TM^{3}Loc: Tightly-coupled Monocular Map Matching for High Precision Vehicle Localization

Tuopu Wen[1], Kun Jiang[1], Benny Wijaya[1], Hangyu Li[1], Mengmeng Yang[1] and Diange Yang[1]

***Abstract*—Vision-based map-matching with HD map for high precision vehicle localization has gained great attention for its low-cost and ease of deployment. However, its localization performance is still unsatisfactory in accuracy and robustness in numerous real applications due to the sparsity and noise of the perceived HD map landmarks. This article proposes the *tightly-coupled monocular map-matching* localization algorithm (TM^{3}Loc) for monocular-based vehicle localization. TM^{3}Loc introduces semantic chamfer matching (SCM) to model monocular map-matching problem and combines visual features with SCM in a tightly-coupled manner. By applying the sliding window-based optimization technique, the historical visual features and HD map constraints are also introduced, such that the vehicle poses are estimated with an abundance of visual features and multi-frame HD map landmark features, rather than with single-frame HD map observations in previous works [1][2][3]. Experiments are conducted on large scale dataset of 15 km long in total. The results show that TM^{3}Loc is able to achieve high precision localization performance using a low-cost monocular camera, largely exceeding the performance of the previous state-of-the-art methods, thereby promoting the development of autonomous driving.**

***Index Terms*—Vehicle localization, HD map, map-matching, autonomous driving, intelligent vehicle.**

I. INTRODUCTION

HIGH-precision and robust self-vehicle localization serves as a prerequisite for navigation, decision making, and control of autonomous vehicles. Existing high-precision localization techniques based on differential RTK (D-RTK), such as GNSS, can theoretically achieve centimeter-level localization precision. However, in real applications such as urban scenarios, a large localization deviation is often seen when the surrounding buildings and trees block the GNSS signal, making using the GNSS sensor alone becomes insufficient [4][5]. As an alternative localization technique, map-based localization has gained a lot of popularity for its role to serve as a complementary localization method.

In recent years, since its first inception in late 2010, High-Definition Maps (HD Maps) has gained tremendous popularity in the intelligent vehicle industry, mainly because it carries road elements with a much higher level of details compared to the traditional navigation maps [6]. Several map building companies have engaged in constructing their HD map databases on a large scale with a series of production and publication standards, such as Navigation Data Standard (NDS) and Local Dynamic Map (LDM). The mainstream approach to building HD Maps is through Mobile Mapping System (MMS) equipped with high precision sensors including LiDAR, RTK and IMU at centimeter-level precision. The resultant map then consists of fine localization features that can support intelligent vehicles' positioning and trajectory planning.

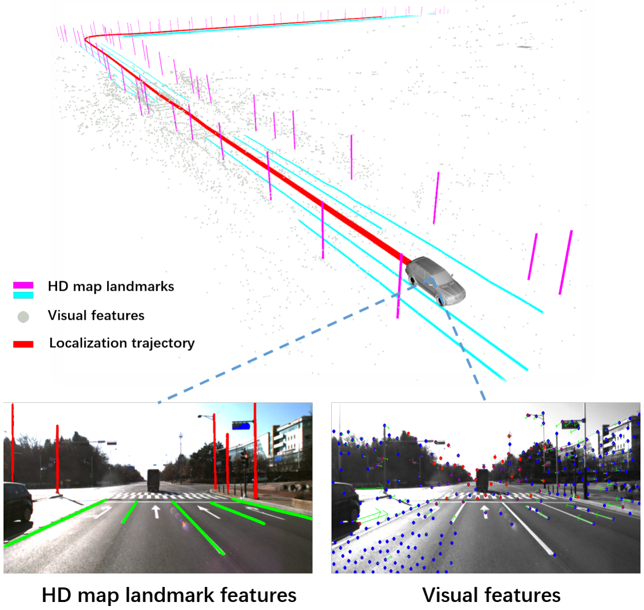


Fig. 1. The proposed TM^{3}Loc algorithm that localizes a vehicle with monocular camera in a pre-built HD map. The algorithm fuses the HD map landmark features and the visual features to estimate the vehicle poses in a tightly-coupled manner, which improves the system localization accuracy and robustness.

The localization feature in the HD map can be divided into a) dense point cloud feature and b) sparse landmark feature. The point cloud feature consists of the original point cloud scanned by 3D LiDAR sensor [7], which maintains the raw geometric information of the point cloud. State-of-the-art map-based localization methods use point cloud HD map to accurately estimate vehicle pose within a maximum error of 0.2 m [8][9][10]. However, equipping intelligent vehicles (IVs) with LiDAR sensors will significantly and undesirably increase the overall sensor cost and the subsequent vehicle production cost. Furthermore, the huge data size of the point cloud map increases the difficulty in implementing the HD

[1] Tuopu Wen, Kun Jiang, Benny Wijaya, Hangyu Li, Mengmeng Yang and Diange Yang are with the School of Vehicle and Mobility, Tsinghua University, Beijing, 100084, China. This work was supported in part by National Natural Science Foundation of China (U1864203 and 52102464), and in part by China Postdoctoral Science Foundation (2019M660622), and in part by Tsinghua University-Didi Joint Research Center for Future Mobility.

Corresponding author: Diange Yang, email: ydg@mail.tsinghua.edu.cn, Mengmeng Yang, email: yangmm_qh@tsinghua.edu.cn

map on IVs. Compared to the landmarks in the point cloud map, the lightweight HD map landmark features are more flexible and easy to use. The HD map landmarks consist of static vectorized semantic landmarks (such as lane lines, poles, and traffic signs), which are much more lightweight than those in the original point cloud map. Matching the HD map landmark features with the images from the low-cost camera is an engineering- and commercial-friendly solution for mass-produced vehicles. As a result, researchers have been investing great efforts in matching these HD map landmarks with features in the camera images. The basic idea behind this approach is to detect semantic landmarks of HD map in the camera image. The vehicle pose can then be obtained by aligning the detected landmarks in the image with their corresponding 3D landmarks in the HD map.

Pink et al. use aerial images to extract lane markings and their respective positions to build a lane-level map, then match the lane marking features in the image with this pre-built map using the simple ICP algorithm [11]. The subsequent works improve the map-matching localization by fusing other sensors such as GNSS and IMU in an Extended Kalman Filter (EKF) framework. Tao et al. proposed to use the EKF fusion framework to integrate lost GPS, dead reckoning, and map-matching features from vision lane markings [12]. The lane markings detected by the video camera system provide the lateral distance and heading angle information of a vehicle. Cai et al. also use the map-matching observation as the additional lateral distance observation in the filter-based fusion frameworks [13]. These works were able to show that the vision-based map matching approach for localization application using pre-built lane level maps is viable and promising.

To make full use of the raw landmark features, research works have deployed additional sensors like LiDAR or replaced the monocular camera with the multi-camera system to restore the 3D information of the environment perception [14][15]. In [16][17][18], the stereo camera systems are utilized to restore the lane markings in 3D space. The stereo images can help calculate the road surface equation by v-disparity [19]; then, the detected lane markings are projected onto that 3D road surface and aligned with the HD map Ma et al. obtain the depth of the 2D detection of lane boundaries and traffic signs through the LiDAR sensor and perform map-matching on the bird's-eye view space constructed by LiDAR scans [20]. Although stereo camera systems or LiDAR can help obtain the 3D information of the detected landmarks, the additional cost of LiDAR and the reliability issue of the stereo camera calibration [21] are the shortcomings of these methods.

In monocular camera settings, due to the depth ambiguity of the monocular camera, projecting the detected landmarks into the 3D space is proven challenging. The common strategy is to assume that the road is flat and transform the detected lane marking features into the bird's-eye view using inverse projection mapping (IPM) [22][23][24]. This method might suffer significant distortion due to the vibration on pitch angle of camera with respect to the ground plane [25]. Some research works investigate the feasibility of performing map-matching on the 2D image space [2][26]. The main challenge is that due to the perspective effect of the camera, the detected 2D landmarks, especially the lane boundaries, can vary drastically in shapes, leading to increased complexity in the parameterization process. In [2][26], the detected lane boundaries are simplified to be straight lines and a point-to-line cost model between HD map landmark detection and projection in the image plane is utilized to solve for the 6-DoF camera pose. Some researchers introduces a neural network to directly predict the 3D layouts of the lane marking from a single image frame [27][28]. Liao et al. only utilize the pole-like landmark features in the image with the assumption that they remain as straight lines in the image plane [29]. With the collection of the simplified model of landmarks, these works can realize monocular map-matching and achieve reasonable performances for vehicle localization.

Firstly, the assumptions of straight lines or flat plane of lane boundaries are not always held true in real applications. As a result, the simplified model of lane boundaries essentially turns into the source of the systematic localization error. The second factor, which is also the vital problem of these vision-based map-matching techniques, is the significant lack of localization constraints due to the sparsity of HD map landmarks in realistic scenarios. Compared to the hundreds of thousands of dense geometric features of point cloud map, the localization clues provided by HD map landmarks are far less. The detected landmarks cannot provide complete localization constraints to solve for the vehicle poses in some cases entirely. Additionally, the detection of landmarks in the image is affected by the detection algorithm and suffer more severe noise interference during image acquisition. The sparsity of HD map landmark observations fundamentally restricted the number of available constraints to reduce the impact of the observation noise and false association, thus resulting in the deterioration of localization accuracy [30]. Recent works also tried to utilize the more general semantic deep learning-based feature instead of HD map landmarks, such as PoseNet [31], DeLS-3D [32], HFNet [33] and DA4AD [34]. These methods demonstrated the feasibility of localizing using deep learning. However, most of these methods can only reach a meter level of localization accuracy for outdoor localization setup, and the result can not be guaranteed due to the generalization issue.

To solve these problems, we propose TM^{3}Loc, a novel localization algorithm for improving the performance of the monocular map-matching process. First, to deal with the map-matching problem in the image plane, we adopt the semantic chamfer matching (SCM) algorithm, inspired by the traditional image alignment method, *chamfer matching*. In chamfer matching, the distance transform is used to model point-to-edge alignment. Lu *et al.* [1] first applied this method in single frame map-matching of lane boundaries, and Pauls *et al.* expanded on it to align with semantic segmentation results such as poles and traffic signs [3]. Similar techniques have also been used in online LiDAR-camera calibration [35][36]. In TM^{3}Loc, we used SCM as a general cost model to match different landmarks. However, instead of optimizing the cost model using global search [35] or automatic derivatives [3], we derived an analytical derivation of chamfer matching cost with respect to the 6-DoF pose on $\mathfrak{se}(3)$ to ensure efficient optimization. Besides, to tackle the inevitable noise of HD map landmark feature detection, an effective outlier rejection strategy is also proposed. With the improved SCM implementation, the monocular map-matching problem can be efficiently and robustly solved in a unified form with various landmark shapes, thereby avoiding the inaccuracy brought by any prior assumption of the target shapes.

Secondly, to deal with the sparsity of HD map landmark features, we introduce the visual features from monocular images in the map-matching process. With the development of visual odometry [37][38] and visual SLAM [39][40][41][42], the visual features are shown to be able to serve as accurate localization constraints for relative pose estimation. As a result, they can be considered as complementary localization information in conjunction with the HD map landmark features. Previous works related to this strategy [43][44] combined these two pieces of information in a loosely coupled manner - i.e., the map-matching-based and the visual feature-based localization results are calculated separately first, and then fused by Kalman Filter or sliding window-based state estimator. However, these loosely-coupled strategies share a common shortcoming: they all require the HD map landmarks to be minimally self-sufficient for solving for the vehicle poses. Nevertheless, in a real application, this requirement clearly cannot always be satisfied.

