Observability-Weighted Visual-Inertial Navigation System

Zhe-Hui Chen, Chin-Chi Hsiao, and Teng-Hu Cheng*

***Abstract*—In recent years, autonomous driving has continuously developed alongside daily technological advancements. Simultaneous Localization and Mapping (SLAM) is one of the significant techniques applied in this field. However, the adoption of autonomous driving techniques for self-driving cars or drones is not yet widespread. The main reason is that the accuracy and robustness of localization systems still need improvement to meet the requirements for autonomous driving and extended applications. This work examines the relationship between the observability and uncertainty of the estimation system and identifies the features that should be prioritized by analyzing their effect on system observability. Based on this analysis, the work aims to determine which features are not significant and can be temporarily excluded from the current estimation in a multi-sensor system. Additionally, this research weighs feature points by their individual observability, considering the influence of each observed feature point to improve estimation accuracy. The study introduces methods to consider system observability in estimation and presents simulation results. Ultimately, applying this method to multiple datasets demonstrates better estimation results compared to other methods.**

***Index Terms*—visual-inertial odometry, simultaneous localization and mapping, estimation and optimization, observability analysis**

I. Introduction

A. Motivation

WHILE a vehicle navigates in three-dimensional (3-D) space, inertial navigation systems (INS) is one of the widely utilized methods to estimate six-degrees-of-freedom (6-DoF) pose. However, because of the biases and noises that interfere the inertial measurement units (IMU) readings, simple integration of the local-linear acceleration and angular velocity measurements would bring severe drifts in a short time, especially, while the cheap IMU is used. In order to mitigate this issue, aided by additional sensors i.e., vision-aided INS is popular. The optical camera is one of the possible exteroceptive sensors, which is of energy-efficient and low cost while receiving abundant information from environment, are helpful sources for INS. Therefore, vision-aided INS (i.e., VINS) has recently prevailed when navigating in the GPS-denied environments (e.g., indoors).

System observability is important for state estimation [1]. Comprehending system observability results in a deep insight about the system geometric properties and determines state parameters needed to initialize an estimator or the minimal measurement modalities. Degenerate motions with additional unobservable directions should be prevented if possible and, in practice, can be identified through system observability. In [2], it proved that velocity, biases, and roll and pitch angles in VINS are observable. In [3], [4], it was analytically derived that the null space of observability matrix (unobservable subspace) of linearized VINS. In [5], [6], the observability analysis for the Lie-derivative-based nonlinear system was presented.

The analysis for system observability is leveraged when developing the EKF-based VINS algorithms [including visual-inertial SLAM (VI-SLAM) and visual-inertial odometry (VIO)] using heterogeneous geometric features.

The environment full of features will affect the estimation results. Specially, the geometric distribution between features and vehicle is a significant factor that consists of the observability matrix of the VIO system. Therefore, this work hopes to develop an estimator that leverages the observability of the feature points measurement, namely observability-weighted visual-inertial system (OW-VINS). The higher the observability of a feature point, the greater its influence on the estimator. In addition, in the VIO system equipped with multiple cameras, dropping out the degenerated camera measurement can help improve the overall performance.

B. Related Works

Vision-aided INS is a classical research thesis with an important body of literature [7] and has recently been reemerging with the advancement of computing and sensing technologies. In addition, the system observability about the VINS is also gradually emphasized. In this section, the briefly review for the related literature is focused on the vision-aided scenarios and the analysis for the nonlinear system.

1) Tightly-Coupled Visual-Inertial Algorithm: Scholastic works on vision-based state odometry/estimation/SLAM are daily extensive in recent years. There are plenty of noticeable approaches like DSO [8], PTAM [9], SVO [10], and ORB-SLAM [11], LSD-SLAM [12]. Any attempts to supply an integral relevant review is obviously incomplete. Nevertheless, in this section, the proposed work ignores the results about vision-only approaches, and pay attention on the most corresponding discussion about monocular visual-inertial state estimation. To deal with the inertial and visual measurements, Loosely-coupled sensor fusion [13], [14] is the simplest way. In the method, vision-only pose estimates obtained from the visual structure from motion is assisted by the independent module of IMU. The extended Kalman filter (EKF) is usually used to fuse, where IMU is used for state propagation and the vision-only pose is used for update. Furthermore, tightly-coupled visual-inertial algorithms are either based on the EKF [15]– [16] or

Zhe-Hui Chen and Teng-Hu Cheng are with the Department of Mechanical Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan. Email: zhehui8805@gmail.com, tenghu@nycu.edu.tw

Chin-Chi Hsiao is with the Mechanical and Mechatronics Systems Research Laboratories, Industrial Technology Research Institute, Taiwan. Phone: +886-3-5913896; Fax: +886-3-5913607; Email: hsiao_cc@itri.org.tw

This research is supported by the National Science and Technology Council, Taiwan (Grant Number NSTC 111-2221-E-A49-175-MY2).

Manuscript received 11 March 2025; revised 4 April 2025; accepted 15 April 2025; Date of publication 30 April 2025.

graph optimization [17], [18], [19], [20]. It jointly optimizes camera and IMU measurements from the raw measurement level. In practice, the data rate of IMUs usually is acquired much higher than the data rate of the camera. Different approaches have been provided to deal with the high rate IMU measurements. Utilizing the IMU for state propagation in EKF-based approaches [13], [15] is the most straightforward approach. In a graph optimization formulation, it proposes an efficient technique called IMU pre-integration in order to avoid repeated IMU re-integration. This technique parametrizes rotation error using Euler angles, which was first introduced in [21]. An on-manifold rotation formulation for IMU pre-integration was developed in [17]. This work derived the covariance propagation using continuous-time IMU error state dynamics. In addition, [22] introduced a closed-form solution to the monocular visual-inertial initialization problem. However, the proposed works are vulnerable to degenerated environment. Especially, the light of the environment has significant influence on the images captured by cameras. In addition, the far objects have small change in the image view while the camera moves in a short time. Those reasons cause the idea-less estimation results, which this work would like to consider and deal with to enhance the accuracy of the estimation.

2) VINS Observability Analysis: With the correspondence between the system observability and the consistent estimation, the observability analysis of VINS is significant research devoted efforts. With the concept of continuous symmetries provided in [23], Martinelli [2] identified that IMU biases, 3-D velocity, and global pitch and roll angles are observable and analytically derived the closed-form solution of VINS. System observability with degenerate motions [24], unknown inputs [25], [26], and minimum available sensors [27] are also examined. Recently, the analytic solutions with observability analysis for cooperative VIO [28] is provided by him. Based on the Lie derivatives and observability matrix rank test [29], Hesch et al. [6] analytically derived that the monocular VINS has four unobservable directions, i.e., the global position of the exteroceptive sensor and the global yaw. In [5], [30], [31], the similar studies for the observability of IMU-camera (monocular, RGBD) calibration were developed. In addition, generic motions cause the extrinsic calibration between the IMU and camera observable. More importantly, as in practice, VINS estimators are built upon the linearized system, it needs to perform the observability analysis for the linearized VINS whose observability properties can be developed when designing an estimator. For instance, the observability analysis for the linearized VINS (without considering biases) is performed in [32] and [33]. Analogously, in [3], [4],[34], they conducted observability analysis for the linearized VINS with full states (including IMU biases) and found the right null space of the observability matrix [3] by analytically deriving the system unobservable directions. Based on those analysis, the observability-constrained (OC)-VINS algorithm was developed. In addition, two degenerate motions (constant acceleration and translation) are identified in [35] which could cause more unobservable directions for monocular camera-aided INS. Those proposed works consider the relationship between the observability and the VINS estimation result, and even consider the relationship to improve the estimation results. However, those works do not thoroughly analyze the relationship between the observed environment and the system observability. In more detail, every feature points observed by cameras should have different observability, which means they have various influence on the estimation system. This work pay attention on the observability of observed feature points and consider their effect with the estimation system to enhance the accuracy of estimation results.

Problems to be resolved: in this work, the key challenge in autonomous navigation is the insufficient accuracy and robustness of localization systems, which hinders the widespread adoption of self-driving technologies in vehicles and drones. Although SLAM techniques are central to autonomous navigation, current multi-sensor estimation frameworks often suffer from observability limitations, leading to degraded performance in real-world scenarios. This research addresses the problem of how to improve localization accuracy by analyzing and prioritizing feature points based on their contribution to system observability, thereby enabling more reliable and efficient state estimation for autonomous navigation systems.

Strengths:

1) **Observability-Aware Feature Selection:** The method introduces a novel perspective by prioritizing feature points based on their contribution to the observability of the estimation system. This principled approach enhances both robustness and accuracy.
2) **Adaptive Feature Weighting:** By assigning weights to feature points based on their individual observability scores, the method dynamically emphasizes more informative observations, improving estimation precision.
3) **Efficiency through Feature Reduction:** The approach allows the temporary exclusion of less significant features, reducing computational burden without compromising estimation quality.
4) **General Applicability:** The method is validated across multiple datasets, demonstrating its effectiveness in various scenarios, suggesting strong generalization ability.
5) **Theoretically Grounded Cost Function:** The use of the trace of the observability matrix as a cost function provides a clear, mathematically sound criterion for evaluating and selecting feature points.

