Article

Improving visual SLAM by combining SVO and ORB-SLAM2 with a complimentary filter to enhance indoor mini-drone localization under varying conditions

Amin Basiri [1],*, Valerio Mariani [1], Arman Neyestani [1] and Luigi Glielmo [1]

[1] Department of Engineering, University of Sannio in Benevento, Piazza Roma 21, 82100 Benevento, Italy.
* Correspondence: basiri@unisannio.it;

Abstract: Mini-drones can be used for a variety of tasks, such as weather monitoring, package delivery, search and rescue, and recreation. Their uses are mostly restricted to outside locations with access to the Global Positioning System (GPS) and/or similar systems since their usefulness, safety, and performance substantially rely on ubiquitously accurate positioning and navigation. Indoor localization is getting better, thanks to technologies like Visual Simultaneous Localization and Mapping (V-SLAM). However, more advancements are still required for mini-drone navigation applications with greater safety standards. In this research, a novel method for enhancing indoor mini-drone localization performance is proposed. By merging Oriented Rotated Brief SLAM (ORB-SLAM2), Semi-Direct Monocular Visual Odometry (SVO), and an Adaptive Complementary Filter, the suggested strategy improves V-SLAM approaches (ACF). The findings demonstrate that, when compared to other widely-used indoor localization algorithms, the suggested methodology performs better at estimating location under various situations (low light, low texture, and dynamic environments).

Keywords: Visual SLAM; Indoor positioning; Mini-drone

1. Introduction

Mini-drones are employed for a variety of tasks, including search and rescue, smart farming, and aerial mapping [1,2].They generally employ Global Navigation Satellite Systems (GNSS) [3–5] like GLONASS, Galileo, and GPS, which are outside positioning systems [6,7]. However, indoor use is not appropriate for these technologies. Additionally, there are restrictions on the size of the payload and a weight maximum that must be met, among other things, which make it difficult for autonomous mini-drones to navigate inside of structures.

Mini-drones may also need quick processing power due to the drone's speed, necessary safety measures, and dynamic changes [8–10].

While standard V-SLAM approaches are used to overcome the aforementioned issues in the literature [11–14], these techniques [15–17] are unable to manage changeable situations since they are mostly relevant for limited regions with distinct picture characteristics and off-board processes with acceptable illumination.

This work suggests an improved V-SLAM method that enhances autonomous mini-drone navigation in interior environments under a variety of situations, which can significantly affect the accuracy of navigation. The suggested approach combines ORB-SLAM2 and SVO, two visual algorithms, using an ACF.

The novelty of the proposed approach consists in the integration of ORB-SLAM2 [18], and SVO [19] algorithms with an ACF [20] which can improve the position estimation. A mini-drone is custom-designed for the Gazebo simulator software to implement the proposed approach. (see Fig.1).

The rest of the paper is organized as follows: Section (2) discusses the literature; Section (3) provides a system overview; Section (5) describes the experimental work undertaken, and Section (6) presents the conclusions.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Preprints* **2022**, *1*, 0. https://doi.org/

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

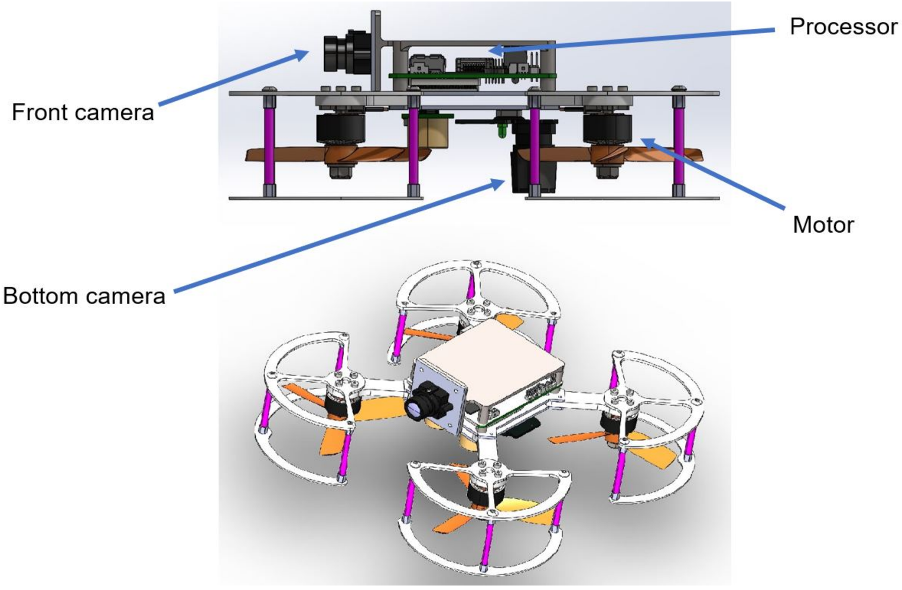


Figure 1. The customized ini-drone's model was utilized for simulations and tests [51].

2. RELATED WORK

Taketomi et al. [21] examined the most recent V-SLAM algorithms from both a technical and historical standpoint in their study, which summarized the state-of-the-art in this field. Researchers got to the conclusion that the usage of cameras and intensive data processing causes V-SLAM algorithms to run slowly, and that a quick approach may be employed to solve this problem. Recently released fast V-SLAM algorithms were compared and categorized in [22–24], following are two V-SLAM examples, namely feature-based approaches and direct methods.

2.1. Feature-based methods

Feature-based methods [25], such as ORB-SLAM, extract the important details from each frame of the images, such as blobs and corners. The mapping and localization are then accomplished using the positions of each feature in the current and previous frames. Artal et al. provide some of the fastest algorithms with feature-based methods (Fig. 2) [18].

2.2. Direct methods

Direct methods [29], such as Large-Scale Direct SLAM (LSD-SLAM), employ the entire amount of data in the image rather than just the features, giving it superior robustness and accuracy compared to feature-based methods. However, compared to ORB-SLAM, LSD-SLAM takes more computing work (Fig. 2). [32].

2.3. Recent works

Guanci et al. [33] present ORB-SLAM2, which is computationally less expensive than ORB-SLAM, has good localization accuracy, and can operate in real-time without GPU processing. The SVO algorithm, developed by Forster et al.[19] combines the benefits of feature-based and direct techniques [19]. Additionally, researchers demonstrate in [26] study that SVO is quicker than LSD-SLAM and ORB-SLAM. It cannot produce maps, however, because: i) only the most recent 5–10 frames are accessible; ii) in-depth feature estimate is lacking; and iii) it is frequently used in downward-looking cameras [26,34].

