2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

ASML-VDIO: Visual-Depth-Inertial Odometry using Selected Accurate and Stable Multi-Modal Landmarks in Structural Environments

Xingjian Luo, Chenglin Pang, Xuankang Wu and Zheng Fang*

***Abstract*— In complex indoor structural scenes such as shopping centers and malls, camera pose estimation using pure point features is easy to fail due to the difficulty in extracting sufficient and stable point features from weak textures or dynamic environments. Recent works have attempted to address these challenges by introducing line features. However, the addition of line features increases the number of parameters and landmarks for BA (Bundle Adjustment), leading to efficiency reduction. This is a common issue in multi-modal SLAM (Simultaneous Localization And Mapping). To address this issue, this paper proposes a novel visual-depth-inertial odometry (ASML-VDIO) framework by combining RGB-D and IMU sensors. To improve the efficiency of BA, the proposed landmark classification method classifies 3D landmarks into accurate landmarks and other landmarks based on spatial consistency verification and depth range limitation. Then, accurate landmarks are fixed, and only other landmarks are optimized in the optimization of BA. Furthermore, to remove line features extracted from dynamic objects (pedestrian, shopping-car, etc), we propose a dynamic line removal method that combines geometric constraints and motion constraints of line features. Finally, the method is evaluated on public and author-collected datasets, showing competitive accuracy and robustness in complex indoor structural scenes while 71% speedup on optimization thread with same constraints.**

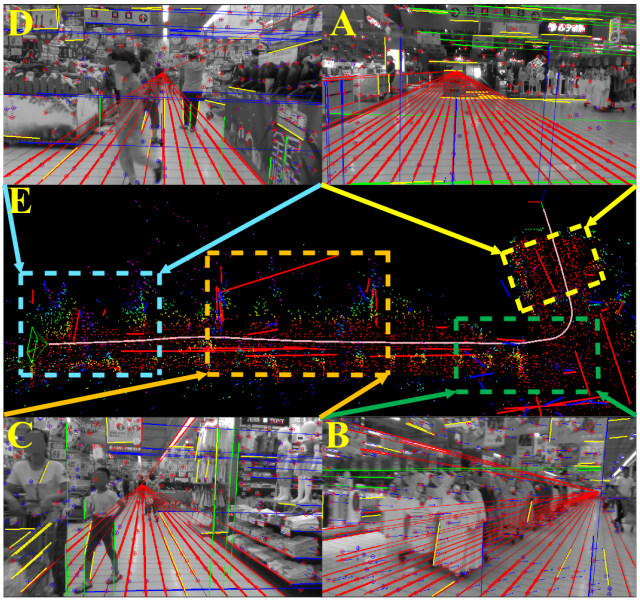

Fig. 1. The visualization of the Market1-1 sequence from the OpenLORIS dataset. The feature extraction is stable in figure A, the image is blurred due to the fast motion of robot in figure B, the dynamic object occlusion exists in figure C and D, and the red point and line markers in figure E are the accurate landmarks selected by the algorithm.

I. Introduction

Simultaneous localization and mapping (SLAM) technology has been widely used in ground autonomous mobile robots. In recent years, the visual-inertial SLAM navigation system (VINS) [1] [2]has become increasingly popular due to its low cost, small size and easy hardware setup. VINS takes advantage of the complementarity between different sensors to provide robust and accurate 6-DoF pose estimation and therefore achieves better results compared to pure visual SLAM methods. Nowadays, VINS is widely installed as the main sensing unit on the ground autonomous mobile robot for pose estimation and environment perception.

Ground autonomous mobile robots like cleaning robots and food delivery robots typically operate in complex man-made structural environments such as shopping centers and malls. These scenes contain large areas of white walls with missing textures or floors with repetitive textures. Point-based VINS methods have difficulty in extracting sufficient reliable features for accurate pose estimation. Therefore, [3] introduces line features to improve the accuracy and robustness of the system, especially in texture-less regions of artificial environments. In addition, ground autonomous mobile robots need to operate for a long time, therefore it is inevitable to accumulate trajectory drift. In [4] [5], the authors show that rotation estimation errors are the main cause of long-term drift. To reduce the trajectory drift, [6] try to utilize extra constraints and regularities such as Manhattan world assumption [7] and Atlanta world assumption [8]. However, these methods only use structural lines with dominant orientations, so SLAM systems integrating these methods are only used in ideal indoor environments. [9] focuses on the use of structural lines without any constraints. It uses vanishing point measurements [10] to constrain the direction of straight lines, which can be better adapted to complex indoor structural scenes. But it is difficult to recover the real scale in IMU degraded scenes. Simultaneously, the introduction of line features increases a large number of line landmarks and optimization variables for optimization of bundle adjustment (BA), which substantially impacts the efficiency of the system. This issue is also a prevalent

This work was supported in part by the National Natural Science Foundation of China under Grants 62073066 and U20A20197, in part by the Fundamental Research Funds for the Central Universities under Grant N2226001, and in part by 111 Project under Grant B16009. (Corresponding author: Zheng Fang.) The authors are all with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China. Zheng Fang is also with the National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang 110819, China and also with the Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, Northeastern University, Shenyang 110819, China. (e-mail:fangzheng@mail.neu.edu.cn).

