2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Graph Matching Pose SLAM based on Road Network Information

Lei He, Ming Yang*, Hao Li, Yuesheng He, Bing Wang and Chunxiang Wang

Abstract— **This paper presents a Graph Matching Pose SLAM to build the perception map which is a prerequisite of localization and environment perception of the intelligent vehicle. Graph-based simultaneous localization and mapping (SLAM) is widely used to build a map with global consistency and requires loop closure to eliminate the accumulative errors. However, loop closure forces the vehicle to re-visit a previously entered area. In this paper, a road network-based graph SLAM method without the requirement of loop closure is proposed to build a consistent map for the real environment. The estimated poses from the original SLAM are linked with the available road network based on the graph matching. Thus the available road network introduces the real topology of the environment to Pose SLAM regularly. In this way, the map is optimized by using the factor graph inference frequently. The proposed method is validated on the KITTI dataset and the real-world experiments. This method outperforms the state-of-the-art methods.**

I. Introduction

An intelligent vehicle is often required to construct consistent models of the environment. In the intelligent vehicle field, these models are usually maps of the surrounding environment that are vital to localization [1] [2] and environment perception. In general, this critical problem of estimating a map under uncertainty is referred to as the simultaneous localization and mapping (SLAM) problem which has been of great interest to many researchers [3].

Most SLAM algorithms concentrate on producing a globally consistent map using a series of sensor observations without taking account of prior information from the environment. The special global priors effectively extend application scenarios for SLAM. For instance, there exists a common scenario where most intelligent vehicles are required to autonomously arrive at a location area with a specific precision global positioning system (GPS) coordinate. Furthermore, due to wrong data associations, even sophisticated SLAM techniques suffer from a drift in the trajectory estimation in absence of constraints. Performing loop closing can eliminate the drift, but it forces the vehicles to detect previously entered areas [4] (e.g., re-visiting the identical intersection). Meanwhile, GPS information can be used to fully compensate for the drift without the requirement of loop closure [5] [6]. Whereas, intelligent vehicles utilizing GPS might compromise when GPS encounters occlusions (e.g., in the presence of tunnels, skyscrapers, or narrow streets).

Nowadays, some effective methods utilizing available global priors are gaining attention in the SLAM community (e.g., aerial photographs information [7] or building information from OpenStreetMap [8]). Using these available priors demonstrate impressive results in the GPS-denied and cluttered environment. However, these two methods still have a defect of the particular sensor requirement, e.g., [8] is extremely dependent on laser range measurements to match building information, so there is a big challenge for those approaches to extend to other sensors. In contrast with those approaches, The proposed method greatly facilitates the acquisition of more general solutions.

This paper proposes a novel and reliable approach to mainly utilize the prior road network for calculating global constraints with a Pose SLAM [9] representation. In this method, these global constraints are obtained by relating the pose estimation sequence with the road network. This process employs the filter-based segment sequence estimation and the graph matching. In the graph matching formula, nodes of road network are used as observation variables while nodes of nearby vehicle's pose estimates are regarded as candidate latent variables. This prior road network contains the real topology information of the environment. The real topology information enables our technique to provide a more accurate solution by correcting the long-term drift. We integrate the ability to relate the obtained vehicle's pose estimate sequence with the road network into the factor graph representation [10] of Pose SLAM. This representation performs inference to optimize poses and the map. Perhaps most notably, our approach does not emphasize particular sensors and can be applied to most types of sensors, such as LiDARs and cameras. Compared to the methods that seek to localize vehicle exploiting the prior map directly [11], the proposed approach can operate reliably even in absence of available priors (e.g., lack of appropriate matches sometimes). The system framework is shown as Fig. 1.

This proposed method makes three main contributions.

- An effective road network-based Pose SLAM is proposed in a factor graph representation. This method aligns the vehicle's trajectory to the road network and estimates the most likely trajectory.
- Compared with methods requiring particular sensors, it just relies on the local pose estimate sequences and it can be integrated with most SLAM frameworks.
- This approach utilizes the environment topology information from the road network to correct the long-term drift and limit the accumulated errors. Our approach outperforms the state-of-the-art methods in the KITTI and real-world experiments.

This work is supported by the National Natural Science Foundation of China (U1764264/61873165), Shanghai Automotive Industry Science and Technology Development Foundation (1733/1807). Ming Yang is the corresponding author.

Lei He, Ming Yang, Hao Li, Yuesheng He, Bing Wang and Chunxiang Wang are with the Key Laboratory of System Control and Information Processing, Ministry of Education, Department of Automation, Shanghai Jiao Tong University, 200240 Shanghai, China (email: MingYang@sjtu.edu.cn).

