Global Localization and Guided Relocalization in Unstructured Environments using Semantic Objects

By

Jacqueline Ankenbauer

B.S. Aerospace Engineering
Massachusetts Institute of Technology, 2022

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Jacqueline Ankenbauer
Department of Aeronautics and Astronautics
May 23, 2023

Certified by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Global Localization and Guided Relocalization in Unstructured Environments using Semantic Objects

by

Jacqueline Ankenbauer

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2023, in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis presents a novel framework for global localization and guided relocalization of a vehicle in an unstructured environment. Compared to existing methods, this pipeline does not rely on cues from urban fixtures (e.g., lane markings, buildings), nor does it make assumptions that require the vehicle to be navigating on a road network. Instead, localization is achieved in both urban and non-urban environments by robustly associating and registering the vehicle's local semantic object map with a compact semantic reference map, potentially built from other viewpoints, time periods, or modalities. Robustness to noise, outliers, and missing objects is achieved through the graph-based data association algorithm. Further, the guided relocalization capability of the pipeline mitigates drift inherent in odometry-based localization after the initial global localization. The pipeline is evaluated on two publicly-available, real-world datasets to demonstrate its effectiveness at global localization in both non-urban and urban environments. The Katwijk Beach Planetary Rover dataset [17] is used to exemplify the pipeline's ability to perform accurate global localization in unstructured environments at as low as 0.58 m accuracy. Demonstrations on the KITTI dataset [15] achieve an average pose error of 3.8 m across all 35 localization events on Sequence 00 when localizing in a reference map created from aerial images. Compared to existing works, this pipeline is more generalizable because it can perform global localization in unstructured environments using maps built from different viewpoints and dates.

Thesis Supervisor: Jonathan P. How
Title: R. C. Maclaurin Professor of Aeronautics and Astronautics

Acknowledgments

I am grateful to have been a part of the AeroAstro Department at MIT for both my Bachelor's and my Master's. MIT has built a community of professors who love teaching their students, teaching assistants who are quick to help, and students who encourage each others' professional goals. I am fortunate to have been a part of MIT.

I would like to extend a huge thank you to my advisor, Jonathan How. No matter how full his calendar is, he always has time to meet with his graduate students.

Thank you to The Boeing Company for your support. This work is supported by The Boeing Company under Cooperative Agreement MRA#2017-656.

I am grateful for Parker Lusk's guidance and advice, whose ability to communicate technical information without losing sight of the big picture is a skill in which I hope to grow. Thank you to Kaveh Fathian for his help getting my research started and for teaching me to think more like a researcher. I am appreciative of the support of all of the graduate students and postdocs in the Aerospace Controls Lab.

I am eternally indebted to my parents for their support, but most importantly, for setting an example of hard work, dedication, and integrity. The entire time I have been alive, they have been in my corner.

Thank you to my sister and her husband, Elizabeth and Leopold Beuken, for paving the way to earning a Master's.

Thank you to my grandparents, Bill and Theresa Barna, for their example of determination, perseverance, and commitment. I am sure my grandfather is reading my thesis from heaven, saying "it's better than a sharp stick in the eye."

Thank you to all of my family on both the Pedlow and Ankenbauer sides, for your support and encouragement.

I owe the deepest gratitude to my husband, Tom Ankenbauer, for taking great interest in my research while keeping an unmovable determination to keep our family the highest priority. His unending love, strength, and support cannot be understated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Global localization is the process of determining a vehicle's pose (i.e., position and orientation) in its environment without an initial estimate. While global navigation satellite system (GNSS) based methods have traditionally been used to provide global positioning in open settings with good satellite visibility, positioning quality quickly degrades in the presence of occlusion, multipath, or spoofing (e.g., in urban canyons, underground, or adversarial settings). To solve this problem, methods have been proposed that leverage onboard measurements to localize a vehicle within a reference map.

Current frameworks typically approach global localization with the assumption that the vehicle is on a road in an urban environment [23]. Many methods localize a vehicle using OpenStreetMap (OSM) [5, 8, 12, 55, 61, 63], which is freely-available and memory efficient, but is limited to urban settings and requires significant effort to map a new area. Methods using semantics often begin with a non-uniform prior, assuming the vehicle is beginning on a road [33, 56] and focus primarily on structures existing only in urban environments (e.g., lane markings, traffic signs, and road structure [5,18, 36,56]). In practice however, applications such as military reconnaissance missions or search and rescue missions cannot make these assumptions. Global localization in an unstructured environment requires localizing with minimal data because information

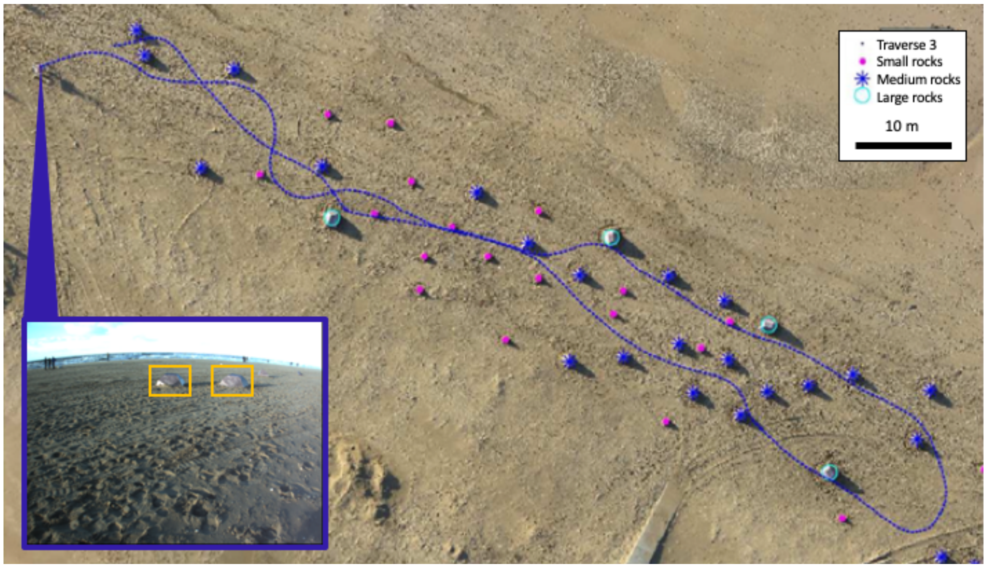


Figure 1-1. The pipeline can localize the ground rover of the Katwijk dataset [17] in an unstructured environment with a ground truth reference map of small, medium, and large rocks. This is accomplished by creating a vehicle object map of the rocks observed while driving (identified by the yellow bounding boxes in image) and registering them with the rocks in the reference map. With a reference map spanning roughly 1 km, 0.68 m accuracy has been achieved on the Katwijk dataset.

such as road structure and lane markings is not present. The environment in Figure 1-1 is an example of an unstructured, non-urban environment. The search space also increases because the vehicle could be in any pose within the reference map, not just on a road.

Changes in the environment such as lighting differences, seasonal changes, or objects being created, moved, or removed cause discrepancies between reference and vehicle maps, which is challenging for current methods. High-density maps are high-quality and can lead to very low position errors, but they are expensive to generate and prone to becoming outdated [14]. Furthermore, map size becomes increasingly important when considering sharing information between vehicles or a central database.

Reference maps created from a different viewpoint than the vehicle (e.g., localizing a ground vehicle in an aerial map or using maps built from opposing ground views) is important because the available reference maps may not be from the same viewpoint as the vehicle (e.g., localizing a ground vehicle in satellite imagery). However, image-based methods which use descriptors, such as visual bag of words [13], suffer from viewpoint changes. These methods may fail when localizing a ground vehicle driving north in a reference map built from a ground vehicle driving south in the same area [29].

In light of these issues, this framework is capable of globally localizing a vehicle in an unstructured environment using a reference map created from an arbitrary viewpoint. Correspondences between objects in the local vehicle map and in the reference map are made by exploiting the geometric consistency of potential associations [30]. Informed by object semantics, the space of potential object associations is reduced, allowing efficient identification of the largest set of geometrically consistent associations using a maximum clique solver [42]. The vehicle's pose is then found by registering the objects in the local vehicle map with their associated reference map objects. Importantly, leveraging geometric consistency in a graph-based manner enables the pipeline to achieve localization in unstructured maps from various viewpoints and with robustness against outliers in both the vehicle and reference maps. Thus, as long as objects can be identified and reconstructed in each map, global localization can be achieved. While globally localizing in an unstructured environment is a challenging problem due to decreased amount of available data, an increased search space, and potentially-outdated maps, this pipeline has been specifically designed to succeed in these environments.

1.2 Contributions

In summary, the contributions of this thesis are:

- A global localization framework robust to outliers and viewpoint changes due to its graph-based object association formulation and use of compact semantic maps.

- A framework capable of localizing in unstructured environments. With no prior assumptions on the existence of an urban setting, the same pipeline has been demonstrated to successfully localize in unstructured environments such as the Katwijk Beach Planetary Rover dataset [17].

- A guided relocalization mode to continually correct the pose estimate after global localization in order to reduce effects of drift.

- A demonstration of successful localization and guided relocalization achieving state-of-the-art performance on real data from the KITTI dataset [15] using a reference map from aerial images captured on a different date with many outliers.

1.3 Thesis Outline

This thesis focuses on the global localization and guided relocalization of a ground vehicle in a reference map. Specifically, the developed pipeline is designed to work in both urban and non-urban settings, with outdated reference maps, and from maps built from extreme viewpoint differences.

Chapter 2 of this thesis discusses similar literature in the areas of global localization, long-term localization, place recognition, and loop closure detection. The current literature focuses on urban environments and assumes that the vehicle being localized is on a road. This pipeline, however, is more generalizable to different applications than similar methods because it is able to work in unstructured environments with no assumption of the vehicle driving on a road.

The pipeline is detailed in Chapter 3, as are the experimental setups on the Katwijk Beach Planetary Rover dataset [17] and the KITTI dataset [15]. The focus of this chapter is on the details of each component of the framework and the experimental setup.

Experiments are conducted to demonstrate the four primary characteristics of this pipeline: global localization in unstructured and structured environments, robustness to outliers, view-invariance, and drift reduction. The experiments and results are presented in Chapter 4.

Chapter 5 concludes the thesis, providing a summary of contributions and directions for future work.

Chapter 2

Literature Review

2.1 Related Work

2.1.1 Global Localization, Place Recognition, and Loop Closure Detection

Global localization has close ties to literature on loop closure detection, place recognition, long-term localization, and image retrieval. Before discussing the related works, it is important to understand the difference between global localization, loop closure detection, place recognition, and long-term localization. While there are many similarities in these concepts (i.e., they can all be used to identify the location of a vehicle), there are subtle yet important differences in the specific question they are answering.

Global localization determines where a vehicle is inside of a reference map. The vehicle knows it is somewhere inside of a reference map and it must explore the area to determine and continually update its current location. Place recognition asks if the vehicle is currently in a place it has seen before. A vehicle begins with some database of locations and must identify at which location within the database it is currently located. Loop closure detection leverages place recognition to determine the transformation between the current pose and the place it has previously seen. As a vehicle explores the environment, it will build a database of places it has been and continually checks if it has revisited a location and if so, adjusts the trajectory appropriately. Long-term localization focuses on aligning two maps using descriptors and data association when the two maps have been created at a different time. Generally, the maps are created on different dates, in different seasons, or with different lighting conditions.

A primary difference between global localization and loop closure detection/place recognition is the concept of a reference map. While global localization is looking for the vehicle's location within some other map, loop closure detection and place recognition search for the vehicle's location across some finite database of locations (e.g., keyframes within the past trajectory). Additionally, there is always an answer to the global localization problem, assuming the vehicle stays within the bounds of the reference map. In place recognition and loop closure, the answer may simply be that the vehicle has not visited its current location previously.

2.1.2 Image-Based Methods

Appearance-based methods use images for localization by finding the most visually similar image in the reference database to the locally captured image [29]. Visual similarity is typically assessed based on low-level information such as color and reflectance values [51,54] or visual features and descriptors [34,44]. Early works such as [7,26] use local feature descriptors to compare and match images taken from different perspectives. Majdik et al. [31] use such features with simulated images from Google Street View to match against images from a quadrotor flying through an urban environment. Methods based on low-level features are impacted the most by changes in the environment such as illumination or seasonal changes and, more importantly, many fail under extreme viewpoint difference between the vehicle and reference images (e.g., aerial-ground).

2.1.3 Cross-View Methods

Cross-view methods, which localize a ground vehicle in aerial or satellite imagery, have been specifically developed to handle extreme viewpoint differences. Current state-of-the-art methods [40, 45, 57] are learning-based and use a Siamese network architecture [20, 50] to return a coarse localization (accuracy of hundreds of meters) across a very large area (e.g., city-wide). When coupled with particle filters, these techniques can provide a higher accuracy (tens of meters) in geo-tracking applications [10]. Cross-view algorithms are typically designed for the extreme air-ground viewpoint difference, but may not be directly applicable for other viewpoint variations (e.g., ground-ground viewed from opposite directions). Air-ground localization can also be achieved by other techniques, such as [2, 14, 60], which can obtain centimeter-level localization by exploiting a high-definition point cloud map of the environment. While accurate, these methods do not work with image maps and they require dense point clouds.

2.1.4 Semantic-Aided Methods

Semantic-aided methods leverage semantic information to assist with localization. Some methods compactly identify objects, their location, and potentially other characteristics to create reference and vehicle maps and then ultimately localize within each other [28, 47, 56]. Other methods use image segmentation [22, 33], semantic lidar point cloud matching [11, 46], general vertical structures [24, 56, 58], lane markings [18, 36], or buildings [32, 43, 49]. Semantic maps are often summarized with descriptors such as random walk descriptors [27, 28], histograms [64], or structural appearance descriptors [21], allowing the vehicle's local observations to be compared with previously seen objects in the reference map more efficiently. Other semantic-aided methods use OpenStreetMap, which is readily available and requires little memory. These methods compare observed roads [5,12,63], buildings [8,55], or both [61] to determine where the vehicle is inside the reference map. Overall, most semantic-aided methods are restricted to working only in urban or suburban settings.

2.1.5 Non-Urban Environments

Most works within global localization, long-term localization, place recognition, and loop closure detection are not suited for non-urban settings due their rigid reliance on urban semantic information (e.g., buildings, roads, lane markings) or their need for rich features within images (e.g., appearance-based methods). There have been works which address the difficulty in successfully running a SLAM system in non-urban environments due to the lack of features and roughness in the road [19, 62]. Specific to localization, works have used topological maps [35], wheel odometry combined with visual orientation tracking [16], or lidar point clouds [39] in order to refine a GPS estimate. Global localization in GPS-denied environments has been achieved by methods using lidar [14], stationary anchors within the region [48], and binary ground-nonground distinction [52, 53]. Despite success with global localization, these methods are either restricted by the size of dense point clouds, have requirements for external hardware in the field (e.g., anchors), are not robust to structural changes, or assume the vehicle is on an off-road trail.

2.2 Placement of This Work

This thesis presents a pipeline within the category of semantic-aided global localization methods. The primary goal is to identify objects as the vehicle explores the area, then match the identified objects to their corresponding object in the reference map in order to estimate the pose of the vehicle. Long-term localization methods such as [56, 59, 64] are similar to this pipeline because they, too, match local map objects to their corresponding global map object. However, long-term localization works focus on the core tools to align maps (e.g., descriptors and data association algorithms), whereas this pipeline consists of a full implementation of mapping and data association in order to continually estimate the pose of a vehicle throughout the entire trajectory. Additionally, long-term localization works use descriptors, whereas this pipeline does not. While descriptors are beneficial for increasing the scalability of a pipeline, they often constrain the problem to a certain environment. For example, the descriptors in [56] encode the direction of roadlines, thus constraining the problem to an urban setting.

