Semantic Perception under Uncertainty with Viewpoint-Dependent Models

Yuri Feldman

Semantic Perception under Uncertainty with Viewpoint-Dependent Models

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Yuri Feldman

Submitted to the Senate
of the Technion — Israel Institute of Technology
Tammuz 5782 Haifa July 2022

This research was carried out under the supervision of Associate Professor Vadim Indelman, in the Faculty of Computer Science.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being:

Journal Articles

[1] Y. Feldman and V. Indelman. "Spatially-Dependent Bayesian Semantic Perception under Model and Localization Uncertainty". In: *Autonomous Robots* (2020)

In Conference Proceedings

[1] Y. Feldman and V. Indelman. "Bayesian Viewpoint-Dependent Robust Classification under Model and Localization Uncertainty". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018

[2] V. Tchuiev, Y. Feldman, and V. Indelman. "Data Association Aware Semantic Mapping and Localization via a Viewpoint-Dependent Classifier Model". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019

The author's part in Tchuiev, Feldman, and Indelman 2019 was (equal) co-development of the theory and co-editing the paper. The experimental implementation and evaluation and results were completely done by Vladimir Tchuiev.

In Professional Workshops

[1] Y. Feldman and V. Indelman. "Towards Self-Supervised Semantic Representation with a Viewpoint-Dependent Observation Model". In: *Workshop on Self-Supervised Robot Learning, in conjunction with Robotics: Science and Systems (RSS)*. July 2020

[2] Y. Feldman and V. Indelman. "Towards Robust Autonomous Semantic Perception". In: *Workshop on Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding, in conjunction with IEEE International Conference on Robotics and Automation (ICRA)*. May 2018

Acknowledgements

I would like to thank my supervisor Prof. Vadim Indelman for the belief in me, endless patience, continued encouragement and guidance throughout this research. Thanks to Vladimir Tchuiev for the collaboration on our IROS paper and fruitful discussions. Thanks to Ariel Dobrovenski and Anton Gulyaev for our work together and the entire ANPL team for the useful feedback at group meetings and pleasant and supportive company throughout.

The generous financial help of the Technion and partial funding of the research from the Israeli Smart Transportation Research Center are gratefully acknowledged.

Contents

List of Figures

Abstract

The term Robotic Perception refers to acquisition of sensor measurements and their processing to produce a representation of a robot's (more generally – autonomous agent's) state and environment. To operate reliably, an autonomous robot needs to be resilient to noisy and partial sensor measurements, actuation errors, and data association ambiguity, i.e. an autonomous robot is bound to operate under uncertainty. Semantic Perception means constructing a representation that is beyond geometric, e.g. capturing categories of environment elements such as places and objects. Perception of semantics is required for autonomous robots to be able to perform a wide range of tasks in less structured environments and alongside humans. However, semantic measurements violate assumptions commonly made by perception systems: they are viewpoint dependent and spatially correlated. Additionally, as semantic measurements are commonly obtained from deep learning models they are also affected by model uncertainty – uncertainty in a learned model's output induced by departure from the training set. This thesis explores approaches to semantic perception formulated as semantic SLAM on the level of objects. We utilize viewpoint-dependent models to capture the spatial variation in a semantic measurement for an object of a given class and show how these can be used in perception under uncertainty for mutual disambiguation of geometry and semantics, facilitating data association and for inference in an induced latent continuous semantic representation space. We start by developing an approach for object classification by a moving robot under localization uncertainty, while accounting for model uncertainty and correlations among viewpoints. We then address object SLAM with unknown data-association using viewpoint-depedent class models for disambiguation, in particular addressing perceptual aliasing. Finally, we propose a novel formulation for object SLAM through inference of object semantic representation vectors over a latent representation space induced by a viewpoint-dependent model learned with weak ground-truth requirements.

