Robust and Efficient Visual-Inertial Odometry with Multi-plane Priors

Jinyu Li, Bangbang Yang, Kai Huang, Guofeng Zhang, and Hujun Bao(✉)

State Key Lab of CAD&CG, Zhejiang University,
ZJU-SenseTime Joint Lab of 3D Vision, Hangzhou Shi, China
bao@cad.zju.edu.cn

Abstract. Planes commonly exist in a human-made scene and are useful for robust localization. In this paper, we propose a novel monocular visual-inertial odometry system which leverages multi-plane priors. A novel visual-inertial-plane PnP algorithm is introduced to use plane information for fast localization. The planes are expanded via a reprojection consensus-based way, which is robust to depth estimation error. A novel structureless plane-distance cost is used in sliding-window optimization, which allows to use a small size window while maintaining good accuracy. Together with modified marginalization and sliding window strategy, the computational cost is significantly reduced. Our VIO system is tested on various datasets and compared with several state-of-the-art systems. Our system can achieve very competitive accuracy, and work pretty well on long and challenging sequences. Our system is also very efficient and can perform 30 fps averagely on an iPhone 7 mobile phone with a single thread.

Keywords: Visual inertial odometry · bundle adjustment · Plane priors · Reprojection consensus · Structureless plane-distance cost

1 Introduction

Cameras and IMUs are already very common on smart mobile phones, which are small and cheap, with low power consumption. Hence they are good choices for addressing mobile localization problem in consumer level applications. Recent advances in visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) communities have give birth to many successful odometry/SLAM systems like [4,5,8,16,18]. However, these systems either require high computation cost with multiple threads or easily drift in challenging situations.

Human-made scenes generally contain rich planar structures, which can benefit odometry/SLAM. Although some methods [9,21] have been proposed to use

This work was partially supported by NSF of China (Nos. 61822310 and 61672457), and the Fundamental Research Funds for the Central Universities (No. 2018FZA5011). This work was done while Jinyu Li was a PhD student at Zhejiang University.

Z. Lin et al. (Eds.): PRCV 2019, LNCS 11859, pp. 283–295, 2019.
https://doi.org/10.1007/978-3-030-31726-3_24

plane information to aid VIO, the computation cost is obviously increased due to plane extraction and management as well as the increase of optimization complexity. In this paper, we propose a new VIO system which can effectively exploit plane structures in the scene to achieve good tracking results. The key contribution is that we propose a novel VIO approach by exploiting plane information in different modules for robust tracking. Especially, we propose a novel structureless plane-distance cost which can enforce plane constraints in sliding-window optimization without increasing much the computation cost. Thus a very robust and efficient VIO system is achieved, which can perform 30 fps averagely on an iPhone 7 mobile phone with a single thread.

2 Related Works

VIO and VISLAM have been studied over decades. MSCKF [15] is an early filtering-based VIO system. Its state vector contains only a fixed number of the pose states. Observations to the landmarks are marginalized in the update phase, and the overall computational time is bounded. Optimization-based systems like OKVIS [12] generally use marginalization technique to linearize old frames into priors, keeping the size of its sliding window bounded.

However, error accumulation in VIO is inevitable. A SLAM system can leverage loops in the trajectory to reduce error accumulation, achieving better accuracy. PTAM [11], an early visual SLAM system, separates tracking and mapping in two threads. This later became the standard of many state-of-the-art SLAM systems. ORB-SLAM [16] improves PTAM in many aspects, including the use of ORB features, the local mapping with the covisibility graph and the global optimization with the essential graph. VINS-Mono [18] is a successful visual-inertial SLAM system, which also uses sliding-window optimization with marginalization, with a 4-DoF pose-graph map optimization. To achieve real-time performance on a mobile device, its mobile version [13] limits its front-end optimization at 10 Hz.

Another type of systems track camera by minimizing the photometric error directly. Early systems such as LSD-SLAM [5] use dense or semi-dense geometry representation, which can lead to heavy computational cost. DSO [4] used a sparse and direct formulation with sliding window optimization similar to OKVIS, thus improving the performance. Due to the small/smooth movements assumption, direct methods can be prone to rolling-shutter distortions and illumination changes.

