

# Research on the Availability of VINS-Mono and ORB-SLAM3 Algorithms for Aviation

BURAK KAAAN ÖZBEK  
Department of Computer Engineering  
Istanbul Commerce University  
Istanbul, TURKEY

METIN TURAN  
Department of Computer Engineering  
Istanbul Commerce University  
Istanbul, TURKEY

**Abstract**— Navigation, which most people have an idea about, is often known as information about how to get from one point to another on the land. Navigation is an interest area for both civil and military aviation. Therefore, we get help from various sensors in order to trace the aircraft route. The most used sensor among them is GPS. It is a sensor that can be out of use although it has high accuracy rates. This research focused on being able to maintain the navigation of an aircraft in the environments where GPS is out of use. There are two visual-inertial navigation systems VINS-Mono and ORB-SLAM3, which are the best known algorithms in the literature, were examined and compared in terms of performance. It has been observed that ORB-SLAM3 outperformed almost twice the VINS-Mono system in various situations.

**Keywords**—slam, visual-inertial navigation, visual odometry

Received: July 10, 2020. Revised: December 1, 2020. Accepted: December 19, 2020. Published: December 23, 2020.

## 1. Introduction

In its simplest definition, navigation is a process finding a way from one point to another. Aircraft navigation is one of the most important application areas, whether military or civil, in order to complete mission. Navigation in aviation is generally carried out by two methods known as dead reckoning and piloting. We can explain pilotage as the method by which the pilot navigates by reference to various visual landmarks such as rivers, cities, airports and houses. However, sometimes the reference points are not easily detected in situations with poor visibility or if the pilot is slightly off course [12]. Demanding more skill and experience than pilotage, dead reckoning is the method the pilot uses while flying overseas, forests or deserts. It is a navigation method that depends only on parameters such as time, airspeed, distance and direction. The pilot must know the distance from one point to another. The pilot will prepare his route in advance on the pre-flight plan map. While the pilot is traveling at a constant speed, he can measure how long to

reach his target and will keep the plane in the right direction with the help of the compass. However, due to the changing wind direction, dead reckoning is not always a reliable method [6]. The sensors that assist in the methods we mentioned above called navigation aids (NAVAIDS). The most important of these sensors is the Global Positioning System (GPS). GPS has proven how reliable and effective it is as the most widely used navigation aid today. GPS can provide navigation service in any weather conditions at any time in the world while improving flight stages from the departure, progress on the route, to the navigation on the airport surface [1]. Despite its accuracy and ease of use, GPS is also prone to be erroneous results. Given the advancing technologies and large investments in aviation, it is inevitable that companies or governments will want to be more vigilant against errors and attacks. In this study, we conducted research on how we can use the Simultaneous Localization and Mapping (SLAM) approach in aviation, which is more common in robotic, in case of the navigation aids (in our case GPS) are exposed to attack or malfunction. In the

first part of this research, SLAM approaches previously proposed in literature were studied. In the second part, the fusion of the camera and the inertial measurement unit were examined. Finally, comparative tests of the two most powerful SLAM frameworks in the literature, VINS-Mono and ORB-SLAM3 were conducted and assumptions on the problems that may be encountered in aircraft mission scenarios were discussed.

## 2 Related Work

Nowadays, aviation is an area that countries attach importance to both to provide transportation in civil aviation and to protect their borders via military aviation. The permanent operation of aircraft navigation systems is very critical for safety and performance of flights. In this section, we examine the previous important studies in the literature on the SLAM systems which frequently encountered in the field of robotics. The researches [7, 22] have created a basis for manipulating the relationship between signs and geometric uncertainty. They illustrated there is a high degree of correlation between estimations of various landmark locations on the map, and those correlations increase with successive observations. When the IEEE Robotics and Automation Conference was held in 1986, the probabilistic approaches had just started to join in the area of robotics and artificial intelligence. Peter Cheeseman, Jim Crowley and Hugh Durrant Whyte were among the researchers working for theoretical methods to mapping and localization problems [2]. According to [2], SLAM is the method by which a mobile vehicle may use the map of the world and assess its location. The critical problem with SLAM as shown in Fig. 1, it needs a simultaneous calculation of both robot and landmark locations. Observations are made between the actual robot and the landmarks because the information about the actual location is not known, or we cannot measure it directly.

