Vision Navigation - Visual Inertial Tracking using Preintegrated Factor

Pei-Ran Huang, Wenjie Xie

August 31, 2024

Abstract

In this work, we implement the visual inertial odometry(VIO) with preintegrated factor. An inertial measurement unit(IMU) provides angular velocity through a gyroscope and acceleration through an accelerometer. Compared to solely a camera with visual odometry(VO), integrating with IMU measurement can make the prediction more robust and accurate. We follow the process of preintegration and finish the strong coupling between IMU and the camera. In summary, the VIO algorithm performs more accurately than the VO algorithm in our experiment.

Code available in the Gitlab. The link :Implementation of Visual Inertial Odometry in Gitlab

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental technology in robotics and autonomous systems[1]. It enables a robot or a device to create a map of an unknown environment while simultaneously determining its position within that map. It is critical when the prior surrounding information is unavailable because it allows the visualization of landmarks and facilitates the environment's visualization. Besides, it is beneficial to estimate the current state of the agent. With the development of visual SLAM, scientists have significantly improved the SLAM technique to be more accurate and robust. It can be widely employed in various applications, including robotics, autonomous vehicle, and drone navigation.

Visual odometry (VO) is a visual SLAM algorithm that estimates the pose and motion of a camera from an image sequence [2]. It involves using one or more cameras to track feature points across successive image frames, allowing for the calculation of the camera's relative motion. This method analyzes frame-to-frame changes to determine the camera's movement. VO is a cost-effective approach to SLAM as it relies solely on cameras, making it more affordable compared to other sensors like LiDAR or radar.

Additionally, VO can capture 3D geometry and texture. However, there are limitations. A monocular camera can only reconstruct a 3D scene up to an unknown scale factor, making it unable to determine absolute distances without additional information. Stereo cameras can reconstruct spatial geometry but are highly dependent on the baseline, leading to less accurate depth estimations if the baseline is too narrow compared to the distance to the observed scene. Furthermore, the precision of depth information degrades with increasing distance. These methods are also sensitive to environmental conditions such as motion blur and changes in lighting.

In order to overcome these limitations, we propose an alternative visual SLAM algorithm known as visual inertial odometry (VIO) [3]. Essentially, VIO combines visual odometry (VO) with inertial data. In addition to the camera, we incorporate an inertial measurement unit (IMU) sensor, which collects inertial measurements such as acceleration and angular velocity using accelerometers and gyroscopes. The integration of the IMU sensor helps address the challenges associated with VO by continuously updating the camera's movement, thus reducing drift and improving accuracy. Moreover, the acceleration data can be utilized to more accurately estimate the scale for 3D reconstruction.

Furthermore, the IMU data can aid in system recovery in the event of temporary visual tracking failures. Additionally, IMU sensors, like cameras, are lightweight and cost-effective. Through the integration of visual and inertial measurements, VIO effectively compensates for the weaknesses of each individual sensor, resulting in more reliable and accurate estimates of position and orientation over time.

In this study, we developed a VIO within the framework of our practical course. We adopted a strong coupling approach and integrated two types of sensor data to implement the VIO algorithm. By preintegrating the IMU measurement and combining the energy functions of the sensors with a hyperparameter, we derived the final residual function. To assess its performance, we compared our VIO method with the original VO method using average trajectory error. Our findings indicate that VIO achieves impressive results compared to VO.

2 Related Work

Visual Inertial Odometry (VIO) is an essential research area in the field of Simultaneous Localization and Mapping (SLAM) that has experienced significant advancements over the last few decades. The primary goal of VIO is to determine the position and path of a moving platform by integrating visual data from cameras with inertial measurements from IMUs.

VIO methods are generally classified based on the number of camera poses involved in the state estimation process and the sensor fusion strategy utilized. These classifications include filter-based methods, fixed-lag smoothing methods, and full smoothing methods. Each of these approaches presents distinct trade-offs in terms of computational complexity, accuracy, and applicability for real-time use.

