2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Efficient Visual-Inertial Navigation with Point-Plane Map

Jiaxin Hu, Kefei Ren, Xiaoyu Xu, Lipu Zhou, Xiaoming Lang, Yinian Mao and Guoquan Huang

***Abstract*— Accurate and real-time global pose estimation relative to a global prior map is indispensable in many applications, such as logistics with micro aerial vehicles and Augmented Reality. Supposed that a pure sparse 3D point map can provide a structureless representation of the environment, then generating a point–plane prior map can further model the environment topology and offer global constraints for an accurate localization. To implement this, we propose a filter-based, large-scale visual-inertial odometry system, termed PPM-VIO, which utilizes a point–plane map to correct the cumulative drift. Our system, detecting coplanar information from sparse point clouds with semantic information, achieves accurate online plane matching via geometric constraints, semantic constraints, and descriptor constraints. To improve the localization performance, we effectively integrate and formulate the global planar measurements and points measurements in a filter-based estimator. The effectiveness of the proposed method is extensively validated on real-world datasets collected in different scenarios. Experimental results demonstrate that, rather than using the point map alone, leveraging the plane information in the prior map can yield better trajectory estimates and broaden the effective scope of the prior map in different scenes.**

I. Introduction

Accurate localization, *i.e.*, estimating the global pose in a given scene, empowers intelligent systems, such as micro aerial vehicles (MAV) and self-driving cars, to estimate their current pose in an environment and thus to navigate to their target place. Over the last decades, visual-inertial odometry (VIO) [1]–[3] has become increasingly prominent in robotic localization due to its complimentary sensing nature, weight, and low cost, particularly in GPS-denied environments, such as indoor, and in the urban canyons. However, one of the most crucial obstacles that limits the application of VIO is that visual-inertial sensors can only provide the relative pose estimation without global position and yaw information.

To overcome this problem, the state-of-the-art methods for online, low-latency visual-inertial localization assume the existence of a prior map to provide global information. For the platforms with limited computational resources, the prior map is often yielded from a set of database images by using Structure-from-Motion (SfM) [4]–[6]. After local features in an image taken by the client camera are extracted, 2D-3D correspondences are established via matching the descriptors associated with the 2D query and 3D map points. To achieve global localization, [4] and [7] optimized the alignment by incorporating global 2D-3D correspondences into the bundle adjustment (BA) of the local map. To limit the computational complexity, [5] applied an extended Kalman filter (EKF)-based algorithm to integrate the various sources of information tightly, thus leading to good estimation accuracy and robustness. Furthermore, the Schmidt-Kalman filter (SKF) [8], which allowed for schmidt states that are treated as nuisance parameters and not updated, was leveraged to incorporate global information directly into the estimator in a consistent manner. The Cholesky-Schmidt-Kalman filter was also proposed by [9], which showed that their method was computationally efficient and consistent compared with simple measurement inflation method.

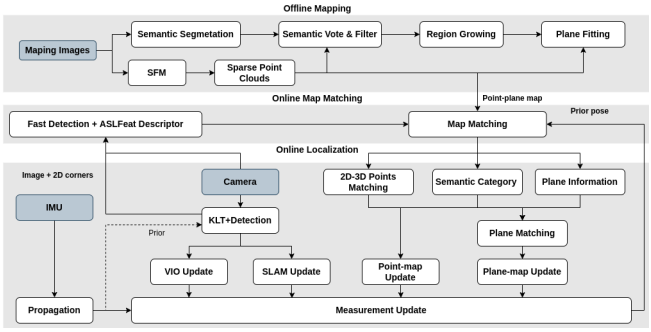


Fig. 1: Overview of our PPM-VIO system.

In real-world applications, the widely existing points are the most common features to be added to the estimator as the visual global information. The planar features can also be applied to improve the robustness of the system. In particular, in some coplanar scenes with uniform textures, such as lawns, there are little point matches but strong coplanarity constraints. For leveraging the plane information in the real-time VIO system, efficient plane extraction and matching approaches are the key problems. Planar constraints applied in real-time VIO systems often rely on depth sensors [11]–[13], which extracted plane from the dense point clouds. PVI-DSO [14], as a semi-dense system, extracts coplanar regularities from the generated 3D mesh, and matching the plane on the basis of geometric information. [15] detected planes from sparse clouds, not only leveraging the planar regularities but also involving both point features and line features, so that richer structural information is used for 3D mesh generation. However, due to the limited information of online point clouds, the above two methods can only perform plane matching through geometric constraints, and there may be mismatches when the depth estimation of the system is inaccurate, which affects the localization accuracy. Besides, to ensure the correctness of plane matching, the plane direction needs to be limited, which weakens the effectiveness of coplanar constraints.