Using a previous motion, Loo et al. [35] attempted to repair the SVO flaws in order to locate the features in the fast action. Prior picture features are used in the preceding motion to calculate the current motion. As a result, they can be changed quickly, but SVO still has issues in settings with subpar features in general. Furthermore, based on

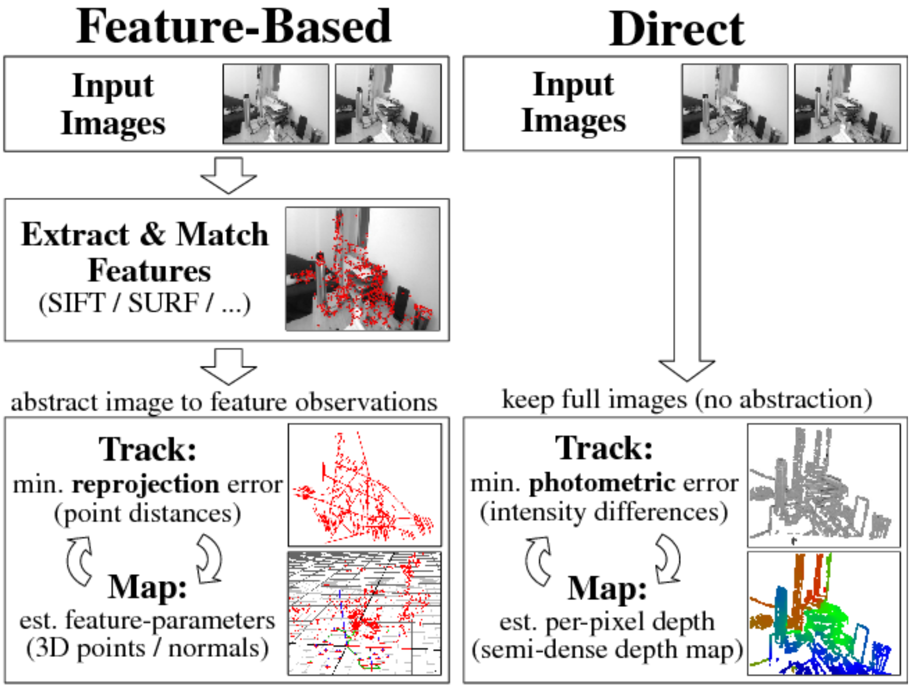

Feature-Based
Direct
Input Images
Input Images
Extract & Match Features (SIFT / SURF / ...)
abstract image to feature observations
keep full images (no abstraction)
Track: min. reprojection error (point distances)
Map: est. feature-parameters (3D points / normals)
Track: min. photometric error (intensity differences)
Map: est. per-pixel depth (semi-dense depth map)

Figure 2. Feature-based methods Abstract visuals to highlight observations and exclude all extraneous data. In contrast, the suggested direct method maps and follows picture intensities directly [47].

the characteristics of the ORB algorithm, ORB-SLAM can fix faults [30]. Additionally, it performs well while creating maps, but ORB-SLAM struggles to locate features in dynamic settings. Together, ORB-SLAM2 and SVO have qualities that allow them to function well in a variety of situations and environments.

To take advantage of both of its benefits, SVO and ORB-SLAM2 are combined in the unique V-SLAM approach that has been presented.

3. SYSTEM OVERVIEW

The system suggested in this paper for the mini-drone consists of two sections: V-SLAM and control components (Fig. 3). Two algorithms are employed in parallel threads in the V-SLAM component [51]. (Fig. 4). ORB-SLAM2 is utilized for localization and mapping, while SVO calculates the mini-position. drone's After extracting data from the ORB-SLAM2 and SVO algorithms, fusion data are extracted based on the weighted average. AFC mixes the ORB-SLAM2 and SVO extracted data based on the error. And the algorithm with the greater mistake rate utilizes a smaller proportion of the data [31,37]. The control section uses a PID controller [40].

3.1. ORB-SLAM

This algorithm consists of three components (Fig. 4): tracking, local mapping and loop closing, [34].

3.1.1. Tracking

Tracking is responsible for localizing the camera in each frame and determining when to insert a new keyframe. If the tracking is lost (e.g., due to occlusions or sudden movements), then matches with the local map points are found by reprojection, and the camera pose is optimized with all matches once again. The tracking thread then determines if a new keyframe should be placed [30,38].

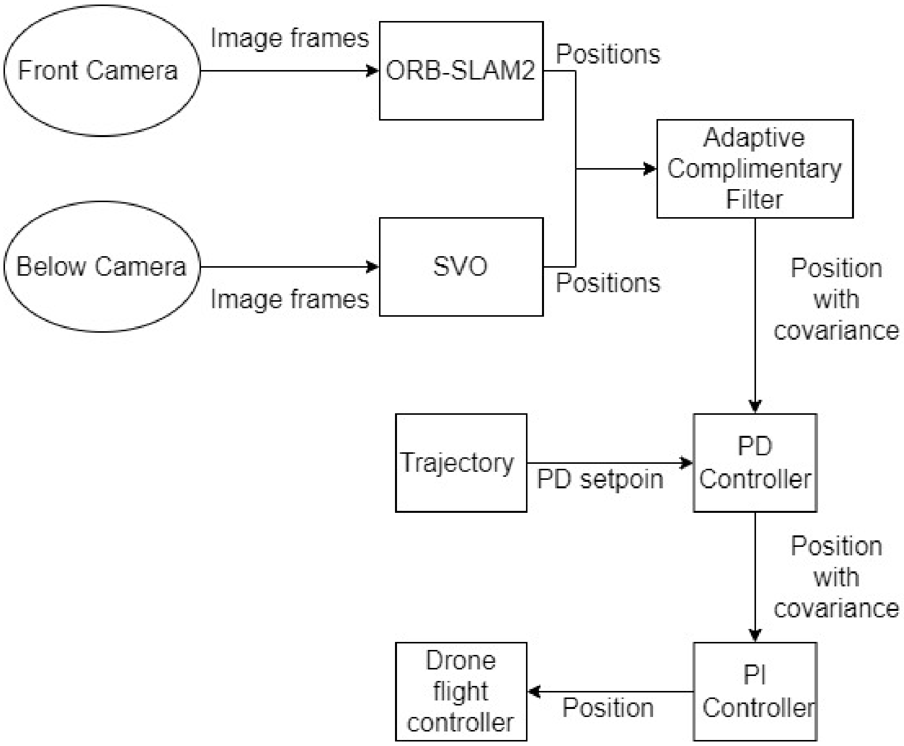


Figure 3. Control system and V-SLAM component [51].

3.1.2. Local mapping

The local mapping process adds keyframes and executes them locally to produce an appropriate reconstruction in the camera pose's surroundings. New correspondences for mismatched ORB in the new keyframe are searched in related keyframes in the visibility graph to triangulate new points. Based on the information obtained during tracking, a stringent point culling strategy is implemented sometime after creation to maintain only high-quality points. The local mapping is also in charge of removing obsolete keyframes [30,49].

3.1.3. Loop closing

It is the task of determining whether or not a drone has returned to a previously visited region after an arbitrary period of trip. If three consecutively compatible frames are detected, the algorithm accepts the current keyframes as a loop candidate. We now have a global graph map. [34].

3.2. SVO

Tracking and mapping are the two threads that make up SVO. The method is still running in a parallel thread with SVO, mostly for mapping and localization on small devices. The tracking thread employs a semi-direct approach of estimating relative posture, whereas the mapping thread employs the depth filter [26,36,50].

3.3. SVO-ORB-SLAM

The solution described in our study employs two parallel threads (indicated in Fig. 4). The ACF is used by weighting the data in each algorithm (ORB and SVO) based on error estimation according to the amount of features discovered at each frame by the FAST corner-detection algorithm in SVO and the ORB feature-detection algorithm in ORB-SLAM. However, before doing this estimation, the data must be synced with the timestamp. The synchronized data is then merged within the ACF .

