2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Vanishing Point Aided LiDAR-Visual-Inertial Estimator

Peng Wang[1], Zheng Fang[1*], Shibo Zhao[2], Yongnan Chen[1], Ming Zhou[1],Shan An[3]

***Abstract*— In this paper, we propose a vanishing point aided LiDAR-Visual-Inertial estimator to achieve real-time, low-drift and robust pose estimation. The proposed method is mainly composed of 3 sequential modules, namely IMU-aided vanishing point (VP) detection module, voxel-map based feature depth association module, and visual inertial fixed-lag smoother module. The IMU-aided VP detection module will detect feature points, line segments and vanishing points to establish robust correspondences in successive frames. In particular, we propose to use 1-line RANSAC method to provide stable VP hypotheses and polar grid to accelerate vanishing point hypothesis validation. After that, we propose a novel voxel-map based feature depth association method, to retrieve depth and assign depth to visual feature efficiently. Finally, the visual inertial fixed-lag smoother module is proposed to jointly minimize error terms. Experiments show that our method outperforms the state-of-the-art visual-inertial odometry and LiDAR-visual estimator in both indoor and outdoor environments.**

I. INTRODUCTION

Robust and accurate state estimation is a challenging and fundamental problem for many applications such as autonomous driving, unmanned aerial vehicles (UAVs) and augmented reality. In recent years, visual-inertial odometry (VIO) [1] has become a very attractive method since it can achieve real-time performance on power and memory constrained devices and provide relatively robust and accurate pose estimation. However, current VIO methods are still far from perfect. For example:

- Suffer from scale drift: In visually or geometrically degraded environments such as long corridors or low-texture scenes, the depth estimation accuracy from triangulation method is very limited, which will cause large drifts in state estimation.
- Suffer from rotation drift: Due to lack of global observation [2]–[4], especially in yaw direction, the monocular VIO system will drift inevitably, which will influence its accuracy performance.

For the scale drift problem, some researchers [5]–[7] use laser scanner to assign depth directly to visual features. However, these methods did not fuse IMU measurement, which is difficult to overcome aggressive motion. Besides, the process of assigning depth is time-consuming, making it difficult to meet real-time requirement on embedded system.

[1] Peng Wang, Zheng Fang, Yongnan Chen and Ming Zhou are with Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China; Corresponding author: Zheng Fang, Email:fangzheng@mail.neu.edu.cn

[2] Shibo Zhao is with Robotics Institute, Carnegie Mellon University, USA, {shiboz}@andrew.cmu.edu

[3] Shan An is with Tech & Data Center, JD.COM Inc.

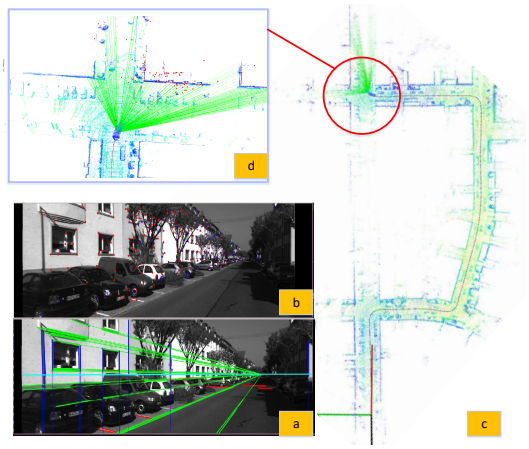

Fig. 1. (a) vanishing point detection result. (b) visual feature point tracking result. (c) LiDAR map and trajectory result. (d) voxel map based feature depth association result.

For the rotation drift problem, most of current works [2], [8] use loop closure strategy to solve it. However, these methods have two major disadvantages. First, it requires mobile robot to revisit previous-visited place. Thus, the drift of pose estimation cannot be eliminated in time. Second, when the system detects loop closure in large-scale environments, the system needs to allocate large computing resources and time to optimize the historical trajectory.

Motivated by the discussion above, we propose a robust and real-time monocular visual-inertial odometry framework which can make full use of laser depth information and vanishing point constraints. The method adopts voxel-map data structure to assign LiDAR depth to visual features efficiently. Besides, the vanishing points provide global rotation constraints to effectively reduce rotation drift. Fig. 1 shows some illustrative results of our proposed method. We carried out various indoor and outdoor experiments to show that our method outperforms the state-of-the-art visual-inertial odometry and LiDAR-visual estimator.

The main contributions of the paper are as follows:

- We propose the first vanishing point aided LiDAR-visual inertial estimator, which utilizes both laser depth and vanishing point information to achieve robust pose estimation in visually and geometrically degraded environments.
- We propose a novel voxel-map based feature depth association module, which can assign depth information to visual features efficiently.
- We propose a novel vanishing point detector pipeline, which can detect vanish point robustly and efficiently.