SVO [8] is a hybrid odometry system which combines direct sparse tracking with indirect formulation for model optimization. It is highly efficient and is capable of tracking at very high framerate, which can remedy the requirement of slow-smooth movement. Nevertheless, it still suffers from many limitations of direct methods.

Lines and planes can be used for robust tracking in structured environment. Many existing methods, like [9] and [17] simply augment the existing bundle adjustment (BA) with additional structure terms. Since they are built on top of typical systems, they introduce additional cost to the system, making the system obviously heavier. Some methods like StructSLAM [22] assume Manhattan scene, which may not be used in general cases. StructVIO [23] extends StructSLAM with inertial measurements, and suffer from similar limitation. Methods like [21] require additional cost to parse planes and can not handle general scenes containing normal objects and planes. In general, there is still a lack of VIO system which can efficiently make use of plane information in a general scene.

3 Visual-Inertial Odometry with Multi-plane Priors

Our framework is illustrated in Fig. 1. Given the input online images and IMU measurements, we first perform feature point tracking on consecutive images and pre-integrate the IMU measurements. We employ a visual-inertial alignment method (Sect. 3.1) to accomplish the initialization. After initialization, the feature point tracking and pre-integration results are sent into the pipeline for localization. We propose a novel visual-inertial-plane PnP (VIP-PnP) to quickly localize the camera pose (Sect. 3.2), which uses information from plane information managed in a local plane map. The output of the VIP-PnP will be integrated with new IMU measurments to get the most up-to-date pose output. After VIP-PnP, the localized frame will be fed into the sliding window, and 3D landmarks are triangulated from newly tracked features. If the last frame in the sliding window is a keyframe, we will do plane expansion via reprojection consensus (Sect. 3.3), and then slide the window, i.e. inserting this new frame and marginalizing the oldest keyframe. A local bundle adjustment is employed, where a novel structureless plane-distance cost is used (Sect. 3.4). If the last frame in the sliding window is not a keyframe, we will directly replace it with the new frame and inherit its IMU measurements. In both cases, the new planes are detected based on the landmarks and added into the local plane map. When there are no planes, all the plane-based modules are disabled, and our system becomes a traditional VIO. Hence our system is still a general purpose VIO which does not fully depend on planes.

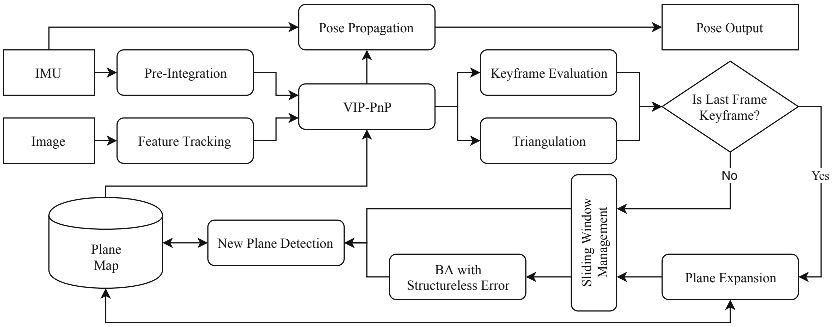

Fig. 1. Pipeline of our visual-inertial odometry with multi-plane priors.

Before describing the algorithm details, we first introduce the convention and major notations for our mathematical formulation. We represent the 3D rotation of image i using Hamiltonian quaternion q_i and use $C(q_i)$ to represent its corresponding rotation matrix. The pose of the device with respect to the world frame is denoted as ${}^w_b p_i, {}^w_b q_i$, and is affixed to the IMU of the device. Camera frame pose ${}^w_c p_i, {}^w_c q_i$ is related to this body frame by rigid extrinsics that can be calibrated beforehand.

We used the pinhole camera model with fixed camera intrinsics K throughout our system. For a landmark point x_k, its projection on camera image I_i will be

$$u_{ik} = \pi_K(C^\top({}^w_c q_i)(x_k - {}^w_c p_i)), \tag{1}$$

where π_K is the projection function with intrinsics K and the corresponding keypoint for u_{ik} is denoted as $\tilde{u}_{ik}$. We will drop the superscript/subscript and use p_i, q_i, x_k for brevity if the frame is clear from the context.

