

KFS-LIO: Key-Feature Selection for Lightweight Lidar Inertial Odometry

Wei Li, Yu Hu, Yinhe Han and Xiaowei Li

Abstract—Feature-based lidar odometry methods have attracted increasing attention due to their low computational cost. However, theoretically analysis of the effect of extracted features on pose estimation is still lacked. In this paper, we propose a method of key-feature selection for lightweight lidar inertial odometry, KFS-LIO, to further enhance the real-time performance by selecting the most effective subset of lidar feature constraints. Aiming at explaining the correlation between the feature distribution and state errors, a quantitative evaluation method of lidar constraints is introduced. In addition, to avoid recalculating the reprojection matrices in de-skewing step, we use the intermediate variables in IMU preintegration to compensate for lidar motion distortion. The experimental results demonstrate that KFS-LIO can reduce half of the LOAM features and provide comparable accuracy with the state-of-the-art odometry.

I. INTRODUCTION

The estimation of navigation information, including position, speed, attitude and three-dimensional map, is a key issue that needs to be solved when completing autonomous navigation tasks. Lidars and cameras are the most widely used sensors to perceive the environment for Simultaneous Localization And Mapping (SLAM). The Lidar using time-of-flight measurements has advantages over cameras, because its ranging accuracy is insensitive to the distance and illumination. With the development of lidar sensor technology, lidar costs are declining, hence the application range of lidar odometry and mapping is expanding.

Despite the high reliability and precision, a moving lidar usually suffers from the motion distortion, for the ranging measurements are not captured instantaneously. In contrast to cameras, another drawback of lidar is that disordered point cloud cannot provide sufficient semantic information, which makes inter-frame matching and loop closure detection challenging. With the development of lidar odometry in the past decade, feature-based matching methods [1]–[4] are proposed to solve the aforesaid problems and require less computational resources than the typical iterative closest point (ICP) [5] approach. The LOAM [1] among them is considered to be a state-of-the-art algorithm, and has been applied widely in subsequent research [6]–[9]. It reduces the amount of computation by extracting edge and planar points according to the roughness of the surrounding region.

Wei Li, Yu Hu and Yinhe Han are with the Research Center for Intelligent Computing Systems, Institute of Computing Technology, the Chinese Academy of Sciences, Beijing, China. {liwei2019, huyu, yinhes}@ict.ac.cn

Xiaowei Li is with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, the Chinese Academy of Sciences, Beijing, China. lxxw@ict.ac.cn

However, in the odometry module of LOAM, it uses the ego-motion estimation obtained from two adjacent lidar frames to compensate for motion distortion. This module requires time-consuming continuous iteration of pose estimation and de-skewing step until convergence. To avoid the unnecessary recalculation, we utilize the intermediate variables in the Inertial Measurement Unit (IMU) preintegration process [10] to complete the de-skewing. In addition, de-skewed lidar points need to be continuously corrected along with the optimization process, so we record lidar features with the associated frames rather than directly adding them to the global map. Because the preintegrated IMU measurements are independent of the state estimation, the reprojection matrices for de-skewing do not need to be recomputed every time the state changes after optimization. Furthermore, we only retain the most effective subset of lidar features by implementing the key-feature selection module.

In this paper, we propose a Key-Feature Selection for Lidar Inertial Odometry, referred to as KFS-LIO. Inspired by the Dilution Of Precision (DOP) concept [11] in the field of satellite positioning, we introduce a quantitative evaluation method for constraints derived from lidar features. According to the evaluation function, a selection algorithm is presented to exclude the least important feature matching results. Additionally, a formula is derived to determine the appropriate opportunity for loop closure detection, thus no need to detect at every moment. The main contributions of our work has three aspects as following:

- A tightly-coupled lidar inertial odometry framework is proposed. In order to enhance the real-time performance, improvements have been made in the de-skewing and key-feature selection module.
- A quantitative evaluation method of constraints derived from lidar features is introduced. The evaluation function can indicate the correlation between lidar constraints and the uncertainty of state estimation.
- A key-feature selection algorithm is presented to exclude the feature matching relationships with less contribution, so as to achieve an expectable compromise between accuracy and computational cost.

II. RELATED WORK

In the development of lidar odometry, researchers have made a lot of efforts to reduce the computational cost. The iterative closest point (ICP) [5] method aligns the dense points between two scans, hence both its accuracy and computational complexity depend on the large quantity of points. The LOAM [1], as a representative of geometry feature

matching methods, realizes a high-frequency odometry by extracting effective lidar cloud features. In this way, the amount of calculation can be reduced because the edge points and planar points to be matched are more sparse. However, LOAM utilizes the ego-motion estimation to iteratively complete distortion correction, which is similar to VICP [12] and will lead to additional computational overhead.

