

UAV navigation in large-scale GPS-denied bridge environments using fiducial marker-corrected stereo visual-inertial localisation



Feng Wang^a, Yang Zou^{a,*}, Cheng Zhang^a, Joao Buzzatto^b, Minas Liarokapis^b, Enrique del Rey Castillo^a, James B.P. Lim^{a,c}

^a Department of Civil and Environmental Engineering, The University of Auckland, Auckland 1010, New Zealand

^b Department of Mechanical and Mechatronics Engineering, The University of Auckland, Auckland 1010, New Zealand

^c School of Engineering, The University of Waikato, Hamilton 3216, New Zealand

ARTICLE INFO

Keywords:

Localisation
Unmanned aerial vehicles (UAVs)
GPS-denied area
Bridge inspection
Stereo visual-inertial odometry
Fiducial markers
AprilTag2-based pose correction

ABSTRACT

The use of Unmanned Aerial Vehicles (UAVs) for bridge inspection has gained popularity recently; however, accurately localising the UAV in GPS-denied areas is still challenging, which hinders the development of fully autonomous UAV-assisted bridge inspection solutions. This paper proposes a fiducial marker-corrected stereo visual-inertial localisation (FMC-SVIL) method, running on a resource-constrained onboard computer, to estimate UAV's global pose underneath bridge girders. The proposed FMC-SVIL utilises an optimised stereo visual-inertial odometry for continuous relative pose estimation between consecutive camera frames and an improved AprilTag2-based measurement algorithm for accurate global referencing and periodic pose corrections. The method is validated through extensive experiments, and the results show that the FMC-SVIL achieved UAV localisation with a root mean square error of 0.416 m in sunny conditions and 0.340 m in cloudy conditions. FMC-SVIL outperforms the leading vision-based simultaneous localisation and mapping (SLAM) algorithms for flights over multiple bridge spans.

1. Introduction

Bridges are critical components of the infrastructure system, which are subject to deterioration during their lifecycle and may experience severe structural failure without proper maintenance [1,2]. As a proactive measure to monitor structural integrity, inspection can identify potential issues and inform necessary maintenance and repair interventions, which is of paramount importance to ensure the continuous safety and functionality of bridges [3,4]. Traditionally, bridge inspections have relied on equipment (e.g., bridge inspection vehicles, mobile elevating platforms) where inspectors need to physically access the bridge to assess the surface damage, which is time-consuming, labour-intensive, and dangerous, especially in hard-to-reach areas [5]. Considering the large number of ageing and deteriorating bridges across the world, developing more efficient and cost-effective bridge inspection approaches is gaining an increasing research interest.

In recent years, numerous research efforts [6,7] have been focused on developing automated and autonomous systems for inspecting bridges, where Unmanned Aerial Vehicles (UAVs) have emerged as a

promising tool due to their high mobility and cost-effectiveness [8]. Moreover, with the incorporation of suitable sensing systems, UAVs are capable of capturing various types of data, such as RGB (red, blue and green) images, point clouds, and thermal images, for different inspection purposes [9]. UAV-assisted bridge inspection often requires a highly skilled pilot to collect the necessary inspection data. However, maintaining the line-of-sight of UAVs and consistent data quality can be challenging due to the complex bridge environment and obstructions from surrounding objects, such as trees. Manual or minimally automated data collection may result in unsatisfactory data quality, such as missing areas and insufficient image resolution [10]. To address these problems, there is a growing trend of reducing human involvement and developing highly autonomous UAV systems for bridge inspection [11]. As a primary step for autonomous UAV navigation, achieving accurate and robust UAV localisation throughout the entire process is crucial [12].

For bridge inspection in open-space areas where the UAV can obtain a stable Global Positioning System (GPS) signal, the UAV can be programmed to automatically execute a pre-defined flight mission consisting of a series of waypoints for collecting the necessary inspection

* Corresponding author.

E-mail address: yang.zou@auckland.ac.nz (Y. Zou).

data [10]. For example, Lin, et al. [11] presented a 2D satellite map-based mission planner for scanning a bridge, which can generate a 3D flight mission containing continuous waypoints according to the selected map area and data collection requirements. The generated 3D flight mission is then automatically executed by DJI Phantom 4 utilising GPS navigation. Similarly, Morgenthal, et al. [13] proposed an automated UAV-assisted bridge inspection framework, where the UAV can automatically implement a flight path for navigation using the Global Navigation Satellite System (GNSS). The path is generated based on an initial 3D bridge model and pre-defined inspection requirements. Despite the high level of automation achieved, these GPS- or GNSS-based UAV navigation methods heavily rely on GPS or GNSS signals, impeding their applicability in areas where GPS signals are unavailable or unreliable (i.e., GPS-denied areas) [14].

The most common GPS-denied area for bridges is underneath the bridge girders, where inspectors may often check those critical structural components (e.g., bottom flanges of bridge girders, bearings, piers, and foundations) that are prone to damage and deterioration over time [15]. To enable UAV navigation in these bridge areas, various techniques have been employed. For example, Abiko, et al. [16], Usenko, et al. [17], and Jung, et al. [18] utilise UAV-mounted sensors, such as laser range finders, visual cameras, and Light Detection and Ranging (LiDAR) to determine the UAV's location. While these methods excel at providing accurate relative positions to bridge surfaces, they face challenges in estimating global positions and are prone to accumulating errors over long distances. Another approach involves using offboard sensors like ultrasonic beacons [19], ultra-wideband beacons [20], or equipment like total stations [21] installed in the field to estimate the UAV's global location. However, the stability of these methods heavily relies on the communication between the UAV and the external equipment, and the installation of such equipment is time-consuming and costly. Real-world bridges often exist within complex built environments and have multiple spans, necessitating cost-effective and time-efficient data collection. These factors make existing methods less than ideal for practical bridge inspection. Therefore, there is a need to develop a low-cost and highly efficient UAV localisation method to support automated inspection in complex and large-scale bridge environments.

To narrow this gap, this paper presents a novel fiducial marker-corrected stereo visual-inertial localisation (FMC-SVIL) method for localising the UAV in enclosed areas beneath bridge girders. The proposed method uses an Inertial Measurement Unit (IMU) built-in stereo camera, a set of pre-installed AprilTag bundles, and combines an optimised stereo visual-inertial odometry (SVIO) with an improved AprilTag2-based measurement (ATM) algorithm. The main novelties of the proposed FMC-SVIL are as follows:

- (1) **An optimised SVIO algorithm.** Existing local pose estimation methods based on stereo vision-based odometry or SLAM algorithms often perform feature matching between left and right images throughout the entire process, which are sensitive to variations in scene depth. The sensitivity limits their accuracy in bridge environments where depth can vary significantly. To address this problem, an optimised SVIO algorithm is proposed by refining the feature-matching strategies to improve the performance of UAV pose estimation in such bridge environments.
- (2) **An improved ATM algorithm.** AprilTag2-based measurement using a single fiducial marker and camera is subject to measurement accuracy and fluctuation, particularly in situations involving significant distances or nearly perpendicular viewing angles. In this study, the AprilTag2 algorithm is improved for accurate UAV pose estimation in bridge environments by adopting a stereo camera and AprilTag bundles. Specifically, a 2×2 AprilTag bundle with four individual AprilTag markers is designed to mitigate the measurement errors associated with near-perpendicular angles. The stereo constraint is incorporated

into an outlier detection process to eliminate erroneous measurements, resulting in more robust and reliable measurements.

- (3) **A weighted multi-sources localisation fusion approach.** A multi-sources localisation fusion approach to integrate the measurement from SVIO and ATM algorithm is proposed based on weighted pose graph optimisation to obtain the reliable and accurate global location of the UAV during long-distance flights under bridge girders.

The efficiency of the proposed FMC-SVIL method was thoroughly validated through extensive experiments. Specifically, a laboratory test was conducted to evaluate the performance of the improved ATM that incorporated a stereo camera and AprilTag bundles. The performance of the weighted pose graph optimisation was validated through a simulation experiment in which a UAV navigated through multiple bridge spans by following two typical trajectories. To evaluate the feasibility of the proposed method in real environments, an experiment involving six field flights was carried out on a real-world concrete girder bridge in both sunny and cloudy weather conditions. The positioning accuracy and computational cost of the FMC-SVIL were lastly compared against two leading vision-based SLAM algorithms (i.e., VINS-Fusion [22], ORB-SLAM3 [23]).

The remainder of this paper is structured as follows. **Section 2** provides a review of related works. In **Section 3**, the proposed FMC-SVIL is described. **Section 4** showcases lab tests and simulation experiments for validating the functionalities of our method's modules. The practical implementation and assessment of the method within a real-world concrete girder bridge environment are presented in **Section 5**. In **Section 6**, we conclude the paper and suggest future research directions.

2. Literature review

This section begins with a state-of-the-art review of UAV-assisted bridge inspection underneath the bridge girder. It is followed by a comprehensive overview of the latest developments in UAV localisation methods in GPS-denied areas. Finally, the research gap and motivation are summarised.

2.1. UAV-assisted bridge inspection underneath bridge girders

For the enclosed GPS-denied areas underneath bridge girders where inspection often takes place, the earliest UAV systems developed to assist the UAV data collection only have the capability of maintaining a constant vertical distance between the UAV and the bottom surface of bridge girders by equipping an upward two-dimensional laser range finder [16] or further integrating an optical flow sensor [24,25] for more accurate altitude estimation. However, these UAV systems were limited to automatic control of the UAV's vertical position, while its movement in the horizontal direction still needed to be manually operated by a pilot.

To further improve the level of autonomy for UAV-assisted bridge inspection underneath the bridge girder, more advanced UAV systems equipped with LiDAR, IMU, visual cameras, and ultrasonic distance sensors were developed by [17,18]. However, the utilised pose estimation methods in these two studies for autonomous data collection are dependent on an initial 3D map of the target bridge environment, which needs to be obtained from a pre-flight. The preparation of the 3D map for large-scale bridges is time-consuming and likely to exceed the time required by manual inspection, restricting these methods into practical uses.

Recent studies illustrate interest in utilising offboard equipment (e.g., total station, ultrasonic beacon system, ultra-wideband beacon system) to assist the UAV flight in the enclosed area underneath bridge girders. In [21], a ground total station is utilised to obtain the global position of UAV in a fixed and time-persistent frame. Moreover, two stereo cameras and an IMU were mounted on the UAV to estimate the

UAV's pose with respect to the local frame in GPS-denied areas. By integrating the position obtained from the total station and UAV onboard sensors, this method can provide a good position estimation even when the UAV flies beyond the line of sight of the total station. One of the main drawbacks of this method is that it requires stable data transformation between the total station and the UAV for pose estimation, which is often challenging in multi-span and long-span bridges.

In addition, beacon-based localisation systems have been utilised for UAV position estimation through the wireless communication between the UAV onboard node and the anchor nodes installed in bridge environments. Ali, et al. [19] utilised an ultrasonic beacon system to estimate the UAV's global location and achieved autonomous UAV navigation underneath the bridge girder, while the effective distance of ultrasonic beacons is normally limited. To address the range limitation, ultra-wideband (UWB) beacons were adopted in [20] to provide accurate positioning data for the UAV, which illustrated a more effective distance than an ultrasonic beacon system. Although these beacon-based methods can provide accurate global UAV location in GPS-denied areas, a pre-arranged positioning network is required, and their working range is limited by the coverage of the distributed anchor nodes. More importantly, these electrical sensors can be sensitive to humidity, and the wireless signals can be interfered with by other wireless communication systems. Despite progress made in UAV localisation in enclosed, GPS-denied areas underneath bridge girders, the automated flight over multiple bridge spans in GPS-denied areas is still a global challenge. Therefore, an urgent need arises for an accurate, cost-effective, and portable global localisation method to improve the autonomy of UAV-assisted bridge inspections underneath bridge girders in real-world applications.

2.2. UAV localisation methods in GPS-denied areas

In the field of robotics, a variety of technologies have been developed over the past years to assist UAV localisation in GPS-denied areas, as presented in Table 1. The wave-based localisation method shares the most similar mechanism with GPS, which uses a set of wireless communication sensors (e.g., Wi-Fi [26], ZigBee [27], Ultra-Wideband sensors [28]) with known positions as anchor nodes. As discussed in section 2.1, this approach enables precise position estimation within a certain range; however, it is susceptible to environmental factors and can be expensive to deploy in large-scale bridge environments.

Instead of implementing local anchor nodes, another strategy is to

use UAV onboard sensors to perform the localisation, among which Inertial Navigation System (INS) [29] is the simplest and most economical technology. In an INS, the position and orientation of the UAV are estimated relative to a known starting point by mathematically integrating the angular velocity and linear acceleration measured by an IMU [30]. Although INS is self-contained and does not require external references [31], its performance is highly prone to drift accumulation, as small errors in angular velocity and acceleration can result in significant localisation errors. Therefore, INS can only be used in a temporary scenario when the GPS signal is lost.

The LiDAR-based localisation method utilises consecutive scans collected by UAV-mounted LiDAR sensors to determine the UAV's position and orientation [32]. This method is robust to poor illumination and well-suited for UAVs operating in constrained and dark environments due to the reduced impact of lighting conditions [33]. However, the cost and power consumption associated with LiDAR sensors can pose challenges, potentially affecting UAV flight endurance. Additionally, the extensive processing required for generating point clouds, object detection, and mapping makes LiDAR-based methods less suitable for small UAV platforms and raises concerns for end-users [34].

The vision-based localisation method, typically utilising lightweight visual sensors (e.g., monocular cameras, stereo cameras, or RGB-D (Red, Green, Blue, and Depth) cameras) to determine the UAV's pose, has gained significant attention and shows potential for wide applications due to its cost-effectiveness and lightweight design [35]. Existing vision-based localisation methods can be mainly divided into two categories according to the reference frame of estimated poses: i) local pose estimation and ii) global pose estimation.

The vision-based local pose estimation generally uses visual odometry (VO) or visual SLAM techniques to track camera poses with regard to a local reference frame located at the start point [36]. The simplest VO system uses only a monocular camera and retrieves the pose changes by incrementally matching the feature of two consecutive frames and further improves the accuracy through optimisation [37]. However, it suffers from scale ambiguity since a monocular camera can only provide 2D projections of the scene and thus cannot determine the UAV location in metric scale [38].

To address the scale problem, the implementation of an RGB-D camera or a stereo camera has been introduced [39]. RGB-D cameras can simultaneously provide an RGB image and its associated depth map to address scale drift with less complexity [40]. However, the depth detection range of RGB-D cameras is limited and is greatly affected by

Table 1
Overview of the UAV localisation technologies in GPS-denied areas.

Technologies	Sensors Employed	Pros	Cons
Wave-based Localisation [26–28]	Wi-Fi, Bluetooth, ZigBee, Ultra-Wideband	<ul style="list-style-type: none"> • Low energy • Inexpensive solution 	<ul style="list-style-type: none"> • The measurement of signal strength is not reliable in realistic settings • Interference and limited coverage • High drift accumulation
Inertial Navigation System (INS) [29]	IMU	<ul style="list-style-type: none"> • Self-contained • Scene-independent • High accuracy • Robust to poor illumination 	
LiDAR-based Localisation [32]	Light Detection and Range Sensor		<ul style="list-style-type: none"> • The reflection of the signal wave is dependent on the material or orientation of the obstacle surface • Expensive solution • Suffer from scale ambiguity • Computationally expensive • Affected by illumination conditions • Measurement error • Limited detection range • Computational burden • Illumination conditions • Depth limitation by stereo baseline • Computational burden • Accumulated drift
Vision-based Local Pose Estimation [37,39,42]	Monocular Camera	<ul style="list-style-type: none"> • Low cost 	
	RGB-D Camera	<ul style="list-style-type: none"> • Directly obtain the depth information 	
	Stereo Camera	<ul style="list-style-type: none"> • Easy to retrieve image scale and depth information 	
Visual-inertial Odometry (VIO) [45,46]	Camera/cameras + IMU	<ul style="list-style-type: none"> • Light-weight sensor package 	
Visual/Visual-inertial SLAM [23,48,53]	Camera/cameras with/without IMU	<ul style="list-style-type: none"> • Reduced drift • Updated map representation • Low cost • Global reference 	<ul style="list-style-type: none"> • Computational complexity • High memory burden • Restricted coverage • Installation cost
Vision-based Global Pose Estimation [50,51]	Camera		

infrared rays [41], constraining its usage in bridge scenarios. Stereo cameras, made up of two calibrated and co-registered cameras with a baseline, can provide 3D perception through triangulation between image pairs [42]. Compared with RGB-D cameras, stereo cameras offer a more effective measurement range. Hence, they are recommended for UAV localisation in outdoor environments [43], making them an ideal option for use in a bridge environment. A primary drawback is that the accuracy of stereo camera-based localisation can be affected by the scene's scale and can decrease with increasing distance from the camera due to the baseline constraints [44].

In the field of robotics and automation, the past few years have witnessed an increasing interest focusing on leveraging scene-independent IMUs to enhance VO systems, leading to the development of sophisticated visual-inertial odometry (VIO) systems. Incorporating IMU measurements into a monocular VO system enables metric-level localisation [45]. In the case of stereo VIO (SVIO) systems, the integration of IMU measurements can facilitate accurate and robust pose estimations at a higher rate and diminish the effect of moving objects on visual sensor estimations [46]. Nonetheless, like other odometry approaches relying solely on onboard sensors, SVIO still suffers from long-term drift [47].