We use inverse parameterization for our landmarks [3]:

$$x_k = \frac{1}{\lambda_k} C({}^w_c q_{\mathrm{ref}(k)}) \cdot \begin{pmatrix} u_k \\ 1 \end{pmatrix} + {}^w_c p_{\mathrm{ref}(k)}. \tag{2}$$

The reference frame $\mathrm{ref}(k)$ is the first keyframe observing x_k. u_k is the shorthand for $u_{\mathrm{ref}(k),k}$.

A plane s is parameterized with its normal n_s and its (signed) distance d_s to the origin. A point x on plane s should satisfy $n_s^\top x - d_s = 0$.

3.1 Initialization and Plane Detection

For each input image, we detect keypoints with the Shi-Tomasi keypoint detector [10]. The keypoints are tracked using the KLT feature tracker [14]. The IMU measurements are pre-integrated into relative motion constraints using the method introduced in [7]. Similar to the visual-inertial alignment method from [19], we build a visual-only SfM from initial frames, and then align them with the pre-integrations to solve the initial states.

A plane detection module is responsible for spawning new planes. These new planes are then used for all the tracking and optimization, and will be extended when possible. We used a 3-point RANSAC [6] for plane detection. After each new frame is added into the sliding window, we detect the new planes from all the landmark points in the local map.

3.2 Visual-Inertial-Plane PnP Tracking

Assume there are already P plane models in the local map. For each new image I_i, we perform a visual-inertial-plane PnP (VIP-PnP) tracking to recover its pose. Let $\{x_k : k = 1 \dots M\}$ be the visible landmarks not belonging to any plane, $\{x_{sk} : k = 1 \dots M_s\}$ be the visible landmarks belonging to plane s, r_{IMU} be the IMU propagation error, Ψ and Φ be the inverse of the covariance matrix for keypoint measurement noise and IMU propagation error correspondingly. VIP-PnP is done by solving the following 1-frame bundle adjustment:

$$\underset{{}^w_b p_i, {}^w_b q_i}{\arg\min} \sum_{k=1}^{M} \|u_{ik} - \tilde{u}_{ik}\|^2_{\Psi} + \|r_{\text{IMU}}({}^w_b p_i, {}^w_b q_i)\|^2_{\Phi} + \sum_{s=1}^{P} \sum_{k=1}^{M_s} \|u^{\perp}_{i,sk} - \tilde{u}_{i,sk}\|^2_{\Psi} \tag{3}$$

In (3), the projected plane point $u^{\perp}_{i,sk}$ is obtained by forcing the landmark on the plane. We cast ray from $I_{\text{ref}(k)}$, and find the intersection depth with plane s:

$$\lambda^{\perp}_{sk} = \frac{n_s^{\top} C({}^w_c q_{\text{ref}(k)}) \binom{u_k}{1}}{d_s - n_s^{\top}\, {}^w_c p_{\text{ref}(k)}}. \tag{4}$$

Then we compute $u^{\perp}_{i,sk}$ with this depth enforced. We are solving the BA “as if” there are some points perfectly lying on some planes.

In a typical VIO, depth estimation can be noisy or even degenerated due to small camera translation, especially when the whole sliding window cannot provide sufficient motion parallax. By incorporating plane priors, the depth estimation becomes much more stable, especially when the motion parallax is small. Thus a smooth and robust tracking can be achieved even without maintaining global map and optimization.

3.3 Plane Expansion via Reprojection Consensus

We perform plane expansion when a new keyframe is pushed into the sliding window. A plane can be continuously tracked and refined over time, by keeping expanding new points. Since the triangulation error easily leads to a large error in depth, we use a reprojection consensus-based method for plane expansion. For landmark x_k and plane s, we can re-cast x_k onto the plane s according to (4). The reprojection errors without/with re-casting are computed as:

$$\epsilon_k = \sum_i \|u_{ik}(\lambda_k) - \tilde{u}_{ik}\|^2, \quad \epsilon^{\perp}_k = \sum_i \|u_{ik}(\lambda^{\perp}_k) - \tilde{u}_{ik}\|^2. \tag{5}$$

If $\epsilon^{\perp}_k \le \max\{\alpha\epsilon_k, \gamma\}$, i.e., the new reprojection error is not greater than a threshold, the x_k is thought to be consistent with plane s, and we add this landmark to the plane. In our experiments, we used $\alpha = 1.2, \gamma = 0.5$.