In contrast with the previous works, we introduce a tightly-coupled sliding window-based optimization strategy to fuse the map-matching and visual feature localization constraints. Similar idea is also utilized in our previous work [26]. The basic concept is to optimize the poses with the minimization target consisting of both visual landmark feature residuals and HD map landmark feature residuals in a certain length of past frames. As a result, the localization estimator can provide the global pose of the current frame even when the current HD map observations are insufficient. Moreover, with the aid of the more abundant and accurate constraints of visual feature landmarks, the system can be more robust against the HD map observation noise and offer a more accurate localization result. However, the calculation of SCM for multi-frames in the tightly-coupled sliding window-based optimization is generally time-consuming, making it challenging to meet the real-time performance. As a further improvement of tightly-coupled strategy in [26], a linearization approximation algorithm that simplifies the SCM residual is also proposed to accelerate the overall optimization process. With this algorithm, the tightly-coupled sliding window-based optimization can be solved in real-time with a negligible performance drop. The contributions of this article are summarized as follows:

1) We adopted the semantic chamfer matching (SCM) as the monocular map matching model and derived its analytical derivatives with respect to the 6-DoF camera pose. Also, an outlier association rejection strategy is proposed, thereby allowing both efficient and robust map-matching optimization.
2) A tightly-coupled sliding window-based optimization algorithm to fuse visual feature and HD map landmark features is proposed. Moreover, a linearization approximation algorithm is proposed to accelerate the calculation of SCM during the optimization to ensure the real-time performance of the whole system.
3) A large scale dataset with HD map landmarks on KAIST Urban Dataset and Shougang Park is built for evaluating the map matching localization algorithms. Experiments are conducted on the proposed dataset, and the results have demonstrated the robustness and high precision of our self-localization algorithm.

II. Method

A. Problem Formulation

We start by considering the on-board camera pose as the equivalent vehicle pose for this study, as the camera is fixed to the vehicle. As such, the 6-DoF pose of camera frame $\mathcal{C}_t$ at time t in global frame $\mathcal{G}$ is defined as ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t} = \{{}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_t}, {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_t}\}$, where ${}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_t} \in \mathbb{R}^3$ represents the translation vector from the origin of $\mathcal{G}$ to the origin of $\mathcal{C}_t$, and ${}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_t} \in \mathrm{SO}(3)$ represents the 3×3 rotation matrix from frame $\mathcal{G}$ to frame $\mathcal{C}_t$. The problem of vehicle localization can be defined as the camera pose ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t}$ estimation relative to the HD map, given the monocular image with HD map landmark observations up to time t.

B. HD map Landmark

The HD map landmarks are composed of various road elements represented in vectorized format. This study utilizes the lane boundaries and poles for localization because these elements provide the basic localization cues and are commonplace in structured road scenarios. The HD map is denoted as $\mathcal{M} = \{\mathcal{M}_i\}$. Each landmark $\mathcal{M}_i$ with its semantic category s_i is modeled as a series of control 3D points $\{\mathbf{m}_{i,j} \in \mathbb{R}^3\}_{j=1:N_i}$ sampled uniformly in the 3D space for a unified representation, with N_i as the total number of control points of landmark $\mathcal{M}_i$.

As for the HD map landmark observations in the image plane, thanks to the development of the deep learning technique, it is now feasible to efficiently extract semantic objects using semantic segmentation [45][46] or panoptic segmentation [47], including lane detection [48][49]. In this study, we design a network based on FCN [45] to detect lane boundaries and poles. However, in real-world applications, the observations of HD map landmarks (e.g., the lane boundaries) are of great variety in shapes when observed from the images. Although the existing polylines representation can model these lane curve observations in most of the structured highway scenarios, it cannot accurately fit the corners and sharp curves in common urban areas. In addition, the complicated lane curve model makes the problem difficult to solve. Thus, this study adopts the semantic segmentation of lane lines and poles as the observation since it provides a straightforward approach to precisely describe the diverse shapes of HD map landmarks.

C. Semantic Chamfer Matching

We start by formulating the map matching problem in a single frame HD map observation. Given the initial camera pose ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t}$, the 3D vector HD map landmarks points $\mathbf{m}_{i,j} \in \mathcal{M}_i$ can be projected into the image space. The map-matching problem is to find an optimal camera pose ${}^{\mathcal{G}}\mathbf{x}^*_{\mathcal{C}_t}$ which can minimize

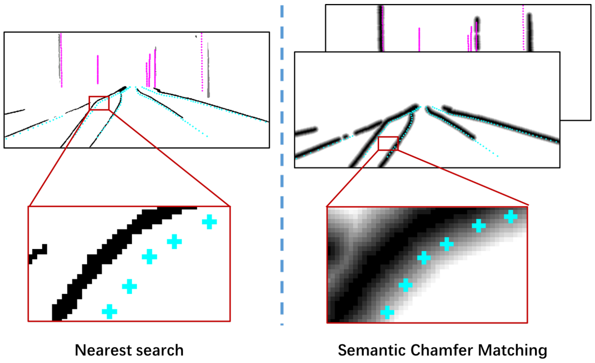


Fig. 2. Corresponding observation searching results in map-matching. The result of the nearest search is shown on the left, and that of the SCM method on the right. In the distance image of SCM, the darker pixel indicates the smaller distance value. The distance from each pixel to its nearest detection pixel can be directly queried in the distance image.

the cost model d between the projected HD map landmark points and their corresponding observations, as formulated in (1):

$$ {}^{\mathcal{G}}\mathbf{x}^*_{\mathcal{C}_t} = \operatorname*{argmin}_{{}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t}} \sum_{\mathcal{M}_i \in \mathcal{M}} \sum_{j=1}^{N_i} d\left(\mathbf{z}_{\mathbf{m}_{i,j}}, \mathbf{m}^{\mathbf{I}_t}_{i,j}\right) \tag{1} $$

where $\mathbf{z}_{\mathbf{m}_{i,j}}$ is the observation of landmark point $\mathbf{m}_{i,j}$ in the image $\mathbf{I}_t$, and $\mathbf{m}^{\mathbf{I}_t}_{i,j}$ represent the 2D projection result of $\mathbf{m}_{i,j}$ given the camera pose ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t}$, i.e.:

$$ \mathbf{m}^{\mathbf{I}_t}_{i,j} = \pi\left(\mathbf{m}^{\mathcal{C}_t}_{i,j}\right) = \pi\left({}^{\mathcal{G}}\mathbf{R}^T_{\mathcal{C}_t}\left(\mathbf{m}_{i,j} - {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_t}\right)\right) \tag{2} $$

Suppose the images have been undistorted beforehand, we adopt the pinhole model $\pi(\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$ as the camera model:

$$ \pi\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \frac{1}{z}\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{z}\mathbf{K}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3} $$

where $\mathbf{K}$ is the intrinsic matrix.

However, finding $\mathbf{z}_{\mathbf{m}_{i,j}}$ in the semantic segmentation results turns out to be non-trivial because of the ambiguity in the matching process between the pixel-wise observations and their corresponding HD map landmark features. A common strategy is to associates each projected 3D landmark point to the nearest pixels of the perceived landmark [17] as shown in Fig. 2.

Given the segmentation result $\mathbf{S}_t$ of frame t, the cost model of HD map control points $\mathbf{m}_{i,j}$ with the detection result is formulated as:

$$ d(\mathbf{z}_{\mathbf{m}_{i,j}}, \mathbf{m}^{\mathbf{I}_t}_{i,j}) = \min_{\mathbf{m}', \mathbf{S}_t(\mathbf{m}')=s_i} \left\| \mathbf{m}' - \mathbf{m}^{\mathbf{I}_t}_{i,j} \right\| \tag{4} $$

With this cost model, the camera pose ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t}$ can be solved using an iterative optimization approach. However, for each optimization iteration, the nearest pixel $\mathbf{z}_{\mathbf{m}_{i,j}}$ of each projected HD map control point $\mathbf{m}^{\mathbf{I}_t}_{i,j}$ should be re-calculated, thus negatively impacting the efficiency if the naive exhaustive searching is applied.

To solve this problem, SCM extends the chamfer matching in [1] into a semantic version to efficiently calculate the cost model $d(\cdot,\cdot)$ of each $\mathbf{m}^{\mathbf{I}_t}_{i,j}$ with its associated segmented pixel $\mathbf{z}_{\mathbf{m}_{i,j}}$. The algorithm first generates the distance images $\{\mathbf{D}^s_t, s \in \mathcal{S}\}$ for each kind of HD map landmarks from the image segmentation result $\mathbf{S}_t$, with $\mathcal{S}$ as the set of semantic categories of all kinds of HD map landmark features. In this work, $\mathcal{S} = \{\text{LaneBoundary}, \text{Pole}\}$. The distance image is formed by augmenting each pixel with its distance to the nearest non-zero pixel. Thus it is essentially a lookup table for querying the nearest distance for all pixels in the image space.

Given a set of projected HD map control points $\mathbf{m}^{\mathbf{I}_t}_{i,j} = (u, v)$ with type s_i, the nearest distance can be approximated by bi-linear interpolation in $\mathbf{D}^{s_i}_t$ as shown in (5):

$$ \min_{\mathbf{m}', \mathbf{S}_t(\mathbf{m}')=s_i} \left\| \mathbf{m}' - \mathbf{m}^{\mathbf{I}_t}_{i,j} \right\| = \mathbf{D}^{s_i}_t(\mathbf{m}^{\mathbf{I}_t}_{i,j}) $$

$$ = \begin{bmatrix} 1-\delta v \\ \delta v \end{bmatrix}^T \begin{bmatrix} \mathbf{D}^{s_i}_t(\bar{u},\bar{v}) & \mathbf{D}^{s_i}_t(\bar{u}+1,\bar{v}) \\ \mathbf{D}^{s_i}_t(\bar{u},\bar{v}+1) & \mathbf{D}^{s_i}_t(\bar{u}+1,\bar{v}+1) \end{bmatrix} \begin{bmatrix} 1-\delta u \\ \delta u \end{bmatrix} \tag{5} $$

where $\bar{u} = \lfloor u \rfloor$ and $\delta u = u - \bar{u}$ and the same for $\bar{v}$ and δv. $\mathbf{D}^{s_i}_t(\cdot,\cdot)$ represents the corresponding pixel value in $\mathbf{D}^{s_i}_t$. During the process of solving the map-matching problem, since the distance map is calculated beforehand, the nearest search at each optimization iteration for each $\mathbf{m}^{\mathbf{I}_t}_{i,j}$ can be calculated as the pixel value query with $O(1)$ computation complexity, which is significantly more efficient than the naive exhaustive search method. In our implementation, we first separate the segmentation result $\mathbf{S}_t$ into $\{\mathbf{S}^s_t\}$ with different semantic categories, and the distance images $\{\mathbf{D}^s_t\}$ are subsequently computed by the efficient two-pass algorithm [50] using the L2 norm with the input of $\{\mathbf{S}^s_t\}$ respectively.

To tackle the noisy image perception, instead of solely using robust loss function in [3], a gating operation is applied as the outlier rejection strategy to the distance image $\mathbf{D}^s_t$, i.e.:

$$ \hat{\mathbf{D}}^s_t(u,v) = \begin{cases} \mathbf{D}^s_t(u,v) & \mathbf{D}^s_t(u,v) < T \\ T & \mathbf{D}^s_t(u,v) \geq T \end{cases} \tag{6} $$

where T is the gating threshold. In our implementation, T is set as 20. By using this strategy, the value of areas with distance larger than T will be constant and has zero gradient. As a result, the projected HD map points lying within these areas will not be counted in the optimization.