Limitations:

1) **Dependence on Accurate System Modeling:** The effectiveness of the observability analysis depends on an accurate system and sensor model. Modeling errors could lead to misjudged feature significance.
2) **Computational Overhead in Observability Evaluation:** Calculating the observability trace and updating feature weights may introduce computational complexity, especially in real-time or resource-constrained applications.
3) **Feature Dependence:** The performance of the method is still inherently tied to the availability and quality of visual features. Poor lighting, textureless surfaces, or occlusion could degrade performance despite the observability-based selection.

C. Contributions:

The present study has developed the data fusion from various measurement sensors. However, the accuracy of the result has no positive correlation with the number of sensors. Bad measurements may degrade the estimation performance. Thus, there is no definitely positive correlation between the accuracy of the result and the number of sensors. By considering the influence of system observability, the proposed method in this work is able to autonomously choose the better sensor to use and further reduce the affection of bad feature points in camera measurements. In spite of more process time, this method results in more accurate and reliable estimation results. The main contributions of this study are summarized as follows:

1) Choosing a better sensor measurement as inputs in the multi-camera perception system will enhance the estimation accuracy by using the observability as a weight for a tightly coupled problem.
2) With consideration of the influence of different visual feature points in the state estimator, the estimation results are more reliable and consistent.

II. Problem Formulation

A. Tightly-Coupled Visual-Inertial Algorithm

The method this work used is a sliding window-base tightly-coupled monocular VIO to enforce the accuracy of the robot state estimation.

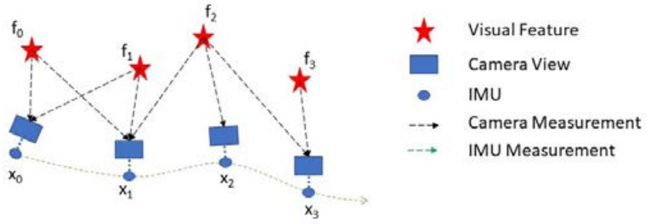


Fig. 1: Visual-Inertial Bundle Adjustment

1) Formulation: The state vector in the sliding window contains the IMU state x_n and the feature state λ_m as Figure 1:

$$\chi = [x_0, x_1, \cdots, x_n, x_c^b, \lambda_0 \cdots \lambda_m] \tag{1}$$

$$x_k = [p_{b_k}^w, v_{b_k}^w, q_{b_k}^w, b_a, b_g], k \in [0, n] \tag{2}$$

$$x_c^b = [p_c^b, q_c^b] \tag{3}$$

In the expression, x_k is the IMU state at the time when the k^{th} image is captured. $p_{b_k}^w$ and $v_{b_k}^w$ denote the IMU position and velocity in the world frame. $q_{b_k}^w$ is the unit quaternion which represents the rotation from the IMU frame to the world frame. b_a and b_g represent the accelerometer and gyroscope biases, respectively. p_c^b and $q_c^b$are the transformation from the camera frame to the IMU frame. λ_m is the inverse depth of the m^{th} feature from its first observation. n is the total number of key frames, and m is the total number of features in the sliding window.

The sum of prior and the Mahalanobis norm of all measurement residuals are minimized to obtain a maximum posterior estimation:

$$\begin{gathered} \min_x \left\{ \left\| r_p - H_p x \right\|^2 + \sum_{k \in B} \left\| r_B \left(\hat{z}_{b_{k+1}}^{b_k}, x \right) \right\|^2 \right. \\ \left. + \sum_{(l,j) \in C} \rho \left(\left\| r_C \left(\hat{z}_l^{c_j}, x \right) \right\|^2 \right) \right\} \end{gathered} \tag{4}$$

where the Huber norm is defined as:

$$\rho(s) = \begin{Bmatrix} 1 & s \geq 1 \\ 2\sqrt{s} - 1 & s < 1 \end{Bmatrix} \tag{5}$$

$r_B\left(\hat{z}_{b_{k+1}}^{b_k}, x\right)$ and $r_C\left(\hat{z}_l^{c_j}, x\right)$ are the residuals for IMU and visual measurements respectively. The detailed definition of those residuals will be represented as (14) in section II-A2 and (15) in Section II-A3. B is the set of all IMU measurements and C is the set of features that have been observed at least twice in the current sliding window. $r_p$and H_p are the prior information from marginalization with the detailed definition is described in [36].

2) *IMU Measurement Residual:* The two continue-time states corresponding to position, velocity, orientation are able to be propagated by IMU information during the continuous time interval.

$$\begin{gathered} p_{b_{k+1}}^w = p_{b_k}^w + v_{b_k}^w \delta t + \frac{1}{2}\left(R_{b_k}^w (\hat{a}_k - b_{a_k}) - g^w\right)\delta t^2 \\ v_{b_{k+1}}^w = v_{b_k}^w + \left(R_{b_k}^w (\hat{a}_k - b_{a_k}) - g^w\right)\delta t \\ q_{b_{k+1}}^w = q_{b_k}^w \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\widehat{\omega}_k - b_{w_k})\delta t \end{bmatrix} \end{gathered} \tag{6}$$

Because the propagation of IMU state contains variables in the states, so this work adopts pre-integration algorithm to avoid re-propagation problem as following equation. Through transforming the world frame to the local frame, preintegrating the parts only corresponding to IMU measurements $\hat{a}_k$ and $\widehat{\omega}_k$ as 8.

$$\begin{gathered} R_w^{b_k} p_{b_{k+1}}^w = R_w^{b_k}\left(p_{b_k}^w + v_{b_k}^w \delta t - \frac{1}{2} g^w \delta t^2\right) + \alpha_{b_{k+1}}^{b_k} \\ R_w^{b_k} v_{b_{k+1}}^w = R_w^{b_k}\left(v_{b_k}^w - g^w \delta t\right) + \beta_{b_{k+1}}^{b_k} \\ q_w^{b_k} \otimes q_{b_{k+1}}^w = \gamma_{b_{k+1}}^{b_k} \end{gathered} \tag{7}$$

where the three pre-integration terms are:

$$\begin{gathered} \alpha_{b_{k+1}}^{b_k} = \iint_{t \in [t_k, t_{k+1}]} R_t^{b_k} (\hat{a}_t - b_{a_t} - n_a) dt^2 \\ \beta_{b_{k+1}}^{b_k} = \int_{t \in [t_k, t_{k+1}]} R_t^{b_k} (\hat{a}_t - b_{a_t} - n_a) dt \\ \gamma_{b_{k+1}}^{b_k} = \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \Omega(\widehat{\omega}_t - b_{w_t} - n_\omega) \gamma_t^{b_k} dt \end{gathered} \tag{8}$$

Since the measurement noises n_a and n_ω are unknown, the pre-integration terms are formulated as following.

$$\begin{bmatrix}\hat{\alpha}_{i+1}^{b_k}\\ \hat{\beta}_{i+1}^{b_k}\\ \hat{\gamma}_{i+1}^{b_k}\end{bmatrix}=\begin{bmatrix}\hat{\alpha}_{i}^{b_k}+\hat{\beta}_{i}^{b_k}\delta t+\frac{1}{2}R\left(\hat{\gamma}_{i}^{b_k}\right)\left(\hat{a}_i-b_{a_i}\right)\delta t^2\\ \hat{\beta}_{i}^{b_k}+R\left(\hat{\gamma}_{i}^{b_k}\right)\left(\hat{a}_i-b_{a_i}\right)\delta t\\ \hat{\gamma}_{i}^{b_k}\otimes\begin{bmatrix}1\\ \frac{1}{2}\left(\widehat{\omega}_i-b_{wi}\right)\delta t\end{bmatrix}\end{bmatrix} \quad (9)$$

where i is the discrete moment corresponding to a IMU measurement within$[t_k,t_{k+1}]$. δt is the time interval between the two IMU measurements i and $i+1$. $\hat{a}_i$ and $\widehat{\omega}_i$ are the acceleration and the angular velocity of IMU at time i. Therefore, a continuous-time linearized dynamics of error terms can be derived:

The covariance matrix $P_{bk+1}^{bk}$and the Jacobian $J_{b_{k+1}}$ can be computed as following:

$$P_{t+\delta t}^{b_k}=(I+F_t\delta t)P_t^{b_k}(I+F_t\delta t)^T+(G_t\delta t)Q(G\delta t)^T \quad (11)$$

$$J_{t+\delta t}=(I+F_t\delta t)J_t,t\in[k,k+1] \quad (12)$$

where $P_t^{b_k}$ and Q are the covariance of $\delta z_t^{b\kappa}$ and noise n_t.