978-1-7281-0560-4/19/$31.00 ©2019 IEEE

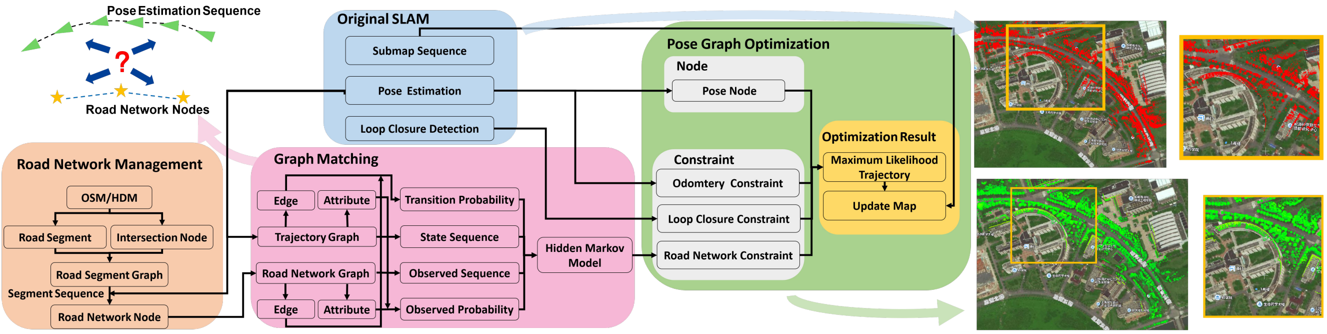


Fig. 1. System framework. The correspondence of the road network and the pose estimation sequence uses the graph matching (red rectangle) after generating road network sequence (orange rectangle). The green rectangle achieves pose graph optimization and outputs a map that is more consistent than the original SLAM (blue rectangle).

II. Road Network Management

Generating a road network is a prerequisite for the Graph Matching Pose SLAM system. This section describes how to create and manage the road network.

Constructing the Road Network. The road network is acquired directly from the OpenStreetMap (OSM) or high definition map (HDM). From the map, crossings and drivable roads with their traffic directions are extracted to output the road network that contains two types of elements, street segments and intersection nodes. With the least squares fitting, each segment is denoted by a consistent representation $segment = (\vec{x}_0, \vec{x}_1, \vec{r}_0, R, L, type)$. $\vec{x}_0$, $\vec{x}_1$ are the starting and ending road nodes in the segment, respectively. The direction of the node is represented by the direction of the vector from the previous road network node to this. $\vec{r}_0$ and R represent the position of the center of the circle by the least squares fitting and the radius of it. L is the length of the segment. There are two types of segments: the linear or arc segments.

The Directed Road Segment Graph. We assume that the road network data is denoted by a directed graph $G = (V, A)$ where vertex set V represents the set of road segments and edge set A defines the connectivity of the roads. The edge (i, j) from node i to node j represents that the vehicle can travel from the i^{th} segment to the j^{th}. Especially, an intersection point typically locates in the center of the street crossing and its position cannot match with the vehicle trajectory. This may hinder the matching procedure from performing the correct association between the road network and the pose estimate sequence. Therefore, the intersection point is substituted by nodes on an arc with a radius R_H.

III. Graph Matching Pose SLAM

Our system is based on a factor graph formulation of Pose SLAM. In addition to the direct association between successive poses, it can integrate environment topology information from a road network (when available). This information is extended to the Pose SLAM framework as the global constraints on the nodes of the factor graph. At first, the filter method is applied to achieve the road segment sequence estimation. Secondly, we define the road network and the pose estimate sequence as two graphs and translate the association problem as a graph matching formulation. Thereafter, the Hidden Markov Model (HMM) are described as a solution to the problem.

A. Matching Road Network and Pose Estimate Sequence

As pose estimates from the original SLAM are obtained at discrete time intervals, we choose a discrete time model. But our method can also be applied to the continue-time SLAM [12] using the sampling strategy. The pose estimation at time t is described as $z_t = (l, \theta)$. l and θ represent the distance and direction displacement to the pose associated with the starting road nodes in the current segment, respectively. With the pose estimate, the road network map and the initial road segment position, the solution to the matching problem for the road network and pose estimate sequence maximizes the product $p(S, k_{0:t}|z_{0:t}, m)$. The k_t represents the index of the road segment that the vehicle travels into at time t, and m represents the road network map. S represents the index set of pose estimates associated with the m. The matching model is factorized as

$$p(S, k_{0:t}|z_{0:t}, m) = \underbrace{p(S|k_{0:t}, z_{0:t}, m)}_{GM} \cdot \underbrace{p(k_{0:t}|z_{0:t}, m)}_{Segment} \quad (1)$$

The problem is divided into two parts. At first, we supply road segment sequence estimation (Segment part) using the Discrete Bayes Filter [13]. Then graph matching (GM) is applied to get a set of latent variables from pose estimates.

