

# A robust pose graph approach for city scale LiDAR mapping

Sheng Yang, Xiaoling Zhu, Xing Nian, Lu Feng, Xiaozhi Qu and Teng Ma<sup>1</sup>

**Abstract**— This paper presents a method for reconstructing globally consistent 3D High-Definition (HD) maps at city scale. Current approaches for eliminating cumulative drift are mainly based on the pose graph optimization under the constraint of scan-matching factors. The misaligned edges in the graph may have negative impacts on the results. To address this problem and further handle inconsistency caused by multi-task acquisitions in urban environments, we introduce a refined structure of the factor graph considering systematical initialization bias, where the scan-matching factors are twice validated through a novel classifier and a robust optimization strategy. In addition, we incorporate a multi-hypothesis extended Kalman filter (MH-EKF) to remove dynamic objects. Quantitative experimental results demonstrate that the proposed method outperforms state-of-the-art techniques in terms of map quality.

## I. INTRODUCTION

HD maps are the new generation maps for autonomous driving. These maps have extremely high precision at centimeter-level accuracy. Since LiDAR techniques are known for their high precisions, a typical data acquisition system for HD maps is usually a car equipped with 3D LiDAR and Integrated Navigation Systems (INS).

Building maps at city scale has to address many challenges, including: (1) Global pose estimation techniques (e.g. GPS or Differential GPS, DGPS) suffer inaccuracy in harsh urban environments; (2) Data captured from different acquisition tasks may be misaligned due to the uncertainty of pose measurements; (3) Dynamic objects prevalently exist in cluttered regions, interfering the quality and usage of these maps.

Additionally, although cooperative or distributable scanning systems have been introduced for 2D LiDARs [1], UAVs [2], and mostly in the field of computer vision [3], [4], such strategy has not been explicitly proposed for 3D range sensors in city-scale tasks. In fact, global pose measurements through INS exist stochastic initialization biases, which need to be solved for globally coherent mapping.

In general, robustly estimating poses through a graph optimization framework heavily relies on the topological correctness of its factors, otherwise the optimization would occasionally fall into a local optimum. Despite some strategies [5] are presented to automatically filter erroneous factors numerically through coherency, simply relying on the back-end filtering is not sufficient for rejecting a certain amount of false registered point cloud pairs.

In this paper, we present a method capable of automatically constructing globally consistent city-scale HD maps

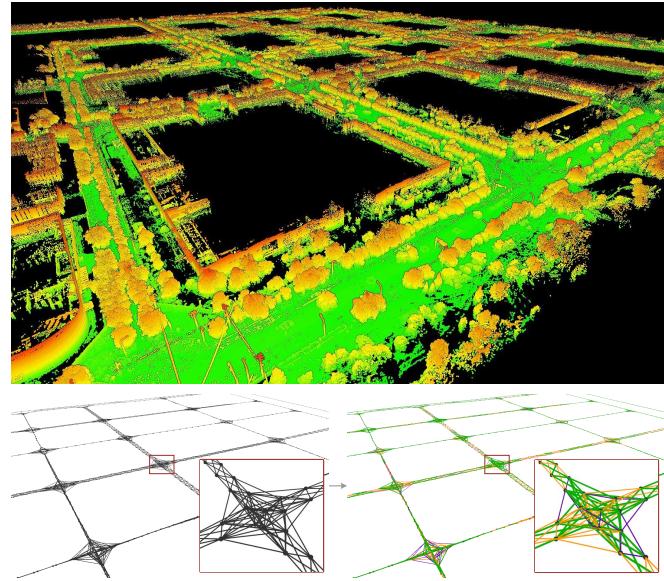


Fig. 1. Our system aligns 3D range data captured through multi-tasks by scanning vehicles (top). Scan matching factors (left) are twice filtered through classification (right-purple) and optimization (right-orange).

from multiple assignments. As shown in Figure 1, we utilize vehicles equipped with a 3D LiDAR and an INS for gathering raw data, which is subsequently fused into a unified dense map according to an enhanced graph structure considering initialization bias in each assignment. Furthermore, scan-matching factors are examined twice before acceptance: Firstly through a classifier considering extracted features from both registration progress and heterogeneous sensors, then in a hierarchical optimization procedure derived from Graham et al. [5]. To filter the dynamic objects, we use a multi-hypothesis extended Kalman filter (MH-EKF) [6] which achieves spatiotemporally consistent tracking to refine our final map. We quantitatively evaluate the effectiveness of the proposed system through various environments (including plain and harsh satellite conditions) and compare with several publicly available systems [7], [8] and a commercial software [9].

## II. RELATED WORK

Reconstructing maps under pose uncertainty is often referred as the Simultaneous Localization and Mapping (SLAM) problem, which can be solved by a factor graph optimization [10], [11], [12]. During the optimization, sensor poses and landmark positions, even calibration parameters, can be simultaneously solved under the constraints of odometry, scan matching, and landmark observations. For earlier

\*This work was done when the authors were with the DiDi Map Business Division.

<sup>1</sup>Corresponding author: matengdamon@didichuxing.com.

SLAM systems, we kindly refer readers to [11], which divides them into two categories: probabilistic approaches [13], [14], [15]; and maximum-a-posteriori (MAP) estimation [7], [16] originated from Lu and Milios [17].

