Novel ORB SLAM System for Robot Navigation in Real-Time

Subhobrata Chakraborty
Department of Computer Science
California State University
Northridge, CA, USA
subhobrata.chakraborty.026@my.csun.edu

Abhishek Verma
Department of Computer Science
California State University
Northridge, CA, USA
abhishek.verma@csun.edu

Amiel Hartman
Department of Mechanical Engineering
California State University
Northridge, CA, USA
amiel.hartman@csun.edu

Nhut Ho
Department of Mechanical Engineering
California State University
Northridge, CA, USA
nhuttho@csun.edu

***Abstract*—Robot navigation is a challenging area of research due to various physical, hardware, and software issues. In this research, an autonomous robot system has been developed, which incorporates a visual inertial SLAM system. Mapping and state estimation rely on the accuracy acquired from fusing the data from the cameras and inertial measurement units (IMU). The fusion of these two sensors makes SLAM systems more accurate and robust. This system is based on the ORB SLAM 3 algorithm, and we have conducted extensive comparison between monocular, monocular-inertial, stereo and stereo-inertial configurations. The results were evaluated on benchmark datasets such EuRoC and TUM.**

***Index Terms*—SLAM, visual inertial, mapping, calibration**

I. Introduction

Extensive exploration into simultaneous localization and mapping frameworks relying on visual data along with visual odometry, leveraging visual feedback either independently or in conjunction with inertial measurement units (IMU), has generated exceptional systems. These systems have demonstrated increased precision and resilience. Contemporary frameworks rely on maximum a posteriori estimation corresponding to bundle adjustment. This also includes geometric bundle adjustment which minimizes the error in reprojection in methods that are based on features whereas in photometric bundle adjustment it reduces the photometric discrepancy of a chosen collection of pixels.

Recently visual odometry algorithms are being incorporated with loop closure methods thereby reducing the boundary between visual odometry (VO) and SLAM systems. Visual SLAM focuses on utilizing the sensors embedded within a robot to reconstruct a 3D map of the environment and simultaneously calculating the position of the robot in real time within the map. Visual odometry focuses on computing the motion of the robot rather than map reconstruction.

The benefit of a SLAM system lies in its capacity to match and utilize previous observations for bundle adjustment [1]. This can be further extended into three categories namely short, mid, and long-term data association. Short-term data association matches map features captured during the most recent time, and it is the one used by most VO algorithms and they forget about features once they are not in the same field of view. This results in more error in trajectory even though the framework is within the desired vicinity. Matching of intermediate-term data association takes care of features that are in close proximity to the camera and the accumulation of the drift remains minimal. These elements can undergo matching and utilization in Bundle Adjustment (BA) similar to short term observations thereby facilitating the achievement of zero drift while traversing mapped regions. These components serve as the cornerstone for heightened precision in contrast to VO algorithms that incorporate loop detection. In the context of long-term data association, the focus is on aligning observations with elements in areas previously explored. This process disregards the accumulated drift and helps in resetting the drift and enables map correction through pose graph optimization.

In this research, we are proposing a system based on ORB SLAM 3 that will be beneficial for autonomous robot navigation in larger environments and utilizes the data association of multiple maps. This permits it to align and utilize map elements of BA fetched from prior sessions of mapping. The paper is structured in the following manner where section II focuses on the related work, section III describes the open-source datasets used in this research, section IV formulates the research methodology and section VI illustrates the experiment results and subsequent discussions. Section VII provides the conclusion to this research and the scope of the future research improvement.

II. Related Work

The inception of Monocular SLAM was initially addressed in MonoSLAM [2, 3, 4] through the utilization of an Extended Kalman Filter (EKF) alongside points of Shi-Tomasi, which were then monitored across subsequent pictures using guided search through correlation techniques. Substantial enhance-

ments in interim data association were achieved using methods ensuring the consistency of feature matches, culminating in the development of mobile visual SLAM systems [5, 6]. In comparison, approaches derived with keyframes as the foundation, conduct map estimation by solely considering some predetermined frames, thereby disregarding data from intervening frames. This approach permits the execution of the more resource-intensive yet more precise Bundle Adjustment (BA) enhancement at the frequency of keyframes. A prominent exemplar of such techniques was PTAM [7], which bifurcated mapping and tracking of camera into two concurrent threads. Methodologies based on keyframes offer superior accuracy compared to filtering at equivalent computational expense [8]. The realization of monocular SLAM on a larger scale was achieved through bundle adjustment relying on sliding-window [9] which was also coupled with double-window [10] enhancement along with a covisibility graph.

Drawing from previous concepts, the ORB SLAM [11, 12, 13] algorithm, harnesses ORB characteristics, where the descriptors facilitate short-range and intermediate-range data matching, while constructing a covisibility graph to streamline tracking and mapping complexity. It achieves relocalization and loop closure by utilizing the DBoW2 [14] library or the bag-of-words library, thus enabling longer-range data matching. ORB SLAM is the sole SLAM system that incorporates data association having all three types because of which the result is exceptionally precise. In this study, we compare the system for monocular-inertial, monocular, stereo-inertial and stereo frameworks and also check the complexity in complex environments. A new map is initiated once encountered with loss of tracking data due to fast movements.

