2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Increasing SLAM Pose Accuracy by Ground-to-Satellite Image Registration

Yanhao Zhang[1(✉)], Yujiao Shi[2], Shan Wang[3,4], Ankit Vora[5], Akhil Perincherry[5], Yongbo Chen[3], and Hongdong Li[3]

***Abstract*— Vision-based localization for autonomous driving has been of great interest among researchers. When a pre-built 3D map is not available, the techniques of visual simultaneous localization and mapping (SLAM) are typically adopted. Due to error accumulation, visual SLAM (vSLAM) usually suffers from long-term drift. This paper proposes a framework to increase the localization accuracy by fusing the vSLAM with a deep-learning based ground-to-satellite (G2S) image registration method. In this framework, a coarse (spatial correlation bound check) to fine (visual odometry consistency check) method is designed to select the valid G2S prediction. The selected prediction is then fused with the SLAM measurement by solving a scaled pose graph problem. To further increase the localization accuracy, we provide an iterative trajectory fusion pipeline. The proposed framework is evaluated on two well-known autonomous driving datasets, and the results demonstrate the accuracy and robustness in terms of vehicle localization. The code will be available at `https://github.com/YanhaoZhang/SLAM-G2S-Fusion`.**

***Index Terms*— visual SLAM, cross-view localization, autonomous driving.**

I. Introduction

Accurate localization is an essential task for autonomous driving. Although GPS is widely used in people's daily lives, the accuracy of a consumer-grade GPS device can degrade rapidly in GPS-compromised areas [1], e.g., the urban areas with high-rising buildings, which does not meet the requirements for autonomous driving [2]. Alternatively, other techniques adopts a pre-rendered 3D high-definition (HD) map for vehicle re-localization [3], [4], [5], [6]. However, it is laborious and expensive to reconstruct and maintain such an HD map, and a pre-built map only supports the vehicle's re-localization. Therefore, the study on self-localization techniques using only on-board sensors for autonomous driving is of great interest among researchers.

Simultaneous localization and mapping (SLAM) is one of the major topics in robotics for sensor self-localization. Among different sensors, cameras are usually cheaper while capturing richer information. The task of visual SLAM (vSLAM) is to build and update a 3D map while simultaneously estimating the camera's trajectory, using the 2D features extracted from the consecutive input images. Since SLAM results are based on the consecutive observation of the same scene, without any global information, the observation error accumulates over time resulting in long-term drift. This is especially a problem for autonomous driving when a vehicle moves from one place to another without any loops.

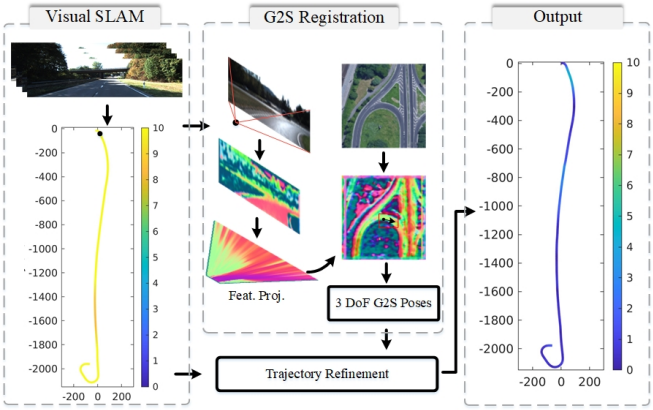

Fig. 1. The proposed framework fuses vSLAM with G2S registration and estimates the camera trajectory with high accuracy. The inputs are poses from stereo SLAM, ground-view images, and satellite images, the output is an updated vehicle trajectory. The example shows the localization error using the colour map (unit: m).

Recently, the problem of ground-to-satellite (G2S) registration has aroused attention in academics. It aims to estimate the 3-DoF pose of a ground-view image w.r.t. a satellite image. Compared to the localization methods relying on an HD map, the cross-view based solution utilizes relatively cheaper and more memory-efficient satellite images. The satellite images can also provide global information for the camera pose reference, making it a preferable option to eliminate the long-term drift from SLAM estimation.

This paper proposes a novel framework for camera pose estimation that combines the merits of vSLAM and G2S registration (Fig. 1). vSLAM experiences long-term drift issues while achieving acceptable accuracy in the short term. In contrast, G2S registration minimizes error accumulation (it concentrates on pose estimation between two views), while the predicted translation, especially along the longitudinal axis, is not robust enough as shown in [7]. Our purpose is to fuse the two methods together and provide a more accurate localization framework for autonomous driving. Particularly, we want to use the pose estimated by G2S registration to

[1]Yanhao Zhang is with the Robotics Institute, University of Technology Sydney, Sydney, Australia (e-mail: yanhao.zhang@uts.edu.au).
[2] Yujiao Shi is with ShanghaiTech University, Shanghai, China (e-mail: shiyj2@shanghaitech.edu.cn).
[3] Shan Wang, Yongbo Chen, and Hongdong Li are with the College of Engineering and Computer Science, Australian National University, Canberra, Australia (e-mails: shan.wang@anu.edu.au; yongbo.chen@anu.edu.au; hongdong.li@anu.edu.au).
[4]Shan Wang is also with Data61, CSIRO, Canberra, Australia.
[5]Ankit Vora and Akhil Perincherry are with Ford Motor Company, Dearborn, USA (e-mails: avora3@ford.com; aperinch@ford.com).
This work was performed while Yanhao Zhang and Yujiao Shi worked at the Australian National University.