In filter-based methods such as those discussed in Mourikis et al.[4] and Bloesch et al. (2015)[5], the prediction step typically involves updating the current camera state estimate using IMU measurements. These methods utilize an extended Kalman filter (EKF) or its variants to maintain an ongoing estimate of the current camera pose and velocity. This predicted state is then continually refined using new information from the camera images. One notable drawback of these filters is their inability to adjust the linearization point for non-linear measurement and state transition models once a measurement has been integrated. This scenario results in the inaccurate prediction.

The fixed-lag smoothing methods, on the other hand, focus on estimating states within a specific time window, while older states are marginalized[3][6]. They generally outperform filter-based methods because they can relinearize past measurements within fixed window. Moreover, fixed-lag smoothers are more robust to outliers, as they can be removed after the optimization process or mitigated using robust cost functions. However, marginalizing states outside the estimation window leads to dense Gaussian priors, which complicates the process.

Full smoothing is an approach to estimate the entire sequence of the state when solving non-linear problem[7][8]. It considers all available measurements simultaneously into a single optimization process. It can globally relinearize all measurement to predict the state more accurate and improve the robustness when introducing non-linearity problem. While full smoothing is more accurate, it becomes impractical for real-time applications as the trajectory and map expand over time.

These methods for estimating the state can be divided into two coupling groups. Coupling is a method, which aims to consolidate sensor data from different sources. The first approach is strong coupling, where the measurements from both the camera and IMU are directly integrated into a single optimization problem. Bloesch et al. [5] introduced the strong coupling method with a filter-based approach. Conversely, weak coupling approaches process visual and inertial measurements separately and fuse their outputs at a later stage, which can simplify the implementation and reduce the computational cost. Sirtlaya et al. [9] present filter-based methods with weak coupling, which allows for more flexibility in handling sensor specific noise and failure modes.

For this project, we have implemented the preintegration theory using IMU measurements within the manifold structure of the rotation group SO(3), as proposed by Forster et al[10]. This approach effectively overcomes the limitations associated with filter-based and smoothing methods discussed earlier, resulting in significant improvements. Additionally, we have utilized Basalt, an open-source and well-documented repository, to facilitate non-linear optimization and IMU measurement preintegration. The functionalities provided by Basalt have played a crucial role in enabling the implementation of the Visual-Inertial Odometry (VIO) method for this project. In summary, the adoption of preintegration theory and the use of the Basalt repository have been key factors contributing to the successful execution of this work.

3 Methodology

Our framework is built upon the final task of a previous practical exercise, where we implemented the partly component of a Visual Odometry algorithm(VO). Specifically, the task involved detecting and matching feature points between two images captured by stereo cameras mounted on a robot. We adopted an approach similar to ORB-SLAM, utilizing the Oriented FAST and Rotated BRIEF(ORB) method[11] to detect keypoints and generate descriptors, and then matching them based on the Hamming distance.[12] Matches are further distilled using the epipolar constraint and RANSAC[13]

The map initialization begins with an initial pair of cameras, where the observed features are triangulated using the matched descriptors between the two cameras. These triangulated features are then added to the map as 3D landmarks (map points). Following this initialization, we iteratively incorporate new camera pairs and their corresponding landmarks using the PnP algorithm within a RANSAC framework. Because of the fixed relative position between the cameras and the robot's body, we could then obtain the robot's location within the world frame.

To enhance the accuracy of both the 3D landmarks (mapping) and the localization, we integrated Bundle Adjustment (BA) into the framework. This optimizes the 6DoF poses of all cameras and the 3DoF positions of all landmarks.

Although the visual odometry demonstrated satisfactory results on the simple dataset used during the exercises, it is important to acknowledge its limitations. Therefore, the objective of this project is to improve the framework of the existing visual odometry baseline by extending it to a Visual-Inertial Odometry (VIO) system through the integration of IMU measurements.