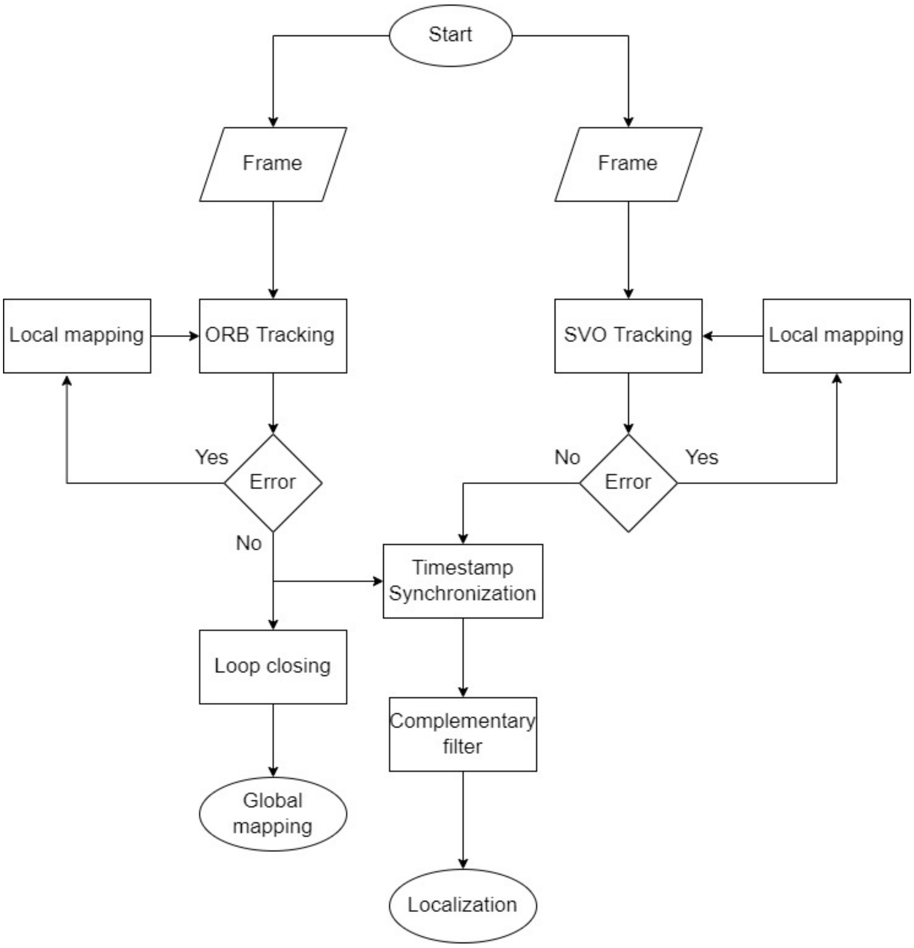


Figure 4. The flowchart for SVO and ORB SLAM [51].

3.4. ACF

ACF was chosen as the basis for the data fusion of SVO and ORB-SLAM2 in this investigation because it requires significantly less processing power than other methods, such as the Kalman filter [48].

The ACF method was used: In two filters, G_l (low pass filter) and G_h (high pass filter), where $\hat{P}$ is the estimated position of drone in (5); to data fusion, need to estimate the error based on the number of feature points in the frame of ORB-SLAM2 and SVO via (1). The filters coefficient is α, P_{orb} in (4) is the position the ORB-SLAM2 algorithm measures, and P_{svo} in (3) is the SVO algorithm's position.

Each algorithm has weight, α is the weight factor that in critical situations causes ORB-SLAM2 and SVO algorithms' data weight to change due to the position estimation error, and ACF uses more data with a minor error. The coefficient α is calculated via (2), as illustrated in Fig. 5. Each algorithm's error is calculated based on how many feature points are in each frame. α_{orb} and α_{svo} satisfy the following two conditions. The filter uses both algorithms; if the SVO's error in finding image features is significant, in finding the features in the images, the filter uses ORB-SLAM's data. If the ORB-SLAM's error in finding image features is significant, the filter uses SVO's data, as indicated in Fig. 6.

$$\text{Error} = \frac{A - N}{A} \tag{1}$$

- A = Accepted feature points number.
- N = Number of feature points found.

$$\alpha = 1 - \text{Error} \tag{2}$$

$$\hat{P}_{svo} = \alpha_{svo}\, P_{svo} \tag{3}$$

$$\hat{P}_{orb} = \alpha_{orb}\, P_{orb} \tag{4}$$

$$\hat{P} = G_I\, \hat{P}_{svo} + G_h\, \hat{P}_{orb} \tag{5}$$

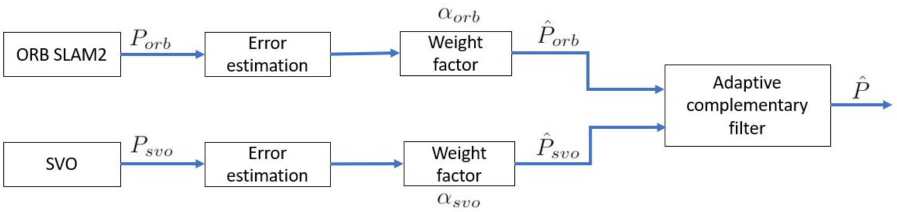


Figure 5. Diagram of computational steps before combining algorithms [51].

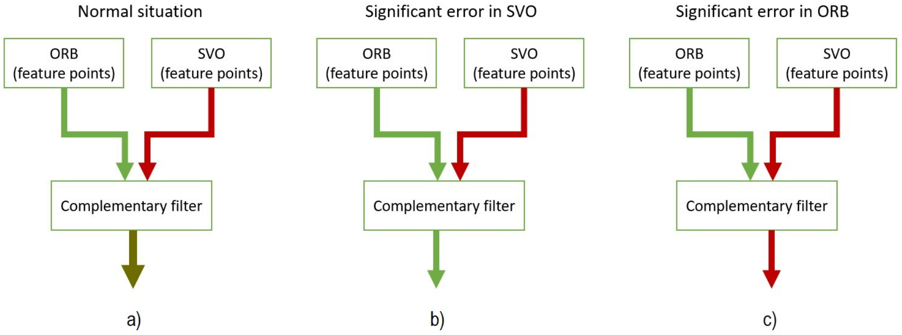


Figure 6. **a**) use data from both algorithms; **b**) use ORB-SLAM's data, and **c**) use SVO's data.

3.5. Controller

V-SLAM algorithms have some drawbacks. One is the problem in initial localization since they need the camera to move about one metre to detect the image's features. While it may not be very safe to have a one-metre error as it can cause a collision or crash, controllers are expected to have higher command ability and stability. Thus, an initial trajectory command is assigned to PD [41] for the initial state. That is, the initial setpoint is set for the PD controller. In the next step, the PD controller output is the PI controller setpoint [42]. In this part, the mini-drone begins flying autonomously. Moreover, the mini-drone requires initial localization. In this part, PI uses optical follow [28] use bottom camera, data based on velocity for position estimation. These are the steps for the first loop for mini-drone control, after which the mini-drone can fly autonomously.