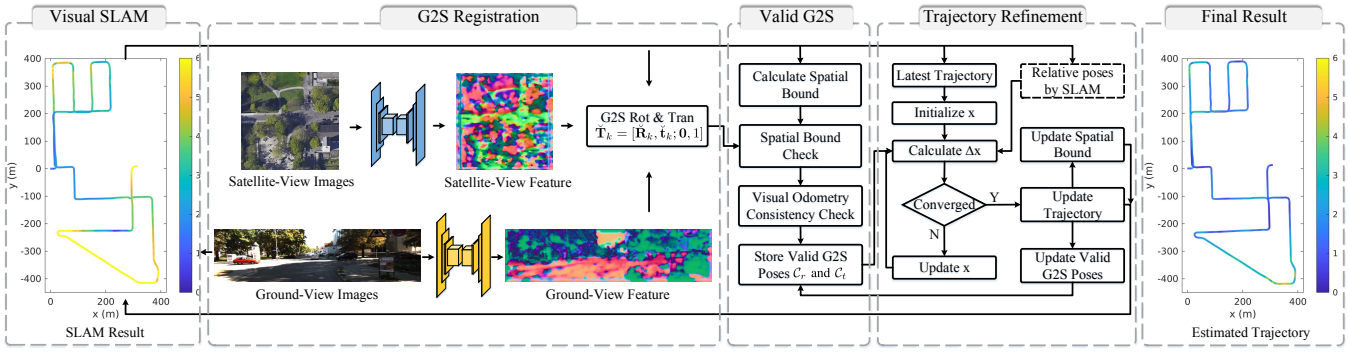

Fig. 2. A flowchart showing the main processes of the proposed framework. For each vehicle pose, we calculate the G2S prediction $\breve{\mathbf{T}}$ using the ground-view and the corresponding satellite images. The valid predictions are selected via a coarse-to-fine procedure, and are fused with the relative poses (from the original SLAM trajectory) by solving a scaled pose graph optimization. The localization error is shown using the colour map (unit: m).

rectify the pose estimated by SLAM.

Considering the pose estimated by G2S registration is not always more accurate than SLAM, we design a coarse-to-fine mechanism to select the valid G2S poses, which is different from the existing methods [8], [9]. Specifically, we first coarsely select the G2S poses based on a spatial bound determined by the trajectory uncertainty, which are then further refined by checking the visual odometry consistency. After this coarse-to-fine filtering, the selected G2S poses are considered valid and fused with the SLAM poses by solving a scaled pose graph problem. To increase the localization accuracy, we update the trajectory iteratively when a G2S pose is considered valid. The proposed framework is evaluated on two well-known autonomous driving datasets, and the results demonstrate that our method achieves around 68%-80% improvement in translation estimation and 45%-65% on rotation estimation compared to the original vSLAM.

The main contributions of this paper are as follows:

- a new localization framework that combines the merits of vSLAM and G2S image registration;
- a coarse-to-fine method to remove false G2S results;
- an iterative trajectory refinement pipeline that fuses the measurements by solving a scaled pose graph problem.

II. Related Work

A. Visual Localization

There has been intensive research on visual localization methods for autonomous driving. Among these methods, vSLAM has traditionally been studied, e.g., [10], [11], [12], [13], [14], [15], [16], [17], where ORB-SLAM3 [10] is the SOTA vSLAM system consisting of visual tracking, local mapping, and loop closure. Deep learning based localization has also been investigated to handle large appearance changes between the query images and the reference maps [18], [19], [20]. The main idea is to train a neural network and use the implicitly calculated correspondence (by the deep features between a query image and a reference image) for camera pose estimation. Similar ideas are also shown to be effective for G2S registration [21], [22].

B. G2S-based Visual Localization

The early works on G2S registration solve an image retrieval problem and achieve a coarse localization [23], [24], [25], [26], [27]. Later, fine-grained methods are proposed to estimate the relative pose from a coarse input pose. To calculate the G2S correspondences, some methods use an additional sensor, e.g., LiDAR, [22], [28], while others vision-only methods calculate feature matches using homography transformations [21], [29], [9] or sparse view-consistent keypoints from a pre-trained network [30]. Recent research has shown that decoupling the estimated rotation and translation can facilitate the overall registration performance [7], [31]. BoostG2SLoc [7] is the SOTA pipeline where a neural pose optimizer is deployed to estimate the azimuth orientation, and a spatial correlation module is used to predict the longitudinal and lateral translation.

III. Problem Description

Given a coarse pose between a paired ground-view and satellite images $\{I^g, I^s\}$, a deep neural network is adopted to predict the relative transformation between I^g and I^s. To be more specific, the output of the G2S prediction is the changes of rotation and translation w.r.t. the input pose. Let $\tilde{\mathbf{T}}_k = [\tilde{\mathbf{R}}_k, \tilde{\mathbf{t}}_k; \mathbf{0}, 1]$ denote a (input) SLAM pose at timestamp k[1]. Using $\tilde{\mathbf{T}}_k$ and the paired images $\{I^g_k, I^s_k\}$, we predict $\{\breve{x}_k, \breve{y}_k, \breve{\theta}_k\}$, indicating the G2S translation shifts (longitudinal and latitudinal) and the azimuth change w.r.t. $\tilde{\mathbf{T}}_k$[2]. Let $\mathbf{T}_k = [\mathbf{R}_k, \mathbf{t}_k; \mathbf{0}, 1]$ denote the updated poses, in the ideal case without noise, we have:

$$\breve{\mathbf{R}}_k^\top = \mathbf{R}_k^\top \cdot \tilde{\mathbf{R}}_k, \quad \breve{\mathbf{t}}_k = \tilde{\mathbf{R}}_k^\top \cdot (\mathbf{t}_k - \tilde{\mathbf{t}}_k) \tag{1}$$

where $\breve{\mathbf{T}}_k = [\breve{\mathbf{R}}_k, \breve{\mathbf{t}}_k; \mathbf{0}, 1]$ represents the G2S pose, $\breve{\mathbf{R}}_k = \exp([\breve{\mathbf{r}}_k]^\wedge)$, $\breve{\mathbf{r}}_k = [0, 0, \breve{\theta}_k]^\top$, $\breve{\mathbf{t}}_k = [\breve{x}_k, \breve{y}_k, 0]^\top$. Here, we do not update the translation along z-axis and the orientation around x, y-axis[3].