The integration of visual and inertial sensors offers resilience against challenges such as motion blur, lower quality texture as well as occlusions. For systems with only one camera, this combination enables the observation of scale. The exploration of tightly coupled approaches traces its path to MSCKF [16], which circumvents the quadratic cost of EKF in the quantity of characteristics through the marginalization of features. Subsequent refinements and extensions, as seen in [17] and [18, 19], for stereo systems, improved upon this initial framework. Among the pioneering tightly coupled visual odometry systems is OKVIS [20, 21], which utilizes keyframes and bundle adjustment, adaptable to both monocular and stereo vision. In contrast to feature-based methods, ROVIO [22, 23] employs an EKF fed with error in photometry through direct method of data association.

ORB-SLAM 3 unveiled the initial visual-inertial SLAM framework proficient of utilizing long-term, mid-term, and short-term data associations, enabling precise regional visual-inertial BA relying on the preprocessing of IMU [24, 25]. Nonetheless, its IMU initialization process proved time intensive, enduring for 15 seconds, negatively impacting resilience and precision. Swifter initialization methods proposed in [26, 27] provides a comprehensive solution to concurrently ascertain gravity, scale, initial velocity accelerometer bias, and depth of visual features. Earlier studies [28] reveal that this approach can lead to notable and unforeseeable errors.

VINS-Mono [29] stands out as a highly precise and resilient monocular-inertial odometry method, incorporating loop closure techniques. The tracking of feature in VINS-Mono employs a Lucas-Kanade tracker, offering slightly enhanced robustness compared to descriptor matching. VINS-Fusion extends this capability to stereo and stereo-inertial configurations. Visual Inertial-Direct Sparse Odometry [15] expands the capabilities of DSO further into the domain of visual-inertial odometry, introducing a type of bundle adjustment which integrates IMU observations along with the error of photometry of carefully chosen high-gradient pixels, resulting in exceptional accuracy. By effectively leveraging data from high-gradient pixels, the system's resilience in areas with lower quality texture is significantly enhanced. The latest advancement BASALT [30] finalizes loops by pairing ORB characteristics, attaining precision levels varying from satisfactory to outstanding. Conversely, Kimera [31], introduces a groundbreaking metric-semantic mapping framework, which combines stereo-inertial odometry with loop closure via pose-graph and DBoW2 enhancement, delivering precision levels comparable to VINS-Fusion [32].

The idea of enhancing tracking resilience during exploration through map establishment and integration was initially introduced in [33]. A multi-map system based on keyframes was proposed in [34], its manual initialization of maps and incapacity to integrate or correlate distinct sub-maps limited its effectiveness. Research into multi-map capability has been explored within collaborative mapping systems, such as those involving several mapping agents and a central server for data [35] reception, or bidirectional [36] informational exchange systems like C2TAM. While MOARSLAM [37] introduced a durable stateless client-server structure to facilitate collaborative multi-device SLAM, its main emphasis was on software structure rather than presenting accuracy outcomes.

VINS-Mono embodies a visual odometry system equipped with both multi-map and loop closure capabilities, relying on the DBoW2 place recognition library. Through experiments, it is illustrated that ORB-SLAM3 achieves superior precision when compared to VINS-Mono on EuRoC, owing to its capacity to utilize the concept of mid-term data association. The Atlas system in ORB SLAM 3, expanding upon DBoW2, introduces an innovative place recognition method of higher-recall and performs more precise map integration using regional BA, resulting in a 3.2 times higher precision compared to VINS-Mono for operation on multiple sessions on EuRoC.

III. Dataset Description

EuRoC [38] is a collection of data used for research and testing in the field of computer vision and robotics. The dataset is primarily designed to evaluate and benchmark the performance of various methods in the context of micro aerial vehicles and their sensor infrastructure. This dataset contains data from several sensors such as inertial measurement units, stereo cameras, etc. The data is collected using micro aerial vehicles in both indoor and outdoor environments. It is a

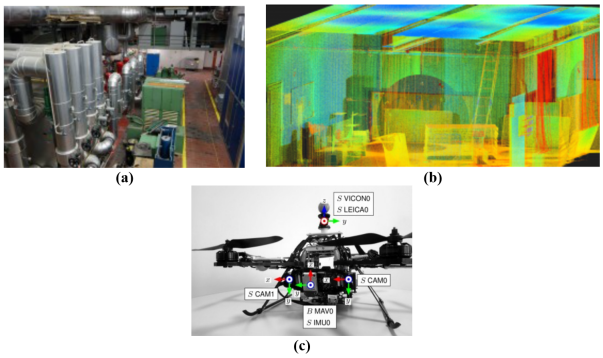


Fig. 1. EuRoC dataset collection and configuration of sensors on the robot for dataset collection. (a) ETH Machine Hall. (b) Ground-truth 3D scan of the video conference room. (c) Hex-copter used for collection of dataset.

Fig. 2. TUM VI dataset for different scenarios for visual inertial odometry benchmark evaluation.

benchmark dataset for conducting experiments in visual odometry, SLAM, sensor fusion and other applications in robot vision. The machine hall environment is depicted in Figure 1 (a), the ground truth 3D scan of the video conference room is depicted in (b) and the sensor infrastructure on the hexcopter is depicted in (c).