Typically for 3D laser mapping, the most commonly used factor graph is the pose graph representation [18], [19], [12], which simply contains factors between pose variables. Odometry factors are mostly obtained through the extended Kalman filter (EKF) [13] coupling GPS, IMU, and wheel encoders, while scan matching factors [19] are inferred through point cloud registration [20], [21]. Besides, Dubé et al. [8] proposed place recognition factors to connect temporally distant frames with matched segments.

In order to enhance the robustness of pose graphs, landmarks are commonly incorporated in feature-based visual-SLAM algorithms [22], [23], but extracting and utilizing reliable landmarks from point clouds for mapping are challenging. Recently, Yu et al. [24] have pioneered a method to extract feature correspondences through a small set of semantic objects for coarse-to-fine ICP optimization. However, robust detection and differentiated parameterization for a wide variety of road objects are under discovered.

**Robustly optimizing factor graphs.** The robustness of a factor graph framework is closely related to the precision and the topological correctness of the factors. In general, the consideration of precision can be achieved through quantified uncertainty (such as the covariance of measurements), while the correctness can be validated through both back-end and front-end filtering.

In the back-end, many works are introduced to reject erroneous factors: Both SC [25] and DCS [26] propose to add a robust kernel for each factor to remove outliers in the optimization progress. Carloni et al. [27] present another feasible way through finding the maximal subset of coherent factors with relaxation. The state-of-the-art ISCC [5] additionally considers false landmark observations and summarizes the aforementioned approaches.

In addition, false factors can also be filtered in the front-end with evaluation, which is commonly addressed in many visual SLAM systems through descriptor matching or optical flow [28]. However, this is seldom considered for scan matching factors in LiDAR-based SLAM, whose registration would occasionally fall into a local optimum and affect global adjustment.

**Multi-task and cooperative mapping systems.** A city-scale map reconstruction is often scheduled and cooperated by several scanning vehicles through multiple assignments. The errors produced by the initialization bias of scanning devices may cause the inconsistency of global integration. Meanwhile, the increasing scale of graph optimization also brings challenges to the scalability and efficiency of graph solvers.

For online scenario, cooperative mapping has been mostly addressed for other types of sensors [1], [2], [3], [4]. While considering 3D range sensors, Dubé et al. [12] proposed a system to scan buildings simultaneously. Compared with their solution of factor graphs, our system, although not per-

formed in real-time, utilizes double checked scan-matching for spatially coherent submaps with initial prediction given by GPS, achieving a city-scale mapping with multiple assignments.

**Detecting Dynamic Objects.** Road objects in LiDAR data can be categorized into three types as static, movable, and moving [29]. Among them, only the first class should be retained in the final map for localization purposes.

For detecting common road objects in a single frame, traditional methods based on geometric fitting and clustering [30], [31] have recently been reformed into deep learning methods. Projecting the data to top view [32] or front view [33], [6] is considered as an efficient way for predicting movable objects in single frames.

For sequential detection and tracking, Dewan et al. [29] propose to use two sequential scans, where the objectness score of each point is predicted through the first frame and the dynamicity is calculated through RigidFlow. Vaquero et al. [6] implement an MH-EKF for tracking 2D bounding boxes based on frame-wise predictions. We follow previous works [6] in our system for consistently removing dynamic objects.

### III. METHODOLOGY

During the mapping process, the 3D range data is organized into four levels as shown in Figure 2. **Frames**, as the basic level, are firstly rectified and stitched into adaptive submaps with estimated odometry. **Submaps** are regarded as rigid segments in the pose graph optimization phase, under the assumption that the continuous tracking through multi-sensor is able to maintain great accuracy in a short span but may drift over long-range runs [24]. Each **transaction** corresponds to an assignment, which contains multiple submaps sharing a same device initialization process. Considering the spatial relevance of submaps both in one transaction and multiple transactions, our final **map** is optimized under the constraints of both *intra*- and *inter*-transaction registrations.

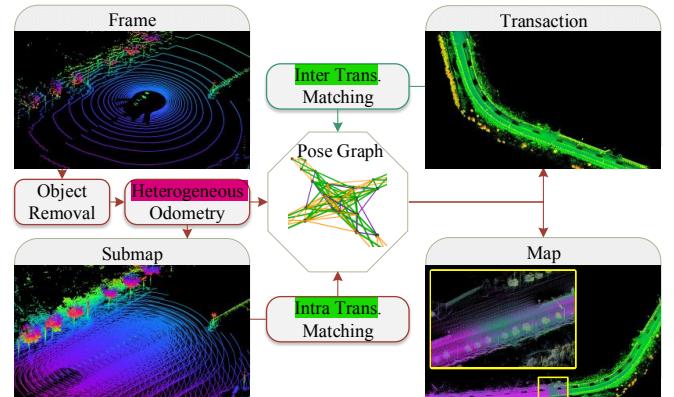


Fig. 2. The data flow of our system, where the final map (from both single transaction and multiple transactions) are optimized under the constraints of odometry and registrations. Red modules are considered for both single and multiple acquisitions, while green only for multi-transaction cases.