D. Tightly-coupled Map-Matching

In this section, we introduce the tightly-coupled sliding window-based optimization strategy that fuses the map-matching process with visual features in TM^{3}Loc. The sliding window-based optimization optimizes the current camera pose together with a batch of historical camera poses in a certain window size [51] and has been successfully applied in popular SLAM systems [38][52]. The optimization window slides forward with time, with each additional new frame marginalizing the oldest frame in a first-in, first-out fashion (FIFO). The marginalization operation converts the original constraints of marginalizing frames into prior information for the states in the sliding window. The full state vector in the sliding window, denote as $\mathcal{X}$, is the length of K. Since visual feature landmarks only provide constraints in the local coordinate, the full state vector is composed of K camera pose states ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}} = \left\{{}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}\right\}_{k=1:K}$ in the first camera frame $\mathcal{C}_0$. As the final output of camera pose is in the global frame $\mathcal{G}$, a global to local 6-DoF transformation state ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$ is also defined in $\mathcal{X}$. The states of visual feature landmarks appearing in the sliding window are also included in the inverse depth form. λ_i is the inverse depth of visual feature landmark $\mathbf{c}_i$ related to the frame with its first observation. It is important to note that the states of HD map landmarks are not estimated since their prior information is sufficiently accurate. The full state vector $\mathcal{X}$ is defined as:

$$\begin{aligned}
\mathcal{X} &= \left[{}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_1}, \cdots, {}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_K}, {}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}, \lambda_1, \lambda_2, \cdots, \lambda_M\right] \\
{}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k} &= \left[{}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k}, {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}\right], n \in [1, K] \\
{}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0} &= \left[{}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_0}, {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}\right] \\
\lambda_m &\in \mathbb{R}, m \in [1, M]
\end{aligned} \tag{7}$$

The kth global camera pose in the sliding window ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_k}$ can be calculated as:

$$\begin{bmatrix} {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_k} & {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_k} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0} & {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k} & {}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k} \\ 0 & 1 \end{bmatrix} \tag{8}$$

The system is to find the optimal state vector $\mathcal{X}$ by minimizes the Mahalanobis distance of all measurement residuals $\mathbf{r}(\mathcal{X})$ in the sliding window:

$$\begin{aligned}
\mathcal{X}^* =& \underset{\mathcal{X}}{\text{argmin}} \left\|\mathbf{r}(\mathcal{X})\right\|^2_{\mathbf{\Omega}} = \\
& \underset{\mathcal{X}}{\text{argmin}} \left\{ \left\|\mathbf{r}_p - \mathbf{H}_p \mathcal{X}\right\|^2 + \sum_{\mathbf{c}_i \in \mathcal{C}} \rho\left(\left\|\mathbf{r}_{\mathcal{C}}(\mathbf{z}^k_{\mathbf{c}_i}, \mathcal{X})\right\|_{\mathbf{\Omega}^{\mathcal{C}}}\right)\right. \\
& \left. + \sum_{\mathcal{M}_i \in \mathcal{M}'} \sum_{j=1}^{N_i} \rho\left(\left\|\mathbf{r}_{\mathcal{M}}\left(\mathbf{z}^k_{\mathbf{m}_{i,j}}, \mathcal{X}\right)\right\|_{\mathbf{\Omega}^{\mathcal{M}}}\right)\right\}
\end{aligned} \tag{9}$$

where $\mathbf{r}_{\mathcal{C}}\left(\mathbf{z}^k_{\mathbf{c}_i}, \mathcal{X}\right)$ is the visual landmark residual and $\mathbf{r}_{\mathcal{M}}(\mathbf{z}^k_{\mathbf{m}_i}, \mathcal{X})$ is the HD map landmark residual. $\mathcal{C}$ and $\mathcal{M}'$ are the sets of 3D visual landmarks and HD map landmarks observed in the sliding window. $\{\mathbf{r}_p, \mathbf{H}_p\}$ is the prior information derived from marginalization during the sliding window-based optimization. $\rho(\cdot)$ is the Huber loss function, a robustify function that makes system robust to outlier noise, as defined in (10):

$$\rho(a) = \begin{cases} a^2, & |a| \leq \delta \\ 2|a|\delta - \delta^2, & |a| > \delta \end{cases} \tag{10}$$

with δ as the parameter that can be adjusted for different levels of outlier suppression strength. To optimize the (9), the Levenberg-Marquardt (LM) algorithm is utilized:

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\Delta\mathbf{x} = -\mathbf{J}^T\mathbf{r} \tag{11}$$

where $\mathbf{J}$ is the jacobian matrix of $\mathbf{r}(\mathcal{X})$ w.r.t. the state vector $\mathcal{X}$. The algorithm iteratively solves for the $\Delta\mathbf{x}$, and $\mathcal{X}$ is updated from k step to $k+1$ step as follows:

$$\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k \oplus \Delta\mathbf{x} \tag{12}$$

The whole framework of the proposed tightly-coupled optimization is visualized in Fig. 3.

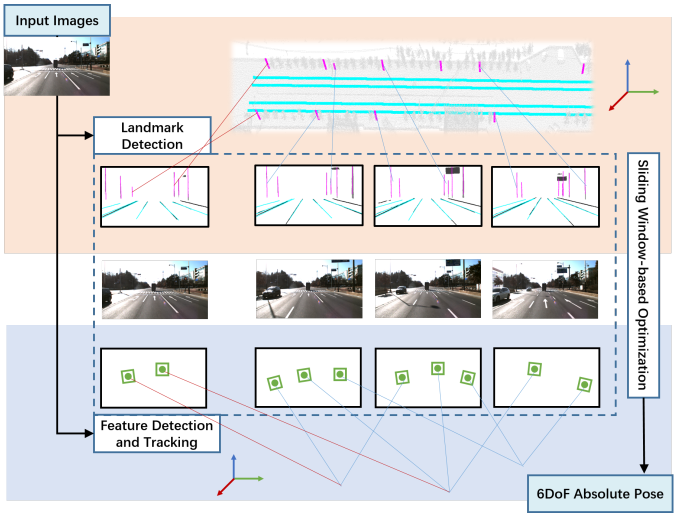

Input Images
Landmark Detection
Feature Detection and Tracking
Sliding Window-based Optimization
6DoF Absolute Pose

Fig. 3. The framework of the proposed tightly-coupled optimization in TM^{3}Loc. During the optimization, the absolute pose of the current input image is estimated by all the HD map landmark constraints (represented in frame $\mathcal{G}$) together with all visual feature constraints (represented in frame $\mathcal{C}_0$) in the sliding window.

Visual landmark residual. The underlying concept of visual landmark residual is finding the image points from different frames corresponding to a common 3D visual landmark point to restrain poses in these frames. In our implementation, the visual feature points are detected using Shi-Tomasi algorithm [53], and tracked using the optical flow [54]. The inverse depth model is adopted to describe the 3D visual landmarks. The observation of visual landmarks is defined in the normalized image plane, which is obtained by applying the inverse camera projection π^{-1} to "lift" the pixels of observed visual landmarks in the image plane to the camera coordinate with the depth of 1. Considering the visual landmark $\mathbf{c}_i$ firstly observed in frame $\mathcal{C}_{k_0}$, the residual of its observation in frame $\mathcal{C}_k$ can be defined as:

$$\begin{aligned}
\mathbf{p}^{\mathcal{C}_\mathbf{0}}_{\mathbf{c_i}} &= {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_{k_0}} \frac{1}{\lambda_i} \mathbf{z}^{k_0}_{\mathbf{c}_i} + {}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_{k_0}} \\
\mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}} &= {}^{\mathcal{C}_0}\mathbf{R}^T_{\mathcal{C}_k} \left(\mathbf{p}^{\mathcal{C}_\mathbf{0}}_{\mathbf{c_i}} - {}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k}\right) \\
\mathbf{r}_{\mathcal{C}}\left(\mathbf{z}^k_{\mathbf{c}_i}, \mathcal{X}\right) &= \pi(\mathbf{z}^k_{\mathbf{c}_i}) - \pi\left(\mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}\right)
\end{aligned} \tag{13}$$

where $\mathbf{z}^k_{\mathbf{c}_i}$ and $\mathbf{z}^{k_0}_{\mathbf{c}_i}$ represent the normalized observations of visual feature $\mathbf{c}_i$ in frame $\mathcal{C}_k$ and $\mathcal{C}_{k_0}$. The Jacobian matrices $\mathbf{J}_{\mathbf{c}_i}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k})$ and $\mathbf{J}_{\mathbf{c}_i}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_{k_0}})$ of $\mathbf{r}_{\mathbf{c}_i}\left(\mathbf{z}^k_{\mathbf{c}_i}, \mathcal{X}\right)$ w.r.t. the camera pose ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}$ and ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_{k_0}}$ are derived on their $\mathfrak{se}(3)$ Lie Algebra manifold. With z denoting the depth of $\mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}$, then

$$\frac{\partial \pi\left(\mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}\right)}{\partial \mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}} = \begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{f_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{f_y}{z^2} \end{bmatrix} \tag{14}$$

$$\mathbf{J}_{\mathbf{c}_i}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}) = \frac{\partial \pi\left(\mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}\right)}{\partial \mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}} \begin{bmatrix} {}^{\mathcal{C}_0}\mathbf{R}^T_{\mathcal{C}_k} & -\left[{}^{\mathcal{C}_0}\mathbf{R}^T_{\mathcal{C}_k} \mathbf{p}^{\mathcal{C}_\mathbf{0}}_{\mathbf{c_i}}\right]_\times \end{bmatrix} \tag{15}$$

$$\mathbf{J}_{\mathbf{c}_i}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}) = \frac{\partial \pi\left(\mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}\right)}{\partial \mathbf{p}^{\mathcal{C}_\mathbf{k}}_{\mathbf{c_i}}} \begin{bmatrix} -{}^{\mathcal{C}_0}\mathbf{R}^T_{\mathcal{C}_k} & {}^{\mathcal{C}_0}\mathbf{R}^T_{\mathcal{C}_k} {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_{k_0}} \left[\frac{1}{\lambda_i} \mathbf{z}^{k_0}_{\mathbf{c}_i}\right]_\times \end{bmatrix} \tag{16}$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix transformation. The Jacobian matrix $\mathbf{J}_{\mathbf{c}_i}(\lambda_i)$ w.r.t the inverse depth λ_i is:

$$\mathbf{J}_{\mathbf{c}_i}(\lambda_i) = \frac{\partial \pi\left(\mathbf{p}_{\mathbf{c}_i}^{\mathcal{C}_k}\right)}{\partial \mathbf{p}_{\mathbf{c_i}}^{\mathcal{C}_\mathbf{k}}} {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}^T \, {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_{k_0}} \frac{1}{\lambda_i^2} \tag{17}$$

HD map landmark residual. The HD map landmark residual is constructed using the SCM as outlined in Sec. II-C. First, the sampled control points of the HD map landmarks are transformed from the global coordinate $\mathcal{G}$ into the local coordinate $\mathcal{C}_0$ by ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$, and then projected into the image plane with local camera pose ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}$. For a sample point $\mathbf{m}_{i,j}$ of HD map landmark $\mathcal{M}_i$, given its corresponding observation $\mathbf{z}_{\mathbf{m}_{i,j}}^k$ in frame $\mathcal{C}_k$, the residual is defined as:

$$\begin{aligned} \mathbf{m}_{i,j}^{\mathcal{C}_0} &= {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}^T \left(\mathbf{m}_{i,j} - {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_0}\right) \\ \mathbf{m}_{i,j}^{\mathcal{C}_k} &= {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}^T \left(\mathbf{m}_{i,j}^{\mathcal{C}_0} - {}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k}\right) \\ \mathbf{r}_{\mathcal{M}}\left(\mathbf{z}_{\mathbf{m}_{i,j}}^k, \mathcal{X}\right) &= d\left(\mathbf{z}_{\mathbf{m}_{i,j}}^k, \pi\left(\mathbf{m}_{i,j}^{\mathcal{C}_k}\right)\right) \end{aligned} \tag{18}$$