Notation and Abbreviations

β_k	Data association at time-step k
$\mathcal{S}_k$	A set of classifier outputs
u_k	Control action taken from pose x_k
$\mathcal{U}_{k:l}$	Set of control actions taken between time-steps k and l
e_i	A latent semantic representation of the object
$\mathbb{N}^n$	The set of vectors of dimension n with positive integer elements.
$\mathcal{Z}_{k:l}$	Set of observations taken between time-steps k and l
$\mathcal{X}_{k:l}$	Sequence of poses between time-steps k and l
$\mathcal{X}_{k:l}^{(rel)}$	Sequence of relative poses between time-steps k and l
c	An object class
f_ψ	A feature extractor
s	Classification output vector
$x^{(rel)} \doteq x \ominus o$	Robot's relative pose with respect to object o
x_k	Robot's pose at time-step k
z_k	Observation obtained at time step k
$\mathcal{H}_k \doteq \{\mathcal{U}_{0:k-1}, \mathcal{Z}_{0:k}\}$	History at time k comprised of observations $\mathcal{Z}_{0:k}$ and user controls $\mathcal{U}_{0:k-1}$
$\mathcal{H}_k^- \doteq \mathcal{H}_k \setminus \{z_k\}$	History at time k before observation k was obtained.
$b[c_k] \doteq \mathbb{P}(c_k \mid \mathcal{H}_k)$	Posterior probability of the object to belong to class c given all measurments and user controls up to time k
$\mathcal{C}$	Set of object classes
$\mathcal{E}$	A set of per-object latent semantic representations
$\mathcal{L}$	Geometric landmarks
$\mathcal{O}$	Set of object poses
o	Object pose
$\mathcal{D}$	Training set
CNN	Convolutional Neural Network
DA	Data Association
DL	Deep Learning
ELBO	Evidence Lower Bound

EM	Expectation Maximization or Minimization, subject to context
GP	Gaussian Process
i.i.d.	Independent Identically Distributed
MCMC	Markov Chain Monte Carlo
MDP	Markov Decision Process
MGR	Most likely to ground truth ratio
MLP	Multi Layer Perceptron
MSDE	Mean squared detection error
POMDP	Partially Observable Markov Decision Process
SfM	Structure from Motion
VAE	Variational Auto-Encoder

Chapter 1

Introduction

1.1 Motivation

Perception under Uncertainty

The term *Robotic Perception* refers to acquisition of sensor measurements and their processing to produce a representation of a robot's state and environment. It is a fundamental constituent in allowing mobile robots to operate autonomously, i.e. without a human operator. An autonomously operating robot uses the constructed state and environment representations for decision making - choosing its next actions w.r.t. its task . As sensor measurements are noisy and partial (i.e. they only carry *partial information* about the state of the robot and its environment), an autonomous robot is compelled to operate under perpetual *uncertainty* w.r.t. to its state, which characterizes the distribution of estimate error as a result of the partial information in sensor measurements. This uncertainty must be maintained and accounted for in decision making in order for the robot to operate safely. Another type of uncertainty that arises in autonomous operation is uncertainty in *data association* - the association of robot sensor measurements to elements in the environment producing them. Incorrect association acts as an outlier in the state estimation and can easily have catastrophic effects on the estimates. A particularly challenging case of data association uncertainty arises as the result of *perceptual aliasing* - whereby different parts of the environment produce the same observations, meaning that data association cannot be correctly resolved based on obtained measurements. To safely address perceptual aliasing, uncertainty in data association needs to be maintained until disambiguating measurements are obtained. This is commonly done with mixture models, leading to challenging mixed - continuous and discrete estimation problems.

Following progress in hardware capabilities and algorithmic approaches (Cadena et al. [11] and Garg et al. [38]), autonomous robots are increasingly present in a growing multitude of domains: from the now-widespread indoors floor-cleaning robots through autonomous mowers and pool cleaners, to underwater and flying inspection drones, to

underground rescue and to space applications. The fundamental task of perception and the set of tools to address it remain highly overlapping across domains. The set of established methods for estimating and maintaining the geometric structure of the robot's environment and its state are known as "Simultaneous Localization and Mapping" - SLAM.