3.1 IMU Model Introduction

With IMU can we measure the acceleration and rotation rate of our robot. For following measurement for IMU ignore we terms of white noise and slowly varying bias, because of the limited time of this project. We would like to adopt model as easy as possible. From this paper[10] can we have the basic mathematical measurement model for IMU. We can update our next state $t+\Delta t$ as following equation (1). The Terms $\mathbf{R}(t)$, $\mathbf{v}(t)$, and $\mathbf{p}(t)$ presents rotation, velocity and position for robot at time step t observed in the world frame. Rotation $\mathbf{R}(t)$ can also be looked as the rotation from body frame to world frame. Here we assume $\mathbf{R}(t)$ is constant in the interval $[t, t+\Delta t]$, as this Δt is very small

$$\begin{aligned}
\mathbf{R}(t+\Delta t) &= \mathbf{R}(t)\mathrm{Exp}(\omega(t)\Delta t) \\
\mathbf{v}(t+\Delta t) &= \mathbf{v}(t) + \mathbf{g}\Delta t + \mathbf{R}(t)\mathbf{a}(t)\Delta t \\
\mathbf{p}(t+\Delta t) &= \mathbf{p}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 + \frac{1}{2}\mathbf{R}(t)\mathbf{a}(t)\Delta t^2
\end{aligned} \tag{1}$$

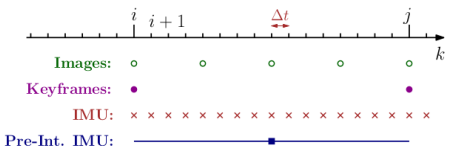

Figure 1: Different rates for IMU and camera.(cite from [10])

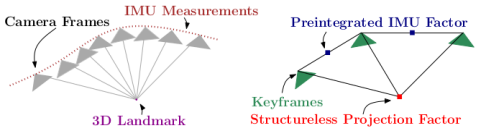

Figure 2: Left: visual and inertial measurements in VIO. Right: factor graph in which several IMU measurements are summarized in a single preintegrated IMU factor and a structureless vision factor constraints keyframes observing the same landmark.(cite from [10])

3.2 IMU Preintegration

Because of difference of Update Frequency between camera's frame and IMU's measurement, generally the frequency for frames is higher than one for measurement.4 It enables us to integrate several consecutive measurement from IMU between last frames and current frames in interval $[t, t+\Delta t]$.5 From these preintegration could we have overall delta state $\mathbf{V}(\Delta t)$, $\mathbf{R}(\Delta t)$ and $\mathbf{P}(\Delta t)$, and then with

these measurement update we current state from the previous one based on equation (1). What's more, these states estimated from preintegration can be used for building cost function and further optimization in the BA processing. The built constraint function is as (2):

$$\begin{aligned} \mathbf{r}_{\Delta \mathbf{R}} &= \operatorname{Log}(\Delta \mathbf{R} \mathbf{R}_j^\top \mathbf{R}_i) \\ \mathbf{r}_{\Delta \mathbf{v}} &= \mathbf{R}_i^\top (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t) - \Delta \mathbf{v} \\ \mathbf{r}_{\Delta \mathbf{p}} &= \mathbf{R}_i^\top \left(\mathbf{p}_j - \mathbf{p}_i - \frac{1}{2}\mathbf{g}\Delta t^2 \right) - \Delta \mathbf{p} \end{aligned} \tag{2}$$

Figure 3 illustrates the factor graph of IMU preintegration and without preintegration between frames, where we compute the preintegrated factor using all the IMU measurements collected between consecutive frames. This preintegrated factor will later act as a constraint during the optimization process. Fixed windows and Local windows respectively presents those keyframes during Local mapping, which is contributing to the total cost but fixed or not in the optimization. In the specific implement, We take 10 keyframes for one iteration and essentially utilize the framework developed in Basalt [14], but omit all bias terms when computing the delta state and calculating the preintegrated factor, which is intended to be minimized.