In this paper, we propose a filter-based, large-scale VIO

The authors are with the Meituan UAV, Beijing, China (e-mail: hujiaxin04 | renkefei | xuxiaoyu | zhoulipu | langxiaoming | maoyinian | huangguoquan@meituan.com).

system that involves a prior map including points and planes (PPM-VIO) to implement highly accurate, real-time localization. On the basis of a sparse point cloud map generated by SFM, we leverage the semantic segmentation and region growing methods to detect precise plane model offline. In the online process, the proposed method not only applies the geometric information but also utilizes both semantic information obtained from the map and descriptor information for matching planes, so that more coplanar features can be detected to offer global constraints for the solution. Besides, we do not require limiting the plane direction. In addition, by drastically compressing the cost of map matching and global fusion, our approach is able to handle large-scale localization and works in real time for clients having computational resources similar to micro aerial vehicles. Overall, the key contributions of our work are as follows:

- We develop a large-scale VIO system based on a point–plane map that entirely runs on devices with limited computational and memory resources while offering accurate and real-time localization.
- To exploit the structural information of the environment, we propose an offline point–plane map building method and an efficient online matching strategy. Our method performs the semantic segmentation and plane detection operations offline, and utilizes semantic, geometric and descriptor constraints to improve the correctness and robustness of online plane matching.
- We provide a global localization approach that efficiently incorporates point and plane global information into the estimator to avoid the affects of drift accumulation.
- We evaluate the proposed PPM-VIO on real-world datasets, which demonstrates that our system improves the estimator accuracy with low computational consumption. Besides, leveraging the plane information in the prior map can significantly improve the localization accuracy rather than using the point map alone, especially in the scenarios with low texture and dominantly planar surfaces.

II. System Overview

Based on the MSCKF estimator [1], the proposed PPM-VIO applies the point-plane prior map to improve the accuracy and robustness of the system. As shown in Fig. 1, the proposed system contains three main modules: offline mapping, online query and localization. In the offline mapping stage, we follow the classic SFM to generate the sparse point clouds, and then utilize the semantic segmentation and voting methods to obtain the semantic label of point clouds. On the basis of semantic point clouds, region growth aggregation algorithm and plane fitting based on RANSAC is applied to generate the accurate plane map. In the online stage, fast detection and ASLFeat [18] descriptor are leveraged to establish the correspondences of 2D query and 3D map points that contain the map's plane information. Besides, semantic label can also be obtained by descriptor matching and nearest neighbor search. In the localization process, the matching points, as point-map global constraints, are added to the local estimator to constraint the pose of the system. Moreover, via the 2D-3D correspondences, semantic label and plane geometric parameters, we can find all the features that are on the map plane to offer coplanar global constraints. Finally, by integrating the global constraints, local windows' MSCKF and SLAM constraints into the estimator, we can obtain accurate global localization without accumulative drift.

III. Offline Mapping

As shown in Fig. 1, we pre-build a point-plane map in the offline stage, and the map is generated according to the following steps.

1) Point Cloud Map Generation: We follow the classic SFM process to build the point cloud map, which will produce accurate camera poses and 3D points, thereby helping us produce accurate planar structure information and global positioning in the subsequent process.

2) Semantic Label Assignment: Since the semantic information of the point clouds is helpful, we utilize a semantic segmentation model to yield the semantic labels for the mapping image. We all know that in the point cloud generated in the last step, a 3D point will be observed by multiple cameras and therefore associated with several semantic classes. Since a 3D point can be associated with a few semantic classes due to possible view occlusion or segmentation errors, its semantic class is voted as the most frequent one.

3) Plane Fitting: The point cloud produced by our method is sparse and thus we choose the region growing method to produce the plane. To improve the efficiency, we remove those landmarks that can not explicitly form a plane, such as trees and vehicles. Then, we compute the curvature of each 3D point using the neighboring points. All 3D points that satisfy the curvature consistency, semantic consistency and distance consistency, will be divided into the same region. Using the RANSAC method, each point cloud region is fitted as a plane with its semantic categories, standard deviations of the distance from the point to the plane and plane covariance leveraged for localization.

IV. Online Visual-Inertial Localization

The state vector of the proposed system is defined as follows:

$$\mathbf{x} = [\mathbf{x}_I^T \quad \mathbf{x}_E^T \quad \mathbf{x}_C^T \quad \mathbf{x}_L^T \quad \mathbf{x}_M^T \quad \mathbf{x}_\Pi^T] \tag{1}$$

$$\mathbf{x}_E = [{}^G_M\mathbf{q}^T \, {}^G\mathbf{p}_M^T] \tag{2}$$

$$\mathbf{x}_C = [\mathbf{x}_{T_1}^\top \dots \mathbf{x}_{T_m}^\top] \tag{3}$$

$$\mathbf{x}_L = [{}^{A_1}\mathbf{f}_1^\top \dots {}^{A_i}\mathbf{f}_i^\top] \tag{4}$$

$$\mathbf{x}_M = [{}^M\mathbf{p}_{f_1}^T \dots {}^M\mathbf{p}_{f_j}^T] \tag{5}$$

$$\mathbf{x}_\Pi = [{}^M\mathbf{p}_{\Pi_1}^T \dots {}^M\mathbf{p}_{\Pi_k}^T], \tag{6}$$