Semantic Perception

While for some applications a purely geometric understanding of the environment can be fully sufficient, robot autonomy in other tasks, which are less structured, require of the robot a more refined understanding of its environment - beyond purely geometry (Garg et al. [38]). For example, a robot navigating an urban environment might need to distinguish static environment features from various road users, traffic signs as well as some elements of the environment that would help it to position itself globally. An agricultural robot might need to distinguish and navigate among crops and a home assistant robot among the different parts of a house, its inhabitants, and the various objects found in it. Perception that goes beyond geometry is called Perception of Semantics, or *Semantic Perception*. Semantics establish a language common to humans in terms of which robot tasks can be specified and constraints established (Kostavelis and Gasteratos [62]). They also carry multiple benefits for autonomously operating robots - providing strong position disambiguation cues e.g. for loop closure detection, and for aiding data association, in particular - resolving perceptual aliasing as will be discussed later on, as well as allowing a sparse representation of the environment, which can be efficiently maintained and communicated, e.g. across a team of robots (Choudhary et al. [17]).

Semantic Measurements

The latest renaissance of deep learning (LeCun, Bengio, and Hinton [70] and Kosman and Di Castro [61]) has provided fairly reliable methods to extract rich semantic information from raw measurements of the environment, such as images of different modalities, audio and video, written text, Lidar and Radar point clouds, and many more - paving the way for Robotic Semantic Perception. Deep learning methods are based on variants of multi-layer Perceptron (MLP) models (shown to be universal approximators by Hornik [44]) with parameters fit to produce desired output - also called *Deep Learning Models* (or *Deep Models*). The per-timestep semantic information thus extracted can be referred to as *semantic measurements*. The availability of semantic information has spurred research in the robotics domain into how it can be incorporated in perception.

One of the key issues (Sünderhauf et al. [120]) that arise when semantic measurements are to be utilized in the context of robotics is that of *domain-*, or *covariate-shift*. Deep learning models are generally fit, or trained, on a finite set of data ("training

set"), with their performance verified on a "validation" and "test" sets, which are generally assumed to follow the same probabilistic distribution as the training set. The validity of the outputs of such models for inputs outside of the training distribution is not guaranteed, and worse - their behavior is known to be arbitrary and unstable. This may not pose a significant problem for common applications in the field of e.g. computer vision (which was originally the focus of modern Deep Learning research Krizhevsky, Sutskever, and Hinton [63] and LeCun et al. [69]), where the input domain can be assumed to be well-represented by the training data and the cost of a single algorithm error is not high or can be mitigated. This however is a major issue for robotics applications, where a robot needs to function safely across all possible application environments, which may differ significantly from those covered by training data (hence, "domain shift"), and where a perception error may lead to incorrect physical actions executed by the robot, possibly entailing serious consequences.

In a general sense, semantic measurements behave differently from geometric measurements which are classically used (and which they complement). A ubiquitous fundamental assumption in geometric observation models is that of measurement independence: measurements taken at different time steps are assumed to be statistically independent given robot and environment state. This assumption is readily broken by semantic measurements, as e.g. multiple successive observations of the same static object from the very same or spatially close, viewpoints add little new semantic information. Treating such measurements as independent amounts to accounting multiple times for the same identical measurement, which leads to an over-confident state estimate, which may be correct, or incorrect - either way not reflecting the true uncertainty. A further particularity of semantic measurements is viewpoint-dependence: empirically, as e.g. an object is viewed from multiple viewpoints, the output of a Deep Model fluctuates as the object appearance and background changes with viewpoint. For an object detector and when viewing an object from uncommon viewpoints (that are likely not represented in the training set) the detection may be completely misclassified (or even missed altogether).

Another major challenge associated with semantics stems from that their discrete nature leads to combinatorial complexity of representation. Humans associate elements of the environment with a finite set of words describing them in terms of predominantly discrete properties: a discrete category e.g. a tree, a chair, a lamppost, a discrete color and characterization - e.g. yellow, blue, tall or wide, etc. A single discrete property translates in the estimation problem into a set of hypotheses associated with the corresponding element of the environment, the maximum number of hypotheses being the number of possible values of the property in the worst case. More than one discrete property results in a combinatorial number of hypotheses, amounting to a severe problem of representation for semantics.