Another challenge in V-SLAM algorithms is the UAV vibration and turbulence, specially when the UAVs use downward-looking cameras and do not move on the x and y axes. This vibration in the vehicle's roll and pitch can cause changes in the x and y values, leading the algorithm to assume that the robot is moving. This is a problem when the robot is at a great height, and every slight vibration in the UAV creates a significant error. Moreover, since SVO estimates the position with the camera below, we have to consider this problem. In this regard, altitude is very important because, as indicated in Fig. 7, if we have two drones moving the same distance but one is at a low altitude like in step 1, then the other one is at a higher altitude like in step 2. The lower drone will see surface features appear to move further, resulting in a higher value. We compensate for vehicle roll and pitch changes.

As indicated in Fig. 7 the drone has rolled, changed degrees, but the flower has moved from the centre of the camera's view in step 2 to the edge of the view in step 3. The expected change in sensor values can be calculated directly from the change in roll, and the pitch gave the formula 6 and 7. We subtract these expected changes from the real values returned by the sensor.

Algorithm 1 SVO-ORB-SLAM algorithm

```
function SVO-ORB-SLAM($\hat{P}, \hat{P}_{svo}, \hat{P}_{orb}, P_{orb}, P_{svo}, \alpha_{orb}, \alpha_{svo}$)
    Step 1: Capture the image from the cameras.
    Step 2: Estimate the $P_{orb}$ and $P_{svo}$.
    Step 3: Calculate the errors of $P_{orb}$ and $P_{svo}$ based on number of features.
    Step 3: Calculate the $\alpha$ for $P_{orb}$ and $P_{svo}$ based on errors.
    for Each image frame $i$ do
        Update the position of the $\hat{P}_{orb}$ and $\hat{P}_{svo}$
        Evaluate the fitness $\alpha_i$)
        if $\alpha_{orb} >= 0.5$ and $\alpha_{svo} < 0.5$ then
            $\hat{P} = \hat{P}_{orb}$;
        end if
        if $\alpha_{svo} >= 0.5$ and $\alpha_{orb} < 0.5$ then
            $\hat{P} = \hat{P}_{svo}$;
        end if
        if $\alpha_{orb} = \alpha_{svo}$ then
            $\hat{P} = (\hat{P}_{orb} + \hat{P}_{svo})/2$;
        end if
    end for
    return;
end function
```

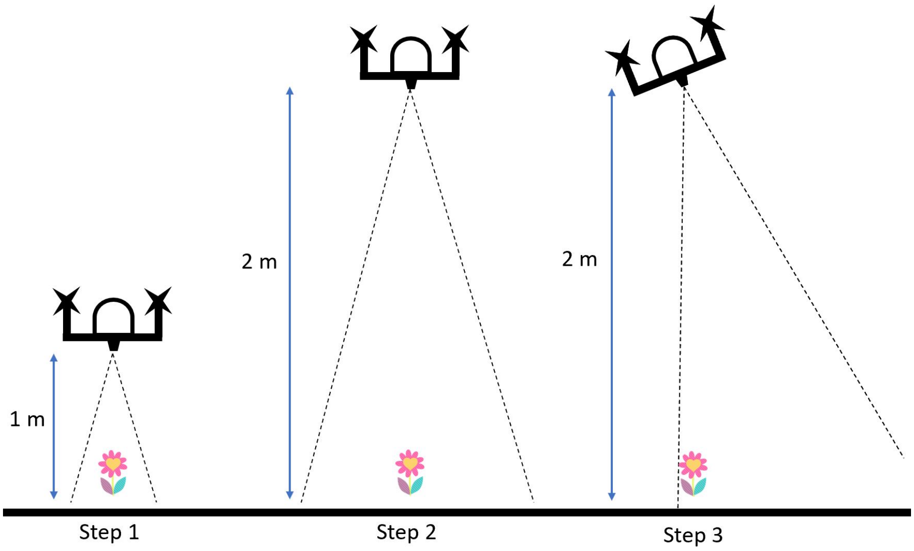


Figure 7. Drone vibrations.

- FOV = Field of View
- S_v = Sensor value
- S_{rp} = Sensor's resolution in pixel
- D_m = Distance moved
- X_e = Expected X value
- a = Altitude
- n = Scalar
- r = Change in roll

$$D_m = \frac{(S_v\, a)}{(S_{rp}\, n)} 2.0\, tan \frac{(FOV)}{(2.0)}. \tag{6}$$

$$X_e = \frac{(r\, S_{rp}\, n)}{(FOV)}. \tag{7}$$

4. Stimulation

All procedures are run in real-time to test the performance of SVO-ORB-SLAM. We ran the simulations in Gazebo 11 and Rviz software on an Intel Core i7-4702MQ, 12 GB RAM, and a 2 GB VGA GeForce GT 740M under the Robot Operating System (ROS) Melodic Morenia version at Ubuntu 18.04 (Bionic).

As shown in Fig. 8, there are two windows, one of which is the window to the right of the Gazebo environment and displays the viewing angle of the drone's front camera. The green spots inside this frame are picture characteristics discovered by the ORB-SLAM2 method. The Rviz environment is shown in the left window of Fig. 8. The green line inside this window represents the mini-drone, drone's while the white cloud points represent the 3D map generated by the ORB-SLAM2 algorithm.

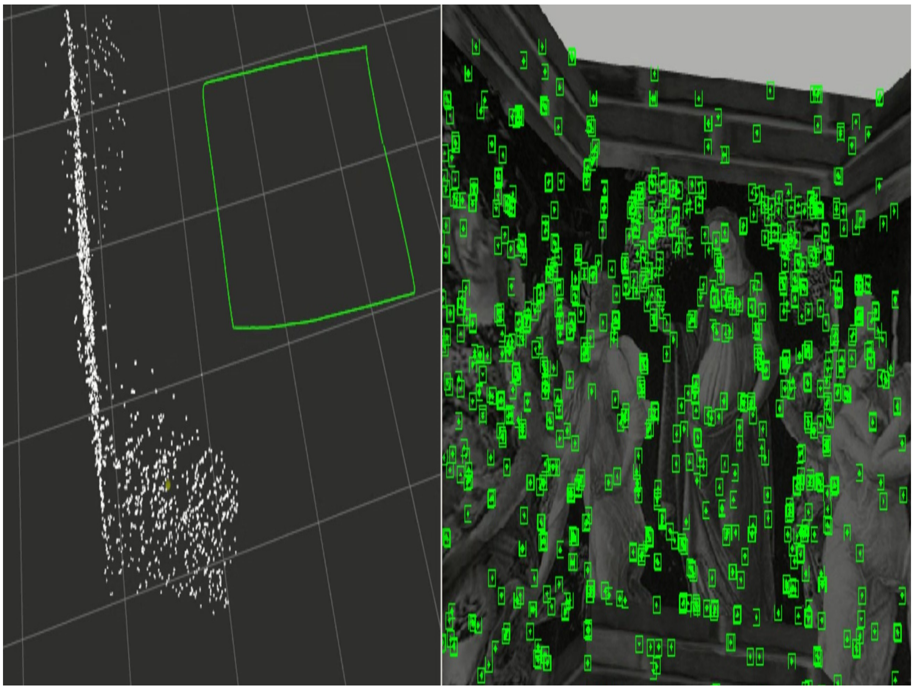


Figure 8. The green points in the ORB-SLAM2 algorithm that represents the features and green line represent the trajectory [51].