II. RELATED WORK

A. Visual-Inertial Odometry

In recent years, many promising methods have been proposed in the VIO area. Nevertheless, most VIO systems [2], [9] only utilize point features to achieve pose estimation. However, point detection is difficult to work well in texture-less and illumination-changing environments. In contrast, line segments are robust in these scenes. For VIO approaches, Kottas and Roumeliotis [10] adopt line features to achieve VIO system. Kong et al. [11] combines point and line features to build a stereo VIO system by utilizing trifocal geometry. Yijia [12] proposes PL-VIO method which integrates line features and point features into the optimization framework. However, these works do not adopt vanishing point, which results in not fully making use of line features. Besides, in man-made environments, some works [13], [14], [15] use the structural information such as line, plane, vanishing point (VP) to achieve robust estimation. However, these works do not utilize depth information from laser scanner and are difficult to initialize successfully in visually-degraded environments.

B. Depth Utilization in VIO System

In visual-inertial odometry system, depth estimation is very important and it decides the accuracy of metric scale. According to different sensors, the depth estimation can be obtained by different ways. For monocular based VIO system [2], a popular approach is to use triangulation to achieve depth estimation. However, such methods are difficult to work well in low-texture environments and cannot provide accurate scale estimation. For stereo based VIO system [8], researchers leverage stereo matching with a fixed baseline for depth estimation. While stereo VIO delivers more reliable depth estimation, it requires self-calibration for long-term operation [16], [17]. For RGBD based VIO system, some works [18] directly assign depths to each feature point via depth image. However, RGB-D cameras cannot work well in outdoor environments [19]. To solve above problems, DEMO [5], LIMO [6] and PL-LOAM [7] use laser scanner to provide depth information for visual odometry (VO) system. However, since these methods need to recreate KD-tree [20] for every single frame, it is very computationally expensive. DVL-SLAM [21] directly projects depth information on the image frame, which is fast and efficient. However, this method only uses single LiDAR scan to provide depth information, which is difficult to provide accurate depth estimation for all visual features.

C. Vanishing Point detection

In order to better constrain the rotation drift, [22] and [23] adopt vanishing point to provide better measurements of rotation. However, these methods have four limitations. Firstly, they use all line segments on the image to detect vanishing points. Secondly, they need to generate many hypotheses by applying 2,3 lines RANSAC [24], [25] methods. Thirdly, the response of one line segment needs to be calculated repeatedly. Fourthly, these methods do not use IMU information as prior, it is a very time-consuming process. To solve above problems, we propose a novel IMU-aided vanishing point detector pipeline, which adopts 1-line RANSAC method to provide more stable VP hypotheses and uses the polar grid to accelerate vanishing point hypotheses validation.

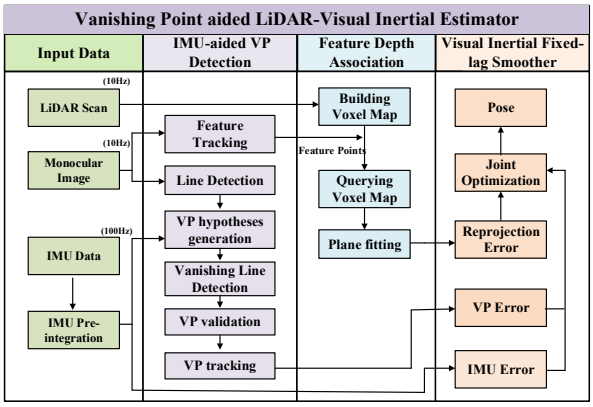

Fig. 2. Overview of Vanishing Point aided LiDAR-Visual Inertial Estimator

III. METHODOLOGY

In this section, we will introduce the pipeline of our system shown in Fig. 2 . It is composed of three sequential modules, namely IMU-aided vanishing point detection (SecIII-A), feature depth association (SecIII-B) and visual inertial fixed-lag smoother (SecIII-C). The visual inertial fixed-lag smoother module will jointly minimize visual re-projection error terms of tracked features and IMU error terms based on IMU pre-integration and VP constraints error terms.

A. IMU-aided VP Detection

1) Point feature detection and tracking: we adopt sparse optical flow method [26] to track visual features efficiently. When the number of tracked features is less than a threshold, we will detect new features by using Shi-Tomashi Corner detector method [27].

2) Line segment detection: we detect line segments with LSD [28] on the undistorted image. Then, we calculate line segments' start points, end points, lengths and angles.

3) Gravity constrained Horizon line detection: The horizon line detection relies heavily on the accuracy of the camera's rotation. If the horizon line detection is inaccurate, accurate horizontal VP hypothesis cannot be generated. Previous work directly [15] uses IMU rotation to calculate horizon line, which may not work well. In order to solve this problem, we use vertical lines based on gravity constraints to refine IMU rotation.

