2021

Semantic Simultaneous Localization And Mapping

Sean L. Bowman
University of Pennsylvania

Semantic Simultaneous Localization And Mapping

Abstract

Traditional approaches to simultaneous localization and mapping (SLAM) rely on low-level geometric features such as points, lines, and planes. They are unable to assign semantic labels to landmarks observed in the environment. Recent advances in object recognition and semantic scene understanding, however, have made this information easier to extract than ever before, and the recent proliferation of robots in human environments demand access to reliable semantic-level mapping and localization algorithms to enable true autonomy. Furthermore, loop closure recognition based on low-level features is often viewpoint dependent and subject to failure in ambiguous or repetitive environments, whereas object recognition methods can infer landmark classes and scales, resulting in a small set of easily recognizable landmarks.

In this thesis, we present two solutions that incorporate semantic information into a full localization and mapping pipeline. In the first, we propose a solution method using only single-image bounding box object detections as the semantic measurement. As these bounding box measurements are relatively imprecise when projected back into 3D space and difficult to associate with existing mapped objects, we first present a general method to probabilistically compute data associations within an estimation framework and demonstrate its improved accuracy in the case of high-uncertainty measurements. We then extend this to the specific case of semantic bounding box measurements and demonstrate its accuracy in indoor and outdoor environments.

Second, we propose a solution based on the detection of semantic keypoints. These semantic keypoints are not only more reliably positioned in space, but also allow us to estimate the full six degree-of-freedom pose of each mapped object. The usage of these semantic keypoints allows us to effectively reduce the problem of semantic mapping to that of the much more well studied problem of mapping point features, allowing for its efficient solution and robustness in practice.

Finally, we present a method of robotic navigation in unexplored semantic environments that robustly plans paths through unknown and unexplored semantic environments towards a goal location. Through the use of the semantic keypoint-based semantic SLAM algorithm, we demonstrate the successful execution of navigation missions through on-the-fly generated semantic maps.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Computer and Information Science

First Advisor

George J. Pappas

Keywords

Localization, Mapping, SLAM

Subject Categories

Computer Sciences | Robotics

SEMANTIC SIMULTANEOUS LOCALIZATION AND MAPPING

Sean L. Bowman

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2022

Supervisor of Dissertation

George J. Pappas, Professor, Electrical and Systems Engineering

Graduate Group Chairperson

Mayur Naik, Professor, Computer and Information Science

Dissertation Committee
Kostas Daniilidis, Professor, Computer and Information Science
Daniel E. Koditschek, Professor, Electrical and Systems Engineering
Camillo J. Taylor, Professor, Computer and Information Science
Nikolay Atanasov, Assistant Professor, Electrical and Computer Engineering,
University of California San Diego

SEMANTIC SIMULTANEOUS LOCALIZATION AND MAPPING

For my parents, Pat and Larry

ACKNOWLEDGEMENT

I would like to first and primarily thank my advisor, George Pappas. From the very beginning he has been nothing but supportive, even in the moments where things were difficult, and has been an absolutely invaluable source of inspiration and knowledge. None of this would have been possible without his continued guidance.

I would additionally like to thank Kostas Daniilidis for his constant supply of good ideas, his ability to quickly and clearly pin down the interesting research directions within a larger problem, and his advice and wise words. I also owe a great deal of gratitude to Nikolay Atanasov, who acted early on as a mentor, source of ideas and inspiration, sounding board for my own thoughts, and motivator. His impact is felt throughout the entirety of this thesis. I also want to thank the other members of my committee, Dan Koditschek and C.J. Taylor for their time, patience, and feedback.

Looking farther back, I would also like to thank my undergraduate advisor Stergios Roumeliotis. His early guidance got me interested in the field of robotics, the problem of SLAM, and his insistence on looking at any problem in a rigorous and analytical way has made me a better researcher and engineer to this day.

Finally, I owe a great deal of thanks to other friends and colleagues that through hang outs or other discussion have kept me sane, especially in the times of COVID-19 and lockdowns.

This thesis is dedicated as a whole to my parents, Pat and Larry; although they're unable to be here to see the completion of this journey, I know they would be proud.

ABSTRACT

SEMANTIC SIMULTANEOUS LOCALIZATION AND MAPPING

Sean L. Bowman

George J. Pappas

Traditional approaches to simultaneous localization and mapping (SLAM) rely on low-level geometric features such as points, lines, and planes. They are unable to assign semantic labels to landmarks observed in the environment. Recent advances in object recognition and semantic scene understanding, however, have made this information easier to extract than ever before, and the recent proliferation of robots in human environments demand access to reliable semantic-level mapping and localization algorithms to enable true autonomy. Furthermore, loop closure recognition based on low-level features is often viewpoint dependent and subject to failure in ambiguous or repetitive environments, whereas object recognition methods can infer landmark classes and scales, resulting in a small set of easily recognizable landmarks.

In this thesis, we present two solutions that incorporate semantic information into a full localization and mapping pipeline. In the first, we propose a solution method using only single-image bounding box object detections as the semantic measurement. As these bounding box measurements are relatively imprecise when projected back into 3D space and difficult to associate with existing mapped objects, we first present a general method to probabilistically compute data associations within an estimation framework and demonstrate its improved accuracy in the case of high-uncertainty measurements. We then extend this to the specific case of semantic bounding box measurements and demonstrate its accuracy in indoor and outdoor environments.

Second, we propose a solution based on the detection of semantic keypoints. These semantic keypoints are not only more reliably positioned in space, but also allow us to estimate the full six degree-of-freedom pose of each mapped object. The usage of these semantic keypoints allows us to effectively reduce the problem of semantic mapping to that of the much more well studied problem of mapping point features, allowing for its efficient solution and robustness in practice.

Finally, we present a method of robotic navigation in unexplored semantic environments that robustly plans paths through unknown and unexplored semantic environments towards a goal location. Through the use of the semantic keypoint-based semantic SLAM algorithm, we demonstrate the successful execution of navigation missions through on-the-fly generated semantic maps.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1

Introduction

1.1 Motivation

In robotics, simultaneous localization and mapping (SLAM) is the problem of mapping an unknown environment while estimating a robot's pose within it. Reliable navigation, object manipulation, autonomous surveillance, and many other tasks require accurate knowledge of the robot's pose and of the surrounding environment. Beginning with the seminal works of Smith and Cheeseman (1986) and Leonard and Durrant-Whyte (1991), the SLAM problem has seen decades of remarkable progress resulting in numerous successful and commercially available SLAM systems. Until somewhat recently, however, most of these methods have focused solely on creating a map of low-level geometric features in the environment such as corners (Hesch et al., 2014), lines (Kottas and Roumeliotis, 2013), and surface patches (Henry et al., 2012).

In contrast, high-level autonomy in unknown environments requires more meaningful maps of objects with *semantic* content, such as windows, tables, and chairs. The goal of this thesis is to address the metric and semantic SLAM problems jointly, taking advantage of object recognition to tightly integrate both metric and semantic information into the sensor state and map estimation. In addition to providing a meaningful interpretation of the scene, semantically-labeled landmarks address two critical issues of geometric SLAM: data association (matching sensor observations to map landmarks) and loop closure (recognizing previously-visited locations). In the process of answering this question, we also present a probabilistic method of data association, allowing for the incorporation of measurements with a very high degree of uncertainty, as some semantic observations are, into a successful SLAM system.

1.2 Related Work

Initial approaches to SLAM were typically based on filtering methods in which only the most recent robot pose is estimated (Durrant-Whyte and Bailey, 2006). This approach is in general very computationally efficient, however because of the inability to estimate past poses and relinearize previous measurement functions, errors can compound (Hesch et al., 2014). More recently, batch methods that optimize over larger fractions of the robot trajectory or even entire trajectories have significantly increased in popularity. Successful batch methods typically represent optimization variables as a set of nodes in a graph (a "pose graph" or a "factor graph"). Two robot-pose nodes share an edge if an odometry measurement is available between them, while a landmark and a robot-pose node share an edge if the landmark was observed from the corresponding robot pose. This pose graph optimization formulation of SLAM traces back to Lu and Milios (1997). In recent years, the state of the art (Kümmerle et al., 2011; Kaess et al., 2012) consists of iterative optimization methods (e.g., nonlinear least squares via the Gauss-Newton algorithm) that achieve excellent performance but depend heavily on linearization of the sensing and motion models. This becomes a problem when we consider including discrete observations, such as detected object classes, in the sensing model.

One of the first systems that used both spatial and semantic representations was proposed by Galindo et al. (2005). A spatial hierarchy contained camera images, local metric maps, and the environment topology, while a semantic hierarchy represented concepts and relations, which allowed room categories to be inferred based on object detections. Many other approaches (Civera et al., 2011; Pronobis, 2011; Stückler et al., 2013; Vineet et al., 2015; Leibe et al., 2007; Pillai and Leonard, 2015) extract both metric and semantic information but typically the two processes are carried out separately and the results are merged afterwards. Fei and Soatto (2018) augment an existing SLAM system with a probabilistic object detector. Rosinol et al. (2020) present a dense semantic mapping system that creates a mesh of triangulated points which is then filled with semantic information via back-projection of images' semantic segmentation. The lack of integration between the metric and the semantic mapping does not allow the object detection confidence to influence the performance of the metric optimization. Focusing on the localization problem only, Atanasov et al. (2014) incorporated semantic observations in the metric optimization via a set-based Bayes filter.

An additional class of algorithms may perform localization and semantic mapping jointly but focus on a dense volumetric representation of the environment. Many works (McCormac et al., 2017; Zhang et al., 2018; Rünz and Agapito, 2017; Barsan et al., 2018) use either RGB-D or stereo cameras to directly create a dense reconstruction of the environment. Zheng et al. (2019) adds semantic labeling to a dense RGB-D reconstruction (e.g. KinectFusion), and derives a method of next best view selection to minimize the uncertainty of both the geometric reconstruction and semantic labeling. Grinvald et al. (2019) combine a geometric-semantic segmentation algorithm with an RGB-D camera with a method of object instance data association across multiple frames to create a semantically-informed volumetric reconstruction of the environment.

The works that are closest to ours consider both localization and mapping and carry out metric and semantic mapping jointly. SLAM++ (Salas-Moreno et al., 2013) focuses on a real-time implementation of joint 3-D object recognition and RGB-D SLAM via pose graph optimization. A global optimization for 3D reconstruction and semantic parsing has been proposed by Kundu et al. (2014). In this work, the 3D space is voxelized and landmarks and/or semantic labels are assigned to voxels which are connected in a conditional random field rather than estimating the continuous pose of objects. Bao and Savarese (2011) incorporate camera parameters, object geometry, and object classes into a structure from motion problem, resulting in a detailed and accurate but large and expensive optimization.

The works that are closest to ours jointly optimize a semantic map and the robot trajectory using only a monocular camera and inertial measurement unit. Nicholson et al. (2018) models objects as ellipsoids in 3D space and derives a measurement model describing how bounding box detections constrain these ellipsoids. Yang and Scherer (2019) similarly simplifies object geometry, presenting a monocular SLAM system that represents semantic objects as cuboids in space. Shan et al. (2020) describes an algorithm that estimates a map of objects represented as both a bounding ellipsoid and semantic keypoint-based model, but operates in a filtering context, marginalizing out objects that have left the field of view.

1.3 The Semantic SLAM Problem

To begin, we will consider a general semantic SLAM problem without a specific sensor configuration or measurement model. In the classical simultaneous localization and mapping problem, a mobile sensor moves through an unknown environment, modeled as a collection $\mathcal{L} \triangleq \{\ell_m\}_{m=1}^{M}$ of M static landmarks. Given a set of sensor measurements $\mathcal{Z} \triangleq \{\mathbf{z}_k\}_{k=1}^{K}$, the task is to estimate the landmark positions $\mathcal{L}$ and a sequence of poses $\mathcal{X} \triangleq \{\mathbf{x}_t\}_{t=1}^{T}$ representing the sensor trajectory. A mathematical statement of the SLAM problem is then the following MAP estimation problem:

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \operatorname*{arg\,max}_{\mathcal{X},\mathcal{L}} \log p(\mathcal{X}, \mathcal{L} | \mathcal{Z}), \tag{1.1}$$

or, with a uniform or uninformative prior on $p(\mathcal{X}, \mathcal{L})$ as is typically assumed, the following ML estimation problem:

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \underset{\mathcal{X}, \mathcal{L}}{\arg\max} \log p(\mathcal{Z} | \mathcal{X}, \mathcal{L}). \tag{1.2}$$

By making certain independence assumptions on the measurements, we are able to decompose this optimization into a form that is known as a *factor graph* optimization. A factor graph is a convenient way of representing an optimization problem for which there exists a clear physical structure or a sparse constraint set. Graphically, a factor is a generalization of an edge that allows connectivity between more than two vertices. A factor f in the graph is associated with a cost function that depends on a subset of the variables $\mathcal{V}$ such that the entire optimization is of the form

$$\hat{\mathcal{V}} = \underset{\mathcal{V}}{\arg\min} \sum_{f \in \mathcal{F}} f(\mathcal{V}). \tag{1.3}$$

For example, consider a simple case of a mobile ground robot equipped with wheel encoders. Along its trajectory, between each pair of poses $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, the integrated wheel encoders report a pose difference $\mathbf{z}_i = \mathbf{x}_{i+1} - \mathbf{x}_i + w_i$, where $w_i \sim \mathcal{N}(0, \mathbf{R}_i)$ is some Gaussian noise. It is then easy to see that the solution for the estimation in Equation (1.2) (or Equation (1.1) with a uniform prior on $p(\mathcal{X}, \mathcal{L})$) is given by

$$\hat{\mathbf{x}}_{1:T} = \underset{\mathbf{x}}{\arg\max} \log p(\mathbf{z}_{1:T-1} | \mathbf{x}_{1:T}). \tag{1.4}$$

Assuming conditional independence of measurements given the trajectory and using the known distribution of $\mathbf{z}$, this can be written as

$$\hat{\mathbf{x}}_{1:T} = \arg\min_{\mathbf{x}} \sum_{i=1}^{T-1} \|\mathbf{z}_i - (\mathbf{x}_{i+1} - \mathbf{x}_i)\|^2_{\mathbf{R}_i} \tag{1.5}$$

which we see is a factor formulation as in (1.3) with

$$f(\mathbf{x}_i, \mathbf{x}_{i+1}) = \|\mathbf{z}_i - (\mathbf{x}_{i+1} - \mathbf{x}_i)\|^2_{\mathbf{R}_i}. \tag{1.6}$$

More generally, suppose a robot receives several different classes of measurements $\mathcal{Z}_1, \ldots, \mathcal{Z}_N$, e.g. odometry, GPS, visual, etc. Assuming measurements are conditionally independent given the trajectory and map, and a uniform prior on $\mathcal{Z}$[1], we can write (1.1) as

$$\begin{align}
\hat{\mathcal{X}}, \hat{\mathcal{L}} &= \arg\max_{\mathcal{X},\mathcal{L}} \log p(\mathcal{Z}|\mathcal{X}, \mathcal{L}) p(\mathcal{X}, \mathcal{L}) \tag{1.7}\\
&= \arg\max_{\mathcal{X},\mathcal{L}} \left[\sum_{i=1}^{N} \log p(\mathcal{Z}_i|\mathcal{X}, \mathcal{L}) + \log p(\mathcal{X}, \mathcal{L}) \right] \tag{1.8}\\
&= \arg\min_{\mathcal{X},\mathcal{L}} \left[-\sum_{i=1}^{N} \log p(\mathcal{Z}_i|\mathcal{X}, \mathcal{L}) - \log p(\mathcal{X}, \mathcal{L}) \right], \tag{1.9}
\end{align}$$

and so we see that negative measurement log-likelihoods correspond exactly to the factors in (1.3). Additionally, we see the inherent modularity in the factor graph formulation; new information or measurement types results in only another additive term to the optimization. For example, suppose an existing SLAM system exists in the form of Equation (1.9) and we wish to additionally include a set of semantic

[1] As mentioned before Equation (1.2), most methods additionally assume a uniform prior on $p(\mathcal{X}, \mathcal{L})$ and perform a maximum likelihood estimation; however later in Chapter 4 we will use this term to capture semantic object structure.

measurements $\mathcal{S}$; the new formulation simply becomes

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \arg\min_{\mathcal{X},\mathcal{L}} \left[-\sum_{i=1}^{N} \log p(\mathcal{Z}_i|\mathcal{X}, \mathcal{L}) - \log p(\mathcal{X}, \mathcal{L}) - \log p(\mathcal{S}|\mathcal{X}, \mathcal{L}) \right]. \qquad (1.10)$$

This modularity not only improves the ease with which problems can be formulated and implemented in terms of existing software packages, but also produces a predictable sparsity that can sometimes significantly improve the performance of their solution in practice (Triggs et al., 2000).

Although the formulation as presented here is widespread and useful in practice, hidden in each term $p(\mathcal{Z}|\mathcal{X}, \mathcal{L})$ is the fact that the data association for each measurement $\mathbf{z} \in \mathcal{Z}$ is *a priori* unknown; before the term can be computed, the question of which landmark ℓ_j generated each specific measurement $\mathbf{z}$ must be answered. In the following chapter we will see the effect of considering this question more explicitly and a method of answering it probabilistically.

Chapter 2

Probabilistic Data Association

2.1 Probabilistic Data Association in SLAM

Consider again the classical localization and mapping problem, in which a mobile sensor moves through an unknown environment, modeled as a collection $\mathcal{L} \triangleq \{\ell_m\}_{m=1}^{M}$ of M static landmarks. Given a set of sensor measurements $\mathcal{Z} \triangleq \{\mathbf{z}_k\}_{k=1}^{K}$, the task is to estimate the landmark positions $\mathcal{L}$ and a sequence of poses $\mathcal{X} \triangleq \{\mathbf{x}_t\}_{t=1}^{T}$ representing the sensor trajectory. Most existing work focuses on estimating $\mathcal{X}$ and $\mathcal{L}$ and rarely emphasizes that the data association $\mathcal{D} \triangleq \{(\alpha_k, \beta_k)\}_{k=1}^{K}$ stipulating that measurement z_k of landmark ℓ_{β_k} was obtained from sensor state x_{α_k} is in fact unknown. A complete statement of the SLAM problem (cf. Equation (1.2)) involves maximum likelihood estimation of $\mathcal{X}$, $\mathcal{L}$, and $\mathcal{D}$ given the measurements $\mathcal{Z}$:

$$\hat{\mathcal{X}}, \hat{\mathcal{L}}, \hat{\mathcal{D}} = \underset{\mathcal{X},\mathcal{L},\mathcal{D}}{\arg\max} \log p(\mathcal{Z}|\mathcal{X}, \mathcal{L}, \mathcal{D}) \tag{2.1}$$

The most common approach to this maximization has been to decompose it into two separate estimation problems. First, given prior estimates $\mathcal{X}^0$ and $\mathcal{L}^0$, the maximum likelihood estimate $\hat{\mathcal{D}}$ of the data association $\mathcal{D}$ is computed (e.g., via joint compatibility branch and bound (Neira and Tardós, 2001) or the Hungarian algorithm (Munkres, 1957)). Then, given $\hat{\mathcal{D}}$, the most likely landmark and sensor states are estimated[1]:

$$\hat{\mathcal{D}} = \arg\max_{\mathcal{D}} p(\mathcal{D}|\mathcal{X}^0, \mathcal{L}^0, \mathcal{Z}) \tag{2.2a}$$

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \arg\max_{\mathcal{X},\mathcal{L}} \log p(\mathcal{Z}|\mathcal{X}, \mathcal{L}, \hat{\mathcal{D}}) \tag{2.2b}$$

The second optimization above is typically carried out via filtering (Mourikis and Roumeliotis, 2007; Bloesch et al., 2015; Forster et al., 2014) or pose-graph optimization (Kaess et al., 2012; Kümmerle et al., 2011).

The above process has the disadvantage that an incorrectly chosen data association may have a highly detrimental effect on the estimation performance. Moreover, if ambiguous measurements are discarded to avoid incorrect association choices, they will never be reconsidered later when refined estimates of the sensor pose (and hence their data association) are available. Instead of a simple one step process, then, it is possible to perform **coordinate descent**, which iterates the two maximization steps as follows:

$$\mathcal{D}^{i+1} = \arg\max_{\mathcal{D}} p(\mathcal{D}|\mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}) \tag{2.3a}$$

$$\mathcal{X}^{i+1}, \mathcal{L}^{i+1} = \arg\max_{\mathcal{X},\mathcal{L}} \log p(\mathcal{Z}|\mathcal{X}, \mathcal{L}, \mathcal{D}^{i+1}) \tag{2.3b}$$

This resolves the problem of being able to revisit association decisions once state estimates improve but does little to resolve the problem with ambiguous measurements since a hard decision on data associations is still required. To address this, rather than simply selecting $\hat{\mathcal{D}}$ as the mode of $p(\mathcal{D}|\mathcal{X}, \mathcal{L}, \mathcal{Z})$, we should consider the entire density of $\mathcal{D}$ when estimating $\mathcal{X}$ and $\mathcal{L}$. Given initial estimates $\mathcal{X}^i$, $\mathcal{L}^i$, an improved

[1] Note that the first maximization in (2.2a) assumes that $p(\mathcal{D}|\mathcal{X}^0, \mathcal{L}^0)$ is uniform. This is true when there are no false positive measurements or missed detections. A more sophisticated model can be obtained using ideas from Atanasov et al. (2016).

estimate that utilizes the whole density of $\mathcal{D}$ can be computed by maximizing the expected measurement likelihood via **expectation maximization** (EM):

$$\begin{aligned} \mathcal{X}^{i+1}, \mathcal{L}^{i+1} &= \arg\max_{\mathcal{X},\mathcal{L}} \mathbb{E}_{\mathcal{D}}\big[\log p(\mathcal{Z}|\mathcal{X},\mathcal{L},\mathcal{D}) \mid \mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}\big] \\ &= \arg\max_{\mathcal{X},\mathcal{L}} \sum_{\mathcal{D}\in\mathbb{D}} p(\mathcal{D}|\mathcal{X}^i,\mathcal{L}^i,\mathcal{Z}) \log p(\mathcal{Z}|\mathcal{X},\mathcal{L},\mathcal{D}) \end{aligned} \tag{2.4}$$

where $\mathbb{D}$ is the space of all possible values of $\mathcal{D}$. This EM formulation has the advantage that no hard decisions on data association are required since it "averages" over all possible associations. To compare this with the coordinate descent formulation in (2.3), we can rewrite (2.4) as follows:

$$\begin{aligned} &\arg\max_{\mathcal{X},\mathcal{L}} \sum_{\mathcal{D}\in\mathbb{D}} \sum_{k=1}^{K} p(\mathcal{D}|\mathcal{X}^i,\mathcal{L}^i,\mathcal{Z}) \log p(\mathbf{z}_k|\mathbf{x}_{\alpha_k},\ell_{\beta_k}) \\ &\qquad = \arg\max_{\mathcal{X},\mathcal{L}} \sum_{k=1}^{K}\sum_{j=1}^{M} \left[\sum_{\mathcal{D}\in\mathbb{D}(k,j)} p(\mathcal{D}|\mathcal{X}^i,\mathcal{L}^i,\mathcal{Z}) \right] \log p(\mathbf{z}_k|\mathbf{x}_{\alpha_k},\ell_j) \\ &\qquad = \arg\max_{\mathcal{X},\mathcal{L}} \sum_{k=1}^{K}\sum_{j=1}^{M} w_{kj}^i \log p(\mathbf{z}_k|\mathbf{x}_{\alpha_k},\ell_j) \end{aligned} \tag{2.5}$$

where

$$w_{kj}^i \triangleq \sum_{\mathcal{D}\in\mathbb{D}(k,j)} p(\mathcal{D}|\mathcal{X}^i,\mathcal{L}^i,\mathcal{Z}) \tag{2.6}$$

is a weight, independent of the optimization variables $\mathcal{X}$ and $\mathcal{L}$, that quantifies the influence of the "soft" data association, and $\mathbb{D}(k,j) \triangleq \{\mathcal{D} \in \mathbb{D} \mid \beta_k = j\} \subseteq \mathbb{D}$ is the set of all data associations such that measurement k is assigned to landmark j. Note that the coordinate descent optimization (2.3b) after expanding the measurement likelihoods has a similar form to (3.18), except that for each k there is exactly one j

such that $w_{kj}^i = 1$ and $w_{kl}^i = 0$ for all $l \neq j$.