IV. Methodology

This section introduces the details of the proposed framework. It consists of three main parts: deep learning based G2S registration, valid G2S pose selection, and the G2S-SLAM fusion. The main processes are outlined in Fig. 2.

[1]In this paper, we use the diacritics $\tilde{\cdot}$ and $\breve{\cdot}$ to represent the poses from SLAM and G2S prediction, respectively.

[2]Here, we assume $\{I^g_0, I^s_0\}$ are aligned and set $\{x_0, y_0, \theta_0\}$ as $\{0, 0, 0\}$ for the first frame.

[3]Owing to the (approximated) parallel projection of satellite images, the G2S registration focuses on a 3-DoF pose estimation.

A. G2S Registration

The G2S registration is based on BoostG2SLoc [7]. From I^g, a U-Net architecture is adopted to extract the deep features F^g which are then projected to the I^s domain to synthesize an overhead view feature map F^{g2s}. Similarly, the deep features F^s are extracted from I^s. The deep features F^{g2s} and F^s enable the network to implicitly learn the correspondence between I^g and I^s for pose prediction.

The camera poses are estimated in two steps. First, based on the deep feature F^{g2s}_{init} (synthesized using the input pose) and F^s, a neural pose optimizer calculates the azimuth change $\breve{\theta}$ w.r.t. the input orientation. Second, the translation shift is calculated using an uncertainty-guided spatial correlation method. To be more specific, using the predicted $\breve{\theta}$, an overhead view feature map $F^{g2s}_{\breve{\theta}}$ is re-synthesized and used as a sliding window to compute its spatial correlation with F^s. A deep uncertainty map is adopted to exclude the infeasible vehicle locations, e.g., building roof and tree areas.

B. Valid G2S Pose Selection

Since the satellite image does not provide 3D information, the valid cross-view features are mainly extracted from the ground, e.g., road marks, lanes, etc. When a vehicle moves forward, the feature changes along the longitudinal direction can be small. In other words, the translation changes on the input signals can be absorbed by the aggregation layers, and hence the predicted translation (especially along the longitudinal axis) can be noisy. To tackle this issue, this paper designs a coarse-to-fine method to remove the false G2S predictions by utilizing the SLAM information.

1) Spatial Bound Check: BoostG2SLoc searches the G2S translation via a fixed-range spatial correlation. On the one hand, a large search range is needed such that the network is able to handle the long-term SLAM drift when the input pose is far from its ground truth. On the other hand, the large search range increases the chance of involving false deep features similar to the true one on F^{g2s}_{θ}, which results in false G2S predictions. These false predictions need to be removed before combining with the SLAM measurements. To achieve this, we calculate a spatial bound proportional to the 3-σ bound of the estimated trajectory covariance[4]. When the input pose has low uncertainty, it is assumed to be close to the ground truth; therefore, a large G2S translation would be regarded as an incorrect prediction. At timestamp k, suppose $\mathbf{\Phi}_k \in \mathbb{R}^{2\times 2}$ is the estimated covariance of the x-y translation, the spatial bound is calculated based on [32]:

$$\mathbf{b}_k(\alpha) = \frac{3}{n}\begin{bmatrix}\cos(\Theta(\mathbf{R}_k)) & -\sin(\Theta(\mathbf{R}_k))\\ \sin(\Theta(\mathbf{R}_k)) & \cos(\Theta(\mathbf{R}_k))\end{bmatrix}\mathbf{\Phi}_k^{1/2}\begin{bmatrix}\cos(\alpha)\\ \sin(\alpha)\end{bmatrix} \quad (2)$$

where $\Theta(\cdot)$ returns the azimuth angle from a rotation matrix, $\alpha \in [0, 2\pi)$, n is the scale factor of the bound. Since the estimate by SLAM is usually accurate at the beginning, we assume the spatial bound $\mathbf{b}_1$ is within a fixed range r, and calculate a scale factor using $n = \text{mean}(\lambda_1, \lambda_2)/r$ where λ_1, λ_2 are the two eigenvalues of $\mathbf{\Phi}_1^{1/2}$. Let $\mathcal{B}_k$ denote the 2D space within the spatial bound, we have $\breve{\mathbf{t}}_k \in \mathcal{B}_k$ meaning the predicted translation is valid w.r.t. the spatial bound check.

2) Visual Odometry Consistency: Although the spatial bound check can remove false G2S predictions inconsistent with the estimated uncertainty, it can not handle the case where false predictions are within the spatial bound[5]. To tackle this, we further check the visual odometry consistency between the G2S predictions and the input poses, according to that the SLAM estimation is usually with acceptable accuracy in the short term. Based on (1), the relative pose by cross-view is $\breve{\mathbf{T}}_{k-1,k} = \breve{\mathbf{T}}_{k-1}^{-1}\mathbf{T}_{k-1,k}\breve{\mathbf{T}}_k$, where $\mathbf{T}_{k-1,k} = \mathbf{T}_{k-1}^{-1}\mathbf{T}_k$ is the relative pose from the input trajectory. If $\breve{\mathbf{t}}_{k-1} \in \mathcal{B}_{k-1}$ and $\breve{\mathbf{t}}_k \in \mathcal{B}_k$, we respectively check if the rotation difference (azimuth angle) and the translation difference are smaller than a threshold. Let $\mathcal{C}_r$ and $\mathcal{C}_t$ denote the selected G2S orientation and translation, we have

$$\begin{aligned}\mathcal{C}_r &= \{\breve{\mathbf{R}}_k : \ \ |\Theta(\breve{\mathbf{R}}_{k-1,k}\cdot\mathbf{R}_{k-1,k}^\top)| < \text{th}_\theta\}\\ \mathcal{C}_t &= \{\breve{\mathbf{t}}_k : \ \ |\mathbf{e}_i^\top\cdot(\breve{\mathbf{t}}_{k-1,k} - \mathbf{t}_{k-1,k})| < \text{th}_t\}\end{aligned} \quad (3)$$

where $\mathbf{e}_1 = [1,0,0]^\top$ and $\mathbf{e}_2 = [0,1,0]^\top$ are for checking the translation consistency along x and y axis, respectively.