To address the issue of accumulated errors, odometry technologies have been extended to include global mapping and loop closure components, resulting in a SLAM system [23,48]. By constructing a map of the environment and continuously updating the robot's pose relative to this map, SLAM can reduce the effect of accumulated errors [35]. While visual and visual-inertial SLAM technologies have shown encouraging results, they still face challenges related to high computational and memory demands, especially in large-scale environments, and generally run on a standard computer [49]. Additionally, various factors, including selected feature descriptors, the efficiency of map databases, and bag-of-words models, can influence the effectiveness of the loop closure. Furthermore, these vision-based local pose estimation methods can only obtain the pose estimation with respect to the initial camera frame, making it challenging to guide UAV navigation in a bridge environment.

The vision-based global pose estimation normally leverages features within an environmental map to achieve UAV localisation with respect to a fixed global coordinate system [50]. Different from the VO or VIO methods, fiducial marker-based localisation can output an absolute position even in an environment without distinctive visual features [51]. A critical requirement for this approach is the consistent visibility of these markers by UAV cameras. However, ensuring comprehensive marker placement to cover all potential areas can be a challenging task. Consequently, there's a pressing need for innovative methods capable of offering reliable localisation even when these markers fade from UAV's view. Kayhani, et al. [52] improved the marker-based localisation by fusing inertial data using an on-manifold Extended Kalman Filter (EKF) for the application in indoor construction environments, which achieved acceptable localisation accuracy in marker-blind zones. However, this method was evaluated in a controlled laboratory environment over a short distance of several metres, and the estimates were observed to drift quickly when IMU data was used solely for prediction.

To summarise, although UAV localisation in GPS-denied areas has gained significant research interest, no existing methods can be directly deployed on a resource-constrained UAV platform to support autonomous bridge inspection in the enclosed areas underneath the bridge girder.

2.3. Summary of literature review

The state-of-the-art UAV localisation methods have distinct strengths and limitations. For instance, the SVIO system offers continuous local pose estimation, but its accuracy decreases with increasing flight distance due to accumulated error. The fiducial marker-based methods accurately estimate the global location of the UAV, but their working

range is limited by marker coverage, resulting in the potential loss of tracking in marker-blind zones. To support UAV-assisted bridge inspection in areas underneath bridge girders, our hypothesis is that SVIO and fiducial marker-based methods can complement each other. Their combined use could yield a robust UAV localisation approach that achieves accurate global pose estimation across large-scale bridge environments. Additionally, fiducial markers are cost-effective, unaffected by environmental conditions, and can be precisely placed on existing bridges at predefined locations or measured accurately after their installation.

3. Methodology

Fig. 1 illustrates the overall framework of the FMC-SVIL, which consists of three modules: SVIO, ATM, and SVIO-ATM fusion. The SVIO module takes measurements from the camera and performs continuous relative pose estimation between the current frame and the previous frame. The ATM module uses stereo image pairs to detect pre-installed AprilTag bundles and outputs a reliable global pose estimation of the UAV when the UAV flies through the pre-installed AprilTag bundles. The SVIO-ATM fusion module is designed to fuse the relative pose estimation from the SVIO module with the global pose estimation from the ATM module using a weighted pose graph optimisation method for the generation of the global pose trajectory of the UAV. In this section, we will first introduce the fundamentals in **Section 3.1**. A detailed explanation of SVIO, ATM, and SVIO-ATM fusion modules will be presented in **Sections 3.2–3.4**, respectively.

3.1. Fundamentals

In this subsection, the mathematical representation of the UAV pose is described first, and then the reference frames and essential operators used in the FMC-SVIL are described.

3.1.1. Mathematical representation of UAV pose

According to [54], UAV localisation refers to the problem of estimating a UAV's position and orientation with six-degree-of-freedom, which is mathematically represented with a 3D special Euclidean group in the form of valid 4×4 transformation matrices in **Eq. (1)**:

$$SE(3) = \left\{ T = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^3 \right\} \quad (1)$$

where, \mathbf{p} is a 3D translation (3×1) vector, \mathbf{R} is a standard (3×3) rotation matrix in the special orthogonal group of $SO(3)$ that represents rotations in 3D and is defined as:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}\mathbf{R}^T = 1, \det(\mathbf{R}) = 1 \right\} \quad (2)$$

The rotation can also be represented by a Hamilton quaternion \mathbf{q} [55], which is defined as a four-dimensional vector with scalar (real) and vector (imaginary) components as:

$$\mathbf{q} = w + xi + yj + zk \quad (3)$$

In this paper, both rotation matrices \mathbf{R} and quaternions \mathbf{q} are utilised to represent rotation in different cases.

3.1.2. Coordinate frames and notations

The coordinate frames used in the proposed FMC-SVIL system for UAV pose estimation are shown in **Fig. 2**. \vec{F}_w is the inertial frame, which is a fixed global frame, normally known as the world frame. This frame is aligned to the bridge centre in FMC-SVIL. The direction of gravity is aligned with the z -axis of the world frame. The reference frame of the UAV at time k is denoted as \vec{F}_{U_k} . Meanwhile, the vehicle frames at time $k-1$, as well as $k+1$, are also presented to indicate the UAV trajectory. In the proposed method, an IMU built-in stereo camera is required to be fixed on the UAV for pose estimation, maintaining a fixed transform

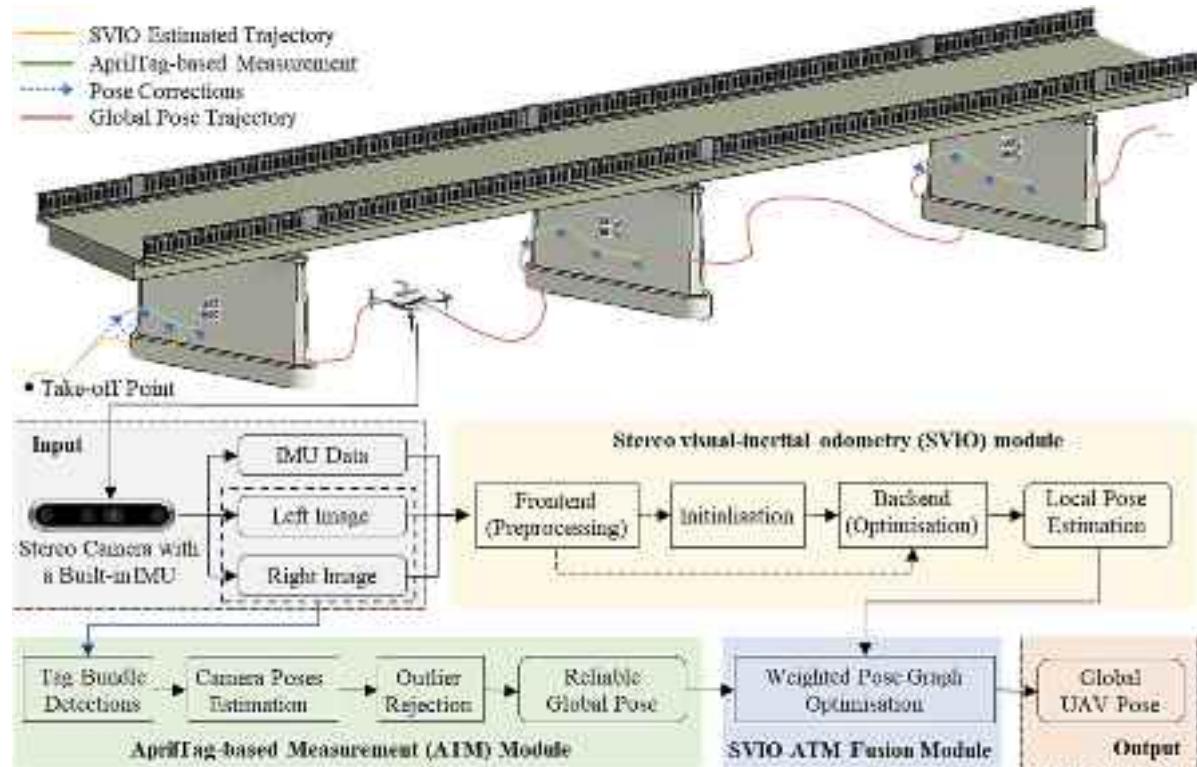


Fig. 1. Overall framework of the proposed FMC-SVIL method.

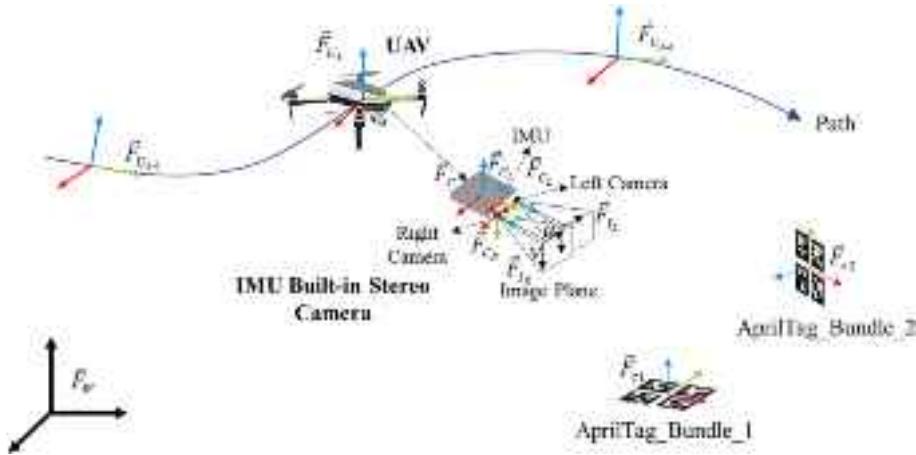


Fig. 2. Coordinate frames used by the FMC-SVIL for UAV pose estimation purposes.

matrix T_C^U from the camera body frame (\vec{F}_C) to the UAV frame (\vec{F}_U). The frames of the left camera, right camera, and IMU are denoted as \vec{F}_{C_L} , \vec{F}_{C_R} , and \vec{F}_I , respectively. For convenience, we define the camera body frame (\vec{F}_C) to be the same as the IMU frame (\vec{F}_I). The transform matrices from the left/right camera frame to the camera body frame, i.e., $T_{C_L}^C$ and $T_{C_R}^C$, can be obtained using visual-inertial sensor calibration tools (e.g., Kalibr [56]) and remain unchanged during the flight. The frames of AprilTag bundles are denoted as \vec{F}_{ij} ($j = 1, 2, \dots$, illustrating the order of the marker bundles), whose poses are known and expressed in the world frame. Additionally, we use q_C^w and p_C^w to indicate the rotation and translation from the camera body frame to the world frame. \oplus represents the multiplication operation between two quaternions. \ominus is the minus operation on the quaternions. $g^w = [0, 0, g]^T$ is the gravity vector in the world frame.

3.2. SVIO module

There are two popular categories of existing SVIO or SLAM systems for the fusion of visual and inertial data, i.e., filtering-based and optimisation-based methods. Filtering-based SVIO methods, such as Robust Visual Inertial Odometry (ROVIO) [54], integrate visual and inertial data using probabilistic state representations in a Kalman-filtering framework but face challenges related to filter initialisation and linearisation errors. Optimisation-based methods, like Open Keyframe-based Visual-inertial SLAM (OKVIS) [45], formulate the fusion problem as a nonlinear optimisation and demonstrate improved accuracy and robustness. The leading methods in this area, such as Visual-inertial Navigation System (VINS)-fusion [22] and ORB-SLAM3 [23], employ the optimisation-based framework and incorporate IMU pre-integration, online calibration, and global optimisation, resulting in

reliable ego-motion estimation. However, their high computational requirements make them unsuitable for resource-constrained small UAV platforms used in bridge inspection applications.

A lightweight SVIO system was developed in this research. Firstly, instead of tracking a large number of features for motion and depth estimation, the system focuses on tracking only a reasonable number of features per image. Two hundred features are selected and tracked throughout the process because this setting illustrated great performance in motion and depth estimation while maintaining computational efficiency in a lab experiment. Secondly, since constant feature detection and image gradient calculations on the full image are computationally expensive, we used an optical flow tracking method to track existing features from the current frame to the previous one and new features are detected when the number of tracked features is less than a certain threshold, i.e., 50%. This threshold is determined according to the triangulation principles that dictate that one point should be seen at least twice for successful depth estimation [57]. Thirdly, only the features within the current sliding window are temporarily stored in memory to reduce the data storage requirements.

In this section, we describe our design of the lightweight SVIO module that provides continuous pose estimation throughout the entire UAV flight. As depicted in Fig. 3, the developed SVIO module consists of three main parts, i.e., frontend, initialisation, and backend. The frontend processes the raw sensor data from the camera, detecting and matching image features between stereo image pairs and temporal image frames. Additionally, the acceleration and angular velocity measurements from the inertial sensor are pre-integrated to create IMU constraints between consecutive image frames. The initialisation procedure performs system initialisation, providing all necessary values for bootstrapping the subsequent nonlinear optimisation. The backend performs a tightly coupled fusion of pre-integrated IMU measurements and feature observations for optimal pose estimation in a sliding window. Details regarding each part are presented below.

3.2.1. Frontend

3.2.1.1. Image pre-processing. For the first image frame, the input stereo image pair is uniformly divided with a fixed number of grids and

enforces a minimum (150) and maximum (200) number of features in each grid, ensuring a uniform distribution of features. Then, a corner feature detector [58] is adopted to identify corner features in the left image of the stereo pair, and Kanade-Lucas-Tomasi (KLT) [59] optical flow algorithm is employed to match features between the left and right images because the descriptor-based methods require much more usage of CPU than optical flow-based algorithms [12]. For the subsequent image pairs, we employ the KLT tracking on the left image stream and right image stream separately. With this mechanism, the range limitation of the stereo baseline can be overcome because the baseline is not used for calculating the depth information during the flight. The grid that loses features is replenished by detecting new features to maintain a minimum number of features for each image. As usual in this field [12,23,45], the keyframe concept is also utilised in the SVIO module. The new coming image frame is selected as a keyframe if the average parallax between consecutive image frames exceeds 10 pixels or the percentage of tracked features versus the detected total number of features is below 50%.

For the first frame, with the matched features and the pre-calibrated baseline between stereo image pairs, true scale landmarks can be triangulated using a Structure from Motion (SfM) framework [57]. The relative rotation and translation from the subsequent frames to the first frame are recovered using a perspective-n-point (PnP) method [60]. During this process, outlier rejection is performed using a RANSAC-based scheme [61], by which the outliers with large errors are removed. In this stage, the first camera frame is set as the reference frame. All frame poses and feature positions are represented with respect to the first frame. The obtained feature observations and image frame poses in this stage will be used in a full bundle adjustment in the backend.

3.2.1.2. Pre-integration. For the IMU measurements that normally have a higher output frequency than cameras, we integrate the IMU measurements between two consecutive visual frames to simultaneously optimise the constraints of vision and IMU. The raw accelerometer and gyroscope measurements from an IMU are normally measured in its body frame, denoted as \vec{F}_{C_i} , which is the same frame as \vec{F}_C . These measurements are modelled as follows [48]:

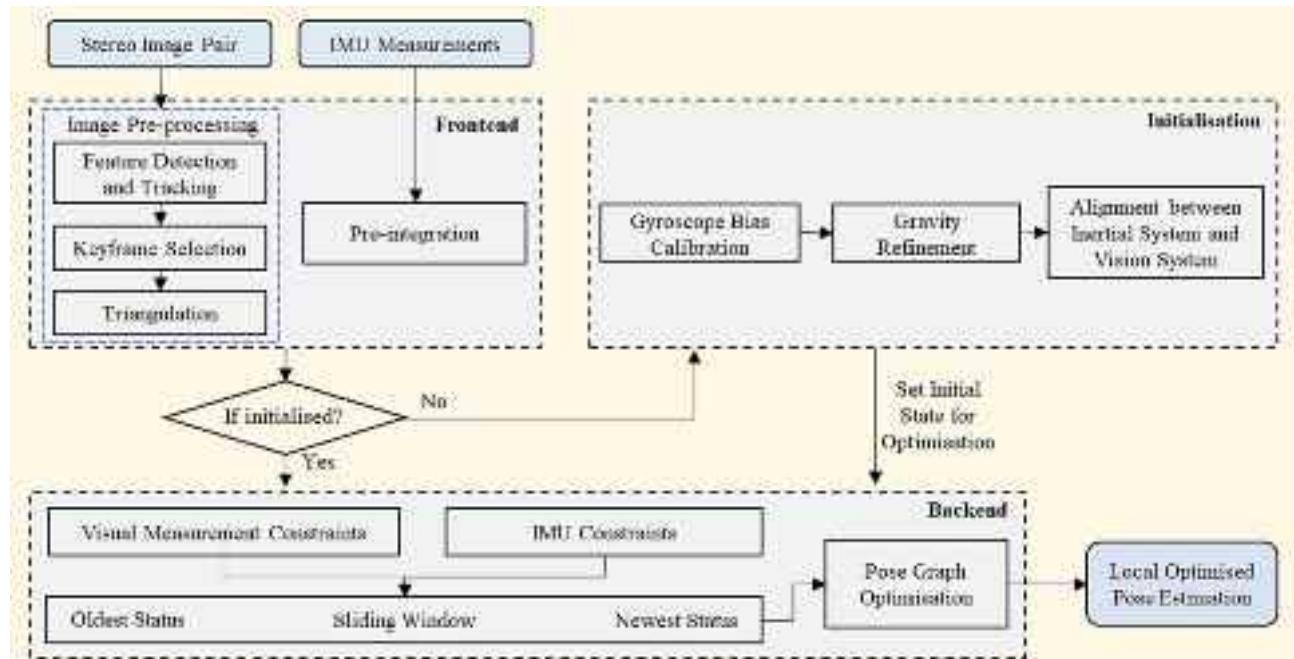


Fig. 3. Pipeline of the SVIO module.

$$\hat{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{R}_w^C \mathbf{g}^w + \mathbf{b}_{at} + \mathbf{n}_a \quad (4)$$

$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \mathbf{b}_{\omega t} + \mathbf{n}_\omega \quad (5)$$

where, $\hat{\mathbf{a}}_t$ (m/s^2) and $\hat{\boldsymbol{\omega}}_t$ (rad/s) are the raw acceleration and angle velocity measured at time t (s). \mathbf{b}_{at} and $\mathbf{b}_{\omega t}$ are acceleration bias and gyroscope bias, respectively, whose derivatives are assumed to be Gaussian white noise. \mathbf{n}_a and \mathbf{n}_ω mean the additive noise in acceleration and gyroscope measurements, which are Gaussian white noise.