The jacobian matrix $\mathbf{J}_{\mathbf{m}_{i,j}}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k})$ and $\mathbf{J}_{\mathbf{m}_{i,j}}({}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0})$ w.r.t ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}$ and ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$ are:

$$\mathbf{J}_{\mathbf{m}_{i,j}}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}) = \frac{\partial \mathbf{D}_k^{s_i}}{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}} \frac{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}}{\partial \mathbf{m}_{i,j}^{\mathcal{C}_k}} \begin{bmatrix} -{}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}^T & \left[\mathbf{m}_{i,j}^{\mathcal{C}_k}\right]_\times \end{bmatrix} \tag{19}$$

$$\mathbf{J}_{\mathbf{m}_{i,j}}({}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}) = \frac{\partial \mathbf{D}_k^{s_i}}{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}} \frac{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}}{\partial \mathbf{m}_{i,j}^{\mathcal{C}_k}} \begin{bmatrix} -{}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}^T \, {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}^T & {}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k} \left[\mathbf{m}_{i,j}^{\mathcal{C}_0}\right]_\times \end{bmatrix} \tag{20}$$

where

$$\frac{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}}{\partial \mathbf{m}_{i,j}^{\mathcal{C}_k}} = \begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{f_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{f_y}{z^2} \end{bmatrix}, \quad \mathbf{m}_{i,j}^{\mathcal{C}_k} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{21}$$

and $\frac{\partial \mathbf{D}_k^{s_i}}{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}}$ is approximated as the pixel gradient of $\mathbf{D}_k^{s_i}$ at $\mathbf{m}_{i,j}^{\mathbf{I}_k}$, i.e.:

$$\frac{\partial \mathbf{D}_k^{s_i}}{\partial \mathbf{m}_{i,j}^{\mathbf{I}_k}} = \frac{1}{2} \begin{bmatrix} \mathbf{D}_k^{s_i}(\bar{u}+1, \bar{v}) - \mathbf{D}_k^{s_i}(\bar{u}-1, \bar{v}) \\ \mathbf{D}_k^{s_i}(\bar{u}, \bar{v}+1) - \mathbf{D}_k^{s_i}(\bar{u}, \bar{v}-1) \end{bmatrix}^T \tag{22}$$

In our implementation, the number of sample points of HD map landmark features is around 150 for single frame. As a result, when performing the LM optimization, the HD map landmark residuals and jacobians needs around $150K$ times calculation for each LM optimization update step, making naive SCM not suitable for tightly-coupled sliding window-based optimization.

Linear approximated residual. Next, we introduce the linearization approximation algorithm for accelerating the HD map residual calculation of SCM. Given the set $\mathcal{M}^k$ of all sample points of HD map landmarks at frame k, when applying LM optimization algorithm in (11), the corresponding block of HD map residuals is:

$$(\mathbf{H}_{\mathcal{M}^k} + \lambda \mathbf{I}) \begin{bmatrix} \Delta^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k} \\ \Delta^{G}\mathbf{x}_{\mathcal{C}_0} \end{bmatrix} = -\mathbf{b}_{\mathcal{M}^k} \tag{23}$$

where $\mathbf{H}_{\mathcal{M}^k}$ and $\mathbf{b}_{\mathcal{M}^k}$ are calculated as:

$$\begin{aligned} \mathbf{H}_{\mathcal{M}^k} &= \sum_{\mathbf{m}_i \in \mathcal{M}^k} \begin{bmatrix} \mathbf{J}_{\mathbf{m}_i}({}^{\mathbf{C}_0}\mathbf{x}_{\mathcal{C}_k}) \\ \mathbf{J}_{\mathbf{m}_i}({}^{G}\mathbf{x}_{\mathcal{C}_0}) \end{bmatrix}^T \begin{bmatrix} \mathbf{J}_{\mathbf{m}_i}({}^{\mathbf{C}_0}\mathbf{x}_{\mathcal{C}_k}) \\ \mathbf{J}_{\mathbf{m}_i}({}^{G}\mathbf{x}_{\mathcal{C}_0}) \end{bmatrix} \\ \mathbf{b}_{\mathcal{M}^k} &= \sum_{\mathbf{m}_i \in \mathcal{M}^k} \begin{bmatrix} \mathbf{J}_{\mathbf{m}_i}({}^{\mathbf{C}_0}\mathbf{x}_{\mathcal{C}_k}) \\ \mathbf{J}_{\mathbf{m}_i}({}^{G}\mathbf{x}_{\mathcal{C}_0}) \end{bmatrix}^T \mathbf{r}_{\mathbf{m}_i} \end{aligned} \tag{24}$$

Since $\mathbf{H}_{\mathcal{M}^k}$ is a symmetric matrix, one can perform Cholesky decomposition as $\mathbf{H}_{\mathcal{M}^k} = \mathbf{J}_{\mathcal{M}^k}^T \mathbf{J}_{\mathcal{M}^k}$. By further introducing $\mathbf{r}_{\mathcal{M}^k} = (\mathbf{J}_{\mathcal{M}^k}^\dagger)^T \mathbf{b}_{\mathcal{M}^k}$, (23) can be transformed as:

$$\mathbf{J}_{\mathcal{M}^k}^T \mathbf{J}_{\mathcal{M}^k} \begin{bmatrix} \Delta^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k} \\ \Delta^{G}\mathbf{x}_{\mathcal{C}_0} \end{bmatrix} = -\mathbf{J}_{\mathcal{M}^k}^T \mathbf{r}_{\mathcal{M}_k} \tag{25}$$

This transformation indicates that the overall HD map residuals at frame k are equivalent to one single residual block $\mathbf{r}$ satisfying:

$$\begin{aligned} \mathbf{r}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}, {}^{G}\mathbf{x}_{\mathcal{C}_0}) &= \mathbf{r}_{\mathcal{M}^k} \\ \frac{\partial \mathbf{r}({}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}, {}^{G}\mathbf{x}_{\mathcal{C}_0})}{\partial \left[\Delta^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}, \Delta^{G}\mathbf{x}_{\mathcal{C}_0}\right]} &= \mathbf{J}_{\mathcal{M}^k} \end{aligned} \tag{26}$$

Normally, $\mathbf{J}_{\mathcal{M}^k}$ is relative to ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}$ and ${}^{G}\mathbf{x}_{\mathcal{C}_0}$. As a result, it should be re-calculated after each round of updates in the LM optimization. However, if ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}$ and ${}^{G}\mathbf{x}_{\mathcal{C}_0}$ have been optimized several times in the previous sliding window-based optimizations, we can assume that they are already close to the local optimum and therefore will change only a little after each round of update. Under this assumption, the proposed algorithm is to replace $\mathbf{J}_{\mathcal{M}^k}$ by a constant jacobian $\bar{\mathbf{J}}_{\mathcal{M}^k}$ that is jacobian of the HD map residuals at the initial value (${}^{\mathcal{C}_0}\bar{\mathbf{x}}_{\mathcal{C}_k}$, ${}^{G}\bar{\mathbf{x}}_{\mathcal{C}_0}$) before optimization, deriving the linear approximated residual $\mathbf{r}_{\text{LA}}$ as:

$$\begin{aligned} \mathbf{r}_{\text{LA}}(\mathbf{z}^k, \mathcal{X}) &= \mathbf{r}_{\mathcal{M}^k} - \bar{\mathbf{J}}_{\mathcal{M}^k} \begin{bmatrix} {}^{\mathcal{C}_0}\bar{\mathbf{x}}_{\mathcal{C}_k} \\ {}^{G}\bar{\mathbf{x}}_{\mathcal{C}_0} \end{bmatrix} + \bar{\mathbf{J}}_{\mathcal{M}^k} \begin{bmatrix} {}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k} \\ {}^{G}\mathbf{x}_{\mathcal{C}_0} \end{bmatrix} \\ &:= \bar{\mathbf{r}}_{\text{LA}} + \bar{\mathbf{J}}_{\mathcal{M}^k} \begin{bmatrix} {}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k} \\ {}^{G}\mathbf{x}_{\mathcal{C}_0} \end{bmatrix} \end{aligned} \tag{27}$$

This approximation avoids the jacobian re-calculation of HD map residuals at each LM optimization round, accelerating the overall state estimation. Notice that the approximation can only be reasonable when the states at frame k are lying within the neighborhood of local optimum. To ensure the approximation is not applied on frames that have not been well solved, the algorithm introduces a variable n_k for each frame k to record the lifetime of frame k in the sliding window. Only frames with n_k larger than a threshold N_T will be approximated.

In the sliding window-based optimization, the K camera poses from the past frames are selected as keyframes. The keyframe selection strategy has been widely studied in the visual SLAM community. In our implementation strategy, the latest frame is added as a new keyframe when it has enough visual parallax with the second latest keyframe in the sliding window, leading to the removal of the oldest keyframe in the sliding window; otherwise, the second latest keyframe will be discarded. With this strategy, the feature landmarks in the sliding window can be observed by the keyframes with enough parallax so that their 3D positions can be estimated more accurately.

In the cost model (4) of SCM, the data association result is strongly related to the initial guess of the state poses. Thus, in order to reduce the false data association in SCM, a good initial guess of the camera pose is required. Therefore, in our implementation, we have proposed the initial guess generation strategy with the aid of visual features. The strategy predicts the initial guess with the current frame observations of the visual feature landmarks, of which their 3D positions have already been estimated in the sliding window process. The results of this prediction serve as the initial guess for the SCM, thereby helping improve the accuracy in generating the final data association for optimization. The whole tightly-coupled sliding window-based map matching pipeline is outlined in Algorithm 1.

Algorithm 1 Tightly-coupled Map Matching

```
Input:
    The HD map landmark feature set $\mathcal{M} = \{\mathcal{M}_i\}_{i=1:N}$;
    The segmentation result $\mathbf{S}_t = \{S_t^s, s \in \mathcal{S}\}$ of image;
    The tracked visual feature $\{\mathbf{c}_i \in \mathcal{C}\}$;
    The initial state $\mathcal{X}$;
Output:
    Optimized pose ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t}$;
Calculate the initial guess ${}^{\mathcal{C}_0}\hat{\mathbf{x}}_{\mathcal{C}_t}$ of ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_t} \in \mathcal{X}$ using visual features.
$\mathbf{D}_t \leftarrow \mathrm{DistanceTransform}(\mathbf{S}_t)$.
Construct $\mathbf{r}_{\mathrm{LA}}\left(\mathbf{z}^k, \mathcal{X}\right)$ for frames with $n_k > N_T$.
while Not Converge do
    for each keyframe $\mathcal{C}_k$ in the sliding window do
        Construct $\mathbf{r}_{\mathbf{c}_i}\left(\mathbf{z}^k_{\mathbf{c}_i} \mathcal{X}\right)$ for $\mathbf{c}_i \in \mathcal{C}$.
        Construct $\mathbf{r}_{\mathbf{m}_{i,j}}\left(\mathbf{z}^k_{\mathbf{m}_{i,j}}, \mathcal{X}\right)$ for $\mathbf{m}_{i,j} \in \mathcal{M}_i$ for frames with $n_k \leq N_T$.
        Calculate the jacobian matrix $\mathbf{J}$ of $\mathbf{r}$ w.r.t. $\mathcal{X}$.
        Calculate $\Delta\mathbf{x}$ using LM optimization.
        $\mathcal{X} \leftarrow \mathcal{X} \oplus \Delta\mathbf{x}$.
    end for
end while
${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_t} \leftarrow {}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0} \cdot {}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_t}$;
```

E. System Initialization

At the beginning of the optimization (9), a good initial value of the full state vector $\mathcal{X}$ is required due to the high non-linearity of the problem. Hence, a novel initialization algorithm is presented here to solve the initial guess of $\mathcal{X}$. The proposed algorithm first solves the Structure-from-Motion (SfM) problem given a sequence of K images to obtain the local camera poses and the inverse depth of visual feature landmarks. Although the estimation of poses and the feature point landmarks from SfM is pretty accurate, due to the depth ambiguity of the monocular camera, this estimation cannot be used as the initial guess directly since its scale is not observable. Moreover, the global-to-local transformation ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$ still remains unknown. To this end, we recover the scale of the local camera pose ${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k}$ and the inverse depth of visual feature landmarks, and ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$ by fusing the observation of the HD map landmarks in a loosely coupled manner. The whole initial pipeline is shown in Fig. 2, with three steps introduced as follows:

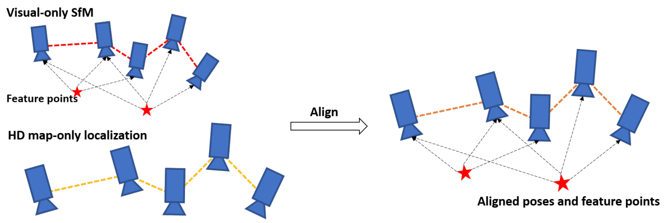


Fig. 4. The pipeline of system initialization shown as three parts. First, the visual-only SfM and the HD map-only localization are separately calculated. Then, the alignment between these two trajectories are conducted to recover the scale of poses and the visual features. Finally, the global-to-local transformation ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$. is obtained.