Algorithm 1: Iterative G2S-SLAM Fusion

Input: SLAM poses, ground and satellite images.
Output: Estimated vehicle trajectory.

Initialize poses $\mathbf{T}_k^0 = \tilde{\mathbf{T}}_k$;
Initialize spatial bound $\mathcal{B}_k^0$;
Initialize G2S prediction for the first frame: $\breve{\mathbf{T}}_0^0 = \mathbf{I}_4$;
Calculate visual odometry weights by Sec. IV-C.2;
for *all other poses* **do**
  Step 1: G2S Pose Prediction:
  Calculate G2S pose $\breve{\mathbf{T}}_k^t$ using the trajectory pose $\mathbf{T}_k^t$ and the images $\{I_k^g, I_k^s\}$; ▷ 't' denotes the latest updated trajectory. $t = 0$ at the beginning.
  Step 2: Check validity:
  Calculate spatial bound $\mathcal{B}_{k-1}^t$ and $\mathcal{B}_k^t$;
  if $\breve{\mathbf{t}}_{k-1}^t \in \mathcal{B}_{k-1}^t$ & $\breve{\mathbf{t}}_k^t \in \mathcal{B}_k^t$ **then**
    Calculate relative pose $\breve{\mathbf{T}}_{k-1,k}^t$ and $\mathbf{T}_{k-1,k}^t$;
    Visual odometry consistency check (3);
  end
  Step 3: Trajectory Refinement:
  if $\breve{\mathbf{T}}_k^t$ *is selected* **then**
    Solve the nonlinear least squares problem (4);
    Update the spatial bound $\mathcal{B}_{t+1}^{t+1}, \cdots$; ▷ For checking the rest G2S predictions.
    Update all selected predictions $\mathcal{C}_r^{t+1}$, $\mathcal{C}_t^{t+1}$. ▷ Making the selected predictions w.r.t. the latest trajectory for the next refinement.
  end
end

C. Scaled Pose Graph based G2S-SLAM Fusion

1) Objective Function: The vehicle trajectory is estimated by fusing the SLAM poses and the selected G2S predictions.

[4]The trajectory estimation is illustrated in Sec. IV-C.4. The covariance is from the information matrix at the solution of (4). To be more specific, we calculate the covariance matrix using the inverse of the Hessian matrix. $\mathbf{\Phi}_k$ is then directly obtained from the covariance matrix.

[5]This can happen either because of the inaccuracy of the estimated uncertainty or because of multiple false deep features within a spatial bound.

For vSLAM, since solving a global Bundle Adjustment can be very expensive owing to the large number of feature points, a pose-graph is typically used for trajectory refinement, e.g., the loop closure refinement [33]. In this paper, we present a similar idea for G2S SLAM fusion, by solving a scaled pose graph problem.

Suppose $\{\tilde{\mathbf{R}}_{i,j}, \tilde{\mathbf{t}}_{i,j}\}$ denotes a relative pose of visual odometry or loop closure from SLAM. $\tilde{\mathcal{V}}_r$ and $\tilde{\mathcal{V}}_t$ are the set of all SLAM relative rotations and translations, respectively. The vehicle poses $\{\mathbf{R}_k, \mathbf{t}_k\}$ are calculated by

$$\begin{aligned}\underset{\{\cdots,\mathbf{R}_k,\mathbf{t}_k,s_k,\cdots\}}{\arg\min} & \sum_{\tilde{\mathbf{R}}_{i,j}\in\tilde{\mathcal{V}}_r} \|\tilde{w}_{i,j}[\log(\tilde{\mathbf{R}}_{i,j}\cdot\mathbf{R}_j^\top\cdot\mathbf{R}_i)]^\vee\|^2_{\tilde{\mathbf{\Sigma}}_r} \\ &+ \sum_{\tilde{\mathbf{t}}_{i,j}\in\tilde{\mathcal{V}}_t} \|\tilde{w}_{i,j}(\tilde{\mathbf{t}}_{i,j} - s_j\mathbf{R}_i^\top(\mathbf{t}_j-\mathbf{t}_i))\|^2_{\tilde{\mathbf{\Sigma}}_t} \\ &+ \sum_{\breve{\mathbf{R}}_l\in\breve{\mathcal{C}}_r} \|[\log(\mathbf{R}_l^\top\cdot\tilde{\mathbf{R}}_l\cdot\breve{\mathbf{R}}_l^\top)]^\vee\|^2_{\breve{\mathbf{\Sigma}}_r} \\ &+ \sum_{\breve{\mathbf{t}}_l\in\breve{\mathcal{C}}_t} \rho(\|(\breve{\mathbf{t}}_l - \tilde{\mathbf{R}}_l^\top(\mathbf{t}_l-\tilde{\mathbf{t}}_l))\|^2_{\breve{\mathbf{\Sigma}}_t}) \\ &+ \sum_{k=1}^{K}\|s_k - s_{k-1}\|^2_{\sigma_s}\end{aligned} \tag{4}$$

where $\log : \mathrm{SO}(3) \to \mathfrak{so}(3)$ is the logarithm map and $[\cdot]^\vee$ returns the vector elements from a skew-symmetric matrix. $\tilde{\mathbf{\Sigma}}_r = \tilde{\sigma}_r\mathbf{I}_3$, $\tilde{\mathbf{\Sigma}}_t = \tilde{\sigma}_t\mathbf{I}_3$, $\breve{\mathbf{\Sigma}}_r = \breve{\sigma}_r\mathbf{I}_3$, $\breve{\mathbf{\Sigma}}_t = \mathrm{diag}(\breve{\sigma}_t^x, \breve{\sigma}_t^y, 0)$, and σ_s are the hyper-parameters to balance the terms.