B. Road Segment Sequence Estimation

The model of road segment sequence estimation consists of computing the distribution $p(k_{0:t}|z_{0:t}, m)$. The algorithm is divided into a prediction step using the state transition model and a measurement step using the observation model

$$p(k_{0:t}|z_{0:t}, m) = \underbrace{p(z_t|k_t, m)}_{measurement} \cdot \underbrace{p(k_t|k_{t-1}, m)}_{prediction} \cdot p(k_{0:t-1}|z_{0:t-1}, m). \quad (2)$$

With equiprobable state transition probabilities, the model can be expressed as

$$p(k_t|k_{t-1}, m) = \begin{cases} \frac{1}{degree(k_{t-1})+2} & k_t \in \{k_{t-1}, \neg road, N(k_{t-1})\} \\ 0 & otherwise, \end{cases} \quad (3)$$

where $N(\bullet)$ and $degree(\bullet)$ represent the connected segments and the degree of the vertex in the segment graph, respectively. The observation model is factored using the product rule as $p(z_t|k_t) = p((z_t)_l|k_t)p((z_t)_\theta|k_t, (z_t)_l)$. The Gaussian model is applied to define the probability of distance part

$$p((z_t)_l|k_t) = \begin{cases} \int_{l_{u_{t-1}}}^{l_{u_{t-1}}+l_{u_t}} N(x;(z_t)_l,\Sigma_l)dx & k_t \in N(k_{t-1}) \\ \int_{-\infty}^{l_{u_{t-1}}} N(x;(z_t)_l,\Sigma_l)dx & k_t = k_{t-1}. \end{cases} \quad (4)$$

Assumed as a zero-mean normal distribution, the direction part is reasonably represented as

$$p((z_t)_\theta|k_t,(z_t)_l) = N((k_t)_\theta;(z_t)_\theta,\Sigma_\theta) \quad (5)$$

$$(k_t)_\theta = \begin{cases} (k_t)_{(\overrightarrow{x_0})_\theta} & k_t \in N(k_{t-1}) \\ (k_t)_{\Delta\theta}\cdot(z_t)_l/(k_t)_L + (k_t)_{(\overrightarrow{x_0})_\theta} & k_t = k_{t-1}. \end{cases} \quad (6)$$

In order to simplify the process, we regard that the crossing belongs to the previous street segment. An angular velocity threshold w is utilized to recognize the turn of the crossing and the length of the route is for the straight. The observation model for the non-road areas is calculated as

$$p(z_t|k_t)_{\neg road} = p((z_t)_l|k_{t-1})(1-p((z_t)_\theta|k_{t-1},(z_t)_l)). \quad (7)$$

Non-road relocalization. As a vehicle enters a non-road area, the process searches for a nearby segment. If the distance to the segment is beyond a threshold d_s, the initialization process for the relocalization is triggered. A fixed number of initialization states N_R is required for this robust relocalization.

C. Graph Matching using HMM

Modeling the Graphs. The two graphs are always directed graphs considering traffic directions. The representation of the road network graph is simply generated by nodes of the road network sequence. Define the set of road nodes $i \in \nu$ as vertices set of road network directed graph. The set of edges $e_{ij} \in \varepsilon$, where $i, j \in \nu$, also represents as the ordered path (i, j) when the vehicle is able to tour from i to j directly. The road length between two nodes i, j is denoted by the weight of edges W_{ij} connecting them. Notably, an additional attributes item P that measures the location and heading properties of road nodes is introduced to the form of the directed graph. With the four items above, the road network directed graph is denoted as $G_N = (\nu, P, \varepsilon, W)$. The pose directed graph is defined as $G_s = (u, Q, \varepsilon_s, M)$. According to a given distance threshold σ_t, the set of vertices is denoted as the set of candidate pose nodes $p \in u$ selected by the distance to the nearby road network nodes. Moreover, the set of edges $d_{p,m} \in \varepsilon_s$ represents the connectivity of nodes p, m corresponding to the two adjacent road network nodes. The weight of edges $M_{p,m}$ denotes the path distance between p, m, and Q stands for the additional attributes item of pose nodes. An HMM-based method is utilized to search for the best matching of these graphs. Each component of this HMM-based method is described overall as follows.

Observed Sequence and Latent Variables. With road network map m and $k_{0:t}$, the sequence of the road network nodes serves as the observed sequence $O = \{o_1, o_2 ... o_n\}$. The set of latent variables $S = \{s_1, s_2 ... s_n\}$ represents the index of associated pose nodes of G_s. For each latent variable s_i, a set of related candidate pose nodes $Z = \{p_m^i, p_{m+1}^i ...\}$ is extracted from $z_{0:t}$ according to the distance threshold σ_t.

Observation Probability. Attribute similarity of two directed graphs forms observation probability with two types of components: the spatial information $P_d(o_i|p_m^k)$ and directional information $P_h(o_i|p_m^k)$. The two parts are integrated as the element $b_{m,i}$ of the observation probability matrix B

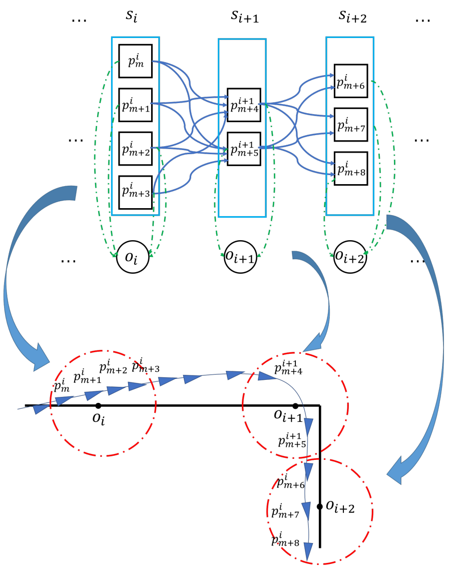


Fig. 2. Hidden Markov Model and trajectory. p_m^i,p_{m+1}^i,p_{m+2}^i,p_{m+3}^i (the blue triangle) are candidate pose nodes associated with the road network node o_i (the black point).