As shown in Fig.3, we first get 2D projection vector $\mathbf{Z}_{c_i} \in \mathbb{R}^2$ of the Z-axis in the world frame (consistent with the direction of gravity) on the middle point of every line segment according to current camera orientation $\mathbf{R}_{\mathrm{w}}^{\mathrm{c}} \in \mathrm{SO}(3)$. Secondly, we classify the line l_i by the angle between this line and its corresponding $\mathbf{Z}_{c_i}$. If the angle is below a

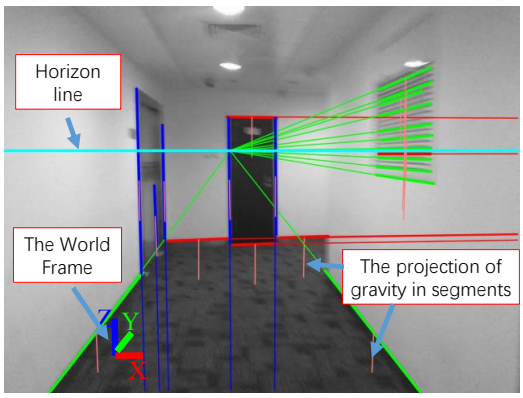

Fig. 3. Illustration of VP detection. Thick red, green and blue line segments are correspond to the X, Y and Z direction of the world frame respectively. A cyan line represents the horizon line. The pink lines represent the projection of the world's Z direction at the midpoint of each extracted line segment.

threshold, we assume this line is in the Z direction. Ideally, these vertical line segments pass through projection of $\mathbf{V}_{\mathrm{z}}^{\mathrm{c}}$ vanishing point of Z direction($\mathbf{V}_{\mathrm{z}} = [0,0,1]^T$) in homogeneous coordinate. So we can refine the $\mathbf{V}_{\mathrm{z}}^{\mathrm{c}}$ by minimizing the distance between $\mathbf{V}_{\mathrm{z}}^{\mathrm{c}}$ and vertical line segments. Then, we calculate unit vector of $\mathbf{V}_{\mathrm{z}}^{\mathrm{c}}$ as $\mathbf{n}_{\mathrm{vz}}^{c} = \mathbf{K}^{-1}\mathbf{V}_{\mathrm{z}}^{\mathrm{c}} / \|\mathbf{K}^{-1}\mathbf{V}_{\mathrm{z}}^{\mathrm{c}}\|$. Finally, we can obtain optimal orientation $\mathbf{R}_{\mathrm{w\_opt}}^{\mathrm{c}} \in \mathrm{SO}(3)$ between $\mathbf{V}_{\mathrm{z}}$ via Eq.(1) and $\mathbf{n}_{\mathrm{vz}}^{\mathrm{c}}$ and optimal horizon line. $\mathbf{HL} = \mathbf{K}^{-T}\mathbf{R}_{\mathrm{w\_opt}}^{c}[0,0,1]^T$, where $\mathbf{K}$ is intrinsic matrix of camera.

$$\mathbf{n}_{\mathrm{vz}}^{\mathrm{c}} = \mathbf{R}_{\mathrm{w}}^{\mathrm{c}}\, \mathbf{V}_{\mathrm{z}} \tag{1}$$

4) VP hypotheses generation: In order to efficiently generate vanishing point hypotheses of X and Y directions with given horizon line $\mathbf{HL}$, We use 1-line RANSAC scheme [15] to select horizon line, and intersect it with horizon line to generate a intersection point $\mathbf{V}_{\mathrm{x}}^{\mathrm{c}}$ and its corresponding vanishing point hypothesis in X direction $\mathbf{n}_{\mathrm{vx}}^{c} = \mathbf{K}^{-1}\mathbf{V}_{\mathrm{x}}^{\mathrm{c}} / \|\mathbf{K}^{-1}\mathbf{V}_{\mathrm{x}}^{\mathrm{c}}\|$ via Eq.(2).

$$\mathbf{V}_{x}^{c} = \mathbf{K}\,\mathbf{n}_{vx}^{c} = \mathbf{K}\,\mathbf{R}_{w}^{c}\,\mathbf{n}_{vx}^{w} = \mathbf{K}\,\mathbf{R}_{w}^{c}\left[\cos(\Phi), -\sin(\Phi), 0\right]^T \tag{2}$$

where $\mathbf{n}_{vx}^{w}$ is the representation of $\mathbf{n}_{vx}^{c}$ in world frame and it denotes a dominant horizontal direction of a local manhattan world w.r.t world frame. Φ is the relative angle between $\mathbf{n}_{vx}^{w}$ and the X axis of world frame.

Once vanishing point of X direction is obtained, the vanishing point of Y direction can be obtained as $\mathbf{n}_{\mathrm{vy}}^{\mathrm{c}} = \mathbf{n}_{\mathrm{vz}}^{\mathrm{c}} \times \mathbf{n}_{\mathrm{vx}}^{\mathrm{c}}$ considering the orthogonal constraint of vanishing point.