An alternative way to register de-skewed laser points is to take advantage of the complementary properties of other sensors, such as GPS, visual odometry and IMU [13]–[15]. For the tightly-coupled lidar inertial odometry methods, which IMU is involved in the optimization process, researchers usually enhance the real-time performance by improving back-end estimation algorithms. LINS [16] designs an iterated error-state Kalman filter to ensure that the system is computationally tractable. LIO-SAM [7] implements the lidar inertial odometry through a factor graph, and takes the loop closure constraint based on the Euclidean distance as a new type of factor. In addition, by proposing the approach of sub-keyframes, the scan-matching in LIO-SAM is performed at a local scale rather than the global scale in LOAM. Although these lidar inertial odometry methods have shown feasible computational efficiency, their front-ends still share the same feature extraction idea with LOAM. Thus the real-time performance can be further improved by selecting the most effective subset of lidar feature constraints.

In fact, the selection of features has already been discussed in the previous research, but it is mainly to eliminate invalid points from a technical point of view, without theoretically analyzing the contribution of each feature to accuracy. LOAM excludes feature points which lie on a plane roughly parallel to the laser beam or on the boundary of an occluded area. Loam_livox [17] removes the points with extreme intensity values and incident angles, as well as those near the edge of the field of view or hidden behind obstacles. LIO-mapping [8] states that edge points cannot improve the odometry results in practice. While Lego-LOAM [6] retains both edge points and plane points, and makes efforts to extract features evenly from all directions.

In order to theoretically explain whether the geometric distribution of feature points will affect the accuracy, we use the DOP concept in the satellite positioning field for reference. The DOP reflects the influence of observation noise on the uncertainty of the estimated state, so a small DOP value indicates an ideal satellite geometric distribution. This paper introduces an evaluation method for lidar feature constraints, which further contributes to the real-time improvement of tightly-coupled lidar inertial odometry.

III. LIDAR-INERTIAL ODOMETRY WITH KEY-FEATURE SELECTION

The lidar-inertial odometry proposed in this paper aims to estimate the position and ego-motion state using constraints derived from sensor observations. In order to describe the odometry process clearly, we define three reference frames. The body frame is denoted by $\{B\}$. Its origin is defined at the center of the mobile platform. For convenience, we

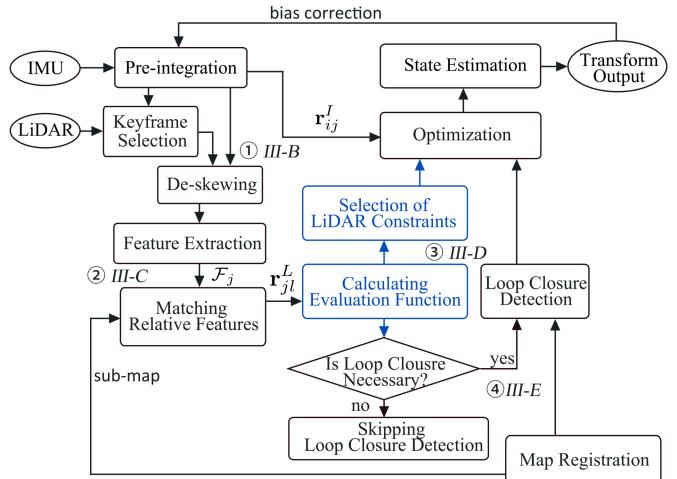


Fig. 1: The framework of the KFS-LIO proposed in this paper. The blue blocks correspond to the key innovations, which are discussed in subsection III-D.

assume that the IMU frame is coincident with frame B . The lidar coordinate frame is denoted by $\{L\}$. The point cloud is measured in this frame. Since sensors are rigidly attached to the mobile platform, the transformation from lidar to IMU is considered to be known from prior calibration. The map frame $\{M\}$, which coincides with the initial body frame, is used to present the trajectory and mapping results. Its z axis is defined as the normal to the surface of the earth ellipsoid. The other two axes are on the plane orthogonal to the z -axis and conform to the right-handed corkscrew rule.

A. Problem Statement and System Overview

We regard the odometry process as a state estimation problem, the state vector at time t_k contains orientation, position, velocity and IMU biases

$$\mathbf{x}_k = \left[\mathbf{R}_k^T, \mathbf{p}_k^T, \mathbf{v}_k^T, \mathbf{b}_k^g, \mathbf{b}_k^a \right]^T. \quad (1)$$

The transformation $\mathbf{T}_k = [\mathbf{R}_k | \mathbf{p}_k]$ from frame B to frame M belongs to $SE(3)$, while velocity \mathbf{v}_k , gyroscope bias \mathbf{b}_k^g , and accelerometer bias \mathbf{b}_k^a belong to \mathbb{R}^3 .