As shown in Fig. 9, the Gazebo environment in the right window displays the viewing angle of the drone's down camera. The green spots inside this window represent image features discovered by the SVO algorithm. The Rviz environment is shown in the left window of Fig.9, and the red line inside this window represents the mini-drone's trajectory as determined by the SVO algorithm.

The mini-drone is flying in the Gazebo (right window) and Rviz environments (left window) in Fig. 10, demonstrating that the yellow line's flying trajectory is the real trajectory, while the red line is SVO's estimate of the trajectory. Meanwhile, the green line represents the trajectory estimate from ORB-SLAM2, and the blue line represents the trajectory estimate from our suggested technique.

5. EXPERIMENTAL

We evaluated the mini-drone in three complicated scenarios to evaluate its performance: (1) a low-light environment, (2) a low-texture environment, and (3) a dynamic environment. Furthermore, we demonstrated that our suggested solution works very rapidly and smoothly, even with poor hardware.

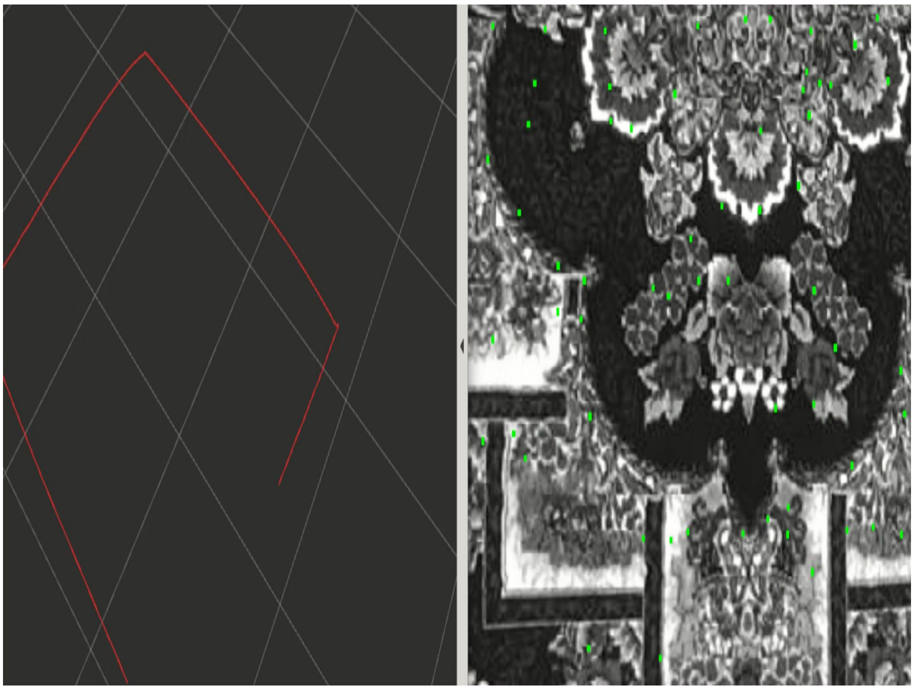


Figure 9. The trajectory and features point in the SVO algorithm [51].

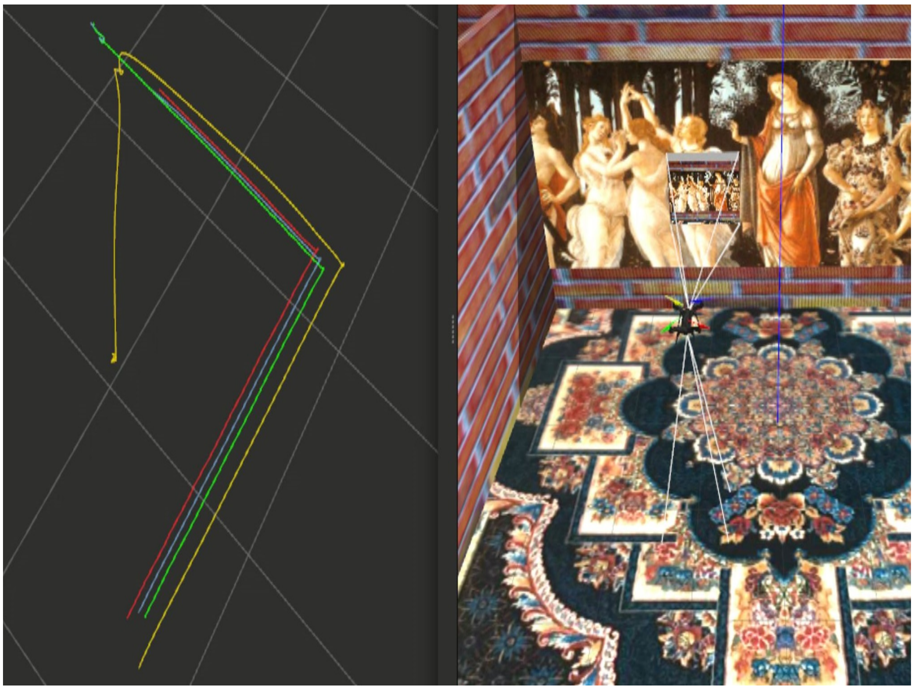


Figure 10. Simulation environment room in the Gazebo [51].

5.1. Low-light environment

Working in low-light situations is one of the most difficult issues in V-SLAM since these approaches rely on data from pictures captured by the camera. V-SLAM feature-detection algorithms perform differently at various brightness levels. Choosing the optimal feature algorithm has always been difficult, but our suggested solution employs two feature algorithms at the same time: the FAST algorithm in SVO and the ORB algorithm in ORB-SLAM2. This advantage allows us to utilize our proposed localization method even in the darkest conditions.

Half of the mini-flight drone's path was evaluated in natural light inside, as shown in Fig. 11, section (**a**). SVO, ORB-SLAM2, and our localization approach all performed admirably. Suddenly, we blocked the sun from the rest of the mini-flight drone's path. As shown in Fig. 11 section (**b**), The SVO has lost its position and cannot locate the drone, but ORB-SLAM2 and our approach can provide an approximation of its position.