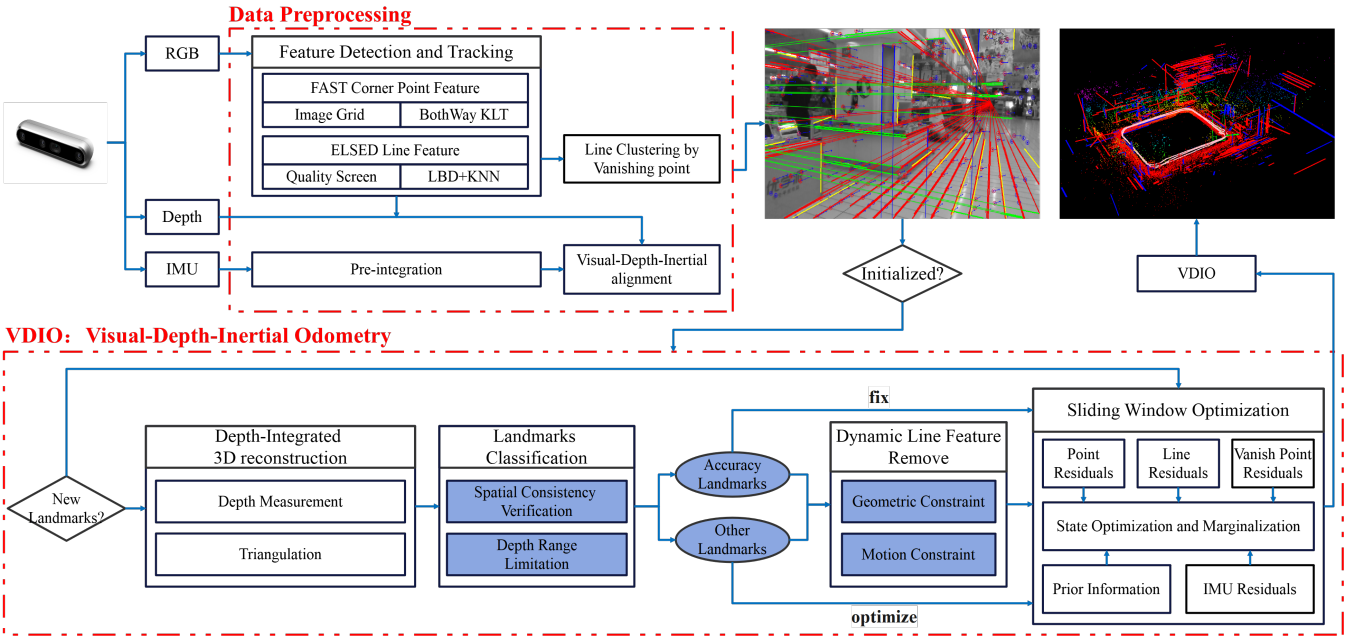

Fig. 2. The framework of ASML-VDIO. The contributing modules are highlighted. The input to the system comes from a consumer-grade RGB-D camera with an integrated IMU that provides RGB, Depth and IMU information. Data preprocessing thread extracts structural information from RGB images and pre-integrates IMU measurements, while depth images are used to obtain depth information of features. In VDIO thread, accurate and stable multi-modal landmarks are obtained by the proposed landmark classification method (see III-E) and dynamic line feature removal method (see III-F), and the pre-integration results of IMU are combined to construct factor graphs to optimize the pose estimation.

challenge in multi-feature SLAM methods.

With the development of sensor technology, some researchers tried to integrate depth information from RGB-D cameras to enhance the initialization and optimization process of VIO. Notably, [11] incorporates depth information on the base of [1]. Moreover, [12] uses EKF to fuse RGB images and depth point clouds. However, both of these methods rely on the assumption of a static environment. Unfortunately, in real-world scenes like shopping malls, the presence of moving individuals and shopping carts can have a negative impact on pose estimation. Although some methods like RANSAC [13] can mitigate the impact of dynamic features to some extent, it overwhelmed when there are a large number of dynamic objects in the scene. Therefore [14] integrates depth feature points with static weights for camera pose estimation, which reduces tracking errors. [15] uses a lightweight target detection model for dynamic target segmentation of depth images to reduce the impact of dynamic objects on the system. However, the above methods only use point features in the environment and ignore line features. Line features in structured environments are less susceptible to the influence of dynamic objects, making them more robust. Furthermore, they can also help compensate for the decrease in system accuracy caused by the removal of a significant number of features.

To address all these issues, this paper proposes a novel visual-depth-inertial odometry method based on RGB, depth and IMU information for indoor complex structural environments, called ASML-VDIO. As shown in Fig 1, it extracts multi-modal features from the environment, cross-verifies and classifies the 3D landmarks into accurate landmarks and other landmarks. Notably, accurate landmarks are fixed during the optimization of BA. Additionally, dynamic line features are removed by checking both geometric and motion constraints. In summary, the main contributions are as follows:

- A novel visual-depth-inertial odometry algorithm is proposed, which improves the accuracy and robustness of pose estimation in indoor complex structured scenes by using selected accurate and stable multi-modal landmarks.
- A multi-modal landmarks classification method is proposed. By classifying the reconstructed 3D landmarks into accurate landmarks and other landmarks, and subsequently fixing the accurate landmarks during the optimization of BA, the speed of back-end optimization is significantly improved.
- A novel method for the removal of dynamic line features is proposed. The main idea lies in using both geometric and motion constraints to remove line features extracted from moving objects such as pedestrians and shopping carts.

The rest of the paper is as follows: Section II describes the overall framework of our algorithm. Section III details the proposed method. Section IV demonstrates and compares the performance of the proposed method on public and author-collected dataset. Finally, Section V summarizes our contributions.