Fig.1 shows an overview of the proposed KFS-LIO. Our algorithm concentrates on a subset of lidar observations, which is called *keyframe*, instead of estimating every state vector when a new observation arrives. In addition, the preintegration technique [10] can not only avoid recomputing high-dimensional matrices in the estimation process, but also provide motion information for keyframe selection. When the movement between the current frame L and the previous keyframe exceeds an artificial threshold, the current frame will be added to \mathcal{K}_n , which denotes the set of all keyframes up to time t_n .

Therefore, in KFS-LIO, the state to be estimated is $\mathbf{X}_n = \{\mathbf{x}_i\}_{i \in \mathcal{K}_n}$. When it is assumed that these measurement errors conform to a zero-mean Gaussian distribution, solving the Maximum A Posteriori (MAP) problem is equivalent to

minimizing the negative log-likelihood, which can be written in the form of the squared Mahalanobis distance

$$\hat{\mathcal{X}}_n = \arg \min_{\mathbf{x}_n} \|\mathbf{r}_0\|_{\Sigma_0}^2 + \sum_{(i,j) \in \mathcal{K}_n} \|\mathbf{r}_{ij}^I\|_{\Sigma_I}^2 + \sum_{j \in \mathcal{K}_n} \sum_{l \in \mathcal{F}_j} \|\mathbf{r}_{jl}^L\|_{\Sigma_L}^2. \quad (2)$$

where \mathcal{F}_j is used to denote lidar features at time t_j . The residual function \mathbf{r}_0 and its covariance matrix Σ_0 depend on the initialization, so will not be discussed in detail here. \mathbf{r}_{ij}^I and Σ_I depend on the IMU preintegration, their definitions will be discussed in III-B. The residual items \mathbf{r}_{jl}^L and Σ_L corresponding to lidar features will be introduced in III-C. After obtaining the residual functions, the feature evaluation and selection method is introduced in III-D. Subsequently, the opportunity decision of loop closure detection in III-E can further avoid unnecessary calculations. Then the weighted least-squares problem of (2) can be solved by the conventional optimization algorithms, such as Gauss-Newton method and Levenberg-Marquardt method; or solved by the mature frameworks, e.g., g^2o [18] and iSAM2 [19].

B. IMU Preintegration and De-skewing

An IMU measures the motion of frame B with respect to an inertial frame. In this paper, we treat frame M as an inertial frame, while ignoring the earth rotation. The preintegration theory [10] makes the assumption that the acceleration and angular rate remain constant within an IMU sampling interval Δt . In the subsequent statement, we ignore the small time gap between lidar and IMU observations caused by loose time synchronization. Considering the effects of biases \mathbf{b} and random noise $\boldsymbol{\eta}$, the motion increments during the time interval between two keyframes at times t_i and t_j can be written as:

$$\begin{aligned} \Delta \mathbf{R}_{ij} &= \prod_{k=i}^{j-1} \text{Exp} \left[\left(\tilde{\omega}_k^B - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd} \right) \Delta t \right] \\ \Delta \mathbf{v}_{ij} &= \sum_{k=i}^{j-1} \Delta \mathbf{R}_{ik} \left(\tilde{\mathbf{a}}_k^B - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad} \right) \Delta t \\ \Delta \mathbf{p}_{ij} &= \sum_{k=i}^{j-1} \left[\Delta \mathbf{v}_{ik} \Delta t + \frac{1}{2} \Delta \mathbf{R}_{ik} \left(\tilde{\mathbf{a}}_k^B - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad} \right) \Delta t^2 \right] \end{aligned} \quad (3)$$

where $\Delta \mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j$, while $\Delta \mathbf{v}_{ij}$ and $\Delta \mathbf{p}_{ij}$ are not related to true physical changes, but to the preintegrated IMU measurements which are independent of the state vector at time t_i . $\text{Exp}(\cdot)$ is the exponential map for $SO(3)$. The superscript B denotes the frame in which the kinematic property is represented, for example, $\tilde{\mathbf{a}}_k^B$ and $\tilde{\omega}_k^B$ denote the measurements of specific force and angular rate at time t_k in frame B . $\boldsymbol{\eta}_k^{ad}$ and $\boldsymbol{\eta}_k^{gd}$ denote the discrete-time random noise. We only give a brief introduction here, for more detailed calculations of the residual error \mathbf{r}_{ij}^I and its covariance Σ_I , please refer to the derivation in [10].