where

$$\mathbf{x}_I = [{}^G\mathbf{p}_I^\top \; {}^G\mathbf{v}_I^\top \; {}^I_G\mathbf{q}^\top \; \mathbf{b}_a^\top \; \mathbf{b}_g^\top]^\top \tag{7}$$

$$\mathbf{x}_{T_i} = [{}^{I_i}_G\mathbf{q}^T \, {}^G\mathbf{p}_{I_i}^T], \tag{8}$$

where $\mathbf{x}_I$ indicates the IMU state, ${}^I_G\mathbf{q}$ is the JPL quaternion parameterizing the rotation from the global frame of reference $\{G\}$ to the IMU local frame $\{I\}$, ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ are the IMU position and velocity in global frame, and $\mathbf{b}_a$,$\mathbf{b}_g$ are the gyroscope and accelerometer biases, respectively. The clone state $\mathbf{x}_C$ contains m historical IMU poses, and $\mathbf{x}_L$ contains i local temporal SLAM features represented in anchored frame $\{A\}$. Additionally, $\mathbf{x}_M$ denotes the matching map features represented in the map frame $\{M\}$, $\mathbf{x}_E$ denotes the transformation between the device's global frame and map frame, and $\mathbf{x}_\Pi$ stores environmental plane features, which are represented by the closest point [16].

A. IMU Propagation

The inertial state $\mathbf{x}_I$ is propagated forward using incoming IMU measurements of linear accelerations $\mathbf{a}_m$ and angular velocities $\boldsymbol{\omega}_m$ based on the following generic nonlinear IMU kinematics [17]:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{a}_m - \mathbf{n}_a, \boldsymbol{\omega}_m - \mathbf{n}_\omega), \tag{9}$$

where $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are the zero-mean white Gaussian noises of the IMU measurements. We can then linearize the IMU kinematics at the current state estimate, and propagate the state and covariance forward.

B. Local Measurement Update

Based on the track length, we divide all the tracked point features into "MSCKF" and "SLAM", which correspond to different update approaches.

1) MSCKF feature update: Consider a 3D feature is detected from the camera image at time k, whose measurement on the image plane is given by:

$$\mathbf{z}_k = h(\mathbf{x}_{T_k}, \mathbf{x}_A, {}^A\mathbf{f}) + n =: \pi({}^{C_k}\mathbf{p}_f) + \mathbf{n}_k \tag{10}$$

$$\pi([x \quad y \quad z]^T) = [x/z \quad y/z]^T \tag{11}$$

$${}^{C_k}\mathbf{p}_f = {}^C_I\mathbf{R}\,{}^{I_k}_G\mathbf{R}({}^G_{I_A}\mathbf{R}p({}^A\mathbf{f}) + {}^G\mathbf{p}_{I_A} - {}^G\mathbf{p}_{I_k}) + {}^C\mathbf{p}_I, \tag{12}$$

where ${}^{I_A}\mathbf{p}_f = p({}^A\mathbf{f})$ is the parameterization of the 3D point feature, and $\mathbf{n}_k$ is a zero-mean Gaussian noise vector with covariance matrix $\sigma^2\mathbf{I}_2$. The linearized residual equation is:

$$\mathbf{r} = \mathbf{H}_C\tilde{\mathbf{x}}_C + \mathbf{H}_L\tilde{\mathbf{x}}_L + \mathbf{n}, \tag{13}$$

where $\tilde{\mathbf{x}}_C$ and $\tilde{\mathbf{x}}_L$ are the estimation errors of the camera pose and feature, respectively, whereas matrix $\mathbf{H}_C$ and $\mathbf{H}_L$ are the corresponding Jacobians. The feature $\tilde{\mathbf{x}}_L$ in the residual equation is marginalized out by multiplying matrix $\mathbf{L}$, which is constructed from the left nullspace of $\mathbf{H}_L$, on both sides. The transformed equation can be directly used for the standard EKF update without storing features in the state.

2) SLAM feature update: For those features that can be reliably tracked longer than the current sliding window, we initialize them into the state, and directly update the state using Eq.(13).

C. Point-map Measurement Update

Building upon the MSCKF, in what follows, we describe in detail how to fully exploit the prior point-map to correct the drift in the local pose estimates.

1) Map Matching: In the online stage, in addition to the real-time VIO system, we will trigger a separate thread for map matching when the current image frame is a key frame and the time interval over the last trigger is greater 1 second.

For the frame triggered map matching, in addition to the points extracted in the VIO process, called VIO points, we will extract some additional points to guarantee enough map matches. ASLFeat is leveraged to calculate the descriptor, which is robust to illumination, but sensitive to scale. To overcome the scale problem, we resize the image to the same scale as the one when constructing the map according to the current flight altitude, and extract descriptors on the resized image, which avoids the establishment of a multi-scale map, and thus reduces the size of the prior map.