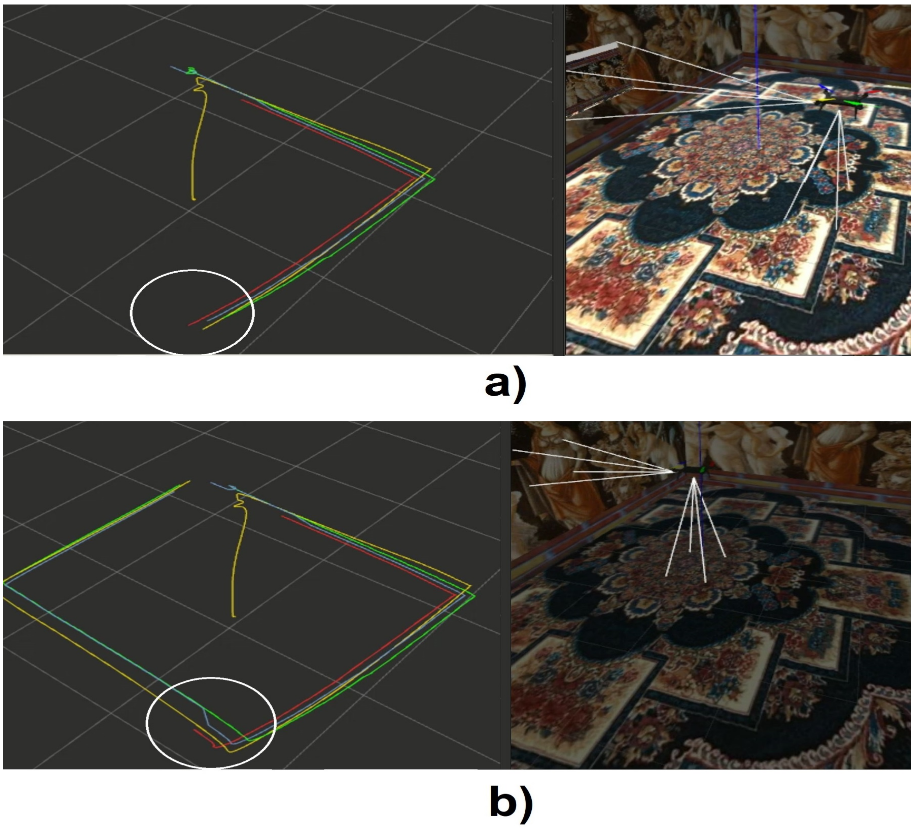


Figure 11. Drones under two different lighting conditions: (**a**) a bright room and (**b**) a dark room [51].

5.2. Low-texture environment

In feature-based approaches, textures, corners, and edges in the environment can assist in locating the image's features. When we observe more characteristics in the image, localisation is improved, hence poor texture in the image is a significant issue for V-SLAM.

Our approach combines the two methods and the usage of front data with ORB-SLAM2 and bottom data with SVO; even when the drone enters an area where the walls are not visible to the camera, such as a huge salon, it can estimate its location using SVO. When entering an area with a surface with few of textures, ORB-SLAM2 can estimate the position. As shown in Fig. 12, in an area devoid of objects and walls, the mini-drone has a low texture; the large surface also has a low texture, and the SVO missed the target location. Our approach can estimate the location using ORB data.

5.3. Dynamic environment

One of the most difficult challenges that drones must always overcome is a dynamic environment. In this situation, we demonstrate that localization works effectively with our approach, despite the fact that moving objects and walls as well as changing maps.

As shown in Fig. 13 (**a**) before modifying the map and (**b**) after changing the map. We made some changes to the map so that it would reflect that ORB-SLAM had failed to locate the position indicated by the white circle. Despite this, the ACF automatically uses all of the position data from SVO in order to estimate the position in order to prevent acquiring a lost position.

tex.JPG

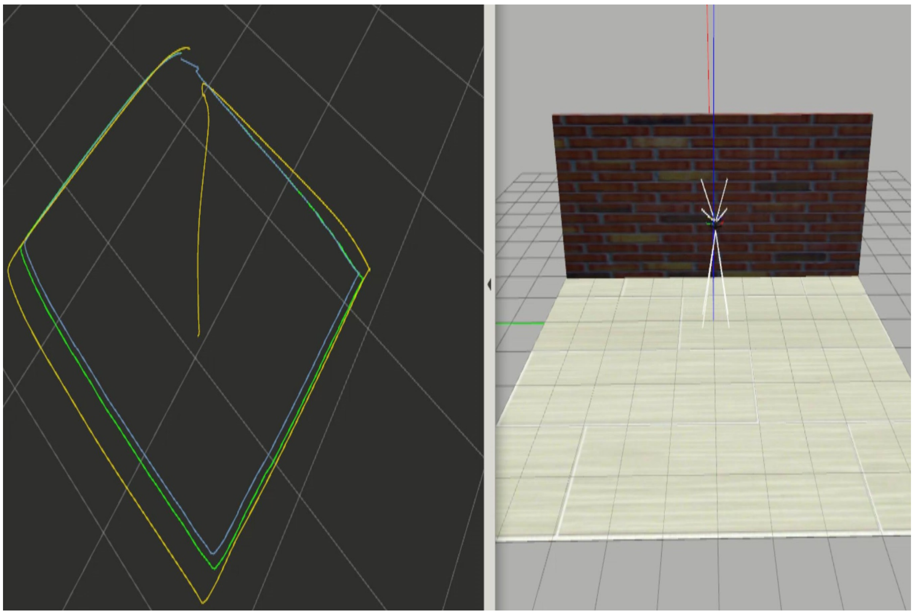


Figure 12. Test the mini-drone in a Low-texture environment [51].

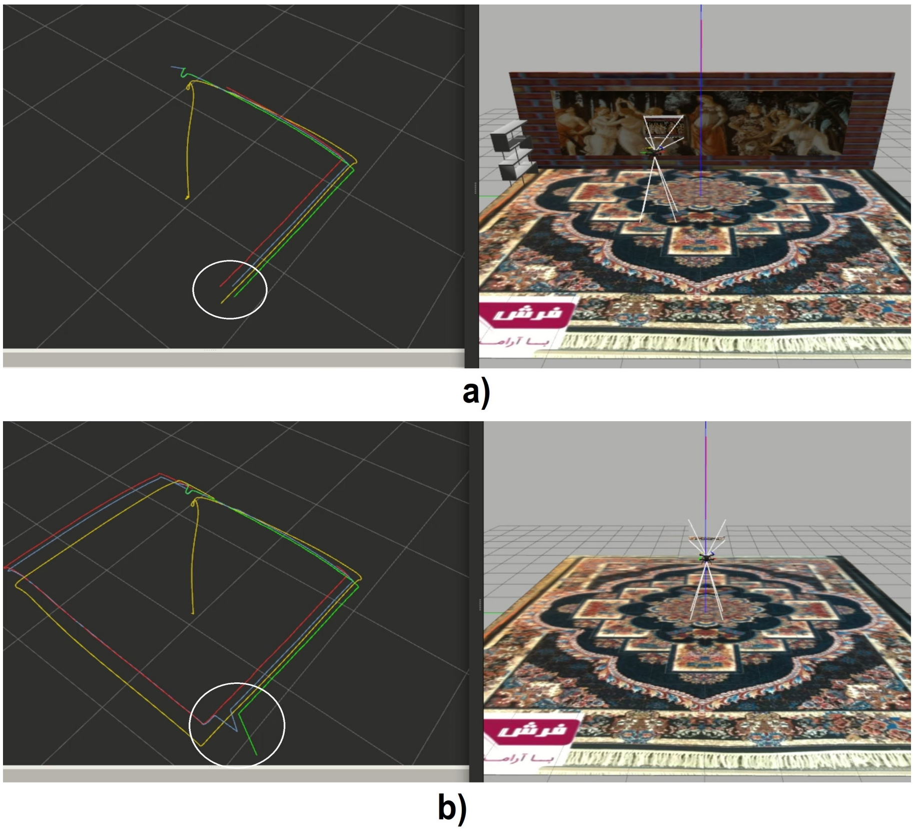
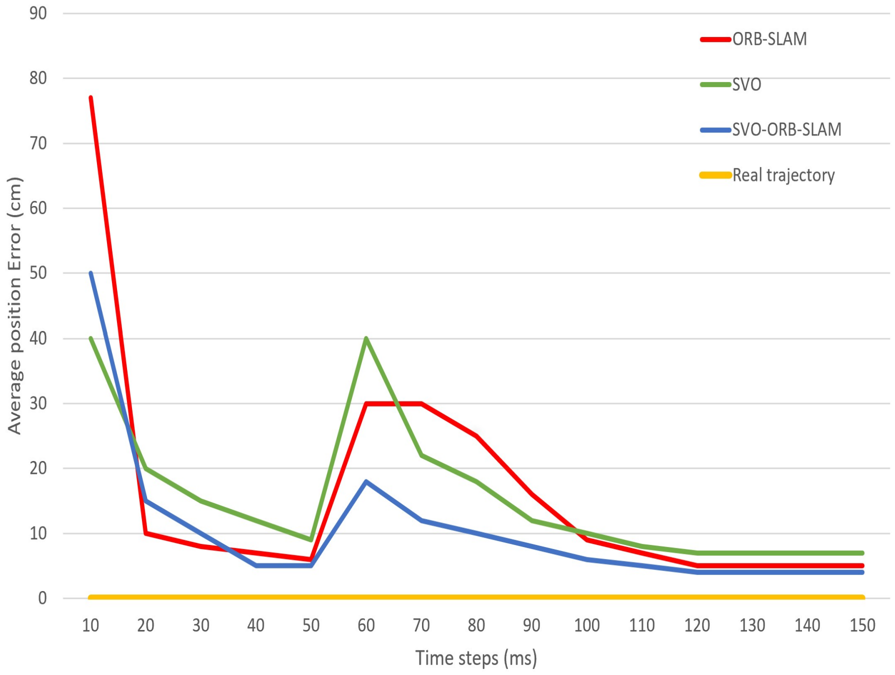