Since motion increment $\Delta \mathbf{T}_{k,k+1} = [\Delta \mathbf{R}_{k,k+1} | \Delta \mathbf{p}_{k,k+1}]$ between time t_k and time $t_k + \Delta t$ can be obtained during the preintegration process, its linear interpolation is used to compensate for the motion distortion. We use $\tilde{\mathcal{P}}_j$ to represent

a lidar cloud, its subscript is consistent with the end time t_j of the sweep. Under the assumption of a constant velocity motion model, the transformation between frame L at time t_k and t can be calculated as:

$$\begin{bmatrix} \Delta \phi_{kt} \\ \Delta \mathbf{p}_{kt} \end{bmatrix} = \frac{t - t_k}{\Delta t} \begin{bmatrix} \Delta \phi_{k,k+1} \\ \Delta \mathbf{p}_{k,k+1} \end{bmatrix}, \quad (4)$$

where $\Delta \phi_{k,k+1}$ consists of the rotation angle and rotation axis of $\Delta \mathbf{R}_{k,k+1}$. For a certain point in $\tilde{\mathcal{P}}_j$ observed at time t , the rotation $\Delta \phi_{kt}$ and translation $\Delta \mathbf{p}_{kt}$ can be used to reprojected the point to the frame L at time t_j .

C. Feature Extraction and Matching

The feature extraction and matching method in this paper is similar to that introduced in LOAM [1]: first a smoothness evaluation variable is used to select edge points and planar points, and then the corresponding edge lines and planar patches in other keyframes will be selected and used to construct residuals. Since the LOAM method needs to utilize the motion estimation obtained from the standalone lidar to correct distorted points, it realizes the feature matching from frame-to-frame at a high frequency and from frame-to-map at a lower frequency. In our algorithm, the motion estimation for de-skewing can be replaced by the intermediate variables in the IMU preintegration, thus only the feature matching from frame-to-submap is retained.

For the de-skewed feature points \mathcal{F}_j , the transformation in the state vector $\mathbf{T}_j = [\mathbf{R}_j | \mathbf{p}_j]$ is used to transfrom \mathcal{F}_j from the frame L at time t_j to frame M . We denote the reprojected features as \mathcal{F}_j^M . When finding the correspondences of \mathcal{F}_j^M , the effective results should mainly exist in the keyframes that are close to the current position \mathbf{p}_j in physical distance. In order to reduce the computational cost, only a set of suitable keyframes is selected to construct a sub-map, and then the feature matching is performed between the current frame and the sub-map. Normally, the sub-map is composed of serval most recent keyframes, but when there is a loop closure exists, the keyframes adjacent to the loop candidate will also be added to the sub-map.

For an edge point l in \mathcal{F}_j^M , $\mathbf{p}_{j,l}^M$ denotes its position represented in frame M , $\mathbf{p}_{i,\alpha}^M$ and $\mathbf{p}_{i,\beta}^M$ denote the positions of its two nearest neighbors in different scans at time t_i . For an planar point l in \mathcal{F}_j^M , $\mathbf{p}_{i,\alpha}^M$, $\mathbf{p}_{i,\beta}^M$, and $\mathbf{p}_{i,\gamma}^M$ denote the positions of three non-collinear points in its corresponding planar patch. The distance between the feature l and its corresponding edge line or planar patch, can be calculated by (5). The resulting residuals \mathbf{r}_{jl}^L derived from lidar features is equivalent to the difference between the true distance $\mathbf{d} = \{d_e, d_p\}$ and the estimated distance $\hat{\mathbf{d}} = \{\hat{d}_e, \hat{d}_p\}$.

$$\begin{aligned} \hat{d}_e &= \frac{\left| (\mathbf{p}_{j,l}^M - \mathbf{p}_{i,\alpha}^M) \times (\mathbf{p}_{j,l}^M - \mathbf{p}_{i,\beta}^M) \right|}{\left| \mathbf{p}_{i,\alpha}^M - \mathbf{p}_{i,\beta}^M \right|} \\ \hat{d}_p &= \frac{\left| \begin{array}{c} (\mathbf{p}_{j,l}^M - \mathbf{p}_{i,\alpha}^M) \\ (\mathbf{p}_{i,\alpha}^M - \mathbf{p}_{i,\beta}^M) \times (\mathbf{p}_{i,\alpha}^M - \mathbf{p}_{i,\gamma}^M) \end{array} \right|}{\left| (\mathbf{p}_{i,\alpha}^M - \mathbf{p}_{i,\beta}^M) \times (\mathbf{p}_{i,\alpha}^M - \mathbf{p}_{i,\gamma}^M) \right|} \end{aligned} \quad (5)$$

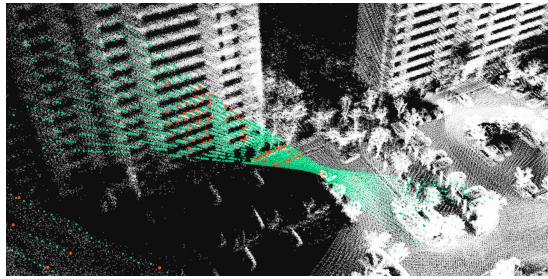


Fig. 2: An example of the extracted edge points (red) and planar points (green). Scan line indicates the line of sight.