For two consecutive frames \vec{F}_{C_k} and $\vec{F}_{C_{k+1}}$, several inertial measurements exist in the time interval $[t_k, t_{k+1}]$. Given the bias estimation, we obtained the pre-integrated position, velocity, and rotation measurements in a local frame \vec{F}_{C_k} , denoted as $\Delta\mathbf{p}_{C_{k+1}}^{C_k}$, $\Delta\mathbf{v}_{C_{k+1}}^{C_k}$, and $\Delta\mathbf{R}_{C_{k+1}}^{C_k}$, as follows:

$$\Delta\mathbf{p}_{C_{k+1}}^{C_k} = \iint_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^{C_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{at}) dt^2 \quad (6)$$

$$\Delta\mathbf{v}_{C_{k+1}}^{C_k} = \int_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^{C_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{at}) dt \quad (7)$$

$$\Delta\mathbf{R}_{C_{k+1}}^{C_k} = \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \boldsymbol{\Omega} (\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega t}) \mathbf{R}_t^{C_k} dt \quad (8)$$

where, $\mathbf{R}_t^{C_k}$ denotes the rotation matrix from the IMU frame at time t to the local frame \vec{F}_{C_k} .

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -[\boldsymbol{\omega}]_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, [\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (9)$$

3.2.2. Initialisation

The primary goal of initialisation is to obtain good initial values for bootstrapping the subsequent non-linear-based optimisation, including

$$\min_{\chi} \left\{ \|\mathbf{r}_p - \mathbf{H}_p \chi\|^2 + \sum_{k \in C_I} \|\mathbf{r}_{C_I}(\tilde{\mathbf{z}}_{C_{k+1}}^{C_k}, \chi)\|_{P_{C_{k+1}}^{C_k}}^2 + \sum_{(l,j) \in C_L} \rho \left(\|\mathbf{r}_{C_L}(\tilde{\mathbf{z}}_l^{C_{lj}}, \chi)\|_{P_l^{C_{lj}}}^2 \right) + \sum_{(l,j) \in C_R} \rho \left(\|\mathbf{r}_{C_R}(\tilde{\mathbf{z}}_l^{C_{Rj}}, \chi)\|_{P_l^{C_{Rj}}}^2 \right) \right\} \quad (14)$$

camera pose, velocity, gravity vector, gyroscope bias, and 3D location of feature points. For the configuration of the stereo camera and IMU system, the camera poses, velocity and 3D feature points have already been determined through triangulation during the image pre-processing. The remaining values requiring initialisation include gyroscope bias and gravity vector.

Considering two consecutive frames \vec{F}_{C_k} and $\vec{F}_{C_{k+1}}$ in the window, we get the relative rotation \mathbf{q}_{k+1}^k from the visual SfM, as well $\Delta\mathbf{R}_{C_{k+1}}^{C_k}$ from IMU pre-integration. By linearising the IMU pre-integration with respect to gyroscope bias and minimising the following cost function using Eq. (10), the initial calibration of gyroscope bias \mathbf{b}_ω can be determined [48].

$$\min_{\delta b_{\omega 0}} \sum_{k \in C_I} \left\| \mathbf{q}_{k+1}^k \otimes \Delta\mathbf{R}_{C_{k+1}}^{C_k} \right\|^2 \quad (10)$$

where, C_I denotes all frames in the window.

After initialising the gyroscope bias, we initialised the gravity vector \mathbf{g}^c , which is the gravity vector in the camera frame. Since the initialisation is completed in a stationary state of the UAV, the gravity vector is the same as the measurements from the accelerometer. After refining the gravity vector, we can get the rotation \mathbf{q}_C^{IR} between the initial reference frame \vec{F}_{IR} and the camera body frame \vec{F}_C by rotating the gravity to the

z -axis. Then, all variables with respect to the camera frame can be transformed into the initial reference frame, which is located at the UAV's take-off point and has its z -axis pointing upward.

3.2.3. Backend

After the initialisation, we proceed with the initial state within a sliding window-based tightly coupled framework based on the work in [62] for UAV pose optimisation in the backend. A nonlinear optimisation framework is formulated to find the camera poses and landmark positions by minimising the reprojection error of landmarks observed in cameras and the errors of IMU. The main states that need to be optimised include the 3D pose of the UAV, depths of visual landmarks, time-variant acceleration bias and gyroscope bias of the IMU. All these states are defined as follows:

$$\chi = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{T}_{C_L}^C, \mathbf{T}_{C_R}^C, \lambda_0, \lambda_1, \dots, \lambda_m] \quad (11)$$

$$\mathbf{x}_k = [\mathbf{p}_{C_k}^{IR}, \mathbf{v}_{C_k}^{IR}, \mathbf{q}_{C_k}^{IR}, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n] \quad (12)$$

$$\mathbf{T}_{C_L}^C = [\mathbf{p}_{C_L}^C, \mathbf{q}_{C_L}^C, \mathbf{T}_{C_R}^C = [\mathbf{p}_{C_R}^C, \mathbf{q}_{C_R}^C]] \quad (13)$$

where, \mathbf{x}_k is the basic system state of k^{th} frame in the sliding window, which corresponds to position ($\mathbf{p}_{C_k}^{IR}$), velocity ($\mathbf{v}_{C_k}^{IR}$), and orientation ($\mathbf{q}_{C_k}^{IR}$) of the camera body expressed in the initial reference frame. \mathbf{b}_a and \mathbf{b}_g are the acceleration bias and gyroscope bias of the IMU. n is the total number of frames in the sliding window. λ_l is the inverse distance of the l^{th} feature from its first observation. m is the total number of features.

The nature of pose graph optimisation is to maximise a posteriori estimation, which consists of the joint probability distribution of robot poses over a period. A visual-inertial bundle adjustment formulation [62] is used to solve the optimisation problem by minimising the Mahalanobis norm of all measurement residuals, which is defined as:

where, $\{\mathbf{r}_p, \mathbf{H}_p\}$ is the prior information from marginalisation. $\mathbf{r}_{C_I}(\tilde{\mathbf{z}}_{C_{k+1}}^{C_k}, \chi)$ is the residual for the IMU constraint between two consecutive frames, which is defined in Eq. (15). $\mathbf{r}_{C_L}(\tilde{\mathbf{z}}_l^{C_{lj}}, \chi)$ is the residual for visual measurements from the left camera, which is defined in Eq. (16). $\mathbf{r}_{C_R}(\tilde{\mathbf{z}}_l^{C_{Rj}}, \chi)$ is the residual for visual measurements from the right camera, which is defined in Eq. (17). $\rho(\cdot)$ is the Huber norm function [63].

$$\mathbf{r}_{C_I}(\tilde{\mathbf{z}}_{C_{k+1}}^{C_k}, \chi) = \begin{bmatrix} \mathbf{R}_{IF}^{C_k} \left(\mathbf{p}_{C_{k+1}}^{IR} - \mathbf{p}_{C_k}^{IR} + \frac{1}{2} \mathbf{g}^{IR} \Delta t_k^2 - \mathbf{v}_{C_k}^{IR} \Delta t_k \right) - \Delta \hat{\mathbf{p}}_{C_{k+1}}^{C_k} \\ \mathbf{R}_{IF}^{C_k} \left(\mathbf{v}_{C_{k+1}}^{IR} + \mathbf{g}^{IR} \Delta t_k - \mathbf{v}_{C_k}^{IR} \right) - \Delta \hat{\mathbf{v}}_{C_{k+1}}^{C_k} \\ 2 \left[\mathbf{q}_{C_k}^{IR-1} \otimes \mathbf{q}_{C_{k+1}}^{IR} \otimes \Delta \hat{\mathbf{R}}_{C_{k+1}}^{C_k}^{-1} \right]_{xyz} \\ \mathbf{b}_{aC_{k+1}} - \mathbf{b}_{aC_k} \\ \mathbf{b}_{\omega C_{k+1}} - \mathbf{b}_{\omega C_k} \end{bmatrix} \quad (15)$$

where, Δt_k is the time interval between time instants. \mathbf{g}^{IR} is the known gravity vector. $[\cdot]_{xyz}$ extracts the vector part of a quaternion \mathbf{q} to represent the rotation errors in the x , y , and z axes. $[\Delta \hat{\mathbf{p}}_{C_{k+1}}^{C_k}, \Delta \hat{\mathbf{v}}_{C_{k+1}}^{C_k}]$ are pre-integrated IMU measurements between two consecutive image frames, calculated through Eqs. (6)–(8).

$$\mathbf{r}_{C_L}(\hat{\mathbf{z}}_l^{C_{Lj}}, \chi) = \begin{bmatrix} \hat{\mathbf{u}}_l^{C_{Lj}} \\ \hat{\mathbf{v}}_l^{C_{Lj}} \end{bmatrix} - \pi \left(\mathbf{R}_{C_L}^{C_L} \left(\mathbf{R}_{IR}^{C_L} \left(\mathbf{R}_{C_i}^{IR} \left(\mathbf{R}_{C_L}^C \frac{1}{\lambda_l} \pi^{-1} \left(\begin{bmatrix} \hat{\mathbf{u}}_l^{C_{Lj}} \\ \hat{\mathbf{v}}_l^{C_{Lj}} \end{bmatrix} \right) \right) + \mathbf{p}_{C_L}^C \right) + \mathbf{p}_{C_i}^{IR} - \mathbf{p}_{C_j}^{IR} \right) - \mathbf{p}_C^{C_L} \right) \quad (16)$$

where, $[\hat{\mathbf{u}}_l^{C_{Lj}}, \hat{\mathbf{v}}_l^{C_{Lj}}]$ denotes the first observation of the l^{th} feature in the i^{th} image using the left camera. $[\hat{\mathbf{u}}_l^{C_{Lj}}, \hat{\mathbf{v}}_l^{C_{Lj}}]$ represents the observation of the same feature in the j^{th} image by the left camera. $(\mathbf{R}_{C_L}^C, \mathbf{p}_{C_L}^C)$ represents the rotation matrix and translation vector from the left camera frame to the camera body frame. $\pi(\cdot)$ and $\pi^{-1}(\cdot)$ are the projection and back projection functions, respectively.

$$\mathbf{r}_{C_R}(\hat{\mathbf{z}}_l^{C_{Rj}}, \chi) = \begin{bmatrix} \hat{\mathbf{u}}_l^{C_{Rj}} \\ \hat{\mathbf{v}}_l^{C_{Rj}} \end{bmatrix} - \pi \left(\mathbf{R}_{C_R}^{C_R} \left(\mathbf{R}_{IR}^{C_R} \left(\mathbf{R}_{C_i}^{IR} \left(\mathbf{R}_{C_R}^C \frac{1}{\lambda_l} \pi^{-1} \left(\begin{bmatrix} \hat{\mathbf{u}}_l^{C_{Rj}} \\ \hat{\mathbf{v}}_l^{C_{Rj}} \end{bmatrix} \right) \right) + \mathbf{p}_{C_R}^C \right) + \mathbf{p}_{C_i}^{IR} - \mathbf{p}_{C_j}^{IR} \right) - \mathbf{p}_C^{C_R} \right) \quad (17)$$

where, $[\hat{\mathbf{u}}_l^{C_{Lj}}, \hat{\mathbf{v}}_l^{C_{Lj}}]$ denotes the first observation of the l^{th} feature in the i^{th} image using the right camera. $[\hat{\mathbf{u}}_l^{C_{Rj}}, \hat{\mathbf{v}}_l^{C_{Rj}}]$ represents the observation of the same feature in the j^{th} image by the right camera. $(\mathbf{R}_{C_R}^C, \mathbf{p}_{C_R}^C)$ represents the rotation matrix and translation vector from the right camera frame to the camera body frame.

To bound the computational complexity, a sliding window including ten keyframes is utilised to limit the size of optimisation. In our work, we employed the Google Ceres solver [64], an open-source C++ library for solving nonlinear least squares problems, to get stable and optimal results efficiently. When a new frame comes, the optimisation will be executed to estimate the camera pose at this moment.

3.3. AprilTag-based measurement module

Fiducial marker-based localisation methods are often used to enhance UAV localisation in environments with poor visual features or challenging lighting conditions. In the proposed FMC-SVIL, the ATM module is designed to provide a robust global pose reference for the match with local pose estimation from the SVIO module, as well as correcting the accumulated drift. Regarding the specific algorithms, ARToolkit [65] was an early fiducial system that used black square markers for 6-DOF (Degree of Freedom) pose estimation. However, computational cost and confusion rates increased with the number of recognised patterns. To address this, Fiala [66] developed ARTag, a planar pattern marker system with a low false negative rate and inter-marker confusion. AprilTag [67], based on ARTag, introduced a lexicode-based system with a minimum Hamming distance between markers, which achieved a low false positive rate. Wang and Olson [68] redesigned the tag detector of AprilTag, introducing AprilTag2, which

showcased an improved detection rate, reduced false positive rate, and decreased computational time. Comparisons between different fiducial marker-based measurement systems by Kalaitzakis, et al. [50] have shown that AprilTag2 presents a competitive performance in terms of detection rate and measurement accuracy. Therefore, the AprilTag2-based algorithm [68] was selected in this paper.

Although the performance of the AprilTag2 has been validated in

indoor lab environments [52], implementing the AprilTag2 algorithm for UAV pose estimation in a real bridge environment poses challenges. One significant challenge is the impact of marker size on the detection range, as any marker with smaller projected side lengths in pixels is undetectable. Additionally, accurately projecting the corner points becomes challenging due to projection noise when the camera is perpendicularly projecting on a single marker, which can result in pose estimation ambiguities and sudden jumps. To enhance the robustness and accuracy of the AprilTag2 system, we have made the following modifications and extensions. Different from the original AprilTag2-

based algorithm designed to estimate the relative pose between a single marker and camera, we arranged four individual markers in a square pattern, aiming at eliminating the measurement error in near perpendicular viewing angles. Additionally, we extended the algorithm for the stereo camera to effectively filter out outlier measurements by referencing the stereo baseline. Fig. 4 presents the configuration of the improved ATM module. With the pre-known individual markers relative to the world frame and the extrinsic of the stereo camera, the camera pose can be accurately estimated in the world frame.

Fig. 5 presents the workflow of the improved ATM module, containing three main stages for global pose estimation. The red dashed line box indicates the newly introduced constraints and conditions in the

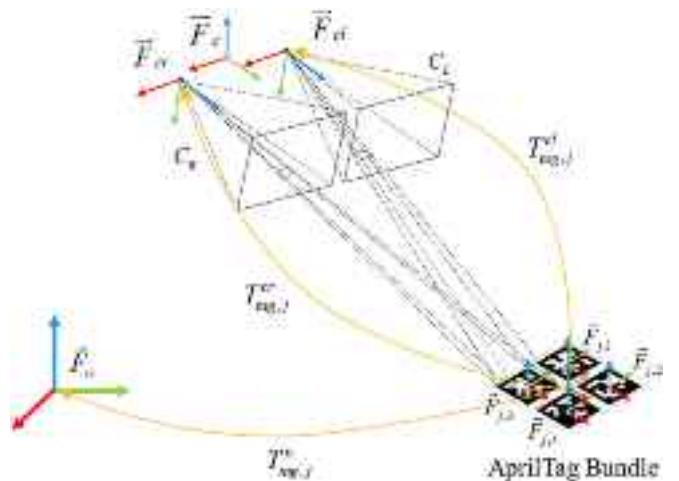


Fig. 4. Overview of the ATM using AprilTag bundles and a stereo camera.