In the rest of this section, we first explain the detailed structure of the factor graph (Sec. III-A), then illustrate

the strategy of removing false submap matching factors (Sec. III-B). In addition, we present a practical way to remove dynamic objects (Sec. III-C) regarding spatio-temporal consistency based on MH-EKF [6] and raw trajectory before calculating the fused odometry.

#### A. Pose Graph Formulation.

Our underlying factor graph  $G = (\mathcal{X}, \mathcal{F}, \mathcal{E})$  (Figure 3) consists of variable nodes  $\mathcal{X}$  to be solved under the constraints of factors  $\mathcal{F}$  through correspondences  $\mathcal{E}$ . Two types of variable nodes are introduced: submap poses  $\theta_{ix} \in \mathcal{X}$  (circles), where  $\theta_{ix}$  is the  $x$ -th submap in the  $i$ -th transaction, and initialization bias  $\phi_i \in \mathcal{X}$  (hexagons), where  $\phi_i$  is augmented for each transaction in order to balance the bias caused by different hardware configurations, e.g., the deployment deviation of the GNSS system with dual GPS receivers and errors from sensor calibration. We incorporate two types of factors as odometry factors  $f_{ix}^{odom} \in \mathcal{F}$  (blue) and submap matching factors  $f_{ix,jy}^{reg} \in \mathcal{F}$ . Specifically,  $f^{reg}$  contains both *intra*-transaction factors (red) and *inter*-transaction factors (brown), which sustainably assure global consistency during the integration of submaps. All variables  $\mathcal{X} \subset SE(3)$ , and factors are given in full 6 Degree-of-Freedom ( $n_d = 6$ ).

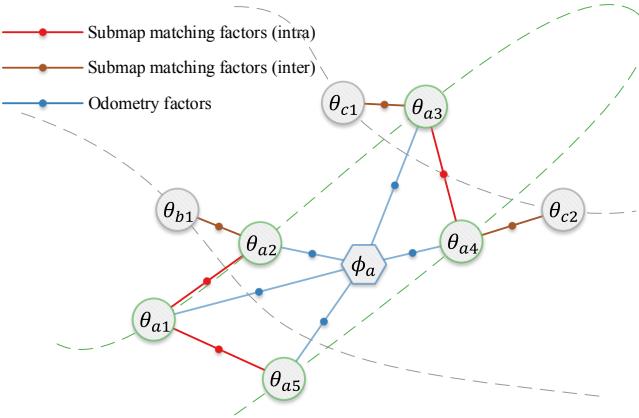


Fig. 3. An exemplary factor graph including three transactions (a-c) in our back-end optimization. Only those factors related to the  $\theta$  (green circles) are shown.

**Adaptive submap construction.** Since the acquired LiDAR data from one frame is sparse, constructing pose variables per frame is neither sufficient for robust pair-wise scan matching, nor practical for solving graphs containing millions of nodes. Hence, hierarchical submap constructions are presented for back-end optimization.

Consecutive frames are firstly rectified and tracked before they are stitched together. For rectification, a 6-DOF spline-slerp interpolation [34] is performed on trajectories that are predicted by IMU [7]. Then in the registration stage, we use the unscented Kalman filter (UKF) [14] to integrate measurements from LiDAR, GPS and IMU. Specially for LiDAR odometry, we match the current frame to a fixed buffer of recently tracked frames (15 in our implementation) through NDT registration [21]. As a consequence, we get the

rectified frame  $F_k$  containing  $w_k$  points with its odometry covariance  $\Omega_{F_k}^{odom}$  and relative transformation  $\mathbf{T}_{k,k-1}^F \in SE(3)$  that registers  $F_k$  to its previous frame  $F_{k-1}$ .

While previous methods use either fixed frame interval or cumulative translation/rotation threshold for partitioning [24], we utilize the uncertainty of localization reflected in  $\Omega_F^{odom}$  to achieve an adaptive strategy that divides the areas with high uncertainty into smaller sections, so that a better non-rigid deformation can be obtained through the subsequent optimization. Specifically, for a constructing submap  $S_x$ , we continuously update its covariance of odometry  $\Omega_x^{odom}$  as:

$$\Omega_x^{odom} = \frac{1}{\sum_k w_k} \cdot \sum_k w_k \cdot \mathbf{T}_{k,0}^F \cdot \Omega_{F_k}^{odom} \cdot \mathbf{T}_{k,0}^F \top, \quad (1)$$

where  $\mathbf{T}_{0,0} = \mathbf{I}$  here for a unified notation. The trace of  $\Omega_x^{odom}$  is calculated and divided into the translation part  $tr_t(\Omega_x^{odom})$  and the rotation part  $tr_r(\Omega_x^{odom})$ . And the criterion for establishing a new submap is defined as:

$$\Gamma_{\gamma_x} > \sigma_\gamma \cdot \exp\left(-\frac{tr_t(\Omega_x^{odom})}{2\sigma_\gamma^2}\right), \gamma \in \{r, t\}, \quad (2)$$

where  $\Gamma_{\gamma_x}$  stands for the cumulative translation  $\Gamma_{t_x}$  or rotation  $\Gamma_{r_x}$  of  $S_x$ , we use  $\sigma_t = 30m, \sigma_r = 30.0^\circ, \sigma_t^e = 0.02m$ , and  $\sigma_r^e = 0.5^\circ$  as parameters in our implementation. For each constructed submap  $S_x$ , we regard its first frame as the reference of submap pose  $\theta_{ix}$ , and the stitched submap is downsampled with the grid-size at 0.05m for registration efficiency.