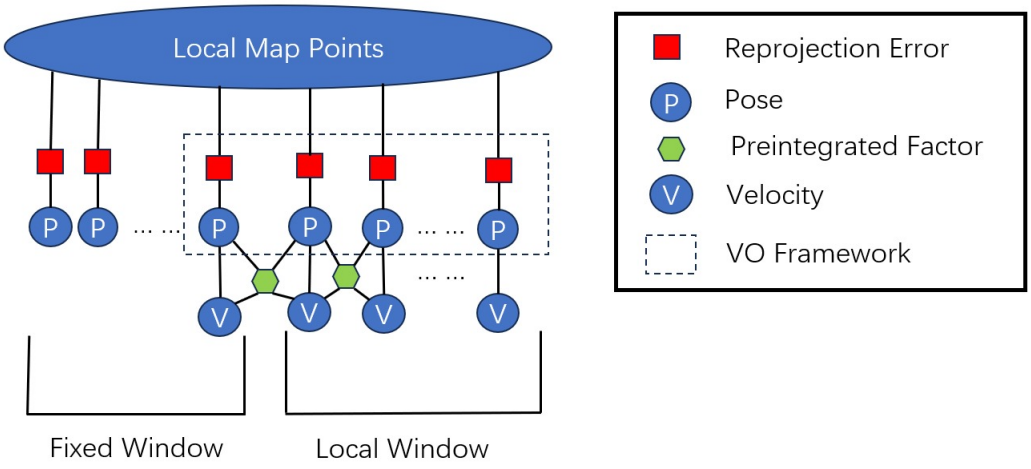

Figure 3: The area in the dashed square box is the original VO framework. Note that the red boxes and green hexagons are the residual terms that we try to minimize while optimizing all the blue bubbles.

3.3 Computation Cost Trade-Off

In addition to performing standard bundle adjustment (BA) on 10 keyframes to minimize reprojection errors, we also actively optimize the residuals from the IMU preintegration. As in VIO framework we introduce additional cost function for consecutive keyframes, it must increase accordingly the computation cost during optimization. We have to apply some adaption to keep algorithm real-time performance. First, we limit the optimization problem to keyframes, which are selected based on specific criteria, rather than including all frames. Additionally, we employ a sliding window approach to further reduce the number of parameters in the optimization and to maintain a relatively constant problem size over time. Older keyframes and their corresponding observed 3D landmarks are marginalized, meaning they are removed from the optimization in the Bundle Adjustment process. According to the finding in this paper[15], the longer the time interval between consecutive keyframes, the less informative the IMU data becomes. We propose a time interval threshold [tmin,tmax] for optimizing the cost associated with IMU measurements. If the interval between two consecutive keyframes

exceeds this threshold, the corresponding preintegration is considered redundant or unreliable, and the associated cost function is disregarded during optimization. Therefore, we do not allow any two consecutive keyframes to differ more than 3.0s or less than 1e-3s. This improvement also, to some extent, enhances the efficiency of the algorithm by conserving computational resources in situations where IMU optimization processing would be unnecessary.

3.4 Combination of Energy Function

As mentioned earlier, we neglect noise and bias when processing data from the IMU. This implies that the constraints for state estimation derived from the IMU data may not be entirely accurate, and thus cannot be fully trusted. To address this, we introduce a weighting parameter λ in the cost function associated with the IMU measurements, witch is expected to be minimized.(3) This allows us to mitigate the adverse effects of bias and white noise.

$$E_{total} = E_{camera} + \lambda E_{IMU} \tag{3}$$

This joint optimization, which is tightly coupling, combines two energy functions directly into a single optimization framework. It significantly enhances the overall robustness and accuracy of the SLAM process.

In this part, we also conduct the experiment to tune the best value of lambda for our research. We implement the 4 different values of lambda with Machine Hall 01 dataset to search for the best performance. According to the table1, the value of lambda with 0.4 can reach the best result among them. Normally, the value of lambda is between 0 and 1. Since the energy function of IMU is smaller than the camera one. Therefore, we set this lambda value as 0.4 for our experiment.

	0.2	0.4	0.6	0.8
MH01	0.386	0.381	0.434	0.864

Table 1: The impact of the value of lambda for total energy function.