Table 2.1 illustrates how this framework compares to other global localization frameworks. The second column lists characteristics with several state-of-the-art works which use the KITTI dataset [15] in their analysis. The third column compares against Viswanathan et al. [53], which is a global localization method for unstructured environments. One primary difference between this pipeline and all of the compared pipelines is that this pipeline begins with a uniform prior, meaning there is no assumption that the vehicle is on a road or trail. Additionally, the methods used for comparison have only been demonstrated to localize a ground vehicle in aerial data whereas this work tests the same aerial-ground configuration in addition to experiments localizing a ground vehicle in ground data.

This work uses semantic object maps and geometric consistency in order to be view-invariant and robust to structural changes in the environment. The generality of the classes being used and assumptions being made (i.e., no reliance on roads) allow this framework to successfully operate in unstructured environments. Furthermore, this method addresses the issue of relocalization after global localization to mitigate effects of drift.

2.3 Conclusion

The pipeline presented in this thesis is a semantic-aided method for global localization in a GPS-denied environment. Most of the similar works span global localization, long-term localization, place recognition, and loop closure detection, and operate exclusively in urban environments. This framework, however, is capable of localizing a vehicle in non-urban environments. Additionally, it is view-invariant, robust to outliers, and frequently refines the pose estimate as more information is received. The next chapter details the pipeline's structure and the experimental setup.

Table 2.1. A comparison of characteristics for several global localization methods. The second column describes all methods against which this pipeline is compared in Table 4.5. The third column, Viswanathan [53], is a state-of-the-art method for global localization in non-urban scenarios the using only onboard sensors and satellite images. All methods are semantic-aided. A dash (−) implies that this characteristic has not been demonstrated on the given pipeline.

Characteristic	Miller, Yan, Brubaker, Floros ([33], [61], [5], [12])	Viswanathan [53]	This Pipeline
Uniform prior*	No	No	**Yes**
Semantic Objects	No	No	**Yes**
Non-urban Environments	No	Yes	**Yes**
Urban Environments	Yes	−	**Yes**
Robustness to Env Changes	−	Yes	**Yes**
Ground-ground	−	−	**Yes**
Drift reduction after GL	Yes	−	**Yes**

*Uniform prior implies the pipeline does not make any assumption that the vehicle is on a road (this primarily affects the size of the search space).

Chapter 3

Localization Framework

This section describes the pipeline in detail as well as the experimental setups when testing with the Katwijk Beach Planetary Rover dataset [17] and the KITTI dataset [15].

3.1 Pipeline Overview

3.1.1 System Overview

Figure 3-1 illustrates the primary components of the pipeline. In this framework, a vehicle explores the environment and creates a vehicle map $\mathcal{M}_{\text{veh}}$ of objects detected and reconstructed by using onboard sensors. The vehicle map $\mathcal{M}_{\text{veh}}$ and reference map $\mathcal{M}_{\text{ref}}$ consist of objects $o_i = (u_i, c_i)$, represented by their 3D centroid $u_i \in \mathbb{R}^3$ and a class, $c_i \in \mathcal{C}$, where $\mathcal{C}$ is a set of classes known a priori. Drift in the trajectory estimate contributes to object reconstruction inaccuracies, so registration is performed on only the r most recently seen objects, denoted $\mathcal{M}^r_{\text{veh}} \subseteq \mathcal{M}_{\text{veh}}$. Objects in $\mathcal{M}^r_{\text{veh}}$ are then associated with their corresponding objects in a the reference map using a maximum clique algorithm. The candidate transformations $T^{\text{i}}_{\text{cand}}$ provided by the maximum clique algorithm are analyzed and either accepted or rejected. If a transformation has been accepted at any point, the pose of the vehicle is estimated using the most recently accepted transformation, T_{cur}. Variables defined throughout this section are listed in Table 3.1 and Table 3.2.

When identifying and analyzing candidate transformations between the vehicle and reference maps, the pipeline leverages two operating modes: global localization and guided relocalization. The pipeline begins in the global localization mode, wherein the vehicle searches for its global pose within a provided reference map. It is emphasized that in this mode, no prior information is leveraged (e.g., no initial guess and no assumption that the vehicle is restricted to roads). Once a candidate transformation between the vehicle's local observations and reference map is accepted (see Section 3.1.5), global localization is achieved and the pipeline switches to guided relocalization. The guided relocalization mode continually updates the accepted transformation by leveraging past information in order to reduce the drift of the SLAM system (see Section 3.1.6). The difference between the two modes are illustrated in Figure 3-2 and detailed in Section 3.1.5 and Section 3.1.6.

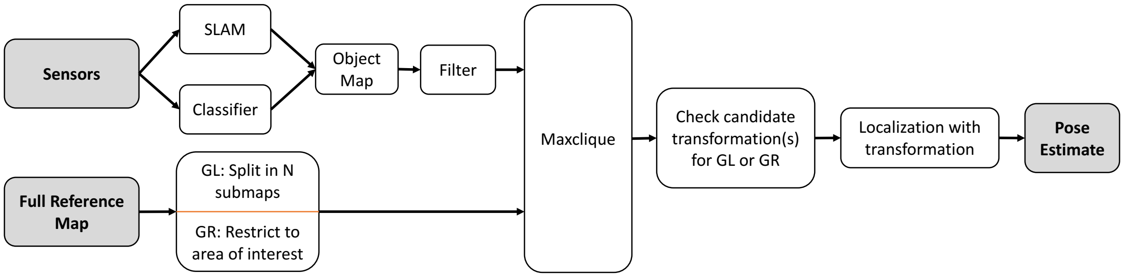

Sensors
SLAM
Classifier
Object Map
Filter
Maxclique
Full Reference Map
GL: Split in N submaps
GR: Restrict to area of interest
Check candidate transformation(s) for GL or GR
Localization with transformation
Pose Estimate

Figure 3-1. Sensors onboard the vehicle are inputs for the SLAM system and the classifier. The output trajectory, point cloud, and bounding boxes are then used to create a vehicle map, which is filtered and prepared before being fed into the maximum clique algorithm. The full reference map is either prepared into several submaps (global localization) or restricted to the area of interest (guided relocalization). The maximum clique algorithm produces candidate registration(s) by aligning these maps. If a transformation has been accepted, the pose of the vehicle is estimated using the current transformation.

3.1.2 Reference Map

The reference map can either be constructed offline or be updating in real-time. As the pipeline searches over the entire reference map during global localization, the reference map is split into k submaps to increase computational efficiency. Given a

Table 3.1. Variables describing the vehicle and reference maps as well as candidate and accepted transformations as discussed in Section 3.1.

Parameter	Description
M_{veh}	The full vehicle map containing all objects seen by the vehicle
$M_{\mathrm{veh}}^{\mathrm{r}}$	A subset of the full vehicle map containing the most recent r objects seen by the vehicle
$M_{\mathrm{veh}}^{\mathrm{r}'}$	A subset of the full vehicle map containing the most recent r' objects seen by the vehicle
M_{ref}	The full reference map containing all objects
$M_{\mathrm{ref}}^{\mathrm{i}}$	A subset of the full reference map containing objects in submap i (used for global localization)
$M_{\mathrm{ref}}^{\mathrm{res}}$	The restricted reference map is a subset of the full reference map used during guided relocalization
T_{cand}	A candidate transformation
$T_{\mathrm{cand}}^{\mathrm{i}}$	The candidate transformation corresponding to submap i (used during global localization)
T_{cur}	The most recently accepted registration

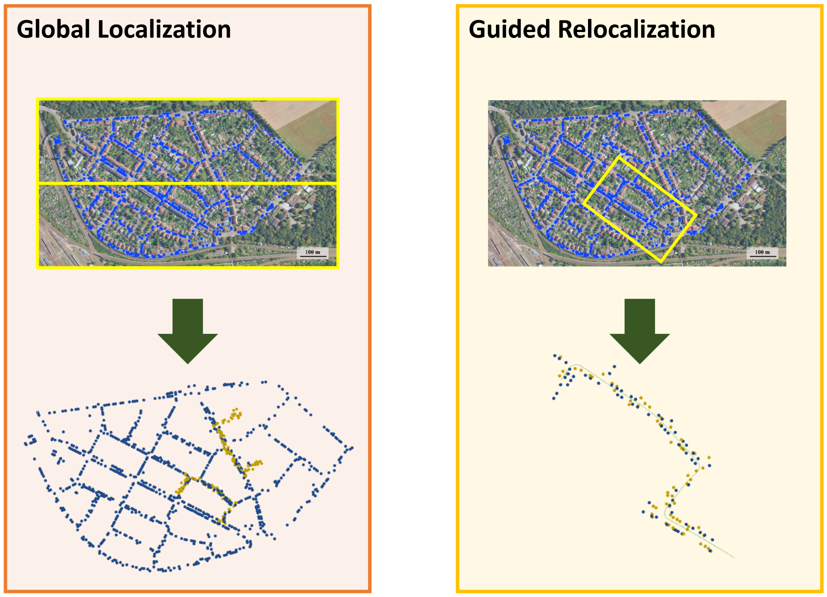


Figure 3-2. (Left) In the global localization mode, the vehicle is localized in the entire reference map, split into k submaps. Thus, k candidate transformations are identified. (Right) In the guided relocalization mode, the vehicle map is localized in a portion of the reference map constrained to the areas of interest. Thus, there is only one candidate transformation.

split reference map, with some probability, the true location of the vehicle map may span multiple submaps, so the k submaps have a specified level of overlap β. In order for global localization to succeed, the correct transformation must be identified by the maximum clique algorithm and accepted by the global localization criteria despite possibly spanning multiple maps. The latter is handled in Section 3.1.5, while the former is handled by choosing appropriate parameters. Candidate transformations causing the vehicle map to span multiple submaps is unlikely to be identified because only few inliers will be identified within each submap. To mitigate this risk, k should be chosen such that each submap is at least large enough to contain an estimated size of $\mathcal{M}^r_{\text{veh}}$, and β should be chosen to be sufficiently large. All of the resulting submaps, $\mathcal{M}^i_{\text{ref}} \subset \mathcal{M}_{\text{ref}}$, $i = 1, \ldots, k$, are input into the maximum clique registration module.

In contrast, after global localization is achieved and the pipeline switches to the guided relocalization mode, the reference map is strategically constrained to objects within distance d_{reloc} of objects in the vehicle map, based on the current transformation. This restricted reference map $\mathcal{M}^{\text{res}}_{\text{ref}}$ is input into the registration module.

3.1.3 Construction of the Vehicle Map

The pipeline constructs $\mathcal{M}_{\text{veh}}$ online by detecting objects and reconstructing their centroids. Sensors onboard the vehicle (e.g., stereo cameras, lidar sensors, IMU) must provide a scaled trajectory as well as enable the classification of objects and the reconstruction of their centroids. A SLAM system takes the data from onboard sensors as input and provides a scaled trajectory and a sparse point cloud at each timestep. The onboard camera images are input into a classifier, which is expected to provide bounding boxes, in pixels, around objects in each frame along with the class of each identified object. In order to minimize false positives, the bounding boxes are required to be of a minimum height p_{h} and width p_{w} in pixels.

At each timestep, objects are classified, and their centroids are reconstructed in real-time. For each timestep, the point cloud is projected onto the image and the points that lie within each bounding box are said to correspond to that object.

The distance to each object is chosen to be the median distance to the 3D points corresponding to that object. The median distance is used because it is assumed some points may lie in the background or foreground of the bounding box, not on the object itself. The set of valid objects in this frame is said to be the objects identified by a bounding box which have at least one corresponding point cloud point and a distance from the camera of less than d_{obj}. The final 3D centroid estimate of each valid object is taken to be the center of the bounding box projected into 3D space by the estimated distance to the object. After the valid objects at timestep t_n are reconstructed, they are compared to all objects seen in the previous timesteps $t = t_0, \ldots, t_{n-1}$. All objects of the same class within a radius ϵ_{fus} are fused together and all objects which have been reconstructed at minimum τ_{sight} times are added to $\mathcal{M}_{\text{veh}}$. As previously mentioned, over long distances, the trajectory drift skews the object centroid estimates in $\mathcal{M}_{\text{veh}}$, so the r most recently seen objects are used to create $\mathcal{M}^r_{\text{veh}}$, which is input into the data association algorithm.

3.1.4 Registration

Robust registration is a core component of the proposed framework. A graph-based formulation is used to solve the registration problem by finding the largest set of geometrically consistent objects that match between each reference submap and the vehicle map. Denoting the association that matches the points p_i and q_i by $a_i = (p_i, q_i)$, two associations a_i and a_j are considered *geometrically consistent* if and only if the distance between the points is preserved (i.e., $\|p_i - p_j\| = \|q_i - q_j\|$). In practice, due to noise and inaccuracies, a threshold ϵ is set and associations are considered consistent when $d(a_i, a_j) \overset{\text{def}}{=} |\, \|p_i - p_j\| - \|q_i - q_j\| \,| < \epsilon$. Now, by denoting the set of associations between objects of the same class in the reference and vehicle maps as $A \overset{\text{def}}{=} \{(o_i, o_j) : \, (u_i, c_i) \in \mathcal{M}_{\text{ref}}, \, (u_j, c_j) \in \mathcal{M}^r_{\text{veh}}, \, c_i = c_j\}$, the problem of finding the

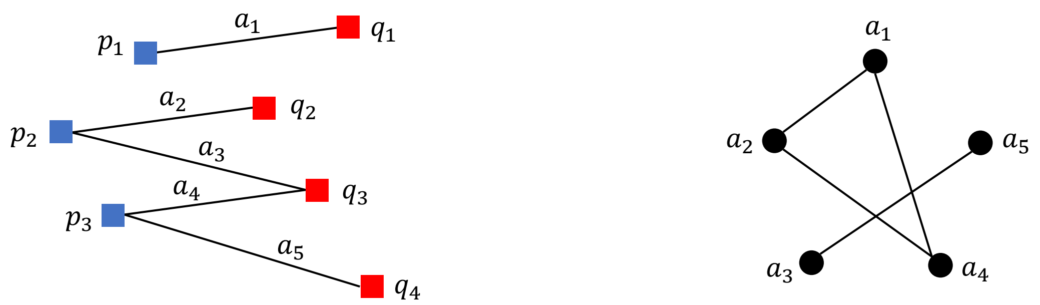


Figure 3-3. Maximum clique formulation for registration. (Left) Points p and q are matched by associations a. (Right) Graph with nodes representing associations and edges indicating their consistency (i.e., associations with (nearly) identical distances between their endpoints). The largest clique $A_{\mathrm{c}}^* = \{a_1, a_2, a_4\}$ is the largest set of consistent associations.

largest set of consistent associations, A_{c}^*, can be defined formally as

$$\begin{aligned} \underset{A_{\mathrm{c}} \subset A}{\text{maximize}} \quad & |A_{\mathrm{c}}| \\ \text{subject to} \quad & d(a_i, a_j) < \epsilon, \ \forall_{a_i, a_j \in A_{\mathrm{c}}}. \end{aligned} \tag{3.1}$$

Problem (3.1) can be modeled as a graph whose vertices represent associations and edges represent consistent associations. The optimal solution is equivalent to the maximum clique of the graph, as illustrated in Figure 3-3. Although typically NP-hard, finding the maximum clique can be solved relatively quickly for sparse graphs (resulting from many inconsistent associations created by an all-to-all scheme) using the parallel maximum clique (PMC) algorithm [42].