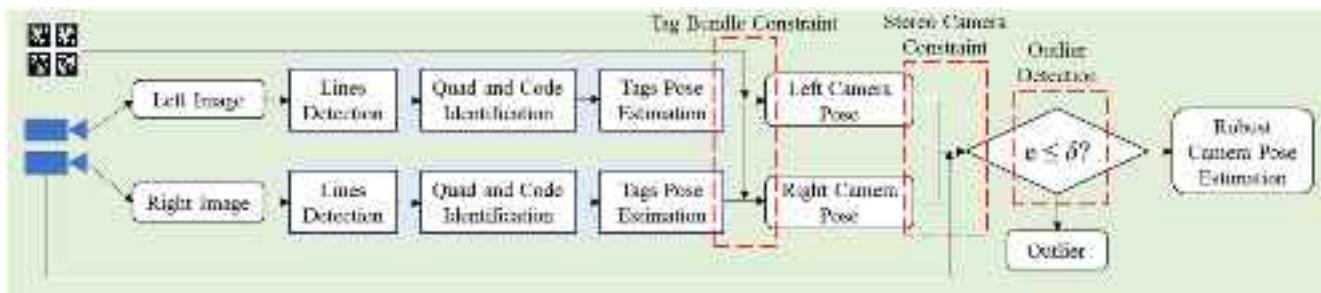


Fig. 5. Workflow of the improved ATM module.

improved ATM compared to the original AprilTag2-based algorithm [68]. Firstly, the image streams from the left camera and right camera are processed in parallel threads to detect the unique pattern of each marker and the sequences of line segments that can form a 4-sided shape (i.e., a quad). Following the detection of the quad shape and identification of its code pattern, the relative transformation between the camera frames ($\vec{F}_{C_L}, \vec{F}_{C_R}$) and each marker frame ($\vec{F}_{j,1}, \vec{F}_{j,2}, \vec{F}_{j,3}, \vec{F}_{j,4}$) can be determined using the camera's intrinsic parameters and by calculating the homography transformation.

Secondly, with the pre-known constraints between individual markers in one marker bundle, four transforms in one bundle [$T_{j,1}^{C_L}, T_{j,2}^{C_L}, T_{j,3}^{C_L}, T_{j,4}^{C_L}$] and [$T_{j,1}^{C_R}, T_{j,2}^{C_R}, T_{j,3}^{C_R}, T_{j,4}^{C_R}$] are further processed to estimate a union transform between the left/right frame and marker bundles by minimising the differences (Eq. (18)) between the union transform and these four measurements, resulting in $T_{tag_j}^{C_L}$ and $T_{tag_j}^{C_R}$.

$$\min \left\{ \sum_{i=1}^4 \left\| \mathbf{r}_{tag} \left(\hat{T}_{j,i}^C, T_{est} \right) \right\|^2 \right\} \quad (18)$$

where, $\mathbf{r}_{tag} \left(\hat{T}_{j,i}^C, T_{est} \right)$ denotes the residual for the estimated pose and the measurements, defined as:

$$\mathbf{r}_{tag} \left(\hat{T}_{j,i}^C, T_{est} \right) = \begin{bmatrix} \mathbf{p}_{est} - \hat{\mathbf{p}}_{j,i}^C \\ [\mathbf{q}_{est}^{-1} \otimes \hat{\mathbf{q}}_j^C]_{xyz} \end{bmatrix} \quad (19)$$

Thirdly, the global pose of the camera body is estimated according to the left and right camera measurements through Eqs. (20) and (21) separately. If the position difference between the pose estimation from the left and right camera is less than the threshold δ (approaching the stereo baseline, which is set at 0.1 m), the measurement is regarded as reliable and will be outputted by respectively calculating the mean value of position and orientation estimations using the Eq. (22). Otherwise, the measurement is denoted as noise and is removed.

$$(\mathbf{p}_1^w, \mathbf{q}_1^w) = \mathbf{T}_j^w \times \left(\mathbf{T}_j^{C_L} \right)^T \times \left(\mathbf{T}_{C_L}^C \right)^T \quad (20)$$

$$(\mathbf{p}_2^w, \mathbf{q}_2^w) = \mathbf{T}_j^w \times \left(\mathbf{T}_j^{C_R} \right)^T \times \left(\mathbf{T}_{C_R}^C \right)^T \quad (21)$$

$$\mathbf{p}_{avg} = 0.5 \times (\mathbf{p}_1^w + \mathbf{p}_2^w), \mathbf{q}_{avg} = SLERP(\mathbf{q}_1^w, \mathbf{q}_2^w, 0.5) \quad (22)$$

where, $(\mathbf{p}_1^w, \mathbf{q}_1^w)$ and $(\mathbf{p}_2^w, \mathbf{q}_2^w)$ are the global pose estimation in translation vector and unit quaternion of the camera from the left camera and right camera, respectively. $SLERP(\cdot)$ represents the Spherical Linear Interpolation (SLERP) function [55].

3.4. SVIO-ATM fusion module

For fusing the estimated states from SVIO and ATM modules, we adopted an optimisation-based fusion method [47] for obtaining a more accurate state estimation, as shown in Fig. 6. The measurement of SVIO and ATM is treated as a general factor. These two factors and their related states form the pose graph. The edge between two consecutive nodes is continuous local constraints, which is from SVIO. Another type of edge is discontinuous global constraints coming from the ATM module, which only exists when the UAV flies through AprilTag bundles. We add different weights to the factors because the measurement from the ATM is much more accurate than the measurement from the SVIO. The width of the line reflects the weight when fusing the SVIO and ATM. The pose estimation from the SVIO system is not globally referenced but is relative to the initial reference frame. The first pose of the camera is set as the origin to boot up the sensor. The estimation of the camera's pose incrementally evolves from the start point. The pose estimation from ATM is globally referenced and works under a fixed global frame whose origin is fixed and known in advance.

The nature of SVIO-AMT fusion is a Maximum Likelihood Estimation (MLE) problem. The MLE consists of the joint probability distribution of camera poses over a period. Variables are global poses of all nodes, $X = \{x_0, x_1, \dots, x_n\}$, where $x_k = \{\mathbf{p}_k^w, \mathbf{q}_k^w\}$. \mathbf{p}^w and \mathbf{q}^w are position and orientation under the world frame. The estimation is formulated as a nonlinear least-squares problem by [47] as:

$$\min_X \left\{ \sum_{k \in S_1} \omega_1 \left\| \mathbf{r}_1 \left(\hat{\mathbf{z}}_{C_{k+1}}^k, X \right) \right\|^2 + \sum_{k \in S_2} \omega_2 \left\| \mathbf{r}_2 \left(\hat{\mathbf{z}}_{C_k}^k, X \right) \right\|^2 \right\} \quad (23)$$

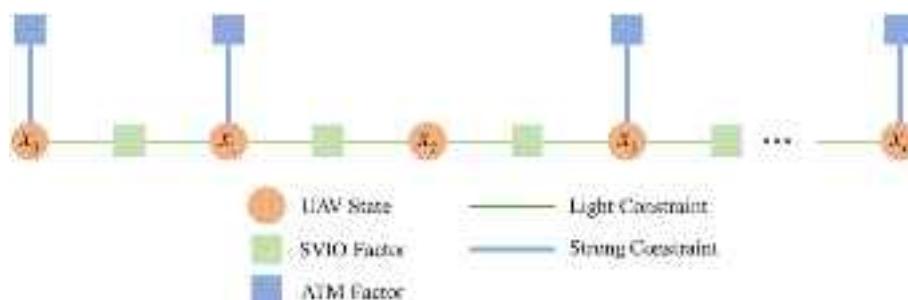


Fig. 6. Graphic illustration of the weighted pose graph for fusing SVIO and ATM.

Table 2

Configurations of the fiducial marker and the camera used in experimental cases.

Cases	Case 1	Case 2	Case 3
Fiducial Marker	AprilTag Id: 0 under the “36 h11” family Size: 160×160 mm 2	AprilTag Bundle (2 × 2) Ids: 0–3 under the “36 h11” family Size: 360×360 mm 2	AprilTag Bundle (2 × 2) Ids: 0–3 under the “36 h11” family Size: 360×360 mm 2
Camera	Intel RealSense D455 Only the left camera was used	Intel RealSense D455 Only the left camera was used	Intel RealSense D455 Both left and right cameras used

where, S_1 and S_2 are the set of pose measurements from SVIO and ATM, respectively. ω_1 and ω_2 are the weights of the SVIO factor and ATM factor. In this paper, we applied weights of 0.1 and 0.9 to SVIO and ATM, respectively. This specific weighting scheme demonstrated superior performance in the simulation environment in effectively correcting accumulated errors from the SVIO while maintaining consistent pose estimation. It should be noted that, for broad applicability, the weights should be adaptable based on the confidence levels associated with measurements from various sources. $\mathbf{r}_1(\hat{\mathbf{z}}_{C_{k+1}}^{C_k}, X)$ is the residual for the SVIO factor, which is defined in Eq. (24). $\mathbf{r}_2(\hat{\mathbf{z}}_{C_k}^w, X)$ represents the residual for the ATM factor, as defined in Eq. (25).

$$\mathbf{r}_1(\hat{\mathbf{z}}_{C_{k+1}}^{C_k}, X) = \begin{bmatrix} \hat{\mathbf{q}}_k^{IR^{-1}} (\hat{\mathbf{p}}_{k+1}^{IR} - \hat{\mathbf{p}}_k^{IR}) \\ \hat{\mathbf{q}}_k^{IR^{-1}} \hat{\mathbf{q}}_{k+1}^{IR} \end{bmatrix} \ominus \begin{bmatrix} \mathbf{q}_k^{w-1} (\mathbf{p}_k^w - \mathbf{p}_{k+1}^w) \\ \mathbf{q}_k^{w-1} \mathbf{q}_{k+1}^w \end{bmatrix} \quad (24)$$

where, $\{\hat{\mathbf{p}}_k^{IR}, \hat{\mathbf{q}}_k^{IR}\}$ and $\{\hat{\mathbf{p}}_{k+1}^{IR}, \hat{\mathbf{q}}_{k+1}^{IR}\}$ are estimated local poses in the initial reference frame from the SVIO module.

$$\mathbf{r}_2(\hat{\mathbf{z}}_{C_k}^w, X) = \begin{bmatrix} \hat{\mathbf{p}}_k^w \\ \hat{\mathbf{q}}_k^w \end{bmatrix} \ominus \begin{bmatrix} \mathbf{p}_k^w \\ \mathbf{q}_k^w \end{bmatrix} \quad (25)$$

where, $\{\hat{\mathbf{p}}_k^w, \hat{\mathbf{q}}_k^w\}$ is the estimated global pose in the World frame from the ATM module.

Once the graph is built, optimisation is conducted to find the configuration of nodes that match all edges as much as possible. Ceres Solver [64] was used to solve this nonlinear problem. After each optimisation, we can get the transformation matrix from the local frame to the global frame, i.e., T_{IR}^w . When the UAV flies out of the valid coverage of marker bundles, the latest T_{IR}^w is utilised to calculate the global pose of the UAV to achieve global pose estimation.

4. Laboratory and simulation validation

4.1. Laboratory test for assessing the improved ATM algorithm

4.1.1. Laboratory test setup

The pose estimation from the ATM module plays a crucial role in the proposed FMC-SVIL by providing a periodic position correction and an accurate global coordinate reference. As described in Section 3.3, we extended the native AprilTag 2-based algorithm [68] to improve its positioning accuracy and robustness by using a fiducial marker bundle and a stereo camera. To evaluate the performance of the modified algorithm, we conducted a comparative experiment in a laboratory environment considering three different configurations, as shown in Table 2. Case 1 only employs a single marker and a single camera. Case 2 uses a single camera and a marker bundle composed of four single markers. Case 3 adopts a stereo camera and a marker bundle for camera pose estimation. Since both distance and orientation between the marker and the camera can influence positioning accuracy [50], this laboratory test was conducted with the consideration of both distances and angles.

Fig. 7 shows the overall layout of the camera with respect to the marker or marker bundle, as well as the equipment for this test. The maximum distance from the camera to the fiducial marker is set as 5 m, while the maximum angle range is 60°. Considering the symmetry, only the right part, marked in a red square in Fig. 7 (a), was tested. The camera used in this test is an Intel RealSense D455 [69], which is composed of two infrared cameras with a baseline of 9.5 mm. The resolution of all the embedded cameras was set to 848 × 480 pixel² at 30 fps. The intrinsic and extrinsic matrixes of the camera were calibrated using an open-source toolbox, Kalibr [56], before the test. The single AprilTag used for Case 1 is id 0 under the “36 h11” family with a size of 160 × 160 mm² printed on A4 paper. For Case 2 and Case 3, the AprilTag bundle consists of four distinct AprilTags (ids 0, 1, 2, 3) with the same size as in Case 1. With a 40 mm gap between consecutive markers, the entire AprilTag bundle measures 360 × 360 mm² and was printed on A2 paper. To ensure the camera was correctly placed on the target position with the corresponding orientation, a set of small reference markers was stuck on the floor. A robotic total station (Trimble SX12 [70]) was used to accurately measure the ground truth of the relative distance between the camera and the centre of the reference markers.

4.1.2. Laboratory test results

In this sub-section, we present and discuss the results of the laboratory test. Fig. 8 illustrates the measured camera location in the x direction (Fig. 8 (a)) and y direction (Fig. 8 (b)) when the camera is placed 3 m away from the single marker with an orientation angle of 30°. Due to camera projection errors and unstable digital signals, the camera

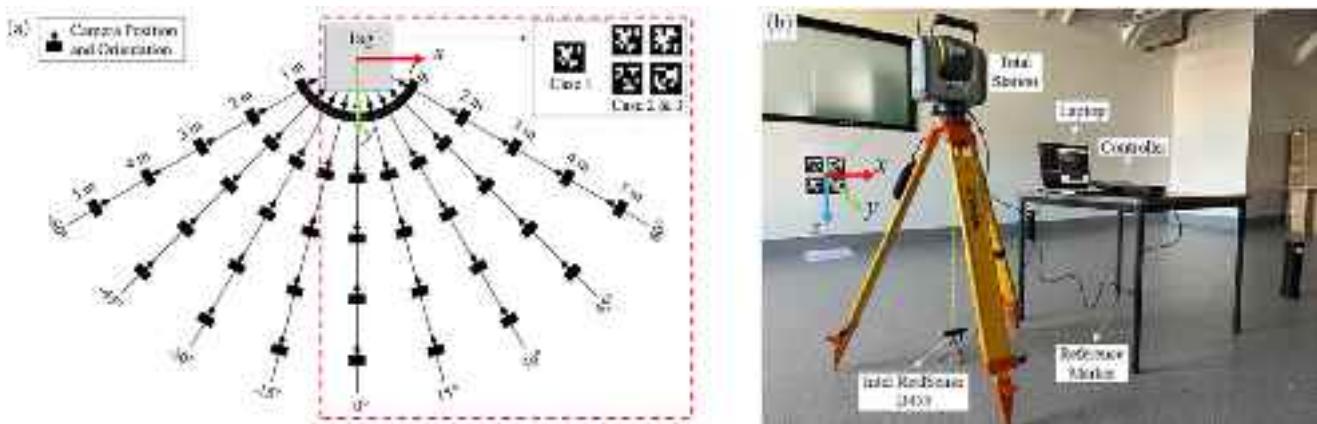


Fig. 7. Overview of the laboratory test setup: a) layout of the planned camera positions, b) equipment.

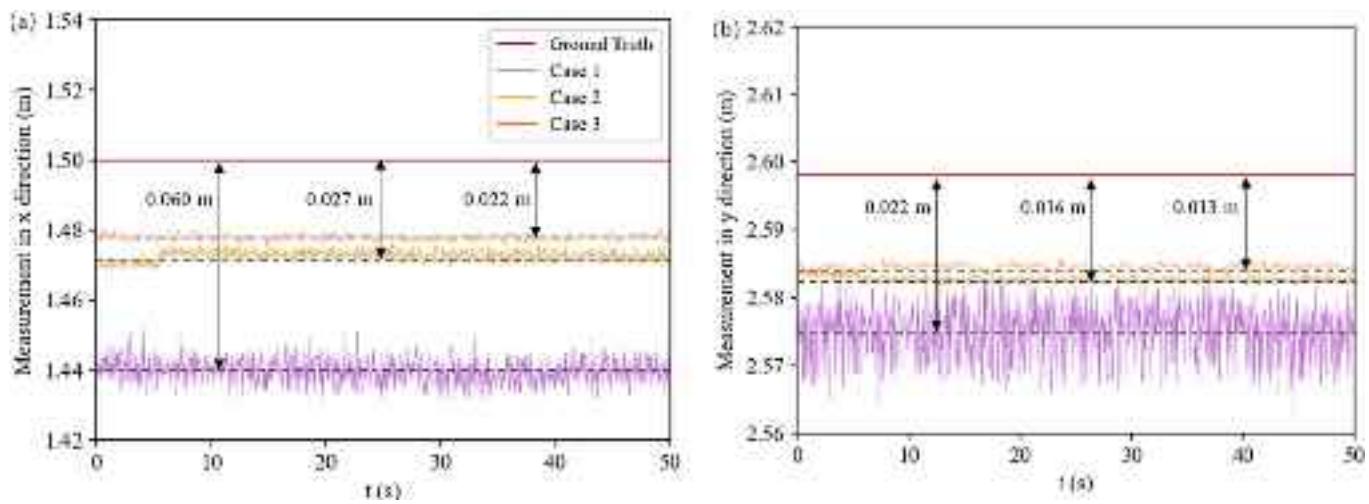


Fig. 8. Example of the raw camera localisation in the: a) x direction, b) y direction.