In order to avoid introducing large error, we do not expand distant points into a local plane area. We represent planes with 12 fan-shaped sectors, 30° each. For a sector τ, its radius r_τ is determined by the currently most distant plane point in it. A new point x_k can be added only if it is within μr_τ distance to the center. We generally set $\mu = 1.2$.

It’s worth mentioning that when the motion degenerates, a landmark at arbitrary depth can still be falsely added by the reprojection criteria. However, it also helps to keep the landmark at a reasonable depth. And these false inclusions will be pruned after the depths becoming observable under sufficient translation.

3.4 Sliding-Window Optimization with Structureless Plane-Distance Cost

We utilize a local bundle adjustment (LBA) to refine the camera poses and the landmark points in the sliding window. We keep N image frames in the sliding window. When a new frame comes, we check the parallax of its keypoint matches with respect to the last keyframe. If the parallax exceeds a threshold, we tag the new frame as a keyframe. If the number of matches is below a lower bound, or there have not been any keyframes for the recent T frames, we also mark the frame as a keyframe. After this keyframe evaluation, the new frame will be added into the sliding window. In the following, we assume $I_1, \ldots, I_N$ be the N the frames already in the sliding window, and I_{N+1} be the new one.

We slide the window with marginalization in the following way: If I_N is a keyframe, we first marginalize out I_1 and all keypoints it observes. Then we add I_{N+1} into the sliding window. If I_N is a non-keyframe, we replace it with I_{N+1} directly. The IMU measurements in between are kept and the pre-integration is updated. This particular order is different from systems like VINS-Mono, where they first add the frame, and then perform the marginalization.

As shown in Fig. 2(c–f), if the marginalization is done after the frame insertion, the result marginalization factor will contain an edge to the new frame. Next time, if this frame is not a keyframe, it will be replaced. And this edge must be marginalized again, resulting in a two-way marginalization in VINS-Mono's implementation. In our system, the marginalization is done before the insertion. As a result, the marginalization factor will constrain the oldest $N-1$ frames. No marginalization is required when replacing I_N.

Before marginalizing the oldest keyframe, we marginalize all related landmarks first, which is similar to VINS-Mono. If not, the information matrix for the related landmarks will become dense, which will significantly increase computation cost. For plannar landmarks, we replace them with the following structureless plane-distance cost, which avoids marginalization.

Structureless Plane-Distance Cost. In the core of our local bundle adjustment, we utilize a structureless plane-distance cost. Based on the linear least square triangulation method, we can triangulate a landmark x_k with all its keypoint observations $\{\tilde{u}_{ik}\}$ on images $\{I_i\}$ by constructing matrix A and vector b as:

$$A_k = \begin{pmatrix} \vdots \\ \tilde{u}_{ikx} r_{i3} - r_{i1} \\ \tilde{u}_{iky} r_{i3} - r_{i2} \\ \vdots \end{pmatrix}, \; b_k = \begin{pmatrix} \vdots \\ \tilde{u}_{ikx} p_{i3} - p_{i1} \\ \tilde{u}_{iky} p_{i3} - p_{i2} \\ \vdots \end{pmatrix}. \tag{6}$$

So x_k can be found by solving $A_k x_k = b_k$. The row vectors r_{ij} are the rows of $KC^\top({}^{w}_{c}q_i)$, i.e., $\left(r_{i1}^\top \; r_{i2}^\top \; r_{i3}^\top\right) = [KC^\top({}^{w}_{c}q_i)]^\top$. Scalars p_{ij} are the components of $-KC^\top({}^{w}_{c}q_i){}^{w}_{c}p_i$, $(p_{i1}, p_{i2}, p_{i3})^\top = -KC^\top({}^{w}_{c}q_i){}^{w}_{c}p_i$. With 2 or more observations that are not degenerated, A_k has more than 4 rows and is a full rank matrix, so we can have the least square solution for x_k.