Figure 13. (**a**) The dynamic environment, before the map was changed, and (**b**) the dynamic environment after the map was changed [51].

The results of a comparison between our suggested technique and SVO and ORB-SLAM2 in terms of the average position error in an environment that is dynamic and has low light and low texture are shown in Fig 14.

Our approach has an error rate of 15.2% on average across all situations, as shown in Table 1. Comparatively, the ORB-SLAM2 method has an error rate of 32.3%, while the SVO method has an error rate of 44.1%.



pos.jpg

Figure 14. Comparison of SVO, ORB-SLAM2 and SVO-ORB-SLAM algorithms [51].

Table 1. Average percentage error during the duration of a low-light, low-texture, and dynamic environment [51].

Algorithms	Average percentage error
SVO	44.1%
ORB-SLAM2	32.3%
SVO-ORB SLAM	15.2%

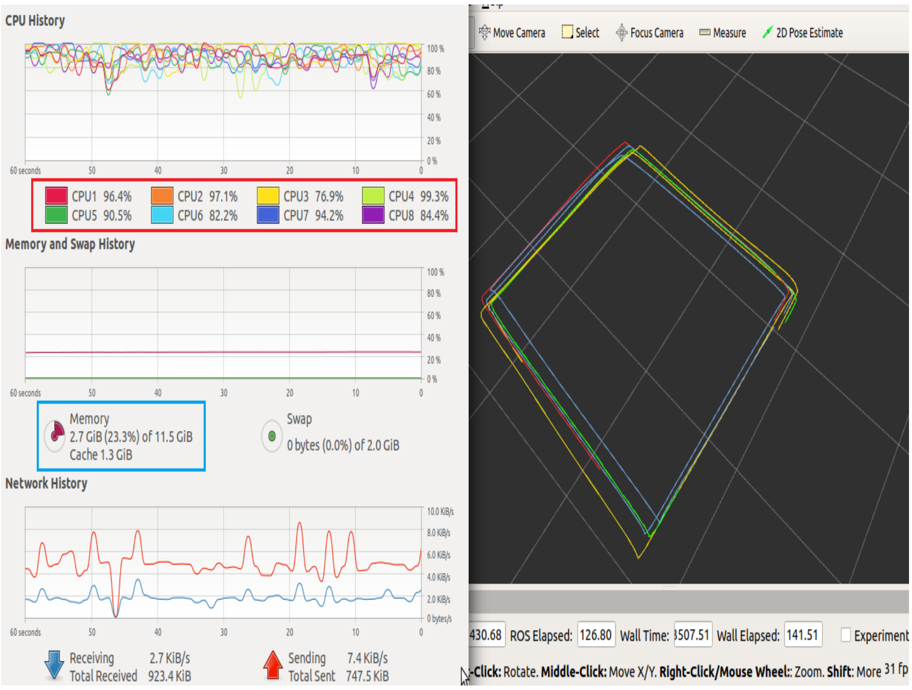
5.4. Processing

It is well known that the V-SLAM algorithms call for a significant amount of computational power in order to operate in an effective manner. Due to the restricted area available, this may be a difficulty for systems that have been shrunk down, such as small drones. This piece makes use of ORB-SLAM2 and SVO, which are the two methods that are currently recognized as being the quickest accessible options. Both of these methods are broken down into much more specific steps as explained in the previous paragraphs. Additionally, a suitable amount of labour has been included in the processing in order to maintain a level of efficiency that may be regarded as adequate. As shown in Fig. 15, the solution that is recommended makes use of just 23.3% of the total RAM that is available, and the processing that it executes on the CPU cores is very smooth.

6. CONCLUSIONS

In this paper, we suggest an approach for mini-drones to use in environments where GPS is unavailable, such as indoors. This strategy is called ORB-SVO SLAM and is based on V-SLAM, it also has the benefits of SVO and ORB-SLAM2. An ACF provides fusion data, which has a higher performance than SVO and ORB-SLAM2 despite having a lower error rate overall. This error rate is 28.9 percent lower than SVO, and 17.1 percent lower than ORB-SLAM2. The findings indicate that our strategy is applicable even in areas with a limited amount of light and a low texture.

(1).png



CPU1 96.4% CPU2 97.1% CPU3 76.9% CPU4 99.3%
CPU5 90.5% CPU6 82.2% CPU7 94.2% CPU8 84.4%

Memory
2.7 GiB (23.3%) of 11.5 GiB
Cache 1.3 GiB

Swap
0 bytes (0.0%) of 2.0 GiB

Receiving 2.7 KiB/s
Total Received 923.4 KiB

Sending 7.4 KiB/s
Total Sent 747.5 KiB

430.68 ROS Elapsed: 126.80 Wall Time: 3507.51 Wall Elapsed: 141.51

-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More 31 fp

Figure 15. Computing process (Memory consumption in the blue rectangle and 8 processor cores in the red rectangle) [51].