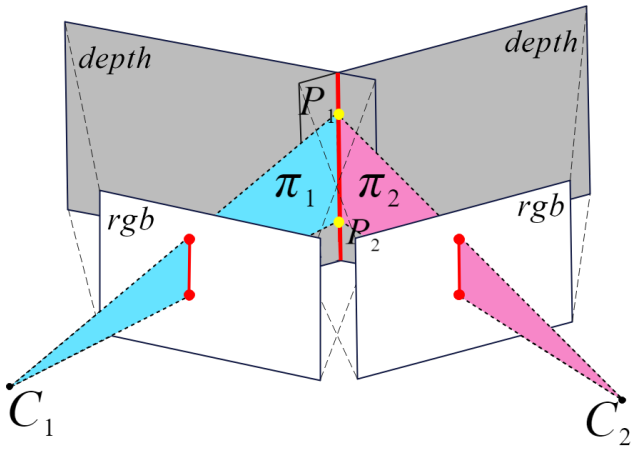

Fig. 3. Illustration of 3D line reconstruction. A 3D line L in space can be recovered by two planes π_1 and π_2 determined by two observation frames C_1 and C_2. It can also be recovered from two 3D endpoints P_1 and P_2 determined by either C_1 or C_2 observation frames.

II. System Overview

Our method is based on VINS-RGBD [11] and UV-SLAM [9]. The proposed framework is shown in Fig 2. We use a consumer-grade RGB-D camera integrated with an IMU as the sensor. The sensor data are directly input into the Data Preprocessing module to extract both point and line features. After the initialization, we integrate depth information to recover 3D landmarks. These landmarks are classified into Accuracy Landmarks and Other Landmarks by the Landmarks Classification module. Subsequently, the Dynamic Feature Removal module removes the dynamic line features. Finally, the accurate and stable multi-modal landmarks are input into the Sliding Window Optimization to get camera poses consecutively. It is notable that during the optimization of BA, we fix the spatial positions of Accuracy Landmarks and only optimize the spatial positions of Other Landmarks.

III. Methodology

A. Feature Detection and Tracking

Our system employ multi-modal features. For point features, we adopt the grid-based feature detection method proposed in Dynamic-VINS [15]. However, unlike [15], we do not need to skip the grid of weak textures in the detection frame because we introduce line features, which enhances robustness in weakly textured scenes. Additionally, we utilize the KLT [16] optical flow method to remove false matches. As for line features, we employ ELSED [17] for detection. Simultaneously, we calculate the LBD [18] descriptor and apply the KNN method for matching. Furthermore, we utilize the 2-line [10] method to detect vanishing points, which allows us to classify line features into structural lines and non-structural lines .

B. Depth-Integrated 3D LandMark Reconstruction

In SLAM, the spatial position of 3D points can be intuitively represented by (x, y, z), while the space position of 3D lines is represented using Plücker coordinates proposed in [19]. Plücker coordinates provide an intuitive and unique

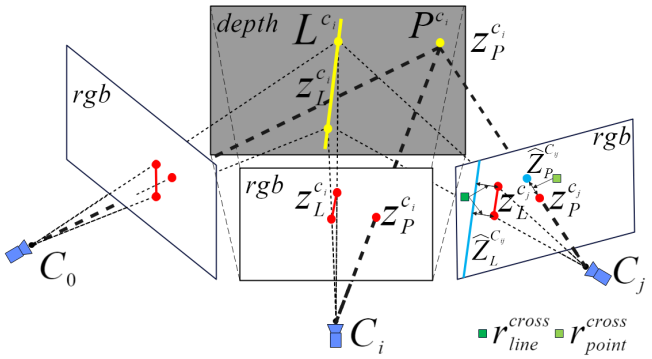

Fig. 4. Illustration of cross reprojection. The dots and solid lines in the image respectively represent point and line landmarks. The yellow ones represent the 3D landmarks recovered using depth measurement in the current observation frame, the red ones represent the observations of the landmarks, and the blue ones represent the re-projection estimations of the landmarks. 3D landmarks obtained from depth measurement are re-projected into the next observation frame. Afterward, the distance between the observed point and the re-projected point is defined as the cross reprojection r_P^{corss} of the point landmark, and the distance between both endpoints of the observed line and the re-projected line is defined as the cross reprojection r_L^{corss} of the line landmark.

representation of a line in 3D space, which is represented as follows:

$$\mathbf{L}(\mathbf{n}, \mathbf{d})^\top \in \mathbb{R}^6, \tag{1}$$

where $\mathbf{n}$ and $\mathbf{d}$ denote the normal and directional vectors of the line, respectively. Plücker coordinates are widely used in the triangulation and reprojection processes of straight lines.

Due to the introduction of depth measurement, the spatial position of landmarks can be recovered using various methods. When depth measurements are zero value, we use the traditional triangulation method. When depth measurements are not zero value, for point features, the depth measurement can be directly assigned to z. For line features, the Plücker coordinates of the line are typically obtained using the dual Plücker matrix, defined as:

$$\mathbf{L}^* = \begin{bmatrix} [\mathbf{d}]_\times & \mathbf{n} \\ -\mathbf{n}^T & \mathbf{0} \end{bmatrix} = \mathbf{X}_1\mathbf{X}_2^\mathbf{T} - \mathbf{X}_2\mathbf{X}_1^\mathbf{T} \in \mathbb{R}^{4\times4} \tag{2}$$

As shown in Fig 3, due to the dyadic nature of points and planes, $\mathbf{X}_1$ and $\mathbf{X}_2$ can represent either two planes or two 3D points' normalized coordinate. It is important to note that the dual Plücker matrix constructed with two endpoints and two planes has opposite normal and directional vectors when obtaining Plücker coordinates.

C. Spatial Consistency Verification based on Cross Reprojection

Using depth measurements can recover the spatial positions of landmarks in current observation frame. However, depth images are not always correct, especially at depth discontinuities such as object boundaries. Moreover, many RGB-D sensors have holes and some patterns (zero measurements) due to their inherent characteristics, even though the depth images have been processed by a hole-filling filter. Consequently, the 3D landmarks recovered through depth measurements need to be further verified.