When there is only one observation $\tilde{u}_{ik}$, or there are insufficient movements in the images, A_k will be ill-conditioned. We use plane information to regularize it: for a landmark x_{sk} belonging to a plane s, we augment the terms in (6) as:

$$A_{sk} = \begin{pmatrix} A_k \\ w_k n_s^\top \end{pmatrix}, \; b_{sk} = \begin{pmatrix} b_k \\ w_k d_s \end{pmatrix}. \tag{7}$$

The augmented row corresponds to the plane constraint $n_s^\top x_{sk} = d_s$, and is weighted by w_k. By augmenting the matrix, the solution to $A_{sk} x_{sk} = b_{sk}$ is regularized by the plane structure. As long as the camera center is not on the plane, A_{sk} is always full-rank. We can then rewrite the closed-form solution of x_{sk} as a function of the related states observing it:

$$x_{sk} = (A_{sk}^\top A_{sk})^{-1} A_{sk} b_{sk} = f(\{{}_b^w p_i, {}_b^w q_i\}, n_s, d_s). \tag{8}$$

Since the landmark x_{sk} should be on the plane, we can minimize the following plane-distance error:

$$r_P(\{{}_b^w p_i, {}_b^w q_i\}, n_s, d_s) = |n_s^\top x_{sk} - d_s|. \tag{9}$$

Although the size of A_{sk} depends on the length of the feature track. $A_{sk}^\top A_{sk}$ and $A_{sk}^\top b_{sk}$ are 3×3 and 3×1. This leads to the efficient evaluation of the cost function and its corresponding Jacobians.

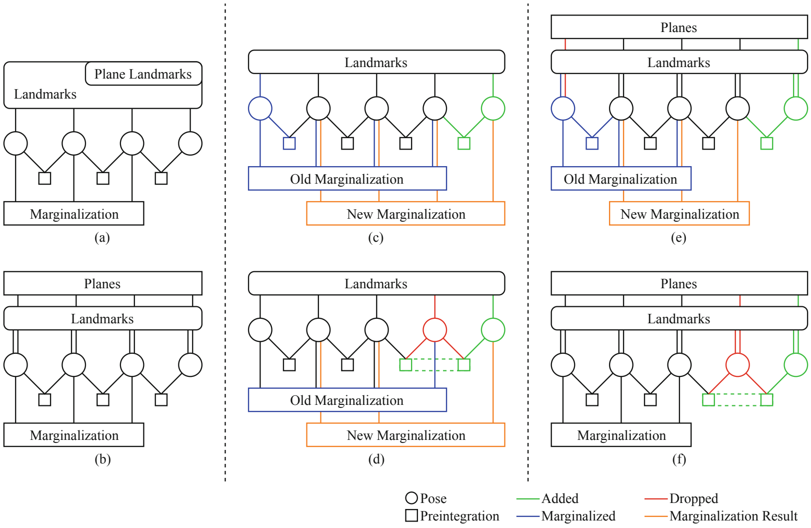

Fig. 2. Structure and marginalization in sliding-window optimization with different strategies: (a) Traditional landmark-only BA; (b) BA with our structureless cost; (c) The oldest-keyframe marginalization process in VINS-Mono; (d) The recent-non-keyframe marginalization process in VINS-Mono; (e) The oldest keyframe is being marginalized in our sliding window; (f) A non-keyframe is being replaced in our sliding window.

Figure 2(a) and (b) illustrate the structure and marginalization in sliding-window optimization. For planar landmarks, we totally remove its original reprojection error terms, and use the structureless cost instead. Given m observations of a landmark, m reprojection error terms are replaced with 1 structureless cost. The state of the plane landmark no longer participate in its corresponding structureless cost. So we can skip these landmarks in the marginalization. Plane parameters are kept fixed during BA. We re-triangulate planar points with refined camera poses after BA, and then update the planes.

4 Experiments

We implemented our system in C++ and use Ceres Solver [1] for solving nonlinear optimization problems. We run our algorithm on public benchmark datasets and evaluate the performance for quantitative results. We also make comparisons with 4 state-of-the-art odometry/SLAM systems: VINS-Mono [18], ORB-SLAM2 [16], SVO2 [8], and DSO [4]. Our system, as a plane-based VIO, will be referred as PVIO in the following.

4.1 Tracking Accuracy and Robustness

We analyze the accuracy of the algorithm by comparing the RMSE of the absolute localization error. We used the suggested configurations from the algorithms, including tuned IMU noise parameters, for the test of VINS-Mono and ORB-SLAM2. The results of SVO2 and DSO on EuRoC dataset are directly from [8]. On TUM dataset, SVO2 performed badly, which always lost quickly. PVIO used the sensor parameters from the specification of datasets, and its sliding window has $N = 8$ frames. Table 1 lists the RMSE of the odometry/SLAM systems on EuRoC [2] and a few results on TUM-VI [20] datasets. The full results on TUM-VI dataset are included in the supplementary material[1].