The pipeline uses this maximum clique method for data association between the reference and vehicle maps, but with an extra constraint compared to Problem (3.1). In addition to the geometric consistency constraint $d(a_i, a_j) < \epsilon$, for two associations to be valid, the distance between the two points in both the reference map and the vehicle map must be larger than a distance d_{in}. Thus, Problem (3.1) becomes

$$\begin{aligned} \underset{A_{\mathrm{c}} \subset A}{\text{maximize}} \quad & |A_{\mathrm{c}}| \\ \text{subject to} \quad & d(a_i, a_j) < \epsilon, \ \forall_{a_i, a_j \in A_{\mathrm{c}}} \\ & \|p_i - q_i\| \geq d_{\mathrm{in}} \\ & \|p_j - q_j\| \geq d_{\mathrm{in}}. \end{aligned} \tag{3.2}$$

The additional constraints in Problem (3.2) protect against finding registrations which have many inliers in a region of the vehicle map with dense objects. While these registrations contain many inliers, they are often incorrect, as regions dense with objects are sometimes created by inaccurate centroid reconstructions and these regions often result in many incorrect associations. Thus, the threshold for distances between associations increases the likelihood that the data association algorithm will find the correct registration.

To find candidate transformations, the registration module periodically solves Problem (3.2) by registering $\mathcal{M}^r_{\text{veh}}$ to each of the reference map submaps ($\mathcal{M}^i_{\text{ref}}$ during global localization or $\mathcal{M}^{\text{res}}_{\text{ref}}$ during guided relocalization). During global localization, an all-to-all association scheme within objects of the same class is used. In other words, an object in each reference submap is initially associated to every object in the vehicle map of the same class. During guided relocalization, previously identified associations from the last relocalization event are leveraged by restricting these objects to be associated only with each other. The remaining objects are associated using an all-to-all association scheme to allow additional associations to be identified.

Solving (3.2) provides the maximum set of valid associations despite many outlier associations generated by the all-to-all association scheme. These associations are then used in the least-square fitting of matched objects via Arun's method [1], which gives the optimal transformation $T_{\text{cand}} \in \text{SE}(3)$ that registers $\mathcal{M}^r_{\text{veh}}$ to $\mathcal{M}_{\text{ref}}$.

3.1.5 Global Localization

The data and decision making flow of this pipeline is detailed in Figure 3-4. During global localization, candidate registrations $\{T^i_{\text{cand}}\}^k_{i=1}$ are identified for each of the k submaps $\{\mathcal{M}^i_{\text{ref}}\}^k_{i=1}$. For a given candidate registration T^i_{cand}, the number of inlier associations identified by Problem (3.2) is denoted as a_i. Because registrations with few associations are less likely to be reliable (e.g., due to perceptual symmetries or the anticipation of a changed environment), candidates with less than τ_{in} inlier associations are rejected. The set of candidates which pass the inlier association threshold are denoted by $\mathbb{V}$.

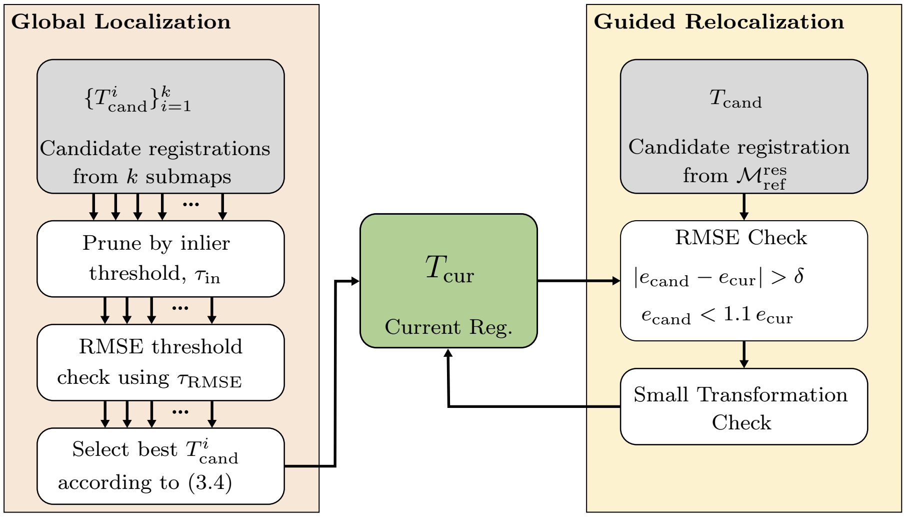

Global Localization
$\{T^i_{\text{cand}}\}^k_{i=1}$
Candidate registrations from k submaps
Prune by inlier threshold, τ_{in}
RMSE threshold check using τ_{RMSE}
Select best T^i_{cand} according to (3.4)
T_{cur}
Current Reg.
Guided Relocalization
T_{cand}
Candidate registration from $\mathcal{M}^{\text{res}}_{\text{ref}}$
RMSE Check
$|e_{\text{cand}} - e_{\text{cur}}| > \delta$
$e_{\text{cand}} < 1.1\, e_{\text{cur}}$
Small Transformation Check

Figure 3-4. Flow chart for accepting or rejecting a new registration in the global localization and guided relocalization modes.

The quality of the i-th registration is evaluated using the root-mean-square error (RMSE), denoted e_i, which measures the average distance between objects in the full vehicle map and their nearest neighbor object of the same class in the reference map. This value provides insight into how well the vehicle map as a whole aligns with the reference map, as opposed to only considering how well the objects associated using A_c^* are matched. An RMSE threshold τ_{RMSE} is used to check that there is at least one candidate transform of sufficient quality. Importantly, $\tau_{\text{RMSE}}(d_{\text{t}})$ is a piece-wise function of distance traveled, d_{t} specified by

$$\tau_{\text{RMSE}}(d_{\text{t}}) = \tau_{\text{RMSE},0} + \alpha_{\text{RMSE}}\text{floor}(d_{\text{t}}/d_{\text{RMSE,GL}}) \tag{3.3}$$

where $\tau_{\text{RMSE},0}$ is the initial value for $\tau_{\text{RMSE}}(d_{\text{t}})$ and d_{RMSE} is the required traveled distance before the threshold increases by α_{RMSE}. The threshold increases as the distance traveled increases to account for the distortion in the map due to drift in the trajectory estimate.

If none of the candidate registrations meet both the τ_{in} and $\tau_{\text{RMSE}}(d_{\text{t}})$ thresholds, the pipeline waits for new candidate transformations and repeats the process. If, however, at least one candidate registration passes these thresholds, the best registration is selected by

$$\begin{aligned} i^* = \; & \underset{i \in \mathbb{V}}{\operatorname{argmax}} \; a_i \\ & \text{subject to} \quad e_i \leq (1+\alpha)\, \underline{e}, \end{aligned} \tag{3.4}$$

where $\underline{e} = \min_{j \in 1,\ldots,k}\{e_j : a_j \geq \tau_{\text{in}}\} \leq \tau_{\text{RMSE}}$ is the best RMSE value in the set of valid transformations and $0 < \alpha \ll 1$. Thus, the result of Equation (3.4) is the candidate transformation with the most associations, provided that the RMSE value is close to the best RMSE value. In simpler words, if multiple transformations have similar RMSE near or below the threshold, the number of inliers is the best indicator of which transformation is accurate. Figure 3-5 illustrates the benefit of the RMSE heuristic in discerning if a candidate registration is of high or low quality.

Once a registration is accepted, it is stored as the current transformation $T_{\text{cur}} \leftarrow T_{\text{cand}}^{i^*}$. Then, the pipeline switches to the guided relocalization mode where T_{cur} will be frequently updated.

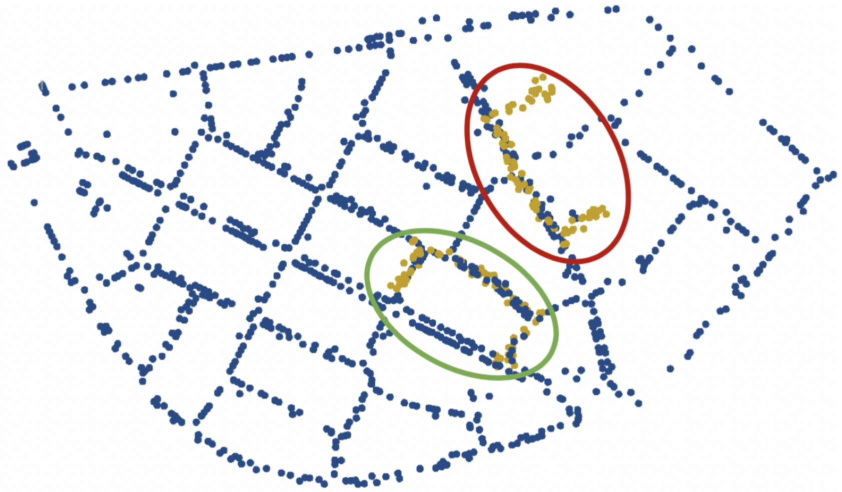


Figure 3-5. The blue points represent objects in the aerial reference map for KITTI Sequence 00. Two copies of the vehicle map are shown in yellow, each transformed by a candidate registration. Despite same number of inliers, the RMSE value allows the registration circled in red to be identified as incorrect and the registration in green to be accepted.

3.1.6 Guided Relocalization

Guided relocalization is used to frequently update T_{cur}, the current transformation between the local and global coordinate frames. These updates are important because as the vehicle moves, the trajectory estimation process accumulates drift. The criteria for accepting the candidate transformation T_{cand} is detailed in the right of Figure 3-4. Note that there is only one candidate transformation T_{cand} for guided relocalization at any given time because only the restricted reference map $\mathcal{M}_{\text{ref}}^{\text{res}}$ (as opposed to k submaps) is being input into the data association algorithm.

During guided relocalization, a candidate registration is compared to the current registration T_{cur} to determine if the candidate registration will be accepted. Unlike the global localization criteria, the new candidate registration does not have a required number of inlier associations because the framework is already confident in the approximate transformation between coordinate frames. The RMSE value is calculated for both the current accepted registration (e_{cur}) and the candidate registration (e_{cand}). These values are calculated as defined in Section 3.1.5, but using the $r' \geq r$ most recently seen objects in the vehicle map such that $\mathcal{M}_{\text{veh}}^{r} \subseteq \mathcal{M}_{\text{veh}}^{r'}$. Calculating the RMSE values with more vehicle map objects than were used to find the candidate transformation provides a better assessment of the quality of each transformation. To accept T_{cand}, the two RMSE values must be sufficiently different ($|e_{\text{cand}} - e_{\text{cur}}| > \delta_{\text{RMSE}}$) and the candidate registration's value must be similar or smaller than the current registration's value ($e_{\text{cand}} \leq (1 + \alpha)\, e_{\text{cur}}$).

The final criteria to accept the candidate transformation is that T_{cand} must be similar enough to T_{cur} in both translation and orientation. The requirements are

$$\begin{aligned} ||t_{\text{cand}} - t_{\text{cur}}|| \quad & \leq \delta_{\text{T}}(d_{\text{t}}^{'}) \\ \cos^{-1}(|q_{\text{cand}} \cdot q_{\text{cur}}|) \quad & \leq \delta_{\theta}(d_{\text{t}}^{'}) \end{aligned} \tag{3.5}$$

where $T_{\text{cand}} = (q_{\text{cand}}, t_{\text{cand}})$ and $T_{\text{cur}} = (q_{\text{cur}}, t_{\text{cur}})$ and $d_{\text{t}}^{'}$ is the distance traveled since the last accepted registration. Similar to relaxing τ_{RMSE} based on distance traveled for global localization, the transformation similarity requirements loosen to account

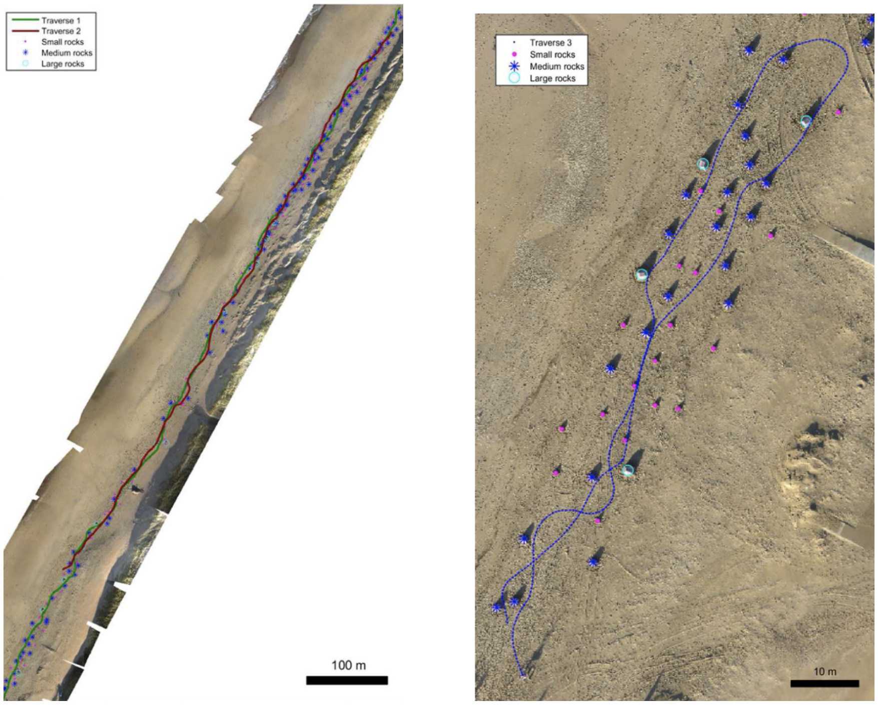


Figure 3-6. Reference map and trajectories for Traverses 1 and 2 (left) and for Traverse 3 (right) of the Katwijk dataset [17]. Ground truth locations for all small, medium, and large rocks are labeled.

for drift as the distance traveled since the last accepted registration increases. These thresholds $\delta_{\mathrm{T}}(d^{\prime}_{\mathrm{t}})$ and $\delta_{\mathrm{T}}(d^{\prime}_{\mathrm{t}})$ change according to

$$\begin{aligned} \delta_{\mathrm{T}}(d^{\prime}_{\mathrm{t}}) &= \delta_{\mathrm{T},0} + \alpha_{\mathrm{T}}\mathrm{floor}(d^{\prime}_{\mathrm{t}}/d_{\mathrm{RMSE,GR}}) \\ \delta_{\theta}(d^{\prime}_{\mathrm{t}}) &= \delta_{\theta,0} + \alpha_{\theta}\mathrm{floor}(d^{\prime}_{\mathrm{t}}/d_{\mathrm{RMSE,GR}}). \end{aligned} \tag{3.6}$$

If the candidate transformation is accepted, $T_{\mathrm{cur}} \leftarrow T_{\mathrm{cand}}$ and the process begins again when the next candidate registration is provided by the registration module.

3.2 Experiment Setup

3.2.1 Katwijk Beach Planetary Rover Experiment Setup

The Katwijk Beach dataset [17] provides a challenging scenario to test global localization. A rover was used to collect data while driving on a beach where small, medium, and large artificial rocks were placed in arbitrary locations along its route (see Figure 3-6). The pipeline localizes the rover in the reference map by identifying and classifying rocks by size and registering the rocks seen by the rover to rocks in the reference map.

Due to the challenging nature of the Katwijk dataset, off-the-shelf visual odometry packages (e.g., ORB-SLAM3 [6]) fail to estimate the robot's trajectory. In addition, the dataset does not provide a ground truth trajectory, so coarse ground truth poses for each camera frame are generated by interpolating the high-precision RTK GPS measurements. For this reason, the generated ground truth is used in this pipeline and the experiments focus on testing the global localization capability only.