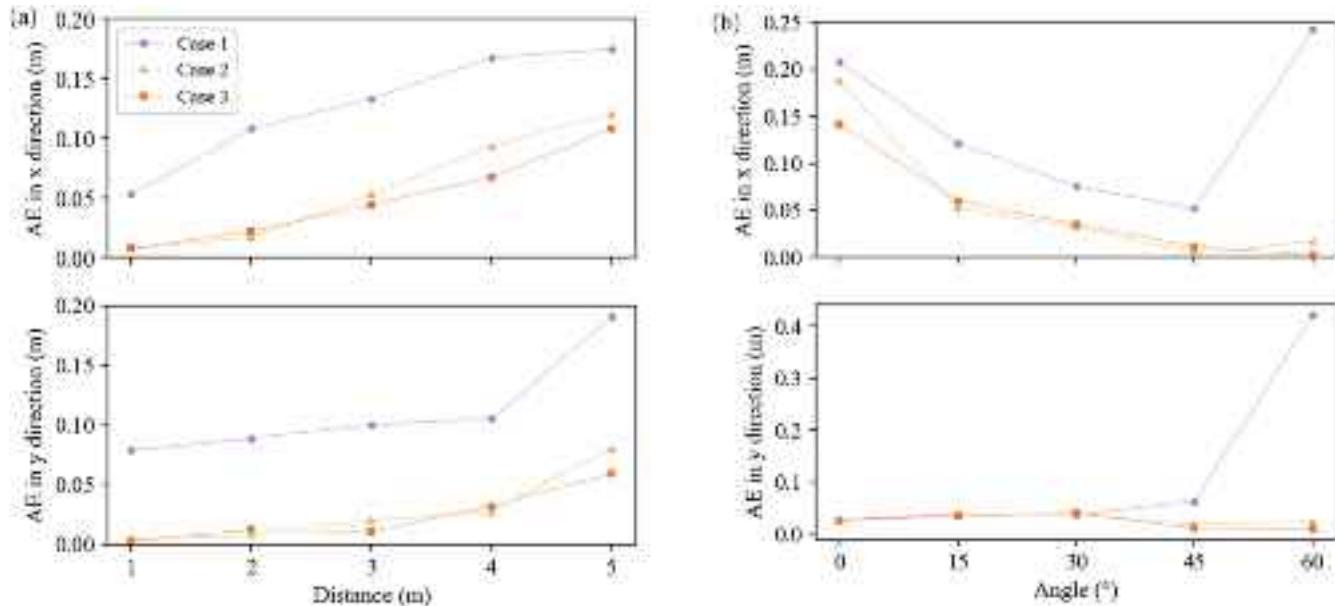


Fig. 9. Statistic results of the laboratory test: a) against the distance, b) against the angle.

location measured from the three cases shows significant fluctuation even when the camera is at rest relative to the marker. Particularly, compared with Cases 1 and 2, Case 3 provides a more stable camera localisation, depicting the accuracy and robustness of the developed ATM module based on a stereo camera and marker bundles. The mean value of the raw measurement within 10 s was calculated and further utilised to evaluate the camera localisation error against the ground truth from the total station. The developed ATM module (Case 3) reached an accuracy of 0.022 m in the x direction and 0.013 m in the y direction, which can be considered a considerable improvement compared with the error of 0.060 m in the x direction and 0.022 m in y direction using the original method using a single marker and a single camera (Case 1).

The absolute error (AE) between the tag-based measurement and ground truth from the total station was calculated for each measurement in the x and y directions. Subsequently, the statistic results were further processed to obtain a uniform result by averaging the AEs in the x and y directions for comparing the influence of distance and orientation on the accuracy of the fiducial marker-based measurement, as shown in Fig. 9. As the distance increases, the measurement error and dispersion linearly

rise in both x and y directions (Fig. 9 (a)). Compared with the single tag and camera configuration (Case 1), the measurement error can be significantly reduced by around 50% with less dispersion by employing a markers bundle (Case 2). The measurement error can be further reduced when using a stereo camera and a tag bundle (Case 3). The maximum error in the x direction and y direction using the improved ATM algorithm can be restricted below 0.1 m when the markers bundle is placed 5 m away from the camera.

In contrast, as the orientation angle increases, the measurement error generally exhibits a decreasing trend in both the x and y directions (Fig. 9 (b)), which implies that smaller orientation angles are associated with larger measurement errors. This phenomenon can be attributed to the sensitivity of camera pose estimation to small errors in detecting the marker shape, especially when the camera orientation approaches perpendicular to the marker. Even slight inaccuracies in detecting the marker shape can result in significant errors in estimating the camera pose. It can also be observed that Case 1 is more susceptible to generating outlier errors compared to Case 2 and Case 3. The developed ATM algorithm can achieve a measurement error below 0.15 m in the x direction and 0.05 m in the y direction. In summary, the positioning

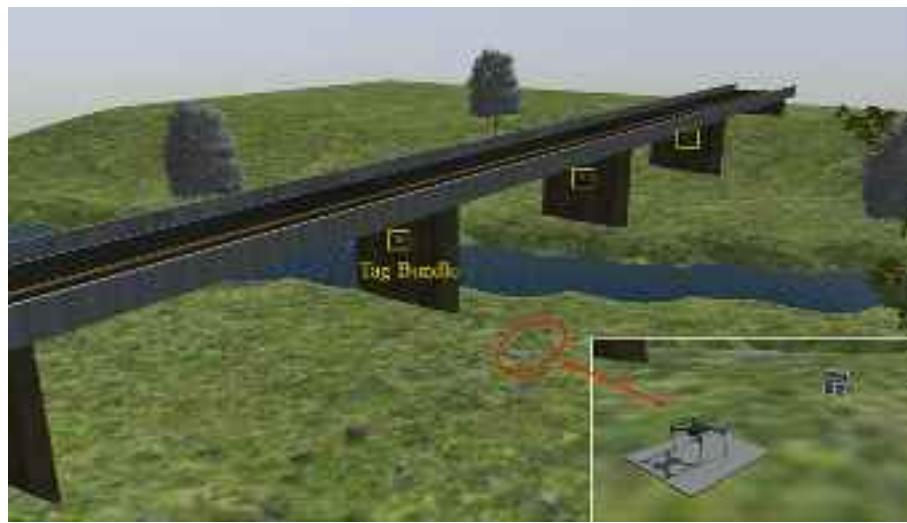


Fig. 10. Simulation environment for validating the developed algorithms.

Table 3
Parameters of the stereo camera in the simulator.

Components	Parameter	Value
Left Camera	Model	pinhole
	Resolution & Update Rate	848 × 480 pixel ² at 30 fps
	Intrinsic Parameters	$f_x = 4.2167055773095 \times 10^2, f_y = 4.2105204030899 \times 10^2,$ $c_x = 4.2012834416868 \times 10^2, c_y = 2.4335375232525 \times 10^2$
	Distortion Coefficients	$k_1 = 1.6686866642401 \times 10^{-3}, k_2 = -8.365170023851 \times 10^{-3},$ $p_1 = 6.3956192041132 \times 10^{-4}, p_2 = -5.319293698194 \times 10^{-4}$
Right Camera	Model	pinhole
	Resolution & Update Rate	848 × 480 pixel ² at 30 fps
	Intrinsic Parameters	$f_x = 4.2309967224877 \times 10^2, f_y = 4.2255428648402 \times 10^2,$ $c_x = 4.2035236798662 \times 10^2, c_y = 2.4272487054559 \times 10^2$
	Distortion Coefficients	$k_1 = 9.8558384426151 \times 10^{-4}, k_2 = -4.218173929427 \times 10^{-3},$ $p_1 = 1.0220456909956 \times 10^{-3}, p_2 = -1.450676474277 \times 10^{-3}$
IMU	Update Rate	200 Hz
	Gyroscope Noise Density	$1.82007349 \times 10^{-3}$
	Gyroscope Random Walk	$1.43678513 \times 10^{-5}$
	Accelerometer Noise Density	$1.37516511 \times 10^{-2}$
	Accelerometer Random Walk	$4.10302365 \times 10^{-4}$

performance of marker-based measurement can be compromised in situations involving long distances and small orientation angles. With the use of markers bundles and stereo cameras, the improved ATM algorithm can offer a more precise and reliable location estimation of the camera with reduced outlier data.

4.2. Simulation experiment for validating the SVIO-ATM fusion module

A simulation environment is considered a safe and efficient alternative to a real-world environment, especially for UAV-related algorithm development and validation [52]. Prior to the real-world experiment, a UAV simulation environment was developed to validate the effectiveness of the SVIO-ATM fusion module of the FMC-SVIL method. With this simulation environment, we assessed the impact of incorporating fiducial marker-based measurements on the overall performance of SVIO by comparing the estimated trajectories from FMC-SVIL with those from the SVIO module.

4.2.1. Simulation setup

The UAV simulation environment was developed based on the Gazebo simulator [71], PX4 Autopilot Package [72], and Robot Operating System (ROS) [73]. As shown in Fig. 10, the environment consists of a concrete girder bridge with a total length of around 120 m. Several trees with varied sizes and a river crossing the bridge were also modelled. The

length of each bridge span is 25 m. Three square tag bundles with a size of $360 \times 360 \text{ mm}^2$ were modelled on the side of each bridge pier at the height of 4 m above the ground. The UAV was modelled based on a standard quadcopter model, Iris, provided by PX4 Autopilot Package. A stereo camera was mounted on the UAV with a built-in IMU. The specific simulation parameters of the stereo camera and IMU are the same as the calibration results of the Intel RealSense D455 camera used in our custom-built UAV, which is developed and adopted for field testing. The parameters are listed in Table 3.

According to [74], the accuracy of a localisation method can be quantified by evaluating the estimated trajectory with respect to the ground truth. In this paper, we employed two typical flight trajectories [75] to validate the developed FMC-SVIL algorithm, namely 2D Zigzag flight trajectory (Fig. 11 (a)) and 3D S-shaped trajectory (Fig. 11 (b)). The specific parameters are summarised in Table 4. For each flight path, the UAV first took off from the start point and ascended to the height of 4 m. Then, it flew to follow the pre-defined waypoints on the trajectory. After travelling to the last waypoint, the UAV went up to 10 m and returned to the take-off point. A flight control algorithm was developed to guide the UAV to complete the flight missions automatically. To obtain the global reference at the take-off point, an additional marker bundle was placed in front of the UAV, by which the global pose of the UAV can be determined. The simulation experiments run on an Alienware m15 R4 [76] with an Intel(R) Core (TM) i9-10980HK CPU and an

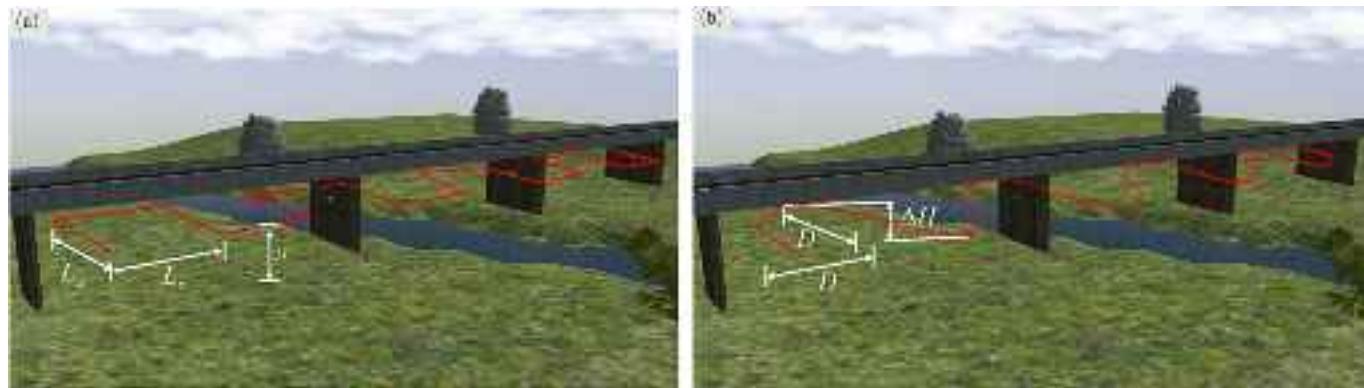


Fig. 11. Two adopted flight trajectories: a) 2D planar Zigzag, b) 3D S-shaped.

Table 4
Parameters of the custom-designed flight trajectories.

ID	Name	Lateral distance (m)	Longitudinal Distance (m)	Height above the ground (m)	Total Longitudinal Length (m)	Total distance (m)
1	2D Planar Zigzag	8	8	4	64	136
2	3D S-Shaped	8	8	2–6	64	109.8

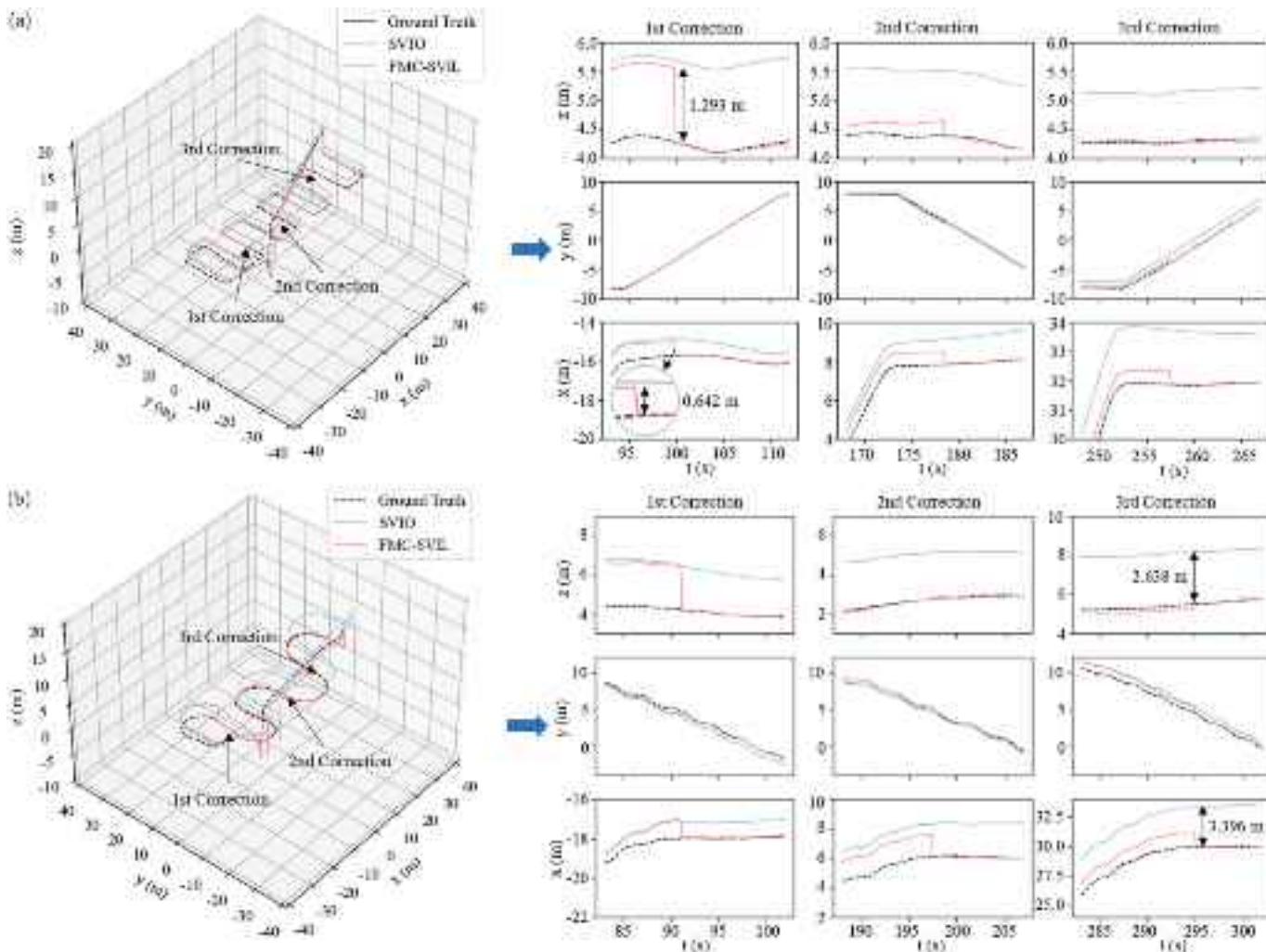


Fig. 12. Comparison of the estimated trajectories against the ground truth: a) Zigzag trajectory, b) 3D S-shaped trajectory.