In the process of map matching, local pose estimator can provide a prior pose that is sufficient for us to find nearby images and 3D points. Based on the prior pose, we can project adjacent point clouds onto the query image, and find the matching points with similar visual descriptors among nearby projected points. Then, the ratio test and PnP RANSAC techniques can be applied to filter out most of the false matches. The flowchart of the above process is shown in Fig. 2.

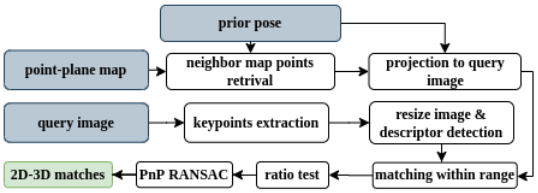


Fig. 2: Flowchart of map matching.

2) Global Fusion with Point Map: In order to bootstrap the localization system, a marker board that has been calibrated offline is placed at the take-off location, which helps us align the VIO local coordinate and map coordinate systems. Subsequently, 2D-3D point matches incorporating the global information can be directly used to update the local estimator. According to the type of matching points (VIO points or other points), we apply different methods for global fusion to achieve a better localization effect.

Matches with VIO Points: For the global 2D-3D matches where the 2D points can be tracked in the sliding window and not added to the state, we add the 3D position of the points to the state as map points $\mathbf{x}_M$. For the point-map measurement, the non-linear measurement model is

$$\mathbf{z} = h(\mathbf{x}_{T_k}, \mathbf{x}_E, {}^M\mathbf{p}_f) + n =: \pi({}^C\mathbf{p}_f) + \mathbf{n} \tag{14}$$

$$\pi([x \quad y \quad z]^T) = [x/z \quad y/z]^T \tag{15}$$

$${}^{C_k}\mathbf{p}_f = {}^C_I\mathbf{R}\,{}^{I_k}_G\mathbf{R}({}^G_M\mathbf{R}\,{}^M\mathbf{p}_f + {}^G\mathbf{p}_M - {}^G\mathbf{p}_{I_k}) + {}^C\mathbf{p}_I. \tag{16}$$

We linearize this measurement model and obtain the following residual:

$$\mathbf{r} = \mathbf{z} - h(\hat{\mathbf{x}}_T, \hat{\mathbf{x}}_E, {}^M\hat{\mathbf{p}}_f) \tag{17}$$

$$\simeq \mathbf{H}_T\tilde{\mathbf{x}}_T + \mathbf{H}_E\tilde{\mathbf{x}}_E + \mathbf{H}^M_f\tilde{\mathbf{p}}_f + \mathbf{n}. \tag{18}$$

where ${}^{M}\mathbf{p}_f \in \mathbf{x}_M$. To improve the computational efficiency, Schmidt-EKF [8] is applied to update the state $\mathbf{x}_M$. In order to keep the memory requirements bounded, we limit the total number of slam features and map features to no more than 50 in our experiments.

For the global 2D-3D matches where the 3D points have been added to the state $\mathbf{x}_L$, the measurement model is

$$\mathbf{r} = {}^{M}\mathbf{p}_f^m - {}^{G}_{M}\mathbf{R}^T(({}^{G}_{I_A}\mathbf{R}p({}^{A}\mathbf{f}) + {}^{G}\mathbf{p}_{I_A}) - {}^{G}\mathbf{p}_M) + \mathbf{n} \quad (19)$$

$$\simeq \mathbf{H}_T^{'}\tilde{\mathbf{x}}_T + \mathbf{H}_E^{'}\tilde{\mathbf{x}}_E + \mathbf{n}. \quad (20)$$

The measurement noise is $\mathbf{n} \sim N(\mathbf{0}, \mathbf{R})$ where $\mathbf{R} = \alpha\mathbf{P}_{ff}$, and $\mathbf{P}_{ff}$ is the covariance of prior map landmark ${}^{M}\mathbf{p}_f^m$. When these "SLAM" and map features are lost in the current window, we marginalize these features from the state.

Matches with Other Points: As the additional extracted matching points have no other observations in the future, we do not estimate the corresponding map points in the state. The linearized measurement model in this case is

$$\mathbf{r} = \mathbf{z} - h(\hat{\mathbf{x}}_T, \hat{\mathbf{x}}_E, {}^{M}\hat{\mathbf{p}}_f) \quad (21)$$

$$\simeq \mathbf{H}_T\tilde{\mathbf{x}}_T + \mathbf{H}_E\tilde{\mathbf{x}}_E + \mathbf{n} = \mathbf{H}_x\tilde{\mathbf{x}} + \mathbf{n}. \quad (22)$$

Considering that the same map landmarks can be matched multiple times, measurement noise inflation is applied to prevent inconsistency. The measurement noise is

$$\mathbf{R} = \sigma_{pix}^2\mathbf{I} + \lambda\mathbf{H}_f\mathbf{P}_{ff}\mathbf{H}_f^T. \quad (23)$$

When the additional extracted matching points have a large number (which is not unusual in the structured environment), QR-decomposition to $\mathbf{H}_x$ can be performed using Givens rotations to reduce the computational complexity.

$$\mathbf{H}_x = [\mathbf{Q}_1 \quad \mathbf{Q}_2]\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}, \quad (24)$$

where $\mathbf{R}$ is an upper triangular matrix, and $\mathbf{Q}_1, \mathbf{Q}_2$ are orthogonal matrices. The measurement model can then be rewritten as $\mathbf{Q}_1^T\mathbf{r} = \mathbf{R}\tilde{\mathbf{x}} + \mathbf{Q}_1^T\mathbf{n}$. This measurement function can directly update the state.