Fusing sensor and inertial data improves the precision and robustness in visual inertial odometry algorithms. TUM [39] visual inertial dataset offers a diverse range of sequences captured in different scenarios for evaluation of visual inertial odometry algorithms. It provides with stereo pair of images with a resolution of 1024 X 1024. The images are captured at 20 frames per second with HDR and photometric calibration. The IMU calculates the acceleration and angular velocities at 200 frames per second along 3 axes. The camera data and the IMU data are time synchronized within the hardware. To evaluate the accuracy of the trajectory, they also provide the ground truth data that is captured with a motion capture system at a frequency of 120 Hz. It also provides RGB-D data along with the ground truth. It contains the color and the depth images recorded at a frame rate of 30 Hz and a sensor resolution of 640 X 480. The accelerometer data was collected from the Kinect sensor. The frame references of the different datasets in TUM are illustrated in Figure 2.

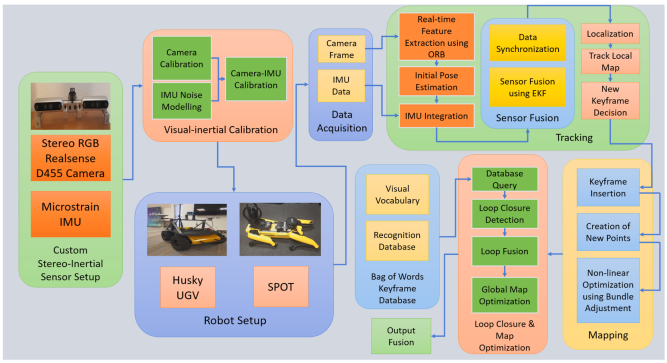


Fig. 3. Proposed system pipeline for visual-inertial SLAM.

IV. Research Methodology

An important computer vision problem is to be able to recover 3-D information of a scene from its image. Usually there is a scene which is defined in some world coordinate frame and at the end of the day when the scene is reconstructed it is important to know where each point lies in the world coordinate frame. Images of the scene are at our disposal where points are measured in pixels. In this research a system pipeline is proposed for the visual inertial SLAM and the components of the system can be segregated as the following: Custom Stereo-Inertial Sensor Configuration: 2 Intel Realsense D455 Cameras, Microstrain IMU. The above sensor configuration was calibrated independently and then mounted on the robots. It can be mounted on any robot and the mapping of the environment can be done. The sensors need to be taken off the robot to do proper calibration. It is essential to excite all the axes during the calibration process which is not possible while mounted on a robot. Robot Configuration: Husky UGV, SPOT.

The above robot setup is used for conducting the experiments. The custom stereo-inertial sensor was mounted on these robots to conduct the experiments. The proposed visual inertial SLAM system used for this research is depicted in Figure 3. Sensor Calibration: This step includes the calibration of the cameras and the IMU. Camera Calibration: It determines the intrinsic and the extrinsic parameters of the camera that includes focal length, optical centre, and lens distortions. This is essential for accurate pixel mapping from 3D world coordinates to 2D image coordinates. IMU Calibration: The IMU is calibrated to account for the sensor noise and the accelerometer and gyroscope bias. This helps to obtain more accurate measurements of accelerations and angular velocities. Data Acquisition: The camera data and the IMU data are acquired from the respective sensors.

Feature Extraction using S: Distinctive features are extracted from the camera images such as keypoints and landmarks. These features are tracked across subsequent frames to establish correspondences that helps in estimating the motion of the camera. Initial Pose Estimation: The tracked features are used to estimate the initial pose of the camera by relating

the camera poses between the frames. IMU Integration: The measurements from the IMU are integrated to estimate the linear and angular velocities of the system. It also involves double integration of accelerometer measurements and integration of gyroscope measurements for orientation changes. Visual Odometry: The tracked features and the estimated camera poses are used to compute incremental transformations between the frames. Data Synchronization: It is ensured that the data from the camera and the IMU are synchronized in time and there is minimum latency between the two. This is also taken care of during the calibration process.

Sensor Fusion using EKF: The estimates from the visual odometry and IMU integration are combined to obtain a more accurate state estimation of the robot. Keyframe Insertion: The new keyframes are inserted into the mapping process once the bad ones are rejected and the good ones are accepted. For the subsequent frames, new keypoints are created. Nonlinear Optimization using Bundle Adjustment: The problem is formulated as a nonlinear optimization task (often a least squares problem) which is used to refine the estimates of camera poses and feature locations. It optimizes the entire trajectory while considering the visual and inertial trajectories in the process.

Database Query: Query is sent to the visual vocabulary database that consists of visual features from the images. It was created using the bag of words approach. Queries from the present image are sent to the database to verify the similarity score between the current image and any of the previous frames. Loop Detection: During the verification of the similarity score, it is possible to determine if there is any loop in the mapping process. Loop Fusion: Once the loop is detected, the loop is corrected in the mapping process thereby correcting the drift in the mapping. This is essential for correcting the accumulated errors in the odometry estimation. Global Map Optimization: The entire map is optimized and refined, and the accumulated drift is corrected further. It involves loop closure constraints thereby improving the consistency of the estimated trajectory.