References

1. Krul, Sander. Visual SLAM for Indoor Livestock and Farming Using a Small Drone with a Monocular Camera: A Feasibility Study. Drones **2021**.
2. Sampedro, Carlos. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. Journal of Intelligent & Robotic Systems **2019**.
3. Walker, Ory. A Framework for Multi-Agent UAV Exploration and Target-Finding in GPS-Denied and Partially Observable Environments. Sensors **2020**.
4. Sidorov, Dmitry. Adopting the empirical CODE orbit model to Galileo satellites. Advances in Space Research **2020**.
5. Pan, Lin. Characterizing inter-frequency bias and signal quality for GLONASS satellites with triple-frequency transmissions. Advances in Space Research **2019**.
6. Zhu, Xiaoqiang. Indoor Intelligent Fingerprint-Based Localization: Principles, Approaches and Challenges. IEEE Communications Surveys & Tutorials **2020**.
7. De Croon, Guido. Challenges of Autonomous Flight in Indoor Environments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) **2018**.
8. Grubesic, Tony H. Uavs and urban spatial analysis. Springer **2020**.
9. Hassanalian, Mostafa. Classifications, applications, and design challenges of drones: A review. Progress in Aerospace Sciences **2017**.
10. Shi, Xiufang. Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges. IEEE Communications Magazine **2018**.
11. García, Sergio. Indoor SLAM for micro aerial vehicles control using monocular camera and sensor fusion. International conference on autonomous robot systems and competitions (ICARSC) **2016**.
12. Zou, Danping. Collaborative visual SLAM for multiple agents: A brief survey. Virtual Reality & Intelligent Hardware **2019**.
13. Vidal, Antoni. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. IEEE Robotics and Automation Letters **2018**.
14. Yang, Tao. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. Electronics **2018**.
15. Wang, Huan. Multi-features visual odometry for indoor mapping of UAV. 3rd International Conference on Unmanned Systems (ICUS) **2020**.
16. Al-Kaff, Abdulla. Survey of computer vision algorithms and applications for unmanned aerial vehicles. Expert Systems with Applications **2018**.
17. Trujillo, Juan-Carlos. Monocular Visual SLAM Based on a Cooperative UAV–Target System. Sensors **2020**.
18. Mur-Artal, Raul. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* **2017**.

19. Forster, Christian. SVO: Fast semi-direct monocular visual odometry. *IEEE international conference on robotics and automation (ICRA)* **2014**.
20. Karunarathne, M Sajeewani. An adaptive complementary filter for inertial sensor based data fusion to track upper body motion. 7th International Conference on Information and Automation for Sustainability **2014**.
21. Taketomi, Takafumi. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* **2017**, 1–11.
22. Chen, Yalan. Visual SLAM algorithms: A survey from 2010 to 2016. *IEEE International Conference on Information and Automation (ICIA)* **2018**, 603–608.
23. Wei, Wenle. A Survey of UAV Visual Navigation Based on Monocular SLAM. IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC) **2018**.
24. Yang, Nan. Feature-based or direct: An evaluation of monocular visual odometry. arXiv preprint arXiv:1705.04300 **2017**.
25. Sachdev, Kanica. A comprehensive review of feature based methods for drug target interaction prediction. *Journal of biomedical informatics* **2019**.
26. Forster, Christian. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics* **2016**.
27. Zhang, Qian. Application of improved fast dynamic Allan variance for the characterization of MEMS gyroscope on UAV. Journal of Sensors **2018**.
28. Zhu, Xudong. Integrating UAV optical imagery and LiDAR data for assessing the spatial relationship between mangrove and inundation across a subtropical estuarine wetland. ISPRS Journal of Photogrammetry and Remote Sensing **2019**.
29. Cremers, Daniel. Direct methods for 3d reconstruction and visual slam. *Fifteenth IAPR International Conference on Machine Vision Applications (MVA)* **2017**.
30. Lv, Qiang. ORB-SLAM-based tracing and 3D reconstruction for robot using Kinect 2.0. *29th Chinese Control And Decision Conference (CCDC)* **2017**.
31. Nowicki, Micha. Simplicity or flexibility? Complementary Filter vs. EKF for orientation estimation on mobile devices. *IEEE 2nd International Conference on Cybernetics (CYBCONF)* **2015**.
32. Endo, Yuki. Indoor positioning and obstacle detection for visually impaired navigation system based on LSD-SLAM. *International Conference on Biometrics and Kansei Engineering (ICBAKE)* **2017**.
33. Yang, Guanci. Rapid relocation method for mobile robot based on improved ORB-SLAM2 algorithm. *Remote Sensing* **2019**.
34. Mur-Artal, Raul. CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction. transactions on robotics **2015**.
35. Loo, Shing. ORB-SLAM: a versatile and accurate monocular SLAM system. International Conference on Robotics and Automation (ICRA) **2019**.
36. Xiong, Wenyi. Improved FAST corner-detection method. The Journal of Engineering **2019**.
37. Kottath, Rahul. Multiple model adaptive complementary filter for attitude estimation. Aerospace Science and Technology **2017**.
38. Tareen, Shaharyar. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. International conference on computing, mathematics and engineering technologies (iCoMET) **2018**.
39. Aldosari, Waleed. Distributed Extended Kalman Filtering Based Techniques for 3-D UAV Jamming Localization. Sensors **2020**.
40. Tang, Wei. Single neural adaptive PID control for small UAV micro-turbojet engine. Sensors **2020**.
41. Santoso, Fendy. Hybrid pd-fuzzy and pd controllers for trajectory tracking of a quadrotor unmanned aerial vehicle: Autopilot designs and real-time flight tests. IEEE Transactions on Systems, Man, and Cybernetics: Systems **2019**.
42. Rabah, Mohammed. Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-PI. IEEE Access **2019**.
43. Suleiman, Amr. Navion: A Fully Integrated Energy-Efficient Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones. IEEE Symposium on VLSI Circuits **2018**.
44. Ganesan. Indoor versus outdoor scene recognition for navigation of a micro aerial vehicle using spatial color gist wavelet descriptors. Visual Computing for Industry, Biomedicine, and Art **2019**.
45. Zhao, Yanming. Information Fusion Based on Complementary Filter for SINS/CNS/GPS Integrated Navigation System of Aerospace Plane. Sensors **2020**
46. Gao, Xiang. Complete and Accurate Indoor Scene Capturing and Reconstruction Using a Drone and a Robot. IEEE Sensors Journal **2020**
47. Schöps, Thomas. Semi-dense visual odometry for AR on a smartphone. IEEE International Symposium on Mixed and Augmented Reality (ISMAR) **2014**
48. Htun, Zaw Min Min. performance comparison of experimental-based Kalman filter and complementary filter for IMU sensor fusion by applying quadrature encoder. International Journal of Scientific and Research Publications **2018**
49. Mur-Artal, Raul. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE transactions on robotics **2015**
50. Liu, Chang. IT-SVO: Improved Semi-Direct Monocular Visual Odometry Combined with JS Divergence in Restricted Mobile Devices. Sensors **2021**.
51. Amin, Basiri. Enhanced V-SLAM combining SVO and ORB-SLAM2, with reduced computational complexity, to improve autonomous indoor mini-drone navigation under varying conditions. 48th Annual Conference of the Industrial Electronics Society IECON 2022 Conference **2022**.