4 Experiment

In order to test our methodology, we conducted an experiment using the EuRoC dataset[16]. The EuRoC dataset was collected on-board a Micro Aerial Vehicle and contains stereo images, synchronized IMU measurements, and corresponding ground truth. It consists of 10 subsets categorized into 3 different levels and 2 rooms: the Machine hall and Vicon room. For our evaluation, we selected sequences from each level in the two rooms. The selected sequences for our experiment are: Machine Hall 01, Machine Hall 03, Machine Hall 05, Vicon Room 1 01, Vicon Room 1 02, and Vicon Room 1 03.

We used the basic visual odometry method as our baseline and compared it with our methodology, visual inertial odometry. The average trajectory error was used as the metric to assess the performance of the two methods. We executed 20 runs for each dataset to obtain the errors. For the IMU energy function, we set the weight λ to 0.4, as this value provided optimal performance during parameter tuning. Subsequently, we excluded outliers that were more than two standard deviations from the mean to obtain the final mean value for each sequence.

5 Results

In the section of results, we compare the average trajectory error and computational time between Visual Odometry (VO) and Visual Inertial Odometry (VIO). Then, We present the results with accompanying images to visualize the trajectory differences. To evaluate the visual SLAM system, [17], specifically comparing the root-mean-squared error (RMSE) of the absolute trajectory error (ATE) in meters between the trajectory estimated by the VO/VIO and the ground truth trajectory provided by the authors. Prior to calculating the ATE, we align the two trajectories using singular value decomposition (SVD).

Figure 4: ETH machine hall

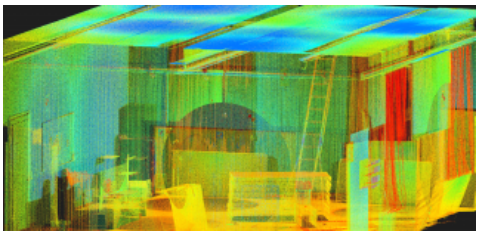

Figure 5: Vicon room

5.1 Quantitative Results

Dataset		Average Trajectory Error(M)	Min Error(M)	Max Error(M)
Machine Hall 01	VO	2.18	0.294	5.60
	VIO	0.91	0.0893	3.072
Machine Hall 03	VO	7.795	1.61	12.505
	VIO	6.40	1.093	17.55
Machine Hall 05	VO	7.1	2.62	11.053
	VIO	5.52	2.25	14.28
Vicon Room 1 01	VO	0.0911	0.0229	0.178
	VIO	0.0823	0.0239	0.176
Vicon Room 1 02	VO	7.118	1.67	12.802
	VIO	5.566	0.775	9.816
Vicon Room 1 03	VO	5.70	1.72	9.61
	VIO	6.61	1.32	10.78

Table 2: Average Trajectory Error of each dataset.

Based on the data presented in Table 12, it is evident that VIO generally outperforms VO and demonstrates greater robustness across the six sequences in the dataset. For instance, in the Machine Hall 01 sequence, VIO shows a significant reduction in average trajectory error, with a 58.26% improvement over VO. Similarly, in the Vicon Room 1 01 sequence, VIO exhibits improvements with a 9.65% and 21.82% reduction in errors, respectively. However, in certain instances, such as Machine Hall 04 and Vicon Room 1 03, VIO exhibits poorer performance on specific metrics, such as Max Error and Average Trajectory Error (ATE). This may be attributed to the more abrupt and rapid movements of the Micro Air Vehicle (MAV) in these sequences, which increase the computational burden on VIO's optimization process compared to VO. Consequently, this can limit the performance of VIO.

Referring to Table 3 on time consumption, the values in this table are very interesting. All sequences that were processed by VIO demonstrates higher efficiency compared to VO. For example, the dataset of Machine Hall 01 shows the obvious difference of time evaluation. The VO method has an execution time of 90.6694 seconds, while the VIO method is slightly faster at 86.8499 seconds. This improvement is likely due to the enhanced and more robust motion estimation provided by IMU Preintegration, which stabilizes the visual odometry process and contributes to more efficient optimization.