In (4), the first two terms represent the measurements by SLAM, where we also estimate a scale s_k for each pose to reflect the trajectory drift of vSLAM[6]. Since the SLAM drift is usually smooth, we restrict the scale change among the neighbouring frames using the fifth term.

The third and the fourth terms represent the measurements from G2S predictions based on (1). For the G2S translation term, we assign $\breve{\sigma}_t^x < \breve{\sigma}_t^y$ since the latitudinal predictions are usually with higher accuracy than the longitudinal predictions. A Huber kernel is used to bring more robustness:

$$\rho(x) = \begin{cases} x^2/2 & |x| < c \\ c\cdot(|x| - c/2) & |x| \geqslant c \end{cases} \tag{5}$$

2) Visual Odometry Weights: We follow [34] to calculate the visual odometry weights. Suppose $N_{i,j}$ denotes the number of covisible features between two frames i and j, $\tilde{w}_{i,j} = \sqrt{N_{i,j}}/\tilde{n}_{i,j}$, where the factor $\tilde{n}_{i,j} = \mathrm{mean}(\cdots, \sqrt{N_{i,j}}, \cdots)$ is to make the weights stable among different dataset.

3) Optimization: The non-linear least squares problem (4) can be solved iteratively using the Gauss-Newton method, where the robust kernel can be represented by reweighting the measurement term [35]. Suppose $\mathbf{x}$ denotes the concatenation of all state variables, $\mathbf{W}$ denotes the stacked weights. In each iteration, the solver linearises the problem at $\mathbf{x}$ by $\mathbf{J}(\mathbf{x})$ (the stacked Jacobian matrix), and calculates the step change to update $\mathbf{x}$:

$$\Delta\mathbf{x} = -(\mathbf{J}(\mathbf{x})^\top\cdot\mathbf{W}\cdot\mathbf{J}(\mathbf{x}))^{-1}\cdot(\mathbf{J}(\mathbf{x})^\top\cdot\mathbf{W}\cdot\mathbf{f}(\mathbf{x})) \tag{6}$$

where $\mathbf{f}(\mathbf{x})$ is the stacked residual. At the solution point $\mathbf{x}^*$, we can calculate the estimated covariance matrix $(\mathbf{J}(\mathbf{x}^*)^\top\cdot\mathbf{W}\cdot\mathbf{J}(\mathbf{x}^*))^{-1}$ for the spatial bound check in Sec. IV-B.1.

4) Iterative Fusion Pipeline: The accuracy of G2S prediction decreases when the trajectory drift becomes out of the search range by BoostG2SLoc. To tackle this issue, we provide an iterative trajectory fusion pipeline. At the beginning, we initialize the vehicle poses $\mathbf{T}_k^0$ using SLAM trajectory. The spatial bound $\mathcal{B}_k^0$ is also initialized using the covariance matrix[7]. Suppose at timestamp t, the predicted G2S pose is selected. We update the trajectory using all the current selected G2S measurements and the results from SLAM by solving (4). Having the new trajectory $\mathbf{T}_k^{t+1}$ ($k = \{1, \cdots, K\}$), we can calculate the spatial bounds $\mathcal{B}_{t+1}^{t+1}, \cdots, \mathcal{B}_K^{t+1}$ (only the spatial bounds for the following poses are needed for the next spatial bound check). We also update the G2S measurements $\mathcal{C}_r^{t+1}$ and $\mathcal{C}_t^{t+1}$ (all selected G2S measurements are used for the next update). More details are shown in Algorithm 1.

V. Experiments and Results

In this section, the evaluations using real datasets are presented. We first introduce the details of data preparation, network referencing, and hyper-parameter setting. The experiment results are then presented followed by an ablation study showing the effectiveness of each module in the framework. For all experiments, we use the stereo SLAM trajectories by ORB-SLAM3 [10] and G2S predictions by BoostG2SLoc [7]. However, we should note that our method is not limited to any specific SLAM or cross-view registration framework.

A. Experimental Setup

1) Dataset Preparation: The framework is evaluated using two publicly available autonomous driving datasets KITTI [36] and FordAV [37]. The satellite images are collected from Google Map [38]. Each satellite image covers a region around $100m\times100m$ with a resolution of 512×512. The ground-view images are re-sized to 256×1024 for network referencing. More details are shown in [21].

2) G2S Registration: The G2S registration is performed by loading the weights pre-trained on the KITTI and FordAV from [7][8]. For all experiments, we set the G2S search range for rotation as $\pm10°$, and the location search range as $20m \times 20m$. We use the same-area G2S model for KITTI and FordAV, respectively.

3) Parameter Setup: We set $r = 0.01$, $\mathrm{th}_\theta = 0.25$, $\mathrm{th}_t = 0.5$ in (3); $\tilde{\sigma}_r = 0.85^2$, $\tilde{\sigma}_t = 0.9^2$, $\breve{\sigma}_r = 1$, $\breve{\sigma}_t^x = 0.003^2$, $\breve{\sigma}_t^y = 0.005^2$, $\sigma_s = 10^2$ in (4) and $c = 1$ in (5) on KITTI. For the experiments on FordAV, we set $r = 0.2$, $\mathrm{th}_\theta = 0.5$, $\breve{\sigma}_t^x = 0.001^2$, $\sigma_s = 9^2$, $c = 6$, and keep the rest hyper-parameters unchanged. The parameters for different trajectories in each dataset are the same.

[6]A similar idea is shown in [33] for monocular SLAM. We find that even for stereo SLAM where the scale is observable, recovering the scale for each pose is helpful to reduce the trajectory error in our framework.

[7]Using the SLAM poses, we calculate the Hessian matrix of (4) without the cross-view measurements

[8]All experiments are conducted on a desktop with the Intel(R) Core(TM) i7-13700KF CPU and the GeForce RTX 3090 GPU.