$$b_{m,i} = P_d(o_i|p_m^k) \times P_h(o_i|p_m^k). \quad (8)$$

The spatial component represents the relative distance between the observed point and the candidate pose node. Assumed as a zero-mean normal distribution, the component is reasonably expressed as

$$P_d(o_i|p_m^k) = \begin{cases} N(\|o_i - p_m^k\|, \sigma_1) & if \quad i = k \\ 0 & otherwise, \end{cases} \quad (9)$$

where $\|\bullet\|$ represents the least squares norm, and p_m^k stands for the m^{th} candidate point of the observed node o_i.

The directional information denotes the direction difference between the candidate pose and the observed node. With the expression of a normal distribution, the component is formalized as

$$P_h(o_i|p_m^k) = \begin{cases} N(\|\theta(o_i) - \theta(p_m^k)\|, \sigma_2) & if \quad i = k \\ 0 & otherwise, \end{cases} \quad (10)$$

where $\theta(o_i)$ and $\theta(p_m^k)$ denote the direction of the road network nodes i and the candidate pose nodes p_m^k, respectively.

Transition Probability. Due to the drift of the pose estimation, the nearest candidate pose point is not usually the best match of the road network point. So the topological information exploiting the weights of edges of two graphs is calculated to improve the matching result. The road length of the two neighboring nodes $i, i+1$ is considered as this topological information of the road network, and the length of the route from the candidate points p to m stands for the topological information of the pose node. With the distance ratio of them, the component takes the form

$$P(p_p^{i+1}|p_m^i) = \frac{A_{G_N}(o_t, o_{t+1})}{A_{G_s}(s_p, s_m)}, \quad (11)$$

where A_{G_N} and A_{G_s} are the adjacency matrices for the graph G_N and G_s, respectively.

Given the observation and transition probabilities, this matching problem is translated into an HMM problem explicitly indicated as the Fig. 2. With equiprobable initial

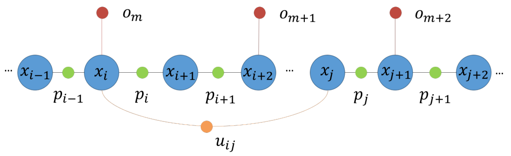


Fig. 3. The factor graph. It is comprised of three types of factors: odometry (green circle), loop closure (yellow) and road network factors (red).

states, the problem is formulated to find the best matching that maximizes the HMM probability

$$\underset{S}{\operatorname{argmax}}\, p(Z,O|S) = \underset{S}{\operatorname{argmax}} \prod_{i=1}^{n} P(p_p^{i+1}|p_m^i)P_h(o_i|p_m^k)P_d(o_i|p_m^k). \quad (12)$$

The problem can be solved by applying the Viterbi algorithm.

D. Pose Graph Optimization using Road Network

As a loop closure is observed or a set of new road network constraints with fixed number N is found, the pose graph optimization thread is triggered. After each optimization procedure, the optimized poses in the map are integrated and the current pose estimate is updated in time. This optimization solution initializes the next procedure effectively. In the pose graph optimization process, three types of factors are considered (Fig. 3).

The odometry factor connects two correlated poses x_{i-1} and x_i using transformation p_{i-1} between them. The loop closure factor indicates that the vehicle re-visits the previously explored place. The road network factor connects the pose estimation x_q and the observed road network node o_m. Here the iSAM2 [10] algorithm is employed to solve it.

IV. EXPERIMENTAL RESULTS

LiDAR data and stereo camera data from the KITTI dataset are used to test the proposed algorithm. Meanwhile, our algorithm is evaluated online in the campus environment using a low-cost 3D LiDAR sensor. This algorithm operates on a hardware arrangement with Intel Core i7-7567U and 16GB RAM. Critical parameters applied in the experiments are summarized in Table I. Two global metrics proposed in [14] with the track alignment strategy and [7] without this strategy are used to evaluate the performance of the algorithm.