D. Feature Evaluation and Selection

Through feature matching, we obtain many constraints derived from lidar measurements, and their corresponding residuals \mathbf{r}_{jl}^L will be used for state estimation. However, in order to reduce the computational cost, we hope to exclude some unnecessary constraints within the acceptable range of accuracy loss, instead of using all data.

1) *Evaluation:* In the lidar odometry process, the line of sight from the mobile platform to a feature point is quite similar to the principle of satellite positioning, as shown in Fig.2. In the research field of satellite, the effect of signal geometry can be quantified using the DOP concept [11]. It indicates how much the measurement noise can affect the uncertainty of the state variable, so a smaller DOP value corresponds to a better geometric distribution. Inspired by the Position Dilution Of Precision (PDOP), an evaluation function for lidar constraints will be derived from the mathematical perspective below.

The state error of position at time t_j is $\delta \mathbf{p}_j$, which is defined as the difference between the true value and the estimation. According to (2), the objective function to be minimized can be expressed as a quadratic in the form of matrix:

$$\mathbf{G}(\mathcal{X}_j) = \mathbf{r}(\mathcal{X}_j)^T \Sigma^{-1} \mathbf{r}(\mathcal{X}_j). \quad (6)$$

Since errors of IMU residual \mathbf{r}_{ij}^I and lidar residual \mathbf{r}_{jl}^L are independent, the effects of IMU and lidar on state error $\delta \mathbf{p}_j$ can be discussed separately. The residual function $\mathbf{r}(\mathcal{X}_j)$ discussed in this subsection is composed of $\mathbf{r}_{jl}^L = \mathbf{d} - \hat{\mathbf{d}}$, where $j \in \mathcal{K}_n$, $l \in \mathcal{F}_j$. Using the Taylor series expansion to first order, the state correction $\delta \mathbf{p}_j$ can be approximated as

$$\delta \mathbf{p}_j = (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1} \mathbf{r}(\mathcal{X}_j), \quad (7)$$

where \mathbf{J} is the Jacobian matrix of $\mathbf{r}(\mathcal{X}_j)$ with respect to \mathbf{p}_j , and Σ represents the weight and is often simplified as an identity matrix. At the results of the iterative optimization, the error covariance matrix for $\delta \mathbf{p}_j$ and \mathbf{p}_j is

$$\text{cov}(\delta \mathbf{p}_j) = (\mathbf{J}^T \mathbf{J})^{-1} \sigma_L^2. \quad (8)$$

The matrix $\mathbf{J}^T \mathbf{J}$ is a 3×3 symmetric matrix and depends on the relative geometry of lidar features according to (5). Note that in (8), we assume that the components of $\mathbf{r}(\mathcal{X}_j)$ are independent and identically distributed. Thus the covariance of $\mathbf{r}(\mathcal{X}_j)$ is defined as a scalar multiple σ_L^2 of the identity.

Algorithm 1: Selection for Lidar Constraints

```

Input:  $\mathbf{r}_{jl}^L$ ,  $\mathbf{J}$ ,  $\lambda$  and  $\mu$ 
Output:  $\mathbf{r}_{jl}^{L,new}$ , and compression rate  $c(\lambda)$ 
1 begin
2   Sort constraints in random order,  $\mathbf{r}_{jl}^{L,new} = \mathbf{r}_{jl}^L$ ;
3   Compute the evaluation value  $E(\mathbf{r}_{jl}^L)$ ;
4   Compute the upper limit:  $E_{sup} = \lambda \cdot E(\mathbf{r}_{jl}^L)$ ;
5   for each element in  $\mathbf{r}_{jl}^L$  do
6     Let  $\mathbf{r}_{jl}^{L,temp}$  consists of elements other than
      the current one in  $\mathbf{r}_{jl}^{L,new}$ ;
7     Delete the corresponding row in  $\mathbf{J}$ , then
      compute  $E(\mathbf{r}_{jl}^{L,temp})$ ;
8     if  $E(\mathbf{r}_{jl}^{L,temp}) \leq \min\{E_{sup}, \mu\}$  then
9        $\mathbf{r}_{jl}^{L,new} = \mathbf{r}_{jl}^{L,temp}$ ;
10      Update  $\mathbf{J}$  with the Jacobian of  $\mathbf{r}_{jl}^{L,new}$ ;
11    end
12  end
13   $c(\lambda) = \frac{\|\mathbf{r}_{jl}^{L,new}\|_0}{\|\mathbf{r}_{jl}^L\|_0}$ ;
14  Return  $c(\lambda)$  and  $\mathbf{r}_{jl}^{L,new}$ ;
15 end

```

A necessary condition of the above assumption is that the noise associated with ranging measurements is independent of the distance. The lidars using time-of-flight measurements meet this requirement, while the cameras can not.