TABLE I

ACCURACY COMPARISON WITH SLAM RESULT USING KITTI DATASET

Sequence	$\mathbf{S}$ by Trajectory Origin						$\mathbf{S}$ by Multiple Ground Truth					
	$\theta^{\dagger}\downarrow$	$\theta^{\S}\downarrow$	$\theta^{\pounds}\%\uparrow$	$\mathbf{t}^{\dagger}\downarrow$	$\mathbf{t}^{\S}\downarrow$	$\mathbf{t}^{\pounds}\%\uparrow$	$\theta^{\dagger}\downarrow$	$\theta^{\S}\downarrow$	$\theta^{\pounds}\%\uparrow$	$\mathbf{t}^{\dagger}\downarrow$	$\mathbf{t}^{\S}\downarrow$	$\mathbf{t}^{\pounds}\%\uparrow$
00	0.731	0.491	32.7 %	4.144	0.946	77.2 %	0.545	0.545	-0.2%	1.081	1.306	-20.8%
01	2.644	0.726	72.6 %	31.978	1.516	95.3%	1.733	0.717	58.6%	15.00	1.529	89.8%
02	1.001	0.211	78.9 %	5.650	0.823	85.4%	0.547	0.214	61.0%	3.661	0.802	78.1%
04	0.037	0.188	-409.8%	0.625	0.363	41.9%	0.413	0.088	78.6%	0.193	0.344	-78.8%
05	0.304	0.231	23.9%	1.292	0.543	58.0%	0.257	0.288	-12.1%	0.922	0.435	52.8%
06	0.832	0.430	48.3%	2.657	1.311	50.6%	0.426	0.367	14.0%	0.912	0.768	15.7%
07	0.291	0.203	30.4%	0.640	0.512	20.0%	0.296	0.278	6.2%	0.419	0.355	15.3%
08	1.990	0.466	76.6%	7.460	1.168	84.4%	1.038	0.523	49.6%	4.323	2.073	52.1%
09	0.949	0.205	78.4%	3.080	1.553	49.6%	0.779	0.286	63.3%	1.903	1.254	34.1%
10	0.784	0.146	81.3%	3.538	0.646	81.7%	0.461	0.190	58.7%	0.906	0.509	43.8%
Avg.	0.956	**0.330**	65.5%	6.106	**0.938**	84.6%	0.650	**0.350**	46.2%	2.932	**0.938**	68.0%

θ: RMSE of absolute azimuth rotation (unit: °); $\mathbf{t}$ RMSE of absolute 2D translation (unit: m);
†: The stereo SLAM result using [10]; §: the result by the proposed method;
$\pounds$: the accuracy improvement using $\frac{\text{SLAM error}-\text{our method error}}{\text{SLAM error}}$;
'↓': smaller error represents higher accuracy; '↑': higher percentage represents larger improvement.

B. Evaluation Metrics

For autonomous driving scenarios, the localization accuracy on the x-y plane is of more concern. This paper adopts two metrics to evaluate the trajectory 2D accuracy.

The absolute mean, median, and root-mean-square error (RMSE) of rotation and translation are used to measure the error between the estimated trajectory and the ground truth[9]:

$$\delta\mathbf{T} = \check{\mathbf{T}}^{-1}\mathbf{S}\mathbf{T} \tag{7}$$

where $\mathbf{T}$ and $\check{\mathbf{T}}$ are the estimated poses and the ground truth. $\mathbf{S}$ is the rigid-body transformation mapping the estimated trajectory to the ground truth [39]. $\mathbf{S}$ is obtained either using the ground truth pose at the trajectory origin, or using multiple ground truth poses by solving a least-squares problem [40].

Following [7] and [21], we also present the translation error along the longitudinal and lateral directions, and the rotation error of azimuth orientation. Specifically, we show the percentage of the correctly estimated translation and rotation within 1 meter and 1°.

C. Assessment on G2S Pose Selection

The valid cross-view selection is important to increase the accuracy of trajectory estimation. To evaluate the proposed G2S pose selection module, we report the error distribution of the raw predictions and the selected G2S poses in Fig. 3. Overall, the selected poses are with smaller errors, indicating that most false G2S predictions are effectively removed.

D. Assessment on Trajectory Estimation

Table I reports a comparison of the estimated rotation and translation RMSE using the stereo SLAM and the proposed framework. To fully reflect the accuracy, we calculate $\mathbf{S}$ in (7) using the ground truth at trajectory origin (better reflecting the true estimate as minimum ground truth information is involved) and multiple ground truth poses (common among literature)[10]. Overall, the proposed framework achieves higher accuracy for vehicle localization. On average, the translation error reduces 64% (from 0.96° to

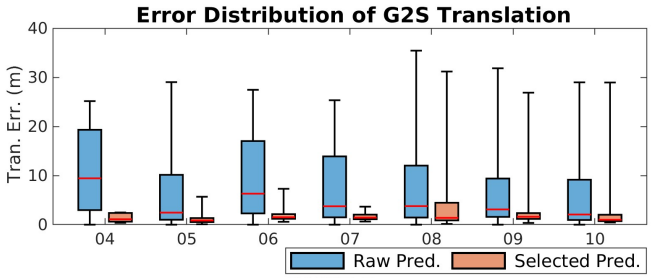

Fig. 3. Error distribution (unit m) of the raw G2S predictions and that of the selected predictions. Here, we use seven sequences ('04'-'10') from KITTI Odometry Benchmark for evaluation. We do not update the trajectory to avoid the effect from other modules.