D. Plane-map Measurement Update

1) Plane Data Association: The prior information provided by the point-plane map includes the following categories:

Plane information: For each point that has been retrieved with the map, whether the point is on a plane is provided by the map. For the points that are on a map plane, we can obtain the corresponding plane information, including plane ID, semantic ID, plane parameter, and corresponding covariance $\mathbf{R}_\Pi$ that is solved by [12].

Semantic information: The semantic labels of all the SLAM features that have been retrieved with the map, is assigned by the corresponding semantic labels of the map points. For the features that are not matched with the map, the nearest neighbor method is applied to assign the semantic labels. Since the map matching module also requires to look for the nearest neighbors of each point, no additional computation is added here.

In addition to the points that have been matched with the map, we aim to find more SLAM features on the plane where the matching point is located, and thus provide more global constraints. To this end, we project the 3D SLAM features with semantic information onto the query image that triggers global localization, and apply 2D Delaunay triangulation to the projection points. However, there might be some invalid triangular patches because coplanarity is not considered in the process of triangle establishment. To remove those triangles that are not coplanar, we filter out the triangles whose semantic labels of three vertices are different, and treat the remaining triangles as candidate triangles, which have the semantic information and three vertices are coplanar. For each matching point with plane information, we find a candidate triangle with the matching point as its vertex or a candidate triangle containing the matching point, as shown in Fig. 3, and then take the candidate triangles that satisfy the following geometric constraints as the seed triangles.

$$\mathbf{l}_1 \cdot \mathbf{n}^G < t_{thresh} \quad (25)$$

$$\mathbf{l}_2 \cdot \mathbf{n}^G < t_{thresh} \quad (26)$$

$$\mathbf{p}_{center} \cdot \mathbf{n} - d^G < p_{thresh} \quad (27)$$

where $\mathbf{n}^G$ is the surface normal vector provided by the map, and $\mathbf{p}_{center}$ is the centroid of triangle. We then traverse the neighboring triangles of the seed triangles, and determine whether a neighboring triangle belong to the same plane, according to the following conditions:

- The semantic category is the same as the seed triangle,
- and it satisfies the geometric constraints of Eq.(25)-(27).

By applying the iterative approach, we find all slam features that locate the retrieved map plane. Given that the strategy of plane data association inevitably has outliers, Mahalanobis gating test is leveraged to reject the outliers before operating the global fusion.

The semantic labels of map features are relatively accurate, but the initial semantic labels of SLAM features are affected by the prior pose and hence may be incorrect. Yet, for the features that pass the Mahalanobis gating test, their semantic labels can be recognized as fixed, thus each SLAM feature will be not matched to a different plane.

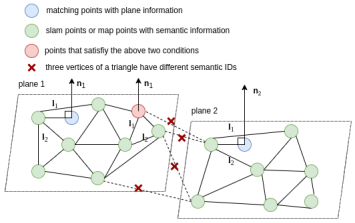


Fig. 3: Based on the map matching, geometric and semantic information, the coplanar regularities of slam features and map features are detected.

2) Global Updates with Plane Map: Once a feature is detected on a map plane, this coplanar constraint always exists. To ensure the computational efficiency, the constraint is updated only when the map matching is triggered. Considering a planar point, ${}^{M}\mathbf{p}_{f_i}$, that is identified on the map plane

${}^{M}\mathbf{p}_{\Pi}$, we have the following point-on-plane measurement constraint:

$$\mathbf{z}^{\Pi} = \frac{{}^{M}\mathbf{p}_{\Pi_k}^{T}}{\left\|{}^{M}\mathbf{p}_{\Pi_k}\right\|} \cdot {}^{M}\mathbf{p}_{f_i} - \left\|{}^{M}\mathbf{p}_{\Pi_k}\right\| + \mathbf{n}. \tag{28}$$