Localization is one of the two problems that SLAM is trying to solve. Visual Odometry (VO) is one of the subtitles of SLAM focused on in order to solve this problem. In the landmark paper [17], the word VO was introduced. This term was used because of its resemblance to the wheel odometry which gradually predicts the motion of the vehicle.

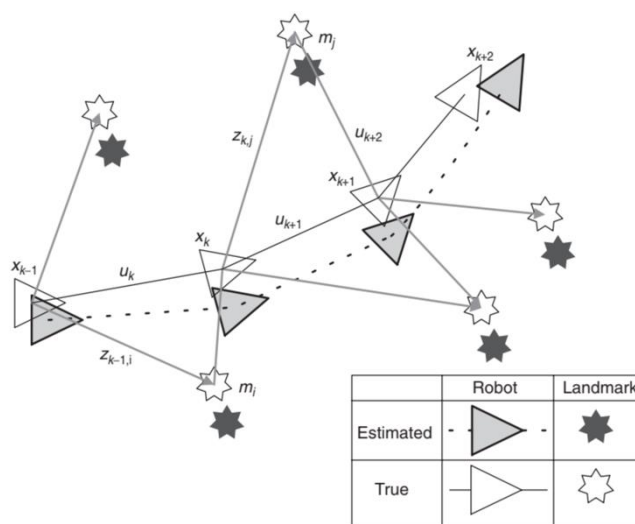


Fig. 1. Simultaneous Localization and Mapping [2]

VO estimates the relative motion of the camera by processing sequential camera images [26]. Similar to wheel odometry, the errors associated with the estimations obtained in visual odometry accumulate over time. Mapping is the second problem we need to solve in order to achieve true autonomy in localization [24]. There is a two-fold need to use a map of the environment. First, to help other functions, the map is also necessary; for example, a map may inform route planning or provide a human operator with an intuitive visualization. Second, the map allows the mistake committed to estimating the robot's state to be reduced. Dead reckoning can easily drift over time in the absence of a map; on the other hand, by revisiting identified areas, the vehicle would "reset" its localization error using a map. Therefore, in all cases in which a prior map is not usable and needs to be constructed, SLAM finds applications [5].

SLAM, on the other hand, by using the following filtering approaches such as EKF-SLAM [23] and particle filter-based SLAM [14], and smoothing approaches such as Graph-SLAM [24] and RGB-D SLAM [12], goals to solve these two problems simultaneously. The map is a large stacking sensor vector and a landmark state in EKF-SLAM and is modeled using Gaussian variables. This map usually referred to as the stochastic map, is maintained by the Extended Kalman Filter (EKF) through the method of prediction (movement of sensors) and correction (sensors observe previously mapped landmarks in the environment) [25]. Graph-SLAM solves the SLAM problem through nonlinear sparse optimization. They turn their intuition into the graphical representation of the SLAM problem.

Graph-based techniques were first described in [20], but the first working solution was given in an enlightening paper [14].

The strength of the graphical SLAM methods is that they scale to several higher-dimensional maps than the EKF-SLAM. The key limiting factor in the EKF-SLAM is the covariance matrix, which includes quadratic space in the size of the map. There are no such drawbacks in graphical approaches [25].

The other key paradigm for SLAM is the particle filters. Each particle is better thought of as a concrete guess as to what the true value of the state could be. Particle filters capture a representative sample from the posterior distribution via collecting a number of such conjectures to obtain a set of conjectures or a set of particles [15].

A general and efficient solution to navigation errors caused by the low-frequency noise of the IMU is to use a filter-based or optimization-based method to fuse visual and IMU measurements. IMU and Vision are combined during the fusing process to form a Visual-Inertial Odometry that not only takes advantage of the visual method's flexibility and is adaptable to a wide variety of scenes but also uses the IMU's high-precision features in the short term. Therefore, visual and inertial sensor-based research into the SLAM algorithm is of great significance and application importance, enabling vehicles to interpret the ambient environment in order to obtain knowledge about localization. [13]