Dataset		Time consuming for execution(s)
Machine Hall 01	VO	90.6694
	VIO	86.8499
Machine Hall 03	VO	67.8577
	VIO	65.1349
Machine Hall 05	VO	47.8860
	VIO	47.4590
Vicon Room 1 01	VO	63.8821
	VIO	62.9647
Vicon Room 1 02	VO	36.9939
	VIO	36.4032
Vicon Room 1 03	VO	44.321
	VIO	43.6565

Table 3: Time consuming of both methods for each dataset.

5.2 Qualitative Results

In addition to the quantitative evaluation, a qualitative assessment was also conducted using the GUI provided during the tutorial. As expected, the alignment between the computed trajectory and the ground truth is more accurate when using VIO. In the earlier section, we identified failure cases as those where errors exceeded more than two standard deviations. We examined these failure cases to determine possible causes and discovered that some frames were significantly mislocalized, leading to substantial differences in position and orientation compared to previous frames. This issue typically arises when the camera moves at high speeds, when the scene is entirely dark or when the quality of mapping is inaccurate initially. While VIO sometimes encounter difficulties under these conditions, the experiments indicate that it is generally more robust than VO. This robustness is particularly noticeable in the Machine Hall 05 sequences, where the drone maneuvers through a dark environment at high speed. VO struggles in such scenarios, often resulting in poor camera localization upon entering dark scenes. However, VIO, aided by the IMU, seems to handle these challenging environments much more effectively, thus avoiding significant mislocalization.

As observed, the overall geometry of the calculated trajectory is accurate for each of the five sequences, with the exception of sequence MH_03, Vicon Room 02, and Vicon Room 03 (see 7 10 11), where both methods perform poorly. In summary, the alignment quality varies significantly across the five sequences due to the varying levels of difficulty, but it is evident that VIO consistently outperforms VO.

On the top of the image analysis, we also record the video when running both methods. Here is the link: the video. Through the video, we can figure out the process of the visual inertial odometry. After the alignment of SVD is done, we can check the predicted trajectory and the groundtruth.

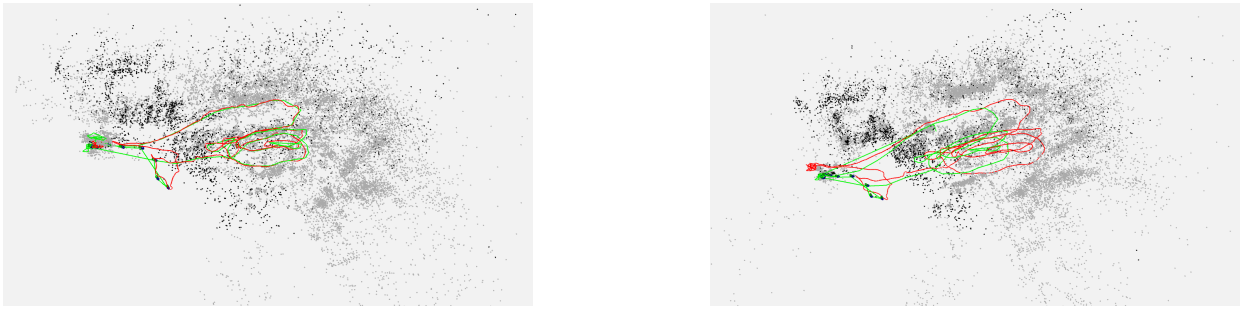

Figure 6: Qualitative results on Machine Hall 01 sequence. The green trajectories are the estimated trajectories from our VIO (left) and baseline VO (right), the red trajectories are ground truth, and the grey and black points are landmarks.