Output Fusion: The final estimations from visual odometry, inertial integration and loop closure are combined to obtain a consistent trajectory and map of the entire system. In order to determine the accuracy of the performance of the algorithms, an evaluation system pipeline was also designed for the evaluation of the visual inertial SLAM system. The primary datasets used for this evaluation are EuRoC and TUM VI and the ground-truth data was collected from the respective dataset sources. The evaluation was performed on the ORB SLAM 3 based visual inertial SLAM systems. The ground truth data and the estimated trajectory data is in the following format: Timestamps, Position of the robot in 3D space (x, y, z), Quaternion Rotation of the Robot (q_x, q_y, q_z, q_w).

Figure 4 illustrates the evaluation system pipeline designed to evaluate the visual-inertial SLAM system. The timestamps, position and the rotation of the robot is not aligned between the ground truth and the estimated trajectory because of which they need to be interpolated first and aligned. If the two trajectories are superimposed it will be noticed that are

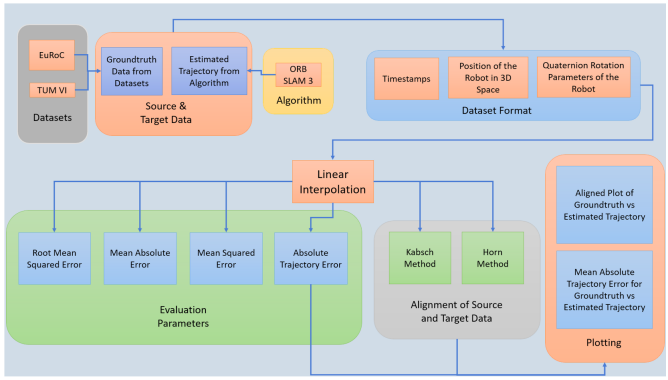


Fig. 4. Evaluation system pipeline for the visual-inertial SLAM system.

completely misaligned. The two data are first interpolated using linear interpolation and then aligned using the Horn method or Kabsch method. Both methods are used for aligning the data. The interpolated data is then used to compute the following evaluation parameters which are used to verify the accuracy of the algorithms: root mean squared error, mean absolute error, mean squared error, absolute trajectory error. The final step involves the alignment of the estimated and the ground truth data.

V. Experiment Environment and Setup

A. Hardware Environment

The hardware setup is equipped with high performance GPU such as NVIDIA RTX 3080 along with the Ryzen 9 5900 HX CPU which consists of 8 cores and 16 threads. GPU VRAM is of 16 GB whereas the RAM was 32 GB. The sensor infrastructure consists of two intel realsense D455 cameras where each of the individual RGB module from each camera were paired together to get a stereo configuration. We used the integrated IMU within the realsense camera for the initial calibration purpose and later switched with the Microstrain IMU. This sensor infrastructure was mounted on the Husky and SPOT robots for testing the ORB SLAM 3 based visual-inertial system.

B. Software Environment

The codebase is developed primarily with C++ and Python. The primary library used was OpenCV for implementation of the computer vision algorithms. Matplotlib was used to create line plots, bar plots, scatter plots, histograms, etc. It was also used in combination with other libraries such as numpy and pandas. We used Robot Operating System (ROS) which is an open-source framework providing tools and services for hardware abstraction, communication between device drivers, etc. We used ROS Melodic and ROS Noetic for our experiments. ROS Melodic operates on Python 2.7 whereas ROS Noetic is based on Python 3. The primary visualization tool used was rviz along with pangolin which is also a flexible framework for visualization, 3D reconstruction, etc. The open source Kalibr [40] tool was used to calibrate the cameras and the IMU. The IMU noise and bias was calculated

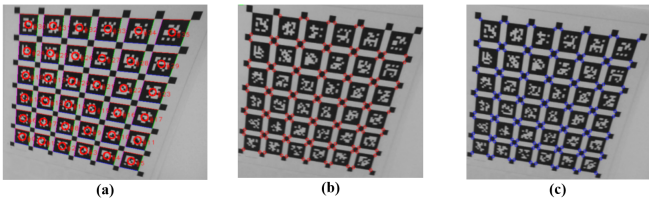


Fig. 5. AprilTag detection and corner detection for the calibration process using the RealSense D455 camera. (a) AprilTag detection using RealSense D455. (b) Reprojection error. Mean is 0.295. Std. Dev. is 0.183. (c) Corner detection of the AprilTag.

using the Allan variance ros package which is also compatible with the Kalibr framework.

VI. Experiment Results and Discussion

The depth parameter of the RealSense camera was set to true with a resolution of 840 x 480. The gyroscope and accelerometer frames per second (FPS) were set to 400 and 250. The calibration was done for both monocular and stereo configuration. The custom stereo inertial system was used for the stereo calibration process. The AprilTag used for the calibration process had the following configuration: size of each tag is 0.088 m, number of AprilTags in each row and column is six, space between each tag is 0.3 m.