### 3. Visual-Inertial Navigation

#### A. Sensor Fusion

In order to provide a comprehensive and complete image of the environment or process of interest, sensor fusion is the process of combining information from a number of different sources/sensors, so that the resulting information is less vague than if these sources is used individually. In autonomous systems and mobile robotics, sensor fusion methods play important role. In principle, sensor fusion processes make it possible to combine information, and provides sufficient knowledge and integrity that decisions can be formulated and implemented [27]. The design of a structure based on whether the fusion is tightly or loosely coupled is another aspect to be decided in the visual-inertial algorithm. Both

camera and Inertial Measurement Unit (IMU) measurements are determined separately in the case of loosely coupled fusion and at the end, fusion is applied to their calculation. Because the fusion of visual and inertial information is not considered to be a raw data level in a loosely coupled process, this makes the device incapable of correcting vision drifts merely by approximation. On the other hand, to estimate the position of the platform, raw measurements of the camera and the IMU is used together in tightly coupled process. More computing resources are needed for a tightly coupled approach, but it is more reliable than a loosely coupled approach [10].

For SLAM applications, there are several ways of collecting environmental measurements. For this reason, the most commonly used sensors are lidars and cameras. Monocular and stereo cameras are the most common cameras that use SLAM systems. In terms of the measuring properties for different applications, there are pros and cons of each kind of sensor. The benefit of LIDAR and stereo cameras is that they gather data on a scale and measure the area's depth. This property will be very useful in the process of data association of any SLAM scheme. However, in the case of LIDAR, for instance, it has the disadvantage of preventing the aircraft from being stealthy due to the active sensor that sends rays to the atmosphere while measurement process.

Stereo and monocular cameras are options for the camera usage. In small scale applications, stereo cameras are useful and flexible, mostly indoor SLAM applications. They are not applicable in large environments because in order to obtain accurate depth information in stereo cameras, the baseline camera and the distance to the landmark ratio should not be too small. On the other hand, using the monocular camera reduces the impact of calibration errors in estimation of motion. One of the biggest advantages of using a monocular camera is that they are cheaper than stereo cameras and easy to deploy. Although monocular cameras suffer from scale uncertainty, IMU in visual-inertial navigation helps to solve this problem.

#### B. VINS-Mono

The VINS-Mono algorithm, a closely coupled, non-linear optimization-based approach is used to obtain visual-inertial odometry by fusing inertial

IMU measurements and visual camera measurements.

Fig. 2 is the general structure for the visual-inertial pose estimation. Most of the Visual-Inertial Odometry (VIO) systems use a camera for visual data and an IMU sensor. In the framework given in Fig. 2, the inputs are the camera image and the IMU data containing acceleration and angular velocity measurements. Based on these inputs, the output is estimated by the 6-DoF (Degrees of Freedom) platform. The framework begins with the extraction of features and the pre-integration of IMU. Inertial and visual poses are combined with their pose, velocity, gyroscope bias and gravity vector for the initial estimate of the platform. These values are modified iteratively by the visual-inertial odometry algorithm. The 6-DoF pose of the platform can be obtained finally.

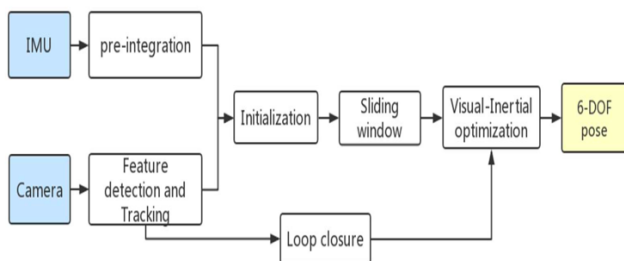


Fig. 2. General structure for the visual-inertial odometry [28]

Two approaches are applied for camera systems. The first is designing a device using a stereo camera and the other is using a monocular camera. The stereo camera approach needs long duration for reliable performance. The baseline is the distance between the two lenses of the stereo camera and the depth range that can be observed and the depth resolution. For that reason, it needs large baseline that is not practical actually for airborne applications. Fig. 3 demonstrates the detailed structure of the VINS-Mono algorithm. It starts with the measurement preprocessing stage in the structure of the VINS-Mono algorithm. This part is common for mentioned famous VIO and visual SLAM algorithms. The preprocessing measurement component contains a function extracting camera data and IMU measurements between two consecutive camera frames. Posing values, velocity, vector, gravity, gyroscope bias and 3D position of the features on the environment are obtained at the initialization. In a further stage, in the VIO module, pre-integrated IMU measurements and observations of features are fused to re-locate the device.

