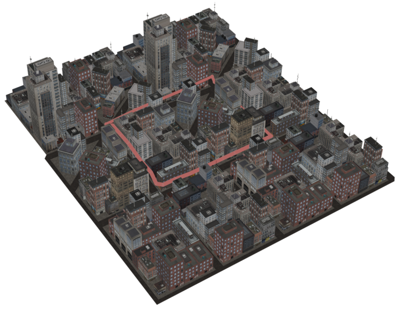

Fig. 5: View of the simulated city. The simulated trajectory is depicted in red.

TABLE I: Evaluation of the trajectory estimates in the simulated dataset. In bold are the best values.

Evaluation Metric	VIO / SLAM Algorithm				
	Vins-Fusion	SVO-GPS	SVO-Loop Closure	ORB-SLAM3	SVO-Digital Twin (Ours)
ATE_P [m]	7.87	7.84	4.02	4.32	**2.78**
ATE_R [deg]	1.30	1.22	1.49	1.49	**0.89**

[1] https://github.com/uzh-rpg/rpg_svo_pro_open

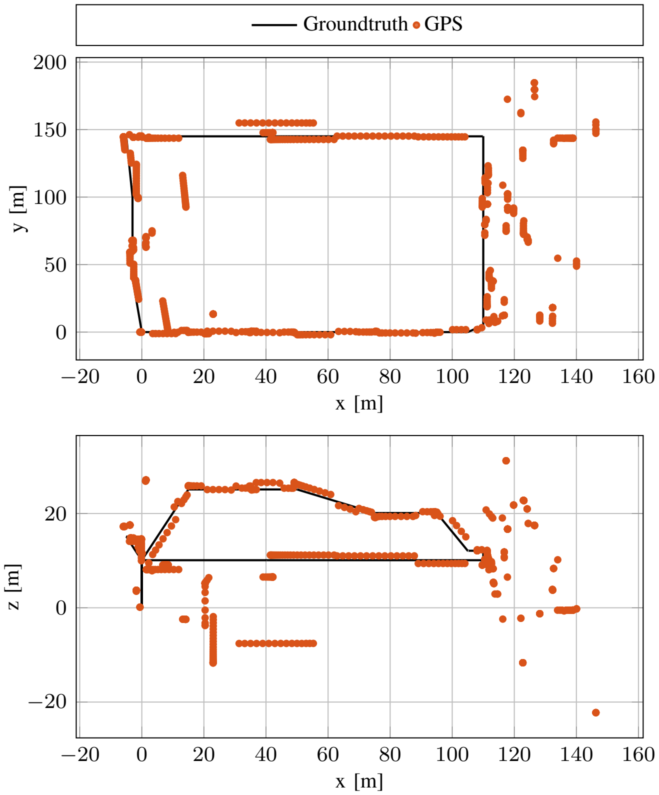

Fig. 6: Visualization of the simulated GPS measurements.

TABLE II: Evaluation of the benefit of the adaptive weighting strategy presented in Sec. III-C2

Evaluation Metric	SVO-Digital Twin	
	w/o Adaptive Weighting	w/ Adaptive Weighting (Ours)
ATE_P [m]	4.54	**2.78**
ATE_R [deg]	2.41	**0.89**

TABLE III: Runtime of the main system components.

System component	Average Runtime [s]
Initial alignment	40.0
ICP-registration	0.44
Pose estimation	0.03

TABLE IV: Evaluation of the trajectory estimates in the real-world dataset. In bold are the best values.

Evaluation Metric	VIO / SLAM Algorithm				
	Vins-Fusion	SVO-GPS	SVO-Loop Closure	ORB-SLAM3	SVO-Digital Twin (Ours)
ATE_P [m]	1.61	3.05	1.60	2.36	**1.09**
ATE_R [deg]	11.30	5.99	4.70	7.04	**3.40**

2) Results: The absolute trajectory errors achieved by our system, SVO-Digital Twin, and the baselines are reported in Table I. The estimated trajectories are plotted, together with the ground truth, in Fig 4. SVO-Digital Twin achieves the best absolute trajectory error in position and rotation by a large margin. Specifically, the ATE_p and ATE_R improve by 31% and 27%, respectively, with respect to the best-performing baselines.