5) Response polar grid: To validate vanishing point hypotheses in the next step, the common practice is to reverify the geometry consistency between vanishing points and line segments, namely the response strength of line segments. And this process is time-consuming. However, we can transform this process into Hough Transform space, and build a VP response map in advance, namely response polar grid. Then, we will obtain the corresponding responses by querying polar grid at constant time to reduce the cost of computing. The implement of polar grid is as follows:

Given a point p on the image plane, we can get its corresponding ray polar coordinate (ϕ, λ) according to Eq. (3) and (4):

$$[X,Y,Z]^T = \left[\tfrac{x-x_0}{f_x}, \tfrac{y-y_0}{f_y}, 1\right]^T \tag{3}$$

$$\begin{bmatrix}\phi \\ \lambda\end{bmatrix} = \begin{bmatrix}\arccos(Z/\sqrt{X^2+Y^2+Z^2}) \\ \mathrm{atan2}(X,Y)+\pi\end{bmatrix} \tag{4}$$

where the intervals of ϕ and λ are $[0,\pi/2)$ and $[0,2\pi)$ respectively. Finally, we update the corresponding response of grid cell $G(\phi_{deg}, \lambda_{deg})$ with following equation:

$$G(\phi_{deg}, \lambda_{deg}) = G(\phi_{deg}, \lambda_{deg}) + \|l_1\|\,\|l_2\|\,\cos(\theta - \frac{\pi}{4}) \tag{5}$$

where $\phi_{deg} = [\phi \times 180/\pi]$, $\lambda_{deg} = [\lambda \times 180/\pi]$, $\|l\|$ indicates the length of line segment l and θ stands for the minimum angle between l_1 and l_2.

From Eq.(5), it can be seen that the horizontal line segments pair with longer length and appropriate orientation have higher weight. If the angle between the lines is too small, the intersection point near two parallel lines would be unstable. If the angle is too large, it indicates that the two lines belong to different structural directions.

6) VP validation: For each horizontal vanishing point hypothesis n_{vx}^{c}, n_{vy}^{c}, we first obtain its polar coordinates (ϕ_x, λ_x), (ϕ_y, λ_y) via Eq.(5) and (4). Then, we set the sum of value of $G(i,j)$ at corresponding position of (ϕ_x, λ_x), (ϕ_y, λ_y) as the response of hypothesis. Finally, according to angle between the line l_i and the line passing l_i's middle point from the most supported vanishing points with the greatest responses, we classify horizontal line segments into X and Y direction line sets L_x, ,L_y.

7) VP Refinement: Inspired by method of [15], we use supported line sets L_x and L_y to refine vanishing point of X direction by minimizing the following constraint equation:

$$E(\mathbf{n}_{\mathrm{vx}}^{c}) = \sum_{i \subseteq L_x} (\mathbf{n}_{\mathrm{vx}}^{c} \cdot \mathbf{u}_{\mathrm{i}})^2 + \sum_{j \subseteq L_y} \left((\mathbf{n}_{\mathrm{vx}}^{c} \times \mathbf{n}_{\mathrm{vz}}^{c}) \cdot \mathbf{u}_{\mathrm{j}}\right)^2 \tag{6}$$

where u_{i} and u_{j} stand for a line represented in normalized image coordinate of corresponding line in L_x and L_y respectively.

8) VP Tracking: In order to deal with multiple Manhattan scenes, where there are multiple dominant vanishing points of X direction, we use method [15] to associate vanishing points with different frames.

B. Voxel-map Based Feature Depth Association

The depth estimation for each 3D point landmark plays an essential role in the accuracy of the VIO system. However, during the initialization and data association process, it's difficult for most of the VIO methods [2] to provide an accurate scale of 3D landmarks when the IMU module is not fully stimulated or the triangulation process is unstable. This results in the drift of pose estimation. To solve this problem, we propose a novel depth association method which can accurately and efficiently assign depth information for each feature point. The specific procedures can be divided into the following steps:

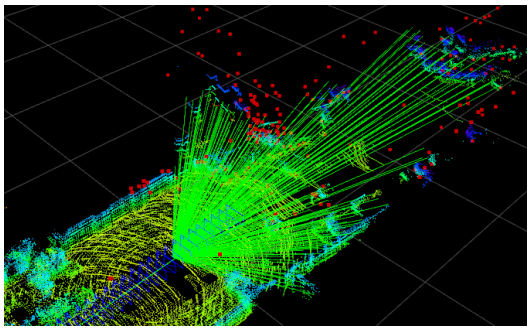

Fig. 4. Obtaining depths of visual feature points (red point) by using ray casting method.

1) Constructing depth map: During the initialization process, we project the current LiDAR scan into the camera frame and obtain its corresponding projection points on the image plane to construct a depth map. After successful initialization process, we register each laser scan into the world frame and build local depth map by using corresponding estimated pose.