**Odometry factors.** The covariance of odometry  $\Omega^{odom}$  reflects the uncertainty of global pose prediction. Intuitively, incorporating such information is beneficial for controlling the scale of optimization. Since global prediction of UKF is coupled with one intialization bias, our odometry factors are defined with the additional variable node  $\phi_i$  for each transaction  $R_i$ , as:

$$f_{ix}^{odom} \propto \exp\left(-\frac{1}{2} \|(\theta_{ix} \ominus \mathbf{T}_{ix}^S) \ominus \phi_i\|_{\Omega_{ix}^{odom}}^2\right), \quad (3)$$

where  $\mathbf{T}_{ix}^S$  is the predicted pose of  $S_{ix}$  through odometry predicted by UKF [14] through combining LiDAR odometry, IMU, and GPS.  $\ominus$  denotes the motion decomposition operator [35].  $\|\mathbf{x}\|_{\Omega}^2 := \mathbf{x}^\top \Omega^{-1} \mathbf{x}$  is the squared Mahalanobis distance with the covariance matrix  $\Omega$ .

**Submap-matching factors.** The global consistency of the final map is ensured through registering spatially coherent submaps  $S_{ix}, S_{jy}$ , which generates constraints as submap matching factors  $f_{ix,jy}^{reg}$  that are defined as:

$$f_{ix,jy}^{reg} \propto \exp\left(-\frac{1}{2} \|(\theta_{ix} \ominus \widehat{\mathbf{T}}_{ix,jy}^S) \ominus \theta_{jy}\|_{\Omega_{ix,jy}^{reg}}^2\right), \quad (4)$$

where  $\widehat{\mathbf{T}}_{ix,jy}^S$  is the estimated transformation from one executable registration algorithm that registers  $S_{ix}$  to  $S_{jy}$  with an estimated covariance  $\Omega_{ix,jy}^{reg}$ . To get robust registration results, the system attempts multiple registration algorithms (e.g. NDT [21] and coarse-to-fine ICP [24]) for each pair.

Their results are subsequently considered in the pose evaluation module (Sec. III-B) for rejecting misaligned pairs that fall into local convergence.

### B. Validating Graph Factors.

The quality of the estimated poses heavily relies on the correctness of the factor graph structure. Unfortunately, false factors occasionally appear in the scan matching progress due to the erroneous registrations. Therefore, we present our two-stage strategy for validating those registration attempts.

**Classifying scan matching factors** With the help of heterogeneous odometry (Sec. III-A), it is practical to evaluate those registration attempts through comparisons among information from multi-sensors. Hence, we train a classifier to delete the tracking factors in the front-end, where features from both pose prediction and submap registration are considered. Specifically, for a candidate pair  $S_{ix}$  and  $S_{jy}$ , we denote the transformation estimated from the INS as  $\tilde{T}_{ix,jy}^S$ , and extract the following features listed in Table I, as they reflect the confidence, variance, and effect of the registration attempt.

Notes	Description	Dimension
$\delta_{ot}$	the translation of $\tilde{T}_{ix,jy}^S$ .	1
$\delta_{or}$	the rotation of $\tilde{T}_{ix,jy}^S$ .	1
$\delta_{gt}$	the translation of $\hat{T}_{ix,jy}^S$ .	1
$\delta_{qr}$	the rotation of $\hat{T}_{ix,jy}^S$ .	1
$\delta_{xt}, \delta_{xr}$	$tr_t(\Omega_{ix}^{odom})$ and $tr_r(\Omega_{ix}^{odom})$	2
$\delta_{yt}, \delta_{yr}$	$tr_t(\Omega_{jy}^{odom})$ and $tr_r(\Omega_{jy}^{odom})$	2
$\delta_{rt}, \delta_{rr}$	$tr_t(\Omega_{ix,jy}^g)$ and $tr_r(\Omega_{ix,jy}^g)$	2
$\delta_z$	the time span between $S_{ix}$ and $S_{jy}$ .	1
$\delta_m$	the RMSE of matched points.	1
$\delta_n$	the percentage of matched points.	1

TABLE I

EXTRACTED RAW FEATURES FOR OUR CLASSIFIER.

Specifically for  $\delta_n$ , we use 0.05m as the distance threshold for counting matched pairs. We next expand these 13-dimensional vectors into 65-dimension using a  $\chi^2$  kernel map [36], and use a random forest classifier [37] to separate correct and wrong registrations. Those classification parameters are acquired by training on a manually labeled dataset including 7626 registration pairs. We split 80% of them for training and the rest for testing, and get an accuracy of 91% (see Sec. IV-D for detailed results categorized through different environments). Notice that those false-positive factors are subsequently validated in the optimization phase.

Representative comparisons of incorporating such front-end strategy are shown in Figure 4, where temporally distant frames are colorized in different colors for visualizing the consistency of the reconstructed map. As a consequence, poles and signs gain better coherence in this high-way scanning scenario. Quantitative effects of such classifier for the quality of final maps are shown in Table II.