A. Monocular Calibration

For the monocular calibration, the integrated IMU of the RealSense D455 camera was used, which published the data at 400 FPS and the rosbag color image data was recorded at 30 FPS. The monocular RGB color module was utilized and the pinhole camera model was used for the calibration. Figure 5 illustrates the calibration of the monocular camera configuration along with the reprojection error. Fig. 5 also illustrates the April Tag detection along with the corner detection, which helps in identifying the intrinsic and extrinsic attributes of the camera along with the coefficients of distortion. However, since the pinhole camera is used for the experiments, distortion coefficients are almost equivalent to zero. Figure 6 (a) shows the estimated poses of the monocular camera configuration and the reprojection error along the x and y axes is represented in (b). It also represents the coverage area of the camera while

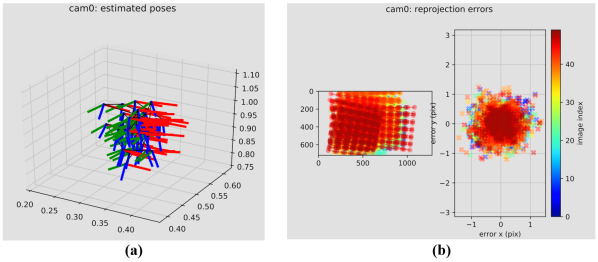


Fig. 6. Estimated camera poses of the monocular camera while capturing calibration data along with the resultant reprojection errors after calibration. (a) Estimated poses of the monocular camera configuration. (b) Camera coverage along the AprilTag and the reprojection error along x and y axis.

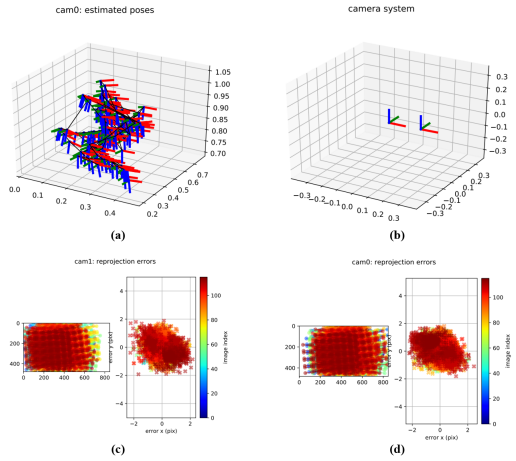


Fig. 7. Stereo camera system reprojection error and estimated poses. (a) Stereo camera configuration. (b) Estimated poses of the stereo system. (c) Reprojection error of first camera. (d) Reprojection error of second camera.

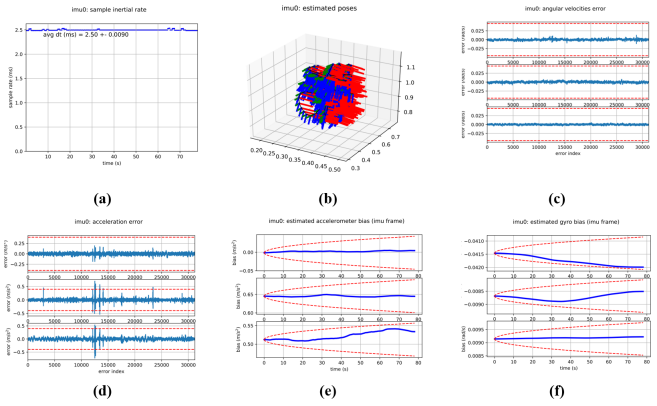


Fig. 8. IMU estimated poses along with the sample inertial rate of the IMU demonstrating steady flow of data with the corresponding accelerometer and gyroscope errors and biases. (a) Estimated poses for the integrated IMU in the RealSense camera. (b) Sample inertial rate of the IMU. (c) Accelerometer error within the 3σ bound (red dashed line). (d) Accelerometer bias within the 3σ bound (red dashed curve). (e) Gyroscope error within the 3σ bound (red dashed line). (f) Gyroscope bias within the 3σ bound (red dashed curve).

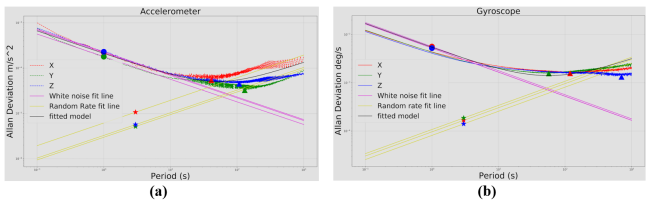


Fig. 9. Allan standard deviation of accelerometer and gyroscope with manually identified noise processes. (a) Allan standard deviation of accelerometer. (b) Allan standard deviation of gyroscope.