2) Assigning depth to visual features: In order to fast retrieve LiDAR points in constant time, we adopt voxel hash structure [29] to store the depth map as a hash table and assign depth in a ray-casting manner. More details as follows:

- Building voxel-map structure: After obtaining the local depth map, we store the local map as a voxel-map representation based on the work of [29]. Specifically, the local depth map is comprised of voxels and each allocated voxel stores a list of 3D LiDAR points and corresponding positions in the world coordinate.
- Querying voxel-map: To assign depth to a given feature point, we use raycasting method and hash value to find its corresponding voxel. If the hash entry exists, the voxel-map will return the pointer to the specific voxel, which contains exactly the corresponding 3D laser points shown as Fig 4. We will use these 3D laser points to calculate the depth of each given visual feature point.
- Plane degradation detection: After obtaining neighboring points, we use the plane fitting method to assign depth. However, some of the neighboring points set $\mathbf{S}$ retrieved from raycasting method may be on the same straight line, resulting in unstable plane fitting. In order to avoid this, we select two points p_1,$p_2 \in \mathbb{R}^3$ from this line. Then, we calculate the non-parallelism $c = \|\sum_{i \subseteq \mathbf{S}} (p_1 - p_2) \times (p_i - p_1)\|$. If c is below a certain threshold, we regard these points are on a same line. The following step will not be processed.
- Assign depth by using plane fitting: We will calculate the local planes corresponding to these neighboring points. After that, we intersect the local planes with the feature rays to obtain features' corresponding depth information.

C. Visual Inertial Fixed-lag Smoother

To precisely estimate the camera pose in real time, we adopt tightly-coupled fixed-lag smoother to jointly minimize IMU error terms, reprojection error terms, and vanishing point error terms.

1) IMU Error: To deal with high frequency IMU measurements between frame k and frame $k+1$, we pre-integrate several consecutive IMU measurements via IMU preintegration method [2] into a pseudo-measurement $\Delta\mathbf{r} = (\hat{\alpha}^{b_k}_{b_{k+1}}, \hat{\beta}^{b_k}_{b_{k+1}}, \hat{\gamma}^{b_k}_{b_{k+1}})$. Then the residuals $\mathbf{e}_{\text{imu}}$ can be formulated as:

$$\mathbf{e}_{\text{imu}} = \left[\delta\alpha^{b_k}_{b_{k+1}}, \delta\beta^{b_k}_{b_{k+1}}, \delta\theta^{b_k}_{b_{k+1}}, \delta\mathbf{b}_a, \delta\mathbf{b}_g\right]^T \tag{7}$$

2) Reprojection Error: When a landmark $\mathbf{l}^{w_i}$ in the world frame is detected in the camera frame c_j at image coordinates $\mathbf{p}^{c_j}$, the residual can be described as:

$$\mathbf{e}_{\text{reproj}} = \mathbf{p}^{c_j} - \mathbf{K}\left(\mathbf{T}^c_b\, \mathbf{T}^b_w \mathbf{l}^{w_i}\right) \tag{8}$$

where $\mathbf{T}^b_w$ is the transformation from world frame w to IMU body frame b. $\mathbf{T}^c_b$ is the transformation from camera frame c to IMU body frame b. If there exists depth prior for this landmark, we will assign a depth prior to this landmark during optimization.

3) Vanishing Point Error: To provide global orientation constraints for camera pose, we need vanishing point of X direction to constrain the camera's yaw. To avoid its over-parameterization during optimization, we use a parameter Φ [30] to represent it according to Eq.(2). We assume it is observed in the camera frame i and the observation is $\mathbf{n}^c_{vxi}$. Then the corresponding vanishing point error can be formulated as:

$$\mathbf{e}_{\text{vanish}} = \mathbf{n}^c_{vxi} \times (\mathbf{R}^{ci}_w \mathbf{n}^w_{vxi}) \tag{9}$$

4) Joint Cost Function: For each new frame, we minimize a joint cost function that consists of IMU terms $\mathbf{e}_{\text{imu}}$, reprojection terms $\mathbf{e}_{\text{reproj}}$, vanishing point terms $\mathbf{e}_{\text{vanish}}$ and a marginalization prior $\mathbf{e}_{\text{m}}$.

$$\begin{aligned}\mathbf{E} = &\sum_{(i,j)\in\mathbf{C}} \mathbf{e}^T_{\text{imu}} W^{-1}_{\text{imu}} \mathbf{e}_{\text{imu}} + \sum_{i\in P}\sum_{j\in obs(i)} \mathbf{e}^T_{\text{reproj}}\, W^{-1}_{\text{reproj}} \mathbf{e}_{\text{reproj}} \\ &+ \sum_{i\in VP}\sum_{j\in obsvp(i)} \mathbf{e}^T_{\text{vanish}} W^{-1}_{\text{vanish}} \mathbf{e}_{\text{vanish}} + \mathbf{e}_{\text{m}}\end{aligned}$$

The set $\mathbf{C}$ contains pairs of frames which are connected by IMU factors. For each landmark P_i, $obs(i)$ represents a set that contains the observations of this landmark from other frames. And for each vanishing point of X direction $VP(i)$, $obsvp(i)$ represents a set that contains the observations of this vanishing point from other frames. W_{reproj}, W_{imu} and W_{vanish} are co-variance matrix for visual and IMU measurements. Through optimizing this cost function, we can obtain the pose of sensor.