As shown in Fig 4, consider the k^{th} landmark is first recovered through depth measurements in the i^{th} observation frame and last recovered in the j^{th} observation frame. The cross reprojection of the k^{th} 3D landmark during its all observation frames is constructed as:

$$\mathbf{r}_{P_k}^{corss} = \mathbf{p} - \tilde{\mathbf{p}}, k \in (i, j) \tag{3}$$

$$\mathbf{r}_{L_k}^{cross} = \begin{bmatrix} d(\mathbf{p}_s, \mathbf{l}^c) \\ d(\mathbf{p}_e, \mathbf{l}^c) \end{bmatrix} , k \in (i, j) \tag{4}$$

where,

$$\begin{aligned} d(\mathbf{p}, \mathbf{l}^c) &= \frac{\mathbf{p}^\top \mathbf{l}^c}{l_d},\ l_d = \sqrt{l_1^2 + l_2^2} \\ \mathbf{p}_s &= (u_s, v_s, 1),\ \mathbf{p}_e = (u_e, v_e, 1) \end{aligned} \tag{5}$$

and $\mathbf{r}_{P_k}^{corss}$ denotes the point landmark cross reprojection and $\mathbf{r}_{L_k}^{corss}$ denotes the line landmark cross reprojection. d denotes the distance between both endpoints of the observed line and the reprojected line. $\mathbf{p}_s$ and $\mathbf{p}_e$ are the endpoints of the observed line in the image. When the $r_{P_k}^{corss}$ or $r_{L_k}^{corss}$ falls below the preset threshold, the k^{th} landmark is considered pass the spatial consistency verification.

D. Depth Range Limitation

By III-C, we rejected the landmarks recovered using incorrect depth measurements. However, RGB-D sensors have a limited range and there are blind and far-field areas. Consequently, any measurements collected from beyond this restricted range are inaccurate. In our implementation, the sensor's range is predefined to limit the reconstruction range of the landmarks. For point landmarks, we determine whether the depth measurement falls within this predefined range. Similarly, for line landmarks, we check whether the depth measurements of the start and end points fall within the specified range. Any point or line landmark that falls outside this predefined range is subsequently classified in III-E. Note that the sensor's range does not need to be equal to the device's real range. It can be manually set by the user. In this way, the user can compensate the increased error of the RGB-D sensor for far-area and blind-area measurements [11].

E. LandMark Classification

We combine III-C and III-D to classify the reconstructed 3D landmarks, as shown in algorithm 1. With the algorithm 1 we obtain the set of different classes of spatial positions recovered by a single landmark in all its observed frames using depth measurements. Where the 3D points are denoted as depth z and the 3D lines are denoted as Plücker coordinates. It is important to note that the landmarks finally stored in the landmark set are the ones that have been transformed to the reference coordinate system using the transformation matrix.

Furthermore, during the optimization of BA, we fix the landmarks with an estimate flag of 1, which indicates its spatial position is accurate. For the landmarks with estimate flags of 2 and 0, representing rough, dubious and triangulate classes, their space positions need to be optimized to improve the accuracy.

As shown in Eqs. (6), (7). For the accurate class and the rough class that have passed the spatial consistency verification, the 3D landmark of current point feature is represented as the mean value of the set of landmarks, while the 3D landmark of line feature is represented as the landmark with the longest tracking time in the observed frame, which means the landmark at the tail of the set. For these landmarks, the estimate flag is set to 1 and 2, respectively. However, for the dubious class, its landmarks have not passed the spatial consistency verification, indicating the presence of noise in the space position recovered with depth measurement. These landmarks and those of the triangulate class landmarks with a zero depth measurement were recovered using the traditional triangulation method, and their estimate flag is set to 0.

$$\boldsymbol{Pos}_p = \begin{cases} Pos_p^{mean}, p^{flag} = 1, p \in \mathcal{C}_{acc} \\ Pos_p^{mean}, p^{flag} = 2, p \in \mathcal{C}_{rou} \\ Triangulate, p^{flag} = 0, p \in \mathcal{C}_{dub} \\ Triangulate, p^{flag} = 0, p \in \mathcal{C}_{tri} \end{cases} \tag{6}$$

$$\boldsymbol{Pos}_l = \begin{cases} Pos_l^{latest}, l^{flag} = 1, l \in \mathcal{C}_{acc} \\ Pos_l^{latest}, l^{flag} = 2, l \in \mathcal{C}_{rou} \\ Triangulate, l^{flag} = 0, l \in \mathcal{C}_{dub} \\ Triangulate, l^{flag} = 0, l \in \mathcal{C}_{tri} \end{cases} \tag{7}$$

Algorithm 1 LandMark Classification

Data: set of all features within observation frames $\mathcal{F}$; accurate landmark set $\mathcal{C}_{acc}$; rough landmark set $\mathcal{C}_{rou}$; dubious landmark set $\mathcal{C}_{dub}$
Result: verified and classified consistent 3D landmark space position in reference frames

```
for feature f in F do
    if f has depth measurement then
        transform f to reference frame;
        get 3D landmark Pos_PL^ref of transformed feature;
        if r_PL^cross < threshold then
            if Pos beyond device range then
                C_rou ⟵ Pos_PL^ref;
            else
                C_acc ⟵ Pos_PL^ref;
            end
        else
            C_dub ⟵ Pos_PL^ref;
        end
    else
        triangulate;
    end
end
```

F. Dynamic Line Feature Remove

Indoor scenes such as shopping centers and malls contain numerous dynamic objects, as shown in Fig 5. Point features