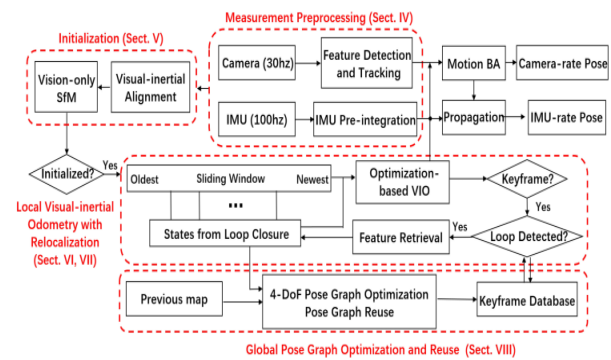


Fig. 3. Structure of VINS-Mono Algorithm [18]

Finally, pose graph optimization module is used to eliminate the drift. Re-localization and graph optimization run simultaneously in the proposed framework. In the re-localization process, loop closures that identify the locations of already been visited are identified. Based on the loop closures observed, the entire pose graph is modified on the correspondence between the loop closure frame and current frame. On the other hand, the residual errors of the edges between frames are minimized in the graph optimization portion. Edges are the relative transformation between two frames. As a result of this optimization, the pose graph changes and has become globally consistent. Fig. 4 displays the pre-integration IMU and the sample trajectory of the IMU. Camera is coupled with the features observed in the field. The basic idea is to align the visual structure with the pre-integration of the IMU.

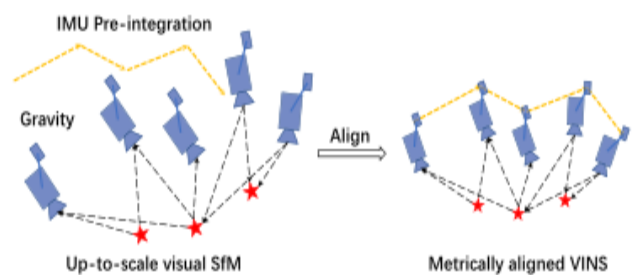


Fig. 4. IMU pre-integration [18]

Fig. 5 shows the flow after the initialization phase. This method is called “sliding window”, based on tightly coupled monocular VIO for state estimation. Fig. 6 describes the strategy for marginalization. In this method, the algorithm tests whether or not the second last frame is a mainframe. If it’s a mainframe, then it’s the oldest frame and marginalized. Marginalized visual and inertial ratios are used as above. But if the second last frame is not a mainframe, it will be removed

with its visual measurements. But inertial measurements are always preserved at the IMU pre-integration phase for additional frames. Re-localization pose graph optimization and loop closure are shown in Fig. 7, in the sliding window method. When a loop was identified with the next key-frame, the key-frame marginalized and re-located. According to the re-location, all poses are optimized in another thread.

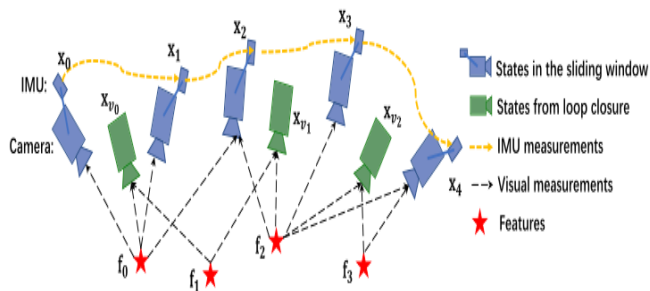


Fig. 5. Sliding window approach [18]