IV. Experiments

The software runs on a laptop computer with 2.3GHz i5 Intel processor, in a Linux system running Robot Operating System (ROS). To thoroughly evaluate and validate the performance of the proposed approach, a variety of experiments were performed on both indoor and outdoor datasets.

A. Robustness and Run-time Comparison of Vanishing Point Detector

To verify the performance of proposed vanishing point detector, we compare our method with state-of-art vanishing point detector method [15] in a long corridor environment with less texture as shown in Fig.5. We first estimate vanishing points of Z, then draw a ray from the vanishing point of X direction to the middle point of each line segment. Compared with [15], our method has a smaller angle between this ray and the line segment marked by the yellow rectangular. It indicates that our algorithm can provide more accurate and robust vanishing point estimation. This is because we use the optimized rotation matrix $R^c_{w\_opt}$ to calculate the horizon line, resulting more accurate vanishing point hypotheses. The more detailed illustration can be found in caption in Fig.5. Also, we compared the real-time performance of our vanishing point algorithm and [15]. Our method only takes 16 ms, while method in [15] takes 33.3ms.

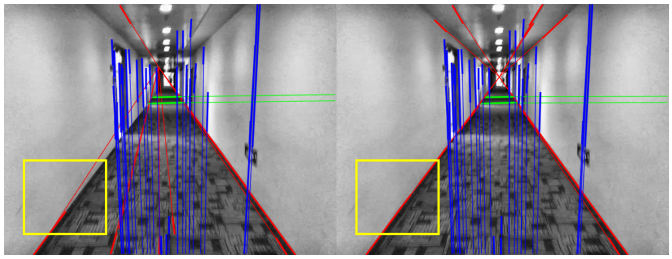

Fig. 5. The performance comparison of [15] (left sub image) and our VP detection (right sub image) method in a long corridor environment. The thick red, green and blue extracted line segments represent the X, Y, Z directions of the vanishing point. The thin lines represent the rays from the midpoint of the lines segment to the corresponding vanishing points. The areas marked by the yellow rectangular are used to compare the angle between line segment and the line through the middle point of the segment.

B. The Comparison of Voxel-map Based Depth Association

In this section, we will compare the real-time performance of voxel-map based depth association method with KD-tree method [20]. We use the voxel filter to build a local voxel map from recent sequential laser scans. In this experiment, the resolution of voxel filter is 0.2m, the voxel grid size for storing voxel map is 2m, the size of voxel map is 100m, and the number of visual feature points that need to be associated depth is 300.

Based on the local voxel map, we query the depths of visual feature points by KD-tree method and our method respectively. The experiment results are shown in Fig.6. It can be seen that when querying the depth of the same number of feature points, the KD-tree method needs to consume about 40-100ms, while our method only consumes about 5ms. This is because that KD-tree method needs to recreate data structure for every querying voxel map. This process will consume a lot of time as the size of LiDAR map increases. In contrast, ray casting method only needs to process the depth of visual features. Its data is much smaller compared with KD-tree based method. Thus, it greatly improve the real-time performance.

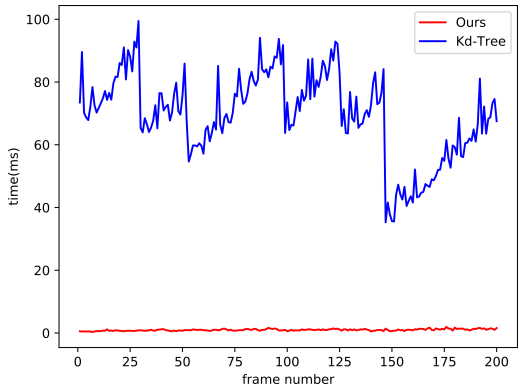

Fig. 6. Comparison of time taken for querying the map as frame number increases.

C. Pose Estimation Performance

1) Evaluation on indoor dataset: We used a custom-built VIO system including an RGB camera with resolution of 640×480 and 100Hz IMU to collect sensor data in a long and textureless corridor. Due to lack the motion capture system, we use the center line of the corridor as the groundtruth. In the experiment, we held the sensors and walked back and forth for three times along the center line of corridor as shown in Fig.5. The width of the corridor is 2m and the length of the trajectory is about 120m. We compared our method with other two state-of-the-art VIO algorithms VINS [2], PL-VINS [13]. We also tested ORB-SLAM3 [31] on our dataset. Since it failed multiple times, ORB-SLAM3 trajectory was excluded from the plot.

The trajectory comparison is shown in Fig.8. We can see that the lateral drift of our method is much smaller than VINS [2] and PL-VINS [13] algorithm, which means our method can provide more accurate yaw estimation.