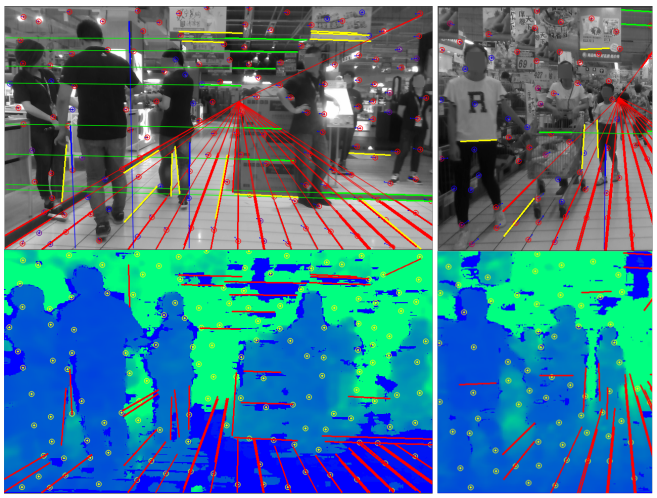

Fig. 5. Illustration of Dynamic Line Feature Detect. The image above shows the visualization of the features of the RGB image. The lines with structural properties can be aggregated to the same vanishing point and are represented as red, green and blue. The yellow lines do not have structural properties. The following image shows the depth image based on the depth magnitude, with the depth values colored from blue to cyan. The line features extracted from moving people have no structural pattern and have similar depth characteristics. Structural lines can be extracted from non-human dynamic objects such as shopping carts.

are easily extracted from moving people and shopping carts. However, most line features extracted from moving people do not have structural properties, they cannot be extended to any of the vanishing points detected in the image. Futhermore, compared to the structural lines in the environment, the lines on human bodies have the property of being similar in depth. The depth measurements of the two endpoints are often very close to each other, since the contour of a person in the depth image forms an approximately planer surface. Based on this geometric constraint, line features on dynamic human bodies can be easily distinguished from other line features.

Furthermore, for unknown non-human dynamic objects such as lines with structural properties on shopping carts, we use the motion constraints to recognize dynamic line features. Given l_i and $l_i^{'}$ matching line feature in two adjacent frames, their 3D positions are recovered from the landmark reconstruction. The 3D line L_i of the i^{th} frame is transformed by the estimated pose T_{sw} of the sliding window into the line $L_i^{T_{sw}}$ with endpoints P_s,$P_e \in \mathbb{R}^3$, the endpoints $P_s^{'}$, $P_e^{'} \in \mathbb{R}^3$ of the 3D line $L_{i'}$ of the $i^{'}$th frame. The spatial distance d_L between the two 3D line landmarks is defined as follows:

$$d_L = d(L_i^{T_{sw}}, L_{i'}) = \frac{\overrightarrow{\mathbf{P}_s\mathbf{P}_s^{'}} \cdot \mathcal{N}}{\|\mathcal{N}\|} \tag{9}$$

$$\mathcal{N} = \overrightarrow{\mathbf{P}_s\mathbf{P}_e} \times \overrightarrow{\mathbf{P}_s^{'}\mathbf{P}_e^{'}} \tag{10}$$

where $\mathcal{N}$ denotes the common vertical line of the two 3D line $L_i^{T_{sw}}$ and $L_{i'}$. By this, the i-th line feature is considered as a dynamic feature when d_L exceeds a preset threshold.

The overall process of determining the dynamic line features is shown in Algorithm 2.We combine geometric constraints and motion constraints of line features to identify moving people or unknown non-human dynamic objects and remove dynamic line features after detecting them.

Algorithm 2 Dynamic Line Judge

Data: Matching Line l_i and $l_{i'}$ between two adjacent frames; Sliding window estimated pose T_{sw};geometric thresholds $\mathcal{T}_g$; motion thresholds $\mathcal{T}_m$.
Result: To determine whether the extracted line features l_i are dynamic lines

```
for Line features per matched pair do
    if l_i corresponding to the vanishing point measurement has the value then
        l_i^{vp} = 1;
    else
        l_i^{vp} = 0;
    end
    calculate the depth difference d_P = l_i^{sp} - l_i^{ep};
    recover 3D line L_i and L_{i'};
    Use T_{sw} to compute L_i^{T_{sw}};
    Use 3D line L_i^{T_{sw}} and L_{i'} to compute the spatial distance d_L;
    if (l_i^{vp} == 0 ∧ d_P ∈ [-T_g, T_g]) ∨ (l_i^{vp} == 1 ∧ d_L ≤ T_m) then
        The line l_i is a dynamic line;
    end
end
```

IV. Experiments

In this section, we evaluate the proposed method using public and author-collected datasets. The public SLAM dataset OpenLORIS-Scene is a real-world indoor dataset using ground mobile robots, providing ground truth information to evaluate SLAM systems in complex indoor environments. It includes 22 sequences in 5 scenarios: office, corridor, home, cafe and market. The challenges come from changed viewpoints and illumination, moving or deforming objects in the scene, degraded sensors and so on. On the other hand, the author-collected dataset Dog-Indoor is an indoor dataset captured using RGB-D cameras (Intel Realsense D435i and D455) mounted on the quadruped robot carrier. It includes 4 sequences and the challenges we foucs come from repetitive textured carpets, untextured white walls, and complex dynamic interference. Quantitative experiments IV-A on the OpenLORIS-Scene dataset [20] and qualitative experiments IV-B on the Dog-Indoor dataset are performed to evaluate the proposed system's accuracy, robustness, and efficiency.