Here, we treat the point-on-plane constraint as a probabilistic compensation for the feature uncertainty of the planar model, rather than a hard constraint. $\mathbf{n} \sim N(\mathbf{0}, \sigma_d^2\mathbf{I})$ denotes the noise of point-on-plane constraint, where σ_d is set to the standard deviation of the distance from the point to the plane when the plane is fitted offline. We linearize this measurement model and obtain the following residual:

$$\mathbf{r}^{(j)} = \mathbf{H}_x^{(j)}\tilde{\mathbf{x}} + \mathbf{H}_{\Pi}^{(j)}\,{}^{M}\tilde{\mathbf{p}}_{\Pi} + \mathbf{n}^{(j)}, \tag{29}$$

where $\mathbf{H}_{\Pi}^{(j)}$ and $\mathbf{H}_x^{(j)}$ are the Jacobians with respect to the plane state and other states, respectively. According to the chain rule, the Jacobians can be computed:

$$\mathbf{H}_{\Pi} = \frac{\partial \tilde{\mathbf{z}}^{\Pi}}{\partial {}^{M}\tilde{\mathbf{p}}_{\Pi}} \tag{30}$$

$$= \frac{1}{{}^{M}d_{\Pi}} *^{M} \mathbf{p}_f^{T} * (\mathbf{I}_{3*3} -^{M} \mathbf{n}_{\Pi} *^{M} \mathbf{n}_{\Pi}^{T}) -^{M} \mathbf{n}_{\Pi}^{T} \tag{31}$$

$$\mathbf{H}_x = \frac{\partial \tilde{\mathbf{z}}^{\Pi}}{\partial \tilde{\mathbf{x}}} = \frac{\partial \mathbf{z}}{\partial {}^{M}\tilde{\mathbf{p}}_f}\frac{\partial {}^{M}\tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}} =^{M} \mathbf{n}_{\Pi}^{T}\frac{\partial {}^{M}\tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}}, \tag{32}$$

where ${}^{M}\mathbf{p}_{\Pi} =^{M} d_{\Pi}^{M}\mathbf{n}_{\Pi}$. The measurements of the plane with a short track length can directly update the state,

$$\mathbf{r}^{(j)} = \mathbf{H}_x^{(j)}\tilde{\mathbf{x}} + \mathbf{n}_s^{(j)}. \tag{33}$$

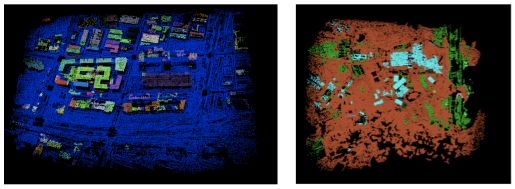
The measurement noise is:

$$\mathbf{R} = \beta\mathbf{H}_{\Pi}\mathbf{R}_{\Pi}\mathbf{H}_{\Pi}^{T} + \sigma_d^2\mathbf{I}, \tag{34}$$

where $\mathbf{R}_{\Pi}$ is the noise of plane obtained from the map. The plane with a long track length can be initialized into the state as a plane feature with initial covariance $\mathbf{R}_{\Pi}$. To improve computational efficiency, we prioritize inserting the planes whose normal directions are significantly different from the planes currently being estimated, which also guarantees that the state estimate can be well constrained. If a feature is found on the plane currently being estimated, then we can directly update its estimate and the state using the formulation Eq.(29). Similar to map points, Schmidt-EKF is leveraged to update the plane state. When the map matching fails for a long time or when the matching succeeds but all the matching points are not on a plane, we marginalize the plane from the state.

V. Real-World Experiments

In this section, to demonstrate the effectiveness and robustness of our method, we conduct experiments in two different environments (town and grassland environment), and the groundtruth is obtained from RTK-GPS. The sensors of the data acquisition platform include an IMU (250Hz), a downward stereo camera (10Hz), and a downward LRF (100Hz) with 120m effective range. We evaluate the estimator the computational cost of each modules, and the accuracy in different scenarios and map types, which show that the proposed algorithm can remarkably reduce the cumulative drift of VIO system with low computational consumption.



(a) Town (b) Grassland

Fig. 4: Point-plane map in experiments. Each color represents a plane.

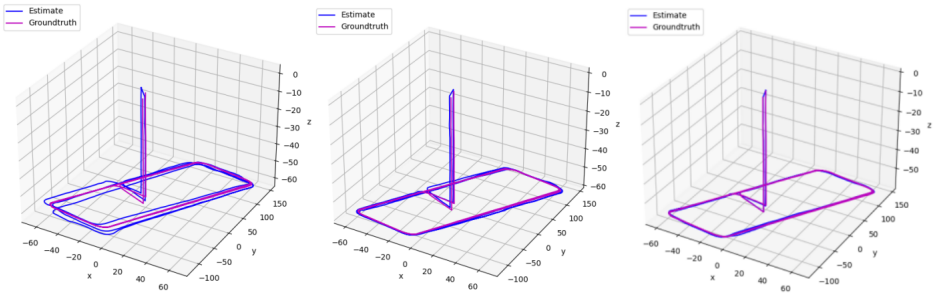
A. Localization Accuracy

The VIO algorithm used in this work is based on our prior work LRF-VIO [19], which performs better in high-altitude MAV flight than other state-of-the-art methods.