Rock classification and bounding box construction is done offline, saved in ROS bags, and played back in real time during the experiments. To identify rocks, a 3D point cloud is reconstructed using stereo camera data (*LocCam*) captured onboard the rover. The ground plane is removed and the 3D points are clustered together in groups of at least 100 points using density-based spatial clustering of applications with noise (DBSCAN). Clusters with a maximum height of less than 0.15 m (half the height of the small rocks) are removed, and the remaining clusters are classified as either small, medium, or large rocks, based on the maximum height of the 3D cluster associated with that object. The size of potential rocks is known a priori. The 3D clusters corresponding to each object are projected back onto the left stereo image and rectangular bounding boxes are constructed around each rock. The 3D point cloud with the ground layer removed and the bounding boxes are saved in a ROS bag file.

As the pipeline runs in real-time, the 3D non-ground point cloud and bounding boxes are used as described in Section 3.1.3. In addition to the minimum width and height for bounding boxes (p_w and p_h), there are two additional requirements for the bounding boxes in order to include the object in the vehicle map. The method of identifying rocks resulted in many false positives, often in very wide bounding boxes using points which were incorrectly retained after removing the ground plane. Thus, the height-to-width ratio must be larger than 0.25. Second, as large and medium rocks were half-in and half-out of the camera frame, many were labeled as a smaller class of rocks because the maximum height of the rock was not in view. To correct for this effect, any bounding boxes completely contained in the leftmost 200 pixels or rightmost 200 pixels of the image are removed.

Within the dataset, there are three Traverses, broken into 8, 6, and 5 5-minute segments, respectively. Algorithm parameters are tuned on Traverse 1, Part 1 and the same values are used for other sequences. The parameters used are provided in Table 3.2. In particular, the object map is built of objects less than $d_{\text{obj}} = 12\,\text{m}$ away which have been seen in at least $\tau_{\text{sight}} = 20$ frames (defined in 3.1.3). The fusion radius for objects across frames is $\epsilon_{\text{fus}} = 1.0\,\text{m}$. In particular, a threshold of $\epsilon_{\text{reg}} = 1.5\,\text{m}$ is used for registration (defined in Section 3.1.4) and the initial RMSE threshold value is set to $\tau_{\text{RMSE},0} = 2\,\text{m}$. The size of the vehicle object map is not restricted (i.e., $\mathcal{M}^r_{\text{veh}} = \mathcal{M}_{\text{veh}}$) given how few rocks the vehicle sees in each segment. For all segments of Traverses 1 and 2, a minimum of $\tau_{\text{in}} = 8$ inliers are required. However, this parameter is loosened to $\tau_{\text{in}} = 6$ for Traverse 3, as it is possible to achieve high confidence of an accurate registration with less inliers because of the significantly lower number of objects in the reference map (45 objects in Traverse 3 as opposed to 212 objects in Traverses 1 and 2).

3.2.2 KITTI Experiment Setup

Experiments on the KITTI dataset demonstrate of the pipeline's robustness to outliers and enable the pipeline to be compared to other methods. For each sequence, two reference maps are considered. One is built by a ground-view lidar scan and the other by an aerial-view image. The SemanticKITTI [3] semantically labeled point cloud from the lidar sensor onboard the KITTI vehicle is used to create the lidar reference map. For each object identified by SemanticKITTI, the median values of the associated point cloud points is used to estimate the centroid. The centroids of the lidar reference maps are in 3D, and all associated errors from experiment localizing in the lidar maps are reported in 3D. The aerial reference map is created by manually annotating Google Satellite images using QGIS [37]. It is possible to automate the annotation by using classifiers trained for aerial images (see Section 5.2.5). The aerial reference map is in two dimensions. When localizing in the aerial reference maps, the

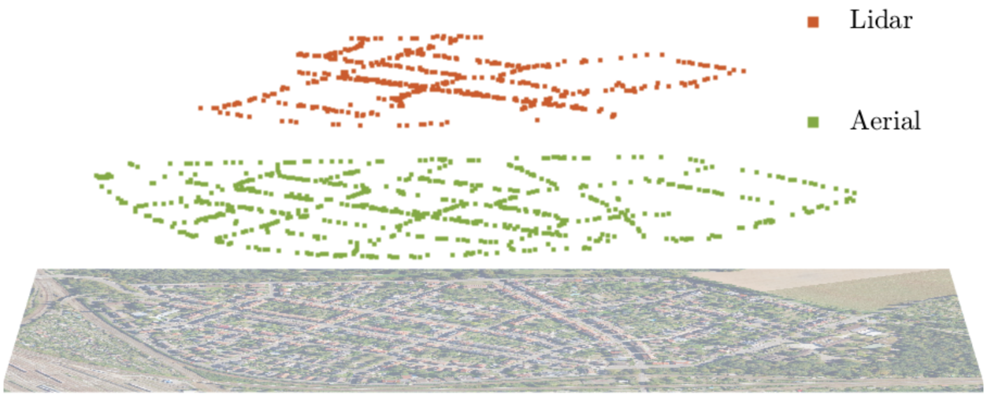


Figure 3-7. Reference object maps corresponding to KITTI Sequence 00 constructed using lidar scans (ground view), and Google Satellite georeferenced images (aerial view). Each square represents a semantic object such as a parking space or traffic sign. The bottom image is for reference.

vehicle map is projected into a 2D plane and the 2D version of Arun's method is used to identify $T_{\text{cand}} \in \text{SE}(3)$. The aerial reference map is larger than the lidar map for each sequence because the aerial images used to create the reference map span a larger area than the roads driven by the KITTI vehicle. Additionally, the aerial reference map contains more outliers when compared to the vehicle map. The data from the lidar scan is collected at the same time as the stereo images used to create the vehicle map, whereas the aerial reference map is created using images taken years apart from the KITTI dataset creation. The aerial and lidar reference maps for Sequence 00 can be seen in Figure 3-7.

Parking spaces and traffic signs are the only object classes used for global localization and guided relocalization, though the vast majority of the objects are parking spaces. To demonstrate robustness to outliers, the classifier identifies parking spaces by identifying cars. Using cars as a proxy for parking spaces leads to noisy estimates because not every parking space is occupied by a car and not every car is located in a parking space. Furthermore, since the parking spots occupied by cars change over time, using semantic object maps from different dates further stresses the algorithm's robustness to outliers. It is important to not that identifying cars and traffic signs in Google Satellite images to create the aerial reference map is challenging due to occlusion and lighting, as depicted in Figure 3-8. Furthermore, traffic signs are difficult to identify from an aerial view because they are very narrow (see Figure 3-9).

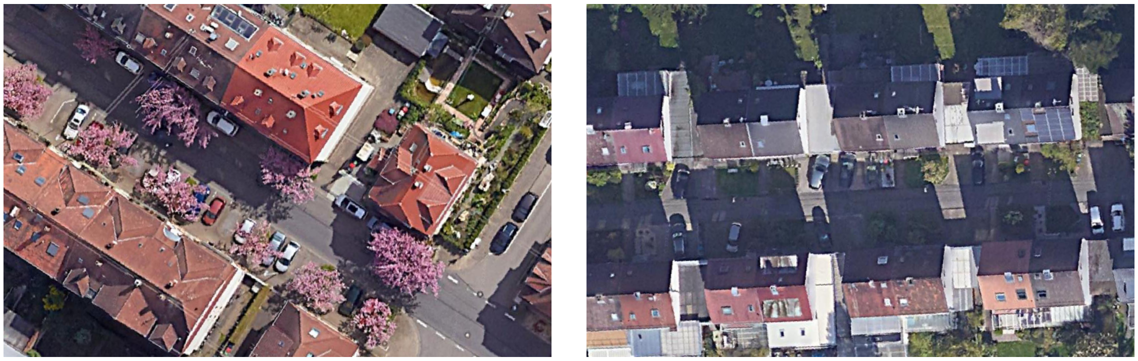


Figure 3-8. Occlusion (left) and lighting (right) in Google Satellite images make identifying cars and traffic signs difficult.

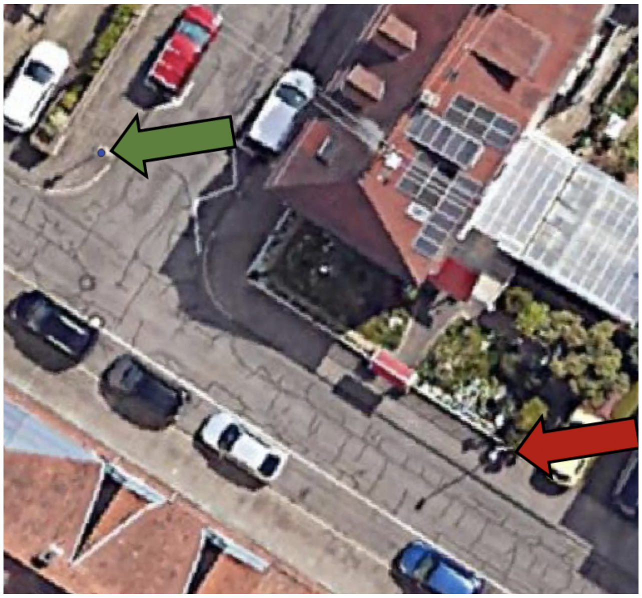


Figure 3-9. Shadows can be used to identify traffic signs in the Google Satellite images. This, however, is unreliable due to lighting and occlusion (see Figure 3-8). Additionally, the difference between traffic signs and street lights is subtle. The traffic sign candidate pointed out by the green arrow is a traffic sign, whereas the candidate identified by the red arrow is not.

When localizing in each reference map, the vehicle's object map is built using the stereo implementation of ORB-SLAM3 [6] for odometry estimation and YOLO [4,38] for object detection. The tracked ORB features from ORB-SLAM create the sparse point cloud used for the 3D reconstruction of objects.

Sequences 00, 02, 06, 07, and 09 of the KITTI dataset are tested due to the number of semantic objects and the lack of symmetry. For each of these sequences, SemanticKITTI [3] identifies greater than 100 stationary cars and traffic signs and the objects in the reference map do not contain high levels of symmetry, as symmetry in surrounding areas leads to failure in global localization (see Section 4.5). Sequences 05 and 08 contain more than 100 stationary cars as identified by SemanticKITTI, but are not included in experiments because of symmetry in the surrounding area. This pipeline succeeds in localizing these sequences in the lidar reference map, but fails in the larger aerial reference maps.

Algorithm parameters are tuned on Sequence 00 and reported in Table 3.2. Specifically, the vehicle map is created using objects within $d_{\text{obj}} = 20\,\text{m}$ from the vehicle and uses a fusion radius of $\epsilon_{\text{fus}} = 3\,\text{m}$. The restricted vehicle map $\mathcal{M}^{r}_{\text{veh}}$ contains up to $r = 75$ objects. No submaps are used for the lidar reference maps, but the aerial reference maps are split into either $k = 2$ or $k = 4$ submaps with no overlap, depending on the total number of reference map objects. Reference maps split into two submaps (Sequences 02, 06, and 09) are split in half along the y-axis, and the remaining sequences (Sequences 00 and 07) are split in half along each axis. Candidate registrations must contain at least $\tau_{\text{in}} = 12$ inliers which are all at a minimum $d_{\text{in}} = 10\,\text{m}$ apart and the threshold to consider associations geometrically consistent is $\epsilon_{\text{reg}} = 2.5\,\text{m}$. The RMSE threshold begins at $\tau_{\text{RMSE},0} = 6\,\text{m}$, but increases by $\alpha_{\text{RMSE}} = 2\,\text{m}$ for every $d_{\text{RMSE}} = 500\,\text{m}$ traveled since the last localization event in order to account for drift. Additionally, the RMSE value is calculated using only parking space objects because the sparsity of traffic signs leads to large RMSE values. Successfully localized trajectories are visualized in Figure 3-10.

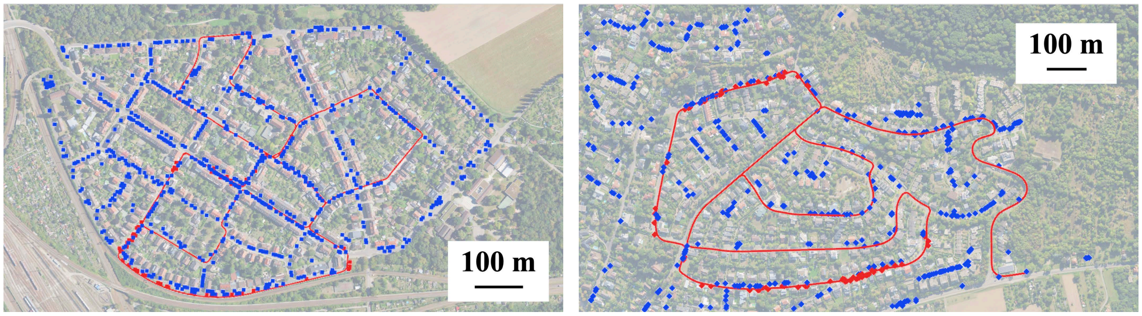


Figure 3-10. The vehicle map and trajectory (red squares and lines) have been successfully localized in reference maps (blue squares) created from Google Satellite images for KITTI Sequences 00 (left) and 02 (right). The background images are provided for reference.

3.3 Conclusion

This pipeline localizes a vehicle in a reference map by geometrically aligning the reference map and the vehicle map. Experiments are conducted using the KITTI benchmark and the Katwijk Beach Planetary Rover dataset in order to demonstrate the pipeline's ability to localize in unstructured environments, view-invariance, robustness to outliers, and drift reduction. The results of these experiments are reported and discussed in Chapter 4.

Table 3.2. Parameter values for experiments with the Katwijk and KITTI datasets. All parameters are described in Section 3.1.

Parameter	Description	Katwijk	KITTI
p_w	Min bounding box width [*pixels*]	20*	0
p_h	Min bounding box height [*pixels*]	20*	0
d_obj	Maximum distance to object to add to vehicle map [m]	12	15
ϵ_fus	Fusion radius for vehicle map [m]	1	3
τ_sight	Min sightings [#]	20	1
r	Max number of objects in vehicle map [#]	–	75
k	Number of submaps [#]	1	Various†
β	Percent overlap [%]	–	0
d_reloc	Required distance between reference and vehicle map objects in guided relocalization [m]	–	10
ϵ_reg	Threshold for considering two associations geometrically consistent [m]	1.5	2.5
d_in	Minimum distance between inliers [m]	0	10
τ_in	Minimum number of inliers [#]	8**	12
$\tau_\text{RMSE,0}$	Initial RMSE threshold for RMSE check [m]	2	6
α_RMSE	Increment to increase RMSE threshold after every d_RMSE traveled [m]	0	2
d_RMSE	Required distance traveled to loosen threshold for RMSE check [m]	–	500
α	Ratio tolerance of smallest RMSE value	0.1	0.1
r'	Maximum number of objects in vehicle map to calculate RMSE for guided relocalization [#]	–	150
δ_RMSE	Minimum difference between candidate and current transformations RMSE for guided relocalization [m]	0.05	0.05
$\delta_\text{T,0}$	Initial maximum difference in translation between candidate and current transformations RMSE for guided relocalization [m]	15	15
$\delta_{\theta,0}$	Initial maximum difference in rotation between candidate and current transformations RMSE for guided relocalization [°]	15	15
$d^{'}$	Required distance traveled between localization events to loosen transformation similarity requirements [m]	500	500
α_T	Increase to δ_T for every $d^{'}$ traveled [m]	15	15
α_θ	Increase to δ_θ for every $d^{'}$ traveled [°]	15	15

*The bounding boxes were additionally required to have a height-to-width ratio of less than 0.25 and to contain some area of the image outside the leftmost and rightmost 200 pixels (see Section 3.2.1).