Since our system is built on [11] and [9], they are used as baselines to demonstrate our improvements. Additionally, we compare other visual inertial odometry methods based on point features and line features such as [1], [3] and [15]. As all these methods are based on [1], we adjusted the parameters of the comparison algorithm to pass as many

TABLE I

THE RMSE OF ATE AND RPE ON OPENLORIS-SCENE DATASETS

Sequence	Vins-Mono [1] (RGB+IMU+P)		PL-VINS [3] (RGB+IMU+PL)		UV-SLAM [9] (RGB+IMU+PLV)		VINS-RGBD [11] (RGBD+IMU+P)		Dynamic-VINS [15] (RGBD+IMU+P)		OURS (RGBD+IMU+PLV)	
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE
cafe1-1	0.329	1.906	0.546	1.924	0.325	**1.700**	0.351	2.222	0.315	1.793	**0.314**	1.896
cafe1-2	0.396	1.818	0.421	1.826	4.004	**1.575**	0.313	1.832	**0.298**	1.811	0.315	1.605
corridor1-1	4.213	1.605	——	——	2.672	1.670	2.710	1.725	2.449	1.725	**2.247**	**1.621**
corridor1-2	——	——	1.477	**1.713**	——	——	1.437	1.737	1.479	1.750	**1.423**	1.772
corridor1-3	——	——	2.405	2.222	——	——	2.150	**2.140**	2.051	2.225	**1.926**	2.326
corridor1-4	2.175	1.771	6.917	2.378	2.163	1.659	1.341	1.657	**1.174**	**1.643**	1.526	1.680
corridor1-5	1.953	1.796	6.749	3.094	2.487	1.641	——	——	——	——	**1.329**	**1.613**
home1-1	0.813	2.105	1.591	2.255	0.872	2.160	0.414	2.030	**0.401**	2.025	0.444	**2.021**
home1-2	0.471	2.249	0.353	2.324	——	——	0.359	**2.237**	0.367	2.542	**0.337**	2.293
home1-3	0.689	2.304	0.850	2.300	1.506	1.934	0.416	2.086	0.392	2.193	**0.376**	**1.912**
home1-4	0.806	2.146	0.657	2.129	1.010	**1.789**	0.329	2.143	**0.292**	2.139	0.365	1.825
home1-5	0.252	2.721	0.253	2.707	0.702	2.452	0.275	2.725	0.278	2.491	**0.248**	**2.323**
market1-1	2.366	1.648	2.224	1.666	2.588	1.631	0.971	1.625	0.849	1.595	**0.687**	**1.582**
market1-2	2.913	1.641	3.058	1.656	2.369	1.627	0.986	1.586	0.945	1.573	**0.877**	**1.569**
market1-3	2.580	1.744	3.086	1.790	1.672	1.672	1.190	1.624	1.174	**1.625**	**1.025**	1.628
office1-1	0.241	2.127	0.245	2.137	0.305	**1.407**	**0.097**	1.815	0.100	1.837	**0.097**	1.549
office1-2	0.263	1.861	0.339	2.032	0.251	1.917	0.114	1.840	0.116	**1.831**	**0.113**	1.860
office1-3	0.121	0,799	0.125	0.799	4.542	1.241	0.150	0.731	0.150	0.700	**0.120**	**0.306**
office1-4	0.239	2.000	0.283	2.043	0.447	**1.184**	0.175	1.732	0.179	1.733	**0.173**	1.740
office1-5	**0.209**	2.029	0.217	2.057	——	——	0.226	1.978	0.238	1.967	0.223	**1.824**
office1-6	**0.067**	1.775	0.072	1.860	0.147	**1.266**	0.081	1.888	0.083	1.882	0.078	1.359
office1-7	0.132	1.590	0.105	1.657	4.974	1.056	0.609	**1.407**	0.098	1.555	**0.096**	1.560

test sequences as possible with the same set of parameter settings. We use the root mean square error RMSE of the absolute trajectory error ATE and relative attitude error RPE to evaluate the accuracy of the algorithm on the dataset. Any sequences that do not run completely or fail are marked as "–". The experimental platform for our experiments is an Intel Core i7-11800H processor with 16GB of RAM.

A. OpenLORIS-Scene Dataset

The results are shown in Table I. The proposed AMSL-VDIO shows the best robustness among the tested algorithms. In most scenes, the fused RGB-D VIO method outperforms the monocular VIO method, mainly because the fused depth can provide better scale estimates in scenes with degraded IMU. The *cafe* and *home* scenes contain some people with slight movements, [15] effectively removes the interference caused by dynamic people and resulting in good results. The *office* scene is mainly a static environment, most algorithms can successfully track and achieve good accuracy. The *corridor* scene includes long corridors with poor lighting and weak textures, the method of fusing line features can make up for the lack of point features and maintain the tracking for a long time, showing good robustness. The *market* scene covers the largest area, while there are many moving pedestrians, shopping carts and other unknown non-human motion objects. Our method makes full use of the verified structural line features in the environment and removes the interference from dynamic lines, resulting in better accuracy compared to [11] and [15] which rely solely on point features. Due to the vanishing point constrains the direction of the line features, [9] shows better RPE. However, the robustness of line feature extraction is problematic, as many line segments in the environment that are not straight lines are also extracted as straight lines, leading to inaccurate triangulated spatial positions. This negatively impacts the system's pose estimation and affects the ATE to some extent. In contrast, our method improves the accuracy and robustness of the line landmarks in 3D space through spatial consistency verification and depth range limitation, which makes our method perform well on ATE.