Fig.9 shows the trajectory error of different methods. From this figure, we can find that error of VINS [2] (red line) increases as the number of frames grows. Although PL-VINS [13] reduces drift by using line features, the error is still big. In contrast, the drift of our method is the smallest, which indicates that using vanishing points can effectively improve the accuracy of state estimation.

2) Evalution on Outdoor dataset: To validate the outdoor performance of our method, we used KITTI [32] dataset, which contains various environments(e.g., urban, highway, and streets). And because the IMU data is discontinuous on other KITTI sequences, we only compared the proposed method with VINS [2] and DEMO [5] and DVL-SLAM [21] on KITTI sequences 02, 07 and 08. For a fair comparison, we disabled explicit loop-closure detection for VINS and DVL-SLAM. We evaluated the accuracy of these methods by using the absolute trajectory error (ATE). Fig.7 presents the intuitive trajectory comparison between ours and other algorithms on KITTI datasets. Table I lists the pose estimation accuracy of the proposed method and other algorithms. It can be seen that our method outperforms these state-of-the-art methods on all sequences. On KITTI02 and KITTI08 sequences, the root mean square (RMSE) of ATE is reduced

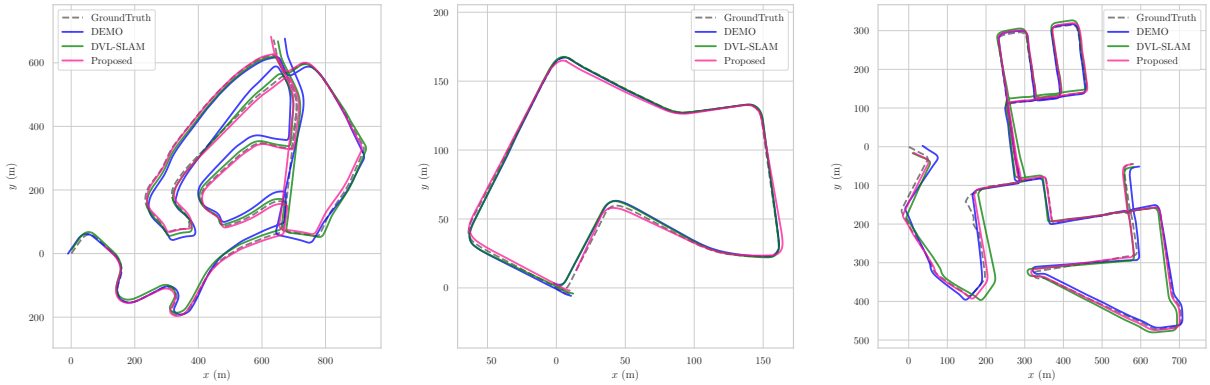

Fig. 7. Trajectories of our method (pink), DEMO (blue) and DVL-SLAM (green) tested on KITTI dataset. The trajectories of groudtruth are drawn in dashed black lines. The left subimage is result of KITTI 02. The middle subimage is result of KITTI 07. The right subimage is result of KITTI 08.

TABLE I

ATE OF THE ESTIMATED TRAJECTORY IN METERS ON THE KITTI DATASET FOR SEVERAL DIFFERENT METHODS

	ATE(in m) Transl. MAX				ATE(in m) Transl. RMSE				ATE(in m) Transl. MIN			
Sequences	DVL	DEMO	VINS	Proposed	DVL	DEMO	VINS	Proposed	DVL	DEMO	VINS	Proposed
KITTI 02	27.003	49.559	fail	**14.545**	14.428	27.086	fail	**7.9829**	1.9364	5.7977	fail	**1.2840**
KITTI 07	**4.871**	5.501	6.629	6.031	**2.363**	2.532	2.401	2.386	**0.2728**	0.9547	0.5748	0.602
KITTI 08	67.552	67.694	fail	**61.113**	17.217	17.531	fail	**9.555**	2.4970	4.3257	fail	**0.9207**

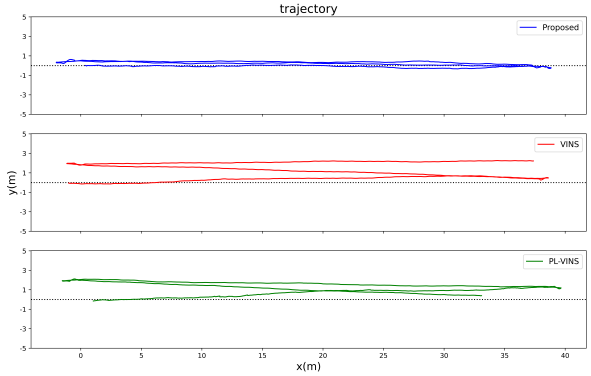

Fig. 8. XY view of indoor sequence trajectories. It shows the trajectories resulted from proposed method (bule), VINS (red), and PL-VINS (green). The black dotted line in the figure represents the centerline of the corridor, and the trajectory provided by our method (blue line) is closer to the centerline of the corridor.

by almost 50%. The reason behind this is that we fuse IMU measurement in our state estimation system, which allows us to provide more accurate rotation estimation. Since VINS does not use depth estimation from laser scanner, it failed in initialization process.