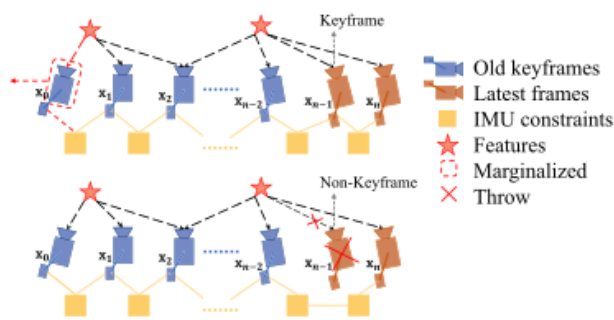


Fig. 6. Marginalization step [18]

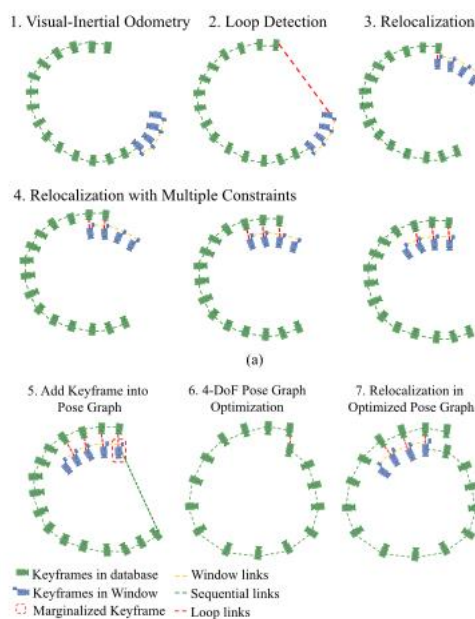


Fig. 6. Re-localization and pose graph optimization [18]

### C. ORB-SLAM3

ORB-SLAM3 is a visual-inertial SLAM system built on ORB-SLAM [16] and ORB-SLAM Visual-Inertial [20] by Carlos Campos et al. Fig. 7 shows the main parts of the ORB-SLAM3 [4]. ORB-SLAM3 claims being the best visual-inertial system in the literature. They selected Oriented Fast and Rotated BRIEF (ORB) [8] for feature extracting as shown in Fig. 8. ORB is incredibly quicker to compute and match, although it's invariant to the point of view. This allows to be aligned with wide baselines, improving the accuracy of the Bundle Adjustment (BA). Atlas is a multi-map representation made up of a number of disconnected maps. There is an active map where the tracking thread locates the incoming frames, and the local mapping thread continuously optimizes and extends with new key frames. Tracking thread processes the information of the sensor and computes the location of the current frame with respect to the active map in real time, while minimizing the error of reprojection of the corresponding map features. It also determines if the current frame will become a keyframe. In visual-inertial mode, body velocity and IMU bias are calculated by using inertial residuals in optimization. If tracking lost, the tracking thread attempts to relocate the current frame to all Atlas maps.

Local mapping thread adds the keyframes and points to the active map, eliminates redundant ones, and refines the map using visual or visual-inertial bundle adjustments, running in a local

keyframe window near the current frame. Closing loop searches for loops with each new keyframe. If a loop is found, calculates a similarity transformation that informs about the drift accumulated in the loop. At the end both sides of the loop are aligned, and the duplicate points are fused.

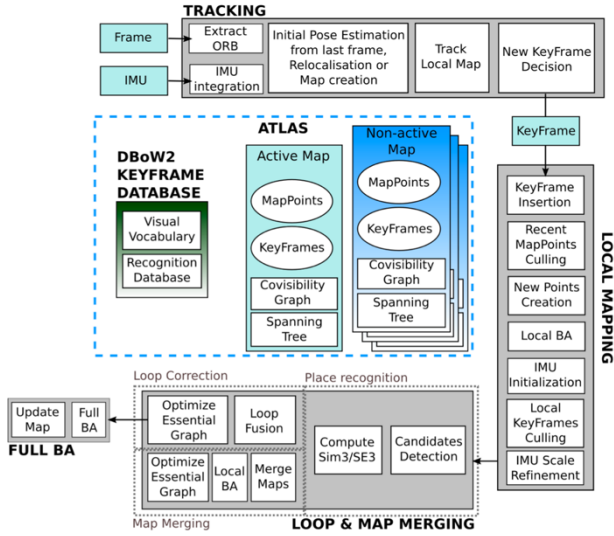


Fig. 7. Main components of ORB-SLAM3 [4]