Visual-only SfM. The monocular visual pose is calculated to determine the up-to-scale frame-to-frame motion, and to obtain a robust visual-only pose result. Its initialization strategy employs a simplified Structure-from-Motion (SfM) process used in [52]. This step outputs the local camera pose $\{{}^{\mathcal{C}_0}\bar{\mathbf{x}}_{\mathcal{C}_k}\}_{k=1}^K = \{{}^{\mathcal{C}_0}\bar{\mathbf{R}}_{\mathcal{C}_k}, {}^{\mathcal{C}_0}\bar{\mathbf{t}}_{\mathcal{C}_k}\}_{k=1}^K$ with obscure scale.

HD map-only localization. To utilize the HD map observation to recover the scale s and ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$, we solve for the global camera poses $\{{}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_k}\}_{k=1}^K = \{{}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_k}, {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_k}\}_{k=1}^K$ from the image sequence by the HD map observations. Firstly, the initial camera pose is solved in the first frame, with the rough initial guess of the camera pose obtained by GNSS. Due to the noise in the GNSS signal, the initial guesses of camera pose is sampled uniformly around the GNSS position at a radius of 5 m. The camera pose of each sampling position is solved by HD map landmarks and their observations using the SCM method, as formulated in (1). The calculated first camera pose will then be used as the initial guess of the successive camera pose, and the same minimization process is performed to obtain a more accurate solution. The optimized cost serves as the criteria to decide whether the current pose is well solved - if the optimized cost is larger than a predefined threshold value, the given camera pose solution will be rejected. It is important to note that the localization is likely to fail in scenarios with sparse HD map landmark observations. As a result, the system should be initialized in the scenarios with plenty of HD map landmarks for reliable initialization results.

Trajectory fusion. After obtaining both the local camera pose from the visual-only SfM and the global camera pose from the HD map-only localization, the scale s and transformation ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$ can be solved.

$$
{}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_k} = {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0} {}^{\mathcal{C}_0}\bar{\mathbf{R}}_{\mathcal{C}_k} \tag{28}
$$

$$
{}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_k} = s {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0} {}^{\mathcal{C}_0}\bar{\mathbf{t}}_{\mathcal{C}_k} + {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_0} \tag{29}
$$

Equation (28) can be solved by converting the rotation matrix into quaternion format, i.e.

$$
{}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_k} = {}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_0} \otimes {}^{\mathcal{C}_0}\bar{\mathbf{q}}_{\mathcal{C}_k} = \lfloor {}^{\mathcal{C}_0}\bar{\mathbf{q}}_{\mathcal{C}_k} \rfloor_R {}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_0} := {}^{\mathcal{C}_0}\bar{\mathbf{Q}}_{\mathcal{C}_k} {}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_0} \tag{30}
$$

where $\lfloor \cdot \rfloor_R$ denote the right multiplication matrix of ${}^{\mathcal{C}_0}\bar{\mathbf{q}}_{\mathcal{C}_k}$. Given K rotation equation of (30) at different timestamps, ${}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}$ can be solved as follows:

$$\begin{bmatrix} {}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_1} \\ \vdots \\ {}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_K} \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{C}_0}\bar{\mathbf{Q}}_{\mathcal{C}_1} \\ \vdots \\ {}^{\mathcal{C}_0}\bar{\mathbf{Q}}_{\mathcal{C}_K} \end{bmatrix} {}^{\mathcal{G}}\mathbf{q}_{\mathcal{C}_0} \tag{31}$$

After solving the rotation ${}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}$, the translation ${}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_0}$ and scale s can be recovered by solving the following equation:

$$\begin{bmatrix} \mathbf{I} & {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}{}^{\mathcal{C}_0}\bar{\mathbf{t}}_{\mathcal{C}_1} \\ \vdots & \vdots \\ \mathbf{I} & {}^{\mathcal{G}}\mathbf{R}_{\mathcal{C}_0}{}^{\mathcal{C}_0}\bar{\mathbf{t}}_{\mathcal{C}_K} \end{bmatrix} \begin{bmatrix} {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_0} \\ s \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_1} \\ \vdots \\ {}^{\mathcal{G}}\mathbf{t}_{\mathcal{C}_K} \end{bmatrix} \tag{32}$$

Note that only the valid HD map-only localization results are used, and at least two valid localization results are required $(K \geq 2)$ to arrive at a unique solution.

After solving s and ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$, the translation of the local camera pose ${}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k}$ and the vision feature landmarks with real scale can be obtained:

$$\begin{aligned} {}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k} &= s^{\mathcal{C}_0}\bar{\mathbf{t}}_{\mathcal{C}_k} \\ \lambda_i &= s^{-1}\bar{\lambda}_i \end{aligned} \tag{33}$$

The rotation matrix ${}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}$ is kept as the result ${}^{\mathcal{C}_0}\bar{\mathbf{R}}_{\mathcal{C}_k}$ from visual-only SfM, since we believe that the pose estimation of visual-only SfM is more accurate than that of the HD map-only localization result for the more abundant visual features offered by the visual-only SfM. The whole initialization algorithm is shown in Algorithm 2.

Algorithm 2 Loosely-coupled Initialization Algorithm

```
Input:
    The set of observations of visual feature point $\mathcal{C}$ at each frame $\{\mathbf{z}^k_{\mathbf{c}_i}\}^K_{k=1}, \mathbf{c}_i \in \mathcal{C}$;
    The set of observations of HD map landmark $\mathcal{M}'$ of each frame $\{\mathbf{z}^k_{\mathbf{m}_{i,j}}\}^K_{k=1}, \mathbf{m}_{i,j} \in \mathcal{M}_i, \mathcal{M}_i \in \mathcal{M}'$;
Output:
    The state vector $\mathcal{X}$ for system initialization.
Calculate the visual-only SfM poses $\{{}^{\mathcal{C}_0}\bar{\mathbf{x}}_{\mathcal{C}_k}\}^K_{k=1}$ with feature point observation set $\{\mathbf{z}^k_{\mathbf{c}_i}\}^K_{k=1}$.
Calculate the HD map-only poses $\{{}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_k}\}^K_{k=1}$ with HD map landmark observation set $\{\mathbf{z}^k_{\mathbf{m}_{i,j}}\}^K_{k=1}$.
Solving the ${}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}$ and scale $s$ with (28) and (29).
${}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k} \leftarrow {}^{\mathcal{C}_0}\bar{\mathbf{R}}_{\mathcal{C}_k}$, ${}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k} \leftarrow s^{\overline{\mathcal{C}_0}}\mathbf{t}_{\mathcal{C}_k}$, $\lambda_i \leftarrow s^{-1}\bar{\lambda}_i$
${}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_k} \leftarrow \{{}^{\mathcal{C}_0}\mathbf{R}_{\mathcal{C}_k}, {}^{\mathcal{C}_0}\mathbf{t}_{\mathcal{C}_k}\}$
$\mathcal{X} \leftarrow \left[{}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_1}, \cdots, {}^{\mathcal{C}_0}\mathbf{x}_{\mathcal{C}_K}, {}^{\mathcal{G}}\mathbf{x}_{\mathcal{C}_0}, \lambda_1, \cdots, \lambda_M\right]$
```

III. Experimental Result

To evaluate the proposed algorithm, experiments are conducted on large scale localization datasets with HD maps. We first evaluate the localization accuracy of TM^{3}Loc on three sequences of a public available dataset, KAIST Urban Dataset and compare it with existing visual(-inertial)-based odometry and map-matching algorithms. Then, the effect of tightly-coupled visual features and the outlier rejection in SCM are examined. Moreover, the linearization approximation algorithm is also been examined to demonstrate its ability of balancing the localization performance and time cost. In the end, the TM^{3}Loc algorithm is tested on three self-recorded sequences in Shougang Park to demonstrate the localization ability of overall system in real applications.

TABLE I
KAIST urban dataset with HD map.

Sequence Name	Scenario	Length	#(Lane boundaries)	#(Poles)
Urban 34	bridge	1.1 km	560	153
Urban 23	highway	3.4 km	679	169
Urban 26	urban	4.0 km	1569	402

A. Experimental Setup & Evaluation Metric

To quantitatively evaluate the localization performance, we have examined the root mean square of Average Trajectory Error (ATE) and Average Rotation Error (ARE) of the estimated trajectories with the reference trajectories. ATE and ARE are calculated as:

$$\begin{aligned} \mathbf{e}_{\text{ATE}} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{t}_i - \mathbf{t}_i^{\text{GT}}\right\|^2} \\ \mathbf{e}_{\text{ARE}} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|2\arccos\operatorname{Re}\left(\mathbf{q}_{\text{GT}}^{-1}\otimes\mathbf{q}_i\right)\right\|^2} \end{aligned} \tag{34}$$

The ATE and ARE are calculated in two kinds of error: relative pose error (RPE) and absolute pose error (APE) with frames at interval of 10 m, thereby evaluating both the local trajectory smoothness and global localization accuracy. Also, the localization error along the lateral, longitudinal and vertical directions are provided separately. The reference trajectories calculated by multi-sensor SLAM algorithm are provided by the official KAIST urban dataset. The average localization error is calculated as the average of error of all sequences weighted by their frame numbers.

B. Localization Result on KAIST Urban Dataset

Since not public available HD maps provided, we manually built HD maps based on the point cloud map of KAIST Urban Dataset. In the dataset, the GNSS data are obtained from high precision VRS-GPS at 1Hz, and the IMU data are received at 100Hz. In addition, for each trajectory, a point cloud map scanned by a set of 2D SICK LiDAR is provided. We chose several classic scenarios in the KAIST urban dataset, including highway, bridge, and urban road, and manually labelled the lane boundaries and poles in the point cloud map. Table I summarizes the detail of selected sequences and their respective HD map information.

To better analyze the performance of the proposed localization algorithm, two visual-inertial odometry (VIO) algorithms VINS-Mono [52], OpenVINS [55] were evaluated for comparing the local trajectory smoothness since visual

TABLE II

Localization results in the KAIST Urban Dataset. For VIO, RPE of ATE and ARE are calculated while APE of ATE and ARE are evaluated for monocular HD map-matching algorithms. ATE_{RPE}, ATE_{APE}, lateral (Δ_x) and longitudinal (Δ_y) localization error are measured in meter (m). ARE_{RPE}, ARE_{APE} are measured in degree.