Eventually, $\widehat{\alpha}_{b_{k+1}}^{b_k}$, $\widehat{\beta}_{b_{k+1}}^{b_k}$, $\widehat{\gamma}_{b_{k+1}}^{b_k}$ can be updated with δb_a and δb_ω by using $J_{t+\delta t}$.

$$\begin{bmatrix}\tilde{\alpha}_{b_{k+1}}^{b_k}\\ \tilde{\beta}_{b_{k+1}}^{b_k}\\ \tilde{\gamma}_{b_{k+1}}^{b_k}\end{bmatrix}=\begin{bmatrix}\hat{\alpha}_{b_{k+1}}^{b_k}+J_{b_a}^{\alpha}\delta b_{ak}+J_{b_\omega}^{\alpha}\delta b_{\omega k}\\ \hat{\beta}_{b_{k+1}}^{b_k}+J_{b_a}^{\beta}\delta b_{ak}+J_{b_\omega}^{\beta}\delta b_{\omega k}\\ \hat{\gamma}_{b_{k+1}}^{b_k}\otimes\begin{bmatrix}1\\ \frac{1}{2}J_{b_\omega}^{\gamma}\delta b_{\omega k}\end{bmatrix}\end{bmatrix} \quad (13)$$

where $J_{b_a}^{\alpha}$, $J_{b_\omega}^{\alpha}$, $J_{b_a}^{\beta}$, $J_{b_\omega}^{\beta}$, and $J_{b_\omega}^{\gamma}$ are the subblock of J_{bk+1}. The details can be found in [37].

The IMU residual is defined as

$$r_B\left(\tilde{z}_{b_{k+1}}^{b_k},x\right)=$$

$$\begin{bmatrix}R_w^{b_k}\left(p_{b_{k+1}}^w-p_{b_k}^w+\frac{1}{2}g^w\Delta t_k^2-v_{b_k}^w\Delta t_k\right)-\tilde{\alpha}_{b_{k+1}}^{b_k}\\ R_w^{b_k}\left(v_{b_{k+1}}^w+g^w\Delta t_k-v_{b_k}^w\right)-\tilde{\beta}_{b_{k+1}}^{b_k}\\ 2\left[\left(q_w^{b_k}\right)^{-1}\otimes q_{b_{k+1}}^w\otimes\left(\tilde{\gamma}_{b_{k+1}}^{b_k}\right)^{-1}\right]_{xyz}\\ b_{ab_{k+1}}-b_{ab_k}\\ b_{\omega b_{k+1}}-b_{\omega b_k}\end{bmatrix} \quad (14)$$

where $[\cdot]_{xyz}$ denotes extracting the vector part of a quaternion. $\tilde{\alpha}_{b_{k+1}}^{b_k}$, $\tilde{\beta}_{b_{k+1}}^{b_k}$, and $\tilde{\gamma}_{b_{k+1}}^{b_k}$ denote the IMU terms only preintegrating the noisy acceleration and gyroscope measurements within the time interval between two continuous camera measurements.

3) *Visual Measurement Residual:* The landmarks are detected and tracked as image features. In addition, this work defines the vision residual that considers the l^{th} feature in the j^{th} image as the reprojaction error in Figure 2.

$$r_c\left(\hat{z}_l^{c_j},\chi\right)=\pi_c^{-1}\left(\begin{bmatrix}\hat{u}_l^{c_j}\\ \hat{v}_l^{c_j}\end{bmatrix}\right)-\frac{P_l^{c_j}}{\left\|P_l^{c_j}\right\|} \quad (15)$$

$$P_l^{c_j}=R_b^c\left(R_w^{b_j}\left(R_{b_i}^w\left(R_c^b\frac{1}{\lambda_l}\pi_c^{-1}\left(\begin{bmatrix}u_l^{c_i}\\ v_l^{c_i}\end{bmatrix}\right)+p_c^b\right)+p_{b_i}^w-p_{b_j}^w\right)-p_c^b\right) \quad (16)$$

Where $\left[\hat{u}_l^{c_j} \quad \hat{v}_l^{c_j}\right]$ is the observation of the l^{th} feature in the j^{th} image, and $\left[u_l^{c_i} \quad v_l^{c_i}\right]$ is the same feature observed in thej^{th} image. π_c^{-1} is a back projection function which projects a pixel location into a unit vector. This residual can be viewed as a re-projection error in the field of computer vision as Figure 1.

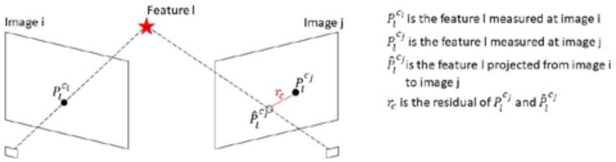

Feature l
Image i
Image j
$P_l^{c_i}$
$P_l^{c_j}$
r_c
$\hat{P}_l^{c_j}$
$P_l^{c_i}$ is the feature l measured at image i
$P_l^{c_j}$ is the feature l measured at image j
$\hat{P}_l^{c_j}$ is the feature l projected from image i to image j
r_c is the residual of $P_l^{c_j}$ and $\hat{P}_l^{c_j}$

Fig. 2: Reprojection Error of The Feature Observation

B. Observability Analysis for VIO

In this section, to simplify the process of analyzing the observability properties of the VIO system, this work considers the situation that only one single point feature is observed by the camera measurement. Specifically, this work first study and find out that there are four unobservable directions of the ideal VIO system. Instead of directly using the above states, this work examines the observability of the system with a new definition of the state vector to more easily analyze the property.

1) *System State:* The state vector for observability analysis is defined as

$$x=\left[q_w^{b_k},b_g,v_{b_k}^w,b_a,p_{b_k}^w,f_m^w\right] \quad (17)$$

Where $p_{b_k}^w$,$v_{b_k}^w$, $q_w^{b_k}$, b_a , b_g are mentioned in the Section II-A1. f_m^w is the 3-D position of the m^{th} feature in the world frame, corresponding the m^{th} feature in Section II-A1.

2) *System Dynamic Model:* The dynamics model of the VIO system is described as follows.

$$\begin{aligned}\dot{q}_w^{b_k}&=\frac{1}{2}\Omega(\omega)q_w^{b_k}\\ \dot{b_g}&=n_{b\omega}\\ \dot{v}_{b_k}^w&=a\\ \dot{b_a}&=n_{ba}\\ \dot{p}_{b_k}^w&=v_{b_k}^w\\ \dot{f}_m^w&=0_{3\times1}\end{aligned} \quad (18)$$

where a and ω are the acceleration and rotation velocity of the IMU, respectively. In addition,

$$\Omega(\omega)=\begin{bmatrix}-[\omega\times] & \omega\\ -\omega^T & 0\end{bmatrix},[\omega\times]=\begin{bmatrix}0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix}$$

From the above definition, the nominal states can be derived as follows.

$$\begin{bmatrix} \delta\dot{\alpha}_t^{b_\kappa} \\ \delta\dot{\beta}_t^{b_\kappa} \\ \delta\dot{\theta}_t^{b_\kappa} \\ \delta\dot{b}_{at} \\ \delta\dot{b}_{\omega t} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & -R_t^{b_k}[\hat{a}_t - b_{at}]_\times & -R_t^{b_k} & 0 \\ 0 & 0 & -[\widehat{\omega}_t - b_{\omega t}]_\times & 0 & -I \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\alpha_t^{b_\kappa} \\ \delta\beta_t^{b_\kappa} \\ \delta\theta_t^{b_\kappa} \\ \delta b_{at} \\ \delta b_{\omega t} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ -R_t^{b_k} & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} n_a \\ n_\omega \\ n_{ba} \\ n_{b\omega} \end{bmatrix}$$
$$= F_t \delta z_t^{b_k} + G_t n_t \tag{10}$$

$$\begin{aligned} \dot{q}_w^{b_k} &= \frac{1}{2}\Omega(\widehat{\omega})\hat{q}_w^{b_k} \\ \dot{b_g} &= 0_{3\times1} \\ \dot{v}_{b_k}^w &= C^T\left(\hat{q}_w^{b_k}\right)\hat{a} + g^w \\ \dot{b_a} &= 0_{3\times1} \\ \dot{p}_{b_k}^w &= \hat{v}_{b_k}^w \\ \dot{f}_m^w &= 0_{3\times1} \end{aligned} \tag{19}$$

where $\hat{a} = a_m - \hat{b}_a$, and $\widehat{\omega} = \omega_m - \hat{b}_{g.} C\left(\hat{q}_w^{b_k}\right)$ is the rotation matrix corresponding to $\hat{q}_w^{b_k}$.