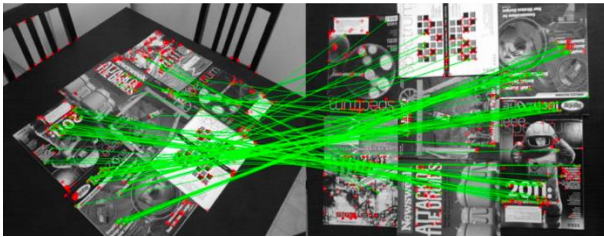


Fig. 8. Matching result using ORB [7]

## 4 Experimental Results

VINS-Mono and ORB-SLAM3 were compared in terms of availability for aviation. We showed trajectories of VINS-Mono and ORB-SLAM3 initially. Eventually a numerical analysis conducted to demonstrate the accuracy of our systems by Root Mean Square Error (RMSE).

### D. Dataset

VINS-Mono and ORB-SLAM3 were tested using a visual-inertial dataset of the EuRoC MAV [3]. Two datasets were provided. The first dataset was recorded in a large machine hall and was intended to test visual-inertial motion estimation algorithms or SLAM frameworks. A 3D location was provided by a laser tracker for ground truth. On the other hand, the second dataset was recorded in the Vicon room fitted with a motion

capture device with an approximate size of 8mx8.4mx4m.

### E. Evaluation

The EuRoC MAV visual-inertial dataset provides 11 sequences. 8 of them were selected for commenting on specific conditions. Each sequence has different environment specifications as given in Table 1. Experiments were executed on an Intel Xeon (R) CPU E5-1620 v4, at 3.50 GHz with 32 GB memory. The trajectories of 2 sequences (MH\_02, V2\_02) with their ground-truth are presented in the Fig. 9 in order to show that how the trajectories were examined. The Root Mean Square Errors (RMSE) of selected sequences in EuRoC datasets is evaluated by Absolute Pose Error (APE) using evo-tool<sup>1</sup> (Table 2). ORB-SLAM3 has outperformed in all cases.

TABLE I. ENVIRONMENT SPECIFICATIONS OF SELECTED DATASETS[3]

Name	Distance/ Duration	Avg. vel. / angular vel.	Conditions
MH_01_easy	80.6m 182s	0.44ms <sup>-1</sup> 0.22rads <sup>-1</sup>	Good texture, bright scene
MH_02_easy	73.5m 150s	0.49ms <sup>-1</sup> 0.21rads <sup>-1</sup>	Good texture, bright scene
MH_03_mediu m	130.9m 132s	0.99ms <sup>-1</sup> 0.29rads <sup>-1</sup>	Fast motion, bright scene
MH_04_difficult	91.7m 99s	0.93ms <sup>-1</sup> 0.24rads <sup>-1</sup>	Fast motion, dark scene
V1_01_easy	58.6m 144s	0.41ms <sup>-1</sup> 0.28rads <sup>-1</sup>	Slow motion, bright scene
V1_02_medium	75.9m 83.5s	0.91ms <sup>-1</sup> 0.56rads <sup>-1</sup>	Fast motion, bright scene
V2_01_easy	36.5m 112s	0.33ms <sup>-1</sup> 0.28rads <sup>-1</sup>	Slow motion, bright scene
V2_02_medium	83.2m 115s	0.72ms <sup>-1</sup> 0.59rads <sup>-1</sup>	Fast motion, bright scene

<sup>1</sup> github.com/MichaelGrupp/evo

The first thing makes an attention when look at RMSEs is that in the same environment conditions but with different flight durations VINS-Mono gave the same error rate in the sequences of MH\_01\_easy and MH\_02\_easy. On the other hand, the accuracy rate of ORB-SLAM3 is noticeably higher in the MH\_01\_easy sequence, where the flight takes longer. Although we might associate this success with better pose optimization in long-term flights, varying accuracy rates depending on the flight time will be challenging. Because, testing the success of ORB-SLAM3 in systems where flight missions such as airplanes and helicopters can change frequently, and error rates are required to be close to zero will be costly in various topics such as time and workload. When the ORB-SLAM3 or similar system is to be integrated into such aircraft, additional developments and studies are required in this regard.