**Traverse 3 only required 6 inliers due to the reduced size of the reference map.

†Lidar reference maps have only one submap. Aerial reference maps with less than 750 objects have two submaps, otherwise they have four (see Section 3.2.2).

Chapter 4

Results

Each of the four claims regarding this pipeline are tested and analyzed with real data. Global localization is achieved on both the Katwijk Beach Planetary Rover dataset [17] and the KITTI dataset [15] in order to demonstrate success in both urban and non-urban environments (see Section 4.1). Experiments in Section 4.2 test the pipeline's robustness to outliers by localizing and relocalizing in the KITTI dataset using two different reference maps with highly different outlier ratios. Further, tests on the Katwijk dataset demonstrate the view-invariance characteristic of the pipeline (see Section 4.3). Drift reduction is the final claim which is demonstrated on the KITTI dataset and reported in Section 4.4.

4.1 Unstructured and Structured Environments

This pipeline can successfully globally localize in structured and unstructured environments alike. The Katwijk dataset was chosen to exemplify the pipeline's capability in unstructured environments because it does not contain structure typical of urban environments (e.g., roads, traffic signs, lane markings). It is demonstrated that the geometry of the small, medium, and large rocks are sufficient to globally localize in an unstructured environment. Traverses 1, 2, and 3 of the Katwijk dataset are split into 8, 6, and 5 five-minute segments, respectively. Each of these segments is treated as a trial to test the ability to globally localize in the reference map. The five most

Table 4.1. Error statistics describing the pipeline's global localization performance in unstructured environments on the Katwijk dataset. Errors are reported in 2D.

Traverse	Part	Global Loc Position Error [m]	Objects to Localize [#]	Objects in Ref Map [#]	Length of Ref Map [m]
1	1	0.68	8	212	1000
1	8	0.97	9	212	1000
3	1	0.65	9	45	110
3	2	1.4	12	45	110
3	4	0.58	7	45	110

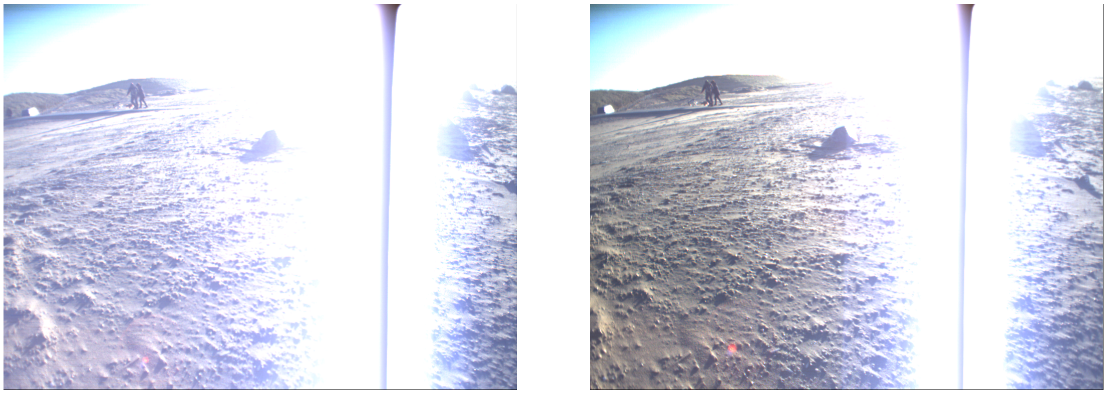


Figure 4-1. Left and right stereo images in Traverse 3, Part 3 of the Katwijk dataset. Sunlight causes glares in the images which makes detection of rocks difficult.

accurate global localizations are reported in Table 4.1. Seven segments do not achieve global localization because the rover does not see a sufficient number of rocks and two additional segments do not achieve global localization because the rover does not identify a sufficient number of inliers. Two segments (Traverse 1, Part 7 and Traverse 3, Part 3) fail (achieve an incorrect global localization) because of misclassifications due to harsh lighting (see Figure 4-1). Despite these challenges, the registration error for global localization is as low as 0.58 m on Traverse 3, Part 4, when localizing in a reference map with 45 objects spanning approximately 110 m. When looking at a larger reference map with 212 objects spanning roughly 1 km, global localization on Traverse 1, Part 1 achieves sub-meter level accuracy of 0.68 m. In Traverse 1, Parts 1 and 8, the rover only needed to identify 8 and 9 objects in order to localize in a map of 212 objects.

In addition to achieving global localization in unstructured environments, the pipeline also successfully globally localizes in structured environments, as demonstrated by experiments with the KITTI dataset. As described in Section 3.2.2, experiments using the KITTI dataset were conducted on Sequences 00, 02, 06, 07, and 09 using two different reference maps. One reference map was created using ground-view lidar scans and the other was created by hand-labeling aerial georeferenced images. Localization in each reference map is achieved using parking spots reconstructed by identifying cars and traffic signs. Table 4.2 reports the global localization accuracy for each of the tested sequences. It is important to note that while the guided relocalization mode is tested on the KITTI dataset, Table 4.2 reports only global localization accuracy in order to compare the pipeline's performance with the Katwijk dataset.

Table 4.2. Error statistics describing the pipeline's global localization performance in structured environments on the KITTI dataset using an aerial reference map created from Google Satellite images. Errors are reported in 2D.

KITTI Seq. [#]	Global Loc Position Error [m]	Objects to Localize [#]	Objects in Ref Map [#]	Size of Ref Map [km^2]
00	7.1	60	942	0.73
02	1.3	68	471	0.43
06	4.1	65	741	0.5
07	2.8	69	942	0.73
09	8.8	115	493	0.70

KITTI Sequence 02 has the lowest global localization error of 1.3 m and the lowest number of objects in the reference map, but Sequence 09 has the highest global localization error of 8.8 m and a very similar number of objects in the reference map. Thus, the number of objects in the reference map is not the primary contributor of the global localization error. Instead, the accuracy of the vehicle map and the amount of symmetry in the reference map contribute more heavily to the size of localization error.

In order to globally localize in KITTI Sequence 09, 115 objects are seen by the ground vehicle, as compared to the 60-69 objects seen in the four other sequences.

This occurs for two reasons. First, the trajectory spans the two submaps used for global localization, so it takes time for sufficient number of inliers to be found with just one of the two reference submaps. Second, there are a few objects identified by the vehicle which are not in the reference map and are located a significant distance from the other objects in the reference map. Thus, the RMSE value is not within the threshold early in the trajectory.

When comparing the global localization error statistics for the Katwijk dataset and the KITTI dataset, the Katwijk dataset clearly requires fewer objects to localize and achieves more accurate global localization. There are two primary causes. First, the Katwijk rover moves significantly slower than the KITTI ground vehicle and sees fewer objects in each frame, which leads to more accurate 3D reconstruction of the objects. Second, the parameters for the KITTI dataset make the pipeline more conservative in accepting a registration because outliers and symmetry are anticipated. Overall, testing on these two datasets demonstrate the pipeline's ability to localize in both structured and unstructured environments.

4.2 Robustness to Outliers

The KITTI dataset was chosen to demonstrate the pipeline's robustness to outliers. The robustness to outliers is quantified by comparing localization accuracy between localizing the KITTI ground vehicle in a reference map created by a lidar scan and a reference map created from a satellite image taken years apart. Traffic signs and parking spaces are the only two classes used to localize the vehicle in a reference map. Parking spots, however, are identified by reconstructing 3D centroids of cars, which provides a noisy estimate of parking spot locations. More details about the experiment setup are in Section 3.2.2. The reference map created from an outdated Google Satellite image poses a more challenging problem than the reference map created from a lidar scan taken at the same time as the stereo images were created. This is both due to the larger size of the aerial map as well as the number of outliers.

To compare the two maps, there are two different object outlier measurements:

reference map object outliers and vehicle map object outliers. The vehicle map object outlier percentage is the percentage of objects in the vehicle map created from ground truth (i.e., no drift) which do not have corresponding objects in the reference map when using the ground truth transformation. The reference map object outlier percentage is the percentage of objects in the reference map which do not have corresponding objects in the vehicle map created from ground truth when using the ground truth transformation. For two objects to correspond, they must be within $\epsilon_{\text{reg}} = 2.5\,\text{m}$ (defined in Table 3.2) of each other and each object can correspond to at most one object. For the five tested KITTI sequences, the vehicle map object outlier percentage ranges from 42-64 % for the lidar reference map (see Table 4.3). Given that the 360-degree lidar sensor and front-facing stereo cameras are on the same vehicle and drive the same trajectory, the lidar sensor will see all of the cars which the stereo cameras see and additional cars which may be beside or behind the vehicle. In other words, the vehicle map outliers with respect to the lidar reference map are due to misclassifications and reconstruction errors. For each KITTI sequence, the vehicle map object outlier percentage is higher with respect to the aerial reference map than with respect to the lidar reference map, ranging from 66-82 % (see Table 4.3). These outliers are due to misclassifications and reconstruction errors as well as changes in the environment.

The reference map outlier percentages indicate the scale of the reference map compared to the scale of the vehicle map in addition to the map errors and environment changes. The reference map object outlier percentage is as high as 97 % and all values are above 87 % for the aerial maps (see Table 4.4). In other words, the registration module can accurately identify accurate associations even though as few as 3 % of the objects in the reference map are viewed by the vehicle along the entire trajectory.

To compare the performance of localization in each reference map, the quality of accepted registrations is considered by plotting the pose estimate error of all localization events in Figure 4-2. It can be observed that Sequences 00 and 07 have the lowest registration error with averages of 4.4 m and 2.5 m when localizing in the aerial reference map. These errors are smaller than the average position error for [61] and [12]

Table 4.3. Error statistics for localizing the KITTI ground vehicle in aerial and lidar reference maps using only parking spaces and traffic signs as semantic objects. The reported position and orientation errors are in 2D for the aerial case and 3D for the lidar case to mirror the dimension of the reference maps.

	KITTI Seq. [#]	Average Position Error [m]	Average Orientation Error [deg]	Distance to Localize [m]	Trajectory Length [m]	Veh Map. Object Outliers [%]	Ref Map. Object Outliers [%]	Objects in Ref. Map [#]
aerial ref. map	00	5.7	1.4	276	3724	80	87	942
	02	10.7	0.7	845	5067	82	90	471
	06	7.1	0.7	975	1233	66	97	741
	07	1.9	1.2	373	695	81	96	942
	09	11.8	0.8	1362	1705	80	94	493
lidar ref. map	00	4.3	2.1	233	3724	52	47	543
	02	10.9	1.3	717	5067	42	52	315
	06	6.9	0.7	941	1233	55	71	119
	07	3.7	1.5	153	695	51	53	180
	09	10.1	1.0	956	1705	64	68	167

by a factor of 10 (see Table 4.5), though methods such as [33] and [5] have superior accuracy on the KITTI dataset. In addition to highly accurate average registration error, tight bounds on the error for Sequences 00, 06, 07, and 09 demonstrate that the registrations are consistently accurate as a whole. Table 4.4, discussed further in Section 4.4, provides more details on the average, median, and standard deviation of relocalization events for each sequence and each reference map. Sequence 02, however, does contain many relocalization error outliers and demonstrates generally lower accuracy. This is attributed to a large amount of drift in the trajectory, as the objects are less dense and so the prepared vehicle map, $\mathcal{M}^r_{\text{veh}}$ as defined in Section 3.1.3, itself contains large amounts of drift.

Table 4.3 and Figure 4-2 combined compare performance on the KITTI dataset using each of the reference maps. When analyzing these error statistics, it is important to understand two key differences between the aerial and lidar maps, apart from the viewpoint difference and number of outliers. First, since Google Satellite only provides 2D data, error statistics corresponding to the aerial reference map are reported in

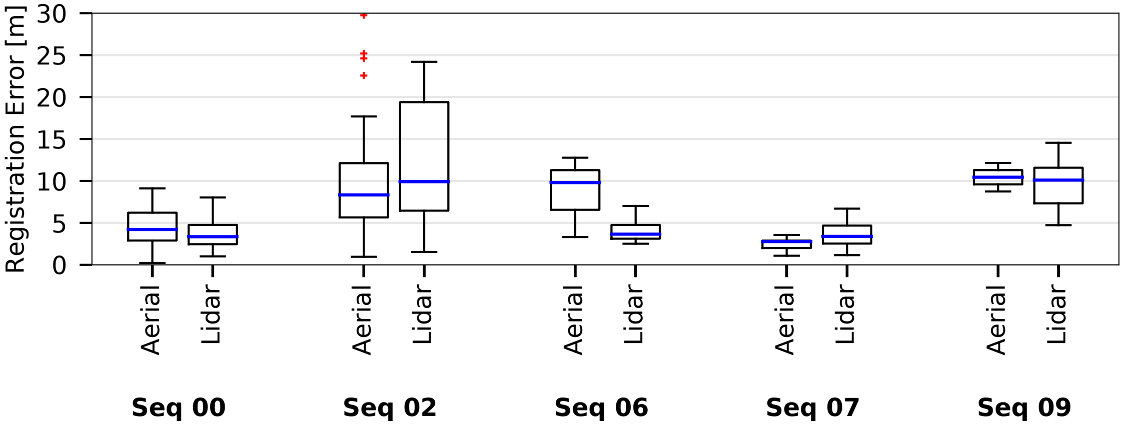


Figure 4-2. Registration accuracy of each KITTI sequence and each reference map (aerial and lidar). Registration accuracy is defined as the pose estimate error at each localization event. Errors are comparable across reference map cases, illustrating the framework's robustness to outliers and viewpoints. Errors are reported in 2D for the aerial case and 3D for the lidar case to mirror the dimension of the reference maps.

2D whereas error statistics corresponding to the lidar reference map are reported in 3D. Second, the time to localize is different across each reference map for the same sequence. Therefore, the number of relocalization events are different across each reference map and the portion of the trajectory considered in the average position error statistic differs in length. Thus, it is possible that a region of the trajectory which is more challenging for the SLAM pipeline is included in the average error statistics for one of the reference maps, but not the other, therefore skewing the error statistics.

Comparing localization accuracy of each sequence across each reference map provides a method to quantify the effects of outliers on the pipeline. The average position errors recorded in Table 4.3 demonstrate how similarly the pipeline performs on the lidar and aerial reference maps. The pipeline performs most similarly across each reference map for Sequences 00 and 07. The difference in average position error between the two maps is 0.2 m for each sequence. Sequences 00, 07, and 09 have differences in error between each reference map of between 1.4 and 1.8 m. Specifically looking at Sequences 07 and 09, global localization in the lidar map occurs 220 and 406 m earlier than localizing in the aerial reference map. All of the other sequences localize in the aerial map within 4 to 128 m of localization in the lidar map. This difference in localization time affects the final average position error. Additionally, the Sequence 09 trajectory has large vertical gain, as the vehicle is driving on a hill. Inaccuracies due to projecting the vehicle map onto a 2D-plane in order to align with the 2D aerial reference map can cause larger position error. It is expected that the remainder of the difference in accuracy is due to the configuration of the object outliers, which may be skewing the transformation by associating cars which are not in the same parking spot. Despite these differences, the pipeline performs similarly when localizing in the aerial reference map or the lidar reference map.

4.3 Viewpoint Variations

The use of semantic maps and a maximum clique data association algorithm make the pipeline view-invariant. In other words, the reference and vehicle maps can be created from any perspective, as long as objects are still identifiable and can be reconstructed. This is exemplified by both the KITTI dataset and the Katwijk dataset.