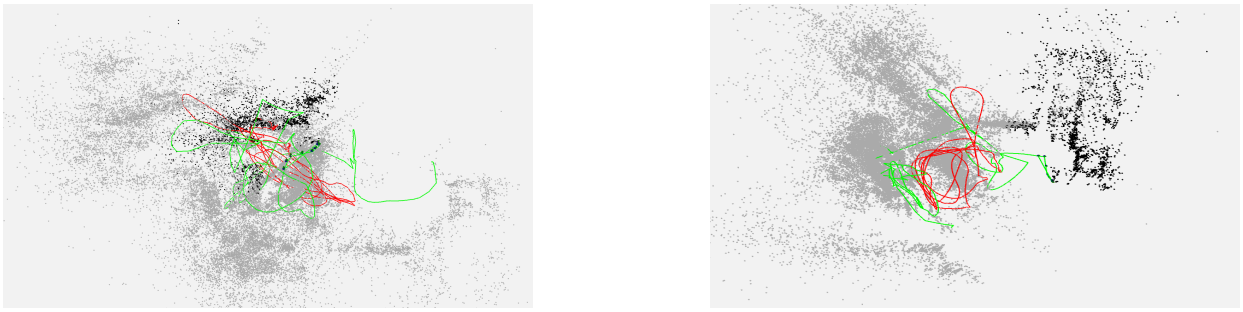

Figure 7: Qualitative results on Machine Hall 03 sequence.VIO (left) and VO (right)

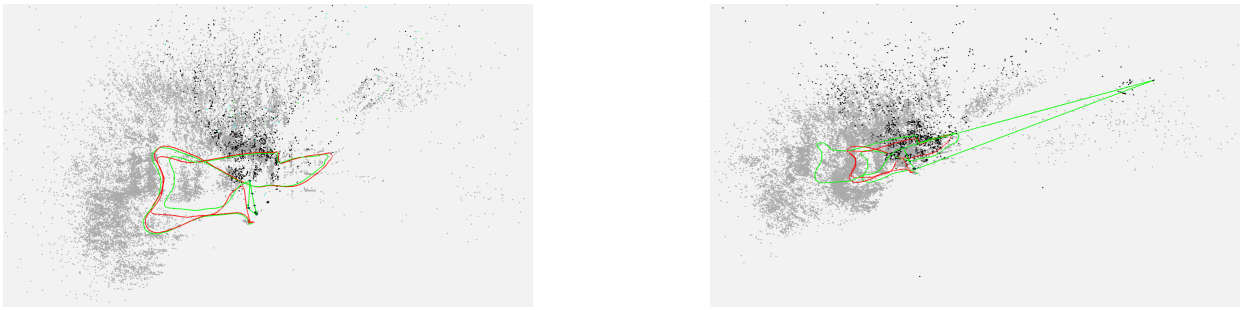

Figure 8: Qualitative results on Machine Hall 05 sequence.VIO (left) and VO (right)

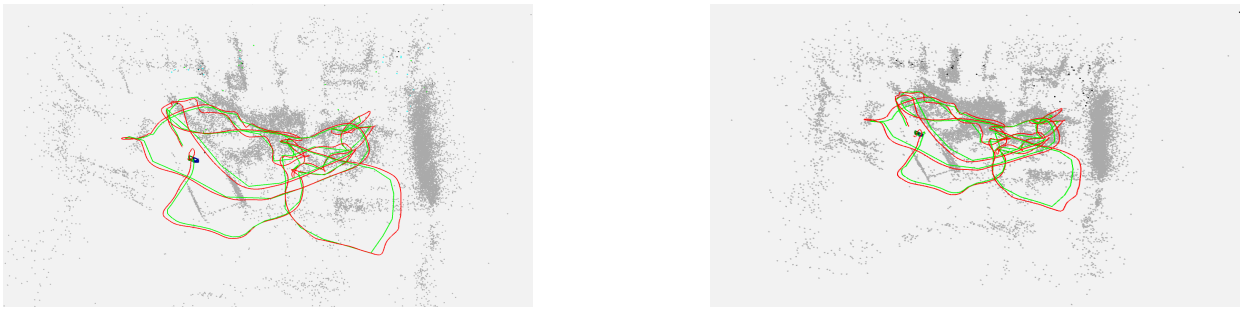

Figure 9: Qualitative results on Vicon Room 01 sequence.VIO (left) and VO (right)