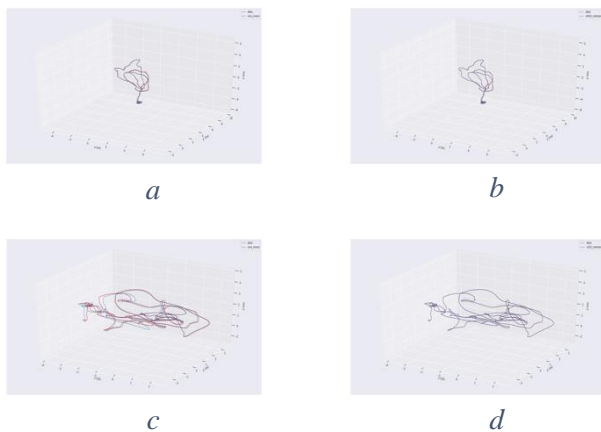


Fig. 9. VINS-Mono in MH\_02 (a) and V2\_02 (c), ORB-SLAM3 in MH\_02 (b) and V2\_02 (d)

When the MH\_03\_medium and MH\_04\_difficult sequences were traced, it cannot be fully interpreted how the dark environment affects the performance of the algorithms due to the change not only in the conditions but also in the distance between sequences. However, we know that the performance of the VINS-Mono reduces the error rate in shorter flight missions and the ORB-SLAM3 achieves high accuracy rates in long term flights. From this point of view, we can say that both algorithms do not experience serious decreases in their performance in dark environments. If it is considered that the effect of the light level on the environment for performance is resolved at the hardware level (high sensitivity cameras), it is not possible to fully understand the capabilities of these algorithms in the dark environment with EuRoC MAV dataset. However, if different types of sensors are used in air vehicles such as planes and helicopters then algorithms need to be compared not only systematically but also according to hardware differences.

TABLE II. RMSE IN DATASETS(METERS)

	VINS-Mono	ORB-SLAM3
MH_01_easy	0.182	0.016
MH_02_easy	0.182	0.065
MH_03_medium	0.404	0.041
MH_04_difficult	0.393	0.110
V1_01_easy	0.144	0.050
V1_02_medium	0.311	0.013
V2_01_easy	0.121	0.041
V2_02_medium	0.275	0.013

In the sequences such V2\_01\_easy and V2\_02\_medium where flight times are close to each other and only the speed of motion of the drone changes in the environment, we observed that the error rate increased approximately 2 times when VINS-Mono performed fast maneuvers. On the other hand, ORB-SLAM3 decreased the error rate even more. In the case, moving objects are also included in the Vicon Room (V1\_01\_easy, V1\_02\_medium, V2\_01\_easy, V2\_02\_medium) sequences [3], it is seen that ORB-SLAM3 yields better results in matching and tracking features.

ORB-SLAM3 gives better results than VINS-Mono in every case. Accuracy rate not affected by flight time, good localization and feature matching ability at high altitudes, ensuring that the accuracy rate does not decrease in blur image that may occur due to changes in weather conditions are the major problems of ORB-SLAM3 that must be solved. However, ORB-SLAM3 can still be a starting point of research for aviation.. Besides, ORB-SLAM3 can be used for aircrafts in comparison of performance in feature matching and tracking in fast motion and optimizations made in long flights.

## 5 Conclusion and Future Work

In this paper, we compared two robust visual-inertial estimators found in the literature. ORB-SLAM3, in addition to its previous systems [15, 18], gave good results with the new fast and high accurate IMU initialization technique [3]. Especially in cases where the drone was moving fast, it was successful in the IMU integration by capturing the features. However, under the same conditions, ORB-SLAM3, which produces almost 10 times higher error only in the sequence where the environment is darker. That is one of the problems to be solved for it. Although our study includes indoor tests on drones, we would like to discuss and study which parts of these systems can be developed and applied at high speeds, high altitudes and various weather conditions such as aircraft and helicopters in the future. While discussing these conditions and dealing with them in our future researches, we think that we will encounter problems

especially in feature extraction and matching. If we consider the researches and where technology is going, we can say that learning-based visual inertial systems can become widespread and it is necessary to focus on them.