We can also show that the EM formulation, besides being a generalization of coordinate descent, is equivalent to the following matrix permanent maximization problem.

Proposition 1. *If $p(\mathcal{D} \mid \mathcal{X}^i, \mathcal{L}^i)$ is uniform, the maximizers of the EM formulation in* (2.4) *and the optimization below are equal:*

$$\mathcal{X}^{i+1}, \mathcal{L}^{i+1} = \arg\max_{\mathcal{X},\mathcal{L}} \mathbf{per}(\mathbf{Q}^i(\mathcal{X}, \mathcal{L})),$$

where **per** *denotes the matrix permanent*[2], *$\mathbf{Q}^i(\mathcal{X}, \mathcal{L})$ is a matrix with elements $[\mathbf{Q}^i]_{kj} := p(\mathbf{z}_k | \mathbf{x}_j^i, \ell_j^i) p(\mathbf{z}_k | \mathbf{x}_j, \ell_j)$ and $\{(\mathbf{x}_j^i, \ell_j^i)\}$ and $\{(\mathbf{x}_j, \ell_j)\}$ are enumerations of the sets $\mathcal{X}^i \times \mathcal{L}^i$ and $\mathcal{X} \times \mathcal{L}$, respectively.*

Proof. First, we rewrite the optimization in (2.4) without a logarithm and similarly expand the expectation:

$$\begin{aligned} \mathcal{X}^{i+1}, \mathcal{L}^{i+1} &= \arg\max_{\mathcal{X},\mathcal{L}} \mathbb{E}_{\mathcal{D}}\big[p(\mathcal{Z} | \mathcal{X}, \mathcal{L}, \mathcal{D}) \mid \mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}\big] \qquad (2.7) \\ &= \arg\max_{\mathcal{X},\mathcal{L}} \sum_{\mathcal{D} \in \mathbb{D}} p(\mathcal{D} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}) p(\mathcal{Z} | \mathcal{X}, \mathcal{L}, \mathcal{D}) \end{aligned}$$

The data association likelihood can then be rewritten as

$$\begin{aligned} p(\mathcal{D} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}) &= \frac{p(\mathcal{Z} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{D}) p(\mathcal{D} | \mathcal{X}^i, \mathcal{L}^i)}{p(\mathcal{Z} | \mathcal{X}^i, \mathcal{L}^i)} \qquad (2.8) \\ &= \frac{p(\mathcal{Z} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{D}) p(\mathcal{D} | \mathcal{X}^i, \mathcal{L}^i)}{\sum_{\mathcal{D}} p(\mathcal{Z} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{D}) p(\mathcal{D} | \mathcal{X}^i, \mathcal{L}^i)} \qquad (2.9) \\ &= \frac{p(\mathcal{Z} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{D})}{\sum_{\mathcal{D}} p(\mathcal{Z} | \mathcal{X}^i, \mathcal{L}^i, \mathcal{D})} \qquad (2.10) \end{aligned}$$

[2] The permanent of an $n \times m$ matrix $A = [A(i,j)]$ with $n \leq m$ is defined as $\mathbf{per}(A) := \sum_\pi \prod_{i=1}^n A(i, \pi(i))$, where the sum is over all one-to-one functions $\pi : \{1, \ldots, n\} \to \{1, \ldots, m\}$.

with the last equality due to the assumption that $p(\mathcal{D}|\mathcal{X},\mathcal{L})$ is uniform. We can next decompose the measurement likelihood

$$p(\mathcal{Z}|\mathcal{X},\mathcal{L},\mathcal{D}) = \prod_k p(\mathbf{z}_k|\mathbf{x}_{\alpha_k}, \ell_{\beta_k}), \tag{2.11}$$

and so

$$\begin{aligned} \mathcal{X}^{i+1}, \mathcal{L}^{i+1} &= \arg\max_{\mathcal{X},\mathcal{L}} \sum_{\mathcal{D}\in\mathbb{D}} p(\mathcal{D}|\mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}) p(\mathcal{Z}|\mathcal{X},\mathcal{L},\mathcal{D}) \\ &= \arg\max_{\mathcal{X},\mathcal{L}} \sum_{\mathcal{D}\in\mathbb{D}} \prod_k \frac{p(\mathbf{z}_k|\mathbf{x}^i_{\alpha_k}, \ell^i_{\beta_k}) p(\mathbf{z}_k|\mathbf{x}_{\alpha_k}, \ell_{\beta_k})}{\sum_{\mathcal{D}} p(\mathcal{Z}|\mathcal{X}^i, \mathcal{L}^i, \mathcal{D})} \end{aligned} \tag{2.12}$$

The result then follows by noting that the normalizing denominator is independent of the optimization variables and from the definition of the matrix permanent. □

Similar to the coordinate descent formulation, the EM formulation (3.18) allows us to solve the permanent maximization problem iteratively. First, instead of estimating a maximum likelihood data association, we estimate the data association distribution $p(\mathcal{D}|\mathcal{X}^i, \mathcal{L}^i, \mathcal{Z})$ in the form of the weights w^i_{kj} (the "E" step). Then, we maximize the expected measurement log likelihood over the previously computed distribution (the "M" step).

2.2 Simulations

To observe the effect of incorporating a probabilistic model of data association and using Proposition 1 on the performance of an actual SLAM algorithm, we implemented a simple 2D SLAM simulator of a bicycle robot equipped with a range and bearing sensor around a field of fixed landmarks. The model is the same used in Bailey et al. (2006), and the MATLAB code is partially based on the accompanying code (Bailey, 2021).

The simulated SLAM state consists of the 2D vehicle state

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} \tag{2.13}$$

and the locations of each of the mapped landmarks $\ell_i \in \mathbb{R}^2$ observed in the environment. The vehicle motion model is taken to be the trajectory of the front wheel of a bicycle subject to rolling motion constraints,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_{k-1} + V_k \Delta T \cos(\phi_{k-1} + \gamma_k) \\ y_{k-1} + V_k \Delta T \sin(\phi_{k-1} + \gamma_k) \\ \phi_{k-1} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix}, \tag{2.14}$$

where $\mathbf{u}_k = [V_k \ \gamma_k]^T$ is the controls vector containing the velocity and steering angle, respectively, and B is the wheelbase between the front and rear axles of the bicycle.

The observation model for a range-bearing measurement generated at time k by landmark $\ell_j = [x_{\ell_j} \ y_{\ell_j}]^T$ is given by

$$\mathbf{h}(\mathbf{x}_i, \ell_j) = \begin{bmatrix} \sqrt{(x_i - x_{\ell_j})^2 + (y_i - y_{\ell_j})^2} \\ \arctan \frac{y_i - y_{\ell_j}}{x_i - x_{\ell_j}} - \phi_k \end{bmatrix}. \tag{2.15}$$

To estimate the robot and map state at time k, we use a general factor graph formulation as given in Equation (3.18):

$$\hat{X}, \hat{L} = \underset{\mathcal{X}, \mathcal{L}}{\arg\max} \sum_{k=1}^{K} \sum_{j=1}^{M} w_{kj}^i \log p(\mathbf{z}_k | \mathbf{x}_{\alpha_k}, \ell_j) + \sum_{t=1}^{T} \log p(\mathbf{z}_{odom,t} | \mathbf{x}_t, \mathbf{x}_{t-1}), \tag{2.16}$$

where we assume the weights on odometric constraints have $w = 1$ due to there being no data association ambiguity.

To measure the effect of probabilistic data association on the estimation performance, we solve for the weights w^i_{kj} in two ways. First, we use the probabilistic formulation as outlined in Section 2.1:

$$w^i_{kj} \triangleq \sum_{\mathcal{D} \in \mathbb{D}(k,j)} p(\mathcal{D}|\mathcal{X}^i, \mathcal{L}^i, \mathcal{Z}). \tag{2.17}$$

We refer to this as the probabilistic formulation.

Second, we use a maximum likelihood formulation of the data association problem, in which we set w^i_{kj} equal to 1 for exactly one j, the mode of the data association distribution, and 0 everywhere else:

$$w^i_{kj} \triangleq \begin{cases} 1 & \text{if } j = \arg\max_s p(\mathbf{z}_k|\mathbf{x}_i, \ell_s) \\ 0 & \text{else }, \end{cases} \tag{2.18}$$

this is equal to the standard maximum likelihood SLAM formulation and we refer to it as the ML formulation here. The resulting least-squares system in both cases is solved with a simple custom Levenberg-Marquardt solver that simply solves the full system at each time step.

An example trajectory solved with the ML formulation under high noise conditions immediately after a first loop closure is shown in Figure 1. The true trajectory is shown as a black line and its estimate is shown in purple. The latest robot position estimate is shown as a triangle along with its associated covariance ellipse in blue. True feature locations are shown as green stars and their estimated locations are shown as red dots; the associated covariance ellipses are plotted around each estimated feature as a red ellipse. These "high" noise conditions are set so the standard deviation of the

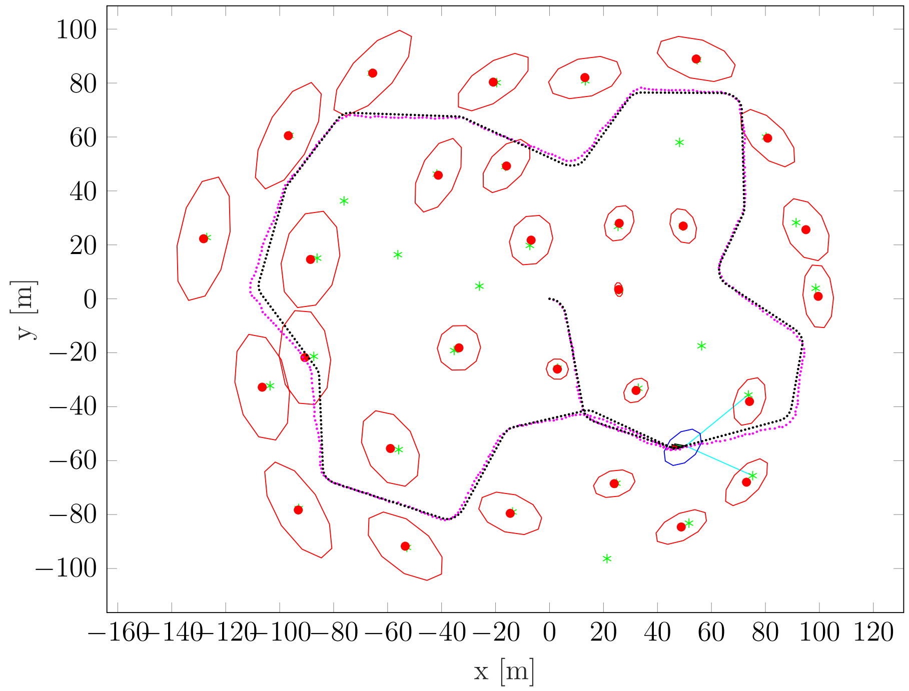

Figure 1: Simulated trajectory after successful loop closure

range measurement noise is $\sigma_{range} = 0.25$ meters, and the standard deviation of the bearing measurement noise is $\sigma_{bear} = 5$ degrees.

An alternate example trajectory at the same point and under the same noise conditions in an ML formulation-solved trajectory after the first loop closure was misdetected is shown in Figure 2. In this case, the detection of one of the features near the beginning of the trajectory was falsely associated with a neighboring feature rather than the true feature that generated it. Due to the ML formulation of the data association solution, this misassigned constraint was given the full weight, resulting in not only the trajectory estimate jumping away from the true position, but also the estimator becoming inconsistent; note how the covariance ellipses of both the robot and some mapped features do not contain the true position.

One would expect under a probalistic formulation for the errors created in such a

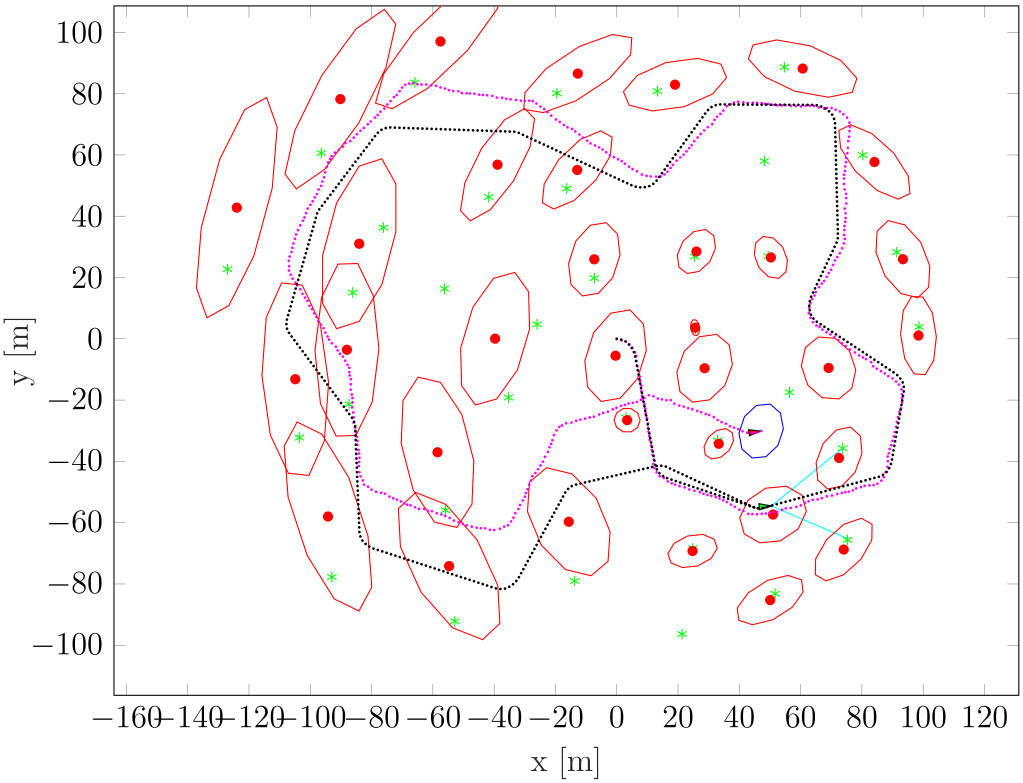

Figure 2: Simulated trajectory after wrong loop closure

situation to be smaller; this incorrect loop closure will have a smaller measurement likelihood, leading to the constraint weight w_{kj} to being smaller, which injects less information into the system. The effect of this is both that the covariance ellipses will be shrunk less by this incorrect loop closure and that the estimate will deviate from its prior position by a smaller amount.

To test this over numerous such trajectories, we ran 1000 Monte Carlo simulations of the same trajectory for both the ML and probabilistic association methods. Each trajectory consisted of two loops around the shown environment, with the successful estimations able to make two loop closures. The final position errors over all such runs are shown as box plots in Figure 3; the boxes show the median, 25th, and 75th percentiles of the final errors with whiskers extending to the set of outlier trials. From this plot it is seen that the both the median error and the bulk of the error distribution of the trials using probabilistic association are lower than those using ML association; this is due to the ML association creating false loop closures and highly confident but wrong associations under the given high measurement noise conditions.

A distribution of all used measurement weights in the probabilistic association case is shown in Figure 4. Most measurements were included in the final estimation with a weight near 1, with a sharply decreasing tail as w decreases past 0.8. The complete lack of any weights less than 0.1 is due to a heuristic threshold in which we discarded any extremely unconfident and ambiguous measurements with $w < 0.1$.

A second set of Monte Carlo experiments was also run in a lower-noise condition, with $\sigma_{bear} = 1$ degree. The box plots of the final position errors are shown in Figure 5, and the histogram of probabilistic measurement constraint weights is shown in Figure 6.

As can be seen in Figure 5, the median position error at the end of the trajectory is very close to 0, reflecting the fact that most associations over the trajectory were performed correctly for both the ML and probabilistic methods. As most associations

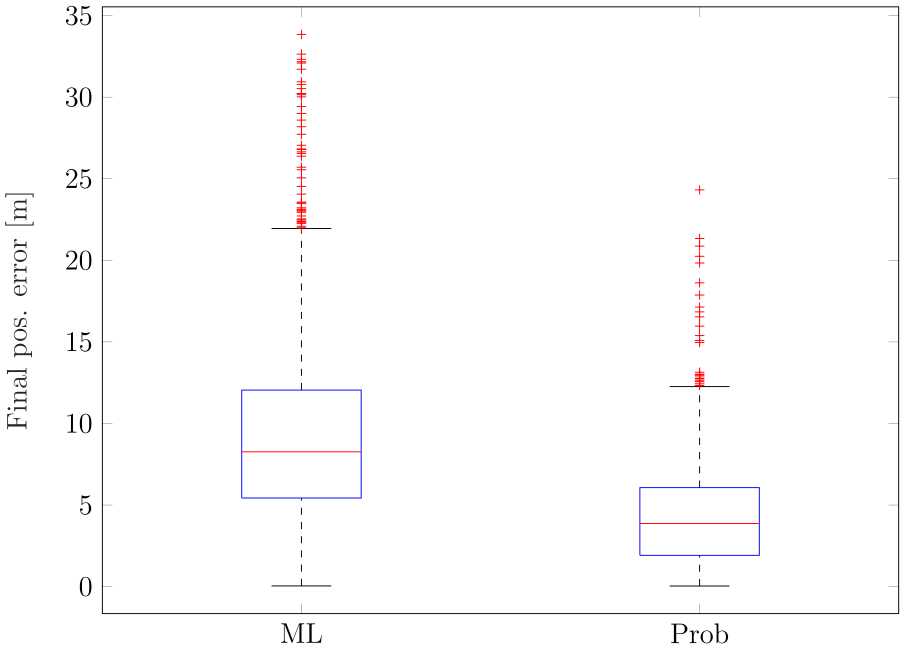

Figure 3: Distributions of final trajectory position errors for ML and Probabilistic association methods under high noise conditions, with $\sigma_{bear} = 5$ degrees.

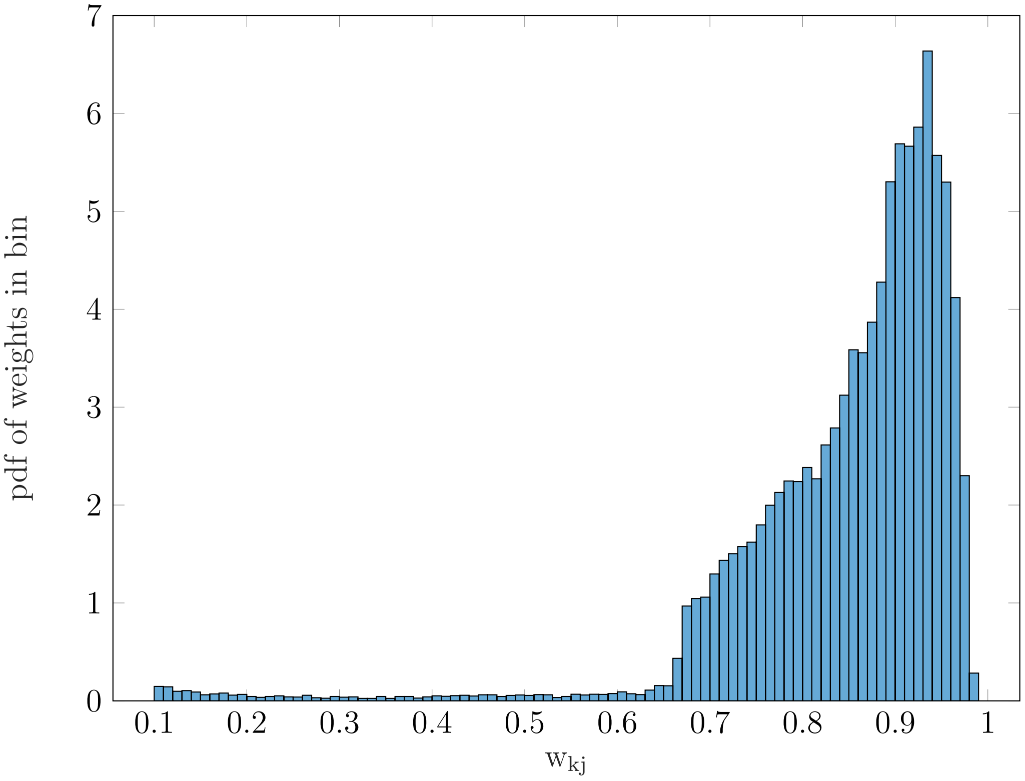

Figure 4: Distribution of all included measurement weights for probabilistic associations under high noise conditions, with $\sigma_{bear} = 5$ degrees.

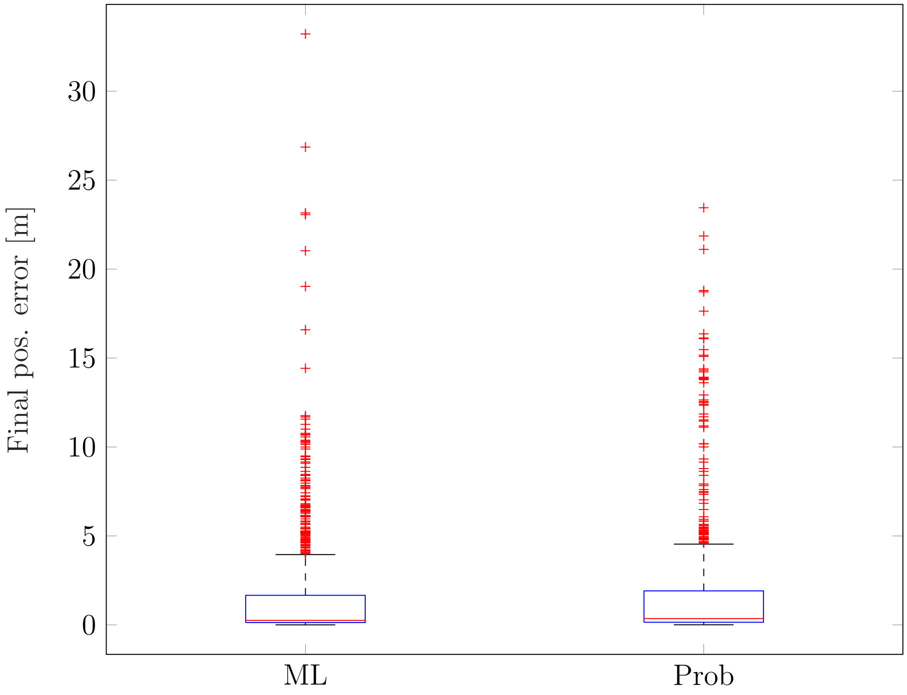

Figure 5: Distributions of final trajectory position errors for ML and Probabilistic association methods under low noise conditions, with $\sigma_{bear} = 1$ degree.

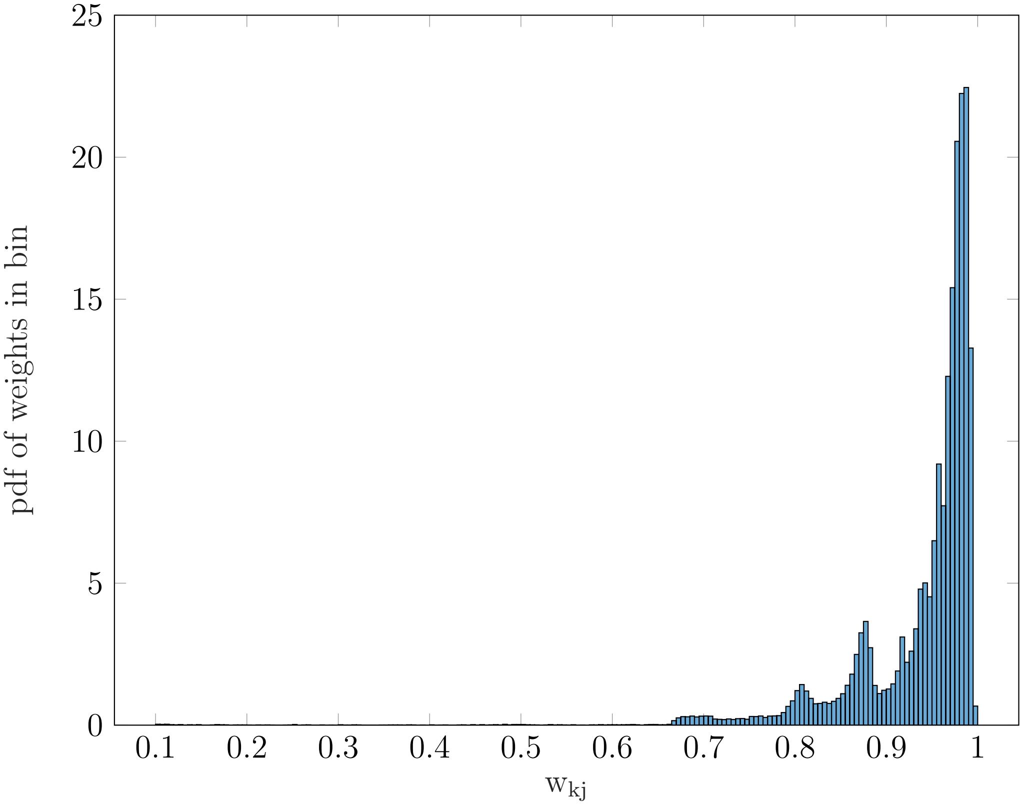

Figure 6: Distribution of all included measurement weights for probabilistic associations under low noise conditions, with $\sigma_{bear} = 1$ degree.

were correct, the ML association method actually has a slightly lower median error (0.25 meters) compared to the probabilistic association method (0.35 meters). This is expected, as if all associations are correct, the ML method will weight them more highly than the probabilistic method. The lower uncertainty in each measurement assignment can also be seen in the histogram in Figure 6, with the majority of weights near 1 and none below 0.6.