The two reference maps considered by the KITTI dataset differ in modality (lidar vs Google Satellite images) and viewpoint (ground and aerial). As discussed in Section 4.2, the pipeline successfully localizes in each of these reference maps for all of the tested KITTI sequences. While Section 4.2 focuses on the difference in outlier ratios, these two reference maps also differ in modality and viewpoint.

The trials for the Katwijk dataset described in Section 4.1 use a ground truth reference map created from GPS and provided by the dataset. These reference maps cannot be said to be built from a specific viewpoint. However, in addition to localizing in the ground truth reference map, Katwijk Traverse 3 was used to create a reference map from the extreme opposite viewpoint of the vehicle's view. Given that Traverse 3 is an "out-and-back" trajectory (see Figure 3-6), the first half of the traverse (Parts 1, 2, and 3) is used to create a reference map into which the second half is localized. In other words, the two halves of the traverse see the same objects, but from an extreme difference in viewpoint. This scenario is challenging to image-based methods, which are likely to fail due to sensitivity to viewpoint [29].

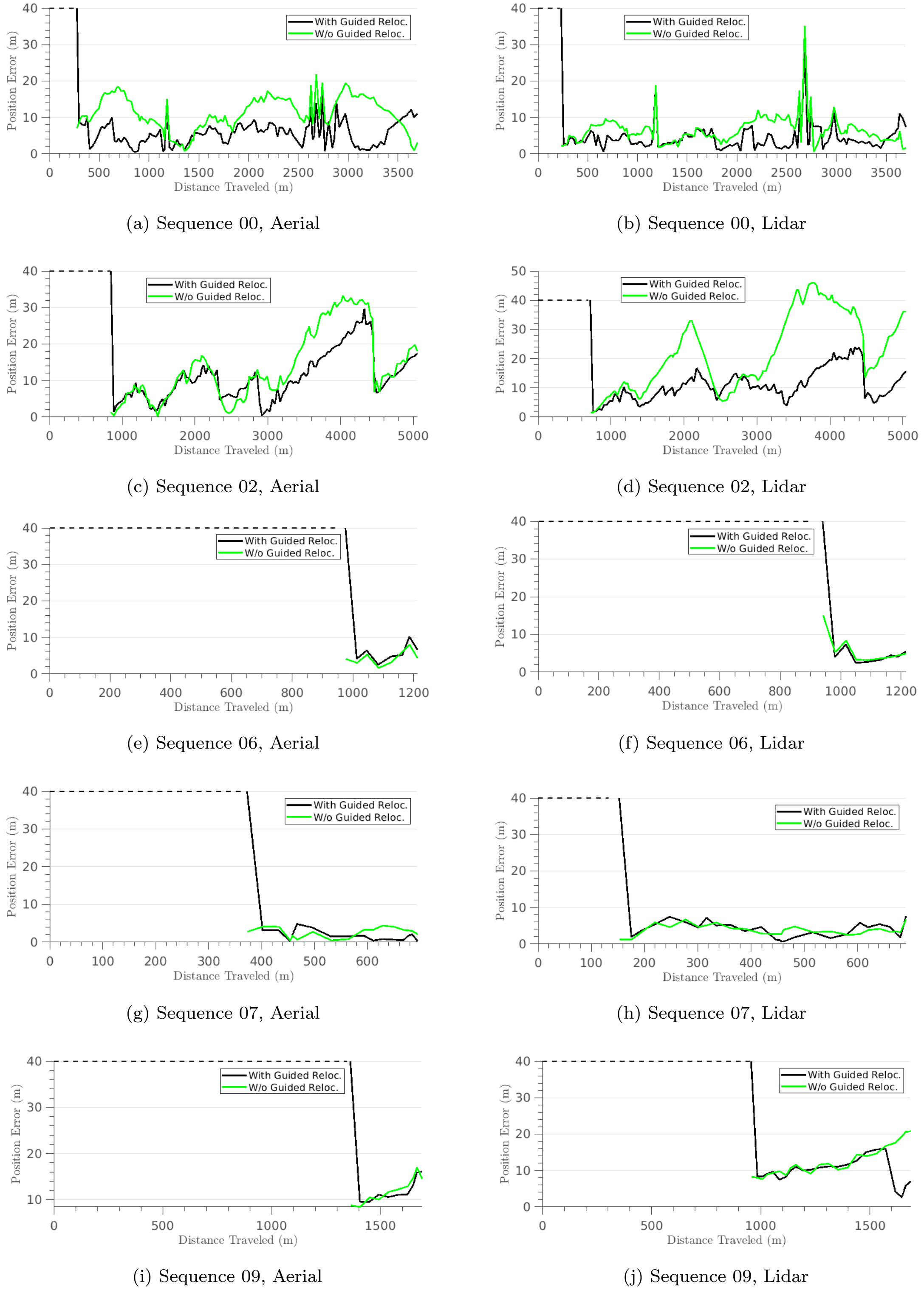
In this pipeline, the object map representation and maximum clique-based association formulation cause the framework to be view-invariant and enables it to localize with 1.2 m accuracy within a reference map spanning approximately 110 m. The error is from inaccuracies in the trajectory and object centroid reconstruction.

4.4 Drift Reduction

This pipeline effectively mitigates error due to drift by using a guided relocalization mode to update pose estimates. The framework's ability to relocalize and reduce effects of drift is demonstrated using the KITTI dataset.

There are two primary sources of error in the pipeline: noisy object maps and noisy registrations. Inaccuracies in the object maps are caused by errors during 3D centroid reconstruction and local pose estimates. The 3D centroid reconstructions are either hand-labeled and prone to human error (aerial reference map) or estimated using point clouds generated by either lidar sensors (lidar reference map) or ORB-SLAM3 tracked features (vehicle map). The accuracy of vehicle pose estimates depends on various factors such as sensors, number of loop closures, and drift from the SLAM module. Incorrect pose estimates distort the vehicle map and directly influence the localization accuracy in the pipeline. In order to increase accuracy of the pipeline, relocalization events (i.e., receiving pose corrections) must be frequent and accurate. After global localization, infrequent relocalizations would allow drift to accumulate between events and contribute toward inaccurate pose estimates across the entire sequence.

To quantify the benefit of guided relocalization, the position error of the full pipeline and the position error using only the global localization transformation are compared. Stated simply, the pipeline is run with and without guided relocalization. Qualitatively, Figure 4-3 illustrates the position error over distance traveled for each of the sequences and each of the reference maps. Due to the length of Sequence 00 and Sequence 02, the effects of using guided relocalization can clearly be seen. In Sequence 00, large spikes in error around 2700 m are due to error in the SLAM



(a) Sequence 00, Aerial

(b) Sequence 00, Lidar

(c) Sequence 02, Aerial

(d) Sequence 02, Lidar

(e) Sequence 06, Aerial

(f) Sequence 06, Lidar

(g) Sequence 07, Aerial

(h) Sequence 07, Lidar

(i) Sequence 09, Aerial

(j) Sequence 09, Lidar

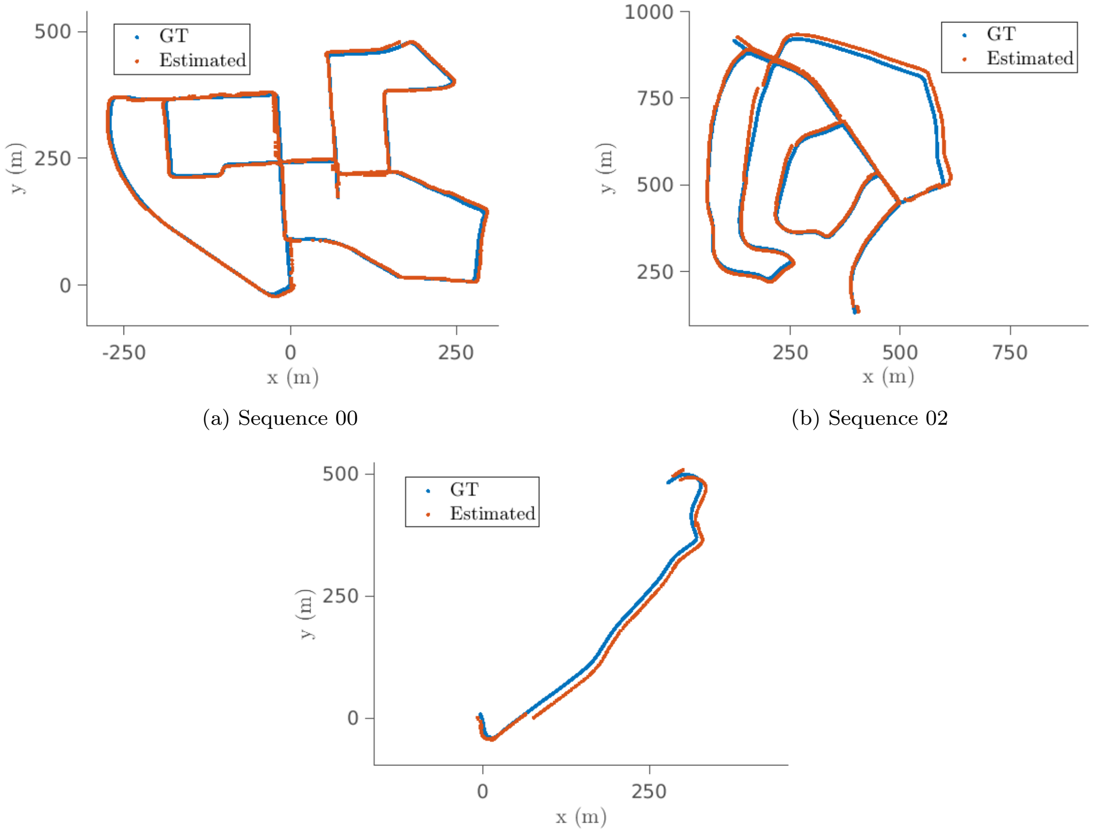
Figure 4-3. Estimated pose error with and without guided relocalization when localizing the KITTI ground vehicle in the aerial and lidar reference maps for each of the tested KITTI sequences. The black line demonstrates the full ability of the pipeline whereas the green line is calculated as if the only accepted transformation is the initial global localization transformation. The error due to drift accumulated after global localization is most obvious on Sequences 00 and 02 because they are the longest sequences.

system due to a sharp turn. For each of the sequences and reference maps, Figure 4-3 illustrates how the position error with guided relocalization is consistently lower than the position error when only using global localization. There are a few instances at which the error with guided relocalization is larger than that without it. These instances generally occur when, by chance, the drift in the trajectory is in such a direction that it makes the global localization transformation more accurate, but the most recently accepted guided relocalization transformation worse. A few of these instances may occur because of a poor accepted registration which is soon corrected.

The estimated and ground truth trajectories after global localization for KITTI Sequences 00, 02, and 09 using the lidar reference map are illustrated in Figure 4-4. The effects of the updated transformations in guided relocalization can be seen in this figure. Most clearly, in Sequence 09, the ground vehicle is traveling to the southwest and at roughly $(x, y) = (75, 0)$, there is an accepted transformation which makes the estimated trajectory (in orange) align better with the ground truth trajectory (in blue). Similar occurrences can be seen in Sequence 00 and Sequence 02.

Error statistics for drift reduction are reported in Table 4.4. Figure 4-5 illustrates the correlation between the frequency of relocalization events and the average position error over the entire sequence. There is a clear correlation that sequences with a low frequency of localization events have high average error. There may be a lower frequency of relocalization events for a couple reasons. For example, large amounts of drift skewing the vehicle map or symmetry in the restricted reference map will make it difficult to find an acceptable registration. Alternatively, the vehicle may not see many objects for a significant period of time and, therefore, will not be able to find an updated transformation, but will continue to accumulate drift. Sequence 00 and Sequence 09 have the highest average error (see Table 4.3). For these sequences, the accepted transformations occur at the lowest rate, which allows for large amounts of drift to be introduced into the pose estimate. For this reason, the correcting effects of newly accepted transformations are most pronounced for these sequences in the trajectory illustrations in Figure 4-4.

The longest sequences, Sequences 00 and 02, unsurprisingly demonstrate the high-



(a) Sequence 00

(b) Sequence 02

(c) Sequence 09

Figure 4-4. Ground truth poses (blue) are plotted with pose estimates localizing in the lidar reference map (orange) at every timestep after global localization for KITTI Sequences 00, 02, and 09. Accumulation of drift over time and the removal of drift after a new relocalization event is most pronounced in Sequences 02 and 09. For example, in Sequence 09, the vehicle moves southwest and accumulates drift until the relocalization event around $(x, y) = (75, 0)$ makes the estimated pose align significantly better with the ground truth pose.

Table 4.4. Performance statistics of guided relocalization for the KITTI ground vehicle. Guided relocalization successfully removes drift in 9 of the 10 trials. As the frequency of relocalization events increases, the average position error with guided relocalization decreases.

	KITTI Seq. [#]	Avg. Position Error with GR [m]	Avg. Position Error w/o GR [m]	Percent of Drift Removed by GR [%]	Avg. Reloc. Error [m]	Median Reloc. Error [m]	Std Reloc. Error [m]	Avg. Dist between Reloc. Events [m]
aerial ref. map	00	5.7	11.1	48.6	4.4	4.0	2.3	59.5
	02	10.8	13.9	22.6	9.6	8.0	6.5	103.0
	06	7.1	6.6	−7.1	8.6	3.3	4.8	85.6
	07	1.9	2.4	20.7	2.5	2.7	0.90	46.0
	09	11.8	12.0	1.7	10.4	8.8	2.4	171.5
lidar ref. map	00	4.3	6.3	31.2	3.8	3.3	1.7	41.6
	02	10.9	22.0	50.5	12.0	9.7	6.8	92.6
	06	6.9	7.2	4.1	4.2	3.3	2.0	73.1
	07	3.7	3.9	5.9	3.7	2.9	1.8	45.2
	09	10.1	12.7	20.5	9.6	8.5	3.1	68.1

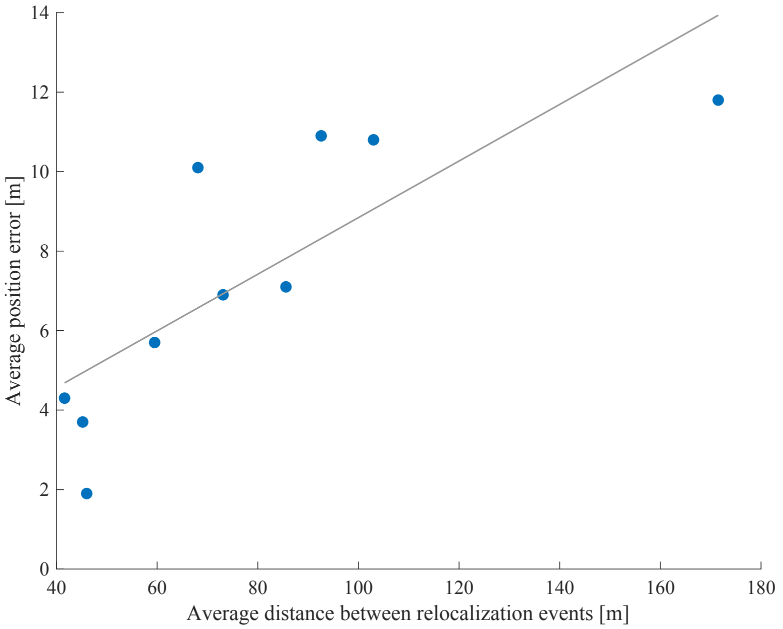


Figure 4-5. Average position error versus average distance between relocalization events for each of the five tested KITTI sequences and each of the two reference maps. The line of best fit is included to demonstrate the positive correlation between the two variables.

est percentage of drift removed by guided relocalization. The percentage of drift removed is negative for Sequence 06 when localizing in the aerial map. In this case, guided relocalization made the pose estimates worse. This is likely due to the short distance the vehicle traveled after global localization (258 m) and the first localization being more accurate than the following. For this case, only 3 localization and relocalization events occurred.