1) Town Sequences: The proposed system is evaluated on the data collected at Hangzhou, whose plane map is shown in Fig. 4(a). In these sequences, point-map matching works better, where the recall ratio (the number of valid matches divided by the number of map matches triggered) is over 0.9 and the average inlier number is over 80. We compare the VIO based on different prior map types with pure VIO in these sequences with different heights and trajectory lengths. Table I shows the absolute trajectory error (ATE) of all methods. The cumulative drift of VIO can be corrected by using a point-map or plane-map that incorporates global information. Note that the most plane normal directions in the town scene are parallel to the direction of gravity, which leads to the weak constraint effect on the horizontal position and yaw, thereby explaining why the accuracy of the point-map is better than the plane-map. The improvement effect of the plane-map will be more obvious in the sequences including the planes with other directions. As shown in Table I and Fig. 5, in the town scene, the proposed PPM-VIO algorithm corrects the cumulative drift of the pure VIO system, which achieves a high localization accuracy with an ATE less than 1 m in up to 3 km trajectory.

2) Grassland Sequences: To verify the effectiveness of our proposed plane-map prior, we further evaluate the proposed system on difficult grassland sequences with challenges like repetitive texture environments like lawns, and non-coplanar environments such as trees. The point-plane-map is shown in Fig. 4(b). In these sequences, it is hard to construct the point-map matching since the number of matching points is less than 15, and the recall ratio is less than 0.3. When the number of matching points is small, the constraint effect of the point-map is limited, but there can be a strong coplanar constraint in this case. As shown in Table II, the plane-map constraint can effectively improve the accuracy of the localization system, but the constraints of point-map are less effective.

We can observe that using the point-plane map prior exhibits better localization performance than the point-map-based VIO system in different scenes, especially in the scene with low texture and dominantly planar surfaces. The



(a) pure VIO (b) VIO with plane-map (c) VIO with point-plane map

Fig. 5: The trajectory estimates of the proposed PPM-VIO on Town5 dataset.

TABLE I: TOWN EXPERIMENT RESULT

	Trajectory Length	Flight Height	VIO	VIO with Plane Map	VIO with Point Map	VIO with Point-Plane Map
	km	m	ATE(m/deg)	ATE(m/deg)	ATE(m/deg)	ATE(m/deg)
Town1	0.4	60	0.89 / 1.14	0.57 / 1.16	0.54 / 1.11	**0.45** / **1.09**
Town2	0.55	60	1.83 / 1.68	0.97 / 2.12	**0.45** / 1.41	0.64 / **1.04**
Town3	0.8	60	1.18 / 0.96	0.77 / 0.77	0.48 / 0.76	**0.41** / **0.63**
Town4	1.2	90	2.35 / 0.81	2.54 / 0.92	**0.65** / 0.77	0.68 / **0.62**
Town5	1.5	60	4.79 / 0.91	1.36 / 0.72	0.68 / 0.70	**0.64** / **0.55**
Town6	2.0	90	5.03 / 1.15	1.96 / 0.93	0.68 / 0.63	**0.65** / **0.62**
Town7	3.0	90	3.97 / 1.28	2.51 / 1.64	0.62 / 0.48	**0.54** / **0.48**

TABLE II: GRASSLAND EXPERIMENT RESULT

	Trajectory Length	Flight Height	VIO	VIO with Plane Map	VIO with Point Map	VIO with Point-Plane Map
	km	m	ATE(m/deg)	ATE(m/deg)	ATE(m/deg)	ATE(m/deg)
Grassland1	0.7	60	3.18 / 2.74	2.23 / 2.38	3.56 / 2.67	**2.00** / **2.14**
Grassland2	1.0	60	4.24 / 2.83	2.67 / 2.12	4.05 / 2.78	**2.32** / **1.90**
Grassland3	1.3	80	2.76 / 0.94	**1.78** / 0.97	1.96 / **0.82**	2.59 / 0.87
Grassland4	1.3	100	4.54 / 2.42	2.49 / **0.70**	3.13 / 0.80	**1.25** / 0.78
Grassland5	2.0	60	7.21 / 4.87	3.65 / 2.51	7.08 / 4.79	**2.32** / **1.90**
Grassland6	2.0	80	7.32 / 4.72	4.24 / 2.75	5.63 / 3.89	**2.87** / **2.50**
Grassland7	2.2	120	10.42 / 5.24	5.43 / 4.11	6.04 / 4.31	**4.43** / **4.10**

proposed point-plane map broadens the effective scope of the prior map in different scenes.

B. Runtime Evaluation

In this section, we evaluate the execution efficiency of the proposed method on a platform with an Intel Core i7-10750H CPU and an NVIDIA GeForce GTX 1650 Ti. Considering the real-time performance of the system, we set the size of the sliding window to 10, the number of extracted points in normal frame to 100, the number of extracted points in map-matching frame to 500, and the maximum number of map points applied in update to 50. From Table. III, we can observe that plane matching, plane updates, and map point updates are very efficient in our system. The most time-consuming part is the map matching module, but this part is only triggered at below 1Hz. Therefore, compared with pure VIO system, the total time of our system is only increased by 2 ms, and the CPU loading is increased by 3%.