Chapter 3

Semantic SLAM with Bounding Box Detections

3.1 Problem Formulation

In this chapter, we focus on a particular formulation of the SLAM problem (Bowman et al., 2017; Atanasov et al., 2018) that in addition to sensor and landmark poses involves *landmark classes* (*e.g.*, door, chair, table) and *semantic measurements* in the form of object detections. We will demonstrate that the expectation maximization formulation (3.18) is an effective way to solve the semantic SLAM problem.

Let the state ℓ of each landmark consist of its position $\ell^p \in \mathbb{R}^3$ as well as a class label ℓ^c from a discrete set $\mathcal{C} = \{1, \ldots, C\}$. To estimate the landmark states $\mathcal{L}$ and sensor trajectory $\mathcal{X}$, we utilize three sources of information: inertial, geometric point features, and semantic object observations. The purpose for the inclusion of these three source of information is that they are all complementary to each other. IMUs and inertial measurements are ideal for tracking the state of the robot over very short periods of time, in cases of temporary occlusion of the camera or feature-less environments to improve robustness, and over periods of very dynamic motion where image processing becomes difficult. Geometric features extracted from images, on the other hand, are able to track motion and build relative motion constraints over medium time frames. Finally, in addition to the already discussed inherently useful

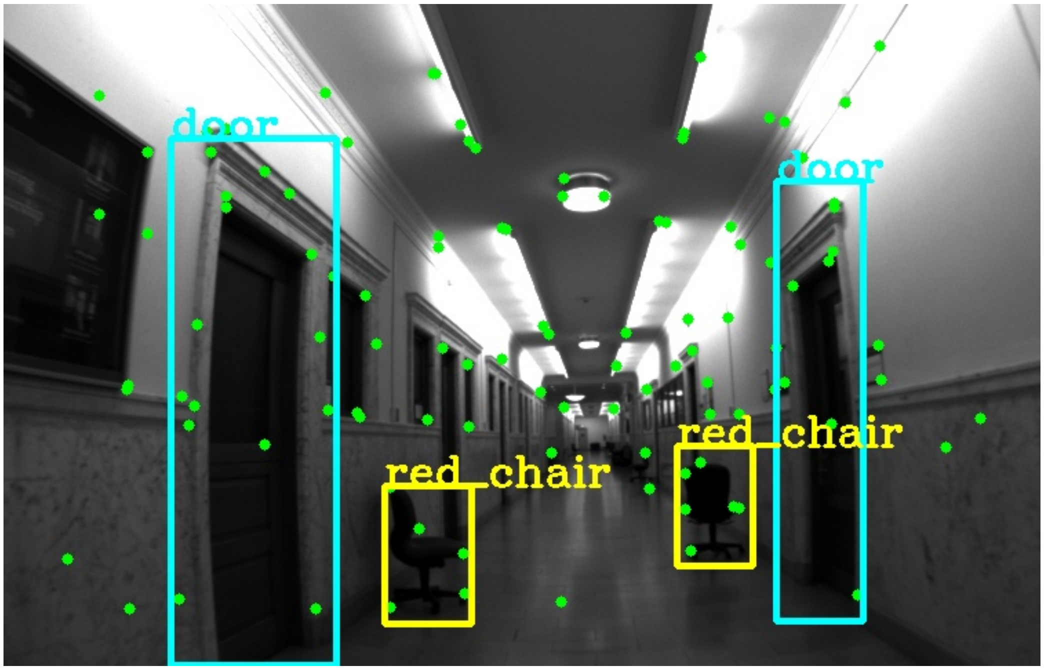

Figure 7: Example keyframe image overlaid with ORB features (green points) and object detections

qualities of building a semantic map, semantic information is useful for the most long-term motion tracking and viewpoint-independent loop closure.

Examples of the geometric features (ORB features) and semantic bounding box observations extracted from a single image can be seen in Figure 7, and will be discussed in detail in the upcoming sections.

3.1.1 Inertial information

We assume that the sensor package consists of an inertial measurement unit (IMU) and one monocular camera. A subset of the images captured by the camera are chosen as *keyframes* (*e.g.*, by selecting every nth frame as a keyframe). The sensor state corresponding to the tth keyframe is denoted x_t and consists of the sensor 6-D pose, velocity, and IMU bias values. We assume that the IMU and camera are time synchronized, so between keyframes t and $t+1$, the sensor also collects a set $\mathcal{I}_t$ of

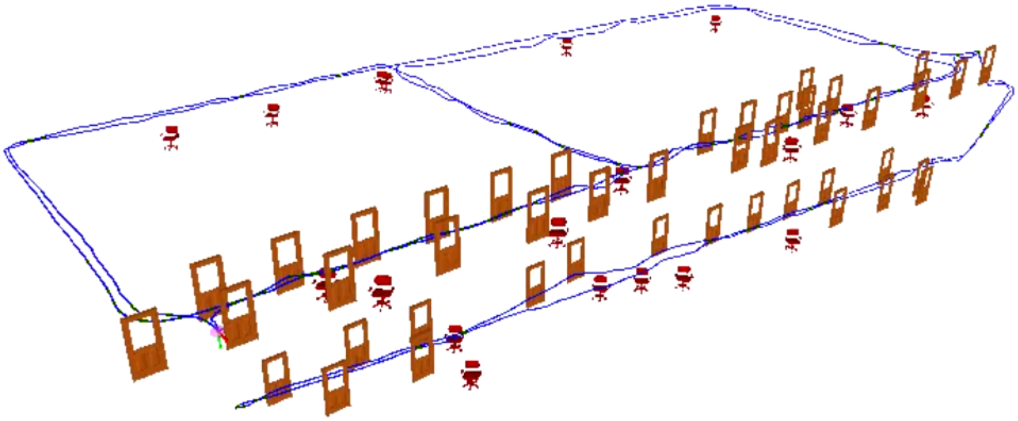

Figure 8: Estimated sensor trajectory (blue) and landmark positions and classes using inertial, geometric, and semantic measurements such as those in Fig. 7.

IMU measurements (linear acceleration and rotational velocity).

3.1.2 Geometric information

In addition to the inertial measurements $\mathcal{I}_t$, we utilize geometric point measurements (*e.g.*, Harris corners, SIFT, SURF, FAST, BRISK, ORB, etc.) $\mathcal{Y}_t$. From each keyframe image, these geometric point features are extracted and tracked forward to the subsequent keyframe. In our experiments we extract ORB features (Rublee et al., 2011) from each keyframe and match them to the subsequent keyframe by minimizing the ORB descriptor distance. Since these features are matched by an external method, we assume that their data association is known.

3.1.3 Semantic information

The last type of measurement used are object detections $\mathcal{S}_t$ extracted from every keyframe image. An object detection $\mathbf{s}_k = (s_k^c, s_k^s, s_k^b) \in \mathcal{S}_t$ extracted from keyframe t consists of a detected class $s_k^c \in \mathcal{C}$, a score s_k^s quantifying the detection confidence, and a bounding box s_k^b. Such information can be obtained from any modern approach for object recognition such as Ren et al. (2015); Bochkovskiy et al. (2020); Srinivas et al.

(2021); Carion et al. (2020). In our implementation, we use a deformable parts model (DPM) detector from Felzenszwalb et al. (2010); Zhu et al. (2014); Dubout and Fleuret (2013), which runs on a CPU in real time. If the data association $\mathcal{D}_k = (\alpha_k, \beta_k)$ of measurement $\mathbf{s}_k$ is known, the measurement likelihood can be decomposed as follows:

$$p(\mathbf{s}_k|\mathbf{x}_{\alpha_k}, \ell_{\beta_k}) = p(s_k^c|\ell_{\beta_k}^c)p(s_k^s|\ell_{\beta_k}^c, s_k^c)p(s_k^b|\mathbf{x}_{\alpha_k}, \ell_{\beta_k}^p). \tag{3.1}$$

The density $p(s_k^c|\ell_{\beta_k}^c)$ corresponds to the confusion matrix of the object detector and is learned offline along with the score distribution $p(s_k^s|\ell_{\beta_k}^c, s_k^c)$. The bounding-box likelihood $p(s_k^b|\mathbf{x}_{\alpha_k}, \ell_{\beta_k}^p)$ is assumed normally distributed with mean equal to the perspective projection of the centroid of the object onto the image plane and covariance proportional to the dimensions of the detected bounding box.

Problem 1 (Semantic SLAM). *Given inertial $\mathcal{I} \triangleq \{\mathcal{I}_t\}_{t=1}^T$, geometric $\mathcal{Y} \triangleq \{\mathcal{Y}_t\}_{t=1}^T$, and semantic $\mathcal{S} \triangleq \{\mathcal{S}_t\}_{t=1}^T$ measurements, estimate the sensor state trajectory $\mathcal{X}$ and the positions and classes $\mathcal{L}$ of the objects in the environment.*

The inertial and geometric measurements are used to track the sensor trajectory locally and, similar to a visual odometry approach, the geometric structure is not recovered. The semantic measurements, in contrast, are used to construct a map of objects that can be used to perform loop closure that is robust to ambiguities and viewpoint and is more efficient than a SLAM approach that maintains full geometric structure.

3.2 Semantic SLAM using EM

Following the observations from Chapter 2, we apply expectation maximization to robustly handle the semantic data association. In addition to treating data association as a latent variable, we also treat the discrete landmark class labels as latent variables in the optimization, resulting in a clean and efficient separation between discrete and continuous variables. As mentioned in Section 3.1.2, the data association of the geometric measurements is provided by the feature tracking algorithm, so the latent variables we use are the data association $\mathcal{D}$ of the semantic measurements measurements and the object classes $\ell^c_{1:M}$. The following proposition specifies the EM steps necessary to solve the semantic SLAM problem. The initial guess $\mathcal{X}^{(0)}$ is provided by odometry integration; the initial guess $\mathcal{L}^{(0)}$ can be obtained from $\mathcal{X}^{(0)}$ by initializing a landmark along the detected camera ray.

Proposition 2. *If $p(\mathcal{D}|\mathcal{X}, \mathcal{L})$ is uniform and the semantic measurement data associations are independent across keyframes*[1]*, i.e.,*

$$p(\mathcal{D}|\mathcal{S}, \mathcal{X}, \mathcal{L}) = \prod_{t=1}^{T} p(\mathcal{D}_t|\mathcal{S}_t, \mathcal{X}, \mathcal{L}), \tag{3.2}$$

the semantic SLAM problem can be solved via the expectation maximization algorithm by iteratively solving for (1) data association weights w^t_{ij} (the "E" step) and (2) continuous sensor states $\mathcal{X}$ and landmark positions $\ell^p_{1:M}$ (the "M" step) via the following equations:

$$w^{t,(i)}_{kj} = \sum_{\ell^c \in \mathcal{C}} \sum_{\mathcal{D}_t \in \mathbb{D}_t(k,j)} \kappa^{(i)}(\mathcal{D}_t, \ell^c) \quad \forall t, k, j \tag{3.3}$$

$$\begin{aligned} \mathcal{X}^{(i+1)}, \ell^{p,(i+1)}_{1:M} = \underset{\mathcal{X}, \ell^p_{1:M}}{\arg\min} \sum_{t=1}^{T} \sum_{\mathbf{s}_k \in \mathcal{S}_t} \sum_{j=1}^{M} -w^{t,(i)}_{kj} \log p(\mathbf{s}_k|\mathbf{x}_t, \ell_j) \\ - \log p(\mathcal{Y}|\mathcal{X}) - \log p(\mathcal{I}|\mathcal{X}) \end{aligned} \tag{3.4}$$

[1] This "naïve Bayes" assumption might not always hold perfectly in practice but it significantly simplifies the optimization and allows for efficient implementation.

where

$$\kappa^{(i)}(\mathcal{D}_t, \ell^c) = \frac{p(\mathcal{S}_t | \mathcal{X}^{(i)}, \mathcal{L}^{(i)}, \mathcal{D}_t)}{\sum_{\ell^c} \sum_{\mathcal{D}_t \in \mathbb{D}_t} p(\mathcal{S}_t | \mathcal{X}^{(i)}, \mathcal{L}^{(i)}, \mathcal{D}_t)},$$

$\mathbb{D}_t$ *is the set of all possible data associations for measurements received at timestep* t*, and* $\mathbb{D}_t(i, j) \subseteq \mathbb{D}_t$ *is the set of all possible data associations for measurements received at time* t *such that measurement* i *is assigned to landmark* j*.*

Proof. Suppose we have some initial guess given by $\theta^{(i)} = \{\mathcal{X}^{(i)}, \ell^{p,(i)}\}$. We can then compute an improved estimate of $\theta = \{\mathcal{X}, \ell^p\}$ by maximizing the expected log likelihood:

$$\theta^{(i+1)} = \arg\max_{\theta} \mathbb{E}_{\mathcal{D}, \ell^c | \theta^{(i)}} \left[\log p(\mathcal{D}, \ell^c, \mathcal{S}, \mathcal{Y}, \mathcal{I} | \theta)\right] \tag{3.5}$$

Expanding the expectation,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}, \ell^c | \theta^{(i)}} \left[\log p(\mathcal{D}, \ell^c, \mathcal{S}, \mathcal{Y}, \mathcal{I} | \theta)\right] \\
&\qquad = \sum_{\mathcal{D}, \ell^c} p(\mathcal{D}, \ell^c | \mathcal{S}, \mathcal{Y}, \mathcal{I}, \theta^{(i)}) \log p(\mathcal{S}, \mathcal{Y}, \mathcal{I}, \mathcal{D}, \ell^c | \theta) && (3.6) \\
&\qquad = \sum_{\mathcal{D}, \ell^c} p(\mathcal{D}, \ell^c | \mathcal{S}, \theta^{(i)}) \log[p(\mathcal{S}, \mathcal{D}, \ell^c | \theta) p(\mathcal{Y} | \theta) p(\mathcal{I} | \theta)] && (3.7)
\end{aligned}$$

Letting $\kappa(\mathcal{D}, \ell^c) \triangleq p(\mathcal{D}, \ell^c | \mathcal{S}, \theta^{(i)})$, a constant with respect to the optimization variables, we continue:

$$\begin{aligned}
\mathbb{E}[\cdot] &= \sum_{\mathcal{D}, \ell^c} \kappa(\mathcal{D}, \ell^c) \log p(\mathcal{S}, \mathcal{D}, \ell^c | \theta) + \sum_{\mathcal{D}, \ell^c} \kappa(\mathcal{D}, \ell^c) \log[p(\mathcal{Y} | \theta) p(\mathcal{I} | \theta)] && (3.8) \\
&= \sum_{\mathcal{D}, \ell^c} \kappa(\mathcal{D}, \ell^c) \log p(\mathcal{S}, \mathcal{D}, \ell^c | \theta) + \log[p(\mathcal{Y} | \theta) p(\mathcal{I} | \theta)] \sum_{\mathcal{D}, \ell^c} \kappa(\mathcal{D}, \ell^c) && (3.9) \\
&= \sum_{\mathcal{D}, \ell^c} \kappa(\mathcal{D}, \ell^c) \log p(\mathcal{S}, \mathcal{D}, \ell^c | \theta) + \log p(\mathcal{Y} | \theta) + \log p(\mathcal{I} | \theta), && (3.10)
\end{aligned}$$

as $\sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) = 1$.

Focusing on the leftmost summation over data associations and landmark classes,

$$\begin{aligned}&\sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) \log p(\mathcal{S},\mathcal{D},\ell^c|\theta) \qquad (3.11)\\ &= \sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) \log p(\mathcal{S}|\mathcal{D},\ell^c,\theta) + \sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) \log p(\mathcal{D},\ell^c|\theta)\end{aligned}$$

Using the assumption that $p(\mathcal{D},\ell^c|\theta)$ is a uniform distribution over the space of data associations and landmark classes, this term doesn't affect which θ maximizes the objective, so for optimization purposes we have

$$\begin{aligned}\sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) \log p(\mathcal{S},\mathcal{D},\ell^c|\theta) &= \sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) \log p(\mathcal{S}|\mathcal{D},\ell^c,\theta) \qquad (3.12)\\ &= \sum_t \sum_i \sum_{\mathcal{D}_t,\ell^c} \kappa(\mathcal{D}_t,\ell^c) \log p(\mathbf{s}_i|\mathbf{x}_t,\ell_{\beta_i}) \qquad (3.13)\end{aligned}$$

Note that if we let $\mathbb{D}(i,j)$ be the subset of all possible data associations that assign measurement i to landmark j, we can further decompose this summation as

$$\sum_{\mathcal{D},\ell^c} \kappa(\mathcal{D},\ell^c) \log p(\mathcal{S},\mathcal{D},\ell^c|\theta) = \sum_t \sum_i \sum_j \sum_{\ell^c} \sum_{\mathcal{D}_t \in \mathbb{D}(i,j)} \kappa(\mathcal{D}_t,\ell^c) \log p(\mathbf{s}_i|\mathbf{x}_t,\ell_j) \quad (3.14)$$

Finally, letting $w_{ij}^t \triangleq \sum_{\ell^c} \sum_{\mathcal{D}_t \in \mathbb{D}(i,j)} \kappa(\mathcal{D}_t,\ell^c)$, we can write the final expectation maximization as

$$\theta^{(i+1)} = \arg\max_\theta \sum_t \sum_i \sum_j w_{ij}^t \log p(\mathbf{s}_i|\mathbf{x}_t,\ell_j) + \log p(\mathcal{Y}|\theta) + \log p(\mathcal{I}|\theta). \qquad (3.15)$$

□

While Proposition 2 allows us to iteratively solve the semantic SLAM problem in a probabilistic EM framework, the computation of the weights $w_{kj}^{t,(i)}$ requires summations over exponentially large data association spaces and can become prohibitively expensive when the number of landmarks or measurements grows large. Similar to the expression in Proposition 1, it is possible to express this weight as a matrix permanent.

Proposition 3. *If $p(\mathcal{D}|\mathcal{X}, \mathcal{L})$ is uniform and the semantic measurement data associations are independent across keyframes, i.e.,*

$$p(\mathcal{D}|\mathcal{S}, \mathcal{X}, \mathcal{L}) = \prod_{t=1}^{T} p(\mathcal{D}_t|\mathcal{S}_t, \mathcal{X}, \mathcal{L}), \tag{3.16}$$

the semantic SLAM problem can be solved via the expectation maximization algorithm by iteratively solving for (1) data association weights w_{ij}^t (the "E" step) and (2) continuous sensor states $\mathcal{X}$ and landmark positions $\ell_{1:M}^p$ (the "M" step) via the following equations:

$$w_{kj}^{t,(i)} = \gamma_t^i l_{kj}^t \, \mathbf{per}\, \mathbf{L}_{-kj}^t \tag{3.17}$$

$$\begin{aligned} \mathcal{X}^{(i+1)}, \ell_{1:M}^{p,(i+1)} = \underset{\mathcal{X}, \ell_{1:M}^p}{\arg\min} \sum_{t=1}^{T} \sum_{\mathbf{s}_k \in \mathcal{S}_t} \sum_{j=1}^{M} -w_{kj}^{t,(i)} \log p(\mathbf{s}_k|\mathbf{x}_t, \ell_j) \\ - \log p(\mathcal{Y}|\mathcal{X}) - \log p(\mathcal{I}|\mathcal{X}) \end{aligned} \tag{3.18}$$

*where γ_t^i is a normalizing factor such that $\sum_k w_{kj}^{t,i} = 1$, **per** denotes the matrix permanent, $\mathbf{L}^t$ is the matrix of individual measurement likelihoods with $l_{kj}^t = p(\mathbf{s}_k^t|\mathbf{x}_t, \ell_j)$, and $\mathbf{L}_{-ij}^t$ is the matrix $\mathbf{L}^t$ with the ith row and jth column removed.*

Proof. From Proposition 2, the weights $w_{kj}^{t,(i)}$ are defined as

$$w_{kj}^{t,(i)} = \sum_{\ell^c \in \mathcal{C}} \sum_{\mathcal{D}_t \in \mathbb{D}_t(k,j)} \frac{p(\mathcal{S}_t | \mathcal{X}^{(i)}, \mathcal{L}^{(i)}, \mathcal{D}_t)}{\sum_{\ell^c} \sum_{\mathcal{D}_t \in \mathbb{D}_t} p(\mathcal{S}_t | \mathcal{X}^{(i)}, \mathcal{L}^{(i)}, \mathcal{D}_t)} \tag{3.19}$$

$$= \gamma_t^i \sum_{\ell^c \in \mathcal{C}} \sum_{\mathcal{D}_t \in \mathbb{D}_t(k,j)} p(\mathcal{S}_t | \mathcal{X}^{(i)}, \mathcal{L}^{(i)}, \mathcal{D}_t), \tag{3.20}$$

where

$$\gamma_t^i = \frac{1}{\sum_{\ell^c} \sum_{\mathcal{D}_t \in \mathbb{D}_t} p(\mathcal{S}_t | \mathcal{X}^{(i)}, \mathcal{L}^{(i)}, \mathcal{D}_t)} \tag{3.21}$$

is a constant normalizing factor.

Now, given a data association, individual measurements are independent and so we can expand

$$w_{kj}^{t,(i)} = \gamma_t^i \sum_{\ell^c} \sum_{\mathcal{D}_t \in \mathbb{D}_t(k,j)} \prod_{s_k \in \mathcal{S}_t} p(\mathbf{s}_k | \mathbf{x}_t, \ell_{\beta_k}). \tag{3.22}$$

where β_k is the landmark index as given by the data association $\mathcal{D}_t$ such that the kth measurement was generated by the β_kth landmark.