**Optimization strategy.** Despite the aforementioned front-end validation process is able to filter most of erroneous loops, the constructed pose graph in one transaction may still contain false-positive factors. Hence, we choose to derive

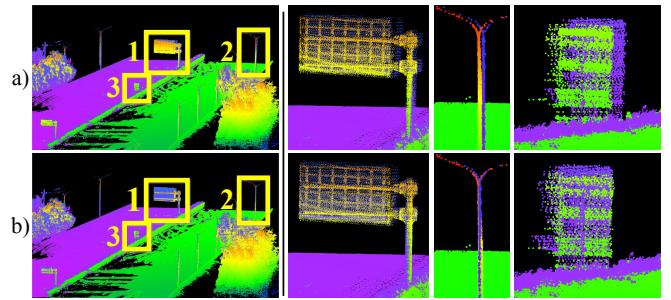


Fig. 4. Detailed view of maps constructed with/without front-end factor filtering. a): without filtering, where false edges result to unexpected optimization. b): with filtering.

from a robust approach [5] rather than traditional graph optimization frameworks [38] for autonomously muting incompatible factors. Our optimization problem is defined as:

$$\begin{aligned} & \max_{\mathcal{X}, \mathcal{S}} \sum_{\mathcal{F}^{reg}} \beta_{ix,jy} \\ \text{s.t. } & \sum_{\mathcal{F}^{odom}} \mathbf{E}(\mathbf{f}_{ix}^{odom}) + \sum_{\mathcal{F}^{reg}} \beta_{ix,jy} \mathbf{E}(\mathbf{f}_{ix,jy}^{reg}) \leq \chi^2(\sigma_p, n_d) \\ & \beta_{ix,jy} \mathbf{E}(\mathbf{f}_{ix,jy}^{reg}) \leq \chi^2(\sigma_p, n_d), \forall \mathbf{f}_{ix,jy}^{reg} \in \mathcal{F}^{reg}, \end{aligned} \quad (5)$$

where  $\beta_{ix,jy} \in \Theta$  are additional binary variables for adaptively picking submap matching factors,  $\mathbf{E}(\cdot) = -\log(\cdot)$  is a log-map for residuals in  $SE(3)$ , and  $\chi^2(\sigma_p, n_d)$  is the inverse  $\chi^2$  Cumulative Distribution Function (CDF), and  $\sigma_p = 0.95$  for consistency verification.

The solution of Eq. 5 is obtained by two steps. The first step is *intra*-transaction filtering, in which the factors are evaluated within the transaction, and the results are staged. In the second step, the optimization algorithm only considers the validity of those factors between transactions. Such strategy effectively controls the size of optimization, since the *inter*-transaction pairs are generated by the overlapped regions, which in practice can be reduced by assignment scheduling.

### C. Spatio-temporal removal of dynamic objects

In order to filter movable objects, we apply the multi-hypothesis extended Kalman filter (MH-EKF) [6] to obtain consistent tracking of road objects. We use the MV3D [32] for detecting dynamic objects in every single frame, and further extract the inlier points for calculating the refined transformation of those candidates.

If the object is successfully detected or a hypothesis is above a threshold, the object will be removed and those laser beams corresponding to removed points are extended along the laser direction to a fitted ground by RANSAC [39] for hole filling. Incorporating spatiotemporal hints to achieve consistent filtering is crucial, especially when objects are far from the vehicle, presenting a very sparse segment that is hard to be detected (Figure 5). In practice, such filtering is performed twice (forward and backward) for a better recall.

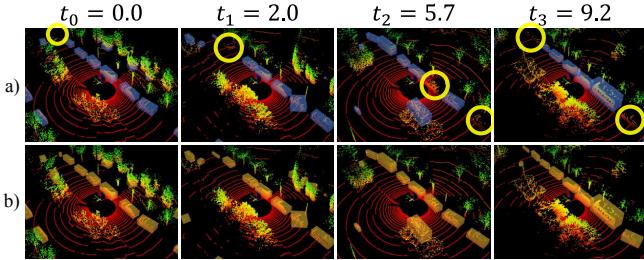


Fig. 5. Removal of dynamic objects. The figure contains four representative frames illustrating a tracking progress for several vehicles from appearance to disappearance. a): single frame detection algorithm, where missing vehicles are circled in yellow. b): detection algorithm enhanced by MH-EKF.

#### IV. EXPERIMENTS AND EVALUATIONS

In this section, we first illustrate the detailed equipment of our scanning system and the standardized data acquisition process (Sec. IV-A), then perform qualitative impression and quantitative evaluation on the mapping results of single transaction (Sec. IV-B) and multiple transactions (Sec. IV-C) in comparison to some publically available systems. Additionally, the detailed statistics of our classifier, dynamic filter, and running time are presented in Sec. IV-D, Sec. IV-E, and Sec. IV-F, respectively.

##### A. Brief Introduction of Our Scanning System.

The influential KITTI odometry benchmark [40] has aroused the production of many LiDAR odometry methods [7], [16], which regards the output of INS as ground truth for trajectory comparisons. In contrast, our approach concentrates on integrating heterogeneous sensors to further enhance the accuracy of mapping, especially for those in cluttered environments with weak GNSS signals. Hence, we assembled several scanning vehicles and create a dataset containing objects with known size for evaluation.