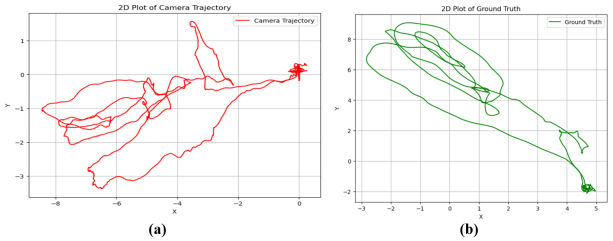


Fig. 10. 2D plot of the camera trajectory of the EuRoC Machine Hall 01 dataset. (a) Estimated camera trajectory. (b) Ground truth trajectory.

collecting the calibration data. Once the monocular camera is calibrated separately, the IMU needs to be calibrated as well. A 22 hour 59 seconds long static IMU data was collected to estimate the transformation between the camera and the integrated IMU of the RealSense camera. The gyroscope and the accelerometer bias of the IMU was calculated using the Allan variance ROS package. Figure 8 (a) represents the estimated poses of the IMU while collecting the rosbag data. The same rosbag data was used for both the camera and camera IMU calibration. Figure 8 (b) represents the sample inertial rate of the IMU, which is a measure of the angular velocity of the IMU at a specific sampling rate. During calibration, the IMU is often subjected to static positions and the inertial rate measurements are collected at regular intervals. These measurements help in characterizing the performance by compensating for the errors and biases. The accelerometer and gyroscope errors and biases are represented in (c), (d), (e), and (f) respectively. The fact that all of these are within the 3σ bound, which is marked with the red-dashed line suggest accurate calibration results. Figure 9 (a) and (b) represents the Allan standard deviation of the accelerometer and gyroscope respectively.

B. Stereo Calibration

To calibrate the stereo pair of cameras, the same intel RealSense D455 cameras were used but an external MicroStrain IMU was used for the visual inertial system calibration. The same calibration process was followed as done for the monocular calibration, but both cameras had to be calibrated separately. One camera was considered to be the global camera coordinate frame and the transformation of the other camera was determined with respect to the first camera. The IMU transformations were determined with respect to both cameras to determine the exact orientation and position of the IMU in the sensor configuration. This is very important for accurate state estimation. The IMU data was published at 100 Hz whereas the stereo camera pair published the color image data at 30 frames per second (FPS). The scale misalignment model was used for calibrating the IMU for both the stereo and monocular sensor configurations. Figure 7 (a) shows the stereo camera sensor configuration along with the estimated poses of the stereo system in Fig. 7 (b), while the reprojection errors of the first and second cameras are represented in Figure 7 (c) and Figure 7 (d) respectively.

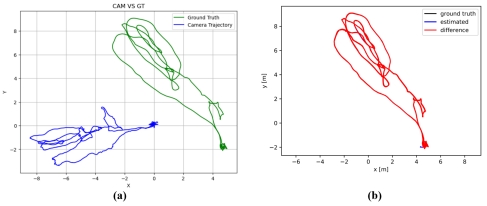


Fig. 11. Unaligned and aligned data of the EuRoC Machine Hall 01 dataset. (a) Unaligned data of the ground truth and estimated trajectory. (b) Aligned data of the ground truth and estimated trajectory using the Horn method.

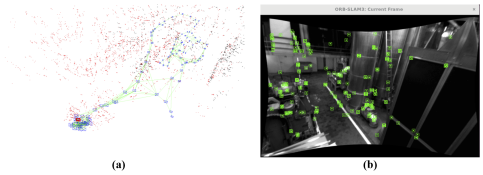


Fig. 12. Sparse map generation and keyframe generation of the EuRoC Machine Hall 01 dataset. (a) Sparse map for Machine Hall 01 dataset. (b) Keyframe and feature detection of the current frame.

C. Stereo Inertial Evaluation on EuRoC Machine Hall 01 Dataset

The visual-inertial system based on ORB SLAM 3 was executed on the EuRoC Machine Hall 01 dataset. The estimated and the ground truth trajectories were plotted against each other to compare the accuracy. The estimated trajectory axis was flipped to align itself to the ground truth data. Figure 10 (a) shows the estimated trajectory and Figure 10 (b) shows the ground truth trajectory of the Machine Hall 01 dataset. Subsequently the ground truth trajectory and the estimated trajectory were aligned using the Horn method and the different evaluation parameters were determined to evaluate the accuracy of the algorithm. Figure 11 (a) shows the unaligned trajectory and Figure 11 (b) shows the aligned trajectory of the ground truth and estimated trajectory of the Machine Hall 01 dataset using the Horn method. The sparse map of the environment was also generated which is represented in Figure 12 (a). Figure 12 (b) represents the keyframe and feature detection of the current frame while generating the sparse map of the environment. Following are the evaluation parameters used to determine the accuracy of the system: Root Mean Squared Error: 0.021876 m, Mean: 0.019318 m, Median: 0.017096 m, Standard Deviation: 0.010266 m, Min: 0.001405 m, Max: 0.129538 m, Max indexes: 1427, Compared pose pairs: 3638

D. Stereo Evaluation on EuRoC Machine Hall 01 Dataset

A similar set of experiments were conducted on the EuRoC Machine Hall 01 dataset using just the stereo pair of cameras and no IMU was used for this experiment. Following are the evaluation parameters used to determine the accuracy of the algorithm: Root Mean Squared Error: 0.034710 m, Mean: 0.025710 m, Median: 0.021142 m, Standard Deviation: 0.023319 m, Min: 0.001223 m, Max: 0.119464 m, Max indexes: 1594, Compared pose pairs: 3638.