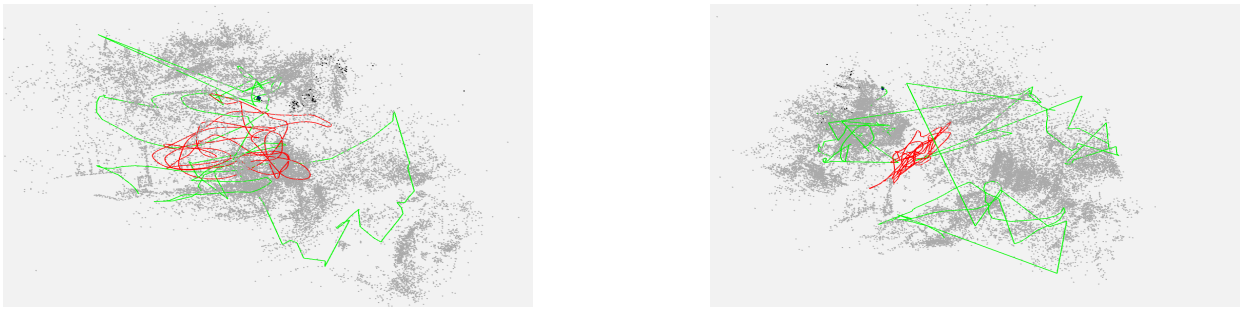

Figure 10: Qualitative results on Vicon Room 02 sequence. VIO (left) and VO (right)

Figure 11: Qualitative results on Vicon Room 03 sequence. VIO (left) and VO (right)

6 Summary

With the guidance of two influential papers[10][6], we have achieved strong-coupled visual-inertial odometry by preintegrating IMU measurements in our project. This method generally improves overall performance compared to visual odometry we done in this pracitical course. Specifically, it reduces the root mean square error (RMSE) of the average trajectory error in the EuRoC dataset[16]. Obviously, with the equipment of IMU sensor, it can enhance the accuracy of the optimization. Besides, it is not the trade-off between the time cost and the capability of prediction. The executed time for both methods is almost the same. Upon observation, it is evident that visual-inertial odometry produces better trajectory results than visual odometry. However, the rapid movement of the agent can leads to the less inaccurate predictions for trajectory. Overall, VIO improves the performance outcomes over the VO algorithm.

7 Future Work

In our project, we are working on replicating visual-inertial odometry, which has demonstrated superior performance compared to visual odometry. However, our current implementation only accounts for poses and velocity, neglecting the biases of the IMU sensors. Incorporating these variables could lead to significant improvements. Additionally, we aim to introduce closed-loop functionality to enhance the project's comprehensiveness.

It is crucial to prioritize reducing computational costs and enhancing real-time performance. This can be achieved by eliminating redundant keyframes based on time intervals or the similarity of detected features between consecutive keyframes.

Visual-inertial odometry is not a new method, and with the rise of artificial intelligence, many visual SLAM methods have been implemented using deep learning. Detailed discussions of state-of-the-art approaches can be found in the following references: [18][19] Additionally, in the future, we may explore extending our implementation to incorporate deep learning methods.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[4] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3565–3572.

[5] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.

[6] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1885–1892.

[7] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras," *International Journal of Computer Vision*, vol. 113, no. 3, pp. 208–219, 2015.

[8] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013.

[9] S. Sirtkaya, B. Seymen, and A. A. Alatan, "Loosely coupled kalman filtering for fusion of visual odometry and inertial navigation," in *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 219–226.

[10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration theory for fast and accurate visual-inertial navigation," *CoRR*, vol. abs/1512.02363, 2015. [Online]. Available: http://arxiv.org/abs/1512.02363

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of The ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2020.

[15] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[16] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[17] D. Prokhorov, D. Zhukov, O. Barinova, K. Anton, and A. Vorontsova, "Measuring robustness of visual slam," in *2019 16th International Conference on Machine Vision Applications (MVA)*, 2019, pp. 1–6.

[18] D. Cai, R. Li, Z. Hu, J. Lu, S. Li, and Y. Zhao, "A comprehensive overview of core modules in visual slam framework," *Neurocomputing*, p. 127760, 2024.

[19] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual slam," *Expert Systems with Applications*, vol. 205, p. 117734, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422010156