For all $\mathcal{D}_t \in \mathbb{D}_t(k,j)$ we have $\beta_k = j$ by definition, so

$$w_{kj}^{t,(i)} = \gamma_t^i \; p(\mathbf{s}_k | \mathbf{x}_t, \ell_j) \sum_{\mathcal{D}_t \in \mathbb{D}_t(k,j)} \prod_{s_m \in \mathcal{S}_t; m \neq k} p(\mathbf{s}_m | \mathbf{x}_t, \ell_{\beta_m}) \tag{3.23}$$

From the definition of the matrix permanent,

$$\mathbf{per}\,\mathbf{L} = \sum_{\pi} \prod_{s=1}^{K_t} l_{s,\pi(s)}^t \tag{3.24}$$

where the first sum is over all one-to-one functions $\pi : \{1, \ldots, K_t\} \to \{1, \ldots, M\}$ and where $K_t = |\mathcal{S}_t|$. This is exactly our definition of a valid data association, and with the definition of $\mathcal{L}$ as given in the statement of the proposition,

$$\mathbf{per}\,\mathbf{L} = \sum_{\mathcal{D}\in\mathbb{D}} \prod_{s=1}^{K_t} l^t_{s,\beta_s} \tag{3.25}$$

$$= \sum_{\mathcal{D}\in\mathbb{D}} \prod_{s_k\in\mathcal{S}_t} p(\mathbf{s}_k|\mathbf{x}_t, \ell_{\beta_k}) \tag{3.26}$$

Similarly, the permanent of $\mathbf{L}^t_{-kj}$ will include a sum over all one-to-one functions $\pi : \{1, \ldots, K_t\} \setminus \{k\} \to \{1, \ldots, M\} \setminus \{j\}$. It is now easy to see that

$$\mathbf{per}\,\mathbf{L}^t_{-kj} = \sum_{\mathcal{D}_t\in\mathbb{D}_t(k,j)} \prod_{s_m\in\mathcal{S}_t;m\neq k} p(\mathbf{s}_m|\mathbf{x}_t, \ell_{\beta_m}) \tag{3.27}$$

and so combining Equations (3.23) and (3.27) we have the final expression

$$w^{t,i}_{kj} = \gamma^i_t\; p(\mathbf{s}_k|\mathbf{x}_t, \ell_j)\,\mathbf{per}\,\mathbf{L}^t_{-kj} \tag{3.28}$$

$$= \gamma^i_t\; l^t_{kj}\,\mathbf{per}\,\mathbf{L}^t_{-kj}. \tag{3.29}$$

□

Crucially, the above proposition allows us to take advantage of matrix permanent approximation algorithms (Jerrum et al., 2004; Law, 2009) that have been developed. Proposition 3 thus allows us to effectively summarize the combinatorially large data association space in polynomial time, making probabilistic data association feasible for even a large number of measurements.

3.2.1 Object classes and data association (E step)

The computation of the weights for a single keyframe require several combinatorial sums over all possible data associations. However, due to the assumption of independent associations among keyframes and the fact that only few objects are present within the sensor field-of-view, it is feasible to compute the summations and hence w_{kj}^t for all keyframes t, measurements k, and landmarks j extremely efficiently in practice. Once the weights $w_{kj}^{t,(i)}$ are computed for each measurement-landmark pair, they are used within the continuous optimization over sensor states and landmark positions. Additionally, maximum likelihood landmark class estimates ℓ^c can be recovered from the computed κ values:

$$\hat{\ell}_{1:M}^c = \arg\max_{\ell^c} p(\ell_{1:M}^c|\theta, \mathcal{Z}) = \arg\max_{\ell^c} \prod_{t=1}^{T} \sum_{\mathcal{D}_t \in \mathbb{D}_t} \kappa(\mathcal{D}_t, \ell^c)$$

3.2.2 Pose graph optimization (M step)

Equation (3.4) forms the basis of our pose graph optimization over sensor states and landmark positions. A pose graph is a convenient way of representing an optimization problem for which there exists a clear physical structure or a sparse constraint set. The graph consists of a set of vertices $\mathcal{V}$, each of which corresponds to an optimization variable, and a set of factors $\mathcal{F}$ among the vertices that correspond to individual components of the cost function. Graphically, a factor is a generalization of an edge that allows connectivity between more than two vertices. A factor f in the graph is associated with a cost function that depends on a subset of the variables $\mathcal{V}$ such that the entire optimization is of the form

$$\hat{\mathcal{V}} = \arg\min_{\mathcal{V}} \sum_{f \in \mathcal{F}} f(\mathcal{V}) \tag{3.30}$$

In addition to providing a useful representation, factor graphs are advantageous in that there exist computational tools that allow efficient optimization (Dellaert, 2012; Kümmerle et al., 2011).

Our graph has a vertex for each sensor state $\mathbf{x}_t$ and for each landmark position ℓ^p_i. Contrary to most prior work in which a hard data association decision results in a measurement defining a single factor between a sensor pose and a landmark, we consider soft semantic data association multiple factors.

Semantic Factors

A measurement $\mathbf{s}_k$ from sensor state $\mathbf{x}_i$ defines factors $f^s_{kj}(\mathbf{x}_i, \ell_j)$ for each visible landmark j. Assuming the number of visible landmarks and the number of received measurements are approximately equal, with this method the number of semantic factors in the graph is roughly squared. Note that since ℓ^c is fixed in (3.4), $p(s^s|\ell^c, s^c)$ and $p(s^c|\ell^c)$ are constant. Thus, $\log p(\mathbf{s}|\mathbf{x}, \ell) = \log p(s^b|\mathbf{x}, \ell^p) + \log p(s^s|\ell^c, s^c)p(s^c|\ell^c)$ and so the latter term can be dropped from the optimization.

Let $h_\pi(\mathbf{x}, \ell^p)$ be the standard perspective projection of a landmark ℓ^p onto a camera at pose $\mathbf{x}$. We assume that the camera measurement of a landmark ℓ^p from camera pose $\mathbf{x}$ is Gaussian distributed with mean $h_\pi(\mathbf{x}, \ell^p)$ and covariance $\mathbf{R}_s$. Thus, a camera factor corresponding to sensor state t, measurement k, and landmark j, f^s_{kj}, becomes

$$f^s_{kj}(\mathcal{X}, \mathcal{L}) = -w^{t,(i)}_{kj} \log p(s^b_k|\mathbf{x}_t, \ell^p_j) \tag{3.31}$$

$$= \|s^b_k - h_\pi(\mathbf{x}_t, \ell_j)\|^2_{\mathbf{R}_s/w^{t,(i)}_{kj}} \tag{3.32}$$

Those semantic factors due to the re-observation of a previously seen landmark are our method's source of loop closure constraints.

Geometric Factors

Following Forster et al. (2015) and Mourikis and Roumeliotis (2007), we incorporate geometric measurements into the pose graph as structureless constraints between the camera poses that observed them. We can rewrite the term corresponding to geometric factors in (3.4) as

$$-\log p(\mathcal{Y}|\mathcal{X}) = -\sum_{i=1}^{N_y} \sum_{k:\beta_k^y=i} \log p(\mathbf{y}_k|\mathbf{x}_{\alpha_k^y}) \tag{3.33}$$

where N_y is the total number of distinct feature tracks, i.e. the total number of observed physical geometric landmarks.

Letting $\rho_{\beta_k^y}$ be the 3D position in the global frame of the landmark that generated measurement $\mathbf{y}_k$, and assuming as before that the projection has Gaussian pixel noise with covariance $\mathbf{R}_y$, we have

$$-\log p(\mathcal{Y}|\mathcal{X}) = \sum_{i=1}^{N_y} \sum_{k:\beta_k^y=i} \|\mathbf{y}_k - h_\pi(\mathbf{x}_{\alpha_k^y}, \rho_i)\|^2_{\mathbf{R}_y} \tag{3.34}$$

For a single observed landmark ρ_i, the factor constraining the camera poses which observed it takes the form

$$f_i^y(\mathcal{X}) = \sum_{k:\beta_k^y=i} \|\mathbf{y}_k - h_\pi(\mathbf{x}_{\alpha_k^y}, \rho_i)\|^2_{\mathbf{R}_y} \tag{3.35}$$

Because we use iterative methods to optimize the full pose graph, it is necessary to linearize the above cost term. The linearization of the above results in an inner cost term of the form

$$c_i = \sum_{k:\beta_k^y=i} \|\mathbf{H}_{ik}^{\rho}\delta\rho_i + \mathbf{H}_{ik}^{\mathbf{x}}\delta\mathbf{x}_{\alpha_k^y} + \mathbf{b}_{ik}\|^2 \tag{3.36}$$

where $\mathbf{H}_{ik}^{\rho}$ is the Jacobian of the cost function with respect to $\rho_{\beta_k^y}$, $\mathbf{H}_{ik}^{\mathbf{x}}$ is the Jacobian with respect to $\mathbf{x}_{\alpha_k^y}$, $\mathbf{b}_{ik}$ is a function of the measurement and its error, and the linearized cost term is in terms of deltas $\delta\mathbf{x}$, $\delta\rho$ rather than the true values $\mathbf{x}$, ρ.

Writing the inner summation in one matrix form by stacking the individual components, we can write this simply as

$$c_i = \|\mathbf{H}_i^{\rho}\delta\rho_i + \mathbf{H}_i^{\mathbf{x}}\delta\mathbf{x}_{\alpha^y(i)} + \mathbf{b}_i\|^2. \tag{3.37}$$

To avoid optimizing over ρ values, and hence to remove the dependence of the cost function upon them, we project the cost into the null space of its Jacobian. We premultiply each cost term by $\mathbf{A}_i$, a matrix whose columns span the left nullspace of $\mathbf{H}_i^{\rho}$. The cost term for the structureless geometric features thus becomes a function of only the states which observe it:

$$c_i = \|\mathbf{A}_i\mathbf{H}_i^{\mathbf{x}}\delta\mathbf{x}_{\alpha^y(i)} + \mathbf{A}_i\mathbf{b}_i\|^2 \tag{3.38}$$

Inertial Factors

To incorporate the accelerometer and gyroscope measurements into the pose graph, we use the method of preintegration factors detailed in Forster et al. (2015). The authors provide an efficient method of computing inertial residuals between two keyframes $\mathbf{x}_i$ and $\mathbf{x}_j$ in which several inertial measurements were received. By "preintegrating" all IMU measurements received between the two keyframes, the relative pose difference (*i.e.* difference in position, velocity, and orientation) between the two successive keyframes is estimated. Using this estimated relative pose, the authors provide expressions for inertial residuals on the rotation ($\mathbf{r}_{\Delta R_{ij}}$), velocity ($\mathbf{r}_{\Delta \mathbf{v}_{ij}}$), and position ($\mathbf{r}_{\Delta \mathbf{p}_{ij}}$) differences between two keyframes as a function of the poses $\mathbf{x}_i$ and $\mathbf{x}_j$. Specifically, they provide said expressions along with their noise covariances $\mathbf{\Sigma}$ such that

$$\begin{aligned} f_i^{\mathcal{I}}(\mathcal{X}) &= -\log p(\mathcal{I}_{ij}|\mathcal{X}) && (3.39)\\ &= \|\mathbf{r}_{\Delta R_{ij}}\|^2_{\mathbf{\Sigma}_{Rij}} + \|\mathbf{r}_{\Delta \mathbf{v}_{ij}}\|^2_{\mathbf{\Sigma}_{vij}} + \|\mathbf{r}_{\Delta \mathbf{p}_{ij}}\|^2_{\Sigma_{pij}} && (3.40)\\ &= \|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\mathbf{\Sigma}_{ij}} && (3.41) \end{aligned}$$

The full pose graph optimization corresponding to equation (3.4) is then a nonlinear least squares problem involving semantic observation terms (see (3.32)), geometric observation terms (see (3.38)), and inertial terms (see (3.41)).

$$\hat{\mathbf{x}}_{1:T}, \hat{\ell}_{1:M} = \underset{\mathcal{X}, \ell_{1:M}}{\arg\min} \sum_{k=1}^{K} \sum_{j=1}^{M} f_{kj}^{s}(\mathcal{X}, \ell_{1:M}^{p}) + \sum_{i=1}^{N_y} f_i^{y}(\mathcal{X}) + \sum_{t=1}^{T-1} f_t^{\mathcal{I}}(\mathcal{X}) \tag{3.42}$$

We solve this within the iSAM2 framework (Kaess et al., 2012), which is able to provide a near-optimal solution with real-time performance.

3.3 Experiments

We implemented our algorithm in C++ using GTSAM (Dellaert, 2012) and its iSAM2 implementation as the optimization back-end. All experiments were able to be computed in real-time.

The front-end in our implementation simply selects every 15th camera frame as a keyframe. As mentioned in section 3.1.2, the tracking front-end extracts ORB features (Rublee et al., 2011) from every selected keyframe and tracks them forward through the images by matching the ORB descriptors. Outlier tracks are eliminated by estimating the essential matrix between the two views using RANSAC and removing those features which do not fit the estimated model. We assume that the timeframe between two subsequent images is short enough that the orientation difference between

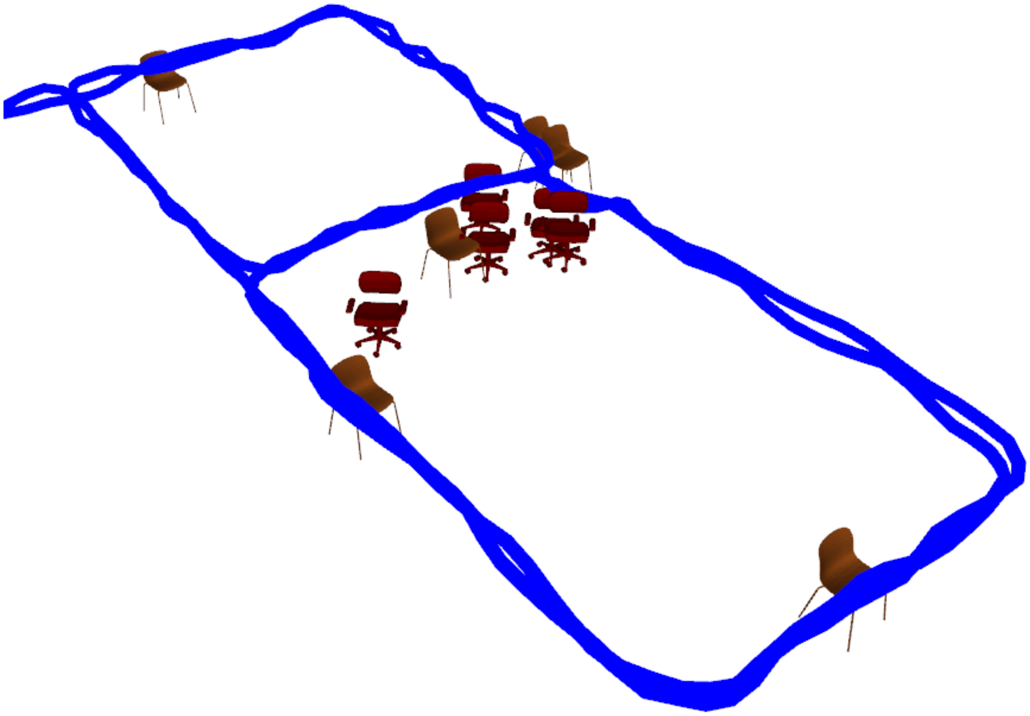

Figure 9: Sensor trajectory and estimated landmarks for the first office experiment

the two frames can be estimated accurately by integrating the gyroscope measurements. Thus, only the unit translation vector between the two images needs to be estimated. We can then estimate the essential matrix using only two point correspondences (Kottas et al., 2013).

The front-end's object detector is an implementation of the deformable parts model detection algorithm (Dubout and Fleuret, 2013). On the acquisition of the semantic measurements from a new keyframe, the Mahalanobis distance from the measurement to all known landmarks is computed. If all such distances are above a certain threshold, a new landmark is initialized in the map, with initial position estimate along the camera ray, with depth given by the median depth of all geometric feature measurements within its detected bounding box (or some fixed value if no

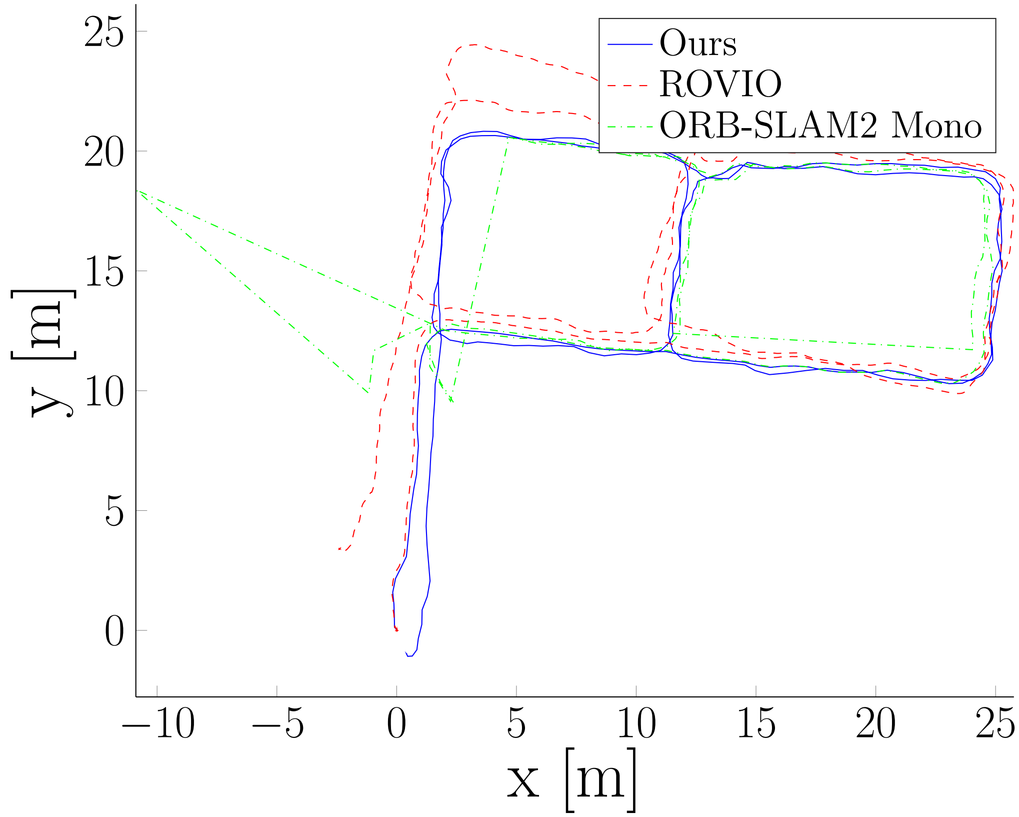

Figure 10: Estimated trajectories in first office experiment.

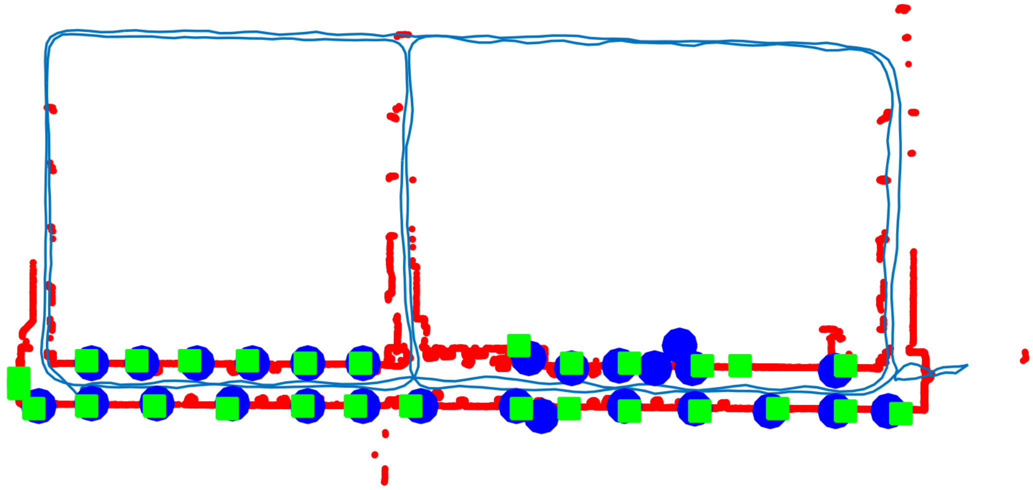

Figure 11: Estimated trajectory in second office experiment from our algorithm (blue line) along with our estimated door landmark positions (blue circles), overlaid onto partial ground truth map (red) along with ground truth door locations (green squares)

such features were tracked successfully).

While ideally we would iterate between solving for constraint weights w_{ij} and poses as Proposition 3 suggests, in practice for computational reasons we solve for the weights just once per keyframe.

Our experimental platform was a VI-Sensor (Nikolic et al., 2014) from which we used the IMU and left camera. We performed three separate experiments. The first consists of a medium length (approx. 175 meters) trajectory around one floor of an office building, in which the object classes detected and kept in the map were two types of chairs (red office chairs and brown four-legged chairs). The second experiment is a long (approx. 625 meters) trajectory around two different floors of an office building. The classes in the second experiment are red office chairs and doors. The third and final trajectory is several loops around a room equipped with a vicon motion tracking system, in which the only class of objects detected is red office chairs. In addition to our own experiments, we applied our algorithm to the KITTI dataset (Geiger et al., 2012) odometry sequences 05 and 06.

The final trajectory estimate along with the estimated semantic map for the first office experiment is shown in Figure 9. The trajectories estimated by our algorithm, by the ROVIO visual-inertial odometry algorithm (Bloesch et al., 2015), and by the ORB-SLAM2 visual SLAM algorithm (Mur-Artal et al., 2015; Mur-Artal and Tardós, 2016), projected into the x-y plane, are shown in Figure 10. Due to a lack of inertial information and a relative lack of visual features in the environment, ORB-SLAM2 frequently got lost and much of the trajectory estimate is missing, but was always able to recover when entering a previously mapped region.

The second office experiment trajectory along with the estimated map is shown in Figure 8. An example image overlaid with object detections from near the beginning of this trajectory is displayed in Figure 7. We constructed a partial map of the top

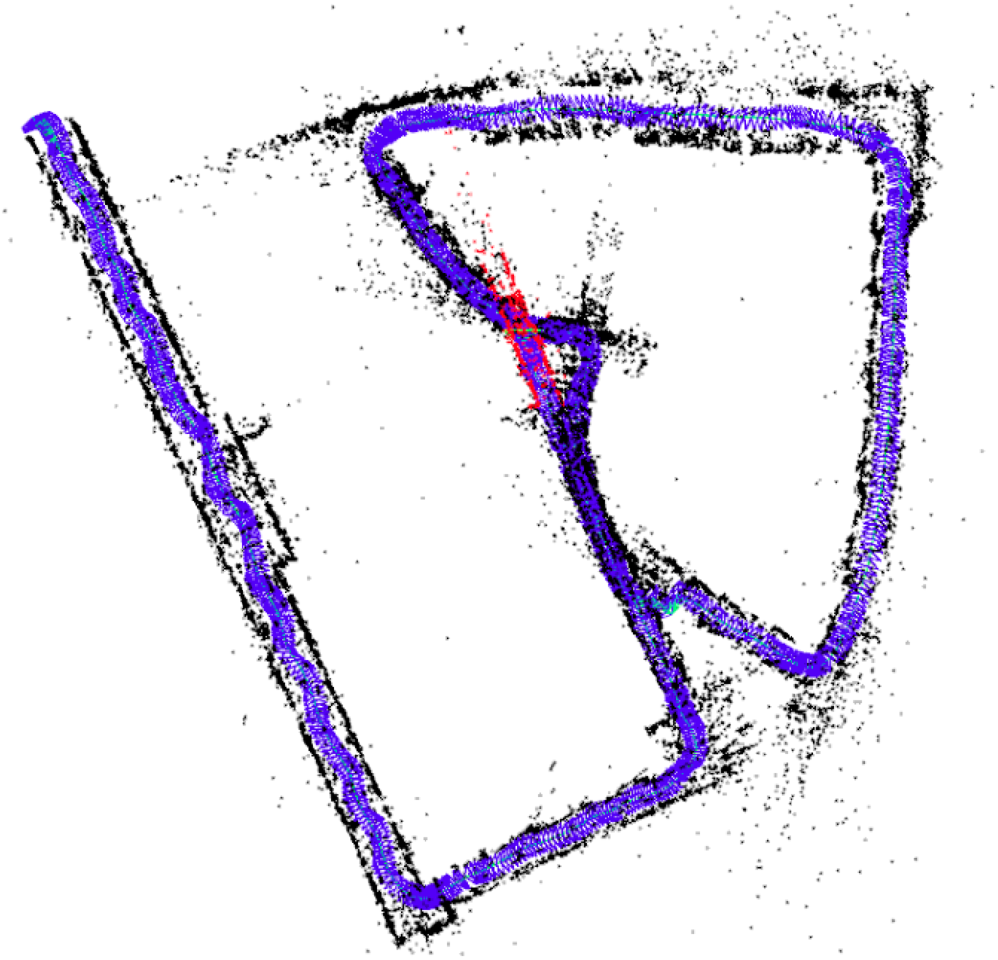

Figure 12: Partial ORB-SLAM2 trajectory after incorrect loop closure in second office experiment.

floor in the experiment using a ground robot equipped with a LIDAR scanner. On this ground truth map, we manually picked out door locations. The portion of the estimated trajectory on the top floor is overlaid onto this partial truth map (the two were manually aligned) in Figure 11, Due to the extremely repetitive nature of the hallways in this experiment, bag-of-words based loop closure detections are subject to false positives and incorrect matches. ORB-SLAM2 was unable to successfully estimate the trajectory due to such false loop closures. A partial trajectory estimate after an incorrect loop closure detection is shown in Fig. 12.