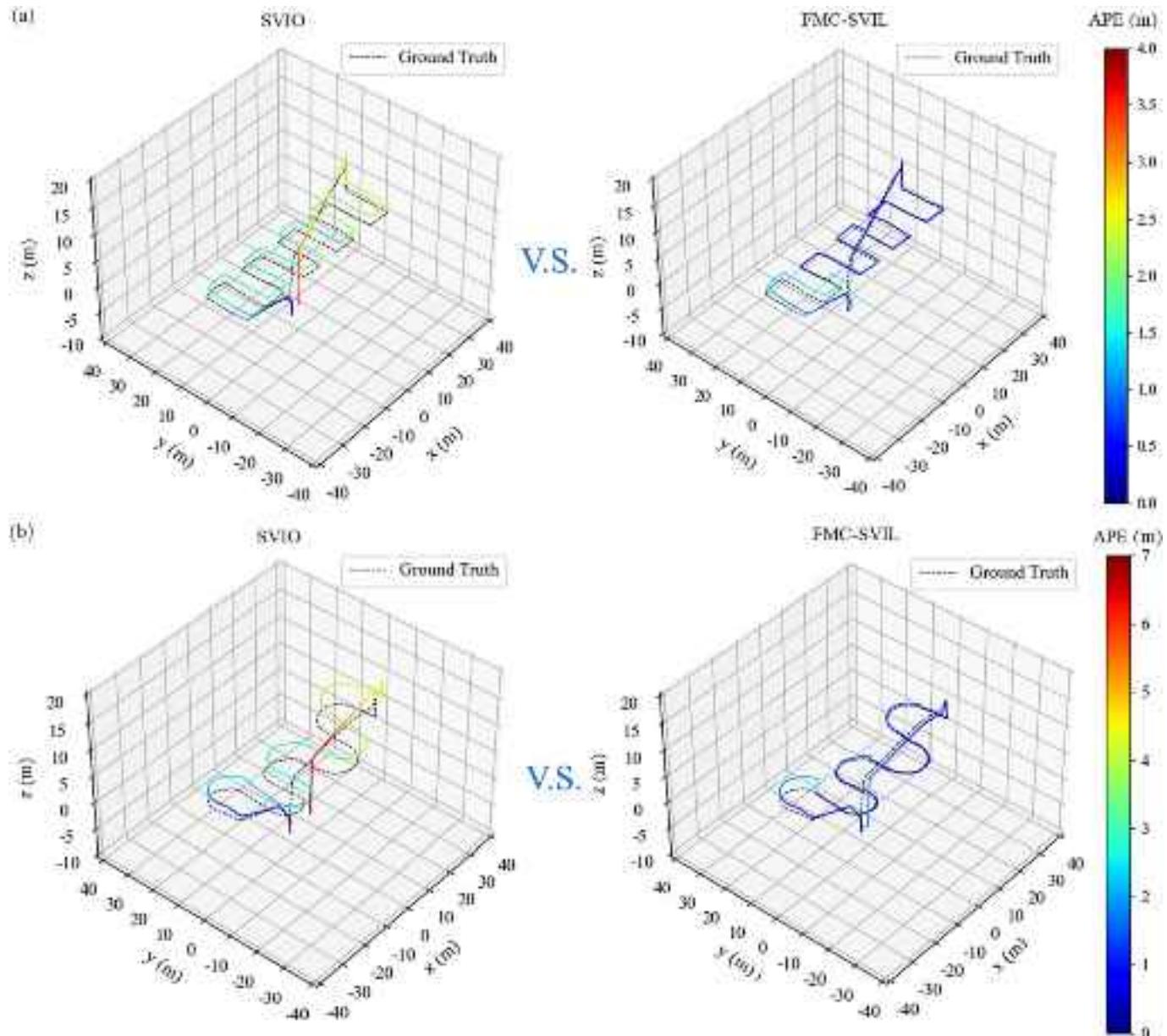


Fig. 13. APE map for the estimated trajectories: a) Zigzag trajectory, b) 3D S-shaped trajectory.

Table 5

Statistics of the comparison between the estimated trajectory and ground truth.

Trajectory	Method	RMSE (m)	Mean (m)	Median (m)	Std (m)	Min (m)	Max (m)
2D Zigzag	SVIO	2.077	1.981	1.900	0.626	0.002	3.320
	FMC-SVIL	0.845	0.710	0.638	0.460	0.001	1.608
3D S-Shaped	SVIO	3.464	3.102	2.743	1.541	0.006	6.824
	FMC-SVIL	1.032	0.800	0.775	0.651	0.005	2.495

NVIDIA GeForce RTX 3080 Laptop Graphics Processing Unit (GPU). The operating system is Ubuntu 20.04 LTS.

To illustrate the efficiency and merits of the proposed FMC-SVIL algorithm in UAV pose estimation, a comparative analysis was conducted. The performance of both the FMC-SVIL and the SVIO was evaluated by comparing their respective estimated trajectories against the ground truth. The ground truth of the UAV was extracted by subscribing to a ROS topic, “ModelStates”, from the Gazebo. To quantify the localisation accuracy, the absolute position error (APE), which measures the discrepancy distance between the true and estimated trajectory

points, was used for investigating the global consistency of a trajectory estimated by localisation algorithms. Prior to calculating the APE, the true and estimated trajectories were aligned based on their timestamps. Subsequently, the open-source package “evo” [77] was employed to calculate the APE and analyse the results.

4.2.2. Simulation results

Fig. 12 presents the raw trajectories in 3D as well as specific fiducial marker-based corrections, which have been aligned to a global reference with respect to the world frame. It is evident that the SVIO module

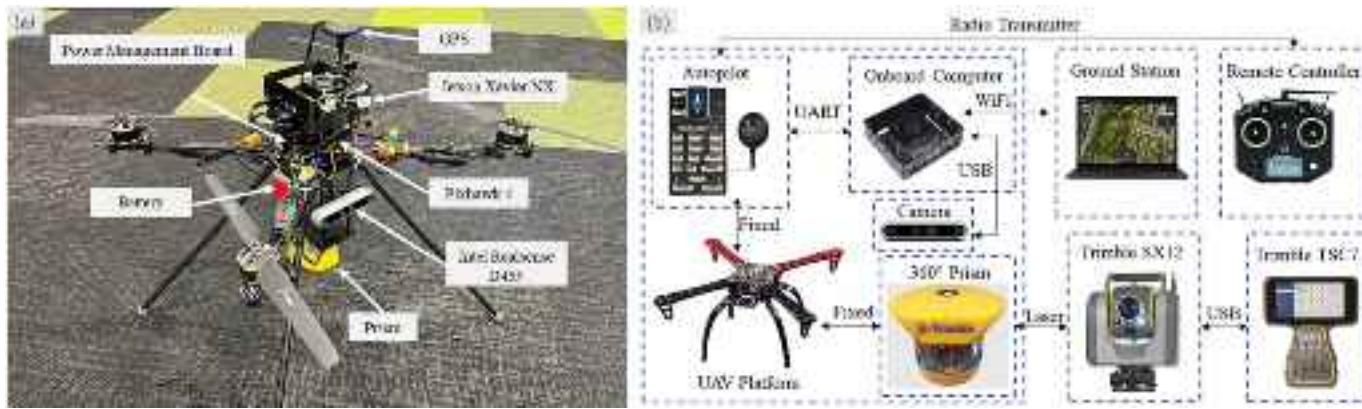


Fig. 14. UAV system for validating the proposed method: a) physical platform, b) system architecture.

exhibits accumulated errors in UAV localisation, while the proposed FMC-SVIL algorithm demonstrated increased accuracy, especially when the UAV flies through the pre-installed AprilTag bundles. The proposed ATM algorithm effectively corrects the UAV localisation for the Zigzag flight trajectory, as shown in Fig. 12 (a). During the first correction, the proposed algorithm reduced the UAV localisation error by approximately 0.642 m in the x direction and 1.293 m in the z direction during the first correction. Similar satisfactory results can also be observed in the second and third ATM-based corrections. In the case of the S-shaped trajectory, the SVIO module shows larger errors, which reached approximately 3.396 m in the x direction and 2.638 m in the z direction during the third correction, as shown in Fig. 12 (b). On the contrary, UAV location estimated by the proposed FMC-SVIL aligned well with the ground truth. Overall, the discrepancy between the pose estimation from the SVIO module and the ground truth accumulated till the end. For the results of FMC-SVIL, notable corrections can be observed when the UAV flies through the pre-installed AprilTag bundles. Through the periodical correction, the FMC-SVIL yields pose estimations that closely approximate the ground truth, effectively limiting the extent of disparity within a narrow range.

To quantify the accuracy of the pose estimation, the APE was calculated for each trajectory. The results are presented in Fig. 13. The APE for the SVIO module is accumulated towards the biggest error of 3.320 m for the Zigzag flight, while this maximum error for a 3D trajectory is 6.824 m. Through the periodical corrections by the fiducial marker-based measurement, the maximum error for the Zigzag trajectory and 3D S-shaped can be reduced to 1.608 m and 2.495 m, respectively. The statistical results of the APE are summarised in Table 5. The Root Mean Square Error (RMSE) of APE is reduced to one-third from the SVIO to the proposed FMC-SVIL algorithm. Another finding is that the APE for FMC-SVIL in both 2D and 3D trajectories between two

consecutive corrections remains low, illustrating the feasibility of the FMC-SVIL for UAV autonomous navigation. In summary, the simulation experiment results illustrated the periodical corrections provided by the ATM module can reduce the accumulated error from the SVIO module. The size and layout of the AprilTag bundles in the simulation are feasible to be employed in real-world bridge experiments for further validation.

5. Real-world experiment validation

To further investigate the performance of the proposed method in a real bridge environment, we built a customised UAV system for this research and tested the proposed FMC-SVIL method for a real-world bridge. The performance of the FMC-SVIL method was evaluated by comparing its pose estimation results with the UAV trajectory obtained by a robotic total station, which serves as a close approximation to the ground truth. Additionally, its performance and computational cost were benchmarked against two prominent vision-based SLAM algorithms with loop closure detection, i.e., VINS-Fusion [22] and ORB-SLAM3 [23]. Fig. 14 presents the overall system architecture of the custom-built UAV platform. Since the proposed method for UAV localisation requires one stereo camera and IMU for pose estimation, Intel RealSense Camera D455 [69] was selected because it provides a wide field of view and is designed with a large baseline. Additionally, it projects infrared light, which performs well even in a low-light environment. The onboard computer used is Nvidia Jetson Xavier NX [78], which is selected for its powerful GPU and CPU performance. The specifications for the custom-built UAV system and the robotic total station used for obtaining ground truth are shown in Table 6.

5.1. Experiment setup

In this study, a classic girder bridge was selected for the real-world test, as shown in Fig. 15. It contains eight spans for a total length of about 200 m. The width of the bridge is 8.5 m. The distance between the centre of two adjacent bridge piers is about 25 m, and the height of the bridge pier is around 7.5 m. We selected three spans with enough space under the bridge deck for UAV flights. One of the spans crosses a river. The standard AprilTag bundles of the same size as in the simulation environment were stuck to each bridge pier, leading to a total of three AprilTag bundles, as shown in Fig. 15 (a). Moreover, an additional AprilTag bundle was put in front of the UAV at the take-off point for the initial global pose estimation.

Considering the light conditions may affect the performance of the vision-based localisation method, we conducted the test in both sunny and cloudy weather conditions. For each weather condition, we conducted three separate tests, resulting in a total of 6 tests. Table 7 presents the overview of these six tests. The first three tests were conducted on a sunny day, and the other three tests were carried out in a cloudy

Table 6
Specifications of the UAV platform and equipment used for the experiment.

Components	Details
Body	Carbon Fibre Frame
Battery	Li-Po Battery with 6000 mAh, 24.2 V
Autopilot	Pixhawk 4
Motors	MN4014 KV400
ESC (Electronic Speed Controllers)	AIR 40 A 6S supporting a signal frequency of up to 621 Hz
Propeller	Polymer Straight Propeller
GPS module	NEO-M9N GNSS module
Cameras	Intel Realsense D455
Onboard Computer	Jetson Xavier NX
Ground Station	Alienware running QGroundControl
Total Station	Trimble SX12 Scanning Total Station (Millimetre precise positioning)
Total Station Controller	Trimble TSC7
Prism	Trimble 360 Prism



Fig. 15. Real-world experiment: a) equipment setup, b) screenshot of the UAV tracking using the total station.

Table 7

Overview of the real-world flight tests.

Cases	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Weather Conditions	Sunny	Sunny	Sunny	Cloudy	Cloudy	Cloudy
Flight Time (s)	460.400	482.675	527.600	521.070	311.070	550.272
Flight Distance (m)	246.597	254.836	307.350	278.836	198.608	247.840
Max. Velocity (m/s)	4.814	2.735	7.218	4.239	2.847	2.375
Avg. Velocity (m/s)	0.527	0.533	0.581	0.539	0.646	0.462

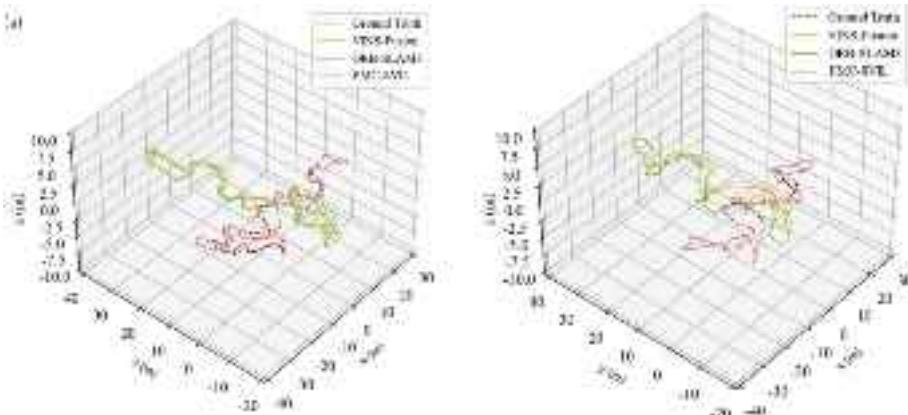


Fig. 16. Example comparison of 3D trajectories for: a) Test 2, b) Test 5.

condition with a gentle breeze. The time stamps of all these estimated trajectories are aligned according to the measurement in the z-direction. During each flight, we manually operated the drone through the selected bridge spans, which consisted of taking off, flying to the far-left end of the chosen spans, traversing all three spans, and then returning to the takeoff location. Due to the lack of a downward vision-assisted localisation system of the custom-built drone, we carefully flew it and kept enough safety distance to the bridge and surrounding trees. Regarding the ground truth of the UAV location, the Trimble SX12 [70] scanning total station with millimetre positioning accuracy was utilised to track the UAV during the flight to obtain its ground truth positions, as shown in Fig. 15 (b). To compare the localisation performance of our proposed method and the leading SLAM algorithms, the images and IMU streams were saved as ROS bag files for post-processing.

5.2. Positioning accuracy evaluation

Fig. 16 presents the estimated raw trajectories of Test 2 and Test 5, which are conducted in sunny and cloudy conditions, respectively. The

trajectories estimated by the VINS-Fusion and ORB-SLAM3 algorithms are referenced to the initial frame at the take-off point, resulting in a significant difference from the ground truth due to the absence of global references. In contrast, the trajectory obtained by the proposed FMC-SViL method exhibits good agreement with the ground truth, highlighting its effectiveness and accuracy in providing global UAV localisation with respect to the bridge frame. It should be noted their estimated trajectories were further transformed to the global frame to evaluate the accuracy of the VINS-Fusion and ORB-SLAM3 algorithms by aligning the first global pose estimated by the ATM module.

For further quantifying the accuracy of the estimated trajectories, we calculated the APE against the ground truth. Fig. 17 (a) and (b) depict the evaluated APE along the entire trajectory of Test 2 and Test 5, respectively. As the UAV's flight distance increased from takeoff to the far end, the APE in UAV pose estimation exhibited a rising trend when employing the VINS-Fusion and ORB-SLAM3 algorithms. When the UAV began its return flight, a decreasing trend was observed, primarily attributed to the efficacy of the loop closure mechanism. Despite the reduction in accumulated errors by the SLAM algorithms, notable errors

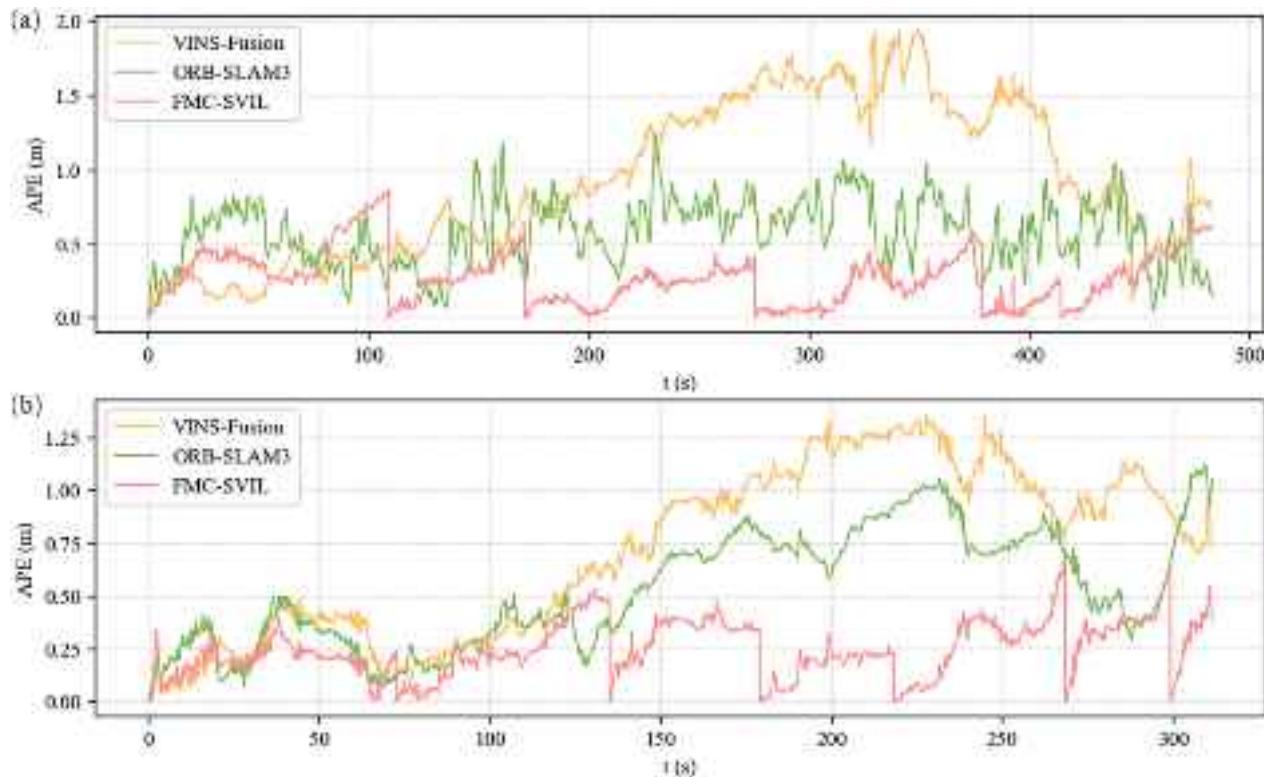


Fig. 17. Example comparison of the absolute position error for: a) Test 2, b) Test 5.