	Method	Urban 34		Urban 23		Urban 26		Average	
		ATE_{RPE}	ARE_{RPE}	ATE_{RPE}	ARE_{RPE}	ATE_{RPE}	ARE_{RPE}	ATE_{RPE}	ARE_{RPE}
VIO	VINS-Mono [52]	1.098	0.348	1.710	0.313	2.474	0.463	2.158	0.419
	OpenVINS [55]	1.237	0.186	0.865	0.205	0.497	0.145	0.749	0.165
	TM^{3}Loc (Ours)	0.159	0.326	0.321	0.242	0.165	0.230	0.203	0.251
		$\text{ATE}_{\text{APE}}/\Delta_x/\Delta_y$	ARE_{APE}	$\text{ATE}_{\text{APE}}/\Delta_x/\Delta_y$	ARE_{APE}	$\text{ATE}_{\text{APE}}/\Delta_x/\Delta_y$	ARE_{APE}	$\text{ATE}_{\text{APE}}/\Delta_x/\Delta_y$	ARE_{APE}
Monocular HD map Matching	Pauls et al. [3]	0.384/0.024/0.383	0.175	0.452/0.054/0.445	0.782	0.208/0.058/0.189	0.306	0.302/0.053/0.291	0.425
	HDMI-Loc [18]	0.228/0.081/0.202	1.379	0.454/0.090/0.435	1.325	0.372/0.158/0.287	1.385	0.369/0.136/0.309	1.373
	Wen et al. [26]	0.177/0.072/0.149	0.565	0.558/0.054/0.555	0.517	0.280/0.053/0.269	0.575	0.337/0.056/0.328	0.562
	TM^{3}Loc (Ours)	0.158/0.045/0.145	0.455	0.330/0.048/0.323	0.608	0.170/0.047/0.157	0.481	0.208/0.047/0.198	0.504

odometry are involved in the proposed method. Also, state-of-the-art HD map-matching localization algorithms Pauls et al. [3], HDMI-Loc [18] and Wen et al. [26] were selected for the comparison of global localization accuracy. In our implementation, VINS-Mono used a monocular camera and IMU data, while OpenVINS used the stereo camera and IMU data. As for algorithm [3], we used IMU and GNSS data to obtain the relative motion estimation instead of vehicle odometry since the motion obtained from IMU and GNSS can provide a completed 6-DoF ego-motion measurement. As for HDMI-Loc, we adjusted the original method to use a monocular camera instead of a stereo camera for the sake of fairness of comparison with other monocular map-matching algorithms. Therefore the subpatch images in HDMI-Loc were generated by inverse projection mapping (IPM) with a fixed homography matrix calculated from the extrinsic parameters between the camera and vehicle coordinates. As for TM^{3}Loc, we set the keyframe length K as 10, the maximal feature point number as 250 and the linear approximation threshold N_T as 3. As for [26], while the keyframe length and feature point number were the same as TM^{3}Loc, since the HD map residuals were modeled as point-to-line distance, the lane boundary observations with large curvature were rejected.

The results first indicate that VIO systems can dramatically drift during the long-term trajectory. This phenomenon commonly exists in visual(-inertial) odometry methods since the accumulative pose error can not be corrected without global pose observations. As for TM^{3}Loc, the RPE error is much smaller than traditional VIO method. This is because the HD map landmark features can provide absolute pose constraints, thereby correcting the scale of the localization result. With the aid of HD map landmark features, the visual feature landmarks estimated in TM^{3}Loc can have a lower drift in scale, resulting in a better local trajectory estimation. With regard to the monocular HD map matching methods, the results show that all methods provide a reasonable localization result, while the proposed TM^{3}Loc has demonstrated a 0.208 m localization error in average, which exceeds other map-matching methods by a large margin. Notice that [26] obtains a similar result in Urban 34, a straight road scenario, compared to TM^{3}Loc, but fails to achieve high precision in the rest two sequences with curve scenarios due to the limitation of point-to-line based HD map residuals. The result demonstrates that the SCM HD map residual model and optimization process of TM^{3}Loc can converge to a better map-matching result.

However, in urban 23, TM^{3}Loc has a much larger longitudinal error than other scenarios with the maximum localization error at 1.56 m. This case happens in some extreme highway scenarios where both the HD map landmarks and the visual features are sparse and far away from the vehicle. In this case, the visual features cannot accurately locate the vehicle, especially in translation motion, thereby resulting in a limited improvement for the vehicle localization. We believe this corner case can be resolved by fusing other sensors, such as IMUs and odometers, into the TM^{3}Loc localization framework.

In a nutshell, our localization system can provide an accurate localization result using only the lightweight HD map landmarks and the monocular camera and has clearly outperformed other existing multi-sensor fusion algorithms, especially when an abundance of visual features are present.

C. *Effect of Visual Features*

In this section, experiments are conducted to examine the proposed method in two aspects to dive deeper into the effect of our visual feature incorporation strategy. The first one is the initial guess generation strategy. In the tightly-coupled optimization, we used the visual features to solve for an initial guess of the current frame before performing the whole optimization (as described in Sec II.D.). The effect of this strategy is examined and discussed. The second one is the complementarity of visual features with HD map constraints. Since the HD map constraints can be insufficient in some cases (e.g. road intersections), solving the poses based solely on the HD map becomes rather sub-optimal. In these cases, both the visual feature and the HD map constraint can be considered in the tightly-coupled optimization process, making up for the deficiency of using only the HD map constraints. The experiments are designed to evaluate the effectiveness of this strategy.

1) *Initial Guess.* We compared the initial guess generation strategy in TM^{3}Loc with a baseline strategy that adopts the constant velocity motion model to predict the initial guess of the current frame. We tested different initial guess generation strategies by evaluating the deviations among their respective

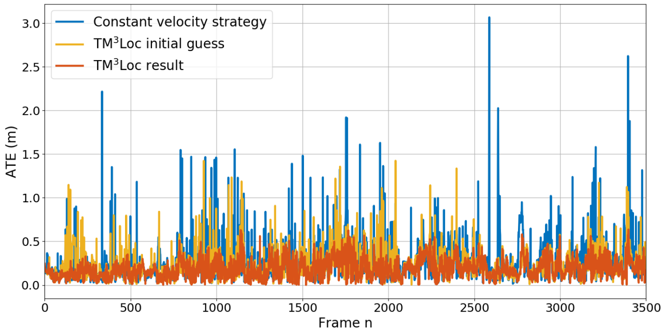


Fig. 5. The localization error of the HD map-only localization with initial guesses provided by the constant velocity model strategy and the TM^{3}Loc's strategy.

final optimized results in the experiment. The baseline method implementation system will be reset as the original TM^{3}Loc result after each estimation round to eliminate the accumulated error. A large error among the optimization results indicates the poor performance of the initial guess strategy. The evaluation of these two initial guess strategies is conducted on the Urban 26 dataset. The system setting is same as experiments in Sec. III-B. The localization error of both strategies is evaluated and plotted in Fig. 5.

From the figure, a big localization error can be observed from the baseline strategy. Although this strategy can provide reasonable results in some frames, the constant velocity model often fails to provide good initial guesses, particularly in urban scenarios where the vehicle movement is more irregular and hard to predict. As for the strategy in TM^{3}Loc, the error of initial guess is smaller, and the final estimation error is much better than the baseline strategy. As indicated, the better initial guesses provide better initial data association and help SCM quickly converge to the optimal solution. In conclusion, the effectiveness of the initial guess generation strategy in TM^{3}Loc is well demonstrated by the experiment results and can further strengthen the optimization effectiveness in the map-matching process.

2) *Complementarity with HD map Constraints.* The experiments are conducted specifically on the scenarios with insufficient HD map constraints to investigate the effect of the tightly-coupled optimization with the visual feature and HD map constraints. One of the common scenarios is the road intersection cases, where only poles can be used as the localization cues. We chose a video clip of 8s in length from Urban 26 at a selected road intersection. Three kinds of localization results are compared: HD map-only localization, visual odometry, and TM^{3}Loc. To better examine the effect of the tightly-coupled optimization, we removed the negative impact of the initial guesses in the HD map-only localization using the same initial guesses for TM^{3}Loc. As for the visual odometry, we first initialized the system identically as in TM^{3}Loc to get the absolute poses and scales, discarded the HD map constraints, and only estimated the poses by visual features of the testing video clip. The localization error of the three kinds of methods is summarized in Table. III.

TABLE III
LOCALIZATION ERROR OF HD MAP-ONLY, VISUAL ODOMETRY AND TM^{3}LOC LOCALIZATION METHODS, AND THEIR RESPECTIVE LATERAL, LONGITUDINAL AND VERTICAL ERRORS.

Method	ATE	Δ_x	Δ_y	Δ_z
HD map only	0.509	0.252	0.193	0.398
Visual odometry	0.228	0.021	0.223	0.039
TM^{3}Loc	0.135	0.031	0.122	0.040

The result first indicates a sizable deviation when localizing solely with HD map constraints, and such localization error is mainly due to the insufficient and noisy HD map constraints. For added clarity, the vertical localization error is also reported, and the result demonstrates that the vertical error plays an even larger role in localization error. The reason is that the poles cannot be treated as a sufficient constraint in a vertical direction, since the length of the detected poles and those of the poles in the HD map can be misaligned due to the data acquisition noises on both the detection results and those of the HD map. This vertical misalignment is further exaggerated at road intersections due to the lack of lane boundaries. As for the visual odometry, the poses are estimated by the visual features. The abundant visual feature constraints can ensure accurate pose estimation initially but will gradually drift off due to its natural characteristics. However, the localization error is much lower than that of the HD map-only localization. In addition, the results further indicate that the height error is no longer the main source of localization error since the visual feature points are distributed in diverse positions and altogether provide sufficient localization constraints in all directions. Lastly, the tightly-coupled optimization result shows a much lower localization error compared to the previous two methods. The reason is that, with the aid of visual feature in tightly-coupled optimization, the vertical constraints are effectively compensated, thereby avoiding the drift caused by insufficient HD map constraints and helping the system converge to better localization results. The tightly-coupled optimization uses visual features to eliminate the ill-posedness of HD map constraints, in the beginning, to achieve a better sensor fusion result effectively. On the contrary, the loosely-coupled optimization solution, which essentially takes a weighted average on the HD map-only and the visual odometry trajectories, has been outperformed due to its inevitable failure to eliminate the localization biases.

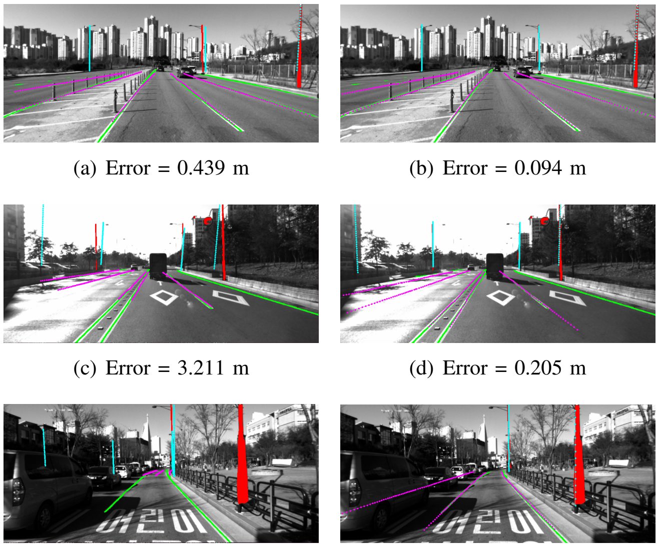
D. Effect of Outlier Rejection Strategy

An ablation study examines the effect of the outlier rejection strategy applied in SCM by comparing the performance of single frame map-matching results using SCM without and with the proposed outlier rejection strategy. We evaluate the map-matching with only SCM on Urban 34, Urban 23 and Urban 26, and summarize the results in Table. IV. The initial poses of SCM are set as the ground truth poses to remove the negative impact of the initial poses. The SCM residuals of every frame are optimized with an LM algorithm with a maximum of 50 iterations. Fig. 6 visualizes several corner case results by projecting the HD map landmarks on the

TABLE IV

ATE of SCM with and without proposed outlier rejection strategy in three sequences of KAIST Urban dataset. The initial guesses are set as ground truth poses.

	Urban 34	Urban 23	Urban 26
SCM(without OR)	2.544	2.378	1.177
SCM(with OR)	0.193	0.355	0.312


(a) Error = 0.439 m

(b) Error = 0.094 m

(c) Error = 3.211 m

(d) Error = 0.205 m

(e) Error = 44.070 m

(f) Error = 0.104 m

Fig. 6. Left: optimizing SCM residuals without outlier rejection strategy. Right: optimizing SCM residuals with proposed outlier rejection strategy. Position errors of estimated poses are reported. The poles and lane boundaries for perception are colored as red and green, while as cyan and magenta for HD map projection.

images using the estimated poses. The localization error of each case is also reported. In these cases, some poles and lane boundaries are not detected due to the hard visual condition, such as daze light or the occlusion by vehicles. As a result, the differences between HD map landmarks and perceived landmarks cause false data association in SCM. When outlier rejection is not applied, the outlier data association will lead to a false convergence direction. However, after applying the proposed outlier rejection strategy, these false data associations can be filtered. Without the effect of wrong residuals, the pose can be solved using the SCM residual model, as shown in the right column of Fig. 6, which demonstrates the effectiveness of the proposed outlier rejection strategy.