The accuracy of each individual localization event is highest for Sequence 00 and Sequence 09 for each reference map, as is the standard deviation. In general, the median relocalization event error is lower than the average relocalization event error. This implies that the majority of relocalization events are highly accurate, but there are often a few less accurate accepted registrations for each sequence which increase the average.

The quantitative results in Table 4.4 and qualitative results in Figure 4-3 and Figure 4-4 demonstrate the high effectiveness of the guided relocalization mode, especially on longer trajectories.

4.5 Discussion

Table 4.5 lists evaluation results of localizing the KITTI ground vehicle in an aerial reference map compared to prior art which similarly tests air-ground localization on the KITTI benchmark. The 2D localization error for each of the five tested sequences is reported twice: once using stereo odometry and once using ground truth odometry. Using ground truth odometry provides the maximum achievable accuracy of the pipeline. Overall, while prior art achieves good accuracy, these methods are restricted to urban environments. This pipeline was designed to work in both urban and non-urban environments and therefore makes no assumptions about roads or lane markings. As a result, the pipeline leverages less information than competing approaches.

The achieved accuracy is competitive to prior art in structured environments. This pipeline outperforms Floros et al. [12] on Sequences 00 and 02, and Yan et al. [61] on

Table 4.5. Aerial-ground localization comparison on the KITTI benchmark for position error and localization time. Dashed line "–" indicates not reported or not localized successfully.

KITTI Seq. [#]	Metric	Miller [33]	Yan [61]	Brubaker [5]	Floros [12]	Ours (GT)	Ours (stereo)
00	Error [m]	2.0	>10	2.1	>10	3.9	5.7
	Time [s]	54.6	–	22	–	42	39
02	Error [m]	9.1	–	4.1	>20	3.9	10.8
	Time [s]	71.5	–	26	–	75	75
06	Error [m]	–	>10	–	–	2.2	7.1
	Time [s]	–	–	–	–	84	90
07	Error [m]	–	>10	1.8	–	2.6	1.9
	Time [s]	–	–	26	–	60	57
09	Error [m]	7.2	>10	4.2	–	2.0	11.8
	Time [s]	75	–	24	–	75	135

Sequences 00, 06, and 07. While in Sequences 00, 02, and 09, the accuracy using stereo SLAM does not surpass Brubaker et al. [5] or Miller et al. [33], these methods assume an urban structure. These error statistics demonstrate the comparable accuracy to other methods regardless of the strict and practical assumption of an unstructured environment.

In general, symmetry in reference maps is challenging for this pipeline. If the geometry of objects in different regions of the reference map look similar, the pipeline may globally localize to the wrong pose. It is the symmetry in the geometry of objects, not the symmetry in road structure, which affects the algorithm's success. Sequence 06 exemplifies this idea, as the symmetry of the road structure makes this sequence highly challenging [5]. However, despite this symmetry, the framework is able to successfully localize because there was little symmetry in parking space and traffic sign locations.

4.6 Conclusion

Experiments on the KITTI and Katwijk datasets demonstrate the four primary characteristics of this pipeline: the ability to localize in urban and non-urban environments alike, view-invariance, robustness to changes in the environment, and the ability to reduce effects drift. Overall, the pipeline demonstrates comparable accuracy in urban environments to other state-of-the-art global localization methods. The following chapter concludes this thesis and discusses ideas to increase scalability and generalizability of the pipeline in the future.

Chapter 5

Conclusion

5.1 Summary of Contributions

The current state of the art global localization, long-term localization, place recognition, and loop closure detection focus on achieving localization in structured GPS-denied environments. Few methods consider non-urban environments, but are limited by either dense point clouds, external hardware, or rely on the vehicle being located on non-urban roads (see Section 2.1.5). As localization technologies are brought into applications such as search-and-rescue or military reconnaissance missions, these methods will be required to localize a vehicle in unstructured environments with few features and no roads. The assumption of an unstructured environment produces a challenging problem because the search space of the algorithm increases and information available in urban settings is no longer present. Because of these real applications, this pipeline is presented for global localization and guided relocalization using potentially-outdated semantic maps created from various viewpoints. Specifically, the semantic map representation and maximum clique data association algorithm enable the view-invariance characteristic, robustness to outliers, and the ability to localize in unstructured settings. Experiments with the Katwijk dataset [17] and the KITTI benchmark [15] demonstrate these three properties of the pipeline as well as the relocalization technique to reduce drift after global localization has been achieved.

5.2 Future Work

Future directions of work could focus on increasing the efficiency of the pipeline such that the pipeline can globally localize within a larger region, in less time, and in more generic settings with fewer semantic objects. The following sections detail potential ideas to leverage in order to increase the pipeline's scalability and generalization.

5.2.1 Descriptors

Experiments with the Katwijk dataset demonstrate localization within a map of a 1km stretch of artificial rocks along the beach. Tests on the KITTI dataset demonstrate localization within a semantic reference maps spanning up to 0.73km^2. In order for the pipeline to efficiently localize in a larger reference map, additional adjustments are needed. Descriptors are one way to increase the scalability of this pipeline.

Currently, when executing the data association algorithm in the global localization mode, an all-to-all association scheme is used within each class (see Section 3.1.4). In order to decrease the run time of the maximum clique algorithm, it would be useful to begin with fewer initial associations than this all-to-all scheme. In order to do so, descriptors can be leveraged. For example, [28] and [27] use random walk descriptors and [64] uses histogram descriptors encoding distance to nearby objects of specific classes in order to prune potential associations early in the data association phase of the frameworks.

To test this idea, histogram descriptors were implemented in the pipeline. The following details how these descriptors were created. For each object in the vehicle map and each object in the reference map, objects within a radius r in the same map are considered. In the creation of the descriptors, the class of the object and nearby objects are ignored. For each of the n objects within r meters from the object in consideration, define the location of object i to be p_i and the location of the object in consideration to be p_{cur}. Let $\bar{v}_{\text{i,cur}}$ be the vector from p_{cur} to p_i. Starting with an arbitrarily chosen object as p_1, order the remaining objects $p_2, \ldots, p_{\text{n}}$ in order of increasing clockwise angles between $\bar{v}_{\text{1,cur}}$ and $\bar{v}_{\text{i,cur}}$.

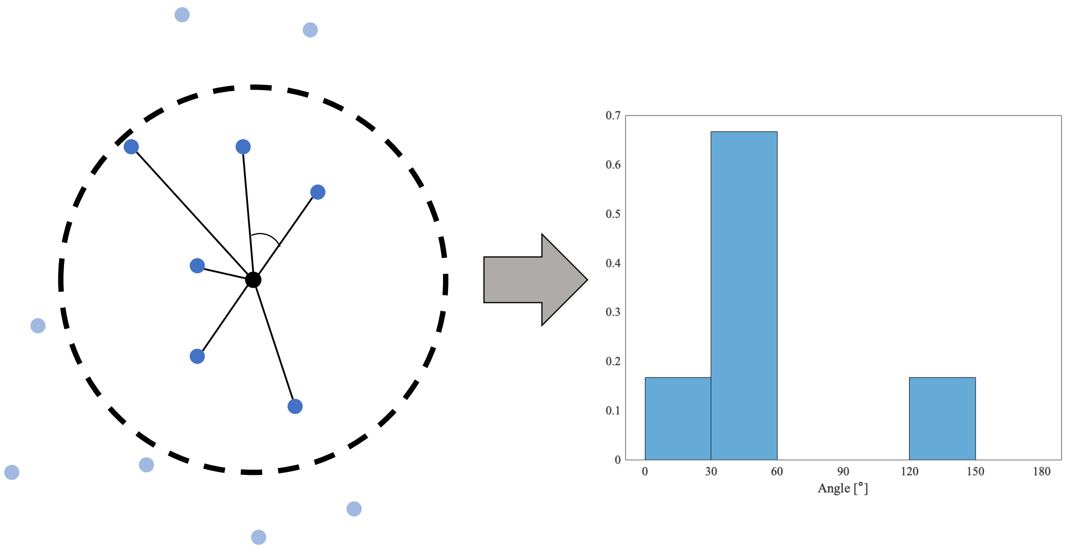


Figure 5-1. The blue and black points represent the 2D locations of objects in a reference map. The histogram descriptor is being created for the black point. A dashed circle of radius r is drawn around the object of interest and the angles between the dark blue objects inside of this boundary are calculated and described by the histogram to the right. The number of bins in the histogram and radius r are parameters which can be tuned.

After the ordering of the nearby objects, the angles between $\bar{v}_{\text{i,cur}}$ and $\bar{v}_{\text{i+1,cur}}$ are represented as a normalized histogram with n_{bin} equal-sized bins dividing 0 to 180 degrees. The descriptor for object o_i is represented by a vector and is denoted by H_i^{ref} if o_i is in the reference map and H_i^{veh} if o_i is in the vehicle map. Figure 5-1 illustrates an example descriptor.

As the pipeline runs, the descriptors are continually created and updated for every object in the reference and vehicle maps. During the data association algorithm, within each class, objects in the vehicle map are associated with objects in the reference map that have the most similar descriptors. The similarity level between descriptors is defined as the 2-norm between the two descriptors, $||H_i^{\text{veh}} - H_j^{\text{veh}}||$. The initial associations for each object in the vehicle map are taken to be the most similar α percent of objects in the reference map.

Tests with this definition of descriptors on KITTI Sequence 00 demonstrated that this formulation is not useful (and sometimes slightly harmful) to the global convergence. With a large enough alpha, the pipeline performed as usual, but with no efficiency advantage. As α decreases, the majority of the correct associations are kept in the initial associations, but enough are lost that pipeline will not reliably find the correct registration and recognize that it is the correct registration (e.g., the correct registration may be a candidate registration, but will be thrown out for too few inliers).

While the discussed histogram of angles has been demonstrated to not be an effective descriptor for the pipeline, a different descriptor may prove useful in increasing scalability. For example, a histogram of distances to nearby objects or the number and class of nearby objects may prove useful.

5.2.2 Approximate Maximum Clique Algorithm

The pipeline assumes the data association method is a maximum clique algorithm. All experiments are run and parameters are tuned for the Parallel Maximum Clique (PMC) [42], which provides the exact maximum clique solution to the given graph formulation of the problem. In order to reduce computation time and therefore increase scalability, an approximate maximum clique algorithm, CLIPPER [30], can be used. CLIPPER takes the same inputs as PMC, but estimates the maximum clique in the graph whereas PMC is guaranteed to find the largest maximum clique. Thus, while CLIPPER can become stuck in local minima, the computational time compared to PMC is much lower, especially as the number of objects in each map increases.

As in PMC (see Section 3.1.4), pairs of potential associations are evaluated based on geometric consistency. A consistency graph is constructed using potential associations as nodes, then edges between two associations are added if the two associations are geometrically consistent. In the CLIPPER algorithm, the edges of the graph are weighted using a predetermined function which encodes the quality of the geometric consistency. More specifically, the weight of the edge between two geometrically consistent associations a_i and a_j is a function of $d(a_i, a_j)$ as defined in Section 3.1.4. An affinity matrix, M, is then created such that entry (i, j) of M is the edge weight between association i and association j in the consistency graph. If the two associations are not geometrically consistent (i.e., there is no edge connecting the two associations in the graph), the entry in M is set to zero. The optimal solution can be found using

$$\begin{aligned} \underset{u \in \{0,1\}^n}{\text{maximize}} \quad & \frac{u^\top M\, u}{u^\top u} \\ \text{subject to} \quad & u_i\, u_j = 0 \quad \text{if } M(i,j) = 0,\ \forall_{i,j}, \end{aligned} \tag{5.1}$$

where u is a binary vector of the chosen associations. However, CLIPPER has a significantly faster runtime by relaxing this problem in order to estimate the maximum clique. The relaxed formulation of Equation (5.1) solved by the CLIPPER algorithm is

$$\begin{aligned} \underset{u \in \mathbb{R}^n_+}{\text{maximize}} \quad & F(u) \overset{\text{def}}{=} u^\top M_d\, u \\ \text{subject to} \quad & \|u\| \leq 1 \end{aligned} \tag{5.2}$$

where $\mathbb{R}_+$ is the set of non-negative reals, $\|\cdot\|$ is the ℓ_2 vector norm, and

$$M_d(i,j) \overset{\text{def}}{=} \begin{cases} M(i,j) & \text{if } \ M(i,j) \neq 0 \\ -d & \text{if } \ M(i,j) = 0 \end{cases} \tag{5.3}$$

where $d > 0$ is a positive scalar [30].

CLIPPER was implemented in this pipeline and tested on KITTI Sequence 00, but it was observed that CLIPPER was unable to find the correct registration, despite testing a various number of submaps k and overlap ratios β (as defined in Table 3.2). However, with an initial association scheme using fast point feature histograms (FPFH) descriptors as opposed to an all-to-all scheme, CLIPPER often, but not always, found the correct solution for KITTI Sequence 00. With appropriate parameters and an appropriate descriptor, the pipeline may succeed with an approximate maximum clique algorithm and, thus, increase scalability of the pipeline.

5.2.3 Map Representation

Both the vehicle and reference maps are represented by objects $o_i = (u_i, c_i)$, defined by their 3D centroid $u_i \in \mathbb{R}^3$ and a class, $c_i \in \mathcal{C}$ (described in Section 3.1.3). In many practical applications such as military reconnaissance and search-and-rescue missions, there may be fewer objects which can appropriately be represented as a point. For example, from an aerial viewpoint, a standalone tree can be well-represented by a point, whereas a clump of trees may be better represented by a polygon. Similarly, while a building may be well-represented by a point, a road may be better represented by a line. Thus, an extension of the map representation to include points, lines, and polygons is a natural next step.

Implementation of this concept takes two steps. First, the vehicle and reference map creation must be able to estimate the points, lines, and polygons. Second, the maximum clique data association algorithm must be able to determine geometric consistency for points, edges, and polygons. For simplicity, consider two classes: roads and clumps of trees, represented by edges and polygons, respectively.

A ground vehicle exploring the area to create either the reference or vehicle map must be able to stitch together lines and polygons across frames. To estimate a road as a line, the vehicle must be able to estimate a line, perhaps using image segmentation, which represents any road viewable by the ground vehicle. Like points, the line estimates for the roads can be fused together across frames if they are within some small fusion distance of each other, and the edge must grow as the vehicle can see more of the road. Polygons are likely more challenging. For example, a stereo camera on the ground vehicle will be able to reconstruct a polygon of a clump of trees in the yz-plane. However, if the reference map is created from an aerial perspective, the polygon will be created in the xy-plane. Perhaps, to create a polygon of a clump of trees, from a ground perspective, the vehicle can identify individual trees as points by identifying tree trunks, and then the map can fuse nearby trees together into a polygon.

To create a semantic map from an aerial perspective, image segmentation is the most reasonable solution. Current methods which use image segmentation from an aerial perspective overtrain the classifier on the satellite images to be used in the experiments [33]. Many variations of the U-net algorithm [41] have been developed to work on satellite images. The biggest challenge to get an aerial image segmentation neural network to work for this pipeline is the time-consuming work of annotating relevant data and then training the network on this data.