A. Testing on the KITTI Dataset

This system is tested on the odometry datasets of the KITTI Vision Benchmark [15] at first. 11 training sequences with the ground truth data are utilized to evaluate the system. We use data from the Velodyne HDL-64E S2 and Point Grey Flea 2, respectively. This algorithm is applied to improve the results of an original visual SLAM framework, ORBSLAM2 [16], and a LiDAR SLAM framework, SuMa [17], respectively. The data is used from the OSM as available road network maps. The ground truth tracks are projected onto the road network maps and errors of the road network maps are calculated as oracle errors [11]. Compared with two original SLAM frameworks, this method supplies better results as illustrated in Table II. The values of improvement η [18] of our method compared to the original SLAM frameworks are also shown in Table II. Without the alignment strategy, the RMSE and S.D. improvement values for SuMa can reach up to **76.27** % and **81.29** % respectively, while they for ORBSLAM2 can reach up to **63.63** % and **71.62** % respectively. The results indicate that the topology constraints from the road networks correct the drifts of the original SLAM frameworks. The proposed method can sharply improve the accuracy of SLAM system, especially in the scenes in absence of loop closure such as KITTI 01. The trajectory comparison results are described in Fig. 4 and Fig. 5.

TABLE I
PARAMETERS OF PROPOSED METHOD

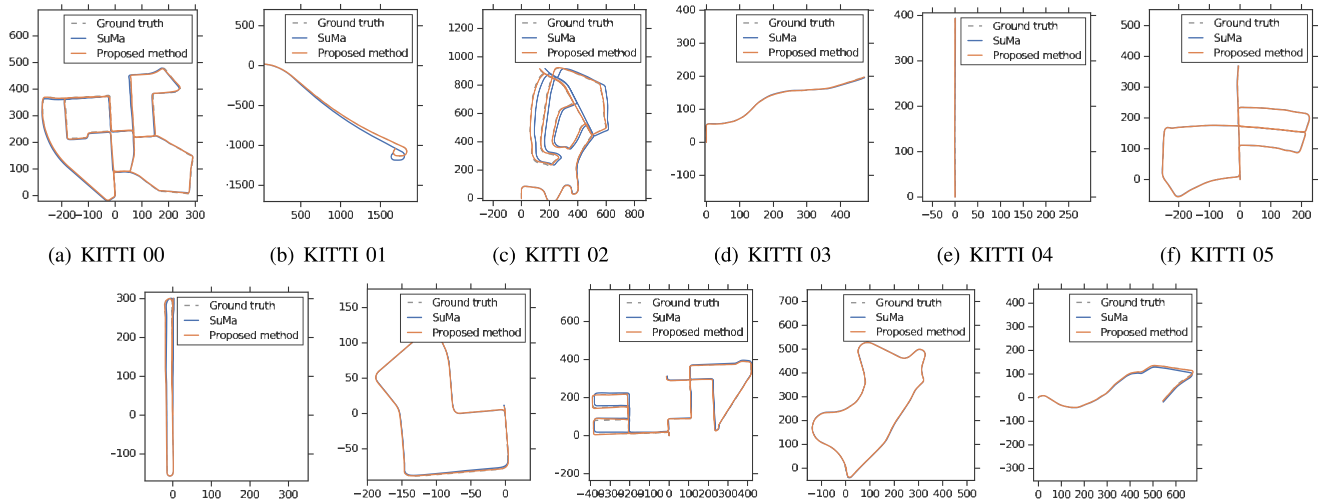
Notation	Value OSM / HDM	Description
L	40 m	Road length threshold to build constraints
N	10	The number of segment in a period
σ_1	1 m / 0.2 m	Position Gaussian noise in the HMM
σ_2	0.5 °	Angle Gaussian noise in the HMM
σ_t	20 m	Distance candidate threshold for HMM
σ_α	2 °	Angle Gaussian noise for Segment part
w	2.5 °/s	Angular velocity threshold for the turning
σ_l	0.1 m	Length Gaussian noise for Segment part
N_R	20	Relocalization pose number
R_H	10 m	Hypothetical corner radius in the crossing
d_s	3 m / 1 m	Distance threshold for the relocalization

B. Testing on the Real-world Environment

1) Testing Platform and Environment: The testing platform and sensor configuration are depicted in Fig. 6. A Velodyne VLP-16, equipped on top of the vehicle, is applied to validate this algorithm. The real-time kinematic (RTK) GPS is used as ground truth. The proposed method is tested in Shanghai Jiao Tong University Minhang Campus online. The driving speed is about 5 m/s during the experiment. This method is tested in an area of 530 m × 970 m where the road network nodes may be unavailable in some subareas as Fig. 8(a). A road network with the lane-level position accuracy of 0.15 ± 0.05 m [11] is generated from the HDM.

2) Trajectory Error: LOAM [19] is employed as LiDAR odometry for our method. Our method is compared with the result of LOAM and a state-of-the-art method of graph-SLAM, the combination of LOAM and Segmatch [20], to evaluate the accuracy [7]. Fig. 7 illustrates that the trajectory of the proposed method is closer to the ground truth than other methods, which proves that the proposed method is effective by using the road network. Fig. 8(a) shows the translation error along the trajectory. Fig. 8(b) describes that our method supplies a considerable improvement. There are some outliers due to the approximatively linearization of the turning behavior in experiments. The mean translation error of the proposed method is **0.43** m and the improvement value of this error is **70.95** %.