E. Effect of Linear Approximated Residuals

To examine how the introduced linear approximated residual effects TM^{3}Loc, we compare the system localization accuracy and the time cost for HD map residual calculation during optimization after applying linearization approximation with N_T varying from 3 to the sliding window size K. The system is tested on sequence Urban 26 of KAIST Urban Dataset with $K = 10$. The maximum optimization round of LM optimization is set as 50. The average HD map residual processing time and localization accuracy are visualized in Fig. 7. As for comparison, we also calculate the computation time of the HD map residual model used in [26], a point-to-line distance model with much simpler computation complexity.

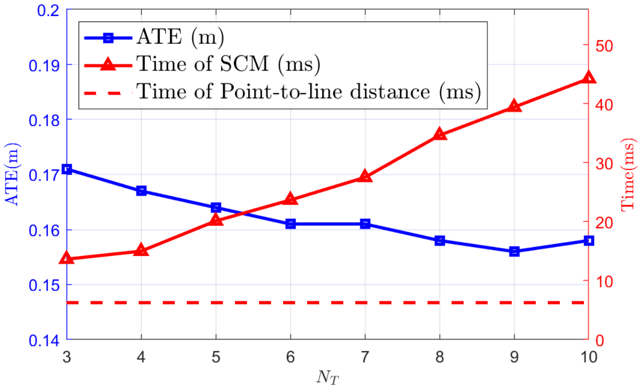
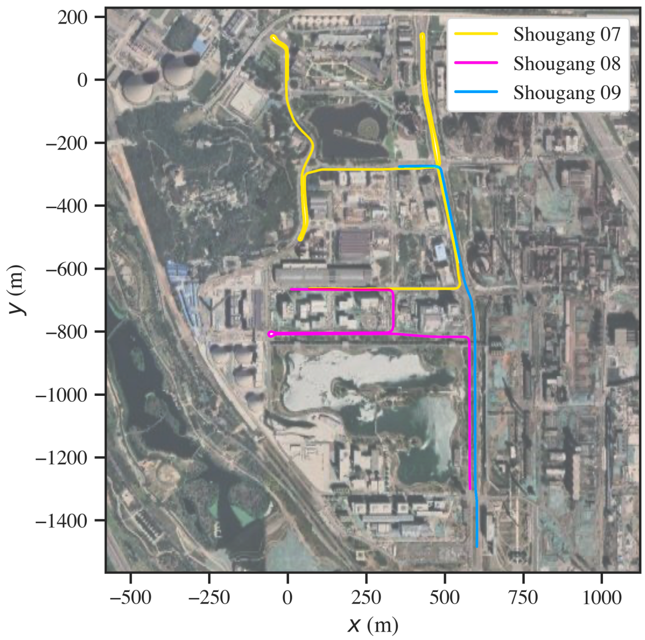


Fig. 7. Curves of localization accuracy (ATE) and computation time (ms) of HD map residuals modeled as SCM and point-to-line [26] with different N_T.

From the result, one can see that there is a great gap in time efficiency between the original SCM and the point-to-line model. After applying linearization approximation, the time for calculating the HD map residuals keeps reducing linearly as N_T becomes smaller and much closer to the time of the point-to-line model. This follows the expectation since the number of HD map residual that needs calculation is linearly relative to the number of frames applied with SCM residuals. Also, the calculation of residual and jacobians of linear approximated residuals is efficient, requiring only few additional computation time cost. On the other hand, while N_T gets smaller, the localization accuracy only slightly decreases (ATE increases around 0.015 m when N_T varies from 10 to 3), indicating that the linear approximation algorithm can well approximate the original SCM HD map residuals, without causing dramatic performance drop. This phenomenon demonstrates the proposed linearization approximation method can greatly reduce the calculation cost of HD map residuals, make it possible to incorporate the HD map residuals into tightly-coupled sliding window-based optimization. Compared to the point-to-line model, with the help of linearization approximation, SCM reaches a better trade-off in localization accuracy and time efficiency.

F. Localization in Actual Application

We demonstrate the effectiveness of the proposed TM^{3}Loc in an actual application in this subsection. Unlike the previous experiments on KAIST Urban Datasets, the image sequences are jointly collected when building the HD map. In the actual application, the HD map is usually built beforehand. The inconsistency between the pre-built HD map and online collected data will finally affect the localization accuracy. In the actual application demonstration, we validate TM^{3}Loc in Shougang Park, an urban scenario with the HD map built by the map builder. The absolute error of the built HD map is around 0.076 m measured by the hand-held RTK device. The data collecting vehicles are equipped with a monocular camera with a resolution of 1920×1080 at 10Hz, an OXTS INS system plugin with RTK signals that can output the vehicle poses at



From the result, one can observe that the overall localization error reaches at 0.255m and at 0.146m in vertical, which grows more significant than the experiments on KAIST Urban Dataset, especially on urban scenarios (Urban 26) despite the different testing places.

Fig. 8. Trajectories of Shougang 07, Shougang 08 and Shougang 09.

100Hz. The computing platform is a Nuvo-6108G computer with i7 6700 CPU and NVIDIA TITAN X GPU for real-time image segmentation processing and localization calculation. The number of extracted visual features for each frame is set as 250, and the number of the keyframe in sliding window K is set as 10, and N_T of linear approximation is set as 3 to ensure the real-time performance. The reference localization poses are obtained from the immediate OXTS INS system output. The algorithm is tested on three recorded sequences, Shougang 07, Shougang 08, and Shougang 09, with 3.2km, 2.0km, and 1.3km. The trajectories of these three sequences are plotted in Fig. 8. The localization result is shown in Table V and the runtime performance is summarized in Table VI.

From the result, one can observe that the overall localization error reaches at 0.255m and at 0.146m in vertical, which grows more significant than the experiments on KAIST Urban Dataset, especially on urban scenarios (Urban 26) despite the different testing places. The result shows that the inconsistency between the pre-built HD map and online collected data affects the localization result of the proposed algorithm. Firstly, due to the change of the road infrastructures, the lane markings and poles in the pre-built HD map can be inconsistent with the actual situations. Fig. 9 visualizes several corner cases when HD map landmarks fail to localize vehicles.

In these cases, the HD map lanedmarks cannot provide valid localization constraints in longitudinal or vertical direction. Although our system can pass through these cases thanks to the tightly-coupled visual features, the estimation can gradually drift if no valid localization provided for a long time.

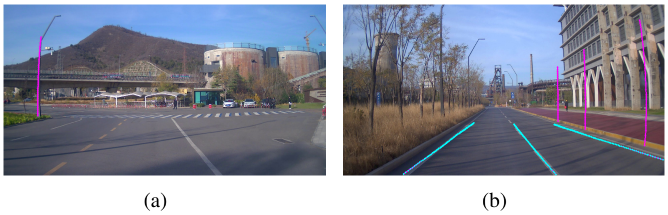

(a)
(b)

Fig. 9. Example images of corner cases that the pre-built HD map can be inconsistent with the actual situations. Pre-built HD map landmarks are projected on the images with estimated poses. One can see the missing lane boundaries in (a) and removed poles in (b).

This results in an increased localization error in longitudinal direction. Secondly, the absolute error of the HD map itself and the differences between the HD map coordinates and online RTK results can also magnify the localization error. However, this will not happen in previous experiments on KAIST Urban Dataset. When conducting localization on the identical sequences used for mapping, the resulting error will not be affected by the misalignment between the reference poses and HD map since they share the exact coordinates. Considering this effect, we also report the aligned ATE with SE(3) alignment. This operation can eliminate the constant biases between reference and estimated poses, therefore reflecting the structure error of the whole trajectory. The result shows that the estimated trajectories reaches at an error of 0.201m in average, which are accurate, especially in Shougang 09 where TM^{3}Loc reaches at an error of 0.114m.

As for the runtime performance, results show that the overall average processing time for each frame is 71.3 ms in a single thread, less than the time interval of subsequent images (100 ms), indicating the system can well meet the real-time requirement. In a word, the experiments demonstrate that the proposed TM^{3}Loc is able to well localize the vehicles for actual applications in terms of accuracy and efficiency.

TABLE V
ATE OF TM^{3}LOC LOCALIZATION METHODS IN THREE SHOUGANG PARK SEQUENCES IN UNIT METER (M), AS WELL AS ARE IN UNIT DEGREE (DEG). MOREOVER, THE ATE WITH SE(3) ALIGNMENT IS ALSO REPORTED.

Sequence	ATE	Δ_x	Δ_y	Δ_z	ATE(*aligned*)	ARE
Shougang 07	0.239	0.070	0.158	0.166	0.216	2.87
Shougang 08	0.299	0.124	0.229	0.146	0.218	2.76
Shougang 09	0.215	0.061	0.191	0.077	0.114	2.05
Average	0.255	0.089	0.188	0.146	0.201	2.69

IV. DISCUSSION

The experiments above showcase the high precision localization ability of the proposed TM^{3}Loc. The combination between the visual feature and HD map residuals can compensate each other when HD map landmark observations are insufficient, making it robust in most cases. However, the system might inevitably suffer the localization drift when the HD map landmark constraints are insufficient for a long

TABLE VI
RUNTIME PERFORMANCE OF TM^{3}LOC IN SHOUGANG PARK DATASETS. THE AVERAGE EXECUTION TIME OF EACH MAIN MODULE IS REPORTED.

	Main module	Time
Frontend	HD map landmark perception	11.3 ms
	HD map query & SCM pre-processing	10.9 ms
	Visual feature detection & tracking	29.3 ms
Backend	Visual feature residuals	5.1 ms
	HD map landmark residuals	13.6 ms
	Linear approximated residuals	0.3 ms
Whole system		71.3 ms

time. For instance, in a highway scenario (e.g., Urban 23), only a few poles in the whole trajectory can provide the longitudinal constraints, resulting in a severe localization drift. Moreover, the system is also sensitive to the incorrect data association of HD map residuals. When the HD map has changed, or the image perception is noisy, the inconsistency between image perception and pre-built HD map might cause the association failure in SCM. Although proposed outlier rejection and initial guess generation strategy can filter out the wrong data association, these strategies might fail in extreme cases when encountering severe perception noise or HD map misalignment. As a result, the wrong data association might ruin the system's performance and result in a bad localization result.

V. CONCLUSIONS

This study proposes TM^{3}Loc to precisely estimate the vehicle poses using a monocular camera and HD map landmarks. In the frontend, the algorithm uses the semantic chamfer matching method that is flexible to model the map-matching cost function for various landmark types and shapes. In the backend, the visual features are incorporated in a tightly-coupled fashion such that the vehicle pose is estimated by jointly optimizing the visual feature constraints and the HD map landmark feature constraints using sliding window-based optimization. Moreover, the linear approximated residual algorithm is introduced to accelerate the optimization, such that the pose estimation can be performed in real-time. The proposed algorithm is evaluated on a large-scale dataset with self-developed HD maps. The results showed that the proposed algorithm could localize the vehicle poses in different scenarios with an excellent accuracy level exceeding that of other methods by a large margin. However, the TM^{3}Loc algorithm is also impacted by the insufficiency of visual features in some extreme corner cases, such as the highway scenarios, and also sensitive to the bad image perception and the severe inconsistency between HD map and detected landmarks. Future works include fusing IMU, GNSS, and odometer sensor data to compensate for the shortcomings of visual features, thereby further improving the robustness and accuracy of the system. In conclusion, the TM^{3}Loc algorithm has been validated as a robust monocular map-matching algorithm for precise vehicle localization that facilitates the low-cost solution for autonomous driving localization.