B. Dog-Indoor Dataset

For carriers like quadruped robots, their inherent motion characteristics introduce significant noise to the IMU sensors. Therefore, we solely employ RGB and Depth information in our experiments on the Dog-Indoor dataset. The algorithms we compare exclude IMU usage and rely solely on RGB and Depth data. In order to verify the robustness of our proposed method in complex dynamic environments, we conducted Multiple-Dynamic Experiments. The experimental results show that our method can effectively withstand the disturbances caused by intricate dynamic environments and can approximate the real trajectory even without IMU. To evaluate the accuracy of our proposed method, we conducted Trajectory Drift Experiments. The experimental results show that our method exhibits reduced drift and greater consistency within structured environments.

1) Multiple Dynamic Experiments: The results are shown in Fig 6. The compared methods includes [11], [15] and VINS-RGBD-FAST. Among them, VINS-RGBD-FAST is the version of [15] with target detection removed. The point features are extracted using the grid-based feature detection method proposed in [15]. [11] traverses the whole image to detect point features, which lacks sufficient robustness and crashes directly at position C. VINS-RGBD-FAST detects point features only in grids with insufficient feature matching, showcasing improved robustness. However, the introduction of some dynamic point features leads to a large

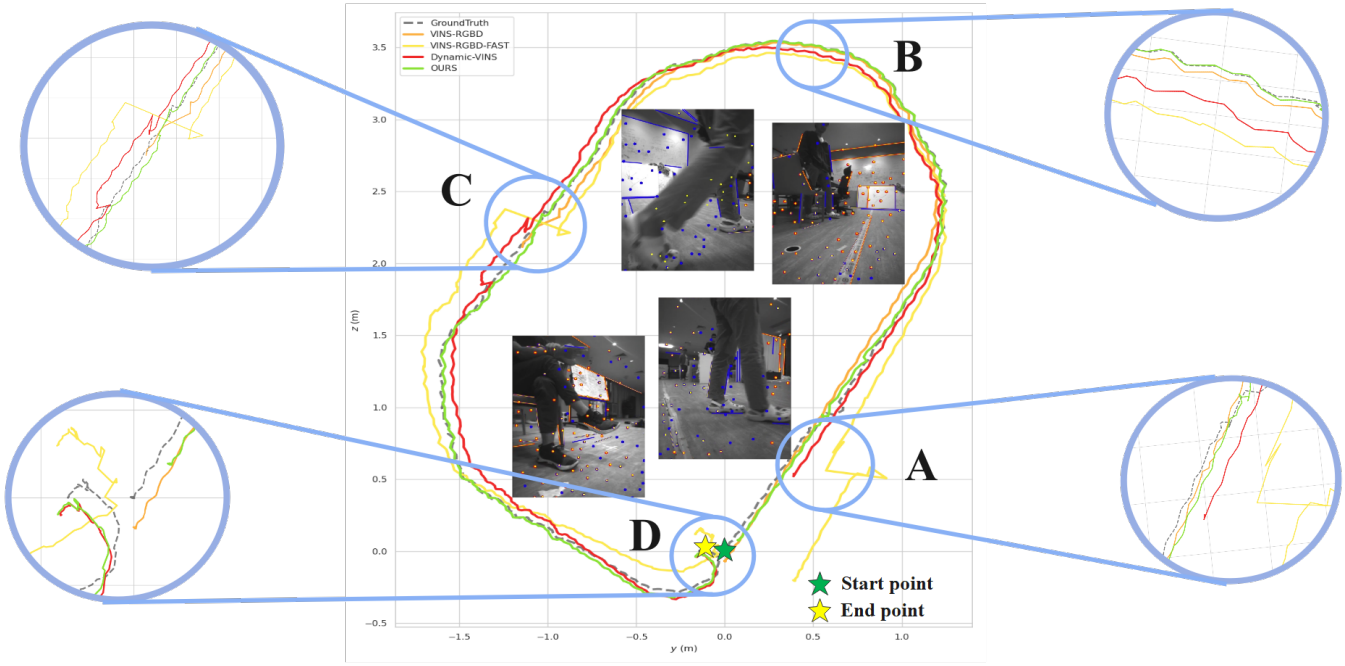

Fig. 6. Qualitative results of evaluated algorithms on Multiple-Dynamic Experiments. In position A, the moving person appears slowly and starts to move slowly, occupying part of the field of view of the quadruped robot. In position C, the moving person appears suddenly and starts to move quickly, occupying half of the field of view of the quadruped robot. In position B, standing people have a small motion of the lower limbs, and in position D, sitting people have a small motion of the upper limbs.

offset in the estimated poses at positions A and C. [15] employs target detection and depth information to remove the dynamic points on moving objects. It can handle slow-moving individuals appearing at position A, but it is ineffective against sudden and rapid movements of individuals at position C. Our method enhances system stability by complementing stable point features with accurate structural lines selected in the structured environment. On one hand, we remove dynamic line features through the integration of geometric and motion constraints applied to the line features. On the other hand, by exploiting structure line features that mostly stationary, our method is closer to the true value in the estimation of the poses.

2) Trajectory Drift Experiments: The results are shown in Fig 7. A-D are OURS, [15] with Loop Closure, [15], and [11] in that order. Even without IMU, the three RGB-D VIO methods still achieve improved scale estimates through the utilization of depth information, highlighting the advantages of fused RGB-D. While a certain number of point features are extracted to maintain system stability, the quality of these point features is notably impacted by factors such as repeated textures and areas with weak or no texture in the environment. Furthermore, the point feature includes some dynamic points that are extracted from moving objects. [15] can remove dynamic points, leading to the drift of its trajectory is smaller than [11]. In addition, it is difficult to constrain the orientation of point features, which leads to a bias in the estimation of rotation in the pure point method. Although the cumulative drift can be reduced to some extent by using loop closure, the loop is difficult to trigger in time and the loop constraint cannot correctly apportion the drift. In contrast, line features possess inherent directionality, and the introduction of vanishing points remove lines that deviate from structural patterns. Therefore, our method is more accurate in rotational estimation and thus maintains minimal drift in repeated motions, and achieving better consistency compared to other methods.