Finally, related to covariate shift but a severe limitation in its own right is the scarcity of tagged, or "groundtruth" training data, coupled with robot state uncer-

tainty (or *Partial Observability*). Training Deep Learning Models requires the expected outputs (i.e. *groundtruth*) to be specified (possibly, implicitly) for the training (as well as validation and possibly also test) data. While generally obtaining groundtruth is more often than not difficult and costly, in the robotics domain the challenge is greatly amplified by two additional factors: one is the richness of information required to allow the robot to adequately interpret its environment, two - is the fact that the state of the agent acquiring the training data (often a robot itself) is usually uncertain, due to the factors outlined above. Thus, even providing adequate expected outputs does not automatically guarantee the ability to obtain a sound Deep Learning Model to solve a task.

1.2 Overview

The present thesis can be categorized as part of the effort towards Semantic Perception for autonomous robots. Focusing on Semantic SLAM, i.e. localization and mapping on the level of semantics, it deals explicitly with the properties of semantic measurements outlined above, addressing the associated challenges.

Viewpoint-Dependent Models

A key observation is that while the output of a semantic feature detector (e.g. Deep Learning Model) fluctuates with viewpoint w.r.t. a semantic element of the environment (e.g., an object), it does so in a *predictable* way, depending on the viewpoint up to noise. Semantic Measurements can be used to construct a statistical model capturing this viewpoint-dependent response, *a viewpoint-dependent semantic measurement model*. A model constructed in this way relates the robot viewpoint relative to the object when capturing a measurement, to the semantic measurement obtained, probabilistically relating geometry and semantics and thus providing for their mutual disambiguation.

Contributions

This thesis explores approaches for semantic perception under uncertainty utilizing viewpoint-dependent models to address challenges associated with semantic measurements in the context of object-based localization and mapping (Object SLAM Salas-Moreno et al. [109]). In a first direction semantic observation models are constructed capturing both viewpoint dependence and inter-viewpoint correlations of semantic measurements. The observation models are subsequently used for object classification under model and localization uncertainty. In a second direction, viewpoint-dependent semantic measurement models are employed to enhance disambiguation in the context of object-based localization and mapping (with multiple objects). A third direction

proposes a novel semantic environment representation with a per-object continuous semantic description implicitly obtained by constructing a viewpoint-dependent model. Specifically, the approaches to semantic perception contributed in this thesis are:

1. An approach for spatially-dependent object classification under model and localization uncertainty, capturing viewpoint dependence of semantic measurements as well as inter-viewpoint correlations.

2. An approach for data association-aware semantic mapping and localization via a viewpoint-dependent classifier model.

3. An approach to semantic mapping and localization via continuous inference in a learned semantic space with a viewpoint-dependent measurement model.

Thesis Structure

The remainder of the thesis is organized as follows. In Chapter 2 we survey literature related to the thesis topics. Chapter 3 through Chapter 5 describe the contributed approaches: Chapter 3 explores object classification under model and localization uncertainty utilising viewpoint dependent models. Chapter 4 describes an approach to semantic mapping and localization while explicitly addressing data association and using viewpoint-dependent models to aid disambiguation. Chapter 5 describes a novel formulation of semantic mapping and localization as a fully continuous inference problem, where inference of semantics occurs in a latent space induced by a learned viewpoint-dependent model. We conclude in Chapter 6 by reviewing the thesis contributions and main findings, and outlining possible future research directions.