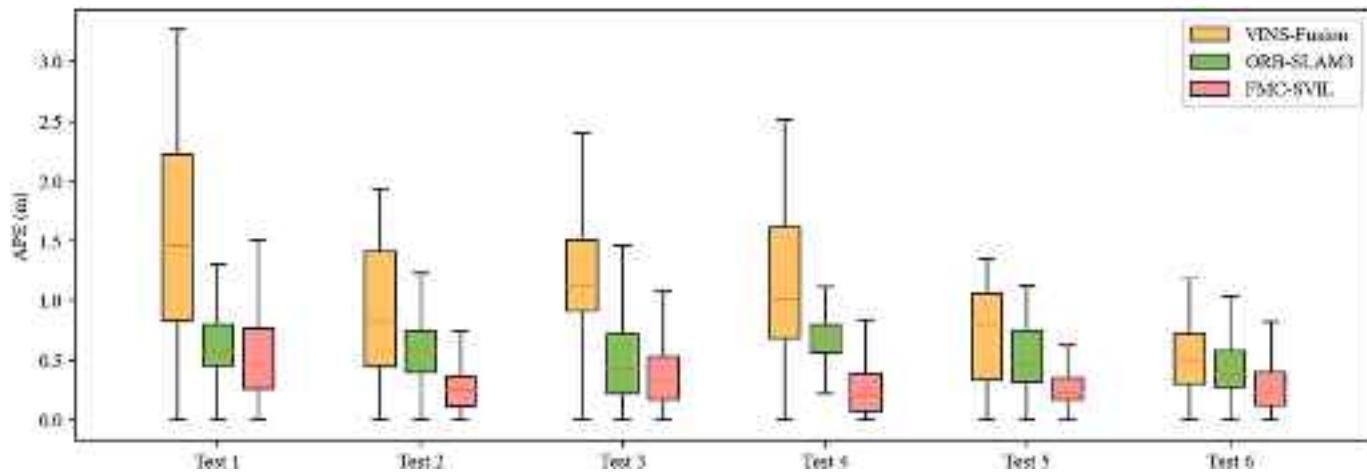


Fig. 18. Statistic results of the absolute position error for all tests.

Table 8

Comparison of the RMSE between the proposed method and the leading SLAM algorithms.

Methods	Test 1 (m)	Test 2 (m)	Test 3 (m)	Average (m)	Test 4 (m)	Test 5 (m)	Test 6 (m)	Average (m)
VINS-Fusion	1.760	1.053	1.322	1.378	1.568	0.819	0.587	0.991
ORB-SLAM3	0.689	0.618	0.615	0.641	0.694	0.596	0.629	0.640
FMC-SVIL	0.608	0.325	0.461	0.465	0.375	0.283	0.362	0.340

persisted. In sunny conditions, the maximum APE for the ORB-SLAM and VINS-Fusion algorithms in Test 2 exceeded 1.225 m and 1.940 m, respectively. In contrast, the proposed FMC-SVIL method consistently demonstrated effective periodic error correction through the ATM module, resulting in a maximum APE below 0.9 m in sunny conditions and below 0.720 m in cloudy conditions.

The statistic results of trajectory APE for all tests are summarised in Fig. 18. Compared with two leading SLAM methods with loop closure, the proposed FMC-SVIL method has a low APE for all tests, demonstrating its accuracy and robustness for UAV localisation in GPS-denied bridge areas. Table 8 compares the RMSE between the proposed FMC-SVIL method and these SLAM algorithms. VINS-Fusion exhibits the

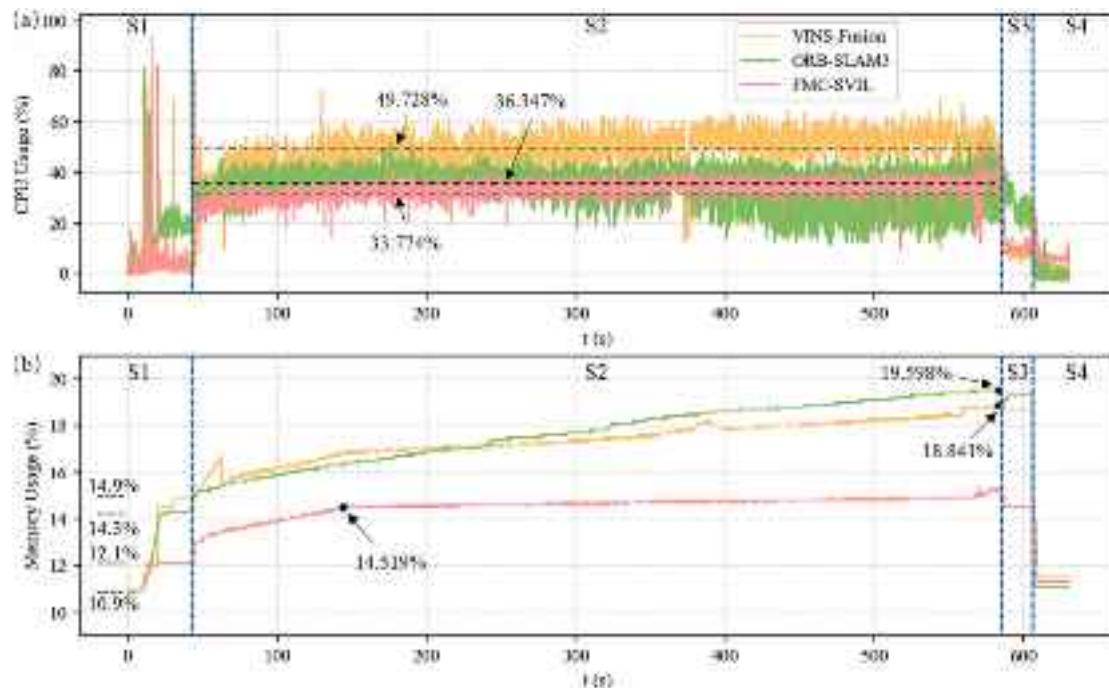


Fig. 19. Comparison of computational cost for Test 2: a) CPU usage, b) memory usage.

largest error, with an average RMSE of 1.378 m in sunny conditions and 0.991 m in cloudy conditions. ORB-SLAM3 can significantly reduce this error compared to VINS-Fusion. The proposed FMC-SVIL method improves upon this further, resulting in an average RMSE of 0.465 m in sunny conditions and 0.345 m in cloudy conditions, depicting its effectiveness and robustness for UAV localisation in GPS-denied bridge environments. As the average accuracy of GPS-based consumer-grade UAVs is around 1–4 m [79], the proposed FMV-SVIO holds the potential to be applied as an alternative for developing fully automated UAV solutions for under-bridge inspections. Additionally, it is found that vision-based localisation methods, including the VINS-Fusion and the proposed FMC-SVIL, can perform more accurate pose estimation in cloudy conditions than in sunny conditions. One reason for this may be that cloudy conditions typically provide more uniform lighting across the scene, resulting in less variation in brightness and contrast, which can improve the stability of visual feature tracking.

5.3. Computational cost evaluation

To further assess the computational efficiency of the proposed method, we conducted a comparison of the CPU and memory usage. All algorithms, i.e., FMC-SVIL, VINS-Fusion and ORB-SLAM algorithms, were executed on a standard laptop running Ubuntu 18.04 to ensure a fair comparison, as there were delays when running VINS-Fusion on the UAV onboard computer (i.e., Jetson Xavier NX). The laptop was equipped with an Intel Core (TM) i7-4720HQ CPU @ 2.6GHz and 16 GB of memory. To ensure a consistent basis for comparison, all algorithms were executed immediately after a system reboot, with no other applications running apart from essential system processes. We recorded the CPU and memory usage at a frequency of 10 Hz. Fig. 19 provides an example of CPU and memory usage for Test 2, which spans four stages denoted as S1 to S4. S1 represents the launch of algorithms without receiving data from the camera, followed by continuous pose estimation using the received raw camera data in S2. S3 signifies a period with no new data from the camera while the algorithms are still running. S4 occurs after the termination of the algorithms.

From Fig. 19 (a), it is observed that the most computationally intensive phase is the loading of the algorithms. When the algorithms

started to process continuous data from the camera, CPU usage stabilised at a consistent level with fluctuations. Once image streams and IMU measurements stopped, CPU usage rapidly decreased and then returned to a minimal level upon procedure termination. To gauge the computational cost of the algorithms, we calculated the mean value of CPU usage during stage 2. The average CPU usage for the proposed FMC-SVIL is 33.774%, demonstrating a 2.573% reduction compared to ORB-SLAM3 and a 15.954% reduction compared to VINS-Fusion. For the memory usage in Fig. 19 (b), a significant disparity can be seen. The loading phase of VINS-Fusion and ORB-SLAM during S1 consumed nearly three times more memory than the proposed FMC-SVIL. This is caused by the need for SLAM algorithms to load a bag-of-words library for place recognition. As the algorithms start to process data from the camera, memory usage for VINS-Fusion and ORB-SLAM3 continuously increases till the end. The memory usage for VINS-Fusion reached a maximum of 18.841%, and the ORB-SLAM3 can go up to 19.598%. In contrast, the memory usage for the proposed FMC-SVIL peaked at 14.519% after a period and remained at this level throughout the process. The improved performance is owing to the removal of loop detection and the optimised feature storage strategies.

6. Conclusions and future work

This paper proposed a new low-cost UAV localisation method for UAV-assisted bridge inspection in GPS-denied environments by combining the use of SVIO and fiducial marker-based measurements. Specifically, we developed a lightweight SVIO method using an optimisation-based VIO framework for obtaining relative pose estimation between consecutive camera frames. The ATM algorithm was extended for use with marker bundles and stereo cameras for more accurate and reliable global pose estimation. By registering the local pose estimation into a global frame and correcting the accumulated error by the SVIO, the proposed method can provide robust global location estimation during long-distance flights. The proposed method was evaluated and validated through laboratory, simulation, and real-world experiments. Based on the experiment results, the following conclusions can be drawn:

1. The improved ATM algorithm for use with 2×2 tag bundles and stereo cameras can provide more robust and accurate pose estimation against the original individual marker-based measurement by around 50%. The measurement error can be restricted below 0.1 m when a marker bundle with the size of $0.36 \times 0.36 \text{ m}^2$ is placed within 5 m (Fig. 9).
2. The optimised lightweight SVIO algorithm can run efficiently on the resource-constrained computer onboard a small UAV platform, which addresses the depth detection limitation caused by the stereo baseline by employing separate feature matching of the left and right image streams.
3. The use of fiducial marker-based measurement as complementary support to SVIO-based localisation can mitigate the accumulated error and register the position data into a global coordinate frame, yielding a more robust and accurate pose estimation (Fig. 12).
4. The proposed FMC-SVIL can correct UAV localisation errors at a certain distance and performs better than the leading stereo vision-based SLAM methods with about 30% accuracy improvement for UAV flights in the GPS-denied areas underneath multi-span bridge girders (Table 8).
5. The proposed FMC-SVIL exhibits lower computational costs and around a 50% reduction in memory usage compared to the VINS-Fusion and ORB-SLAM3 algorithms (Fig. 19).
6. The FMC-SVIL demonstrates superior positioning accuracy compared to GPS-based localisation (commonly used in commercial UAVs), showcasing its potential to be used in high-level UAV autonomy in inspecting GPS-denied bridge environments.

The main contributions of this paper can be summarised as:

1. An FMC-SVIL method was proposed to accurately determine the global UAV location in GPS-denied bridge areas, enabling reliable UAV navigation across multiple spans underneath bridge girders.
2. A lightweight SVIO algorithm, specifically designed for a resource-constrained UAV onboard computer, was developed. By processing the left and right image streams separately and incorporating the pair constraints during the optimisation stage, the range limitation of the stereo baseline can be overcome to deal with the depth-varying bridge environment.
3. An improved ATM algorithm using 2×2 marker bundles and stereo cameras was introduced. By incorporating marker bundle constraints and stereo baseline constraints, the improved ATM can reduce measurement fluctuation and outlier data, resulting in more precise and reliable location estimation.
4. A weighted pose graph optimisation framework for pose fusion from multiple sources was introduced. By assigning different weights to each source, the fusion output is more likely to achieve accurate pose estimation close to the ground truth.

In most bridge environments, the proposed method can be readily implemented with only one fiducial marker installation per bridge pier, significantly reducing the installation workload. Once installed, these fiducial markers can be used permanently for future routine bridge inspections. For specific scenarios where the bridge environment lacks distinctive features or there is a substantial distance between bridge piers, the accumulated positional error may exceed tolerances required by autonomous navigation. Under these circumstances, additional fiducial markers can be installed on the undersides of the bridge girders. Despite its significant benefits and potential, the proposed method still faces challenges. For instance, within the current methodology, the positions of the fiducial markers are presumed to be known. However, in real-world applications, it may be challenging to accurately measure the poses of the on-site markers, and any measurement errors will eventually affect the UAV localisation accuracy. Additionally, pre-installed markers can be damaged by extreme weather, leading to a substantial workload in maintaining these markers.

In the future, the proposed method can be further enhanced by pursuing several key directions. Firstly, we aim to refine our fusion techniques by investigating adaptive weight strategies for effectively combining measurements from both ATM and SVIO modules, thus improving overall accuracy and reliability. Secondly, to bolster precision, we will address potential errors in determining fiducial marker poses, integrating comprehensive error modelling directly into the fusion module, as demonstrated in the study [80]. Thirdly, we intend to diversify our data sources, exploring the integration of Real-Time Kinematic (RTK)-GPS positioning data and barometer measurements to augment pose estimation derived from FMC-SVIL. By incorporating a broader range of data streams, our goal is to establish a more robust UAV localisation system capable of excelling across the entire bridge environment.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] S. Dorafshan, M. Maguire, Bridge inspection: human performance, unmanned aerial systems and automation, *J Civ Struct Health Monit* 8 (3) (2018) 443–476, <https://doi.org/10.1007/s13349-018-0285-4>.
- [2] X. Yang, E. del Rey Castillo, Y. Zou, L. Wotherspoon, Semantic segmentation of bridge point clouds with a synthetic data augmentation strategy and graph-structured deep metric learning, *Autom Constr* 150 (2023) 104838, <https://doi.org/10.1016/j.autcon.2023.104838>.
- [3] T. Omar, M.L. Nehdi, Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography, *Autom Constr* 83 (2017) 360–371, <https://doi.org/10.1016/j.autcon.2017.06.024>.
- [4] S. Chen, D.F. Laefer, E. Mangina, S.M.I. Zolanvari, J. Byrne, UAV bridge inspection through evaluated 3D reconstructions, *J Bridg Eng* 24 (4) (2019) 05019001, [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0001343](https://doi.org/10.1061/(ASCE)BE.1943-5592.0001343).
- [5] P.J. Chun, J. Dang, S. Hamasaki, R. Yajima, T. Kameda, H. Wada, T. Yamane, S. Izumi, K. Nagatani, Utilization of unmanned aerial vehicle, artificial intelligence, and remote measurement technology for bridge inspections, *J Rob Mechatr* 32 (6) (2020) 1244–1258, <https://doi.org/10.20965/jrm.2020.p1244>.
- [6] S.T. Nguyen, H.M. La, A climbing robot for steel bridge inspection, *J Intell Robot Syst* 102 (4) (2021) 75, <https://doi.org/10.1007/s10846-020-01266-1>.
- [7] J.-K. Oh, G. Jang, S. Oh, J.H. Lee, B.-J. Yi, Y.S. Moon, J.S. Lee, Y. Choi, Bridge inspection robot system with machine vision, *Autom Constr* 18 (7) (2009) 929–941, <https://doi.org/10.1016/j.autcon.2009.04.003>.
- [8] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, *Autom Constr* 94 (2018) 112–126, <https://doi.org/10.1016/j.autcon.2018.06.006>.
- [9] C. Zhang, Y. Zou, F. Wang, E. del Rey Castillo, J. Dimyadi, L. Chen, Towards fully automated unmanned aerial vehicle-enabled bridge inspection: where are we at? *Constr Build Mater* 347 (2022) 128543, <https://doi.org/10.1016/j.conbuildmat.2022.128543>.
- [10] F. Wang, Y. Zou, E. del Rey Castillo, Y. Ding, Z. Xu, H. Zhao, J.B.P. Lim, Automated UAV path-planning for high-quality photogrammetric 3D bridge reconstruction, *Struct Infrastruct Eng* (2022) 1–20, <https://doi.org/10.1080/15732479.2022.2152840>.
- [11] J.J. Lin, A. Ibrahim, S. Sarwade, M. Golparvar-Fard, Bridge inspection with aerial robots: Automating the entire pipeline of visual data capture, 3D mapping, defect detection, analysis, and reporting 35 (2) (2021) 04020064, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000954](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000954).
- [12] K. Sun, K. Mohta, B. Pfammatter, M. Watterson, S. Liu, Y. Mulgaonkar, C.J. Taylor, V. Kumar, Robust stereo visual inertial odometry for fast autonomous flight, *IEEE Robot Automat Lett* 3 (2) (2018) 965–972, <https://doi.org/10.1109/LRA.2018.2793349>.
- [13] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, V. Rodehorst, Framework for automated UAS-based structural condition assessment of bridges, *Autom Constr* 97 (2019) 77–95, <https://doi.org/10.1016/j.autcon.2018.10.006>.
- [14] F. Vanegas, K.J. Gaston, J. Roberts, F. Gonzalez, A framework for UAV navigation and exploration in GPS-denied environments, in: In: 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2-9 March, 2019, pp. 1–6, <https://doi.org/10.1109/AERO.2019.8741612>.