REFERENCES

[1] Yan Lu, Jiawei Huang, Yi-Ting Chen, and Bernd Heisele. Monocular localization in urban environments using road markings. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 468–474. IEEE, 2017.

[2] Zhongyang Xiao, Kun Jiang, Shichao Xie, Tuopu Wen, Chunlei Yu, and Diange Yang. Monocular vehicle self-localization method based on compact semantic map. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3083–3090. IEEE, 2018.

[3] Jan Hendrik Pauls, Kursat Petek, Fabian Poggenhans, and Christoph Stiller. Monocular localization in HD maps by combining semantic segmentation and distance transform. *IEEE International Conference on Intelligent Robots and Systems*, pages 4595–4601, 2020.

[4] Ashwani Kumar, Takeshi Oishi, Shintaro Ono, Atsuhiko Banno, and Katsushi Ikeuchi. Global coordinate adjustment of 3d survey models in world geodetic system under unstable gps condition. *20th ITS World Congress Tokyo 2013*, 01 2013.

[5] Ashwani Kumar Aggarwal. Machine Vision Based Self-Position Estimation of Mobile Robots. *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, 6(10):20–29, 2015.

[6] Rong Liu, Jinling Wang, and Bingqi Zhang. High Definition Map for Automated Driving : Overview and Analysis. 2019.

[7] Keisuke Yoneda, Hossein Tehrani, Takashi Ogawa, Naohisa Hukuyama, and Seiichi Mita. Lidar scan feature for localization with highly precise 3-d map. In *Intelligent Vehicles Symposium*, 2014.

[8] Guowei Wan, Xiaolong Yang, Renlan Cai, Hao Li, Yao Zhou, Hao Wang, and Shiyu 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)*, pages 4670–4677. IEEE, 2018.

[9] Xingxing Zuo, Patrick Geneva, Yulin Yang, Wenlong Ye, Yong Liu, and Guoquan Huang. Visual-Inertial Localization with Prior LiDAR Map Constraints. *IEEE Robotics and Automation Letters*, 4(4):3394–3401, 2019.

[10] Wendong Ding, Shenhua Hou, Hang Gao, Guowei Wan, and Shiyu Song. Lidar inertial odometry aided robust lidar localization system in changing city scenes. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4322–4328. IEEE, 2020.

[11] Oliver Pink. Visual map matching and localization using a global feature map. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7. IEEE, 2008.

[12] Zui Tao, Ph Bonnifait, Vincent Fremont, and Javier Ibanez-Guzman. Mapping and localization using gps, lane markings and proprioceptive sensors. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 406–412. IEEE, 2013.

[13] Hao Cai, Zhaozheng Hu, Gang Huang, Dunyao Zhu, and Xiaocong Su. Integration of gps, monocular vision, and high definition (hd) map for accurate vehicle localization. *Sensors*, 18(10):3270, 2018.

[14] Youngji Kim, Jinyong Jeong, and Ayoung Kim. Stereo Camera Localization in 3D LiDAR Maps. *IEEE International Conference on Intelligent Robots and Systems*, pages 5826–5833, 2018.

[15] Pengtao Yang, Dongliang Duan, Chen Chen, Xiang Cheng, and Liuqing Yang. Multi-Sensor Multi-Vehicle (MSMV) Localization and Mobility Tracking for Autonomous Driving. *IEEE Transactions on Vehicular Technology*, 69(12):14355–14364, 2020.

[16] Andreas Schindler. Vehicle self-localization with high-precision digital maps. In *2013 IEEE intelligent vehicles symposium workshops (IV Workshops)*, pages 134–139. IEEE, 2013.

[17] Markus Schreiber, Carsten Knöppel, and Uwe Franke. Laneloc: Lane marking based localization using highly accurate maps. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 449–454. IEEE, 2013.

[18] Jinyong Jeong, Younggun Cho, and Ayoung Kim. HDMI-Loc: Exploiting High Definition Map Image for Precise Localization via Bitwise Particle Filter. *IEEE Robotics and Automation Letters*, 5(4):6310–6317, 2020.

[19] Andreas Wedel, Uwe Franke, Hernán Badino, and Daniel Cremers. B-spline modeling of road surfaces for freespace estimation. In *2008 IEEE Intelligent Vehicles Symposium*, pages 828–833. IEEE, 2008.

[20] Wei-Chiu Ma, Ignacio Tartavull, Ioan Andrei Bârsan, Shenlong Wang, Min Bai, Gellert Mattyus, Namdar Homayounfar, Shrinidhi Kowshika Lakshmikanth, Andrei Pokrovsky, and Raquel Urtasun. Exploiting sparse semantic hd maps for self-driving vehicle localization. *arXiv preprint arXiv:1908.03274*, 2019.

[21] Luis R. Ramírez-Hernández, Julio C. Rodríguez-Quiñonez, Moises J. Castro-Toscano, Daniel Hernández-Balbuena, Wendy Flores-Fuentes, Raúl Rascón-Carmona, Lars Lindner, and Oleg Sergiyenko. Improve

three-dimensional point localizatión accuracy in stereo vision systems using a novel camera calibration method. *International Journal of Advanced Robotic Systems*, 17(1):1–15, 2020.

[22] R. P. D. Vivacqua, M. Bertozzi, P. Cerri, F. N. Martins, and R. F. Vassallo. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(2):582–597, 2018.

[23] Wonje Jang, Jhonghyun An, Sangyun Lee, Minho Cho, Myungki Sun, and Euntai Kim. Road Lane Semantic Segmentation for High Definition Map. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June(Iv):1001–1006, 2018.

[24] Liuyuan Deng, Ming Yang, Bing Hu, Tianyi Li, Hao Li, and Chunxiang Wang. Semantic Segmentation-Based Lane-Level Localization Using Around View Monitoring System. *IEEE Sensors Journal*, 19(21):10077–10086, 2019.

[25] Wonje Jang, Junhyuk Hyun, Jhonghyun An, Minho Cho, and Euntai Kim. A lane-level road marking map using a monocular camera. *IEEE/CAA Journal of Automatica Sinica*, 9(1):187–204, 2022.

[26] Tuopu Wen, Zhongyang Xiao, Benny Wijaya, Kun Jiang, Mengmeng Yang, and Diange Yang. High Precision Vehicle Localization based on Tightly-coupled Visual Odometry and Vector HD Map. *IEEE Intelligent Vehicles Symposium, Proceedings*, (Iv):672–679, 2020.

[27] Noa Garnett, Rafi Cohen, Tomer Pe'Er, Roee Lahav, and Dan Levi. 3D-LaneNet: End-to-end 3D multiple lane detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:2921–2930, 2019.

[28] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-LaneNet: A Generalized and Scalable Approach for 3D Lane Detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12366 LNCS:666–681, 2020.

[29] Ziwei Liao, Jieqi Shi, Xianyu Qi, Xiaoyu Zhang, Wei Wang, Yijia He, Ran Wei, and Xiao Liu. Coarse-to-fine visual localization using semantic compact map. *arXiv preprint arXiv:1910.04936*, 2019.

[30] Andre Welzel, Pierre Reisdorf, and Gerd Wanielik. Improving Urban Vehicle Localization with Traffic Sign Recognition. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015-Octob:2728–2732, 2015.

[31] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 International Conference on Computer Vision, ICCV 2015:2938–2946, 2015.

[32] Peng Wang, Ruigang Yang, Binbin Cao, Wei Xu, and Yuanqing Lin. Dels-3d: Deep localization and segmentation with a 3d semantic map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5860–5869, 2018.

[33] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[34] Yao Zhou, Guowei Wan, Shenhua Hou, Li Yu, Gang Wang, Xiaofei Rui, and Shiyu Song. Da4ad: End-to-end deep attention-based visual localization for autonomous driving. In *European Conference on Computer Vision*, pages 271–289. Springer, 2020.

[35] Jesse Levinson and Sebastian Thrun. Automatic Online Calibration of Cameras and Lasers. 2016.

[36] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research*, page 0278364919843996, 2019.

[37] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016.

[38] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[39] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[40] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[41] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[42] Jinyong Jeong, Younggun Cho, and Ayoung Kim. Road-SLAM : Road marking based SLAM with lane-level accuracy. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1736–1743, 2017.

[43] Julius Ziegler, Henning Lategahn, Markus Schreiber, Christoph G Keller, Carsten Knöppel, Jochen Hipp, Martin Haueis, and Christoph Stiller. Video based localization for bertha. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1231–1238. IEEE, 2014.

[44] Zhongyang Xiao, Diange Yang, Tuopu Wen, Kun Jiang, and Ruidong Yan. Monocular localization with vector hd map (mlvhm): A low-cost method for commercial ivs. *Sensors*, 20(7):1870, 2020.

[45] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[46] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[47] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

[48] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 286–291, 2018.

[49] Tuopu Wen, Diange Yang, Kun Jiang, Chunlei Yu, Jiaxin Lin, Benny Wijaya, and Xinyu Jiao. Bridging the gap of lane detection performance between different datasets: Unified viewpoint transformation. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[50] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *IJCAI*, 1977.

[51] Gabe Sibley, Larry Matthies, and Gaurav Sukhatme. A sliding window filter for incremental slam. In *Unifying perspectives in computational and robot vision*, pages 103–112. Springer, 2008.

[52] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[53] Jianbo Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[54] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc 7th Intl Joint Conf on Artificial Intelligence (IJ CAI)*, 1981.

[55] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. Openvins: A research platform for visual-inertial estimation. In *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.

Tuopu Wen received the B.S. degree from Electronic Engineering, Tsinghua University, Beijing, China in 2018. He is currently working toward the Ph.D. degree at the School of Vehicle and Mobility of Tsinghua University, Beijing, China. His research interests include computer vision, high definition map, and high precision localization for autonomous driving.

Kun JIANG received the B.S. degree in mechanical and automation engineering from Shanghai Jiao Tong University, Shanghai, China in 2011. Then he received the Master degree in mechatronics system and the Ph.D. degree in information and systems technologies from University of Technology of Compiègne (UTC), Compiègne, France, in 2013 and 2016, respectively. He is currently an assistant research professor at Tsinghua University, Beijing, China. His research interests include autonomous vehicles, high precision digital map, and sensor fusion.

Benny Wijaya received the B.S. degree in mechanical engineering from University of Newcastle, Newcastle, Australia in 2015. Then he received the Master degree in mechanical engineering from Tsinghua University, Beijing, China in 2020. He is currently working toward the Ph.D. degree at the School of Vehicle and Mobility in Tsinghua University, Beijing, China. His research interests include, sensor fusion, confidence modelling, and high definition map for autonomous driving.

Hangyu Li received the B.S. degree from Tsinghua University, Beijing, China in 2021 and is now a Master of Philosophy student in Hong Kong University of Science and Technology. His research interests mainly focus on high precision localization and evaluation of autonomous vehicles.

Mengmeng Yang received the Ph.D. degree in Photogrammetry and Remote Sensing from Wuhan University, Wuhan, China in 2018. She is currently an assistant research professor at Tsinghua University, Beijing, China. Her research interests include autonomous vehicles, high precision digital map, and sensor fusion.

Diange Yang is a professor at the School of Vehicle and Mobility of Tsinghua University. He received his B.S. and Ph.D. from Tsinghua University in 1996 and 2001. Now, he is the Dean of the School of Vehicle and Mobility and his research work mainly focuses on Intelligent Connected Vehicles and Autonomous Driving. Professor Yang has authored 12 software copyrights and registered more than 60 national patents, he also published 120 papers. He has received numerous awards during his career, including the Distinguished Young Science Technology talent of Chinese Automobile Industry in 2011, the Excellent Young Scientist of Beijing in 2010. He is also the recipient of the Second Prize of National Technology Invention Rewards of CHINA in 2010 and in 2013.