According to the stated derivation, it can be seen that the matrix $(\mathbf{J}^T \mathbf{J})^{-1}$ quantify how lidar residual errors translate into components of the covariance of $\delta \mathbf{p}_j$. When only considering the main diagonal elements of $\text{cov}(\delta \mathbf{p}_j)$, the evaluation function $E(\cdot)$ of features can be computed by:

$$E(\mathbf{r}_{jl}^L) = \sqrt{\text{tr}[(\mathbf{J}^T \mathbf{J})^{-1}]}, \quad (9)$$

where $\text{tr}(\cdot)$ is the trace of a matrix. The difference between the DOP and the evaluation function introduced in this paper is that the Jacobian in the former is directly constituted by the line-of-sight vectors, while the latter needs to calculate partial derivatives of the lidar residual function.

2) *Selection:* Before the feature selection, some inferences can be obtained according to the evaluation method:

- The fewer lidar constraints available, the larger evaluation value and estimation error.
- The evaluation of lidar feature constraints is affected by the relative geometry of points on the corresponding edge lines and planar patches.

In this work, the selection is aimed at the feature matching relationships, rather than directly deleting some feature points from the submap based on the evaluation results. To make an appropriate compromise between accuracy and computational cost, we define an accuracy magnification factor λ , which indicates how many times the state error can be tolerated. We also define the upper bound μ of the evalua-

ation function. The selection method for lidar constraints is described as Algorithm 1. Note that since each element in r_{jl}^L corresponds to a set of feature matching relationships, the element is used to indicate whether a relationship is retained. This selection method can be further accelerated, if a single element in the loop is replaced by a subset containing multiple elements.

The result of this selection method is affected by the order of features, so the selected set may not be the optimal solution. Nevertheless, in the next section of experiments, the effectiveness of this selection method will be demonstrated.

E. The Appropriate Opportunity for Loop Closure Detection

When a loop closure occurs, a decrease in the uncertainty of the state estimation is expected. In this case, if the state estimation errors are small, such as when the odometry has just started, or when the last loop closure detection has just finished, there is actually no need to perform the detection. In this paper, we set a threshold to determine the appropriate opportunity for detection. If the pose is taken as the main consideration, the threshold is set as (10). When the state error exceeds this threshold, it is considered that the accuracy of the estimation problem needs to be improved, thus then the loop closure detection is performed at each keyframe.

$$\sqrt{\text{tr} \left[(\mathbf{J}^T \boldsymbol{\Sigma}^{-1} \mathbf{J})^{-1} \right]} \quad (10)$$

Here \mathbf{J} is the Jacobian matrix of $\mathbf{r}(\mathcal{X}_n)$ with respect to \mathbf{x}_n . The residual function $\mathbf{r}(\mathcal{X}_n)$ contains the lidar residuals \mathbf{r}_{jl}^L and IMU residuals \mathbf{r}_{ij}^I . $\boldsymbol{\Sigma}$ is the corresponding covariance matrix.

IV. EXPERIMENTS AND ANALYSIS

In this section, we evaluate the performance of KFS-LIO in terms of accuracy and computational cost, and compare it with LOAM [1] and LIO-SAM [7]. All methods are implemented in C++ and executed on a computer with the Intel Xeon E5-2650 CPU@2.20GHz, using the Robot Operating System (ROS) [20] in Ubuntu Linux.

The following experiments are performed on the KAIST urban dataset [21], which was collected by a mapping vehicle equipped with two 3D lidar sensors, an Xsens MTi-300 IMU and some other navigation sensors. We choose KAIST dataset to evaluate the algorithms because it provides a development tool that can adjust the publishing speed of sensor measurements. In addition, the Virtual Reference Station (VRS)-GPS can provide positioning results with an accuracy within 10cm, thus its measurements are used as the ground truth. All methods only use the Velodyne VLP-16 lidar on the right side, which was mounted at the rear of the vehicle and tilted about 45 degrees.