Our scanning system consists of an INS including pre-calibrated 3-axis gyrometers, 3-axis accelerometers, and a GPS enhanced by RTK with an additional fixed base station. We use Velodyne HDL-32E as our scanner and implement an off-line calibration between the scanner and the INS based on Levinson et al. [41]. Vehicles are driven under 40km/h on urban roads and 60km/h on highways, lasting several hours in Beijing, China. We manually placed and measured some representative objects (e.g. road signs and landmarks, shown on the bottom of Figure 6) in the covered area for quantitative evaluation, on both common roads (a1-a4) and areas with weak GNSS signals (b1-b4), where each pair (e.g. a1-a2) has same measured objects for multi-transaction evaluations.

##### B. Single transaction cases.

We choose 3 publically available systems to compare the quality of mapping: LOAM [7], SegMatch [8], and a commercial system Inertial Explorer (IE) [9]. LOAM takes both range data and IMU as input for LiDAR odometry estimation, but this system does not include back-end optimization strategy for handling loops. In order to launch

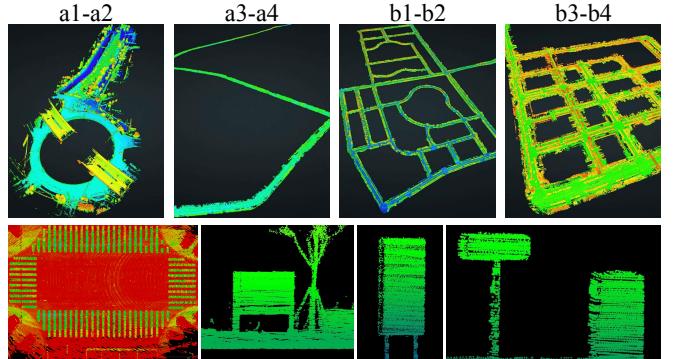


Fig. 6. Scenes and measured signs used in our evaluation. Top row: overview of scenes, including viaduct, highway, and two cluttered urban scenes. Bottom row: representative signs for quality measurements.

**SegMatch**, both RTK poses and raw LiDAR frames are sent as input for construction. IE is a common way in the industry of refining RTK trajectories. In addition, we implement a version without the front-end scan-matching classifier (Ours-NC) as a baseline for demonstrating the effectiveness of our presented strategy. All 8 scans are sent to each system, and their output trajectories are used for frame stitching.

Reconstruction quality of these transactions are visualized in Figure 7 and measured in Table II. Specifically in Figure 7, we show the reconstructed map in different levels colorized by height. For detailed road objects such as signs and fences (pairs on the right), we use two colors (green and orange) to draw temporally distant frames, which is typically produced by passing through such same place at different time. The results of such objects illustrate that our system achieves better globally consistent mapping compared with IE. In the quantitative perspective, the measured objects are found and measured 3 times for averaging in the final point cloud, and the RMSE of each scenario is summarized to Table II. Specifically for LOAM, since it does not contain back-end optimization, we only measure such targets that are acquired by one-pass scanning to avoid loop closure. As a consequence, LOAM is able to produce precise trajectories but requires sufficient point cloud features, SegMatch occasionally failed in some scenes due to the missing or wrong place recognition factors, especially those with insufficient or dynamic segments (the fails stand for unrecognizable places). In summary, our system outperforms those above systems in maintaining high accuracy during long-time scanning, where the front-end classifier has positive impact on the final quality.

##### C. Multiple transaction cases.

For evaluating our ability on integrating multiple transactions, we compare the proposed pose graph structure to the original graph structure for mapping, where initialization biases  $\phi$  are not included, and those odometry factors  $f^{odom}$  are considered as unary factors for its corresponding variables. As shown in Table III, the strategy of replacing unary priors by relative poses are beneficial for stitching multiple assignments based on different hardware configurations.