Therefore, the error state between the true state and the nominal state, $\tilde{x} = x - \hat{x}$, can be obtained as

$$\dot{\tilde{x}} = \begin{bmatrix} F & 0_{15\times3m} \\ 0_{3m\times15} & 0_{3m} \end{bmatrix} \tilde{x} + \begin{bmatrix} G \\ 0_{3m\times12} \end{bmatrix} n \tag{20}$$

where n is the noise $n = [(n_\omega)^T(n_{b\omega})^T(n_a)^T(n_{ba})^T]^T$. F is the errorstate transition matrix, and G is the input noise matrix defined as.

$$F = \begin{bmatrix} -[\widehat{\omega}\times] & -I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ -C^T\left(\hat{q}_w^{b_k}\right)[\hat{a}\times] & 0_3 & 0_3 & -C^T\left(\hat{q}_\omega^{b_k}\right) & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & I_3 & 0_3 & 0_3 \end{bmatrix} \tag{21}$$

$$G = \begin{bmatrix} -I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & -C^T\left(\hat{q}_\omega^{b_k}\right) & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix} \tag{22}$$

3) Discrete-Time Implementation: Because the IMU measurements are received at a sample time, they are sampled at a constant rate $1/\delta_t$, where $\delta_t = t_{k+1} - t_k$. The state estimation is propagated using numerical integration, so the discrete-time state transition matrix Φ_k^{k+1} from time-step t_k to t_{k+1} is desired.

$$\begin{aligned} \dot{\Phi}_k^{k+1} &= F\Phi_k^{k+1} \\ \Phi_0 &= I_{18} \end{aligned} \tag{23}$$

The discrete-time system noise covariance matrix Q_k and propagation covariance matrix $P_{\mathrm{k+1|k}}$ are also derived as

$$Q_k = \int_{t_k}^{t_{k+1}} \Phi_\tau^{k+1} G Q_c G^T (\Phi_\tau^{k+1})^T d\tau \tag{24}$$

$$P_{\kappa+1|k} = \Phi_k^{k+1} P_{k|k} \left(\Phi_k^{k+1}\right)^T + Q_k \tag{25}$$

where Q_c depends on the IMU noise characteristics.

4) Measurement Model ∶ During motion, the camera observes plenty of visual features, and the motion and feature position are estimated by utilizing the camera measurements. For simplification, this work considers the camera residual with only one feature point f_i. The camera measurement z_i is the projection of the 3-D point f_i^b, expressed in the IMU frame as

$$z_i = \frac{1}{z}\begin{bmatrix} x \\ y \end{bmatrix} + \eta_i \tag{26}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = f_i^b = C(q_w^b)(f_i^w - p_b^w) \tag{27}$$

where η_i is the measurement noise. Therefore, the error state of the measurement model is:

$$\begin{aligned} \tilde{z}_i &= z_i - \hat{z}_i \\ &= H_i \tilde{x} + \eta_i \end{aligned} \tag{28}$$

where $\hat{x}$ is the nominal measurement computed by (26)-(27), and the measurement Jacobian H_i is defined as

$$H_i = H_c\left[H_q \quad 0_{3\times9} \quad H_p| \quad 0_3 \cdots \quad H_{f_i} \cdots \quad 0_3\right], \tag{29}$$

where

$$H_c = \frac{1}{z^2}\begin{bmatrix} z & 0 & -x \\ 0 & z & -y \end{bmatrix}$$

$$H_q = [C(q_w^b)(f_i^w - q_b^w) \times]$$

$$H_p = -C(q_w^b)$$

$$H_{f_i} = C(q_w^b)$$

While a new feature is observed by the camera, this work initializes it into the state vector. Since one feature observation does not provide enough information to resolve the 3-D position of the feature point, this work utilize multiple observations to recover the features. In order to compute the initial feature position estimation, uncertainty, and cross-correlation with the current state, this work use a bundle-adjustment method to solve it over a sliding window.

5) Observability Analysis: To simply understand the observability property of the VIO system, this work considers the case when only one feature point is observed. If the observability matrix $M(x)$ is full column rank, this work can say that the VIO system would be observable, vice versa.

The observation matrix M is defined as [38]:

$$M(x) = \begin{bmatrix} H_1 \\ H_2\Phi_1^2 \\ \vdots \\ H_k\Phi_1^k \end{bmatrix} \tag{30}$$

where Φ_1^k is the state transition matrix from time-step 1 to time-step k, and H_k is the linearized measurement model at time-step k.

Therefore, the nullspace N of the observability matrix can be derived, which is the span of the unobservability direction for the VIO system.

$$N_1 = \begin{bmatrix} 0_3 & C(q_w^{b_1})g^w \\ 0_3 & 0_{3\times1} \\ 0_3 & -[v_{b_1}^w \times]g^w \\ 0_3 & 0_{3\times1} \\ I_3 & -[p_{b_1}^w \times]g^w \\ I_3 & -[f^w \times]g^w \end{bmatrix} = [\,N_{t,1}|\ N_{r,1}] \tag{31}$$

The detailed derivation and proof can refer to [3]. The 18 × 3 block column $N_{t,1}$ corresponding to the global translation motion, i.e., amount of vehicle translation is same with the feature point translation. The 18×1 column $N_{r,1}$ corresponding to the global rotation motion of vehicle and the feature about the gravity vector. In conclusion, the VIO system has four unobservable direction corresponding 3-D global translation and 1-D global rotation about the gravity vector.

III. Estimation Based on Visual Observability

To enforce the robustness and consistence of the VIO result, his work take observability into account during the estimation process in (46). The feature points with higher observability values will improve the estimation result. In this paper, this work visualizes the observability of every feature point in the image and redesign the estimation formula based on the observability.

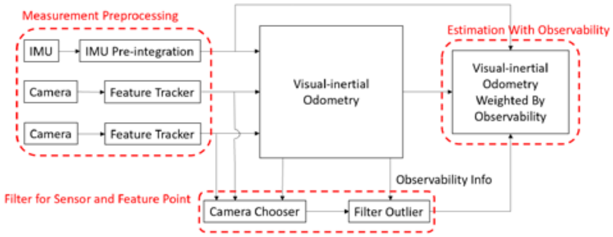

Measurement Preprocessing
IMU
IMU Pre-integration
Camera
Feature Tracker
Camera
Feature Tracker
Visual-inertial Odometry
Estimation With Observability
Visual-inertial Odometry Weighted By Observability
Observability Info
Filter for Sensor and Feature Point
Camera Chooser
Filter Outlier

Fig. 3: A block diagram illustrating the pipeline of the visual-inertial odometry weighted by observability

A. Overview

The architecture of the visual-inertial state estimator weighted by the observability is shown in Figure 3. First, in the measurement preprocessing stage, the system receives the information from the IMU and the two cameras, and it pre-integrates the IMU measurement between two consecutive frames and extracts the individual feature points to track. In order to get the observability of each feature point, the three-dimension pose of the feature point is required. Secondly, this work temporally uses the state estimation proposed from [37] to get the 3D pose of all feature points. Therefore, this work can calculate the observability for feature points with the 3D pose. With the observability information, the camera chooser determines which camera is better as the visual input based on their image observability. In addition, this work finds that the more number the features in the image with low observability, the more possibility the estimation is wrong. Thirdly, during the outlier filtering process, the feature point which has an observability value of three times larger than the standard deviation is not considered in the estimator. Since, the observability value represents how important the feature point is for the estimator. Finally, the estimator weighted by observability (46) is modified from (4), which weighs every camera residual by their individual feature observability value. Theoretically, the more feature points with high observability, the more accurate the estimation result is.

B. Relation between Estimation Uncertainty and System Observability

To improve the accuracy of estimation problem, the system observability [39] is considered.

The time derivative of system output y up to order $n-1$:

$$Y = \begin{bmatrix} y \\ y^{(1)} \\ \vdots \\ y^{(n-1)} \end{bmatrix} = g\left(x, u, \dots, u^{(n-r-1)}\right) \tag{32}$$

where x represents the system states, u represents the system input. The superscript denotes the order of the time derivative. n and r are the degree of states and the relative degree of system, respectively.

This work gets a linear approximation of the system output by computing the first-order Taylor series expansion.

$$Y \approx g(x_0, u_0) + dY(x_0)\Delta x \tag{33}$$

where x_0 and u_0 are the linearization point. $dY\ (x_0)$ is the derivative of Y with respect to the state at point x_0 . Δx represents the deviation of states from the linearization point.

Through the least-square estimator [40], this work can get an approximate solution for Δx , if the measurements Y are interfered by the measurement noise with covariance R.

$$\Delta x = (dY^T R^{-1} dY)^{-1} dY^T R^{-1} (Y - g) \tag{34}$$

and state covariance [40]

$$P = (dY^T R^{-1} dY)^{-1} \tag{35}$$

Therefore, citing Cramer-Rao lower bound from [41], the state covariance P is inversely proportional to the value of $dY^T R^{-1} dY$. By the way, the measurement Jacobian dY is equivalent to the observability matrix M in (30). Thus, in order to improve the performance of estimation, the trace of the observability matrix is considered as a metric.