A. Evaluation for the KFS-LIO framework

To analyze the accuracy of KFS-LIO, experiments were carried out on three datasets in [21]. The urban08 and urban15 datasets were collected in the residential area, while urban07 dataset was collected in the apartment complex



(a) urban08

(b) urban15

Fig. 3: The mapping results of KFS-LIO, which show a good alignment with Google Map.

TABLE I: Relative errors for position estimation

DataSet	Distance (km)	LOAM	LIO-SAM	KFS-LIO
Urban07	2.549	4.96%	0.96%	0.93%
Urban08	1.56	2.24%	1.57%	1.52%
Urban15	5.43	2.63%	1.63%	1.65%

scenario. For all experiments in this subsection, the accuracy magnification factor for KFS-LIO is set as $\lambda = 1.2$. From the qualitative perspective, the mapping results of KFS-LIO can achieve the lane-level accuracy as shown in Fig.3 .

As for the quantitative perspective, the Root Mean Square Error (RMSE) is used to measure the gap between the estimation and the ground truth provided by VRS-GPS. The relative errors are listed in Table I, which are RMSE results compared to the path length of dataset. The corresponding odometry results are shown in Fig.4. From the results, it can be seen that KFS-LIO has better performance in accuracy than LOAM. In the meantime, KFS-LIO shows comparable accuracy with LIO-SAM when they are both required to run in real-time. We take the experimental results on urban07 as an example, the relative RMSE of KFS-LIO is 0.93%, slightly better than that of LIO-SAM, which is 0.96%. These two relative errors are significantly smaller than the result of LOAM, because there are some loop closure constraints available in urban07 dataset.

B. Evaluation for the Key-Feature Selection Module

The key-feature selection module proposed in this paper also can be used in other lidar odometry methods. To verify the effectiveness of the KFS module, we add it to LOAM. Thus the impact of IMU on state estimation can be avoided.

We set the accuracy magnification factor λ to 1.2, 2.0, and 2.5 for experiments. The LOAM with random feature selection (RFS) is used as the comparison method, and the standard LOAM is regarded as the baseline. The RFS module is designed to randomly delete lidar constraints to ensure that its average compression rate is almost equal to that of KFS module. Table II shows the experimental results of different feature selection modules on urban08 dataset. In Table II, *Aver.c* is the compression rate defined in the subsection III-D, and *Aver.E* is the average value of the evaluation function. At the same time, the resulting boxplots are shown in Fig.5, and the trajectories are shown in Fig.6.

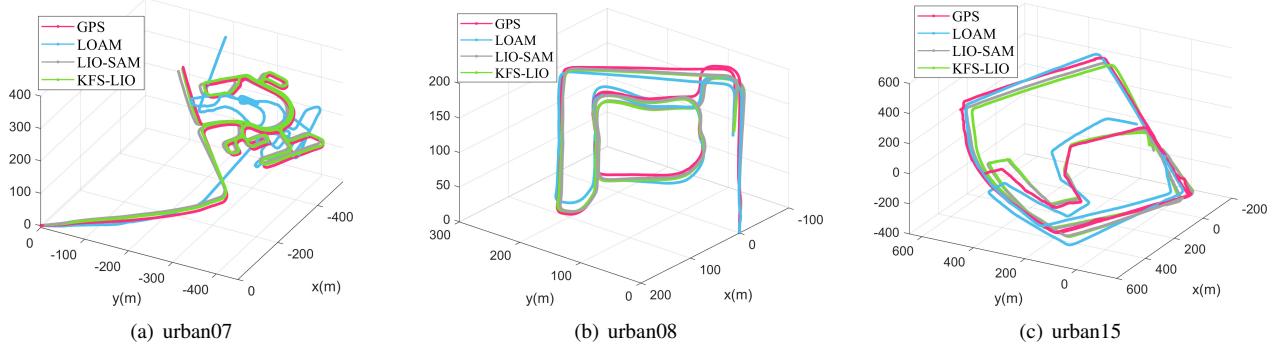


Fig. 4: The trajectories of GPS, LOAM, LIO-SAM and KFS-LIO.