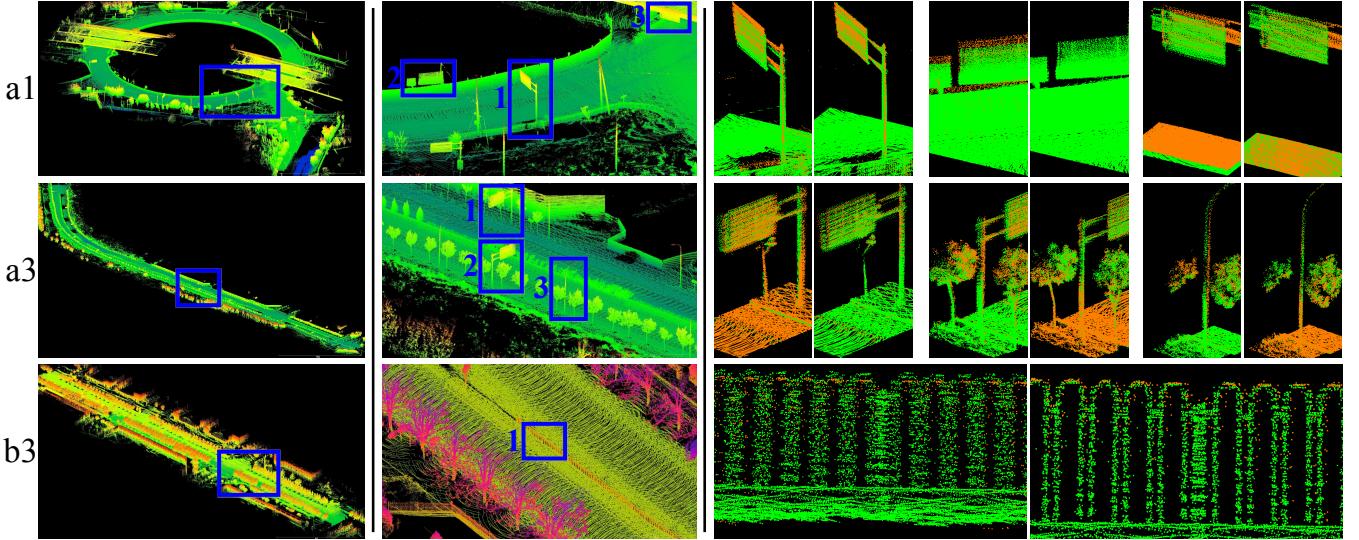


Fig. 7. Visualized results of mapping systems. Left: reconstructed roads colorized by height. Middle-left: detailed views of roads colorized by height. Right: Comparison of reconstructed objects, where each pair contains IE [9] (left) and ours (right), and temporally discontinuous frames are colorized by different colors.

ID	Signs	LOAM(cm)	SegMatch(cm)	IE(cm)	Ours-NC(cm)	Ours(cm)
a1	7	159.6	Failed	12.5	6.9	3.9
a2	7	172.5	Failed	17.2	5.7	5.0
a3	12	83.9	Failed	7.6	5.1	4.3
a4	12	67.3	Failed	7.2	6.4	5.4
b1	10	51.7	108.6	10.0	4.3	3.7
b2	10	37.4	118.6	14.3	5.7	4.0
b3	5	60.0	52.4	30.4	8.3	7.4
b4	5	50.4	76.3	16.7	8.7	8.4
Avg.	-	85.4	89.0	14.5	6.4	5.3

TABLE II

THE RMSE OF MEASURED SIGNS BASED ON LOAM [7],  
SEGMATCH [8], IE [9] AND OURS.

ID	Signs	Unary(cm)	Ours(cm)
a1+a2	7	10.3	5.8
a3+a4	12	7.5	6.9
b1+b2	10	9.5	5.3
b3+b4	5	14.5	10.6
Avg.	-	10.5	7.2

TABLE III

THE RMSE OF MEASURED SIGNS BY GRAPH STRUCTURE WITHOUT  
BIAS VARIABLES AND OURS.

#### D. Result of random forest classification.

The statistical results of random forest classifier on the testset in Sec. III-B is listed in Table IV, where the labeled registration pairs are from both common roads(a) and areas with weak GNSS signals(b). In summary, we acquired 91.3% precision and are able to filter most of the false-positive candidates, which ensures a cleaner set of loop candidates for the back-end optimization to converge better results. Despite that our recall is relatively lower than the precision, the system is able to accept sufficient scan matching factors for loop closure.

ID	TP	TN	FP	FN	Pre.	Rec.	Acc.	F1.
a	242	9	21	117	93.6	84.1	92.3	92.2
b	726	22	74	315	93.9	79.7	91.3	91.1
Total	954	45	88	439	91.3	81.6	91.0	90.8

TABLE IV

THE CONFUSION MATRIX, PRECISION, RECALL, ACCURACY, AND F1 SCORE OF TESTSET FOR LOOP VALIDATION, THE LAST FOUR ROWS ARE LISTED IN PERCENTAGE.

#### E. Accuracy of dynamic objects filtering.

Table V demonstrates the effectiveness of our proposed spatio-temporal removal strategy based on MH-EKF [6]. To perform such an evaluation, the number of successfully and erroneously detected vehicles are counted manually. The listed results clearly show the effectiveness of our method, as the tracking strategy significantly improves the recall, which is crucial for acquiring a cleaner map.

Method	TP	FP	FN	Pre.	Rec.	F1.
MV3D	668	276	192	70.7	77.7	74.1
MV3D + Tracking	776	284	110	<b>73.2</b>	<b>87.5</b>	<b>79.8</b>

TABLE V

THE ENHANCEMENT FROM APPLYING A SPATIO-TEMPORAL TRACKING FOR CONSISTENT FILTERING.

#### F. Running time.

We listed the total length and the scale of our constructed graph for each transaction in Table VI, and also show the time spent for scanning, graph optimization and total processing. Our system is executed on a server with 48 cores and 128GB memories, where plenty of registrations are parallelly performed but still occupies a large proportion (nearly 70%) of calculations. In contrast, the optimization

progress only occupies a small proportion of the total running time. Eventually, our total processing time is acceptable for processing daily acquisitions.

ID	len.(km)	$\ \mathcal{S}\ $	$\ \mathcal{X}^{reg}\ $	$\ \mathcal{X}_f^{reg}\ $	Scan(min)	Opt.(sec)	Tot.(min)
a1	2	97	161	68	8	0.09	38
a2	3.5	168	1009	586	9	1.07	112
a3	34	1501	2177	998	54	1.04	350
a4	34	1681	3004	1566	55	3.60	323
b1	30	2662	3361	2195	67	4.74	735
b2	30	1285	3403	2407	76	2.02	748
b3	22	1563	3327	2711	85	1.90	574
b4	30	2312	6403	4953	120	2.02	704

TABLE VI

THE ATTRIBUTES AND TIME COSTS OF EACH TRANSACTION. FROM LEFT TO RIGHT: TOTAL LENGTH, NUMBER OF SUBMAPS/RAW/FILTERED SCAN MATCHING FACTORS, TIME OF SCANNING/OPTIMIZATION/TOTAL.