C. Estimation with Observability

In order to quantify the observability matrix, a scalar value of the observability matrix needs to be evaluated. The quantitative method used is to take the trace value of the observability matrix, which means the inverse of average estimation uncertainty [39].

$$cost = Tr(M^T M) \tag{36}$$

Specially, there are several features received by camera, which means the row number of the measurement model H in is larger than the number of the states, so the observability matrix is

$$M(x) = [H]$$
$$H = \frac{\partial r_c\left(\hat{z}_l^{c_j}, \chi\right)}{\partial \chi} \tag{37}$$

From the definition of tightly-couple VIO system (15) defined in Section II-A, the linearized measurement model is

$$H = \begin{bmatrix} H_1 \\ \vdots \\ H_l \end{bmatrix} \tag{38}$$

$$H_l = \begin{bmatrix} \frac{1}{z_l^{c_j}} & 0 & \frac{-x_l^{c_j}}{\left(z_l^{c_j}\right)^2} \\ 0 & \frac{1}{z_l^{c_j}} & \frac{-y_l^{c_j}}{\left(z_l^{c_j}\right)^2} \end{bmatrix} \begin{bmatrix} \left(-R_b^c R_w^{b_j}\right)^T \\ H_{l\_bottom} \end{bmatrix} \tag{39}$$

$$H_{l\_bottom} = \left(R_b^c \left[R_w^{b_j} \left(R_{b_i}^w Pt^b + p_{b_i}^w - p_{b_j}^w\right) - p_c^b\right] \times\right)^T \tag{40}$$

$$Pt^b = R_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} u_l^{c_i} \\ v_l^{c_i} \end{bmatrix}\right) + p_c^b \tag{41}$$

where l is the subscript of the feature point in the sliding window. $x_l^{c_j}$, and $y_l^{c_j}$, and $z_l^{c_j} arethepositionof\ l^{th}$ feature point in the world frame.

Subsequently, the cost function can be rewritten as

$$\begin{aligned} cost &= Tr(H^T H) \\ &= Tr\left[\left(-R_b^c R_w^{b_j}\right)^T A \left(-R_b^c R_w^{b_j}\right)\right] + Tr[U^T A U] \end{aligned} \tag{42}$$

$$A = \begin{bmatrix} \frac{1}{\left(z_l^{c_j}\right)^2} & 0 & \frac{-x_l^{c_j}}{\left(z_l^{c_j}\right)^3} \\ 0 & \frac{1}{\left(z_l^{c_j}\right)^2} & \frac{-y_l^{c_j}}{\left(z_l^{c_j}\right)^3} \\ \frac{-x_l^{c_j}}{\left(z_l^{c_j}\right)^3} & \frac{-y_l^{c_j}}{\left(z_l^{c_j}\right)^3} & \frac{x_l^{c_j}}{\left(z_l^{c_j}\right)^3} + \frac{y_l^{c_j}}{\left(z_l^{c_j}\right)^3} \end{bmatrix} \tag{43}$$

$$U = R_b^c \left[R_w^{b_j} \left(R_{b_i}^w \left(R_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} u_l^{c_i} \\ v_l^{c_i} \end{bmatrix}\right) + p_c^b\right) + p_{b_i}^w - p_{b_j}^w\right) - p_c^b \times\right] \tag{44}$$

Thus, the optimal control input for the position $p_{b_k}^w$ and quaternion $q_{b_k}^{\omega}$ of the system depends on the Jacobian of cost function.

1) Normalization of the Cost Function: When the vehicle moves in the environment, the number of observed feature points is various. The column number of (38) is same as the number of feature points. Thus, the normalization of the cost function is an unavoidable problem to deal with. Through the mathematical induction, the normalized cost function is derived as

$$cost = \frac{Tr(H^T H)}{m} \tag{45}$$

where m is the number of feature points.

2) Camera Chooser: Unmanned vehicle moving in a degenerated environment will result in degraded estimation. Typically, the camera measurement has low system observability in those degenerated environments. In other words, the higher the system observability is, the more accurate the estimated result is. Hence, in the two-camera case, this work choose which camera should be used in the current estimation and the other camera will temporary drop out, according to their individual observability from (45). In the experiments, the camera measurements with higher observability values will be reserved, and those with lower observability values are not included in the system.

After getting the observability of every feature point (39), this work found that the estimation result is severely affected by the outlier feature points. In order to enforce the consistence of the estimation, filtering out the outlier feature points is a significant pre-process before estimation. If the observability value of the feature point is three times larger than the standard deviation, the feature point will be filtered out from the estimation process. Therefore, the estimation result with the pre-processing will be more robust and consistent.

3) Estimation With Observability: The observability value represents the influence of the measurement with the estimated states. As mentioned earlier, considering the observability value in the estimation process will improve the accuracy. Focusing on more observable measurement will increase the estimation reliability. Therefore, the observability of each feature point is considered in the estimation process as (46).

$$\min_{x}\left\{\left\|r_p - H_p x\right\|^2 + \sum_{k\in B}\left\|r_B\left(\hat{z}_{bk+1}^{bk}, x\right)\right\|^2 + \sum_{(l,j)\in C}\rho\left(\left\|\frac{M_l}{M_{total}} r_C\left(\hat{z}_l^{c_j}, x\right)\right\|^2\right)\right\} \tag{46}$$

$$M_l = Tr(H_l^T H_l)$$

$$M_{total} = \sum_{l\in C} M_l$$

where H_l has be early defined in (39), M_l is the trace of the observability of the l-th feature pointy and M_{total} is the total trace of the observability of feature points in set C.

Remark 1: The cost function is designed to maximize the trace of the observability matrix. Based on equation (36), this cost can serve as a metric to evaluate the quality of individual feature points. That is, the higher the observability trace, the more valuable the feature point is for state estimation. Given the selected feature points, we further improve localization accuracy in equation (46) by assigning higher weights to the more informative features, resulting in more accurate state estimation.

IV. Simulation

To verify the estimation performance by considering the feature observability, two datasets are used to evaluate the estimation result. The first dataset, NTU VIRAL, was collected by an unmanned aerial vehicle equipping two cameras and the IMU. The main purpose of using the dataset is to confirm if the performance of OW-VINS is better than that of VINS-Mono. The second dataset, Hilti SLAM, is recorded from a handheld platform mounting cameras facing around and an IMU. The aim is to evaluate how OW-VINS improves the estimation by using the cameras facing the opposite directions.

A. Individual Observability Value

In order to evaluate the observability with the distribution of feature point, Figure 4 illustrates the individual observability of every feature point. The green circles are the observed feature points in the image, and the red numbers are the observability values of the feature points. There are two regular patterns can be observed in Figure 4. Firstly, the closer the feature point is to the camera, the larger its observability value is. Secondly, the closer the feature point is to the corners of the image, the bigger its observability value is. In fact, the laws are reasonable since the feature point close to the camera or close to the edge of the image has obvious displacement in the image during motion. Videos can be found in the multimedia attachment[1].

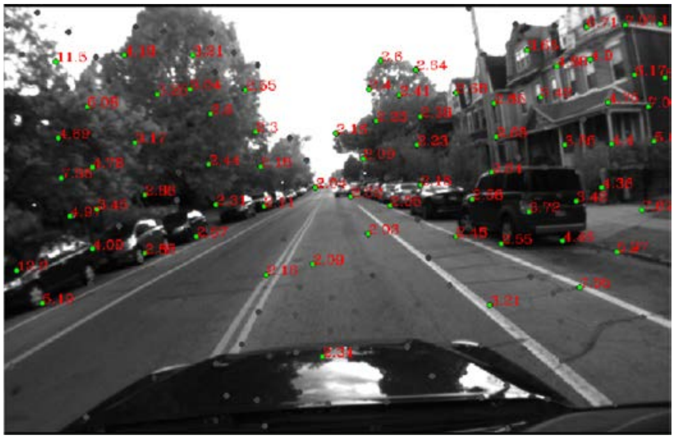


Fig. 4: Individual observability of every feature point

B. NTU VIRAL Dataset

NTU VIRAL dataset [42] is a visual-inertial-ranging-lidar dataset for autonomous aerial vehicle. It is equipped with two times-synchronized cameras with 10 fps and multiple inertial measurement units with 385 Hz. In addition, both cameras are facing the same direction as shown in Figure 5. The comprehensive sensor suite resembles that of an autonomous driving car but features distinct and challenging characteristics of aerial operations. They conducted the flight tests in a variety of indoor and outdoor conditions. The basic information for data collection is described in Table I. In eee sequences, the area is surrounded by tall building structures where visual features can be detected on nearby objects such as buildings, road markings, and trees, which collected at the School of EEE central carpark. In nya sequences, low lighting conditions are difficult for visual SLAM to perform, which collected inside the Nanyang Auditorium. Figure 7 illustrates the image captured in eee_01 and nya_01, which demonstrates the image in nya_01 is obviously darker than that of eee_01. In sbs sequences, some low-rise buildings with large glass surfaces surround this area, where visual features may only be detected on far way objects, which can include noisy depth. Besides, it is collected at the School of Bio, Science's front square.