TABLE III: TIMING ANALYSIS OF DIFFERENT APPROACHES

Module	Ours	VIO
Point detection (<1Hz)	5.5ms	0ms
Descriptor extraction (<1Hz)	5.5ms	0ms
Map matching and RANSAC (<1Hz)	23ms	0ms
Point updates (<1Hz)	2.5ms	0ms
Plane matching (<1Hz)	0.38ms	0ms
Plane updates (<1Hz)	0.6ms	0ms
IMU propagation	0.34ms	0.33ms
Slam and msckf update (10Hz)	10.4ms	10ms
Total time	26ms	24ms
CPU loading (%)	108	105

VI. CONCLUSION

In this paper, we present an efficient, large-scale VIO system leveraging the global information from prior point-plane map, which can exploit the available geometrical information in structured environments more efficiently. Using both an accurate plane detection based on semantic sparse clouds and a precise plane matching on the basis of point descriptor, semantic information and geometric constraints, we can detect all the points which locate on the map plane. Besides, an efficient point-plane map fusion strategy is applied to improve the localization accuracy. The effectiveness of the proposed method is extensively validated on the MAV datasets with different scenarios. Experimental results demonstrate that leveraging the plane information in the prior map can significantly improve the localization accuracy than the point map alone and the pure VIO system, especially in the environment with low texture and dominantly planar surfaces, exhibiting promising application prospects.

REFERENCES

[1] Mourikis, A. I., Roumeliotis, S. I. (2007, April). A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In ICRA (Vol. 2, p. 6).

[2] Huang, G. (2019, May). Visual-inertial navigation: A concise review. In 2019 international conference on robotics and automation (ICRA) (pp. 9572-9582). IEEE.

[3] Geneva, P., Eckenhoff, K., Lee, W., Yang, Y., Huang, G. (2020, May). Openvins: A research platform for visual-inertial estimation. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4666-4672). IEEE.

[4] Middelberg, S., Sattler, T., Untzelmann, O., Kobbelt, L. (2014, September). Scalable 6-dof localization on mobile devices. In European conference on computer vision (pp. 268-283). Springer, Cham.

[5] Lynen, S., Sattler, T., Bosse, M., Hesch, J. A., Pollefeys, M., Siegwart, R. (2015, July). Get out of my lab: Large-scale, real-time visual-inertial localization. In Robotics: Science and Systems (Vol. 1, p. 1).

[6] Lynen, S., Zeisl, B., Aiger, D., Bosse, M., Hesch, J., Pollefeys, M., ... Sattler, T. (2020). Large-scale, real-time visual–inertial localization revisited. The International Journal of Robotics Research, 39(9), 1061-1084.

[7] Ventura, J., Arth, C., Reitmayr, G., Schmalstieg, D. (2014). Global localization from monocular slam on a mobile phone. IEEE transactions on visualization and computer graphics, 20(4), 531-539.

[8] Geneva, P., Maley, J., Huang, G. (2019). An efficient schmidt-ekf for 3D visual-inertial SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12105-12115).

[9] DuToit, R. C., Hesch, J. A., Nerurkar, E. D., Roumeliotis, S. I. (2017, May). Consistent map-based 3D localization on mobile devices. In 2017 IEEE international conference on robotics and automation (ICRA) (pp. 6253-6260). IEEE.

[10] Ataer-Cansizoglu, E., Taguchi, Y., Ramalingam, S., Garaas, T. (2013). Tracking an RGB-D camera using points and planes. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 51-58).

[11] Kaess, M. (2015, May). Simultaneous localization and mapping with infinite planes. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4605-4611). IEEE.

[12] Yang, Y., Geneva, P., Zuo, X., Eckenhoff, K., Liu, Y., Huang, G. (2019, May). Tightly-coupled aided inertial navigation with point and plane features. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 6094-6100). IEEE.

[13] Hosseinzadeh, M., Latif, Y., Reid, I. (2017). Sparse point-plane SLAM. In Australasian Conference on Robotics and Automation.

[14] Xu, B., Li, X., Li, J., Yuen, C., Dai, J., Gong, Y. (2022). PVI-DSO: Leveraging Planar Regularities for Direct Sparse Visual-Inertial Odometry. arXiv preprint arXiv:2204.02635.

[15] Li, X., He, Y., Lin, J., Liu, X. (2020, October). Leveraging planar regularities for point line visual-inertial odometry. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5120-5127). IEEE.

[16] Geneva, P., Eckenhoff, K., Yang, Y., Huang, G. (2018, October). Lips: Lidar-inertial 3d plane slam. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 123-130). IEEE.

[17] Chatfield, A. B. (1997). Fundamentals of high accuracy inertial navigation (Vol. 174). Aiaa.

[18] Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., ... Quan, L. (2020). Aslfeat: Learning local features of accurate shape and localization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6589-6598).

[19] Hu, J., Hu, J., Shen, Y., Lang, X., Zang, B., Huang, G., Mao, Y. (2022, May). 1D-LRF Aided Visual-Inertial Odometry for High-Altitude MAV Flight. In 2022 International Conference on Robotics and Automation (ICRA) (pp. 5858-5864). IEEE.