Chapter 2

Literature Review

2.1 Geometric Perception: SLAM

We start by briefly reviewing the SLAM techniques which the present work expands on towards perception of semantics. The foundation of the currently prevalent approaches to SLAM has been largely developed in the late 2000's, with the standard formulation being based on probabilistic joint maximum-a-posteriori inference over variables of interest (Cadena et al. [11]) - describing robot and environment state - a smoothing, rather than filtering approach (Strasdat, Montiel, and Davison [116]), first proposed about a decade before it took strong hold (Lu and Milios [72], Gutmann and Konolige [41], and Dellaert and Kaess [22]). Two key elements in modern SLAM approaches are exploitation (and maintenance) of the sparsity of the square root information matrix, in an incremental way (Kaess, Ranganathan, and Dellaert [51], Kaess et al. [49], and Polok et al. [101]), to continuously update a MAP estimate in real time. A commonly used formalism to describe a SLAM problem is a *factor graph* (Kschischang, Frey, and Loeliger [64]). A factor graph is a bipartite undirected graph, with a node for each variable and for each probabilistic factor in the joint posterior decomposition, and edges connecting variable nodes with factors they participate in. In the context of visual SLAM, i.e. SLAM operating on RGB images, a distinction can be mentioned between direct SLAM methods and indirect or *feature-based* SLAM methods. Direct SLAM approaches (e.g. Newcombe, Lovegrove, and Davison [89], Engel, Schöps, and Cremers [26], and Engel, Koltun, and Cremers [25]) optimize per-pixel photometric error between frames as function of the estimated poses and scene geometry, while feature-based approaches (e.g. Klein and Murray [58], Campos et al. [12], and Rosinol et al. [107]) pre-process frames by extracting features which are assigned to variables (*landmarks*) in the estimation problem. The approach taken in this thesis is closer to the latter, as objects (semantic features) in the environment are treated as landmarks.

2.2 Semantic SLAM, Object-Centric Perception

We briefly review the context of this research in the field of semantic SLAM. Comprehensive surveys which cover more diverse aspects are provided in Kostavelis and Gasteratos [62], Sünderhauf et al. [121], Cadena et al. [11], and Sünderhauf et al. [120].

While focusing on geometric perception, early work proposed to complement metric information with a hierarchical qualitative map (Kuipers and Byun [66], Kuipers [65], and Kortenkamp [60]), sometimes called "cognitive map", which was backed by psychology research of human spatial perception (Tolman [127]). At least as early as mid 2000's work explicitly introduced semantics into the representation hierarchy (Tapus, Tomatis, and Siegwart [123] and Galindo et al. [37]), in particular in the form of object positions and labels (as well as place labels, directly extending past cognitive map approaches), citing the necessity of semantic map information for human-machine interaction and efficiency of representation (Vasudevan et al. [128] and Ranganathan and Dellaert [103]). Early methods were severely limited by object detection performance. Ranganathan and Dellaert [103] proposed an inference formulation joint with object detection based on image cues and learned models, which they perform using MCMC (Barbu and Zhu [7]). Later work continued to develop the object-centric approach. Bao et al. [6] formulated a joint energy-based SfM with points (landmarks), objects, and (planar) regions, which they solved through global optimization - simulated annealing (Kirkpatrick, Gelatt Jr, and Vecchi [57]). Choudhary et al. [16] formulated SLAM with objects segmented as part of inference based on the assumption of objects being supported by a planar surface. Salas-Moreno et al. [110] implemented what was probably the first large-scale capable, real-time 3D SLAM system using a spatial representation based on objects (as opposed to augmenting a geometric map with detected objects). They detected objects based on depth measurements using generalized Hough transform-based method (Ballard [5] and Drost et al. [24]), requiring CAD models for all mapped objects to operate. Object detections were used as measurements to refine estimates, but without notion of detection or classification uncertainty, which was not part of the inference. With the introduction of deep object detectors (Girshick et al. [39], Redmon et al. [105], and Liu et al. [71]) they were incorporated in semantic inference as virtual sensors providing a measure of classification uncertainty. Omidshafiei et al. [92] and Mu et al. [87] formulated hierarchical inference of object SLAM jointly with class model parameters. Bowman et al. [10] implemented real-time inference mapping and performing loop closures with objects while also using point landmarks for visual odometry. Sünderhauf et al. [122] and McCormac et al. [80] proposed SLAM systems which simultaneously compute 3D models for discovered objects (but only McCormac et al. [80] incorporated the objects in probabilistic inference). Nicholson, Milford, and Sünderhauf [90] computed bounding quadrics to represent detected objects in inference (as opposed to representing an object with only centroid location). A more recent parallel line of research aims to reconstruct a dense map (e.g. mesh) augmented with