We include in Table II an ablation study to validate our adaptive weighting strategy. The improved ATE errors validate the benefits of the proposed adaptive weighting strategy.

We include the runtime of the main system components in Table III. We fix a time length for the initial alignment step. After this time has passed, we check if the convergence criterion (see, Sec. III-D) is met. If yes, the alignment is deemed successful. If not, we accumulate more GPS measurements until this criterion is met. The runtime of pose estimation is the time elapsed from when an image is captured until the camera pose of such an image is estimated.

B. Real world experiments

1) Setup: We flew a Flyability Elios 3 [2], equipped with a GPS antenna, a monocular fisheye camera, and an IMU in a neighborhood featuring tall buildings in Zurich, Switzerland. The flight path lasts circa 5 minutes and covers over 500 meters. The drone altitude varies between 0 and 17 meters. The drone flies three loops around a building at different altitudes, which increase by approximately 4 meters per loop. The drone maintained a 45-degree angle towards the building during most of the flight, ensuring visibility of both the structure and the surrounding environment. The ground truth was generated using a structure-from-motion framework, COLMAP [38]. We use the city digital twin from Google Photorealistic 3D Tiles [3], c.f. Fig. 8. The camera trajectory obtained from COLMAP is aligned to the world frame by using an ICP algorithm that takes as inputs the point cloud generated by COLMAP and the point cloud of the city map. We show 2 images taken at the same location but in two different laps in Fig. 9. The different camera point of view makes visual loop closure difficult, indeed, neither SVO Loop Closure nor ORB-SLAM3 can close a loop in this case. Contrary, our map registration is performed successfully in this situation.

2) Results: The absolute trajectory errors achieved by our system, SVO-Digital Twin, and the baselines are reported in Table IV. The estimated trajectories are plotted, together with the ground truth, in Fig 7. SVO-Digital Twin achieves the best absolute trajectory error in position and rotation by a large margin. Specifically, the ATE_p and ATE_R improve by 32% and 28%, respectively, with respect to the best-performing baselines. Since SVO-GPS tightly couples the GPS with visual and inertial data, its performance is degraded due to the high uncertain GPS measurements.

Furthermore, we evaluated the frame alignment strategy proposed in Sec III-D against aligning the VSLAM pose estimates with the world-frame referenced ground truth poses.

[2] https://www.flyability.com/elios-3
[3] https://3d-tiles.web.app

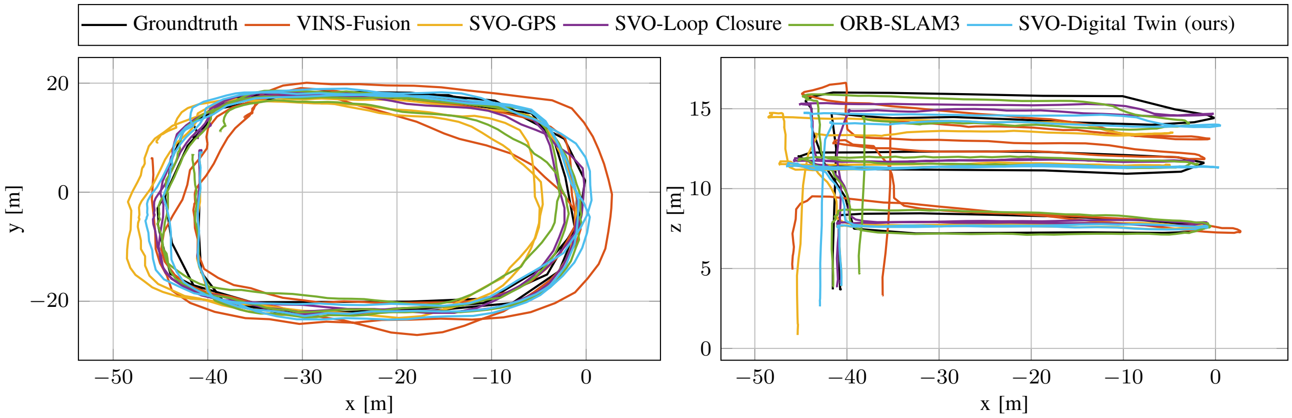

Fig. 7: Real-world data. Trajectory estimated by the SVO-Digital Twin and the baselines together with the ground truth trajectory. Left: Top-down view, right: Side view.