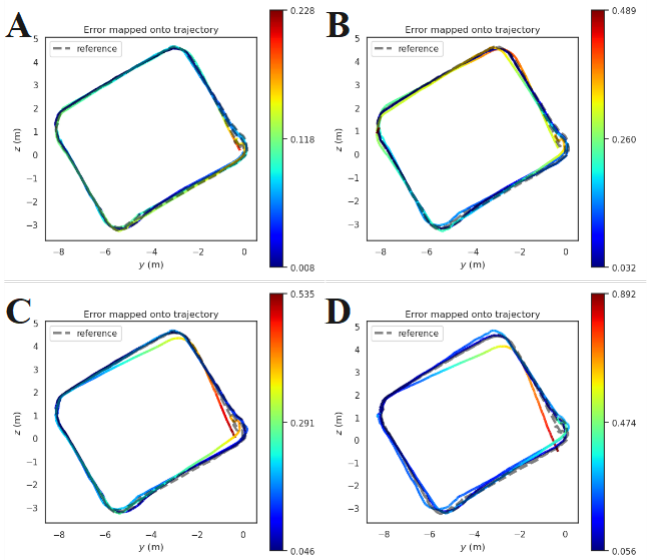

Fig. 7. Qualitative results of evaluated algorithms on Trajectory Drift Experiments. The start and end points of the trajectory are the same. The error of the trajectory is based on the color from red to blue.

TABLE II

AVERAGE COMPUTATION TIME[MS] OF EACH MOUDLE AND THREAD ON ON OPENLORIS DATASETS MARKET1-1

Platforms	Methods	Point Tracking	Point Detection	Line Tracking	Line Detection	Tracking Thread	State Optimization	Optimization Thread	Object Detection
Intel Core Processor	VINS-Mono [1]	2.0412	10.8957	*	*	15.6788	46.0194	48.8102	*
	VINS-RGBD [11]	1.9609	10.1792	*	*	15.5010	46.4869	48.5167	*
	Dynamic-VINS [15]	2.6260	0.7785	*	*	5.3881	46.4026	47.5624	20.0621
	PL-VINS [3]	2.4642	9.2247	0.3289	22.5407	38.4382	64.0022	66.6098	*
	UV-SLAM [9]	4.5618	14.8853	1.0925	31.5451	55.8696	120.2557	142.2338	*
	ASML-VDIO	4.7079	0.4890	0.5121	19.7089	30.3130	**38.5855**	**41.0390**	*

C. Runtime Analysis

This part compares [1], [11], [15], [3], [9] and AMSL-VDIO for runtime analysis. These methods are expected to track and detect 150 feature points and 60 feature lines, and the frames in [15] and AMSL-VDIO are divided into 7x8 grids. The depth measurement range is set to 0.3-6m. The average computation times of each module and thread are calculated on OpenLORIS market scenes 1-1 sequence and the results are shown in Table II. It should be noted that the average computation time is only to be updated when the module is used. Thanks to fixing the accurate landmarks during BA optimization, our algorithm shows the best optimization speedup compared to other algorithms, even with the introduction of line features and vanishing points. Compared to [9] with the same optimization constraint, our speedup in the State Optimization module is 68% and in the Optimization Thread is 71%.

V. CONCLUSIONS

This paper presents a novel visual-depth-inertial odometry method based on three types of information provided by a consumer-grade RGB-D camera with an integrated IMU, which includes RGB, depth and IMU, aiming to make use of accurate and stable multi-modal landmarks selected in a structured environment. The proposed multi-modal landmarks classification method classifies the 3D landmarks into accurate landmarks and other landmarks based on spatial consistency verification and depth range limitation. Additionally, through the proposed dynamic line feature removal method, we combine geometric constraints and motion constraints to remove dynamic line features. Finally, we fix accurate landmarks and only optimize other landmarks in the optimization of BA, resulting in significant improvement of back-end optimization speed. The validation experiments on public and author-collected datasets show that the system exhibits competitive accuracy and robustness in structural environments.

REFERENCES

[1] 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.

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

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

[4] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher, "Real-time manhattan world rotation estimation in 3d," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1913–1920.

[5] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li, "Divide and conquer: Efficient density-based tracking of 3d sensors in manhattan worlds," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*. Springer, 2017, pp. 3–19.

[6] 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.

[7] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. IEEE, 1999, pp. 941–947.

[8] 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.

[9] H. Lim, J. Jeon, and H. Myung, "Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1518–1525, 2022.

[10] 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.

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

[12] C. Chu and S. Yang, "Keyframe-based rgb-d visual-inertial odometry and camera extrinsic calibration using extended kalman filter," *IEEE Sensors Journal*, vol. 20, no. 11, pp. 6130–6138, 2020.

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

[14] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017.

[15] J. Liu, X. Li, Y. Liu, and H. Chen, "Rgb-d inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9573–9580, 2022.

[16] C. Tomasi and T. Kanade, "Detection and tracking of point," *Int J Comput Vis*, vol. 9, no. 137-154, p. 3, 1991.

[17] I. Suárez, J. M. Buenaposada, and L. Baumela, "Elsed: Enhanced line segment drawing," *Pattern Recognition*, vol. 127, p. 108619, 2022.

[18] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of visual communication and image representation*, vol. 24, no. 7, pp. 794–805, 2013.

[19] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Computer vision and image understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[20] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song *et al.*, "Are we ready for service robots? the openloris-scene datasets for lifelong slam," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3139–3145.