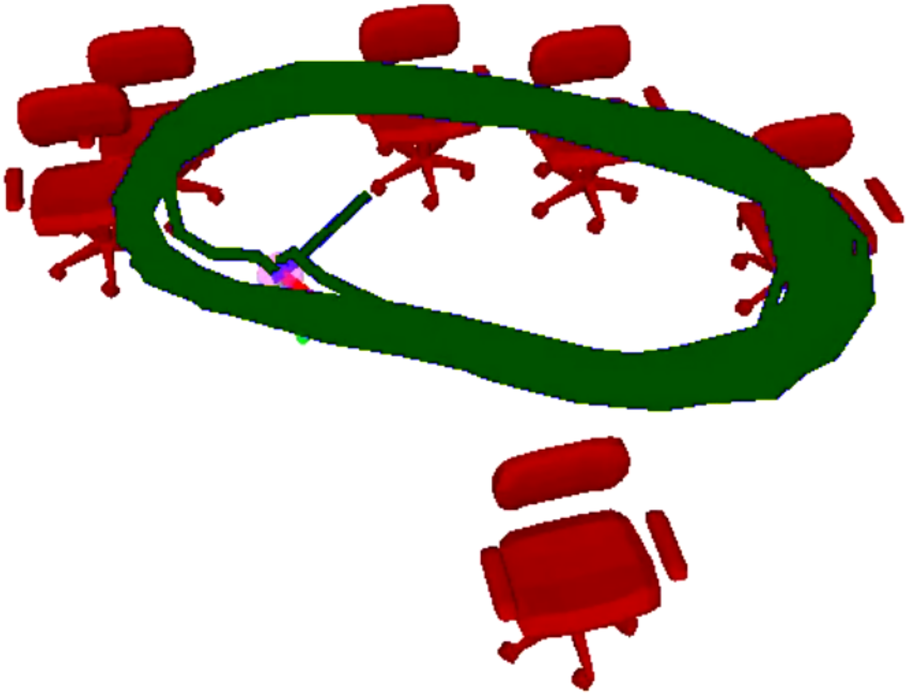

Figure 13: Sensor trajectory and estimated landmarks for the vicon experiment

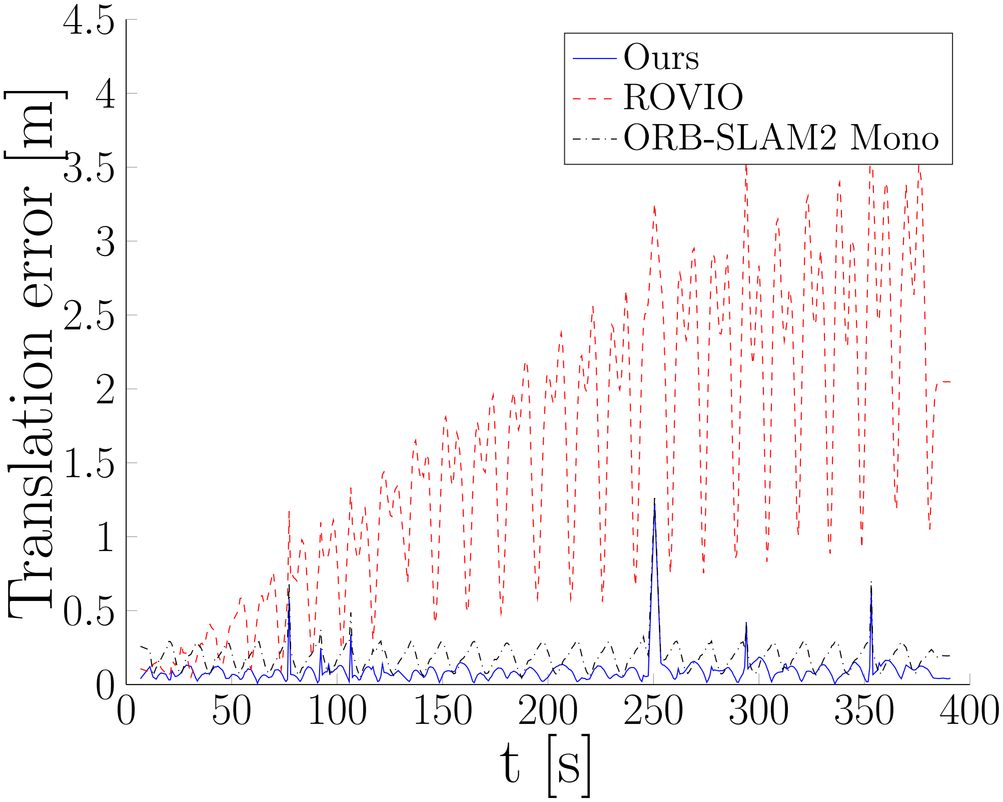

Figure 14: Position errors with respect to vicon ground truth.

The vicon trajectory and the estimated map of chairs is shown in Figure 13. We evaluated the position error with respect to the vicon's estimate for our algorithm, ROVIO, and ORB-SLAM2 and the results are shown in Figure 14. Note that the spikes in the estimate errors are due to momentary occlusion from the vicon cameras.

We also evaluated our algorithm on the KITTI outdoor dataset, using odometry sequences 05 and 06. The semantic objects detected and used in our algorithm were cars. Rather than use inertial odometry in this experiment, we used the VISO2 (Geiger et al., 2011) visual odometry algorithm as the initial guess $\mathcal{X}^{(0)}$ for a new keyframe state. Similarly, we replaced the preintegrated inertial relative pose (cf. Section 3.2.2) with the relative pose obtained from VISO in the odometry factors. The absolute position errors over time for KITTI sequence 05 with respect to ground truth for our algorithm, VISO2, and ORB-SLAM2 with monocular and stereo cameras are shown in Figure 15. The same for sequence 06 are shown in Figure 16. Finally, the mean translational and rotational errors over all possible subpaths of length (100, 200, ..., 800) meters are shown in Figure 18.

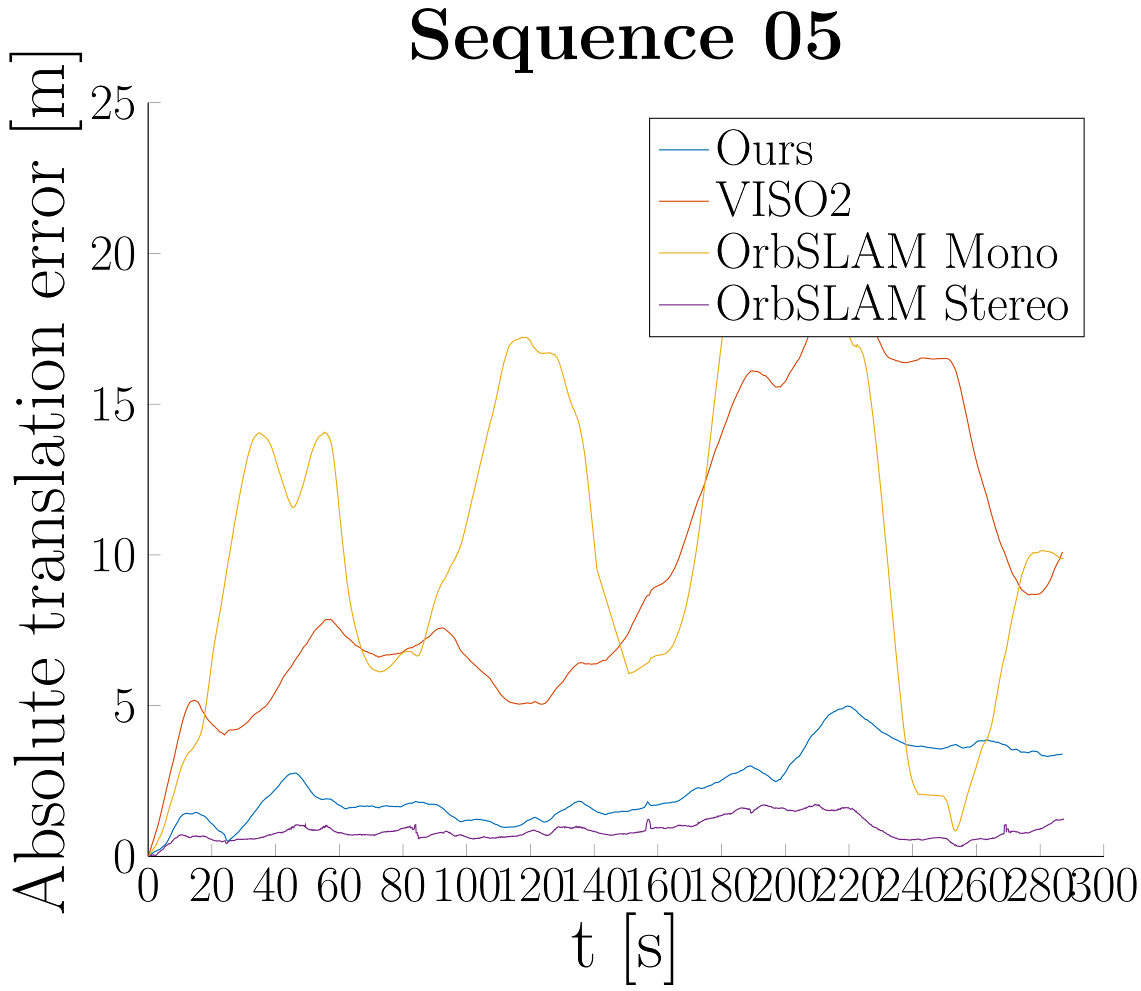

Figure 15: Norm of position error between estimate and ground truth, KITTI seq. 05

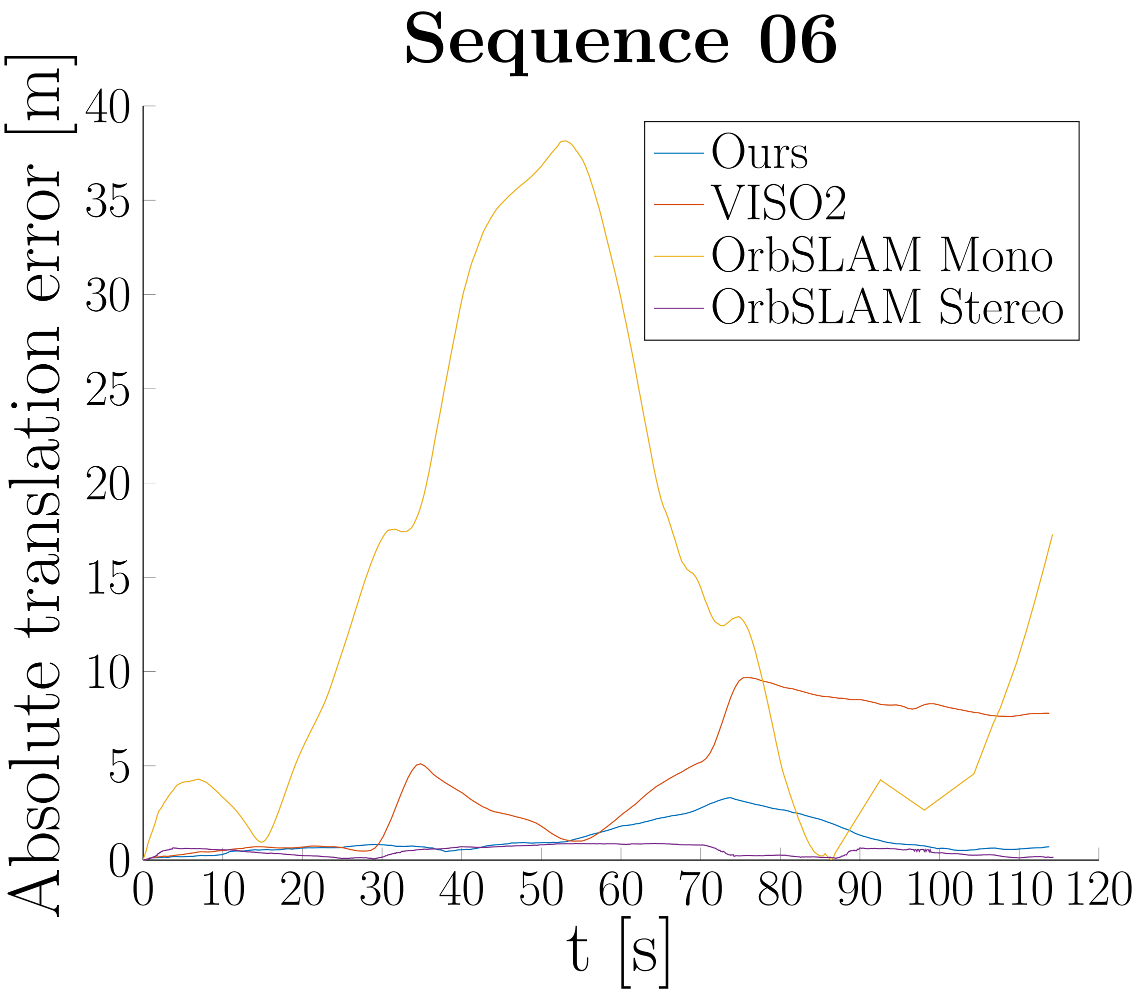

Figure 16: Norm of position error between estimate and ground truth, KITTI seq. 06

KITTI Sequence 05		
Method	Trans. err [%]	Rot. err [deg/m]
Ours	1.31	0.0038
VISO2	4.08	0.0050
ORBSLAM2 Mono	5.39	0.0019
ORBSLAM2 Stereo	0.63	0.0017

Figure 17: KITTI sequence 05 mean translational and rotational error over path lengths (100, 200, . . . , 800) meters.

KITTI Sequence 06		
Method	Trans. err [%]	Rot. err [deg/m]
Ours	0.77	0.0037
VISO2	1.81	0.0036
ORBSLAM2 Mono	6.71	0.0015
ORBSLAM2 Stereo	0.29	0.0013

Figure 18: KITTI sequence 06 mean translational and rotational error over path lengths (100, 200, . . . , 800) meters.

Chapter 4

Semantic SLAM via Semantic Keypoints

4.1 Introduction

While the methods described in the preceding section as demonstrated can fairly robustly perform object-level SLAM over long trajectories given only bounding box measurements, it suffers from several issues. First, objects are represented solely as a position $\ell^p \in \mathbb{R}^3$, while we would desire a richer representation as a full pose in $\ell^p \in SE(3)$. Second, even under a probabilistic association framework, the small two-dimensional measurement space and ambiguities to scale and rotation render data association difficult and often ambiguous. Finally, triangulation of bounding box centroids sometimes may result in unreliable object localization and may have initialization issues.

To ameliorate these issues, we now focus on a more complex object representation. In this chapter, an object is represented as its pose $o \in SE(3)$ in addition to a set of *semantic keypoints* $\ell_i \in \mathbb{R}^3$. These semantic keypoints consist of semantically meaningful points on the object that can be reliably found across different instances of the object class and meaningfully located in space. For example, the object class *car* may have among its semantic keypoints those of "front left wheel" and "rear right headlight." Using the methods of Pavlakos et al. (2017), an object's semantic

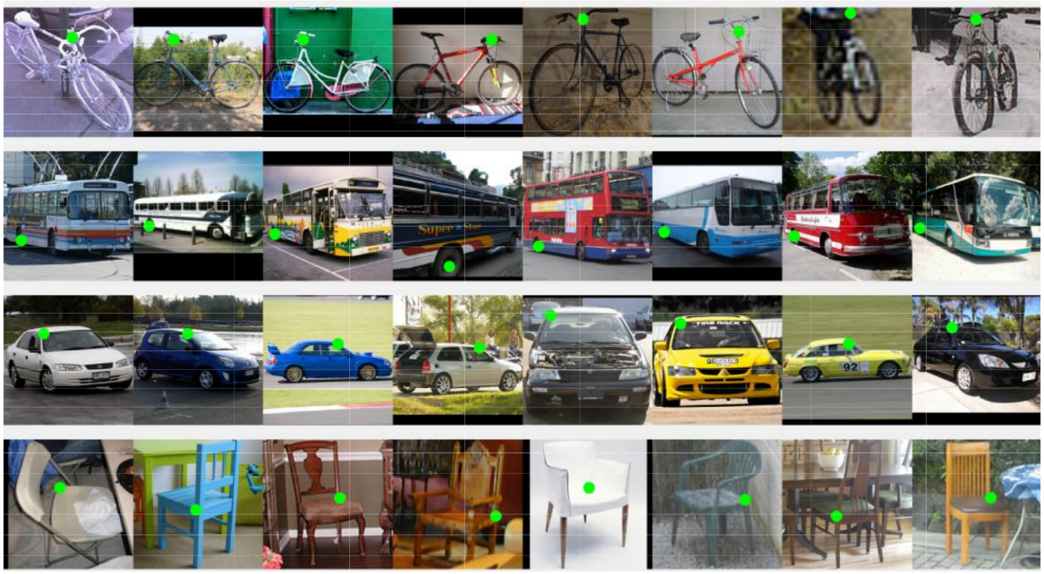

Figure 19: Example detected semantic keypoints for the object classes bicycle, bus, car, and chair (from Pavlakos et al. (2017))

keypoints are able to be reliably detected and identified across various viewpoints. For example, in Figure 19 various semantic keypoint detections for the object classes bicycle, bus, car, and chair are shown.

To account for intraclass variation of the shape of particular object instances, an object with in a particular class is represented as a static deformation of a shape model. This shape model consists of two components: (1) the mean shape (taken over all representative instances from the class) of each of its p semantic keypoints relative to its own pose o, along with several modes of possible shape variability (computed by principal component analysis). More specifically, let $\mathbf{S} \in \mathbb{R}^{3\times p}$ be a matrix consisting of an object's p keypoints represented in the object's own frame stacked horizontally. We then have

$$\mathbf{S}(c) = \mathbf{B}_0 + \sum_{i=1}^{k} c_i \mathbf{B}_i, \tag{4.1}$$

where $\mathbf{B}_0$ is the object class's mean shape and $\mathbf{B}_1, \ldots, \mathbf{B}_k$ are the modes of possible

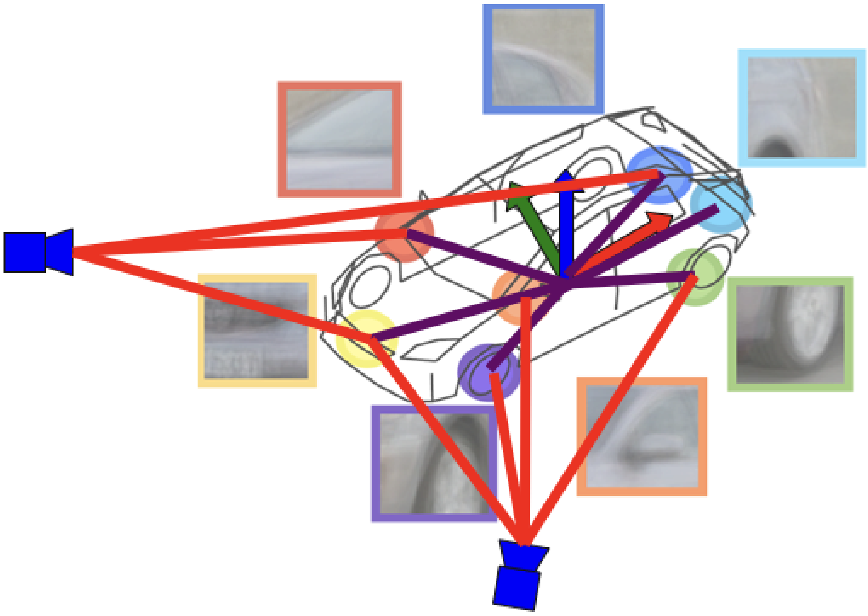

Figure 20: Example factor structure for a car object observed from two camera poses

shape variability (Pavlakos et al., 2017), written as a function of the deformation coefficients $c \in \mathbb{R}^k$.

Intuitively, repeated observations of a keypoint ℓ_j are used to triangulate it in space; the deformable shape model of the known object class is then used to indirectly estimate both the deformation coefficients c and the overall object pose o. See Figure 20 for an example of a car being observed from two camera poses. The semantic keypoints, denoted by colored circles and their associated image patches, are constrained in space by the corresponding image observations, denoted by red lines drawn to the camera positions. The object pose, represented by the axis in the middle of the car, is then constrained by the deformable object structure, denoted by the purple lines drawn to the keypoints.

The full SLAM problem under the expanded keypoint-based object model can now be stated as the following MAP estimation problem (cf. Equation (1.1)):

$$\begin{aligned}\hat{\mathcal{X}}, \hat{\mathcal{O}}, \hat{\mathcal{L}}, \hat{\mathcal{C}} &= \operatorname*{arg\,max}_{\mathcal{X},\mathcal{O},\mathcal{L},\mathcal{C}} \log p(\mathcal{X}, \mathcal{O}, \mathcal{L}, \mathcal{C} | \mathcal{Z}) &\quad (4.2)\\ &= \operatorname*{arg\,min}_{\mathcal{X},\mathcal{O},\mathcal{L},\mathcal{C}} \left[-\sum_{i=1}^{N} \log p(\mathcal{Z}_i | \mathcal{X}, \mathcal{O}, \mathcal{L}, \mathcal{C}) - \log p(\mathcal{X}, \mathcal{O}, \mathcal{L}, \mathcal{C}) \right], &\quad (4.3)\end{aligned}$$

where the second equality is from Equation (1.9), and where $\mathcal{O}$, $\mathcal{L}$, and $\mathcal{C}$ are the sets of all objects, semantic keypoints, and deformation coefficients, respectively. As an object measurement solely measures that object's semantic keypoints, and as our object model priors are independent of any particular trajectory, we can further simplify this as

$$\hat{\mathcal{X}}, \hat{\mathcal{O}}, \hat{\mathcal{L}}, \hat{\mathcal{C}} = \operatorname*{arg\,min}_{\mathcal{X},\mathcal{O},\mathcal{L},\mathcal{C}} \left[-\sum_{i=1}^{N} \log p(\mathcal{Z}_i | \mathcal{X}, \mathcal{L}) - \log p(\mathcal{O}, \mathcal{L}, \mathcal{C}) \right]. \quad (4.4)$$

Notice that the usual factor graph terms $p(\mathcal{Z}_i | \mathcal{X}, \mathcal{L})$ involve only the semantic keypoint positions $\mathcal{L}$; the full object poses are only determined through the object structure "prior" term $p(\mathcal{O}, \mathcal{L}, \mathcal{C})$.

4.2 Semantic Measurement Model

Formally, an object in the map consists of four elements: its class o^C, its pose $o \in SE(3)$, the positions of its keypoints $\ell_i \in \mathbb{R}^3$, $i = 1, \ldots, p$, and its deformation coefficients $c \in \mathbb{R}^k$. Note that we include the landmark positions ℓ_i explicitly as an optimization variable as we allow them to deviate from the positions implied from the object pose o and deformation parameters c.

When a camera $\mathbf{x}$ observes this object o, the measurement $h(\mathbf{x}, o)$ consists of projections of each of the object's semantic keypoints onto the image plane:

$$h(\mathbf{x}, o) = \begin{bmatrix} h_\pi(\mathbf{x}, \ell_1)^T & \cdots & h_\pi(\mathbf{x}, \ell_p)^T \end{bmatrix}^T, \tag{4.5}$$

where $h_\pi(\mathbf{x}, \ell)$ is the standard perspective projection of a point at ℓ onto a camera at pose $\mathbf{x}$.

The likelihood of a single semantic measurement $\mathbf{z} = [z_1^T \ \cdots \ z_p^T]^T$ is given as

$$p(\mathbf{x}, o, \ell, c|\mathbf{z}) = p(o, c|\mathbf{x}, \ell, \mathbf{z})p(\mathbf{x}, \ell|\mathbf{z}). \tag{4.6}$$

Note that the actual measurement $\mathbf{z}$ observes only the semantic keypoints ℓ and not the overall object pose o, and that given a particular class an object's pose and structure is uniquely determined by the position of its set of keypoints $\{\ell\}$. Thus, we have $p(o, c|\mathbf{x}, \ell, \mathbf{z}) = p(o, c|\ell)$, and so

$$\begin{aligned} p(\mathbf{x}, o, \ell, c|\mathbf{z}) &= p(o, c|\ell)p(\mathbf{x}, \ell|\mathbf{z}) && (4.7) \\ &= \frac{p(\ell|o, c)p(o, c)}{p(\ell)} \frac{p(\mathbf{z}|\mathbf{x}, \ell)p(\mathbf{x}, \ell)}{p(\mathbf{z})} && (4.8) \\ &\propto p(\mathbf{z}|\mathbf{x}, \ell)p(\ell|o, c)p(o, c), && (4.9) \end{aligned}$$

where we assume uniform priors $p(\ell)$, $p(\mathbf{x}, \ell)$, and $p(\mathbf{z})$.