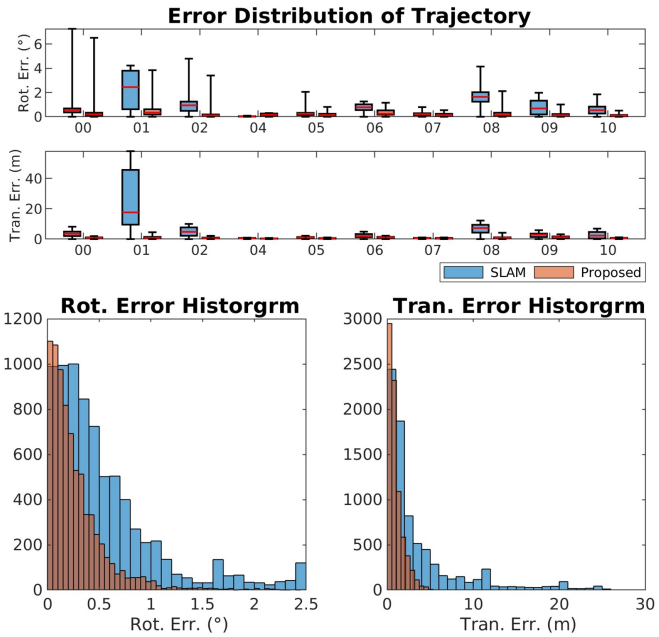

Fig. 4. A comparison of localization error distribution. The RMSE is shown in Table I. The 1st-2nd rows are the rotation (unit °) and translation (unit m) error distribution of each sequence, where $\mathbf{S}$ is by trajectory origin. The 3rd row reports the histograms of all rotation and translation errors, where $\mathbf{S}$ is by multiple ground truth. Overall, the error by the proposed framework is lower and more concentrated.

[9]To better reflect the global localization accuracy, we use the absolute error rather than the relative translation/rotation error for evaluation.

[10]In this paper, we present the result using $\mathbf{S}$ by trajectory origin, unless otherwise noted.

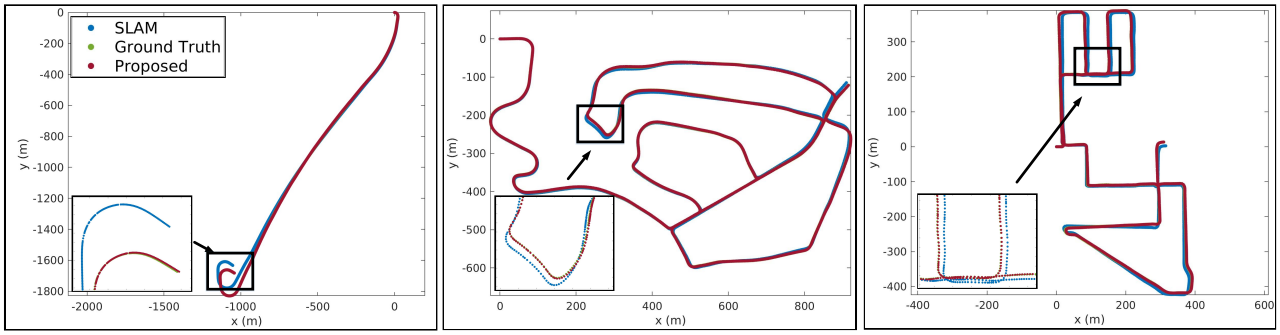

Fig. 5. Examples of the estimated trajectories on KITTI. The first figure shows a scenario without loop closure. For the second and the third figures, the trajectories estimated by SLAM are with loop closure. For all results, our estimated trajectories (red) are very close to the ground truth (green).

$0.34°$) and the rotation error reduces 83% (from $6m$ to $1m$). Fig. 4 shows a comparison of the error distribution. Overall, the localization error by the proposed framework is lower and more concentrated, which illustrates the robustness of our method.

TABLE II

ACCURACY COMPARISON WITH G2S PREDICTION

Meth.	Azimuth (°)		Longitudinal (m)		Lateral (m)	
	mean↓	1°(%) ↑	mean↓	1m (%) ↑	mean↓	1m (%) ↑
[7]	0.163	99.9%	7.651	20.4%	0.746	79.6%
Ours	0.231	98.0%	0.544	84.1%	0.485	89.9%

We report the mean and the percentage of the estimated results larger than the threshold. Results are averaged across ten KITTI sequences.

Fig. 5 and Fig. 6 visually report the estimated trajectory on KITTI and FordAV datasets[11]. We can see that the proposed framework improves the localization accuracy, especially for the scenario without loop closure (the 1st figure in Fig. 5 and all figures in Fig. 6) which is common in real applications for autonomous driving.

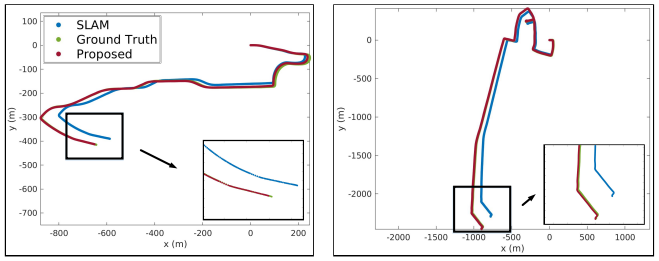

Fig. 6. Evaluation on FordAV. On average, the RMSE is $\{1.08°, 65.24\text{m}\}$ by [10], and $\{0.80°, 7.17\text{m}\}$ by the proposed method.

Table II presents a comparison of the estimated trajectory with the G2S prediction in terms of the longitudinal, lateral, and azimuth error. We can see that the translation estimates (especially along the longitudinal direction) by the proposed methods are with higher accuracy, while the rotation error using both methods is small.

E. Ablation Study

We conduct an ablation study to evaluate the contribution of each module in the proposed framework. The translation errors by removing different configurations are present in Table. III. We can see that the result using the full proposed framework achieves the best performance.

TABLE III

ABLATION STUDY ON DIFFERENT CONFIGURATIONS

	All G2S	SPB	VOC	No s	Non-Iter	Full
mean↓	2.557	1.708	1.190	2.521	1.786	0.808
median↓	1.703	1.838	1.103	2.278	1.121	0.714
RMSE↓	3.367	1.880	1.352	2.953	2.288	0.938

All G2S: non-iterative G2S-SLAM fusion using all G2S poses; SPB: using only spatial bound to check G2S poses; VOC: using only visual odometry consistency to check G2S poses; No s: without scale estimate in (4); Non-Iter: non-iterative G2S-SLAM fusion (updating the trajectory once using all selected G2S poses, rather than the pipeline in Sec. IV-C.4); Full: results using all proposed modules.
Results are averaged across ten KITTI sequences.