TABLE I
COMPARISON BETWEEN STEREO AND STEREO-INERTIAL CONFIGURATION IN METERS

	Stereo	Stereo-Inertial
RMSE	0.034710	0.021876
Mean	0.025710	0.019318
Median	0.021142	0.017096
Standard Deviation	0.023319	0.010266

TABLE II
COMPARISON BETWEEN MONOCULAR AND MONOCULAR-INERTIAL CONFIGURATION IN METERS

	Monocular	Monocular-Inertial
RMSE	2.582047	0.109435
Mean	2.376143	0.093516
Median	2.659100	0.081721
Standard Deviation	1.010402	0.056841

The comparison between the evaluation parameters of stereo and stereo-inertial configuration is depicted in Table I.

E. Monocular Inertial Evaluation on EuRoC Machine Hall 01 Dataset

One camera and one IMU were used to perform this experiment. Following are the evaluation parameters used to determine the accuracy of the algorithm: Root Mean Squared Error: 0.109435 m, Mean: 0.093516 m, Median: 0.081721 m, Standard Deviation: 0.056841 m, Min: 0.003959 m, Max: 0.628301 m, Compared pose pairs: 2729

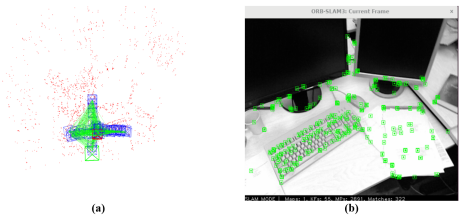
F. Monocular Evaluation on EuRoC Machine Hall 01 Dataset

One camera was used to perform this experiment because this evaluation was based on the monocular configuration. Following are the evaluation parameters used to determine the accuracy of the algorithm: Root Mean Squared Error: 2.582047 m, Mean: 2.376143 m, Median: 2.659100 m, Standard Deviation: 1.010402 m, Min: 0.551272 m, Max: 4.491108 m, Compared pose pairs: 3638.

The comparison between the evaluation parameters of monocular and monocular-inertial configuration is depicted in Table II.

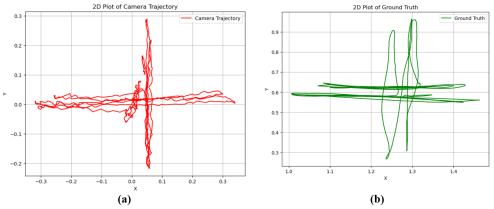
G. Evaluation on TUM RGBD Dataset

We also conducted experiments on RGB and depth datasets. While considering rgbd datasets associations had to be created between the rgb and the depth data. In the rgb and depth text



(a) (b)

Fig. 13. Sparse map and keyframe detection of TUM RGBD Freigburg 1 xyz dataset. (a) Sparse map generation. (b) Keyframe and feature detection.



(a) (b)

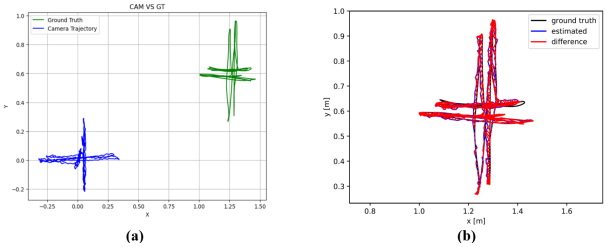
Fig. 14. Estimated camera trajectory and ground truth trajectory of TUM RGBD Freigburg 1 xyz dataset. (a) Estimated camera trajectory. (b) Ground truth data.

files there was information about the timestamps, translation, and quaternion rotation data. Finally, an association was created between the two where the timestamps were mapped between both the rgb and depth data along with the images. Once that was done, the algorithm was executed on the rgbd dataset. Unlike RGB datasets RGBD datasets provide depth information which helps in more accurate state estimation and creation of map of the environment. It also assists in feature detection and matching across frames leading to more robust feature correspondences. Depth data also helps in accurate 3D reconstruction of the environment as well as obstacle avoidance.

Figure 13 (a) illustrates the sparse map generated for the the TUM VI Freigburg 1 xyz dataset along with the current frame illustrating the feature detection in Figure 13 (b) while mapping the environment. Subsequently, we also estimated the trajectory of the camera and compared it with the ground truth of the dataset. Figure 14 (a) illustrates the estimated trajectory of the freigburg 1 xyz dataset along with the ground truth data represented in Figure 14 (b). The ground truth data and the camera trajectory were aligned using the Horn method which is represented in Figure 15 (b). Figure 15 (a) shows the unaligned data of the camera trajectory and the ground truth data while fig shows the aligned data of the camera trajectory and the ground truth data.

H. RGBD Evaluation of Freigburg 1 xyz Dataset

Root Mean Squared Error: 0.015258 m, Mean: 0.013320 m, Median: 0.011903 m, Standard Deviation: 0.007442 m, Min: 0.001529 m, Max: 0.048537 m, Compared pose pairs: 792.



(a) (b)

Fig. 15. Unaligned and aligned data of the Freigburg 1 xyz dataset using the Horn method. (a) Unaligned data of the camera trajectory and the ground truth. (b) Aligned data of the estimated trajectory and the ground truth.