V. CONCLUSION

In this paper, we presented an accurate and efficient LiDAR-visual inertial estimator which fully exploits the LiDAR depth and vanishing point. To improve the accuracy and real-time performance of VP detection, we employed an IMU-aided VP detetion method which combines 1-line RANSAC method with polar grid to generate more stable VP.

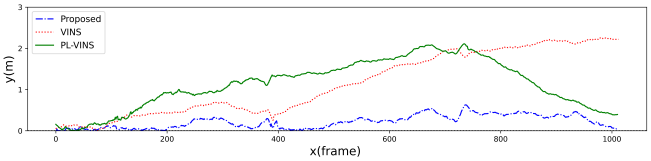

Fig. 9. Comparison accuracy of the proposed method (blue), VINS (red) and PL-VINS (green) over the duration of the indoor dataset.

For further reducing resource consumption, we employed a voxel-map based depth association scheme which uses voxel-hashing data structure to store local map and ray casting method to retrieve depth information for visual features. The proposed state estimation method was evaluated in indoor as well as outdoor environments. The experiments show that our approach is more accurate than the state-of-the-art visual-inertial odometry and LiDAR-visual odometry. In the future, we will try to adopt visual line features to further improve the accuracy of our system.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (No. 62073066, U20A20197), Science and Technology on Near-Surface Detection Laboratory (No. 6142414200208), the Fundamental Research Funds for the Central Universities(No.N182608003), and Aeronautical Science Foundation of China (No. 201941050001), and Major Special Science and Technology Project of Liaoning Province (No.2019JH1/10100026).

REFERENCES

[1] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5303–5310.

[2] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[3] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[4] D. Adams, *The Hitchhiker's Guide to the Galaxy*. San Val, 1995. [Online]. Available: http://books.google.com/books?id=W-xMPgAACAAJ

[5] J. Zhang, M. Kaess, and S. Singh, "A real-time method for depth enhanced visual odometry," *Autonomous Robots*, vol. 41, no. 1, pp. 31–43, 2017.

[6] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7872–7879.

[7] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, "Lidar-monocular visual odometry using point and line features," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1091–1097.

[8] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019.

[9] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6319–6326.

[10] D. G. Kottas and S. I. Roumeliotis, "Efficient and consistent vision-aided inertial navigation using line observations," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1540–1547.

[11] X. Kong, W. Wu, L. Zhang, and Y. Wang, "Tightly-coupled stereo visual-inertial navigation using point and line features," *Sensors*, vol. 15, no. 6, pp. 12 816–12 833, 2015.

[12] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[13] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, and H. Zhang, "Pl-vins: Real-time monocular visual-inertial slam with point and line," *arXiv preprint arXiv:2009.07462*, 2020.

[14] X. Li, Y. He, J. Lin, and X. Liu, "Leveraging planar regularities for point line visual-inertial odometry," *arXiv preprint arXiv:2004.11969*, 2020.

[15] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 5219–5225.

[16] T. Dang, C. Hoffmann, and C. Stiller, "Continuous stereo self-calibration by camera parameter tracking," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1536–1550, 2009.

[17] X. Yin, X. Wang, X. Du, and Q. Chen, "Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5870–5878.

[18] Z. Shan, R. Li, and S. Schwertfeger, "Rgbd-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, p. 2251, 2019.

[19] N. Ragot, R. Khemmar, A. Pokala, R. Rossi, and J.-Y. Ertaud, "Benchmark of visual slam algorithms: Orb-slam2 vs rtab-map," in *2019 Eighth International Conference on Emerging Security Technologies (EST)*. IEEE, 2019, pp. 1–6.

[20] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf, "Computational geometry," in *Computational geometry*. Springer, 1997, pp. 1–17.

[21] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual slam using sparse depth for camera-lidar system," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[22] D. G. Kottas and S. I. Roumeliotis, "Exploiting urban scenes for vision-aided inertial navigation." in *Robotics: Science and Systems*, 2013.

[23] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "Structslam: Visual slam with building structure lines," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.

[24] J.-C. Bazin and M. Pollefeys, "3-line ransac for orthogonal vanishing point detection," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4282–4287.

[25] X. Lu, J. Yaoy, H. Li, Y. Liu, and X. Zhang, "2-line exhaustive searching for real-time vanishing point estimation in manhattan world," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 345–353.

[26] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[27] S. Jianbo and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[28] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2008.

[29] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[30] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "Structvio: visual-inertial odometry with structural regularity of man-made environments," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 999–1013, 2019.

[31] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam," *arXiv preprint arXiv:2007.11898*, 2020.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.