Accuracy. As shown in Table 1, PVIO has comparable accuracy to VINS-Mono. As for ORB-SLAM, since it does not recover true scale, we scale the camera trajectory and align it with the ground truth, which hence has lower RMSE. SVO2 and DSO are also visual only, whose recovered camera trajectories are also scaled. Despite that, we can still achieve better accuracy on many sequences. We also analyse the error accumulation on several sequences, which are included in the supplementary material due to the limited space. TUM-VI is a challenging dataset, where many sequences contain vigorous movement, and all the sequences are rather long. PVIO still achieves very competitive accuracy.

Robustness. We compare the keyframes involved in the local BA: PVIO has 8 frames, VINS-Mono has 10 frames, while ORB-SLAM2 can have as much as 30 frames. With such a small sliding window, a traditional VIO will easily have

[1] http://www.cad.zju.edu.cn/home/gfzhang/projects/SLAM/PVIO/pvio-supp.zip.

Table 1. The RMSE (m) of localization for different algorithms. "+/−Loop" means loop-closure turned on/off. For SVO2, "E+P" means edgelet+prior and "BA" means bundle adjustment. See [8] for the explanations about E+P and BA. For PVIO, "+/−Plane" means with/without plane priors. For the values in parenthesis, the corresponding trajectory is less than 80% complete. × means the trajectory is less than 50% complete (lost). The best results for visual-inertial algorithms are bolded.

Dataset		ORB-SLAM2		SVO2		DSO	VINS-Mono		PVIO	
		−Loop	+Loop	E+P	BA		−Loop	+Loop	−Plane	+Plane
EuRoC [2]	MH_01	0.02	0.03	0.10	0.06	0.05	0.16	0.15	0.19	**0.13**
	MH_02	0.03	0.03	0.12	0.07	0.05	0.18	0.26	**0.16**	0.21
	MH_03	0.17	0.05	0.41	×	0.18	0.20	**0.11**	0.31	0.16
	MH_04	0.15	0.37	0.43	0.40	2.50	0.35	0.37	0.29	**0.29**
	MH_05	0.06	0.04	0.30	×	0.11	0.30	**0.28**	0.79	0.34
	V1_01	0.03	0.03	0.07	0.05	0.12	0.09	0.10	0.10	**0.08**
	V1_02	0.15	0.03	0.21	×	0.11	0.11	0.09	×	**0.09**
	V1_03	(0.49)	0.10	×	×	0.93	0.19	0.18	×	**0.16**
	V2_01	0.03	0.03	0.11	×	0.04	0.09	0.08	0.11	**0.05**
	V2_02	0.15	0.03	0.11	×	0.13	**0.16**	0.17	×	0.20
	V2_03	(0.73)	(0.40)	1.08	×	1.16	0.29	0.37	×	**0.29**
TUM-VI [20]	Room1	×	0.10	×	×	0.06	0.07	**0.07**	1.65	0.26
	Room2	×	0.12	×	×	0.11	0.07	**0.07**	0.12	0.15
	Room3	×	(0.04)	×	×	0.12	0.12	**0.12**	0.18	0.18
	Corridor1	×	×	×	×	5.43	0.59	0.59	×	**0.23**
	Outdoors1	×	×	×	×	×	74.55	81.57	×	**22.26**

robustness problems especially when the motion parallax is insufficient. In contrast, PVIO can still track robustly. We also tried disabling all the plane-related modules. Without using plane priors, PVIO failed to track some sequences on EuRoC dataset, and diverged on almost all long sequences in TUM-VI.