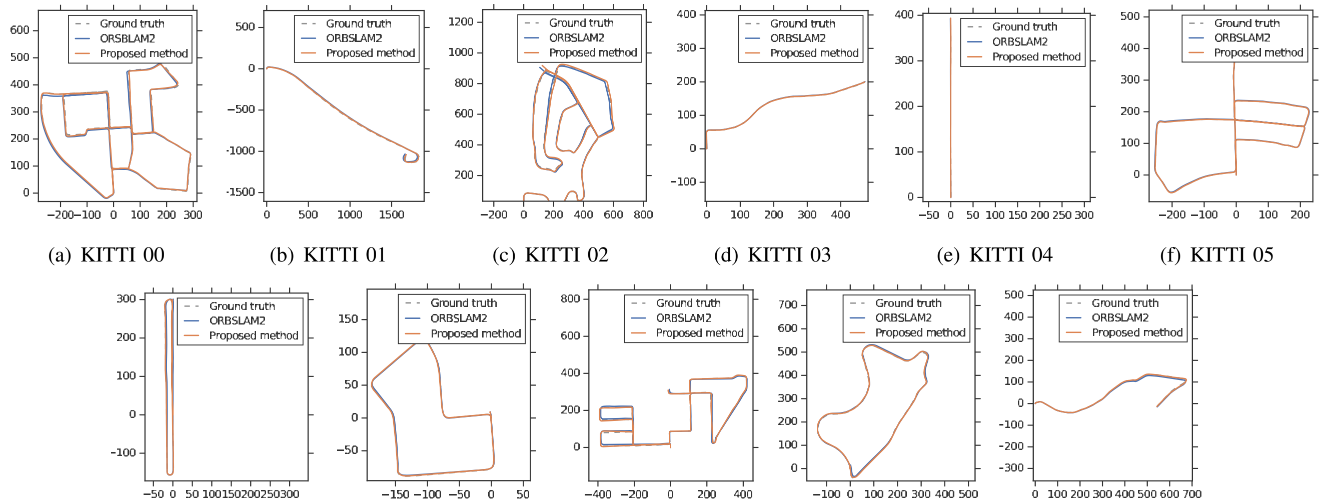
3) Map Alignment: The point cloud map in the experiment is shown as Fig. 9. The edges of point cloud are sharp, indicating good accuracy on the map. The point cloud is



(a) KITTI 00 (b) KITTI 01 (c) KITTI 02 (d) KITTI 03 (e) KITTI 04 (f) KITTI 05

(g) KITTI 06 (h) KITTI 07 (i) KITTI 08 (j) KITTI 09 (k) KITTI 10

Fig. 4. The LiDAR SLAM trajectories of the KITTI training set.



(a) KITTI 00 (b) KITTI 01 (c) KITTI 02 (d) KITTI 03 (e) KITTI 04 (f) KITTI 05

(g) KITTI 06 (h) KITTI 07 (i) KITTI 08 (j) KITTI 09 (k) KITTI 10

Fig. 5. The visual SLAM trajectories of the KITTI training set.

TABLE II

QUANTITATIVE EVALUATION IN THE KITTI DATASET

			00			**01**			**02**			**03**			**04**			**05**	
		RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.
Aligned	S	1.14	1.01	0.54	14.59	13.81	4.72	8.00	7.10	3.69	1.01	0.92	0.42	0.34	0.31	0.12	0.67	0.61	0.28
	S+R	**0.94**	**0.81**	**0.48**	**3.78**	**3.19**	**2.03**	**4.98**	**4.24**	**2.61**	**0.82**	**0.78**	**0.27**	**0.18**	**0.17**	**0.06**	**0.46**	**0.38**	**0.26**
	$\eta_{(S+R)}$	**17.53%**	**19.96%**	**11.09%**	**74.09%**	**76.90%**	**56.99%**	**37.75%**	**40.28%**	**29.27%**	**18.81%**	**15.22%**	**35.71%**	**47.63%**	**46.06%**	**49.06%**	**31.08%**	**36.95%**	**8.74%**
	O	1.30	1.16	0.60	9.58	9.07	3.10	6.57	5.44	3.68	0.76	0.64	0.40	0.25	0.21	0.13	0.75	0.69	0.31
	O+R	**0.99**	**0.84**	**0.53**	**4.23**	**4.02**	**1.32**	**4.05**	**3.45**	**2.12**	**0.52**	**0.46**	**0.23**	**0.21**	**0.19**	**0.08**	**0.66**	**0.59**	**0.30**
	$\eta_{(O+R)}$	**23.60%**	**27.57%**	**11.75%**	**55.85%**	**55.68%**	**57.42%**	**38.36%**	**36.58%**	**42.39%**	**31.58%**	**28.13%**	**42.50%**	**16.16%**	**7.92%**	**38.46%**	**11.68%**	**14.41%**	**3.23%**
Without aligned	S	6.98	6.33	2.91	81.63	62.41	52.61	29.98	26.25	14.48	1.49	1.09	1.02	0.91	0.68	0.61	1.16	1.04	0.51
	S+R	**2.64**	**2.55**	**0.68**	**3.96**	**3.03**	**2.54**	**5.88**	**5.10**	**2.94**	**1.12**	**1.04**	**0.42**	**0.35**	**0.32**	**0.16**	**0.61**	**0.55**	**0.27**
	$\eta_{(S+R)}$	**62.21%**	**59.74%**	**76.60%**	**95.15%**	**95.15%**	**95.17%**	**80.39%**	**80.57%**	**79.70%**	**24.83%**	**4.59%**	**58.82%**	**61.06%**	**53.15%**	**74.59%**	**47.09%**	**47.08%**	**46.75%**
	O	7.79	7.01	3.39	22.56	19.80	10.81	13.34	11.37	6.97	1.88	1.66	0.88	0.78	0.70	0.33	1.62	1.45	0.70
	O+R	**3.79**	**3.70**	**0.82**	**5.33**	**4.84**	**2.22**	**5.41**	**5.01**	**2.03**	**1.32**	**1.20**	**0.55**	**0.24**	**0.21**	**0.13**	**1.23**	**1.20**	**0.28**
	$\eta_{(O+R)}$	**51.36%**	**47.24%**	**75.72%**	**76.37%**	**75.56%**	**79.46%**	**59.45%**	**55.94%**	**70.88%**	**29.79%**	**27.71%**	**37.50%**	**68.77%**	**70.44%**	**61.05%**	**24.19%**	**17.46%**	**60.62%**
Oracle error		-	0.80	0.60	-	1.30	1.20	-	1.30	0.80	-	2.50	1.80	-	-	-	-	1.30	1.10