## V. CONCLUSION

We have presented a framework for acquiring 3D high-definition maps. A refined pose graph structure is designed for optimizing multiple transactions robustly, and a reliable classifier is incorporated for filtering erroneous scan matching factors. In the future, our system can be augmented with **landmark extraction modules**, and further supports modern autonomous driving applications.

## ACKNOWLEDGEMENTS

We would like to thank the reviewers for their constructive comments. We would also like to thank Xian Zhang and Baoli Li for their support throughout the project, Zhengqiang Yan for vehicle preparation, Ye Wang and Zugang Chen for 3D visualization platform.

## REFERENCES

- [1] S. Saeedi, M. Trentini, M. Seto, and H. Li, “Multiple-robot simultaneous localization and mapping: A review,” *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.
- [2] P. Schmuck, “Multi-uav collaborative monocular slam,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3863–3870.
- [3] K. Konolige and J. Bowman, “Towards lifelong visual maps,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1156–1163.
- [4] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard, “Real-time 6-dof multi-session visual slam over large-scale environments,” *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1144–1158, 2013.
- [5] M. C. Graham, J. P. How, and D. E. Gustafson, “Robust incremental slam with consistency-checking,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 117–124.
- [6] V. Vaquero, I. del Pino, F. Moreno-Noguer, J. Solà, A. Sanfeliu, and J. Andrade-Cetto, “Deconvolutional networks for point-cloud vehicle detection and tracking in driving scenarios,” in *European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–7.
- [7] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and Systems*, vol. 2, 2014.
- [8] 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)*, 2017, pp. 5266–5272.
- [9] “Inertial explorer,” <https://www.novatel.com/products/software/inertial-explorer/>.
- [10] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based slam,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [11] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. L. Neira, I. D. 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, 2016.
- [12] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, “An online multi-robot slam system for 3d lidars,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1004–1011.
- [13] S. J. Julier and J. K. Uhlmann, “New extension of the kalman filter to nonlinear systems,” in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, 1997, pp. 182–194.
- [14] E. A. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.
- [15] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, “The unscented particle filter,” in *Advances in neural information processing systems*, 2001, pp. 584–590.
- [16] J. Zhang and S. Singh, “Low-drift and real-time lidar odometry and mapping,” *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.
- [17] F. Lu and E. Milios, “Globally consistent range scan alignment for environment mapping,” *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.
- [18] G. H. Lee, F. Fraundorfer, and M. Pollefeys, “Robust pose-graph loop-closures with expectation-maximization,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 556–563.
- [19] E. Mendes, P. Koch, and S. Lacroix, “Icp-based pose-graph slam,” in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 195–200.
- [20] 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, 1992, pp. 586–607.
- [21] M. Magnusson, “The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection,” Ph.D. dissertation, Örebro universitet, 2009.
- [22] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [23] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, “Pl-slam: real-time monocular visual slam with points and lines,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4503–4508.
- [24] F. Yu, J. Xiao, and T. Funkhouser, “Semantic alignment of lidar data at city scale,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1722–1731.
- [25] N. Sünderhauf and P. Protzel, “Towards a robust back-end for pose graph slam,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1254–1261.
- [26] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, “Robust map optimization using dynamic covariance scaling,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 62–69.
- [27] L. Carlone, A. Censi, and F. Dellaert, “Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2667–2674.
- [28] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part i: The first 30 years and fundamentals,” *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.
- [29] A. Dewan, G. L. Oliveira, and W. Burgard, “Deep semantic classification for 3d lidar data,” *arXiv preprint arXiv:1706.08355*, 2017.
- [30] A. Petrovskaya and S. Thrun, “Model based vehicle detection and tracking for autonomous urban driving,” *Autonomous Robots*, vol. 26, no. 2–3, pp. 123–139, 2009.
- [31] A. Teichman, J. Levinson, and S. Thrun, “Towards 3d object recognition via classification of arbitrary object tracks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4034–4041.
- [32] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in *IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534.
- [33] B. Li, T. Zhang, and T. Xia, “Vehicle detection from 3d lidar using fully convolutional network,” *arXiv preprint arXiv:1608.07916*, 2016.
  - [34] K. Shoemake, “Animating rotation with quaternion curves,” in *ACM SIGGRAPH Computer Graphics*, vol. 19, no. 3, 1985, pp. 245–254.
  - [35] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Autonomous Robot Vehicles*, 1990, pp. 167–193.
  - [36] A. Vedaldi and A. Zisserman, “Efficient additive kernels via explicit feature maps,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 3, pp. 480–492, 2012.
  - [37] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
  - [38] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A general framework for graph optimization,” in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.
  - [39] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” in *Readings in Computer Vision*, 1987, pp. 726–740.
  - [40] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.
  - [41] J. Levinson and S. Thrun, “Unsupervised calibration for multi-beam lasers,” in *Experimental Robotics*, 2014, pp. 179–193.