semantics (categories of map elements), as opposed to a sparse map of objects inside a geometric representation - e.g. occupancy grid or pointcloud. McCormac et al. [81] generate and fuse semantic labels for a dense map constructed by the method of Whelan et al. [130]. They motivate dense mapping by being useful for robot control, e.g. collision detection. However, a dense representation is arguably less efficient and difficult to use for planning. Zhi et al. [136] construct a dense semantic map by estimating then fusing latent per-timestep "scene codes" using a learned decoder as observation model. Rosinol et al. [107] and Hughes, Chang, and Carlone [46] present a system with a hierarchical representation based on dense semantic SLAM - combining advantages of dense and sparse mapping. They construct a metric-semantic mesh in real time then process it to extract entities on multiple hierarchical levels (objects and agents, places, rooms, buildings). Sucar et al. [117] perform semantic SLAM by jointly localizing and learning an implicit scene representation inspired by "Neural Radiance Fields" Mildenhall et al. [83].

2.3 Fusion of Semantic Measurements

One simple common form of semantic measurements is (for example, object) classification obtained at the output of a classifier unit. Such classification can take the form of (in order of increasing richness) the most likely class identity (equivalent to a "1-hot" vector, i.e. vector with all elements zero but for one 1 corresponding to the index of the class in question), the most likely class identity along with detection score, which can usually be interpreted as probability, or an entire classification vector, specifying a categorical distribution over classes of interest. Most likely class output may be unstable. As an example at the time of this writing state of the art top-1 (most likely class) accuracy on the ImageNet dataset (Russakovsky et al. [108]) is 91% (Yu et al. [135]). Accuracy for a model deployed on a robot (and hence subject to runtime and power limitations), acting on measurements obtained in diverse environments can be expected to be lower. The possible implication of a single mis-classification to methods that perform sparse mapping with objects is a failed or wrong data association in a loop closure and as a result, complete corruption of the estimate, explaining why perception methods commonly filter semantic measurements.

Perception methods performing fusion of classifier measurements can be roughly split into methods directly fusing classifier scores (Patten et al. [96], Pillai and Leonard [99], and Omidshafiei et al. [92]), and methods matching classifier measurements to a statistical model (Bowman et al. [10], Patten, Martens, and Fitch [97], Teacy et al. [126], Atanasov et al. [2], Omidshafiei et al. [92], Velez et al. [129], Becerra et al. [8], Tchuiev and Indelman [125], and Mu et al. [87]). Patten et al. [96] and Pillai and Leonard [99] use a classifier unit producing a distribution over classes, i.e. a categorical vector describing class likelihood given an input measurement, fusing measurements by directly applying Bayes rule. Bowman et al. [10] use most-likely-class measurements with the classifier

confusion matrix as the measurement model. In a joint hierarchical Bayesian inference formulation, Omidshafiei et al. [92] use a Dirichlet class model with categorical vector measurements. In a sense this method belongs to both groups, as only a single noise parameter is inferred per-class, amounting to an assumption that noise-free classifier output is a 1-hot-vector denoting class. Mu et al. [87] use a Dirichlet class model with a general vector noise parameter with a categorical measurement model (using only the identity of the most - likely class output by the classifier), likewise performing a joint hierarchical inference of states and model parameters. Tchuiev and Indelman [125] also use a Dirichlet class model with a general noise parameter, incrementally maintaining a Dirichlet classification posterior to quantify model uncertainty.