After the maps incorporate edges and polygons, the data association algorithm must be able to determine geometric consistency for each of the object representations. Denote the midpoint of an edge p_i as m_{p_i} and denote an association matching two edge objects p_i and q_i by $a_i = (p_i, q_i)$. Two associations a_i and a_j can be considered geometrically consistent if $|\sphericalangle p_i p_j - \sphericalangle q_i q_j| \leq \epsilon_1$ and $||(m_{p_i} - m_{p_j}) - (m_{q_i} - m_{q_j})|| \leq \epsilon_2$. The first condition ensures the angle between lines are preserved and the second condition ensures the distance between lines are preserved.

Polygons pose a more challenging geometric consistency problem. One option is to treat the polygon as a collection of edges, and so the geometric consistency definition developed for edges would be required to hold for all of the edges of the polygon. However, if the reference and vehicle maps have chosen to represent the same object with a different number of edges, this method would fail. The area encompassed by the polygon and the centroid could factor into the definition geometric consistency. Although, more issues arise if the vehicle can only see a portion of the entire polygon, whereas the reference map contains the entire polygon. All in all, extending the semantic maps to include different representations other than points is vital to the generalization of this pipeline to real-world off-road scenarios, though it brings with it many new challenges.

5.2.4 Adaptive Vehicle Map

The prepared vehicle map $M^{\mathrm{r}}_{\mathrm{veh}}$ as defined in Table 3.1 is a subset of the full vehicle map, containing the r most recently observed objects. It is important to use a subset of the full vehicle map in order to reduce the size of the data association problem. The subset of the vehicle map is chosen is by using the objects most recently seen by the vehicle in order to reduce the effects of drift on the vehicle map. However, there are other methods with which the prepared vehicle map could be chosen.

In order to test the idea of an adaptive vehicle map, experiments were conducted with a prepared vehicle map consisting of a combination of objects which were associated with reference map objects in the most recently accepted registration and the most recently seen objects. First, if there were n inliers in the most recently accepted registration, these n objects in the vehicle map were chosen and put into the prepared vehicle map. Then, the most recently seen $r - n$ objects which are unique from the n inlier objects are added to the prepared vehicle map as well. The purpose of choosing this scheme was to minimize chances of having unusable candidate transformations during the guided relocalization phase. It is important to note that these experiments were conducted when the guided relocalization mode localized in the full reference map, not the restricted reference map.

The results of these trials demonstrated that the adaptive vehicle map was sometimes harmful to the pipeline. For example, sometimes the most recently accepted registration was of poor quality, but had a high number of inliers. The vehicle map would then include the older objects which were inliers for this poor quality registration made the pipeline continually produce the same poor registration. Additionally, keeping a significant amount of older objects hurt the pipeline's ability to correct for drift. Overall, the pipeline would often become stuck using a poor quality registration due to the large number of inliers and to the number of old objects in the vehicle map.

While this scheme did not prove useful to the pipeline, other constructions of a adaptive vehicle map may increase the effectiveness of guided relocalization and global localization. For example, if the pipeline is in the global localization mode and the localizing vehicle has seen an object of a rare class (e.g., traffic signs in the KITTI dataset were rare compared to parking spaces), this object may be disproportionately useful in localizing the vehicle compared to any one car. Thus, it may be useful to keep this object in the prepared vehicle map for a longer period of time, even if it is not one of the r most recently seen objects. With more time, a better method of preparing the vehicle map for the data association module could likely be found.

5.2.5 Aerial Reference Map Automation

The aerial reference map used in KITTI experiments is created by hand-labeling the identifiable cars and traffic signs from Google Satellite images. This process consists of importing the Google Satellite layer into QGIS [37] and clicking on each car and traffic sign, then importing this data into a text file to be loaded in the pipeline as a reference map. Given the size of the reference maps (see Table 4.3), this process is time-consuming.

Ideally, the process of creating a reference map from a satellite image would be automated using an image segmentation neural network trained on satellite images. Several recent works have focused on aerial classifiers [9, 25] and there are many open source versions of the U-Net algorithm [41] trained for satellite images in urban environments. However, these algorithms may not classify the desired semantic objects or work on unstructured datasets. Additionally, implementing these neural networks are not trivial because the network likely must be retrained to handle satellite images similar to the images used for any specific dataset. Collecting and annotating relevant data and then retraining the neural network is time-consuming. After the neural network is retrained, the masks must be processed to create a reference map in the desired representation. Thus, while automating the creation of a reference map from satellite images would reduce time in the long run, implementing this is time-intensive in the short-term.

5.3 Conclusion

This thesis presented a pipeline for global localization and guided relocalization of a vehicle's pose in unstructured environments using semantic object maps created from various viewpoints. Experiments with the Katwijk dataset and the KITTI benchmark demonstrate the pipeline's view-invariant property, robustness to outliers, and capability of localizing in unstructured environments.

Bibliography

[1] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE TPAMI*, (5):698–700, 1987.

[2] Ioan Andrei Barsan, Shenlong Wang, Andrei Pokrovsky, and Raquel Urtasun. Learning to localize using a lidar intensity map. *arXiv preprint arXiv:2012.10902*, 2020.

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *IEEE/CVF ICCV*, 2019.

[4] Marko Bjelonic. YOLO ROS: Real-time object detection for ROS. `https://github.com/leggedrobotics/darknet_ros`, 2016–2018.

[5] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Map-based probabilistic visual self-localization. *IEEE TPAMI*, 38(4):652–665, 2015.

[6] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE T-RO*, 37(6), 2021.

[7] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *IEEE CVPR*, 2011.

[8] Younghun Cho, Giseop Kim, Sangmin Lee, and Jee-Hwan Ryu. Openstreetmap-based lidar global localization in urban environment without a prior lidar map. *IEEE RA-L*, 7(2):4999–5006, 2022.

[9] J Ding, N Xue, Y Long, GS Xia, and Q Lu. Learning roi transformer for detecting oriented objects in aerial images. In *IEEE CVPR*, 2019.

[10] Lena M Downes, Dong-Ki Kim, Ted J Steiner, and Jonathan P How. City-wide street-to-satellite image geolocalization of a mobile ground agent. In *IEEE/RSJ IROS*, pages 11102–11108, 2022.

[11] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMatch: Segment based place recognition in 3d point clouds. In *IEEE ICRA*, 2017.

[12] Georgios Floros, Benito Van Der Zander, and Bastian Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *IEEE ICRA*, 2013.

[13] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE T-RO*, 2012.

[14] Abel Gawel, Renaud Dubé, Hartmut Surmann, Juan Nieto, Roland Siegwart, and Cesar Cadena. 3d registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation. In *IEEE SSRR*, pages 27–34, 2017.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE CVPR*, 2012.

[16] Matthew Grimes and Yann LeCun. Efficient off-road localization using visually corrected odometry. In *IEEE ICRA*, pages 2649–2654, 2009.

[17] Robert A Hewitt, Evangelos Boukas, Martin Azkarate, Marco Pagnamenta, Joshua A Marshall, Antonios Gasteratos, and Gianfranco Visentin. The Katwijk beach planetary rover dataset. *IJRR*, 37(1):3–12, 2018.

[18] Mahdi Javanmardi, Ehsan Javanmardi, Yanlei Gu, and Shunsuke Kamijo. Towards high-definition 3d urban mapping: Road feature-based registration of mobile mapping systems and aerial imagery. *Remote Sensing*, 2017.

[19] Kaijin Ji, Huiyan Chen, Huijun Di, Jianwei Gong, Guangming Xiong, Jianyong Qi, and Tao Yi. Cpfg-slam: A robust simultaneous localization and mapping based on lidar in off-road environment. In *IEEE IV*, pages 650–655, 2018.

[20] Dong-Ki Kim and Matthew R. Walter. Satellite image-based localization via learned embeddings. In *IEEE ICRA*, pages 2073–2080, 2017.

[21] Giseop Kim, Sunwook Choi, and Ayoung Kim. Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 38(3):1856–1874, 2021.

[22] Jonghwi Kim and Jinwhan Kim. Fusing lidar data and aerial imagery with perspective correction for precise localization in urban canyons. In *IEEE/RSJ IROS*, pages 5298–5303. IEEE, 2019.

[23] Joshua Knights, Kavisha Vidanapathirana, Milad Ramezani, Sridha Sridharan, Clinton Fookes, and Peyman Moghadam. Wild-places: A large-scale dataset for lidar place recognition in unstructured natural environments. *arXiv preprint arXiv:2211.12732*, 2022.

[24] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Alexander Kleiner, Giorgio Grisetti, and Wolfram Burgard. Large scale graph-based SLAM using aerial images as prior information. *Autonomous Robots*, 30(1), 2011.

[25] Wentong Li, Yijie Chen, Kaixuan Hu, and Jianke Zhu. Oriented reppoints for aerial object detection. In *IEEE CVPR*, pages 1829–1838, 2022.

[26] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, pages 791–804. Springer, 2010.

[27] Shiqi Lin, Jikai Wang, Meng Xu, Hao Zhao, and Zonghai Chen. Topology aware object-level semantic mapping towards more robust loop closure. *IEEE RA-L*, 6(4):7041–7048, 2021.

[28] Yu Liu, Yvan Petillot, David Lane, and Sen Wang. Global localization with object-level semantics and topology. In *IEEE ICRA*, 2019.

[29] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE T-RO*, 32(1):1–19, 2015.

[30] Parker C. Lusk, Kaveh Fathian, and Jonathan P. How. CLIPPER: A Graph-Theoretic Framework for Robust Data Association. In *IEEE ICRA*, 2021.

[31] András L Majdik, Yves Albers-Schoenberg, and Davide Scaramuzza. Mav urban localization from google street view data. In *IEEE/RSJ IROS*, 2013.

[32] Bogdan C Matei, Nick Vander Valk, Zhiwei Zhu, Hui Cheng, and Harpreet S Sawhney. Image to lidar matching for geotagging in urban environments. In *IEEE/CVF WACV*, pages 413–420, 2013.

[33] Ian D. Miller, Anthony Cowley, Ravi Konkimalla, Shreyas S. Shivakumar, Ty Nguyen, Trey Smith, Camillo Jose Taylor, and Vijay Kumar. Any way you look at it: Semantic crossview localization and mapping with lidar. *IEEE Robotics and Automation Letters*, 6(2):2397–2404, 2021.

[34] Masafumi Noda, Tomokazu Takahashi, Daisuke Deguchi, Ichiro Ide, Hiroshi Murase, Yoshiko Kojima, and Takashi Naito. Vehicle ego-localization by matching in-vehicle camera images to an aerial image. In *ACCV*, pages 163–173. Springer, 2010.

[35] Teddy Ort, Liam Paull, and Daniela Rus. Autonomous vehicle navigation in rural environments without detailed prior maps. In *IEEE ICRA*, 2018.

[36] Oliver Pink. Visual map matching and localization using a global feature map. In *IEEE CVPR*, pages 1–7. IEEE, 2008.

[37] QGIS Development Team. *QGIS Geographic Information System.* Open Source Geospatial Foundation, 2009.

[38] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[39] Ruike Ren, Hao Fu, Hanzhang Xue, Xiaohui Li, Xiaochang Hu, and Meiping Wu. Lidar-based robust localization for field autonomous vehicles in off-road environments. *Journal of Field Robotics*, 38(8):1059–1077, 2021.

[40] Royston Rodrigues and Masahiro Tani. Are these from the same place? seeing the unseen in cross-view image geo-localization. In *IEEE/CVF WACV*, pages 3753–3761, 2021.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[42] Ryan A Rossi, David F Gleich, and Assefaw H Gebremedhin. Parallel maximum clique algorithms with applications to network analysis. *SIAM Journal on Scientific Computing*, 37(5):C589–C616, 2015.

[43] Turgay Senlet, Tarek El-Gaaly, and Ahmed Elgammal. Hierarchical semantic hashing: Visual localization from buildings on maps. In *IEEE International Conference on Pattern Recognition*, pages 2990–2995, 2014.

[44] Turgay Senlet and Ahmed Elgammal. A framework for global vehicle localization using stereo images and satellite and road maps. In *IEEE/CVF ICCV*, pages 2034–2041, 2011.

[45] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. *NeurIPS*, 32, 2019.

[46] Tao Song, Shan He, and Xinkai Wu. Semantic assisted loop closure detection for automated driving. In *CICTP 2022*, pages 690–698. 2022.

[47] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. In *IEEE ICRA*, 2018.

[48] Hannes Stoll, Peter Zimmer, Frank Hartmann, and Eric Sax. Gps-independent localization for off-road vehicles using ultra-wideband (uwb). In *IEEE ITSC*, 2017.

[49] Yicong Tian, Chen Chen, and Mubarak Shah. Cross-view image matching for geo-localization in urban environments. In *IEEE CVPR*, 2017.

[50] Yuxin Tian, Xueqing Deng, Yi Zhu, and Shawn Newsam. Cross-time and orientation-invariant overhead image geolocalization using deep local features. In *IEEE/CVF WACV*, pages 2512–2520, 2020.

[51] Lucas De Paula Veronese, Edilson de Aguiar, Rafael Correia Nascimento, Jose Guivant, Fernando A Auat Cheein, Alberto Ferreira De Souza, and Thiago Oliveira-Santos. Re-emission and satellite aerial maps applied to vehicle localization on urban environments. In *IEEE/RSJ IROS*, pages 4285–4290, 2015.

[52] Anirudh Viswanathan, Bernardo R Pires, and Daniel Huber. Vision based robot localization by ground to satellite matching in gps-denied situations. In *IEEE/RSJ IROS*, pages 192–198, 2014.

[53] Anirudh Viswanathan, Bernardo R Pires, and Daniel Huber. Vision-based robot localization across seasons and in remote locations. In *IEEE ICRA*, pages 4815–4821. IEEE, 2016.

[54] Ankit Vora, Siddharth Agarwal, Gaurav Pandey, and James McBride. Aerial imagery based lidar localization for autonomous vehicles. *arXiv preprint arXiv:2003.11192*, 2020.

[55] Olga Vysotska and Cyrill Stachniss. Improving SLAM by exploiting building information from publicly available maps and localization priors. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 85:53–65, 2017.

[56] Huayou Wang, Changliang Xue, Yu Tang, Wanlong Li, Feng Wen, and Hongbo Zhang. LTSR: Long-term semantic relocalization based on HD map for autonomous vehicles. In *IEEE ICRA*, pages 2171–2178, 2022.

[57] Tingyu Wang, Zhedong Zheng, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, and Yi Yang. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE T-CSVT*, 32(2):867–879, 2021.

[58] Xipeng Wang, Steve Vozar, and Edwin Olson. Flag: Feature-based localization between air and ground. In *IEEE ICRA*, pages 3178–3184, 2017.

[59] Zhihao Wang, Silin Li, Ming Cao, Haoyao Chen, and Yunhui Liu. Pole-like objects mapping and long-term robot localization in dynamic urban scenarios. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 998–1003. IEEE, 2021.

[60] Ryan W Wolcott and Ryan M Eustice. Visual localization within lidar maps for automated urban driving. In *IEEE/RSJ IROS*, 2014.

[61] Fan Yan, Olga Vysotska, and Cyrill Stachniss. Global localization on Open-StreetMap using 4-bit semantic descriptors. In *ECCV*, 2019.

[62] Yi Yang, Di Tang, Dongsheng Wang, Wenjie Song, Junbo Wang, and Mengyin Fu. Multi-camera visual SLAM for off-road navigation. *Robotics and Autonomous Systems*, 128:103505, 2020.

[63] Zhichao Ye, Chong Bao, Xinyang Liu, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Crossview mapping with graph-based geolocalization on city-scale street maps. In *IEEE ICRA*, pages 7980–7987, 2022.

[64] Yachen Zhu, Yanyang Ma, Long Chen, Cong Liu, Maosheng Ye, and Lingxi Li. Gosmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data. In *IEEE/RSJ IROS*, pages 5151–5157, 2020.