#### REFERENCES

- [1] Aviation. (2020). Retrieved October 29, 2020, from <https://www.gps.gov/applications/aviation/>
- [2] Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics and Automation Magazine*, 13(3), 108–117. <https://doi.org/10.1109/MRA.2006.1678144>
- [3] Burri, Michael & Nikolic, Janosch & Gohl, Pascal & Schneider, Thomas & Rehder, Joern & Omari, Sammy & Achtelik, Markus & Siegwart, Roland. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*. 35. 10.1177/0278364915620033.
- [4] Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. M., & Tardós, J. D. (2020). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM, 1–15. Retrieved from <http://arxiv.org/abs/2007.11898>
- [5] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," in *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
- [6] Dead Reckoning. (2020, April 03). Retrieved November 09, 2020, from <https://www.flightliteracy.com/dead-reckoning/>
- [7] Durrant-Whyte, H. F. (1988). Uncertain Geometry in Robotics. *IEEE Journal on Robotics and Automation*, 4(1), 23–31. <https://doi.org/10.1109/56.768>
- [8] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2564–2571, doi: 10.1109/ICCV.2011.6126544.
- [9] Gamini Dissanayake, M. W. M., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3), 229–241. <https://doi.org/10.1109/70.938381>
- [10] Ismail, N. A., Wen, T. C., Salam, M. S., Nawi, A. M., & Mohamed, S. E. N. (2020). A review of visual inertial odometry for object tracking and measurement. *International Journal of Scientific and Technology Research*, 9(2), 355–361.
- [11] Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, 31(5), 647–663. <https://doi.org/10.1177/0278364911434148>
- [12] Houston, S. (2019, October 30). How Pilots Use Air Navigation to Fly. Retrieved October 29, 2020, from <https://www.thebalancecareers.com/how-do-pilots-navigate-282803>
- [13] K. Sun et al., "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," in *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, April 2018, doi: 10.1109/LRA.2018.2793349.
- [14] Lu, F., Milios, E. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots* 4, 333–349 (1997). <https://doi.org/10.1023/A:1008854305733>
- [15] Montemerlo, Michael & Thrun, Sebastian & Koller, Daphne & Wegbreit, Ben. (2002). FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Proceedings of the National Conference on Artificial Intelligence*.
- [16] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>
- [17] Nistér, D., Naroditsky, O., & Bergen, J. (2004). Visual odometry. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1. <https://doi.org/10.1109/cvpr.2004.1315094>
- [18] Qin, T., Li, P., & Shen, S. (2018). VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4), 1004–1020. <https://doi.org/10.1109/TRO.2018.2853729>
- [19] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.
- [20] ———, "Visual-inertial monocular SLAM with mapreuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [21] Sato, C. (1988). On Robotics. *Journal of the Robotics Society of Japan*, 6(1), 83–83. <https://doi.org/10.7210/jrsj.6.83>
- [22] Smith, R. C., & Cheeseman, P. (1986). On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4), 56–68. <https://doi.org/10.1177/027836498600500404>
- [23] Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3), 52–57. <https://doi.org/10.1145/504729.504754>
- [24] Thrun, S., & Montemerlo, M. (2006). The graph SLAM algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research*, 25(5–6), 403–429. <https://doi.org/10.1177/0278364906065387>
- [25] Thrun, S., & Leonard, J. J. (2016). Simultaneous Localization and Mapping. In 1060700160 808555050 B. Siciliano & 1060700161 808555050 O. Khatib (Authors), *Springer handbook of robotics* (p. 879). Berlin: Springer.
- [26] Webster, J. G., Huang, S., & Dissanayake, G. (2016). Robot Localization: An Introduction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1–10. <https://doi.org/10.1002/047134608x.w8318>
- [27] Yousif, K., Bab-Hadiashar, A., & Hoseinnezhad, R. (2015). An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intelligent Industrial Systems*, 1(4), 289–311. <https://doi.org/10.1007/s40903-015-0032-7>
- [28] Zheng, W., Zhou, F., & Wang, Z. (2015). Robust and accurate monocular visual navigation combining IMU for a quadrotor. *IEEE/CAA Journal of Automatica Sinica*, 2(1), 33–44. <https://doi.org/10.1109/JAS.2015.7032904>

#### Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0  
[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)