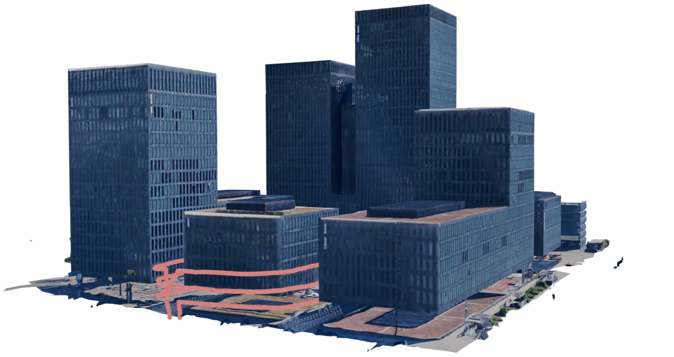

Fig. 8: Rendered map from Google Photorealistic 3D Tiles of the neighborhood in Zurich where we collected the real-world data. The trajectory flown by our drone is depicted in red.

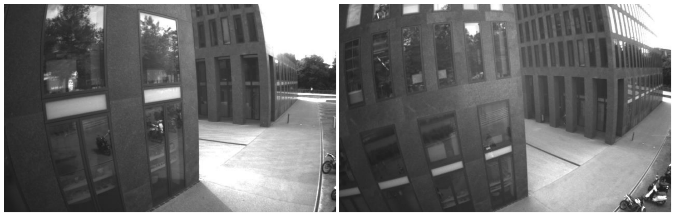

Fig. 9: Images of the same part of the building taken in two consecutive laps. Neither SVO-Loop Closure nor ORB-SLAM3 can close a loop based on visual descriptor matching. Contrary, our map registration is performed successfully in this situation.

Our alignment strategy achieves a translation error of 0.50 [m] and a heading error of 0.79 [deg]. Otherwise, using only GPS measurement in our alignment strategy results in a translation error of 8.11 [m] and a heading error of 6.77 [deg].

V. Discussion

In this section, we discuss the failure cases of the global map registration and the generalizability of our approach to any *Digital Twin*.

The registration between the local VSLAM point cloud and the global map can be incorrect for two main reasons. The first reason is 3D aliasing. Our adaptive weighting strategy can detect this failure case and provide a solution, c.f. Tab. II. The second reason is inaccuracies in either the global map or the local 3D point cloud that lead to incorrect matches in the ICP-based registration. Inaccuracies in the global map are mainly the consequence of changes in the environment. This case is detected by the outlier rejection strategy that is part of our ICP algorithm, and the registration is deemed unsuccessful. Inaccuracies in the local 3D point cloud are due to large drift that could accumulate in the VSLAM system after a long period of unsuccessful global map registrations. In this case, the solution is to re-align the local VSLAM reference frame to the global frame using GPS or any other global measurement. A possible implementation of such a solution can be based on the approach proposed in [16], where the VIO local frame is re-aligned to the global frame after a long time of GPS outage.

Our method is not restricted to the availability of Google 3D tiles maps. Indeed, it can work with any *Digital Twin* that represents the geometry of the environment via 3D points, meshes, etc. In this case, the camera's initial position in the Digital Twin needs to be known and it can be set manually. Google 3D tiles are one possible representation of a city's Digital Twin. Another example of the city's Digital Twin that can be used in our system is the open-source maps provided by OpenStreetMap [4]. OpenStreetMap maps provide 3D information of the buildings but they do not have texture information included in most places. A method relying on image features would not be able to use OpenStreetMap maps.

VI. Conclusion

In conclusion, this paper presents a novel approach to reduce the drift of VIO/VSLAM systems by leveraging geometric information from digital twins. By introducing a method that localizes the VIO/VSLAM-generated point cloud to a city digital twin via point-to-plane matching, we address the limitations of methods that rely on GPS or visual localization techniques. Our experiments, both in a high-fidelity GPS simulator and real-world drone flight, demonstrate the superiority of this method over existing systems, particularly in reducing drift and handling viewpoint variations.

[4] https://www.openstreetmap.org

References

[1] "Drone strategy 2.0: Creating a large-scale european drone market," https://ec.europa.eu/commission/presscorner/detail/en/ip_22_7076, accessed: 2024-08-30.