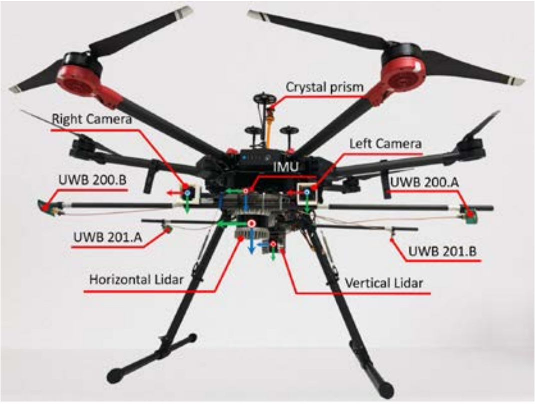
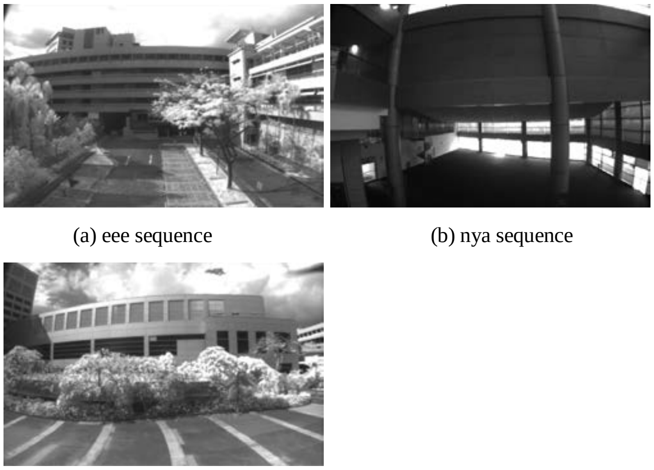


Fig. 5: Hardware Structure of NTU VIRAL Dataset

[1] https://www.youtube.com/watch?v=42jcwil4aCg

TABLE I: The Information of NTU VIRAL Dataset

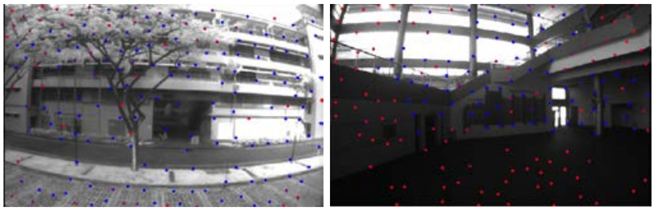
Sequence	Path Length (m)	Duration (s)	Remark
eee_01 (Figure 8a)	237	398.7	Collected at the School of EEE central carpark
eee_02 (Figure 8b)	171	321.1	Collected at the School of EEE central carpark
nya_01 (Figure 9a)	160	396.3	Collected inside the Nanyang Auditorium
nya_02 (Figure 9b)	249	428.7	Collected inside the Nanyang Auditorium
sbs_01 (Figure 10a)	202	354.2	Collected at the School of Bio. Science's front square
sbs_02 (Figure 10b)	184	373.3	Collected at the School of Bio. Science's front square



(a) eee sequence (b) nya sequence

(c) sbs sequence

Fig. 6: Environment where the NTU VIRAL datasets are Collected



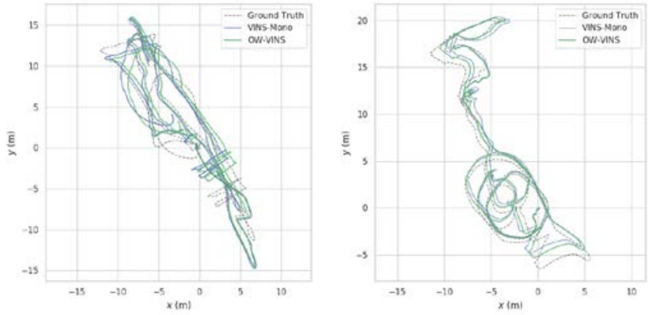
(a) Feature Points in eee_01 Dataset (b) Feature Points in nya_01 Dataset

Fig. 7: Feature Points in Different Datasets

In the datasets, this work utilizes them to check whether switching cameras as input based on its observability is better than consistently using the same camera as input. VINS-MONO always use the left camera as its image input. Instead, OW-VINS firstly determines which camera should be used by computing their observability value, and secondly estimates the states by considering every feature point's observability in the image. In Table II, it is clear that OW-VINS is almost much better than VINS-Mono at least 10 percentages. In eee sequences, OW-VINS improves the estimation accuracy about 10 percentages, which means it is suitable for these outside environments. In nya sequences, there is no significant difference between both methods. The reason is that the inside environment has few visual features since it lacks in enough light to identify surrounding object. In sbs sequences, although the environment is full of far visual features that cause noisy depth, OW-VINS is able to filter those noisy feature points to improve the estimation accuracy by at least 35 percentage. Unfortunately, there is no obvious improvement in RPE in Table III. The estimated trajectories by VINS-Mono and OW-VINS are illustrated from Figure 8a to Figure 10b. The estimated trajectory using OW-VINS fits the ground truth better than the that of VINS-Mono.

TABLE II: Position APE (m) of VINS-MONO and WO-VINS

Sequence	VINS-MONO	OW-VINS	Progress (%)
eee_01 (Figure 8a)	2.912062	2.502542	14
eee_02 (Figure 8b)	1.557913	0.856959	45
nya_01 (Figure 9a)	1.160654	1.083436	6
nya_02 (Figure 9b)	1.436416	1.499788	-4
sbs_01 (Figure 10a)	6.955026	2.559598	63
sbs_02 (Figure 10b)	3.020185	1.859777	38

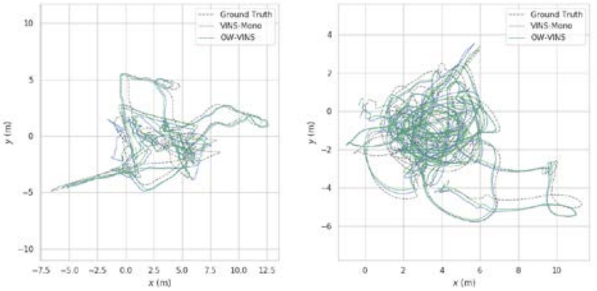


(a) eee_01 (b) eee_02

Fig. 8: Trajectories of Sequence eee Generated from VINS-Mono and OW-VINS

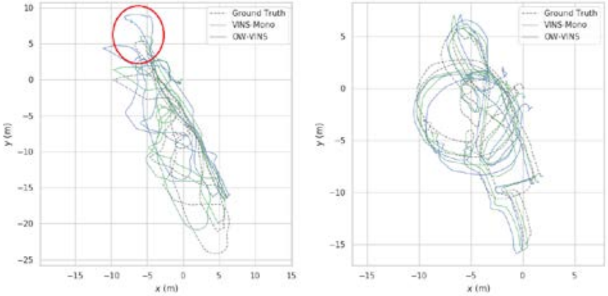
TABLE III: Position RPE (m) of VINS-MONO and WO-VINS

Sequence	VINS-MONO	OW-VINS	Progress (%)
eee_01 (Figure 8a)	0.113202	0.113270	-0.06
eee_02 (Figure 8b)	0.094816	0.094236	0.6
nya_01 (Figure 9a)	0.111705	0.096028	14
nya_02 (Figure 9b)	0.111275	0.111755	-0.4
sbs_01 (Figure 10a)	0.113973	0.114006	-0.02
sbs_02 (Figure 10b)	0.117595	0.116775	0.6



(a) nya_01 (b) nya_02

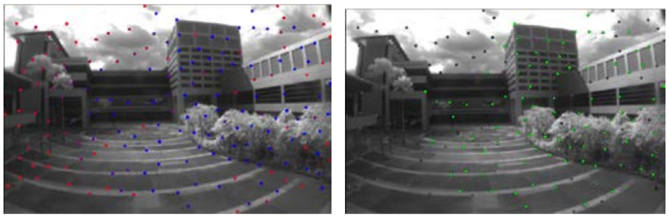
Fig. 9: Trajectories of Sequence nya Generated from VINS-Monoand OW-VINS



(a) sbs_01 (b) sbs_02

Fig. 10: Trajectories of Sequence sbs Generated from VINS-Mono and OW-VINS

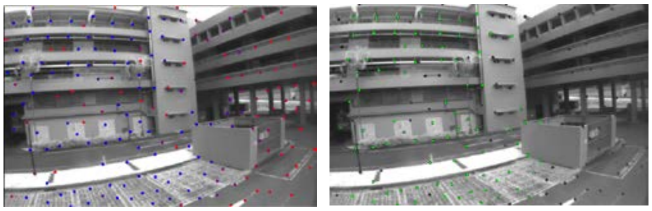
Specifically, Figure 10a shows the estimated trajectory using VINS-Mono has more serious drift than that of OW-VINS in the red area. Figure 11 shows the image captured when the vehicle moves around the areas in the red circle in Figure 10a. Meanwhile, the feature points in the image rapidly moves when the cameras are rotating. That results in plenty of new feature points to be captured in the estimation. Because those new feature points are not enough to be tracked in the past, it leads to unconvinced estimation. In addition, the far feature points have few displacements in the image, which can cause unstable estimation. Therefore, OW-VINS will filter out those noisy feature points from the estimation as shown in Figure 11. Videos can be found in the multimedia attachment[2].