VII. Conclusion and Future Work

In this research, we introduced a visual inertial SLAM system based on ORB SLAM 3 with a goal to perform autonomous robot navigation. We compared the performance of the system with open-source datasets such as EuRoC and TUM VI. We evaluated the performance between stereo and stereo-inertial systems as well as monocular and monocular-inertial systems for the EuRoC datasets. We also evaluated the performance of the algorithm on RGBD datasets on the freigburg 1 xyz dataset. The results were significantly better when switching from monocular to stereo system. We also used our custom stereo inertial setup to test the SLAM system and calibrated the stereo setup along with the IMU and evaluated the performance in real world scenario. The system has performed significantly well in terms of accuracy when tested on open-source datasets. There is scope of improvement when operating in real-time where the localization is lost when moving too fast in an environment.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, et. al, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. on Robotics*, vol. 32, no. 6, pp. 1309-1332, 2016.

[2] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," *Proc. of 9th IEEE Int. Conf. on Computer Vision*, pp. 1403-1410, 2003.

[3] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, 2007.

[4] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 932-945, 2008.

[5] J. Civera, O. G. Grasa, A. J. Davison, and J. M. Montiel, "1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *J. of Field Robotics*, vol. 27, no. 5, pp. 609-631, 2010.

[6] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, et. al, "Mapping Large Loops with a Single Hand-Held Camera," *Robotics: Science and Systems*, vol. 2, no. 2. 2007.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pp. 225-234, 2007.

[8] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual SLAM: why filter?," *Image and Vision Computing*, vol. 30, no. 2, pp. 65-77, 2012.

[9] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," *Robotics: Science and Systems*, vol. 2, no. 3, 2010.

[10] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," *Int. Conf. on Computer Vision*, pp. 2352-2359, 2011.

[11] C. Campos, R. Elvira, J. J. Rodríguez, J. M. Montiel, et. al, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Trans. on Robotics*, vol. 37, no. 6, pp. 1874-1890, 2021.

[12] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.

[13] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. on Robotics*, vol. 33, no. 5, pp. 1255-1262, 2017.

[14] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1188-1197, 2012.

[15] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," *IEEE Int. Conf. on Robotics and Automation*, pp. 2510-2517, 2018.

[16] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3565-3572, 2007.

[17] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. of Robotics Research* 32, no. 6 (2013): 690-711.

[18] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, et. al, "A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS," *IEEE Int. Conf. on Robotics and Automation*, pp. 165-172, 2017.

[19] M. K. Paul and S. I. Roumeliotis, "Alternating-stereo VINS: Observability analysis and performance evaluation," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4729-4737. 2018.

[20] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, et. al, "Keyframe-based visual-inertial slam using nonlinear optimization," *Proc. of Robotic Science and Systems*, 2013.

[21] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, et. al, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. of Robotics Research* 34, no. 3 (2015): 314-334.

[22] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 298-304, 2015.

[23] M. Bloesch, M. Burri, S. Omari, M. Hutter, et. al, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. of Robotics Research* 36, no. 10 (2017): 1053-1072.

[24] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. on Robotics* 33, no. 1 (2016): 1-21.

[25] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. on Robotics* 28, no. 1 (2011): 61-76.

[26] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robotics and Automation Letters* 2, no. 1 (2016): 18-25.

[27] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *Int. J. of Computer Vision* 106, no. 2 (2014): 138-152.

[28] C. Campos, J. M. Montiel, and J. D. Tardós, "Fast and robust initialization for visual-inertial SLAM," *IEEE Int. Conf. on Robotics and Automation*, pp. 1288-1294, 2019.

[29] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. on Robotics* 34, no. 4 (2018): 1004-1020.

[30] V. Usenko, N. Demmel, D. Schubert, J. Stückler, et. al, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters* 5, no. 2 (2019): 422-429.

[31] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," *IEEE Int. Conf. on Robotics and Automation*, pp. 1689-1696, 2020.

[32] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," arXiv preprint arXiv:1901.03638 (2019).

[33] E. Eade and T. Drummond, "Unified loop closing and recovery for real time monocular SLAM," *British Machine Vision Conf.*, vol. 13, p. 136. 2008.

[34] R. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," *IEEE Int. Symposium on Wearable Computers*, pp. 15-22, 2008.

[35] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3962-3970, 2013.

[36] L. Riazuelo, J. Civera, and J. M. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems* 62, no. 4 (2014): 401-413.

[37] J. G. Morrison, D. Gálvez-López, and G. Sibley, "MOARSLAM: Multiple operator augmented RSLAM," *Int. Symp. on Distributed Autonomous Robotic Systems*, pp. 119-132, Japan, 2016.

[38] M. J. N. P. G. Burri, T. Schneider, J. Rehder, S. Omari, et. al, "The EuRoC micro aerial vehicle datasets," *Int. J. of Robotics Research* 35, no. 10 (2016): 1157-1163.

[39] D. Schubert, T. Goll, N. Demmel, V. Usenko, et. al, "The TUM VI benchmark for evaluating visual-inertial odometry," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1680-1687, 2018.

[40] "Kalibr," in https://github.com/ethz-asl/kalibr.