TABLE II: The average compression rate and evaluation value for KFS module and Random module

Factor	$\lambda = 1.2$		$\lambda = 2.0$		$\lambda = 2.5$	
Method	KFS	RFS	KFS	RFS	KFS	RFS
Aver.c	86.1%	86.0%	61.7%	62.0%	47.8%	48.0%
Aver.E	0.26	0.30	0.31	0.48	0.36	0.61

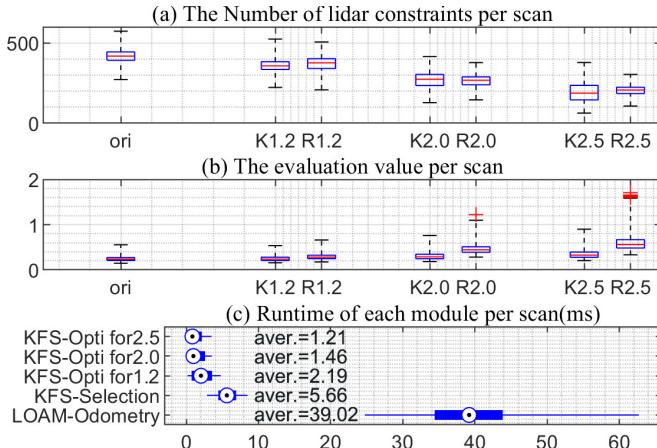


Fig. 5: The boxplot of the constraint numbers(a), the evaluation value(b), and the runtime(c) per scan.

We can see that when $\lambda = 1.2$, 86.1% of the original features can provide a trajectory with a similar accuracy to LOAM. When λ is set to 2.0 and 2.5, 61.7% and 47.8% of the original features are retained, and the resulting trajectories still show no obvious drift. By contrast, the methods with RFS module have failed. In Fig.5, K1.2 is the abbreviation of “KFS $\lambda = 1.2$ ”. From the evaluation value of Fig.5(b), it can be concluded that the smaller evaluation value does indicate the better set of constraints. In practice, it is found that more feature points are needed when rotating, otherwise the accuracy will be degraded; while keeping approximately 35% of the original features is sufficient to maintain acceptable accuracy on the straight path. Additionally, in order to ensure the accuracy and reliability, the upper bound μ is best set to no more than 0.6 when λ reaches a larger value.

Fig.5(c) compares the runtime of each module. Since

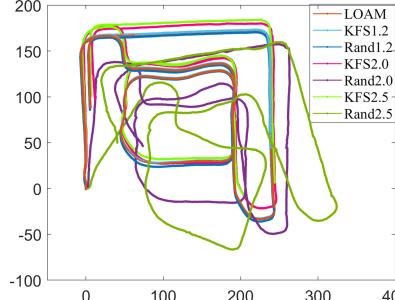


Fig. 6: The trajectories of different feature selection methods.

the odometry module in LOAM iteratively complete the de-skewing, feature matching and optimization, it takes an average of 39 milliseconds to run. As for the KFS module, the runtime is 5.66ms on average. “KFS-Opti for 2.5” in Fig.5(c) represents the runtime of optimization after adding KFS module with parameter $\lambda = 2.5$. It can be seen that the optimization time decreases with the reduction of features.

V. CONCLUSION AND DISCUSSION

A key-feature selection for lidar inertial odometry is proposed for real-time motion estimation and mapping. In order to reduce the computational cost, we introduce a quantitative evaluation method of lidar constraints and present a selection module which only retains the most effective subset of lidar features. The experimental results showed that the extracted lidar features from LOAM are redundant, and the number of feature points can be reduced to 50% according to the proposed method. In the future work, the accuracy magnification factor in KFS module can be adjusted adaptively according to the prior motion information, and the number of feature points is expected to be further reduced.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China under Grant No.62003323, in part by the National Key R&D Program of China under Grant No. 2018AAA0102701, and in part by the Innovation Project of State Key Laboratory of Computer Architecture, ICT, CAS under Grant No. CARCH5203.

REFERENCES

- [1] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [2] B. Steder, G. Grisetti, and W. Burgard, "Robust place recognition for 3d range data based on point features," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1400–1405.
- [3] R. Zlot and M. Bosse, "Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine," in *Field and service robotics*. Springer, 2014, pp. 479–493.
- [4] J. Serafin, E. Olson, and G. Grisetti, "Fast and robust 3d feature extraction from sparse point clouds," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4105–4112.
- [5] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [6] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.
- [7] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," *arXiv preprint arXiv:2007.00258*, 2020.
- [8] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3144–3150.
- [9] J. Zhang and S. Singh, "Enabling aggressive motion estimation at low-drift and accurate mapping in real-time," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5051–5058.
- [10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.
- [11] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.
- [12] S. Hong, H. Ko, and J. Kim, "VICP: Velocity updating iterative closest point algorithm," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1893–1898.
- [13] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4670–4677.
- [14] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "IN2LAMA: Inertial lidar localisation and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6388–6394.
- [15] Z. Wang, J. Zhang, S. Chen, C. Yuan, J. Zhang, and J. Zhang, "Robust high accuracy visual-inertial-laser slam system," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6636–6641.
- [16] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8899–8906.
- [17] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3126–3131.
- [18] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: A general framework for (hyper) graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, 2011, pp. 9–13.
- [19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [21] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.