(a) Feature points of VINS-Mono (b) Feature points of OW-VINS

Fig. 11: Observed Feature Points of sbs_01

The deeper the circle in red, the newer the feature point is in Figure 11a. The deeper the circle in blue, the more times the feature point is in track in Figure 11a. It is obvious that many red feature points and far feature point, like the clouds, are filtered by OW-VINS in Figure 11b. The method of OW-VINS reduces the influence of those noisy feature points since their observability is low. Therefore, the improvement from OW-VINS to VINS-Mono in sbs sequences is better compared to eee sequences and nya sequences. Similarly, there is a serious drift trajectory in the red area in Figure 8b, and the main reason is caused by the noisy feature points when the camera is rotating as shown in Figure 12. The UAV rotates the camera to the right, which results in numerous new feature points captured in the right-hand side of the image in Figure 12a. As mentioned above, the red circle denotes the feature point that is not tracked and also possess certain noise with estimation. Instead, OW-VINS filter those noisy feature points in Figure 12b, which lacks in the features around the corners of the image. Thus, OW-VINS obtains better performance than VINS-Mono in Figure 8b.



(a) Feature points of VINS-Mono (b) Feature points of OW-VINS

Fig. 12: Trajectories of Sequence sbs Generated from VINS-Mono and OW-VINS

C. Hilti SLAM Challenge

Hilti SLAM Challenge [43] is a up-to-date dataset for SLAM. Together with the Oxford Robotics Institute and the Robotics and Perception Group from University of ZÃŒrich, they have created benchmarks for SLAM problems in environments with challenging features such as ill light conditions, difficult geometries, and fast movements. The sensor suite consists of a Sevensense Alphasense Core camera head with 5 x 0.4MP global shutter cameras, and a Hesai PandarXT-32 as shown in Figure 13. All these sequences shown from Figure 14a to Figure 16bare characterized by the featureless areas and varying illumination conditions that are typical real-world scenarios and pose great challenges to evaluate SLAM algorithms that have been developed in confined lab environments. The basic information for data collection is described in Table IV.

[2] https://www.youtube.com/watch?v=P6kWT8X7Dn4

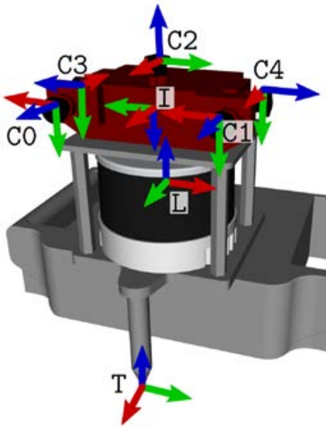


Fig. 13: Hardware Structure of Hilti Dataset

(a) Basement (b) Basement 4

Fig. 14: Environment where the Hilti datasets are collected

(a) Lab (b) Campus 2

Fig. 15: Environment where the Hilti datasets are collected

(a) Exp04 Construction Upper Level 1 (b) Exp06 Construction Upper Level 3

Fig. 16: Environment where the Hilti datasets are collected

In the datasets, this work used two cameras C3 and C4 as shown in Figure 13 as our camera measurement inputs. Instead of the two cameras facing the same direction in Section IV-B, C3 and C4 face to the opposite direction as shown in Figure 19. The main idea is to verify that the estimated trajectory is more accurate by switching between the cameras when they have various image views in the same environment. VINS-MONO consistently used the right camera C3 as its image input. Instead, OW-VINS firstly determines which camera should be used based on their observability, and secondly estimates the states with considering every feature point's observability in the remaining image. With Table V, it is obvious that OW-VINS performs much better than VINS-Mono for more than 40 percentage in most datasets. However, there is no obvious improvement in RPE in Table VI. The local drift caused by frequently switching the camera input will result in big difference in Exp04 Construction Upper Level 1 and Exp05 Construction Upper Level 2 in Table VI. Nevertheless, the global trajectory results are well estimated in Table V.

Fig. 17 depicted the APE described in Table V, and it clearly shows that OW-VINS consistently achieves lower APE across all sequences, indicating better positioning performance. Similarly, Fig. 18 illustrates the RPE results described in Table VI. As with the APE results, OW-VINS shows slight improvements or maintains parity in most sequences.

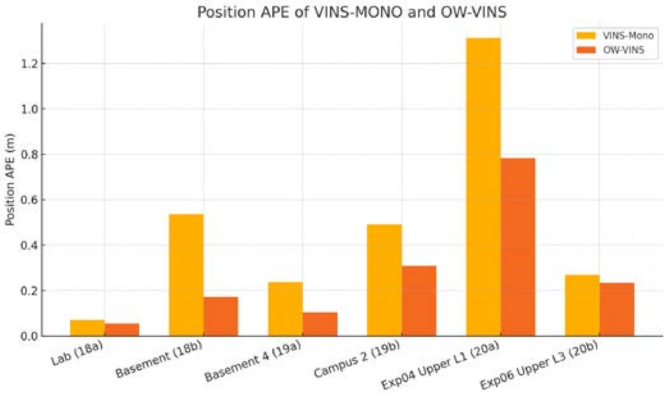


Fig. 17: The bar chart depicts the Position APE (in meters) for each sequence, comparing VINS-MONO and OW-VINS across six new sequences.

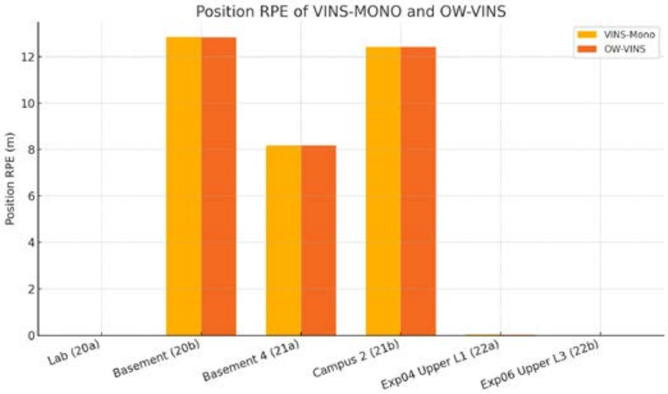


Fig. 18: The bar chart illustrating Position RPE (in meters) for each sequence, comparing VINS-MONO and OW-VINS across six test sequences.

TABLE IV: The Information of HILTI Dataset

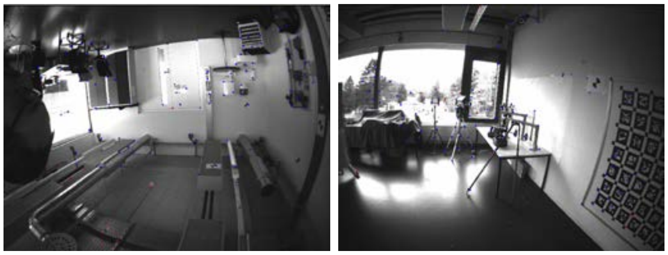
Sequence	Path Length (m)	Duration (s)
Lab (Figure 20a)	267	135.6
Basement (Figure 20b)	46	99.7
Basement 4 (Figure 21a)	97	331.9
Campus 2 (Figure 21b)	157	359.7
Exp04 Construction Upper Level 1 (Figure 22a)	79	125.5
Exp06 Construction Upper Level 3 (Figure 22b)	93	150.5

TABLE V: Position APE (m) of VINS-MONO and WO-VINS

No.	Sequence	VINS-Mono	OW-VINS	Progress (%)
1	Lab (Figure 20a)	0.070082	0.055219	21
2	Basement (Figure 20b)	0.535126	0.170664	67
3	Basement 4 (Figure 21a)	0.237276	0.103817	56
4	Campus 2 (Figure 21b)	0.490775	0.309003	38
5	Exp04 Construction Upper Level 1 (Figure 22a)	1.311918	0.783308	41
6	Exp06 Construction Upper Level 3 (Figure 22b)	0.268005	0.233442	11

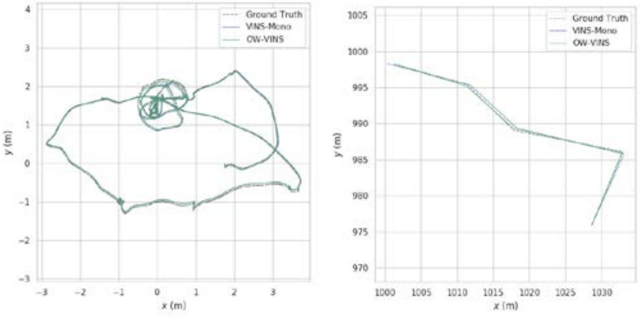
TABLE VI: Position RPE (m) of VINS-MONO and WO-VINS