VI. LIMITATION

Although achieving promising results, there are several limitations. First, the reliance on SLAM for G2S selection means that tracking loss within SLAM will impact the proposed method, particularly we find that some challenges causing tracking loss (e.g., illumination changes from the ground view images) also lead to inaccurate G2S predictions. Second, the proposed method requires more computational resources. Finally, satellite images are not accessible for certain environments, e.g., tunnels or indoor parking areas.

VII. CONCLUSION

This paper proposes a framework for vehicle localization. The framework combines the stereo SLAM and the G2S cross-view registration to improve the camera localization accuracy. The G2S poses are predicted using a deep-learning based method, and their validities are checked using a coarse-to-fine method. The selected prediction is then fused with the SLAM poses by solving a scaled pose graph problem. Detailed validation using real experiments is conducted, and the results illustrate the localization accuracy as well as the potential value of this framework to be applied for autonomous driving.

In the future, we plan to investigate a more tightly coupled fusion method by combining the 3D maps by SLAM. We also plan to investigate more advanced G2S methods. Our goal is to develop a SLAM-G2S-Fusion system for autonomous driving.

[11]FordAV is more challenging than KITTI for visual localization, e.g., the rapid illumination changes between the consecutive images can cause tracking loss of the vSLAM system. Because of this, the evaluation is conducted using part of the trajectories without tracking loss from SLAM.

REFERENCES

[1] L. Xiong, R. Kang, J. Zhao, P. Zhang, M. Xu, R. Ju, C. Ye, and T. Feng, "G-vido: A vehicle dynamics and intermittent gnss-aided visual-inertial state estimator for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 845–11 861, 2021.

[2] T. G. Reid, S. E. Houts, R. Cammarata, G. Mills, S. Agarwal, A. Vora, and G. Pandey, "Localization requirements for autonomous vehicles," *arXiv preprint arXiv:1906.01061*, 2019.

[3] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 176–183.

[4] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 9–16.

[5] L. Liu, H. Li, and Y. Dai, "Efficient global 2d-3d matching for camera localization in a large-scale 3d map," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2372–2381.

[6] L. Liu, Y. Lin, X. Liang, Q. Xu, M. Jia, Y. Liu, Y. Wen, W. Luo, and J. Li, "Cyberloc: Towards accurate long-term visual localization," *arXiv preprint arXiv:2301.02403*, 2023.

[7] Y. Shi, F. Wu, A. Perincherry, A. Vora, and H. Li, "Boosting 3-dof ground-to-satellite camera localization accuracy via geometry-guided cross-view transformer," *Accpeted by International Conference on Computer Vision (ICCV)*, 2023.

[8] S. Hu and G. H. Lee, "Image-based geo-localization using satellite imagery," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1205–1219, 2020.

[9] P.-E. Sarlin, D. DeTone, T.-Y. Yang, A. Avetisyan, J. Straub, T. Malisiewicz, S. R. Bulò, R. Newcombe, P. Kontschieder, and V. Balntas, "Orienternet: Visual localization in 2d public maps with neural matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 632–21 642.

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[11] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[14] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[15] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[16] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[17] Y. Wang, Y. Ng, I. Sa, A. Parra, C. Rodriguez, T. J. Lin, and H. Li, "Mavis: Multi-camera augmented visual-inertial slam using se2 (3) based exact imu pre-integration," *arXiv preprint arXiv:2309.08142*, 2023.

[18] L. Von Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "Gn-net: The gauss-newton loss for multi-weather relocalization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 890–897, 2020.

[19] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers, "LM-Reloc: Levenberg-Marquardt based direct visual relocalization," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 968–977.

[20] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3247–3257.

[21] Y. Shi and H. Li, "Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 010–17 020.

[22] S. Wang, Y. Zhang, A. Vora, A. Perincherry, and H. Li, "Satellite image based cross-view localization for autonomous vehicle," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3592–3599.

[23] S. Hu, M. Feng, R. M. Nguyen, and G. H. Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7258–7267.

[24] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5624–5633.

[25] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] A. Toker, Q. Zhou, M. Maximov, and L. Leal-Taixé, "Coming down to earth: Satellite-to-street view synthesis for geo-localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6488–6497.

[27] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 990–11 997.

[28] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, "Continuous self-localization on aerial images using visual and lidar sensors," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7028–7035.

[29] Z. Xia, O. Booij, M. Manfredi, and J. F. Kooij, "Visual cross-view metric localization with dense uncertainty estimates," in *European Conference on Computer Vision*. Springer, 2022, pp. 90–106.

[30] S. Wang, Y. Zhang, A. Perincherry, A. Vora, and H. Li, "View consistent purification for accurate cross-view localization," *Accpeted by International Conference on Computer Vision (ICCV)*, 2023.

[31] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, "Uncertainty-aware vision-based metric cross-view geolocalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 621–21 631.

[32] Y. Zhang, T. Zhang, and S. Huang, "Comparison of ekf based slam and optimization based slam algorithms," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 1308–1313.

[33] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," *Robotics: science and Systems VI*, vol. 2, no. 3, p. 7, 2010.

[34] Y. Chen, L. Zhao, Y. Zhang, S. Huang, and G. Dissanayake, "Anchor selection for slam based on graph topology and submodular optimization," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 329–350, 2021.

[35] N. Chebrolu, T. Läbe, O. Vysotska, J. Behley, and C. Stachniss, "Adaptive robust kernels for non-linear least squares problems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2240–2247, 2021.

[36] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[37] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1367–1376, 2020.

[38] Google, "Maps Static API," https://developers.google.com/maps/documentation/maps-static/overview, accessed: September-2023.

[39] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[40] B. K. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *JOSA A*, vol. 5, no. 7, pp. 1127–1135, 1988.