Let us first examine the first term in (4.9), $p(\mathbf{z}|\mathbf{x}, \ell)$, and begin to compute log-probabilities as required in (4.4). As the measurements $\mathbf{z}$ are simply perspective projections of the keypoints onto an image plane with some additive (Gaussian) measurement noise, we have

$$\log p(\mathbf{z}|\mathbf{x}, \ell) \propto \log \prod_{i=1}^{p} p(z_i|\mathbf{x}, \ell_i) \tag{4.10}$$

$$\propto -\sum_{i=1}^{p} \|z_i - h_\pi(\mathbf{x}, \ell_i)\|^2_{\mathbf{R}}, \tag{4.11}$$

where $\mathbf{R} \in \mathbb{R}^{2\times 2}$ is the image measurement covariance matrix.

Next, let us examine the second term $p(\ell|o, c)$. This probability relates to the deformable object structure, and describes how likely a given object configuration is given the learned object basis structure. Let ${}^G\bar{q}_O$ and ${}^Gp_O$ be the rotation and position, respectively, of the object with respect to the global frame. Following equation (4.1), we have

$$\ell_i = R({}^G\bar{q}_O)\left(\mathbf{b}_0^i + \sum_{j=1}^{k} c_j \mathbf{b}_j^i\right) + {}^Gp_O, \quad i = 1, \dots, p \tag{4.12}$$

$$= R({}^G\bar{q}_O)\sigma_i(c) + {}^Gp_O, \quad i = 1, \dots, p, \tag{4.13}$$

where $\mathbf{b}_j^i$ is the ith column of $\mathbf{B}_j$, and $\sigma_i(c)$ is the ith structure-determined keypoint position in the local frame with deformation coefficients c (the ith column of $\mathbf{S}(c)$ as given in Equation 4.1).

Because the deformable shape model may not perfectly capture all intraclass variation, and because keypoint positions will not be estimated perfectly due to image noise and state uncertainty, we allow for estimated keypoints ℓ to vary from their structure $\sigma(c)$ by introducing a gaussian noise term $w_{st} \sim \mathcal{N}(0, \mathbf{R}_{struct})$. Here $\mathbf{R}_{struct}$ acts as more of a parameter describing how closely the learned object model fits the actual object class than a true measurement noise and should be chosen to be a relatively small value. We then write a probabilistic expression for ℓ_i as

$$\ell_i = R({}^G\bar{q}_O)\sigma_i(c) + {}^Gp_O + w_{st}, \quad i = 1, \ldots, p. \tag{4.14}$$

We can now write the desired log-probability as

$$\begin{align} \log p(\ell|o, c) &= \log \prod_{i=1}^{p} p(\ell_i|o, c) \tag{4.15} \\ &\propto - \sum_{i=1}^{p} \|\ell_i - R({}^G\bar{q}_O)\sigma_i(c) - {}^Gp_O\|^2_{\mathbf{R}_{struct}}. \tag{4.16} \end{align}$$

Finally, let us examine the term $p(o, c)$. We assume that the deformation coefficients are independent of the object pose and that the pose prior $p(o)$ is uniform, so we have $p(o, c) \propto p(c)$. As in Pavlakos et al. (2017), we use the term $p(c)$ as a simple regularizer on the coefficients c:

$$\log p(c) \propto -\lambda \|c\|_2^2, \tag{4.17}$$

where λ is a chosen regularization parameter.

Combining equations (4.9), (4.11), (4.16), and (4.17), we can now write the expression for the full semantic measurement log-probability,

$$\begin{split} - \log p(\mathbf{x}, o, \ell, c|\mathbf{z}) \propto \sum_{i=1}^{p} \|z_i - h_\pi(\mathbf{x}, \ell_i)\|^2_{\mathbf{R}} \\ + \sum_{i=1}^{p} \|\ell_i - R({}^G\bar{q}_O)\sigma_i(c) - {}^Gp_O\|^2_{\mathbf{R}_{struct}} + \lambda\|c\|_2^2. \end{split} \tag{4.18}$$

In practice, a single object is necessarily observed from multiple different camera poses. While each observation alters the measurement probability (equation (4.11)) associated with the object, the structure probabilities (equations (4.16) and (4.17)) remain the same. Suppose an object is observed by a set of measurements $\{\mathbf{z}_i\}_{i=1}^K$. We can write the full log-probability associated with this object as

$$\begin{aligned} -\log p(\mathbf{x}, o, \ell, c | \mathbf{z}_{1:K}) \propto \sum_{k=1}^{K} \sum_{i=1}^{p} \|[\mathbf{z}_k]_i - h_\pi(\mathbf{x}, \ell_i)\|^2_{\mathbf{R}} \\ + \sum_{i=1}^{p} \|\ell_i - R({}^G\bar{q}_O)\sigma_i(c) - {}^G p_O\|^2_{\mathbf{R}_{struct}} + \lambda \|c\|_2^2, \end{aligned} \tag{4.19}$$

where $[\mathbf{z}_k]_i$ is the ith keypoint measurement in measurement $\mathbf{z}_k$.

More generally, we can now consider the entire trajectory and thus the entire SLAM problem. Assuming a known data association, let β_k be the index of the mapped object that generated the kth semantic measurement, i.e. such that $\mathbf{s}_k$ is a measurement of object o_{β_k}. Next, as the number and structure of semantic keypoints per object varies with respect to the class of that object, let $p(o^C)$ be the number of keypoints for an object with object class o^C, and let $\sigma_i(c|o^C)$ be the local position of object o's keypoint as before *given* that object o is of class o^C. Finally, slightly abusing notation, let $\ell_i(o)$ be the ith keypoint of the keypoints that belong to object o. We can now write the following proposition.

Proposition 4. *Under the semantic keypoint measurement model described in Section 4.2, the Semantic SLAM problem (Problem 1) can be solved with the following factor graph estimation problem:*

$$\begin{aligned} \hat{\mathcal{X}}, \hat{\mathcal{L}} = \operatorname*{arg\,min}_{\mathcal{X}, \mathcal{L}} \sum_{t=1}^{T} \sum_{\mathbf{s}_k \in \mathcal{S}_t} \sum_{i=1}^{p(o^C_{\beta_k})} \|[\mathbf{s}_k]_i - h_\pi(\mathbf{x}_t, \ell_i(o_{\beta_k}))\|^2_{\mathbf{R}} \\ + \sum_{j=1}^{M} \sum_{i=1}^{p(o_j^C)} \|\ell_i(o_j) - R({}^G\bar{q}_{O_j})\sigma_i(c|o_j^C) - {}^G p_{O_j}\|^2_{\mathbf{R}_{struct}} \\ + \lambda \sum_{j=1}^{M} \|c_j\|_2^2 - \log p(\mathcal{Y}|\mathcal{X}) - \log p(\mathcal{I}|\mathcal{X}). \end{aligned} \tag{4.20}$$

As in Section 3.2.2, we can write this in an even more explicitly factor-graph based formulation by writing expressions for the semantic keypoint and semantic object structure factors. Similar to the definition of β_k, let α_k be the index of the sensor pose at which semantic measurement k was taken, i.e. such that measurement $\mathbf{s}_k$ was taken at pose $\mathbf{x}_{\alpha_k}$. Let a semantic keypoint factor for measurement $\mathbf{s}_k$ be given as

$$f_k^{key}(\mathcal{X}, \mathcal{L}) = \sum_{i=1}^{p(o_{\beta_k}^C)} \|[\mathbf{s}_k]_i - h_\pi(\mathbf{x}_{\alpha_k}, \ell_i(o_{\beta_k}))\|_{\mathbf{R}}^2, \tag{4.21}$$

and let a semantic structure factor for object o_j be given as

$$f_j^{struct}(\mathcal{X}, \mathcal{L}) = \sum_{i=1}^{p(o_j^C)} \|\ell_i(o_j) - R({}^G\bar{q}_{O_j})\sigma_i(c|o_j^C) - {}^Gp_{O_j}\|_{\mathbf{R}_{struct}}^2 + \lambda\|c_j\|^2. \tag{4.22}$$

Equation 4.20, and hence the full Semantic SLAM problem, is then equal to the following factor graph optimization:

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \arg\min_{\mathcal{X},\mathcal{L}} \sum_{k=1}^{K} f_k^{key}(\mathcal{X}, \mathcal{L}) + \sum_{j=1}^{M} f_j^{struct}(\mathcal{X}, \mathcal{L}) + \sum_{i=1}^{N_y} f_i^y(\mathcal{X}) + \sum_{t=1}^{T-1} f_t^{\mathcal{I}}(\mathcal{X}), \tag{4.23}$$

where as defined in Section 3.2.2, f^y and $f^{\mathcal{I}}$ are geometric and inertial factors defined in Equations (3.38) and (3.41), respectively, and N_y is the total number of distinct geometric feature tracks.

Note that unlike the exposition in Chapter 2 and the probabilistic association methods used in Chapter 3, in the above formulation we have assumed a known data association. It is worth exploring why this is the case. When extracting an object measurement as the centroid of a detected bounding box, a large of inherent measurement error is present; while a difference of a few pixels in the border of a bounding box may almost be imperceptible to a human observing the measurement quality on an image, the resulting difference of a few pixels in its centroid can have a very large effect on triangulation and estimation quality. In contrast, the detection of semantic keypoints is associated with a much smaller error; the movement of a keypoint detection in an image by several pixels is often easily noticeable, and errors typically seen in real world detections are on the order of pixels.

Furthermore, even if a bounding box centroid were able to be extracted without any associated measurement error, the measurement model itself used in Chapter 3 is highly ambiguous as it is invariant to several transformations of the camera and object pose. Scaling the distance between the camera and the object, scaling the size of the object, and rotating the object about any axis do not affect the received centroid measurement. As a result of this small (2-dimensional) measurement space, very distant objects (in either position or orientation in space) may generate very close (in the measurement space $\mathbb{R}^2$) measurements, resulting in numerous highly ambiguous measurements encountered and data association distributions that in general wide and multi-modal.

In contrast, the semantic keypoint measurement model used in this chapter alleviates many of the problems mentioned above. In general, due to the richer measurement model and much larger measurement space, sets of semantic keypoint measurements $\mathbf{s}_i$, $\mathbf{s}_j$ are only close in the $2p$-dimensional measurement space if objects o_i and o_j are close in both position and orientation in physical space, resulting in much fewer ambiguous measurements encountered. In practice, this results in a much more "peaky" data association distribution; if we were to implement the methods of Chapter 2 and compute data association weights w_{kj}, it is almost always be the case that for a given measurement $\mathbf{s}_k$, there exist one j such that $w_{kj} \approx 1$, and $w_{ks} \approx 0$ for all $s \neq j$. Examining the equations in Proposition 2, we can see that this then reduces exactly to the case of computing a maximum likelihood data association and assuming it known in the factor graph optimization over $\mathcal{X}$, $\mathcal{L}$. Additional comparison with the data association simulation results from Section 2.2 would suggest that a maximum likelihood association would produce better performance in practice for these relatively low noise measurements, and this is indeed what we saw in experiments.

4.3 Implementation

While the factor graph optimization in Chapter 3 to solve the Semantic SLAM problem given bounding box measurements is largely similar to a typical point feature visual SLAM estimation, the inclusion of the semantic structures f_j^{struct} produce multiple complicated relationships between variables in the pose graph. As a result, methods designed to solve the former may fail to work well or efficiently on the latter; in particular, after implementing Equation (4.23) with GTSAM as the optimization backend, we observed iSAM and iSAM2 to require many more linearizations than before and produce poor performance in many situations.

As a result, the implementation of semantic keypoint SLAM requires a more thoughtful implementation strategy. In particular, we adopt a sliding window solution method. Upon receiving the Tth keyframe and extracting the semantic measurements $\mathcal{S}_T$, a sliding window of length W is created. Within the estimation, the pose variables $\mathbf{x}_1, \ldots, \mathbf{x}_{T-W}$ are frozen, to be held constant in the factor graph optimization, leaving $\mathbf{x}_{T-W+1}, \ldots, \mathbf{x}_T$ as free variables in the estimation.

Let $\mathcal{O}_t$ be the set of semantic objects that were observed in the tth keyframe, i.e. using the data association notation from the preceding section, $\mathcal{O}_t = \{o_{\beta_k} \mid \mathbf{s}_k \in \mathcal{S}_t\}$. The objects observed at any point in the sliding window are then given by $\cup_{t=T-W+1}^{T} \mathcal{O}_t$ and are allowed to vary in the optimization carried out at time step T; all other objects $\mathcal{L} \setminus \cup_{t=T-W+1}^{T} \mathcal{O}_t$ are treated as constants. In this way, we are able to efficiently perform a local optimization while ensuring all relevant objects' pose is updated given all known information.

As a sliding window in the above fashion obviously does not allow for loop closures to be properly handled, we add a separate loop closing thread that is run in parallel to the sliding window optimization. On detection of a loop closure (defined as the observation of an object o_j that is not in the sliding window's visible set $o_j \notin \cup_{t=T-W+1}^{T-1} \mathcal{O}_t$), a separate loop closure thread is started. This loop closure thread optimizes the same factor graph, Equation (4.23), as the sliding window, per-keyframe optimizing thread, but over an expanded set of variables. Let $\mathbf{x}_a$ be the first $\mathbf{x}$ that observed the loop-closing object o_j; the set of variables allowed to vary within the loop closure optimization is then the sensor poses $\mathbf{x}_a, \mathbf{x}_{a+1}, \ldots, \mathbf{x}_T$ and the objects $\cup_{t=a}^{T} \mathcal{O}_t$. This (potentially large, slow) optimization is allowed to run in the background until completion, at which point the sensor poses and objects are updated with the result.

We implemented the keypoint-based Semantic SLAM system as described, and specifically implemented Equation (4.23), using Google's Ceres nonlinear optimization library (Agarwal et al.) as the optimization backend. The Ceres solver was used to solve both the sliding window local optimization and the background loop closing optimizations. For the front end of our implementation, we chose the simple method of selecting every 10th camera frame as a semantic keyframe. The front end applies to each image the Faster R-CNN object detector (Ren et al., 2015) to detect object bounding boxes. To each detected bounding box, we applied the semantic keypoint detector from Pavlakos et al. (2017) to detect the object's semantic keypoints. Next, the Mahalanobis distance between each measurement and each object in the map of the same class is computed, and a simple maximum likelihood data association is performed with the Hungarian algorithm (Munkres, 1957). The resulting keypoint measurements and their data associations were then used within a custom factor graph library built around the Ceres solver as mentioned above.

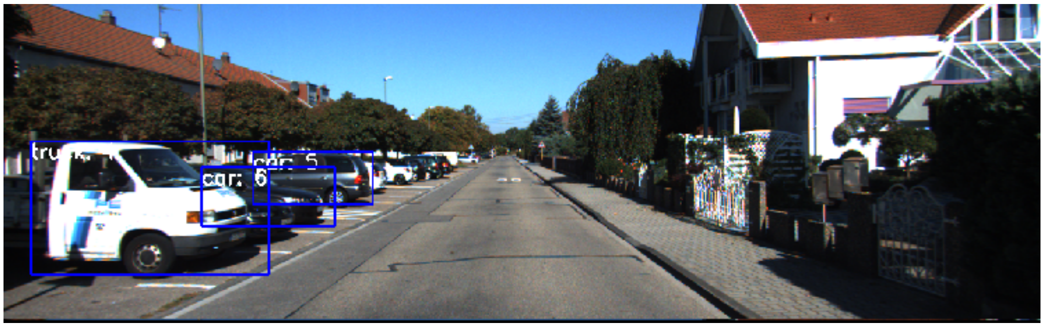

Figure 21: Image from early in the KITTI dataset trajectory 05, showing a line of parked cars.

4.3.1 Visual-Semantic SLAM on KITTI

In the first set of experiments, we apply our algorithm again to the outdoor KITTI (Geiger et al., 2012) dataset. The KITTI dataset consists of a vehicle equipped with several sensors driving through an urban environment, and parked cars were used as the estimated semantic objects. Rather than including inertial odometry factors as shown in Equation (4.23), we instead include simple relative pose factors computed using the VISO2 (Geiger et al., 2011) visual odometry algorithm. We applied our algorithm to trajectory 05 in the KITTI Geiger et al. (2012) outdoor dataset.

See Figure 21 for an example image taken from early in the KITTI dataset trajectory 05, showing a challenging example of a line of parked cars with numerous occlusions and that is traversed at relatively high speed. Our algorithm's estimate of the trajectory along with the estimated cars is shown in Figure 22. Although some detections were missed, the detections and estimated poses and keypoints are very accurate given the conditions.

In Figure 23, our algorithm's trajectory and map estimate after a longer trajectory is shown. Even in long trajectories with numerous objects in the map and several loop closure situations, our algorithm is able to localize not only the camera's position along the trajectory, but also the position and orientation of parked cars along the

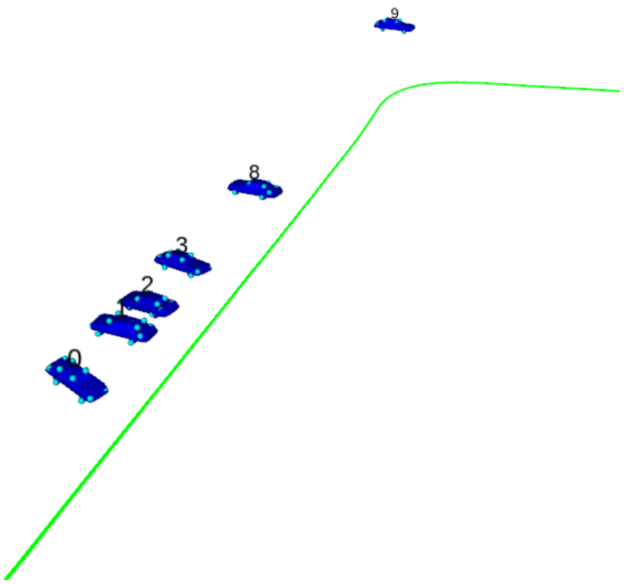

Figure 22: Resulting trajectory from our keypoint-based Semantic SLAM system, showing the area of space that is captured in the photograph from Figure 21.

path.

4.3.2 Clearpath Husky Experiments

We additionally experimented with the application our semantic factors to a dataset collected with a Clearpath Husky robot, as shown in figure 24. LiDAR and camera data was collected from trajectories in an urban environment and processed offline. See Figure 25 for an example image collected along the trajectory along with semantic keypoints detected on a window, and see Figure 26 for the system's estimate of the robot trajectory and map at the time the picture in Figure 25 was taken. Note the one detected and localized window shown in the estimate, along with the four semantic keypoints that correspond to the window corners.

Continuing the trajectory Figure 27 shows a later point in the experiment after a longer path through the urban environment, showing the estimated trajectory, occupancy grid, and several estimated window objects.

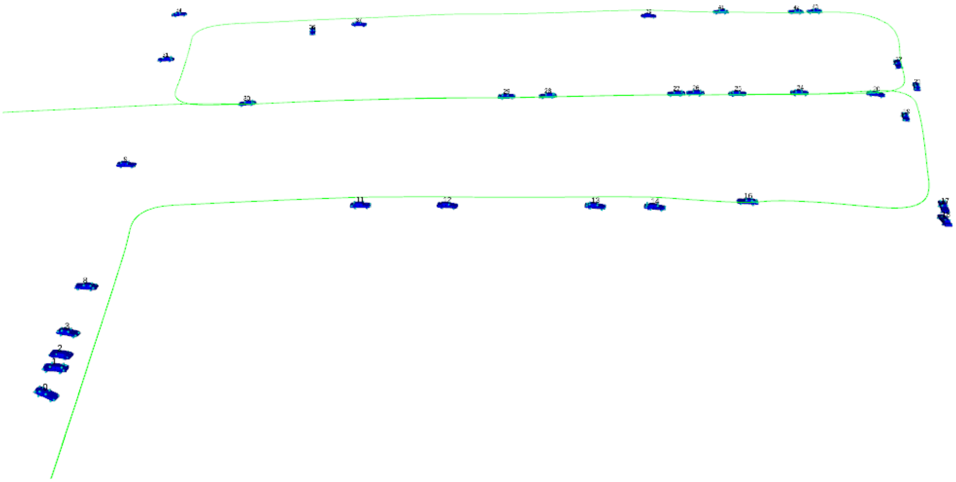

Figure 23: Estimated trajectory and objects from KITTI trajectory 05 after a longer duration

Our method is also able to perform well at single-object localization up close, with applications of manipulation or other interaction where precise pose estimates are necessary. A robot was driven on a straight line trajectory towards a black crate placed on the ground and images were continuously taken of the crate. See Figure 28 for an example of an image as the robot nears the crate, and Figure 29 for the estimated trajectory and crate pose along with the position of the crate's semantic keypoints. Note how the keypoints line up directly over the occupancy grid-shown obstacle that the crate represents, as well as the subjective quality of the keypoint localization relative to their displayed positions on the crate in Figure 28.

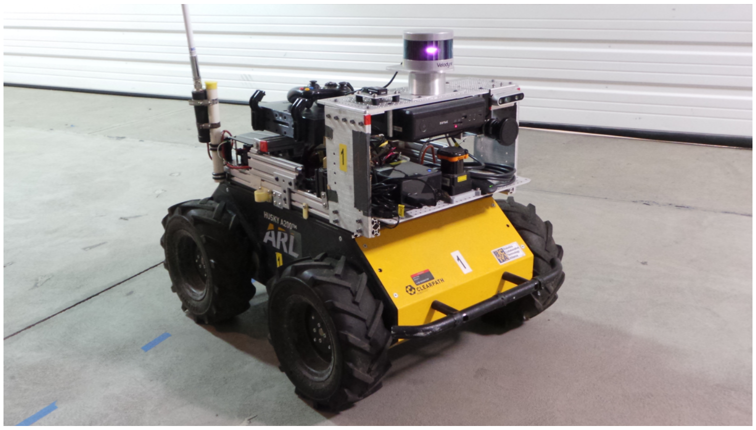

Figure 24: Clearpath Husky robot used in first series of experiments

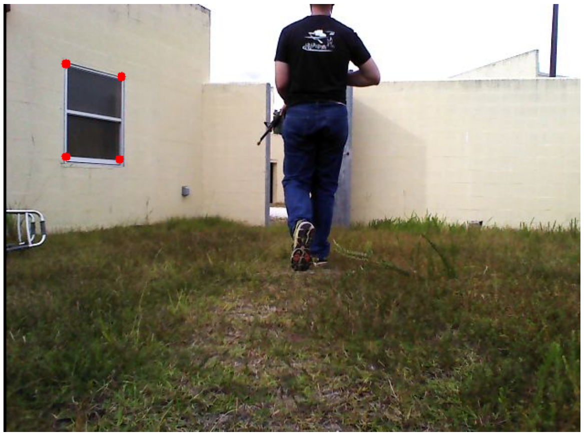

Figure 25: Example image collected from Husky robot along with semantic keypoint detections

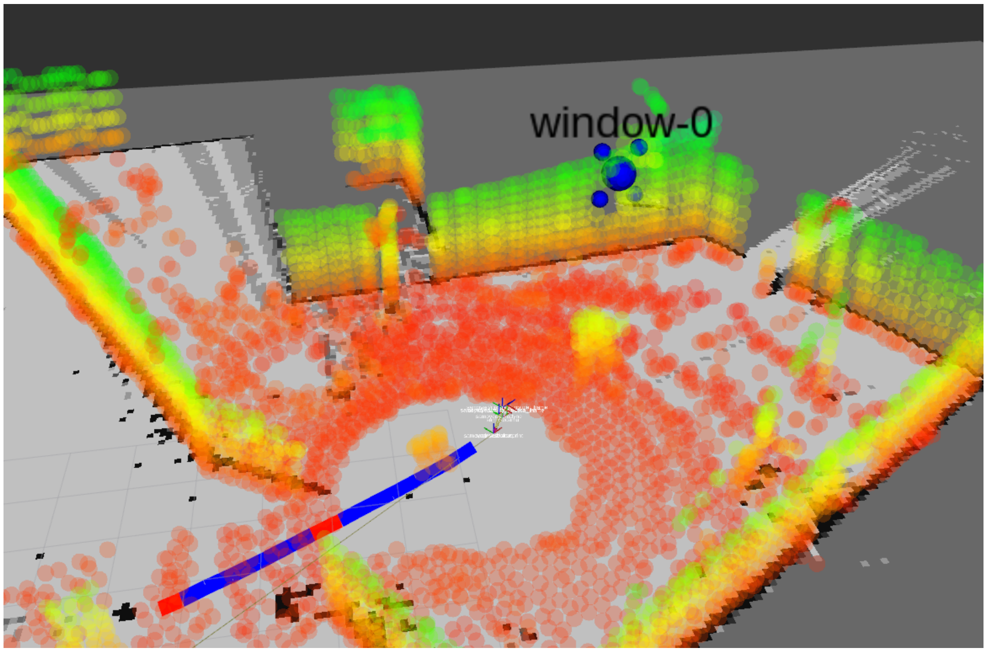

Figure 26: Estimate of the robot trajectory, map, and detected window object at the time at which the image in Figure 25 was taken. The central sphere of the window corresponds to the object position while the four bordering spheres represent the semantic keypoint locations. The grid on the ground displays the estimated occupancy grid map, and the translucent points display the most recent LIDAR measurement data.