No.	Sequence	VINS-Mono	OW-VINS	Progress (%)
1	Lab (Figure 20a)	0.005866	0.006110	-4
2	Basement (Figure 20b)	12.851721	12.837234	0.1
3	Basement 4 (Figure 21a)	8.181421	8.179194	0.1
4	Campus 2 (Figure 21b)	12.420817	12.422723	0.01
5	Exp04 Construction Upper Level 1 (Figure 22a)	0.031891	0.016871	66
6	Exp06 Construction Upper Level 3 (Figure 22b)	0.013691	0.015211	-15



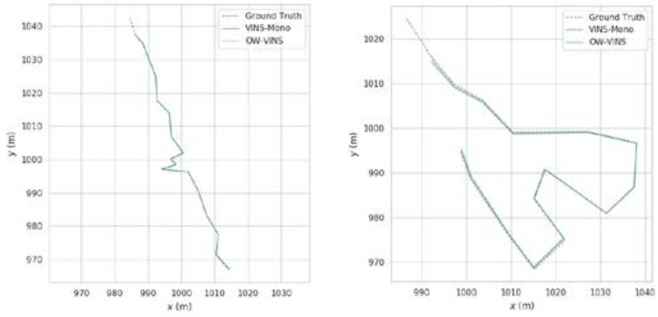
(a) Camera C3 View (b)Camera C4 View

Fig. 19: Camera Views at The Same Time



(a) Lab (b) Basement

Fig. 20: Trajectories Generated from VINS-Mono and OW-VINS

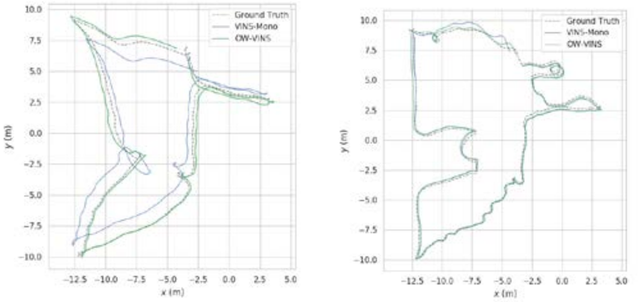


(a) Basement 4 (b) Campus 2

Fig. 21: Trajectories Generated from VINS-Mono and OW-VINS

The detailed estimated trajectories VINS-Mono and OW-VINS are illustrated from Figure 20a to Figure 22b. The trajectory obtained by using OW-VINS fits more to the gound truth than the that of VINS-Mono. Especially, in the Exp04 Construction Upper Level 1 sequence as shown in Figure 22a, there is a severe drift estimated by VINS-Mono in the red circle area. Instead, OW-VINS utilizes another image with higher observability value as the camera measurement to avoid drift. Videos can be found in the multimedia attachment[3].

[3] https://www.youtube.com/watch?v=bDqoSY81M7k



(a)Exp04 Construction Upper Level 1 (b) Exp06 Construction Upper Level 3

Fig. 22: Trajectories Generated from VINS-Mono and OW-VINS

V. Conclusion And Future Work

The observability analysis and application of VIO algorithm have been developed and engaged in recent years. One of the main contributions of this work is utilizing the observability of the estimation system to evaluate the distribution of the feature points. OW-VINS has a better estimation result than VINS-Mono by considering the observability of the feature points as weights in the estimator. The other contribution is finding out which camera is not proper for the estimation in real-time to avoid incredible measurement resulting in drift.

Based on the experimental results, this work concludes that OW-VINS is capable of discarding failed sensors to stabilize and improve estimation performance. In recent years, more and more diverse sensor measurements, like RGBD cameras, wide-range cameras, radar, or GPUs, are applied in the field of unmanned vehicles. A potential and unavoidable danger is the failure of sensing and localization. Obviously, the more amount sensors are, the higher the probability of sensor failure. Fortunately, OW-VINS can filter the improper camera input in this work, which may be able to deal with the aforementioned problems. Therefore, future work will find out the occurring of the failed sensors and drop them to reduce estimation error through the sensor observability.

References

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[2] A. Martinelli, “Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination,” *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 44–60, 2012.

[3] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, “Consistency analysis and improvement of vision-aided inertial navigation,” *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 158–176, Feb. 2014.

[4] D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis, “On the consistency of vision-aided inertial navigation,” in *Experimental Robotics.* Springer, 2013, pp. 303–317.

[5] C. X. Guo and S. I. Roumeliotis, “Imu-rgbd camera 3d pose estima-tion and extrinsic calibration: Observability analysis and consistency improvement,” in *2013 IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2935–2942.

[6] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, “Camera-imu-based localization: Observability analysis and consistency improvement,” *Int. J. Rob. Res.*, vol. 33, no. 1, pp. 182–201, 2014.

[7] J. Farrell, *Aided navigation: GPS with high rate sensors.* McGraw-Hill, Inc., 2008.

[8] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2017.

[9] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *6th IEEE and ACM Int. Symp. Mix. Augment. Real.*, 2007, pp. 225–234.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *IEEE Int. Conf. Robot. Autom.*, 2014, pp. 15–22.

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

[12] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *Eur. Conf. Computerv*. Springer, 2014, pp. 834–849.

[13] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments,” in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 957–964.

[14] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A robust and modular multi-sensor fusion approach applied to mav navigation,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 3923–3929.

[15] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint Kalman filter for vision-aided inertial navigation,” in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3565–3572.

[16] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust visual inertial odometry using a direct EKF-based approach,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2015, pp. 298–304.

[17] S. Shen, N. Michael, and V. Kumar, “Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 5303–5310.

[18] Z. Yang and S. Shen, “Monocular visual–inertial state estimation with online initialization and camera–imu extrinsic calibration,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 39–51, 2016.

[19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimiza-tion,” *Int. J. Rob. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[20] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, 2017.

[21] T. Lupton and S. Sukkarieh, “Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions,” *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, 2011.

[22] A. Martinelli, “Visual-inertial structure from motion: observability vs minimum number of sensors,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 1020–1027.

[23] A. Martinelli, “State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration,” *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 239–255, 2011.

[24] A. Martinelli, “Visual-inertial structure from motion: Observability and resolvability,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 4235–4242.

[25] A. Martinelli, “Nonlinear unknown input observability: The general analytic solution,” *arXiv preprint arXiv:1704.03252*, 2017.

[26] A. Martinelli, “Nonlinear unknown input observability: Extension of the observability rank condition,” *IEEE Trans. Automat. Contr.*, vol. 64, no. 1, pp. 222–237, 2018.

[27] A. Martinelli, “Visual-inertial structure from motion: observability vs minimum number of sensors,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 1020–1027.

[28] A. Martinelli, “Closed-form solution to cooperative visual-inertial structure from motion,” *arXiv preprint arXiv:1802.08515*, 2018.

[29] R. Hermann and A. Krener, “Nonlinear controllability and observability,” *IEEE Trans. Autom. Control*, vol. 22, no. 5, pp. 728–740, 1997.

[30] F. M. Mirzaei and S. I. Roumeliotis, “A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation,” *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1143–1156, 2008.

[31] J. Kelly and G. S. Sukhatme, “Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration,” *Int. J. Rob. Res.*, vol. 30, no. 1, pp. 56–79, 2011.

[32] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” *Int. J. Rob. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[33] M. Li and A. I. Mourikis, “Improving the accuracy of ekf-based visual-inertial odometry,” in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 828–835.

[34] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, “Towards consistent vision-aided inertial navigation,” in *Algorithmic Found. Robotics X*. Springer, 2013, pp. 559–574.

[35]K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, “Vins on wheels,” in *IEEE Int. Conf. Robot*. Autom. (ICRA), 2017, pp. 5155–5162.

[36] K. Eckenhoff, L. Paull, and G. Huang, “Decoupled, consistent node removal and edge sparsification for graph-based slam,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016, pp. 3275–3282.

[37] T. Qin, P. Li, and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[38] Z. Chen, K. Jiang, and J. C. Hung, “Local observability matrix and its application to observability analyses,” in *[Proceedings] IECON’90: 16th Annu. Conf. IEEE Ind. Electron. Soc.*, 1990, pp. 100–103.

[39] B. Hinson, “Observability-based guidance and sensor placement,” Ph.D. dissertation, University of Washington, 2014.

[40] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman & Hall, 2004.

[41] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, 2nd ed., ser. CRC Appl. math. nonlinear sci. Chapman & Hall, 2011.

[42] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, “Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint,” *Int J Rob Res*, vol. 41, no. 3, pp. 270–280, 2022.

[43] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon, “Hilti-oxford dataset: A millimetre-accurate benchmark for simultaneous localization and mapping,” *arXiv preprint arXiv:2208.09825*, 2022.