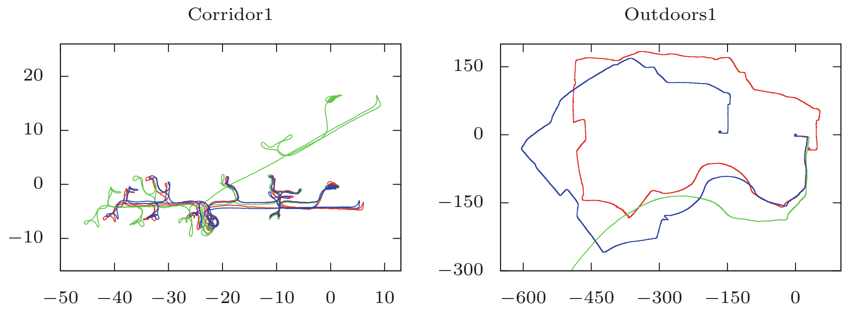

Fig. 3. Trajectories of: — PVIO, — VINS-Mono, — DSO. Axes are in meters.

TUM-VI is a very challenging dataset, where all sequences contain vigorous movement, and many of them are rather long. On almost all sequences, our system successfully tracks the data without lost or divergence. Only VINS-Mono performed better in terms of completeness. DSO diverges occasionally, and the trajectories become completely useless after divergence. ORB-SLAM is almost incapable of running on TUM-VI, which repeatedly gets lost and re-localizes in room sequences, and completely gets lost in other sequences. DSO can only track for few frames, and then fail to continue further.

The Outdoors1 sequence in TUM-VI dataset is a 2656m-long sequence. Figure 3 shows the top-down view of the trajectories from PVIO, VINS-Mono and DSO. Only VINS-Mono and PVIO can get reasonable results in this sequence. PVIO, albeit of being a VIO method, achieving 22.26m RMSE, which is smaller than 1% of the total length. VINS-Mono fails to detect loops in the end, and has significant error accumulation in its orientation estimation. PVIO, on the other hand, successfully takes advantage of the information provided by the ground plane and produces a less distorted trajectory. As a general purpose VIO, plane-priors give PVIO extra robustness, result in good accuracy.

4.2 Efficiency

With the multi-plane priors, the size of the sliding window can be effectively reduced. At the same time, the revisited marginalization strategy and the structureless cost also helped to reduce the computation time. In a canonical system, one can enforce plane constraints by adding additional point-to-plane distance error to the bundle adjustment. We also implemented such bundle adjustment, and name the corresponding VIO system as Ref-VIO. We run VINS-Mono, Ref-VIO and PVIO on the same computer with i7-7700 3.6GHz×4 and 16G memory. We also measure the running time for different parts of the systems. We set the sliding window size of VINS-Mono to 8 frames, and also disable its backend. So three systems have fair competition. Table 2 shows the running time of different components on sequence V1_01_easy.

Table 2. Running time (ms) of VINS-Mono (frontend), Ref-VIO and PVIO.

Module	VINS-Mono	Ref-VIO	PVIO
Keypoint Tracking	8.34	7.42	7.40
Pre-Integration	0.44	0.04	0.04
Plane Management	–	1.04	1.09
Non-Keyframe PnP	17.78	0.93	0.90
Non-Keyframe Marg	0.68	–	–
Keyframe BA	19.18	30.59	19.87
Keyframe Marg	32.91	3.26	2.81
Keyframe Average	60.87	42.35	31.02
All Frames Average	44.72	14.80	13.53

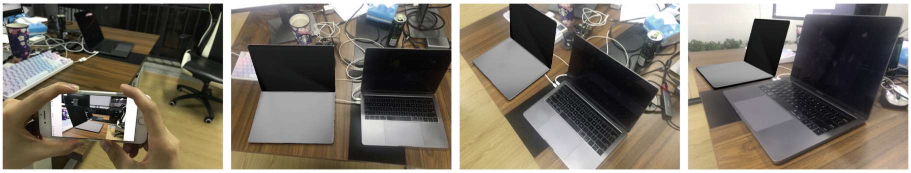

Fig. 4. AR effect on a mobile phone. A virtual "laptop" is placed next to the real one.

As we can see, if we directly use point-to-plane distance in BA, the computation cost will significantly increase. By replacing traditional reprojection error with structureless plane-distance cost, the keyframe BA in PVIO takes almost the same time as the normal BA in VINS-Mono. In the meantime, VINS-Mono uses a 3-frame BA with older frames fixed in its non-keyframe PnP, while PVIO only solves 1 frame, without involving any historical frames. The modified marginalization strategy also significantly reduce the computation time. Summing up all the accelerations, the keyframe processing time of PVIO is only 1/2 of VINS-Mono, and all frames average is taking less than 1/3 of VINS-Mono.