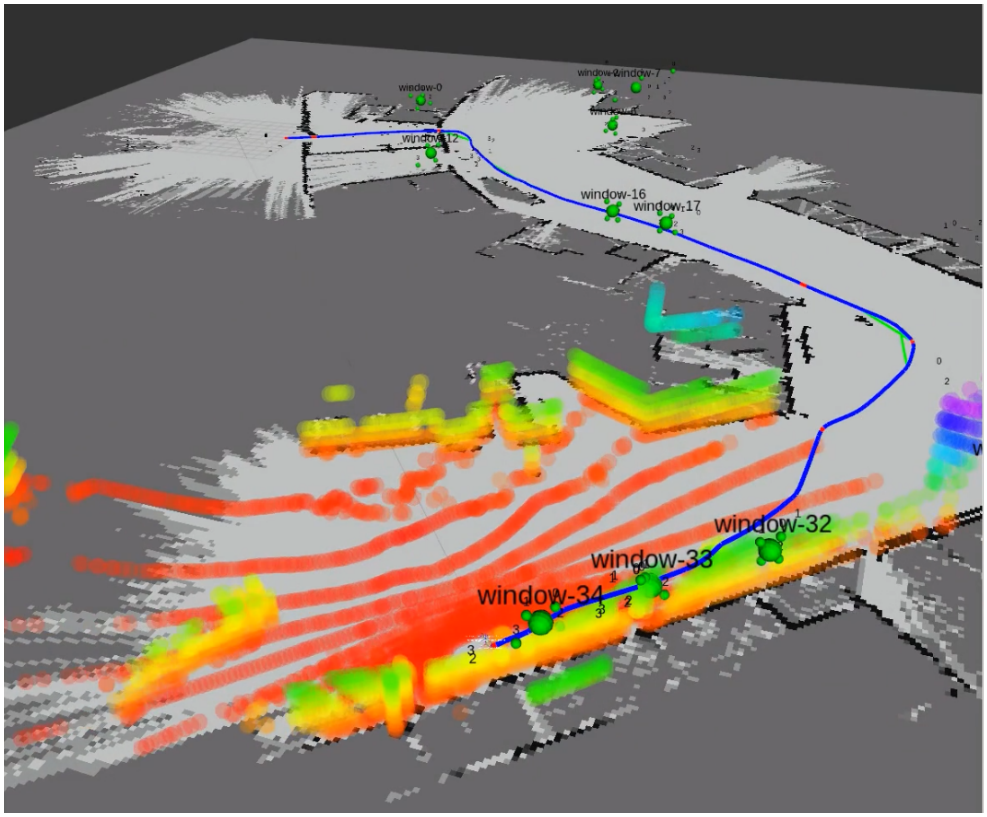

Figure 27: Estimate of the trajectory after a longer path through the same environment as seen in Figures 25 and 26.

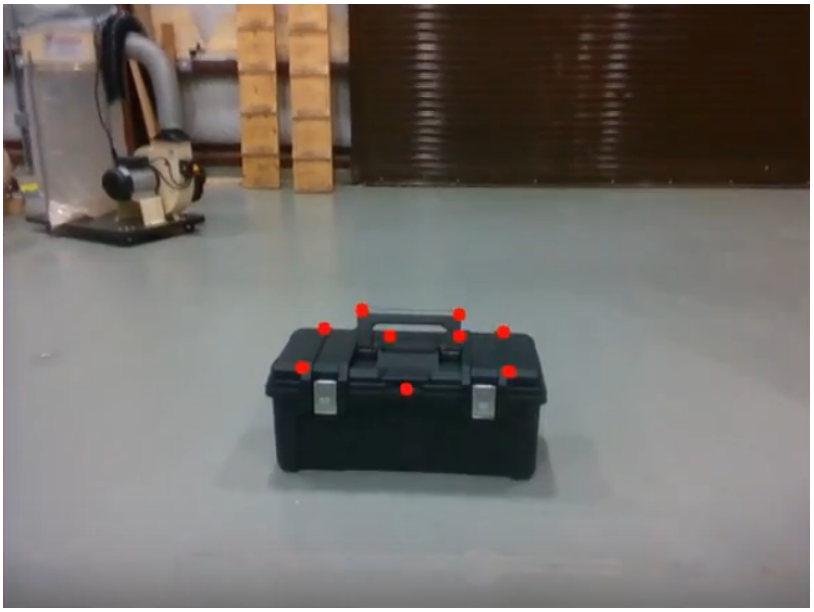

Figure 28: Image as a robot nears a black crate placed on the ground along with detected semantic keypoints.

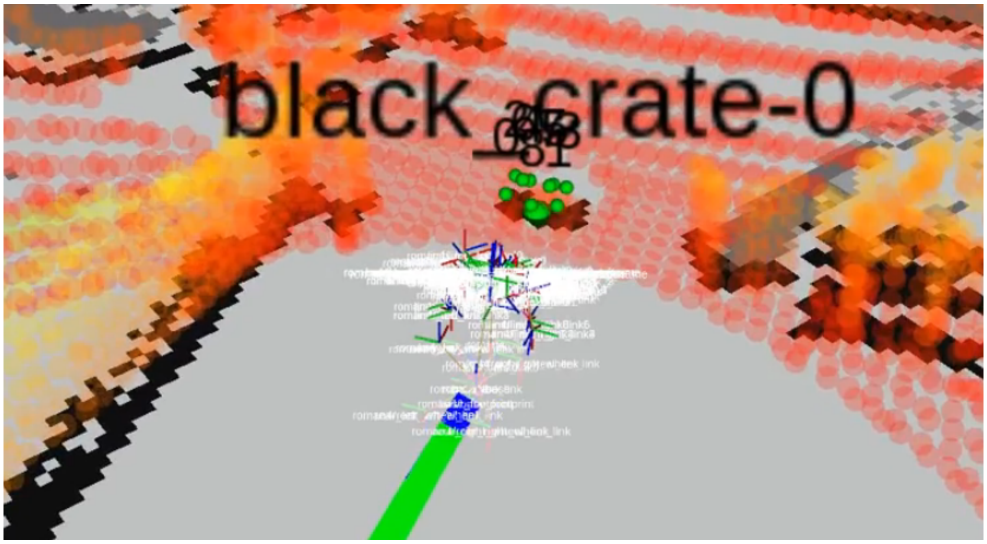

Figure 29: Estimate of the trajectory as the robot approaches the crate along with its detected pose and 3D keypoint positions.

Chapter 5

Reactive Planning in Unexplored Semantic Environments

5.1 Introduction

The inclusion of semantic information within a SLAM framework not only can improve the accuracy of mapping and localization systems and provide human-understandable maps to an operator, but can also be used to improve autonomy and plan high level semantically-meaningful missions. In particular, in this chapter we consider a particular application of the methods outlined in Chapter 4: the problem of navigation in unexplored semantic environments.

The problem of *navigation* is a fundamental problem in robotics. In the case of navigating a perfectly known environment, the problem is reducible to a purely reactive (i.e. closed loop state feedback based) solution (Rimon and Koditschek, 1992). In the case of an imperfectly known and explored environment, "doubly reactive" methods (methods that not only construct the robot's trajectory online but also the control vector field that generate it) have shown success in the case of sufficiently "nice" obstacles (spaced sufficiently far apart and convex) (Paternain et al., 2018; Ilhan et al., 2020).

Densely cluttered or non-convex obstacles, however, have generally required incremental and random sampling-based planning, the probabilistic completeness guarantees of which can be slow to realize in practice (Noreen et al., 2016). Furthermore, the setting of navigation within an imperfectly known environment has received little theoretical attention. Some exceptions include considerations of optimality in unknown spaces, online modifications to temporal logic specifications or deep learning algorithms that assure safety against obstacles, or the use of trajectory optimization along with offline computed reachable sets for online policy adaptations. However, none of these advances has achieved simultaneous guarantees of obstacle avoidance and convergence. In this chapter we present an algorithm that extends these two guarantees of obstacle avoidance and convergence to the setting of an environment containing non-convex and unknown or moving targets.

5.2 Problem Formulation

We consider a circular robot with radius r, centered at position $\mathbf{x} \in \mathbb{R}^2$, navigating a compact, polygonal and potentially non-convex workspace $\mathcal{W} \subset \mathbb{R}^2$ with known boundary $\partial\mathcal{W}$, towards a target location $\mathbf{x}_d \in \mathcal{W}$. The robot is assumed to possess a sensor with fixed range R for recognizing familiar objects and estimating the distance to nearby obstacles. We further define the enclosing workspace as the convex hull of the closure of the workspace $\mathcal{W}$, i.e.

$$\mathcal{W}_e \triangleq \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} \in \text{Conv}(\overline{\mathcal{W}})\}. \tag{5.1}$$

The workspace $\mathcal{W}$ is cluttered by a finite but unknown number of disjoint and fixed obstacles, denoted by $\tilde{\mathcal{O}} \triangleq \{\tilde{O}_1, \tilde{O}_2, \dots\}$. This set $\tilde{\mathcal{O}}$ also includes any non-convex "intrusions" of the boundary of the physical workspace $\mathcal{W}$ into $\mathcal{W}_e$. We define the *freespace* $\mathcal{F}$ as the set of collision free placements of the robot within the physical workspace, i.e. the set of collision-free placements of the closed ball $\overline{B(\mathbf{x}, r)}$ centered at $\mathbf{x}$ with radius r in $\mathcal{W}$:

$$\mathcal{F} \triangleq \left\{ \mathbf{x} \in \mathcal{W}_e \mid \overline{B(\mathbf{x}, r)} \subseteq \mathcal{W}_e \setminus \bigcup_i \tilde{O}_i \right\}, \tag{5.2}$$

and we similarly define the *enclosing freespace* as

$$\mathcal{F}_e \triangleq \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} \in \mathrm{Conv}(\overline{\mathcal{F}})\}. \tag{5.3}$$

We assume that none of the positions of any of the obstacles $\tilde{O}_i$ are a-priori known, however we assume that a subset $\tilde{P} \triangleq \{\tilde{P}_i\}_{i \in \mathcal{N}_P} \subseteq \tilde{\mathcal{O}}$ of these obstacles, indexed by $\widetilde{\mathcal{N}_P} \triangleq \{1, \ldots, N_P\}$, is "familiar" in the sense of having a known and readily recognizable polygonal geometry, that the robot can instantly identify and localize. In this chapter, this corresponds exactly to the set of known semantic objects which are detectable and localizable via the methods of Chapter 4. The remaining obstacles in $\tilde{\mathcal{C}} \triangleq \tilde{\mathcal{O}} \setminus \tilde{\mathcal{P}}$ are assumed to be strongly convex with an additional curvature constraint as in Assumption 2 from Arslan and Koditschek (2019) but are otherwise completely unknown to the robot.

To simplify the notation, we neglect the robot dimensions, and assume that the robot is a point navigating within the freespace $\mathcal{F}$ by dilating each obstacle in $\tilde{\mathcal{O}}$ by r. We denote the set of dilated obstacles in $\tilde{\mathcal{O}}$, $\tilde{\mathcal{P}}$, and $\tilde{\mathcal{C}}$ by $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{C}$, respectively. We then describe each polygonal obstacle $P_i \in \mathcal{P} \subseteq \mathcal{O}$ by an *obstacle function* $\beta_i(\mathbf{x})$, a real-valued map providing an implicit representation of the form

$$P_i = \{\mathbf{x} \in \mathbb{R}^2 \mid \beta_i(\mathbf{x}) \leq 0\} \tag{5.4}$$

that is constructable by the robot after it has localized the obstacle P_i.

We finally require the following separation assumptions on the obstacles:

Assumption 1. *Each obstacle $C_i \in \mathcal{C}$ has a positive clearance $d(C_i, C_j) > 0$ from any other obstacle $C_j \in \mathcal{C}$, $j \neq i$.*

Assumption 2. *Each obstacle $C_i \in \mathcal{C}$ satisfies $d(C_i, \partial\mathcal{F}) > 0$.*

Assumption 3. *For each $P_i \in \mathcal{P}$, there exists an $\epsilon_i > 0$ such that the set*

$$S_{\beta_i} \triangleq \{\mathbf{x} \mid \beta_i(\mathbf{x}) \leq \epsilon_i\} \tag{5.5}$$

has a positive clearance $d(S_{\beta_i}, C) > 0$ from any obstacle $C \in \mathcal{C}$.

We additionally impose an assumption stating that a solution exists:

Assumption 4. *The freespace $\mathcal{F}$ is path-connected.*

Considering our robot with first order dynamics $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x})$ and equipped with these assumptions, the navigation problem consists of finding a Lipschitz continuous controller $\mathbf{u} : \mathcal{F} \rightarrow \mathbb{R}^2$ that leaves the freespace $\mathcal{F}$ positively invariant and directs the robot towards the goal $\mathbf{x}_d \in \mathcal{F}$.

5.3 Approach and Planning Space Construction

An overview of the solution is as follows. We interpolate a sequence of spaces between the physical space and a topologically equivalent but geometrically simple model space. Within this simpler model space, we design a control input which we can then transform through the inverse of the diffeomorphism between the physical and model space to find the commands in the physical space. In the following sections, we outline the distinct representations of the environment that we refer to as the planning spaces. An outline of the planning spaces and their relation can be seen in Figure 30

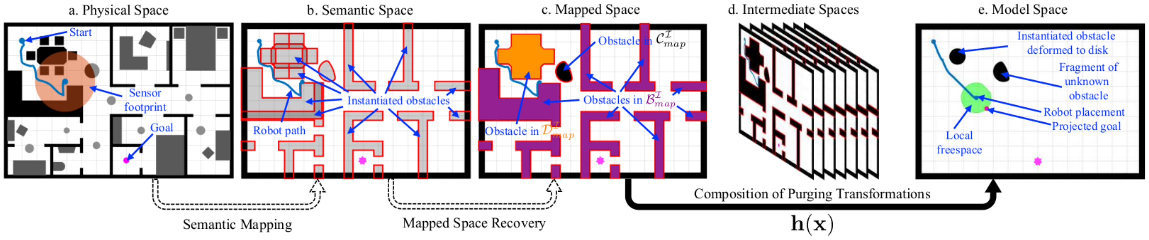

Figure 30: Illustration of the planning spaces (from Vasilopoulos et al. (2020a).)

5.3.1 Physical Space

The *physical space* denotes the actual physical workspace; this is the workspace $\mathcal{W}_e$ punctured by the obstacles $\tilde{\mathcal{O}}$, the knowledge of which is inaccessible to the robot. The robot navigates in this space towards the desired location $\mathbf{x}_d$ and discovers and localizes new objects along the way. Let $\tilde{\mathcal{P}}_{\mathcal{I}} = \{\tilde{P}_i\}_{i \in \mathcal{I}} \subseteq \tilde{\mathcal{P}}$ denote the set of physically instantiated familiar objects, i.e. the set of objects whose geometry and pose is either known to the robot before hand (for example the workspace intrusions from a known wall or room layout), or those objects that have been detected and localized online and whose pose has been estimated by the semantic SLAM system. This set is indexed by a set $\mathcal{I} \subseteq \mathcal{N}_P$.

5.3.2 Semantic Space

The semantic space $\mathcal{F}^{\mathcal{I}}_{sem}$ describes the robot's actual current and continuously updated information about the environment. This consists of the $|\mathcal{I}|$ instantiated familiar obstacles as well as the observable portions of the unrecognized obstacles in the space. We denote this latter set of unrecognized obstacles within the semantic space by $\mathcal{C}_{sem} \triangleq \{C_i\}_{i \in \mathcal{J}_C}$, which is indexed by a set $\mathcal{J}_C \subseteq \mathcal{N}_C$. Similarly, we denote the former set of familiar obstacles within the semantic space as the set $\mathcal{P}^{\mathcal{I}}_{sem} \triangleq \sqcup_{i \in \mathcal{I}} P_i$. Note in particular the use of a disjoint union in the construction here; it will become important when we consider the Mapped Space. Within this space, as we are considering the sets of dilated obstacles, the robot is treated as a single point.

5.3.3 Mapped Space

Note that Assumptions 1 through 3 do not exclude obstacles in $\mathcal{P}$ from overlapping with each other, and as such, the semantic space does not explicitly contain topological information about the explored environment. Hence, we form a *mapped space* by taking (non-disjoint) unions of elements of $\mathcal{P}^{\mathcal{I}}_{sem}$, creating a new set of consolidated familiar obstacles $\mathcal{P}^{\mathcal{I}}_{map} \triangleq \{P_i\}_{i \in \mathcal{J}^{\mathcal{I}}}$. This set is indexed by the set $\mathcal{J}^{\mathcal{I}}$, with $|\mathcal{J}^{\mathcal{I}}| \leq |\mathcal{I}|$. The space additionally includes copies of the unknown obstacles, $\mathcal{C}_{map} = \mathcal{C}_{sem}$, because the assumptions preclude these obstacles from overlapping.

The next step is to separate the mapped familiar objects that intersect the boundary of the freespace with those that do not. In the construction of the diffeomorphism to the simple model space, those that intersect the boundary should be merged into the boundary itself, while those that do not should be deformed into disks. Thus, for any connected component P of $\mathcal{P}^{\mathcal{I}}_{map}$ that intersects the boundary $\partial\mathcal{F}_e$, we let $B \triangleq P \cap \mathcal{F}_e$ and include B in a new set $\mathcal{B}^{\mathcal{I}}_{map}$ indexed by $\mathcal{J}^{\mathcal{I}}_{\mathcal{B}}$. Similarly, the rest of the components in $\mathcal{P}^{\mathcal{I}}_{map}$ that do not intersect $\partial\mathcal{F}_e$ are included in a set $\mathcal{D}^{\mathcal{I}}_{map}$, indexed by $\mathcal{J}^{\mathcal{I}}_{\mathcal{D}}$.

5.3.4 Model Space

Lastly, we have the *model space* $\mathcal{F}^{\mathcal{I}}_{model}$. This space is a topologically equivalent but geometrically simplified version of the mapped space $\mathcal{F}^{\mathcal{I}}_{map}$. This model space $\mathcal{F}^{\mathcal{I}}_{model}$ has the same boundary as $\mathcal{F}_e$, and the $|\mathcal{J}_C|$ unrecognized visible obstacles in the mapped space are simply copied into the model space. The $|\mathcal{J}^{\mathcal{I}}_{\mathcal{D}}|$ consolidated familiar obstacles in $\mathcal{D}^{\mathcal{I}}_{map}$ are deformed to disks, and the boundary consolidated obstacles $\mathcal{B}^{\mathcal{I}}_{map}$ are merged into the boundary $\partial\mathcal{F}_e$ to make $\mathcal{F}^{\mathcal{I}}_{map}$ and $\mathcal{F}^{\mathcal{I}}_{model}$ topologically equivalent through a mapping $\mathbf{h}^{\mathcal{I}}$, which we will describe next.

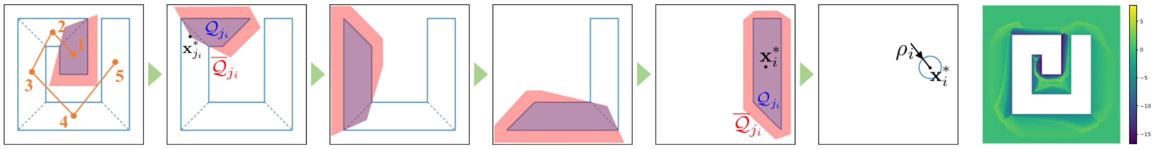

Figure 31: Diffeomorphism construction via convex decomposition (from Vasilopoulos et al. (2020a))

5.4 Diffeomorphism Construction

The construction of the diffeomorphism $\mathbf{h}^{\mathcal{I}}$ between $\mathcal{F}_{map}^{\mathcal{I}}$ and $\mathcal{F}_{model}^{\mathcal{I}}$ relies on the convex decomposition of each obstacle $P \in \mathcal{P}_{map}^{\mathcal{I}}$. We assume for each obstacle that the robot has access to such a decomposition; here, we compute such a decomposition of the obstacle polygons using Greene's method and its C++ implementation in CGAL. As shown in Figure 31, such a decomposition results in a tree of convex polygons $\mathcal{T}_{P_i} \triangleq (\mathcal{V}_{P_i}, \mathcal{E}_{P_i})$ corresponding to P_i, where $\mathcal{V}_{P_i}$ is a set of vertices identified with each component convex polygon and $\mathcal{E}_{P_i}$ is a set of edges corresponding to polygon adjacency. We can therefore pick any polygon as the root of the tree $\mathcal{T}_{P_i}$ and construct the tree based on the adjacency properties.

As the present work is focused largely on the reactive planning controller design and diffeomorphism construction as an application of the semantic mapping methods outlined in Chapter 4, we leave the precise mathematics and further details of the construction of $\mathbf{h}^{\mathcal{I}}$ to the technical report (Vasilopoulos et al., 2020b) and proceed with a high level description. The map $\mathbf{h}^{\mathcal{I}}$ is constructed iteratively in several steps by composing individual purging transformations for all leaf polygons of all obstacles P in $\mathcal{B}_{map}^{\mathcal{I}}$ and $\mathcal{D}_{map}^{\mathcal{I}}$. This composition continues, during execution time, until all root polygons have been reached.

The construction of each of these individual purging transformations can be seen in Figure 31. Each leaf polygon is associated with a *center* (denoted for polygon j_i

as $\mathbf{x}_{j_i}^*$) as well as a *polygonal collar* $\overline{\mathcal{Q}}_{j_i}$, shown in red. The purging transformation is then defined in terms of a function σ_{j_i} that smoothly varies from 0 outside of the collar $\overline{\mathcal{Q}}_{j_i}$ to 1 inside the polygon j_i, allowing for a smooth transformation collapsing the leaf polygon j_i into its parent to be written as

$$\mathbf{h}_{j_i}^{\mathcal{I}} \triangleq \sigma_{j_i}(\mathbf{x})\mathbf{x}_{j_i}^* + (1 - \sigma_{j_i}(\mathbf{x}))\mathbf{x}. \tag{5.6}$$

The final step is then the transformation of each root polygon into a disk, as shown in the second to last transformation in Figure 31. This transformation is constructed in a similar fashion along with a *deforming factor* for each obstacle; see Vasilopoulos et al. (2020b) for details.