[2] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz *et al.*, "Self-driving cars: A survey," *Expert systems with applications*, vol. 165, p. 113816, 2021.

[3] L. Gionfrida, D. Kim, D. Scaramuzza, D. Farina, and R. D. Howe, "Wearable robots for the real world need vision," *Science Robotics*, vol. 9, no. 90, 2024.

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

[5] G. Huang, "Visual-inertial navigation: A concise review," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[6] G. Cioffi and D. Scaramuzza, "Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2020.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[8] 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 Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[9] A. Oertel, T. Cieslewski, and D. Scaramuzza, "Augmenting visual place recognition with structural cues," *IEEE Robot. Autom. Lett.*, 2020.

[10] P.-E. Sarlin, D. DeTone, T.-Y. Yang, A. Avetisyan, J. Straub, T. Malisiewicz, S. R. Bulo, R. Newcombe, P. Kontschieder, and V. Balntas, "Orienternet: Visual localization in 2d public maps with neural matching," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023.

[11] V. Panek, Z. Kukelova, and T. Sattler, "Visual localization using imperfect 3d models from the internet," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023.

[12] P.-E. Sarlin, E. Trulls, M. Pollefeys, J. Hosang, and S. Lynen, "Snap: Self-supervised neural maps for visual positioning and semantic understanding," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[13] Y. Ye, T. Cieslewski, A. Loquercio, and D. Scaramuzza, "Place recognition in semi-dense maps: Geometric and learning-based approaches," in *British Mach. Vis. Conf. (BMVC)*, 2017.

[14] 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*, 2018, accessed 2024-08-30.

[15] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018.

[16] S. Boche, X. Zuo, S. Schaefer, and S. Leutenegger, "Visual-inertial slam with tightly-coupled dropout-tolerant gps fusion," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2022.

[17] S. Cao, X. Lu, and S. Shen, "Gvins: Tightly coupled gnss-visual-inertial fusion for smooth and consistent state estimation," *IEEE Trans. Robot.*, 2021.

[18] J. Yin, T. Li, H. Yin, W. Yu, and D. Zou, "Sky-gvins: a sky-segmentation aided gnss-visual-inertial system for robust navigation in urban canyons," *Geo-spatial Information Science*, 2023.

[19] W. Lee, K. Eckenhoff, P. Geneva, and G. P. Huang, "Intermittent gps-aided vio: Online initialization and calibration," *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 5724–5731, 2020.

[20] A. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza, "Micro air vehicle localization and position tracking from textured 3d cadastral models," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014.

[21] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "Mav urban localization from google street view data," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2013.

[22] H. Bao, W. Xie, Q. Qian, D. Chen, S. Zhai, N. Wang, and G. Zhang, "Robust tightly-coupled visual-inertial odometry with pre-built maps in high latency situations," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[23] C. Chen, B. Wang, C. X. Lu, N. Trigoni, and A. Markham, "Deep learning for visual localization and mapping: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[24] R. Sadli, M. Afkir, A. Hadid, A. Rivenq, and A. Taleb-Ahmed, "Map-matching-based localization using camera and low-cost gps for lane-level accuracy," *Procedia Computer Science*, vol. 198, pp. 255–262, 2022.

[25] 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 Robot. Autom. Lett.*, vol. 8, no. 2, pp. 1029–1036, 2023.

[26] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2013.

[27] G. Cioffi, T. Cieslewski, and D. Scaramuzza, "Continuous-time vs. discrete-time vision-based slam: A comparative study," *IEEE Robot. Autom. Lett.*, 2022.

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

[29] M. Barczyk, S. Bonnabel, and F. Goulette, "Observability, covariance and uncertainty of icp scan matching," *arXiv*, 2014, accessed 2024-08-30.

[30] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 376–380, 1991.

[31] 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, 02 2014.

[32] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, 2017.

[33] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[34] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[35] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, p. eabl6259, 2022.

[36] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," *Conf. on Robot. Learning (CoRL)*, 2020.

[37] S. Ollander, F.-W. Bode, and M. Baum, "Simulation-based analysis of multipath delay distributions in urban canyons," *arXiv*, 2020, accessed: 2024-08-30.

[38] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.