To further verify the efficiency of PVIO, we successfully run PVIO on an iPhone 7 mobile phone. The image is captured at 640×480 (30fps), while IMU is incoming at 100 Hz. The whole system runs in a single thread, and can perform metric tracking and AR on the camera image. The average speed can reach 30fps. Figure 4 shows the AR effect in our demo App.

5 Conclusions and Disscusions

We presented a new robust and efficient VIO system, which exploits multi-plane priors in the tracking and the local mapping. With the design of the structureless plane-distance cost, we can incorporate multi-plane prior constraints into bundle adjustment without introducing much computation cost. Compared to other state-of-the-art systems, our proposed VIO system can get competitive accuracy. Even on long and challenging sequences, our system can track successfully, whereas many other systems fail. Especially, our VIO system is very efficient and requires much less computation cost compared to the complex SLAM systems such as ORB-SLAM and VINS-Mono. Our VIO can perform in real-time even on an iPhone 7 with a single thread. To further improve the robustness and efficiency of our VIO system, we would like to explore the possibilities in using more structure information in the future.

References

1. Agarwal, S., Mierle, K., Others: ceres solver. http://ceres-solver.org
2. Burri, M., et al.: The EuRoC micro aerial vehicle datasets. Int. J. Rob. Res. **35**(10), 1157–1163 (2016)

3. Civera, J., Davison, A., Montiel, J.: Inverse depth parametrization for monocular SLAM. IEEE Trans. Rob. **24**(5), 932–945 (2008)
4. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018)
5. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
7. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: On-manifold preintegration for real-time visual-inertial odometry. IEEE Trans. Rob. **33**(1), 1–21 (2017)
8. Forster, C., Zhang, Z., Gassner, M., Werlberger, M., Scaramuzza, D.: SVO: semidirect visual odometry for monocular and multicamera systems. IEEE Trans. Rob. **33**(2), 249–265 (2017)
9. Lee, G.H., Fraundorfer, F., Pollefeys, M.: MAV visual SLAM with plane constraint. In: IEEE International Conference on Robotics and Automation, pp. 3139–3144. IEEE, Shanghai, May 2011
10. Shi, J.T.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600. IEEE Computer Society Press, Seattle (1994)
11. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 1–10. IEEE, Nara, November 2007
12. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. Int. J. Rob. Res. **34**(3), 314–334 (2015)
13. Li, P., Qin, T., Hu, B., Zhu, F., Shen, S.: Monocular visual-inertial state estimation for mobile augmented reality. In: IEEE International Symposium on Mixed and Augmented Reality, pp. 11–21. IEEE, Nantes, October 2017
14. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI 1981 , vol. 2, pp. 674–679. Morgan Kaufmann Publishers Inc. (1981)
15. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint kalman filter for vision-aided inertial navigation. In: IEEE International Conference on Robotics and Automation, pp. 3565–3572. IEEE, Rome, April 2007
16. Mur-Artal, R., Tardos, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017)
17. Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F.: PL-SLAM: Real-time monocular visual SLAM with points and lines. In: IEEE International Conference on Robotics and Automation, pp. 4503–4508. IEEE, Singapore, May 2017
18. Qin, T., Li, P., Shen, S.: VINS-Mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Rob. **34**(4), 1004–1020 (2018)
19. Qin, T., Shen, S.: Robust initialization of monocular visual-inertial estimation on aerial robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4225–4232. IEEE, Vancouver, September 2017

20. Schubert, D., Goll, T., Demmel, N., Usenko, V., Stueckler, J., Cremers, D.: The TUM VI benchmark for evaluating visual-inertial odometry. In: International Conference on Intelligent Robots and Systems, October 2018
21. Yang, S., Song, Y., Kaess, M., Scherer, S.: Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1222–1229. IEEE, Daejeon,October 2016
22. Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., Yu, W.: StructSLAM: visual SLAM with building structure lines. IEEE Trans. Veh. Technol. **64**(4), 1364–1375 (2015)
23. Zou, D., Wu, Y., Pei, L., Ling, H., Yu, W.: StructVIO : visual-inertial odometry with structural regularity of man-made environments. arXiv:1810.06796 [cs], October 2018