			06			**07**			**08**			**09**			**10**			**Average**	
		RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.
Aligned	S	0.61	0.57	0.23	1.08	0.99	0.44	3.68	3.38	1.46	1.20	1.10	0.48	1.39	1.34	0.38	3.52	3.20	1.46
	S+R	**0.52**	**0.47**	**0.21**	**0.85**	**0.79**	**0.32**	**2.58**	**2.19**	**1.36**	**0.84**	**0.86**	**0.43**	**0.97**	**0.91**	**0.32**	**2.07**	**1.79**	**1.06**
	$\eta_{(S+R)}$	**15.19%**	**17.04%**	**8.80%**	**21.30%**	**20.20%**	**27.27%**	**29.84%**	**35.06%**	**6.89%**	**29.98%**	**22.00%**	**11.17%**	**30.43%**	**31.80%**	**16.80%**	**41.08%**	**44.14%**	**27.14%**
	O	0.63	0.61	**0.15**	0.59	0.55	0.20	3.27	2.93	1.44	3.18	2.57	1.88	1.19	1.05	0.54	3.06	2.66	1.48
	O+R	**0.46**	**0.42**	0.18	**0.39**	**0.33**	**0.20**	**1.97**	**1.66**	**1.06**	**1.90**	**1.64**	**0.96**	**1.04**	**0.90**	**0.51**	**1.88**	**1.62**	**0.93**
	$\eta_{(O+R)}$	**27.25%**	**31.35%**	-16.22%	**33.58%**	**39.09%**	**0.21%**	**39.77%**	**43.24%**	**26.72%**	**40.15%**	**36.05%**	**48.93%**	**12.80%**	**14.03%**	**5.19%**	**38.60%**	**38.88%**	**37.11%**
Without aligned	S	2.47	1.89	1.59	1.80	1.39	1.14	13.97	13.82	2.03	2.37	2.08	1.14	5.72	4.78	3.14	14.57	12.61	6.80
	S+R	**0.71**	**0.66**	**0.24**	**1.31**	**1.14**	**0.64**	**6.62**	**6.62**	**1.53**	**1.14**	**1.09**	**0.35**	**2.36**	**2.19**	**0.88**	**3.46**	**3.21**	**1.27**
	$\eta_{(S+R)}$	**71.43%**	**64.83%**	**85.09%**	**27.22%**	**17.99%**	**43.86%**	**52.61%**	**52.09%**	**24.56%**	**51.72%**	**47.65%**	**69.13%**	**58.78%**	**54.27%**	**71.83%**	**76.27%**	**74.57%**	**81.29%**
	O	2.02	1.74	1.02	0.96	0.86	0.43	11.73	11.44	2.60	7.33	6.04	4.16	5.50	4.50	3.11	8.53	7.61	3.67
	O+R	**0.93**	**0.92**	**0.18**	**0.84**	**0.79**	**0.27**	**2.61**	**2.45**	**0.92**	**2.84**	**2.56**	**1.23**	**1.72**	**1.48**	**0.87**	**3.10**	**2.91**	**1.04**
	$\eta_{(O+R)}$	**53.78%**	**47.38%**	**82.07%**	**12.62%**	**7.92%**	**36.75%**	**77.71%**	**78.61%**	**64.57%**	**61.28%**	**57.67%**	**70.38%**	**68.75%**	**67.02%**	**72.13%**	**63.63%**	**61.78%**	**71.62%**
Oracle error		-	-	-	-	0.60	0.40	-	1.10	0.90	-	1.20	1.20	-	1.10	0.8	-	1.20	1.00

”S” and ”O”indicate SuMa and ORBSLAM2, and ”R” represents the road network constraint. η represents the value of improvement [18].