- [15] D. Reagan, A. Sabato, C. Nierzrecki, Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges, *Struct Health Monit* 17 (5) (2018) 1056–1072, <https://doi.org/10.1177/1475921717735326>.
- [16] S. Abiko, Y. Sakamoto, T. Hasegawa, S. Yuta, N. Shimaji, Development of constant altitude flight system using two dimensional laser range finder with mirrors, in: In: 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Munich, Germany, 3-7 July, 2017, pp. 833–838, <https://doi.org/10.1109/AIM.2017.8014121>.
- [17] V. Usenko, L. Von Stumberg, J. Stückler, D. Cremers, TUM flyers: Vision—Based MAV navigation for systematic inspection of structures, in: O.K. Bruno Siciliano (Ed.), Springer Tracts in Advanced Robotics, Springer International Publishing, 2020, pp. 189–209, https://doi.org/10.1007/978-3-030-34507-5_8.
- [18] S. Jung, S. Song, S. Kim, J. Park, J. Her, K. Roh, H. Myung, Toward autonomous bridge Inspection: A framework and experimental results, in: 16th International Conference on Ubiquitous Robots (UR), IEEE, Jeju, Korea (South) 24, June, 2019, p. 27, <https://doi.org/10.1109/urai.2019.8768677>.
- [19] R. Ali, D. Kang, G. Suh, Y.-J. Cha, Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures, *Autom Constr* 130 (2021), 103831, <https://doi.org/10.1016/j.autcon.2021.103831>.
- [20] S. Jiang, Y. Wu, J. Zhang, Bridge coating inspection based on two-stage automatic method and collision-tolerant unmanned aerial system, *Autom Constr* 146 (2023), 104685, <https://doi.org/10.1016/j.autcon.2022.104685>.
- [21] D. Benjumea, A. Alcántara, A. Ramos, A. Torres-Gonzalez, P. Sánchez-Cuevas, J. Capitan, G. Heredia, A. Ollero, Localization system for lightweight unmanned aerial vehicles in inspection tasks, *Sensors* 21 (17) (2021), <https://doi.org/10.3390/s21175937>.
- [22] T. Qin, S. Cao, J. Pan, P. Li, S. Shen, VINS-Fusion, Retrieved 15 January 2023 from, <https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>, 2019.
- [23] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M.M. Montiel, J.D. Tardós, ORB-SLAM3: an accurate open-source library for visual, visual–inertial, and multimap SLAM, *IEEE Trans Robot* 37 (6) (2021) 1874–1890, <https://doi.org/10.1109/TRO.2021.3075644>.
- [24] A.P. Tomiczek, J.A. Bridge, P.G. Ifju, T.J. Whitley, C.S. Tripp, A.E. Ortega, J. J. Poelstra, S.A. Gonzalez, Small unmanned aerial vehicle (sUAV) inspections in GPS denied area beneath bridges, in: Structures Congress 2018, ASCE, Fort Worth, Texas, 19–21 April, 2018, pp. 205–216, <https://doi.org/10.1061/9780784481332.019>.
- [25] T. Whitley, A. Tomiczek, C. Tripp, A. Ortega, M. Mennu, J. Bridge, P. Ifju, Design of a small unmanned aircraft system for bridge inspections, *Sensors (Switzerland)* 20 (18) (2020) 1–20, <https://doi.org/10.3390/s20185358>.
- [26] W. Nie, Z.C. Han, M. Zhou, L.B. Xie, Q. Jiang, UAV detection and identification based on WiFi signal and RF fingerprint, *IEEE Sensors J* 21 (12) (2021) 13540–13550, <https://doi.org/10.1109/JSEN.2021.3068444>.
- [27] L. Yu, Q. Fei, Q. Geng, Combining Zigbee and inertial sensors for quadrotor UAV indoor localization, in: In: 2013 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 12–14 June, 2013, pp. 1912–1916, <https://doi.org/10.1109/ICCA.2013.6565087>.
- [28] K. Guo, Z. Qiu, C. Miao, A.H. Zaini, C.-L. Chen, W. Meng, L. Xie, Ultra-wideband-based localization for quadcopter navigation, *Unmanned Syst* 04 (01) (2016) 23–34, <https://doi.org/10.1142/s2301385016400033>.
- [29] E. Petritoli, F. Leccese, M. Leccisi, Inertial navigation systems for UAV: uncertainty and error measurements, in: In: 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), , Torino, Italy, 19–21 June, 2019, pp. 1–5, <https://doi.org/10.1109/MetroAeroSpace.2019.8869618>.
- [30] D. Scaramuzza, Z. Zhang, Visual-inertial odometry of aerial robots, in: arXiv preprint, 2019, <https://doi.org/10.48550/arXiv.1906.03289> arXiv:1906.03289.
- [31] S.A. Berrabah, Y. Baudoin, GPS data correction using encoders and inertial navigation system (INS) sensors, in: Y. Baudoin, M.K. Habib (Eds.), Using Robots in Hazardous Environments, Woodhead Publishing, 2011, pp. 269–282, <https://doi.org/10.1533/9780857090201.2.269>.
- [32] A. Shetty, G. Xingxin Gao, adaptive covariance estimation of LiDAR-based positioning errors for UAVs, *NAVIGATION* 66 (2) (2019) 463–476, <https://doi.org/10.1002/navi.307>.
- [33] M. Petrlik, T. Krajník, M. Saska, LiDAR-based stabilization, navigation and localization for UAVs operating in dark indoor environments, in: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 15–18 June, 2021, pp. 243–251, <https://doi.org/10.1109/ICUASS1884.2021.9476837>.
- [34] V. Pritzl, M. Vrba, P. Štěpán, M. Saska, Cooperative navigation and guidance of a micro-scale aerial vehicle by an accompanying UAV using 3D LiDAR relative localization, in: In: 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, 21–24 June, 2022, pp. 526–535, <https://doi.org/10.1109/ICUAS54217.2022.9836116>.
- [35] Y. Alkendi, L. Seneviratne, Y. Zweiri, State of the art in vision-based localization techniques for autonomous navigation systems, *IEEE Access* 9 (2021) 76847–76874, <https://doi.org/10.1109/ACCESS.2021.3082778>.
- [36] G. Balamurugan, J. Valarmathi, V.P.S. Naidu, Survey on UAV navigation in GPS denied environments, in: In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System, Paralakhemundi, India, 3–5 October, 2016, pp. 198–204, <https://doi.org/10.1109/SCOPES.2016.7955787>.
- [37] S. Mansur, M. Habib, G.N.P. Pratama, A.I. Cahyadi, I. Ardianto, Real time monocular visual odometry using optical flow: Study on navigation of quadrotors UAV, in: In: 2017 3rd International Conference on Science and Technology - Computer (ICST), Yogyakarta, Indonesia, 11–12 July, 2017, pp. 122–126, <https://doi.org/10.1109/ICSTC.2017.8011864>.
- [38] A. El Amin, A. El-Rabbany, Monocular VO scale ambiguity resolution using an ultra low-cost spike rangefinder, *Positioning* 11 (04) (2020) 45–60, <https://doi.org/10.4236/po.2020.114004>.
- [39] M.C.P. Santos, M. Sarcinelli-Filho, R. Carelli, Indoor waypoint UAV navigation using a RGB-D system, in: In: 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Cancun, Mexico, 23–25 Nov, 2015, pp. 84–91, <https://doi.org/10.1109/RED-UAS.2015.7440994>.
- [40] I. El Bouazzaoui, S.R. Florez, A. El Ouardi, Enhancing RGB-D SLAM performances considering sensor specifications for indoor localization, *IEEE Sensors J* 22 (6) (2021) 4970–4977, <https://doi.org/10.1109/JSEN.2021.3073676>.
- [41] W. Wei, L. Tan, G. Jin, L. Lu, C. Sun, A survey of UAV visual navigation based on monocular SLAM, in: In: 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, , Chongqing, China, 14–16 December, 2018, pp. 1849–1853, <https://doi.org/10.1109/IPOEC.2018.8740355>.
- [42] M. Warren, P. Corke, B. Upcroft, Long-range stereo visual odometry for extended altitude flight of unmanned aerial vehicles, *Int J Robot Res* 35 (4) (2016) 381–403, <https://doi.org/10.1177/0278364915581194>.
- [43] Y. Liu, C. Wang, Hybrid real-time stereo visual odometry for unmanned aerial vehicles, *Opt Eng* 57 (7) (2018), 073104, <https://doi.org/10.1117/1.OE.57.7.073104>.
- [44] S. Mo, B. Yingcai, Q. Hailong, L. Jiaxin, G. Zhi, L. Feng, B.M. Chen, A brief survey of visual odometry for micro aerial vehicles, in: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October, 2016, pp. 6049–6054, <https://doi.org/10.1109/IECON.2016.7793198>.
- [45] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, P. Furgale, Keyframe-based visual–inertial odometry using nonlinear optimization, *Int J Robot Res* 34 (3) (2015) 314–334, <https://doi.org/10.1177/0278364914554813>.
- [46] Y. Fan, R. Wang, Y. Mao, Stereo visual inertial odometry with online baseline calibration, in: In: 2020 IEEE International Conference on Robotics and Automation, Paris, France, 31 May–31 August, 2020, pp. 1084–1090, <https://doi.org/10.1109/ICRA40945.2020.9197581>.
- [47] T. Qin, S. Cao, J. Pan, S. Shen, A general optimization-based framework for global pose estimation with multiple sensors, in: arXiv preprint, 2019, <https://doi.org/10.48550/arXiv.1901.03642> arXiv:1901.03642.
- [48] T. Qin, P. Li, S. Shen, VINS-Mono: A robust and versatile monocular visual–inertial state estimator, *IEEE Trans Robot* 34 (4) (2018) 1004–1020, <https://doi.org/10.1109/TRO.2018.2853729>.
- [49] I. Abaspour Kazerouni, L. Fitzgerald, G. Dooly, D. Toal, A survey of state-of-the-art on visual SLAM, *Expert Syst Appl* 205 (2022) 117734, <https://doi.org/10.1016/j.eswa.2022.117734>.
- [50] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, N. Vitzilaios, Fiducial markers for pose estimation, *J Intell Robot Syst* 101 (4) (2021), <https://doi.org/10.1007/s10846-020-01307-9>.
- [51] L. Jayatilleke, N. Zhang, Landmark-based localization for unmanned aerial vehicles, in: In: 2013 IEEE International Systems Conference (SysCon), Orlando, FL, USA, 15–18 April, 2013, pp. 448–451, <https://doi.org/10.1109/SysCon.2013.6549921>.
- [52] N. Kayhani, W. Zhao, B. McCabe, A.P. Schoellig, Tag-based visual–inertial localization of unmanned aerial vehicles in indoor construction environments using an on-manifold extended Kalman filter, *Autom Constr* 135 (2022), <https://doi.org/10.1016/j.autcon.2021.104112>.
- [53] M. Labb  , F. Michaud, RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation, *J Field Robot* 36 (2) (2019) 416–446, <https://doi.org/10.1002/rob.21831>.
- [54] M. Bloesch, M. Burri, S. Omari, M. Hutter, R. Siegwart, Iterated extended Kalman filter based visual–inertial odometry using direct photometric feedback, *Int J Robot Res* 36 (10) (2017) 1053–1072, <https://doi.org/10.1177/0278364917728574>.
- [55] K. Shoemake, Animating rotation with quaternion curves, *SIGGRAPH Comp Graph* 19 (3) (1985) 245–254, <https://doi.org/10.1145/325165.325242>.
- [56] P. Furgale, H. Sommer, J. Maye, J. Rehder, T. Schneider, L. Oth, Kalibr, Retrieved 13 June 2023 from, <https://github.com/ethz-asl/kalibr>, 2022.
- [57] S. Ramalingam, S.K. Lodha, P. Sturm, A generic structure-from-motion framework, *Comput Vis Image Underst* 103 (3) (2006) 218–228, <https://doi.org/10.1016/j.cviu.2006.06.006>.
- [58] S. Jianbo, Tomasi, Good features to track, in: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June, 1994, pp. 593–600, <https://doi.org/10.1109/CVPR.1994.323794>.
- [59] B. Lucas, D.T. Kanade, An iterative image registration technique with an application to stereo vision, in: In: 7th International Joint Conference on Artificial Intelligence, Vancouver, Canada, 24–28 August, 1981, pp. 674–679, <https://doi.org/10.5555/1623264.1623280>.
- [60] V. Lepetit, F. Moreno-Noguer, P. Fua, EPnP: an accurate O(n) solution to the PnP problem, *Int J Comput Vis* 81 (2) (2009) 155–166, <https://doi.org/10.1007/s11263-008-0152-6>.
- [61] B. Kitt, A. Geiger, H. Lategahn, Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme, in: 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, 21–24 June, 2010, pp. 486–492, <https://doi.org/10.1109/IVS.2010.5548123>.
- [62] T. Qin, J. Pan, S. Cao, S. Shen, A general optimization-based framework for local odometry estimation with multiple sensors, in: arXiv preprint, 2019, <https://doi.org/10.48550/arXiv.1901.03638> arXiv:1901.03638.
- [63] J.H. Peter, Robust estimation of a location parameter, *Ann Math Stat* 35 (1) (1964) 73–101, <https://doi.org/10.1214/aoms/117703732>.
- [64] Sameer Agarwal, Keir Mierle, T.G.S. Team, Ceres Solver, Retrieved 13 June 2023 from, <https://github.com/ceres-solver/ceres-solver>, 2022.

- [65] H. Kato, M. Billinghurst, Marker tracking and HMD calibration for a video-based augmented reality conferencing system, in: Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), San Francisco, CA, USA, 20-21 October, 1999, pp. 85–94, <https://doi.org/10.1109/IWAR.1999.803809>.
- [66] M. Fiala, ARTag, a fiducial marker system using digital techniques, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, 20–25 June 592, 2005, pp. 590–596, <https://doi.org/10.1109/CVPR.2005.74>.
- [67] E. Olson, AprilTag: A robust and flexible visual fiducial system, in: 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9-13 May, 2011, pp. 3400–3407, <https://doi.org/10.1109/ICRA.2011.5979561>.
- [68] J. Wang, E. Olson, AprilTag 2: efficient and robust fiducial detection, in: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), Daejeon, Korea (South), 9-14 October, 2016, pp. 4193–4198, <https://doi.org/10.1109/IROS.2016.7759617>.
- [69] Intel RealSense Depth Camera D455, Retrieved 13 June 2023 from, <https://www.intelrealsense.com/depth-camera-d455/>.
- [70] Trimble SX12 Scanning Total Station, Retrieved 13 June 2023 from, <https://geospatial.trimble.com/products-and-solutions/trimble-sx12>.
- [71] Gazebo - A dynamic multi-robot simulator, Retrieved 13 April 2022 from, [http://github.com/gazebosim/gazebo-classic](https://github.com/gazebosim/gazebo-classic).
- [72] PX4 Drone Autopilot, Retrieved 20 February 2022 from, <https://github.com/PX4/PX4-Autopilot>, 2022.
- [73] ROS - Robot Operating System. Retrieved 20 April 2022 from <https://www.ros.org/>.
- [74] J. Jeon, S. Jung, E. Lee, D. Choi, H. Myung, Run your visual-inertial odometry on NVIDIA Jetson: benchmark tests on a micro aerial vehicle, IEEE Robot Automat Lett 6 (3) (2021) 5332–5339, <https://doi.org/10.1109/LRA.2021.3075141>.
- [75] J.O. Araujo, J. Valente, L. Kooistra, S. Munniks, R.J.B. Peters, Experimental flight patterns evaluation for a UAV-based air pollutant sensor, Micromachines 11 (8) (2020) 768, <https://doi.org/10.3390/mi11080768>.
- [76] Alienware M15 R4 Gaming Laptop, Retrieved 13 June 2023 from, <https://www.dell.com/ae/p/alienware-m15-r4-laptop/pd>.
- [77] M. Grupp, Evo: Python Package for the Evaluation of Odometry and SLAM, Retrieved 10 April 2023 from, <https://github.com/MichaelGrupp/evo>, 2017.
- [78] Jetson Xavier NX Developer Kit, Retrieved 13 June 2023 from, <https://developer.nvidia.com/embedded/learn/get-started-jetson-xavier-nx-devkit>, 2022.
- [79] M. Gowda, J. Manweiler, A. Dhekne, R.R. Choudhury, J.D. Weisz, Tracking drone orientation with multiple GPS receivers, in: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, Association for Computing Machinery, New York City, New York, October, 2016, pp. 280–293, <https://doi.org/10.1145/2973750.2973768>.
- [80] N. Kayhani, B. McCabe, A.P. Schoellig, Stochastic modeling of tag installation error for robust on-manifold tag-based visual-inertial localization, in: R. Gupta, M. Sun, S. Brzev, et al. (Eds.), Proceedings of the Canadian Society of Civil Engineering Annual Conference 2022, Springer, Whistler, Canada, 25–28 May, 2022, pp. 41–54, https://doi.org/10.1007/978-3-031-34593-7_3.