5.5 Reactive Planning Algorithm

Equipped with the diffeomorphism $\mathbf{h}^{\mathcal{I}}$ between the mapped space $\mathcal{F}_{map}^{\mathcal{I}}$ and the model space $\mathcal{F}_{model}^{\mathcal{I}}$, we can now describe the reactive planning algorithm itself. Because we assume the space is a priori unexplored and new obstacles enter the robot's field of view as it progresses through the environment, new obstacles are incorporated and the semantic map is modified over time. Thus, we give a hybrid systems description of the controller, where each mode is defined by an index set $\mathcal{I} \in 2^{\mathcal{N}_P}$ of familiar obstacles stored in the semantic map, the guards describe the sensor trigger events where a previously unexplored obstacle is discovered, and the resets describe transitions to new modes that are equal to the identity in the physical space but may result in discrete jumps of the robot position in model space as a result of the diffeomorphism $\mathbf{h}^{\mathcal{I}}$ being updated to account for the newly discovered object.

In each mode $\mathcal{I}$, the robot with dynamics $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x})$, $\mathbf{u} \in \mathbb{R}^2$, is given as (Vasilopoulos et al., 2020a)

$$\mathbf{u}^{\mathcal{I}}(\mathbf{X}) = k \left[D_{\mathbf{x}}\mathbf{h}^{\mathcal{I}}\right]^{-1} \cdot \left(\mathbf{v}^{\mathcal{I}} \circ \mathbf{h}^{\mathcal{I}}(\mathbf{x})\right), \tag{5.7}$$

where $D_{\mathbf{x}}$ is the derivative operator with respect to $\mathbf{x}$, and the control input $\mathbf{v}^{\mathcal{I}}$ in the model space is given as

$$\mathbf{V}^{\mathcal{I}}(\mathbf{y}) = -\left(\mathbf{y} - \Pi_{\mathcal{LF}(\mathbf{y})}(\mathbf{y}_d)\right), \tag{5.8}$$

where $\mathbf{y} = \mathbf{h}^{\mathcal{I}}(\mathbf{x})$ and $\mathbf{y}_d = \mathbf{h}^{\mathcal{I}}(\mathbf{x}_d)$ are the robot's position and desired position in the model space, respectively, and $\Pi_{\mathcal{LF}(\mathbf{y})}$ is the projection onto the convex local freespace for $\mathbf{y}$, $\mathcal{LF}(\mathbf{y})$, defined as the Voronoi cell separating $\mathbf{y}$ from all model space obstacles.

The main result of Vasilopoulos et al. (2020a) and this chapter is then the following Theorem:

Theorem 1. *With $\mathcal{I}$ the terminal mode of the hybrid controller, the reactive controller in* (5.7) *leaves the freespace $\mathcal{F}^{\mathcal{I}}_{map}$ positively invariant, and asymptotically reaches a constant $\mathbf{x}_d$ with its unique continuously differentiable flow from almost any placement $\mathbf{x} \in \mathcal{F}^{\mathcal{I}}_{map}$, while strictly decreasing $\|\mathbf{h}^{\mathcal{I}}(\mathbf{x}) - \mathbf{h}^{\mathcal{I}}(\mathbf{x}_d)\|$ along the way.*

Proof. See Vasilopoulos et al. (2020b). □

5.6 Experiments

The reactive planner from Section 5.5 and semantic mapping pipeline as described in 4.3 were integrated into an architecture as overviewed in Figure 32.

In addition to the semantic mapping pipeline outlined in Section 4.3 and the reactive planner, we included the approach from Kolotouros et al. (2019) to estimate the 3D mesh of detected people in the robot's field of view as an additional mapped

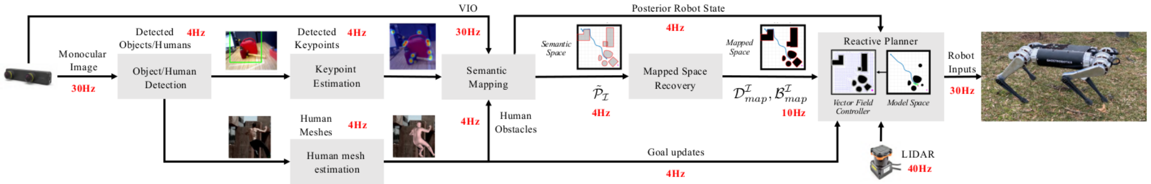

Figure 32: Online reactive planning and semantic mapping architecture (from Vasilopoulos et al. (2020a)).

semantic obstacle. The main computer used is an Nvidia Jetson AGX Xavier GPU, responsible for running both the navigation and perception algorithms. The GPU communicates with a Hokuyo LIDAR used to detect unknown obstacles, and a ZED Mini stereo camera used for both a visual-inertial odometry into the semantic SLAM pipeline and as the camera input to the object and human detectors. The mapping pipeline was implemented in the same way as described in Section 4.3, and the reactive controller was implemented in C++ using Boost Geometry (Schäling, 2014) for the underlying polygon operations and runs at 30Hz.

The reactive planning system was tested on two separate robots: the Turtlebot (TurtleBot2, 2019), and the more dynamic Ghost Spirit legged robot (Ghost Robotics, 2021). Several experiments were run using the two platforms; see Figure 33 for some representations of the environments and a visualization of the platforms used in our experiments.

The experiments used the human detection component of the architecture to set the desired location $\mathbf{x}_d$: the robot was given the task of navigating to a (moving) target while avoiding obstacles in an unknown environment. Figure 34 shows a Spirit robot following a human in a previously unexplored hallway environment, containing both catalogued obstacles localized with the semantic mapping pipeline (chairs), and unknown obstacles avoided via LiDAR.

In this and several similar environments the controller and mapping pipelines

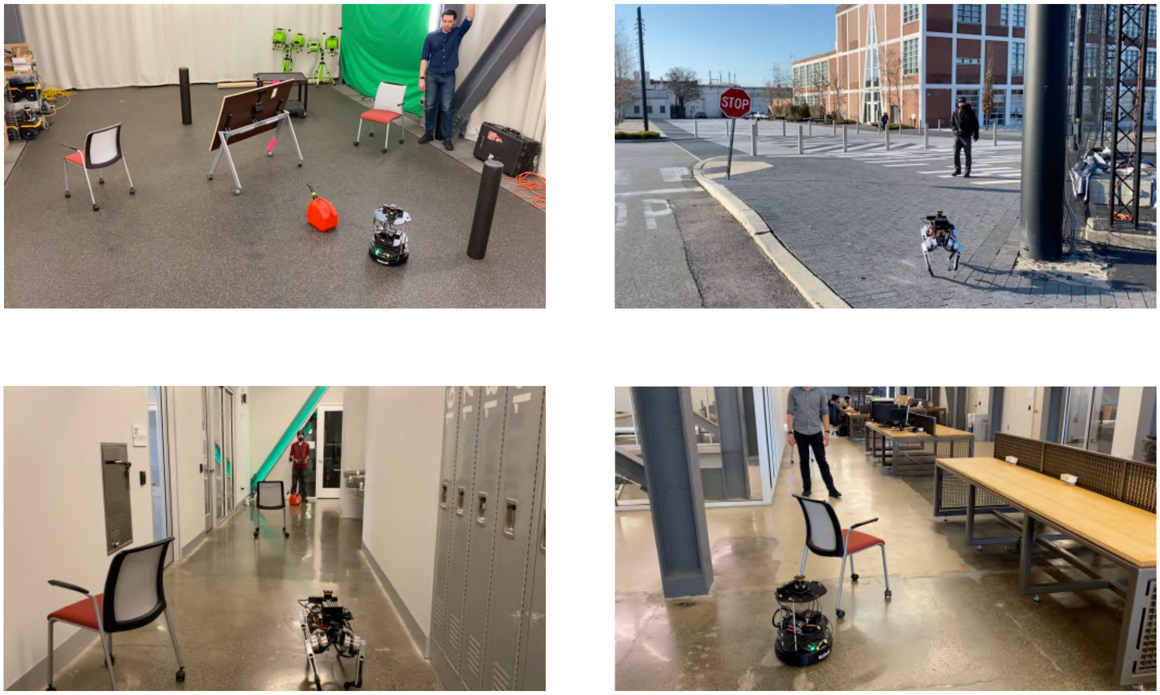

Figure 33: Types of environments and platforms used in our experiments (from Vasilopoulos et al. (2020a)).

proved robust and adaptable and were able to successfully complete the goal while avoiding detected obstacles.

A similar experiment was performed, as seen in Figure 35 in which a Turtlebot was given the task of following a detected human until a “stop” gesture (raising the hand) is detected. At that point, the Turtlebot was instructed to return to its start position. As can be seen in the trajectory history and mapped objects in the rightmost two images in Figure 35, the robot successfully followed the person to the opposite corner of the room while avoiding the mapped objects, and subsequently returned to its original position in the bottom right.

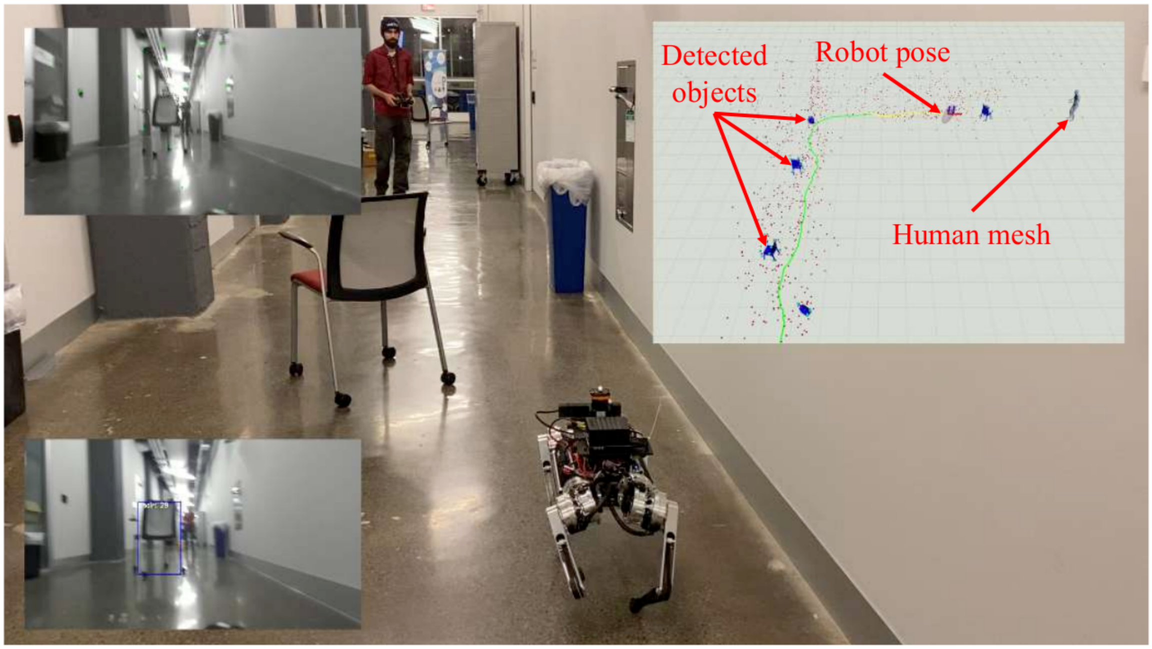

Figure 34: Ghost Spirit following a human while avoiding obstacles in a previously unexplored environment. Shown on the left are the output of the ZED camera and the object detector, and on the right the mapping system's internal representation of the world is shown, with the robot trajectory shown in green, the detected objects in blue, triangulated geometric features (cf. Section 3.2.2) in red, and the detected human mesh in grey (from Vasilopoulos et al. (2020a))

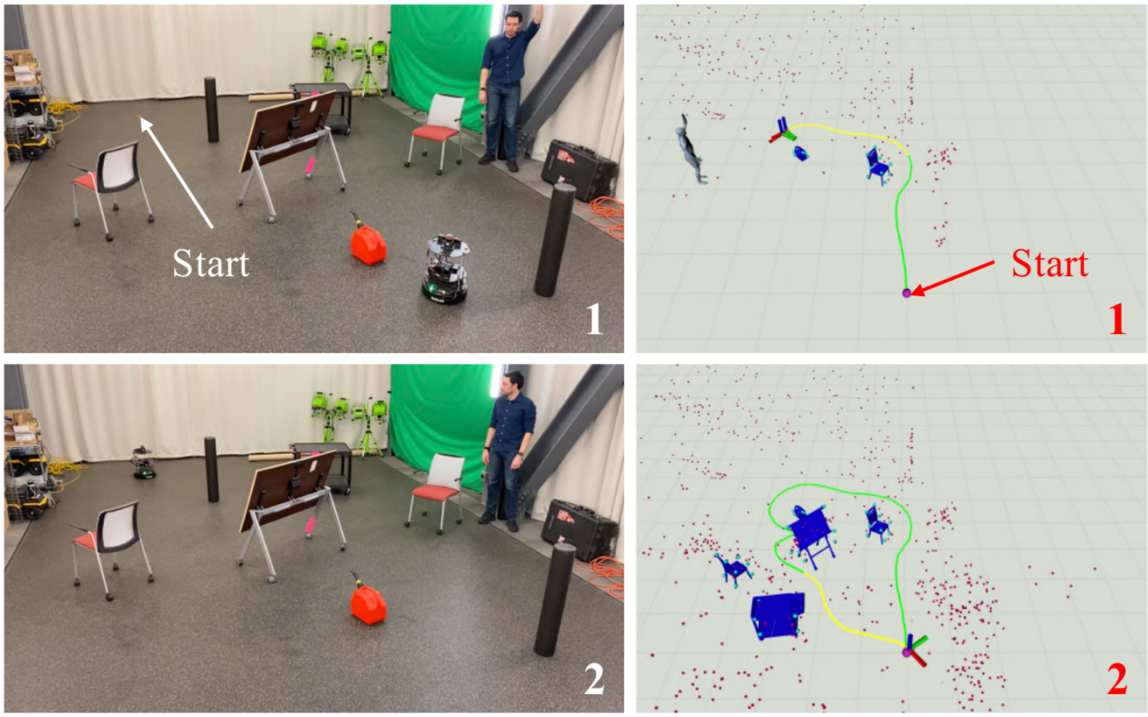

Figure 35: Top: The Turtlebot follows a human until a stop gesture is given and detected. Bottom: the Turtlebot safely returns to its starting position

Chapter 6

Conclusions

High-level autonomy in unknown environments requires advancements past typical metric SLAM; it requires robust and efficient semantic mapping system. In this thesis, we have presented two main contributions to semantic mapping and localization. The first is the methods of probabilistic data association presented in Chapter 2. By exploiting the full shape of the data association distribution rather than simply its modes, even highly ambiguous measurements such as object bounding box detections from a single image are able to be integrated into a SLAM system and used to create a robust semantic map of an unknown environment. This was shown in the direct simulations in Chapter 2, and also in Chapter 3 and the experiments within, where these methods were show to be effective in incorporating semantic information into a SLAM system in both small-scale indoor and large-scale outdoor environments.

The second primary contribution is the keypoint-based semantic SLAM system described in Chapter 4. By effectively reducing the problem of multi-object pose estimation to the much simpler and more heavily studied problem of SLAM with point features (the semantic keypoints), we presented a system that was able to both precisely localize the 6 degree-of-freedom pose of catalogued semantic objects and optimize the resulting factor graph estimation in an efficient way. The effectiveness of this was also shown in several experiments.

Finally, a high-level usage of the semantic SLAM system was given with the reactive planning algorithm described in Chapter 5. Through the use of the keypoint-based semantic SLAM algorithm, a robot is able to robustly navigate through unexplored semantic environments and perform logically complex tasks presented in a high-level way.

BIBLIOGRAPHY

Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

Omur Arslan and Daniel E Koditschek. Sensor-based reactive navigation in unknown convex sphere worlds. *The International Journal of Robotics Research*, 38(2-3): 196–223, 2019. doi: 10.1177/0278364918796267. URL https://doi.org/10.1177/0278364918796267.

N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Semantic Localization Via the Matrix Permanent. In *Robotics: Science and Systems (RSS)*, 2014.

Nikolay Atanasov, Menglong Zhu, Kostas Daniilidis, and George Pappas. Localization from semantic observations via the matrix permanent. *The International Journal of Robotics Research*, 35(1-3):73–99, 2016. doi: 10.1177/0278364915596589.

Nikolay Atanasov, Sean L. Bowman, Kostas Daniilidis, and George J. Pappas. A unifying view of geometry, semantics, and data association in slam. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5204–5208. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/722. URL https://doi.org/10.24963/ijcai.2018/722.

T. Bailey, J. Nieto, and E. Nebot. Consistency of the fastslam algorithm. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 424–429, 2006. doi: 10.1109/ROBOT.2006.1641748.

Tim Bailey. SLAM Simulations. http://www-personal.acfr.usyd.edu.au/tbailey/software/slam_simulations.htm, 2021.

S. Bao and S. Savarese. Semantic Structure from Motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2025–2032, 2011.

Ioan Andrei Barsan, Peidong Liu, Marc Pollefeys, and Andreas Geiger. Robust dense mapping for large-scale dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7510–7517, 2018. doi: 10.1109/ICRA.2018.8462974.

M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 298–304, 2015. doi: 10.1109/IROS.2015.7353389.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J. Pappas. Probabilistic data association for semantic slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017. doi: 10.1109/ICRA.2017.7989203.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

J. Civera, D. Galvez-Lopez, L. Riazuelo, J. Tardos, and J. Montiel. Towards Semantic SLAM Using a Monocular Camera. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1277–1284, 2011.

Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sept 2012. URL `https://research.cc.gatech.edu/borg/sites/edu.borg/files/downloads/gtsam.pdf`.

Charles Dubout and François Fleuret. Deformable part models with individual part scaling. In *British Machine Vision Conference*, number EPFL-CONF-192393, 2013.

H. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE*, 13(2):99–110, June 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1638022.

Xiaohan Fei and Stefano Soatto. Visual-inertial object detection and mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010.

C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15–22, 2014. doi: 10.1109/ICRA.2014.6906584.

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical Semantic Maps for Mobile Robotics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2278–2283, 2005.

Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3d Reconstruction in Real-time. In *Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Ghost Robotics. Spirit 40, 8 2021. URL `http://ghostrobotics.io`.

Margarita Grinvald, Fadri Furrer, Tonci Novkovic, Jen Jen Chung, Cesar Cadena, Roland Siegwart, and Juan Nieto. Volumetric instance-aware semantic mapping and 3d object discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, 2019. doi: 10.1109/LRA.2019.2923960.

Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research (IJRR)*, 31(5): 647–663, 2012.

Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency Analysis and Improvement of Vision-aided Inertial Navigation. *IEEE Trans. on Robotics (TRO)*, 30(1):158–176, 2014.

B. Deniz Ilhan, Aaron M. Johnson, and D. E. Koditschek. Autonomous stairwell ascent. *Robotica*, 38(1):159–170, 2020. doi: 10.1017/S0263574719000535.

Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51 (4):671–697, July 2004. ISSN 0004-5411. doi: 10.1145/1008731.1008738. URL `http://doi.acm.org/10.1145/1008731.1008738`.

Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. *The International Journal of Robotics Research (IJRR)*, 31(2):216–235, 2012.

Nikos Kolotouros, Georgios Pavlakos, Michael Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. pages 2252–2261, 10 2019. doi: 10.1109/ICCV.2019.00234.

D. G. Kottas, K. Wu, and S. I. Roumeliotis. Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3172–3179, Nov 2013. doi: 10.1109/IROS.2013.6696807.

Dimitrios G. Kottas and Stergios I. Roumeliotis. Efficient and Consistent Vision-aided Inertial Navigation using Line Observations. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1540–1547, 2013.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A General Framework for Graph Optimization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and JamesM. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8694 of *Lecture Notes in Computer Science*, pages 703–718. Springer International Publishing, 2014. ISBN 978-3-319-10598-7. doi: 10.1007/978-3-319-10599-4_45. URL http://dx.doi.org/10.1007/978-3-319-10599-4_45.

Wai Jing Law. *Approximately Counting Perfect and General Matchings in Bipartite and General Graphs*. PhD thesis, Duke University, 2009.

B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383146.

J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, pages 1442–1447 vol.3, 1991. doi: 10.1109/IROS.1991.174711.

F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Auton. Robots*, 4(4):333–349, 1997.

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635, 2017. doi: 10.1109/ICRA.2017.7989538.

Anastasios Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3565–3572. IEEE, 2007.

James Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial & Applied Mathematics (SIAM)*, 5(1):32–38, 1957.

Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *CoRR*, abs/1610.06475, 2016. URL http://arxiv.org/abs/1610.06475.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5): 1147–1163, 2015. doi: 10.1109/TRO.2015.2463671.

José Neira and Juan Tardós. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *IEEE Trans. on Robotics and Automation (TRO)*, 17(6): 890–897, 2001.

Lachlan Nicholson, Michael Milford, and Niko Sunderhauf. Quadricslam: Dual quadrics as slam landmarks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Janosch Nikolic, Joern Rehder, Michael Burri, Pascal Gohl, Stefan Leutenegger, Paul T Furgale, and Roland Siegwart. A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 431–437. IEEE, 2014.

Iram Noreen, Amna Khan, and Zulfiqar Habib. Optimal path planning using rrt* based approaches: A survey and future directions. *International Journal of Advanced Computer Science and Applications*, 7, 11 2016. doi: 10.14569/IJACSA.2016.071114.

Santiago Paternain, Daniel E. Koditschek, and Alejandro Ribeiro. Navigation functions for convex potentials in a space with convex obstacles. *IEEE Transactions on Automatic Control*, 63(9):2944–2959, 2018. doi: 10.1109/TAC.2017.2775046.

G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2011–2018, 2017.

Sudeep Pillai and John Leonard. Monocular slam supported object recognition. In *Proceedings of Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

Andrzej Pronobis. *Semantic Mapping with Mobile Robots*. dissertation, KTH Royal Institute of Technology, 2011.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992. doi: 10.1109/70.163777.

Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020. doi: 10.1109/ICRA40945.2020.9196885.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Int. Conf. on Computer Vision*, pages 2564–2571, 2011. doi: 10.1109/ICCV.2011.6126544.

Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478, 2017. doi: 10.1109/ICRA.2017.7989518.

R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013.

Boris Schäling. *The boost C++ libraries.* XML Press, 2014. ISBN 9781937434366 1937434362.

Mo Shan, Qiaojun Feng, and Nikolay Atanasov. Orcvio: Object residual constrained visual-inertial odometry. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5104–5111, 2020. doi: 10.1109/IROS45743.2020.9341660.

Randall C. Smith and Peter Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986. doi: 10.1177/027836498600500404.

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16519–16529, June 2021.

Jörg Stückler, Benedikt Waldvogel, Hannes Schulz, and Sven Behnke. Dense real-time mapping of object-class semantics from RGB-D video. *Journal of Real-Time Image Processing*, pages 1–11, 2013.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-44480-0.

TurtleBot2. Open-source robot development kit for apps on wheels, 2019. URL `https://www.turtlebot.com/turtlebot2`.

Vasileios Vasilopoulos, Georgios Pavlakos, Sean Bowman, J. Caporale, Kostas Daniilidis, George Pappas, and Daniel Koditschek. Reactive semantic planning in unexplored semantic environments using deep perceptual feedback. *IEEE Robotics and Automation Letters*, PP:1–1, 06 2020a. doi: 10.1109/LRA.2020.3001496.

Vasileios Vasilopoulos, Georgios Pavlakos, Sean L. Bowman, J. Diego Caporale, Kostas Daniilidis, George J. Pappas, and Daniel E. Koditschek. Technical report: Reactive semantic planning in unexplored semantic environments using deep perceptual feedback, 2020b.

Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A. Prisacariu, Olaf Kähler, David W. Murray, Shahram Izadi, Patrick Perez, and Philip H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. doi: 10.1109/TRO.2019.2909168.

Liang Zhang, Leqi Wei, Peiyi Shen, Wei Wei, Guangming Zhu, and Juan Song. Semantic slam based on object detection and improved octomap. *IEEE Access*, 6: 75545–75559, 2018. doi: 10.1109/ACCESS.2018.2873617.

Lintao Zheng, Chenyang Zhu, Jiazhao Zhang, Hang Zhao, Hui Huang, Matthias Niessner, and Kai Xu. Active scene understanding via online semantic reconstruction. *Computer Graphics Forum*, 38(7):103–114, 2019. doi: https://doi.org/10.1111/cgf.13820. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13820`.

Menglong Zhu, Nikolay Atanasov, George Pappas, and Kostas Daniilidis. Active Deformable Part Models Inference. In *European Conference on Computer Vision (ECCV)*, volume 8695 of *Lecture Notes in Computer Science*, pages 281–296. Springer, 2014.