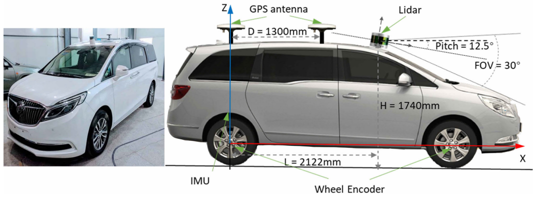


Fig. 6. The Experimental Platform.

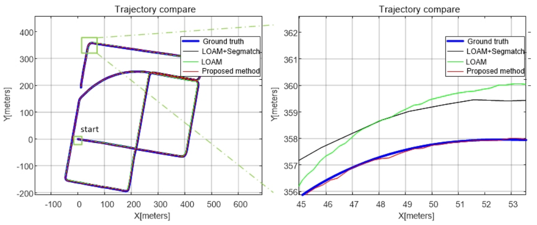
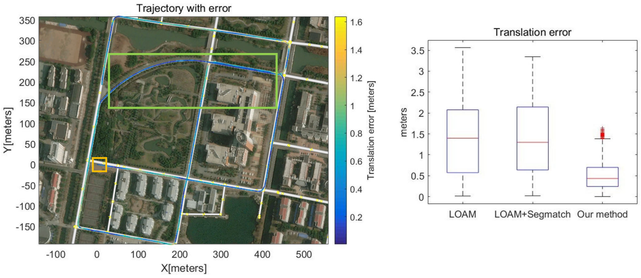


Fig. 7. The 3.1 km trajectory.



(a) Error with trajectory (b) The boxplot

Fig. 8. The Trajectory error. In Fig. 8(a), the road network (white line) may be lack of segments (green box) and the yellow box is the start.

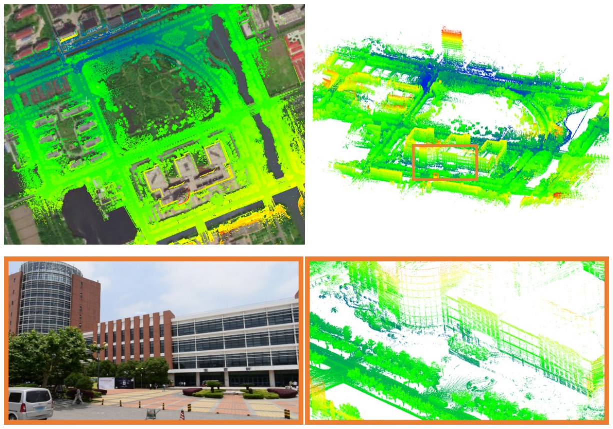


Fig. 9. Generating the point cloud map in the experimental route.

projected onto the satellite picture and it has good alignment with the outlines of buildings.

V. CONCLUSION

In the paper, a Pose SLAM method combined with road networks is proposed. The combination of the filter-based road segment sequence estimation and the graph matching is employed to obtain global road network constraints at first. Then a factor graph representation performs inference to optimize poses and the map regularly. As the road network constraints introduce the real topology of the environment to correct the long-term drift of the original SLAM, this method facilitates major precision improvements for open original SLAM implementations in the experiments. In the future, we plan to apply this method to the continue-time SLAM.

REFERENCES

[1] L. Liang, Y. Ming, L. Guo, C. Wang, and W. Bing, "Precise and reliable localization of intelligent vehicles for safe driving," in *International Conference on Intelligent Autonomous Systems*, 2016.

[2] L. Li, M. Yang, C. Wang, and B. Wang, "Gaussian mixture model-signature quadratic form distance based point set registration," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 998–1003.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec 2016.

[4] M. Labb and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based slam," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2661–2666.

[5] A. U. Shamsudin, K. Ohno, R. Hamada, S. Kojima, T. Westfechtel, T. Suzuki, Y. Okada, S. Tadokoro, J. Fujita, and H. Amano, "Consistent map building in petrochemical complexes for firefighter robots using slam based on gps and lidar," *ROBOMECH Journal*, vol. 5, no. 1, p. 7, 2018.

[6] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and M. E. López, "Real-time hierarchical outdoor slam based on stereovision and gps fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 440–452, 2009.

[7] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, "Large scale graph-based slam using aerial images as prior information," *Autonomous Robots*, vol. 30, no. 1, pp. 25–39, 2011.

[8] O. Vysotska and C. Stachniss, "Improving slam by exploiting building information from publicly available maps and localization priors," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 85, no. 1, pp. 53–65, 2017.

[9] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact pose slam," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 78–93, Feb 2010.

[10] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[11] M. A. Brubaker, A. Geiger, and R. Urtasun, "Map-based probabilistic visual self-localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 652–665, 2016.

[12] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3d lidar-based online mapping," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9, 2018.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE/RSJ International Conference on Intelligent Robots Systems*, 2012.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[17] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[18] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[19] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[20] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5266–5272.