All methods described above use classifier measurement models that are not explicitly parametrized by viewpoint. While this makes class inference trivially agnostic to pose estimation errors, downsides include overly-pessimistic noise model, as variations in measurements due to viewpoint changes are interpreted as noise, and more importantly, over-confident inference due to not accounting for the reduced information value of spatially correlated measurements. It is therefore not surprising that active perception methods utilizing semantic measurements tend to use viewpoint-dependent models, which allow to better quantify information value of future views for planning. Atanasov et al. [2] use a measurement model based on VP-Tree object detector (Atanasov et al. [2]), parameterized by object class and orientation relative to camera, to perform non-myopic planning for inference of object class and orientation. Closely related to our method Velez et al. [129] and Teacy et al. [126] use Gaussian Processes (GPs) to model detector responses for objects of a given class, capturing spatial correlations by learning parameters of the GP kernels. Patten, Martens, and Fitch [97] also use per-class GP observation models of point-cloud features. Aydemir, Bishop, and Jensfelt [3] explored inference with viewpoint-dependent models with a general classifier output vector. However, the above assume robot localization is known, which in reality is a simplification. Becerra et al. [8] relax the assumption of known localization, however, discretize pose space both for localizing the robot and in the viewpoint dependent class model representation.

2.4 Model Uncertainty

Model uncertainty is uncertainty in model output that is due to the covariate shift, or difference between training and test distribution (Gal and Ghahramani [35] and Kendall and Gal [53]). As modern classifiers are ubiquitously based on Machine Learning and robots may be deployed in environments different from the ones the classifier had been trained on, the issue of covariate shift needs to be addressed. Several methods have been proposed for efficiently obtaining the (approximate) predictive uncertainty in the output of Deep Neural Networks associated with covariate shift, including MC-Dropout (Gal and Ghahramani [35]), Bootstrapping (Osband et al. [93]), ensembles (Lakshmi-

narayanan, Pritzel, and Blundell [68]) and, more recently, Prior Networks (Malinin and Gales [75]) and recently Harakeh, Smart, and Waslander [43] and Harakeh [42] explore obtaining the covariate shift-induced predictive uncertainty for object detectors. While these are used, among else, for efficient exploration and active learning (Osband et al. [93] and Gal, Islam, and Ghahramani [36]), safe decision making (Malinin et al. [76], Lütjens, Everett, and How [73], and McAllister et al. [79]), semantic segmentation (Kendall, Badrinarayanan, and Cipolla [52]), camera localization (Kendall, Grimes, and Cipolla [54]), and the more closely related methods of Miller et al. [85, 84] incorporate model uncertainty into merging strategy of object bounding box detections in a single frame, we are not aware of existing approaches for fusion of classification measurements over multiple frames/viewpoints attempting to account for covariate shift, except for the authors work (Tchuiev and Indelman [125], Feldman and Indelman [29], and Tchuiev, Feldman, and Indelman [124]). Tchuiev and Indelman [125] fuse class measurements carrying model uncertainty information represented with a Dirichlet distribution, however without considering localization inference or spatial correlation among measurements. Tchuiev, Feldman, and Indelman [124] perform classification as part of hybrid inference under localization uncertainty, however not addressing correlations among viewpoints or model uncertainty.

2.5 Data Association and Mixed Inference

SLAM systems can commonly be split into a *front end*, preprocessing arriving measurements and adapting them to the internal representation, and a *back end*, which performs inference, fusing all available measurement information to produce estimates. Data association (DA) is commonly handled by the front end, avoiding the mixed inference problem which arises when modeling data association uncertainty in inference. In such methods if the backend assumes that data association is solved, errors in data association introduced by the front end usually would lead to estimate corruption (Lajoie et al. [67]). In such methods the backend sometimes attempts to discover and reject wrong associations (*outlier rejection*) base on consistency checks, e.g. Mangelson et al. [77] and references therein (done e.g. in the recent Kimera and Hydra semantic mapping libraries presented in Rosinol et al. [106] and Hughes, Chang, and Carlone [46]).

Perceptual aliasing is a case of data association uncertainty which cannot be solved by the front end, since it requires additional measurements, which can only be incorporated through inference, i.e. backend. Perceptual aliasing is a common case in an environment with repetitive elements, such as are especially many artificial environments (Pathak, Thomas, and Indelman [95]), motivating research into how to address data association as part of inference.

An early work on DA is joint probability data association, JPDA (Fortmann, Bar-Shalom, and Scheffe [33]) which considers all possible DA hypotheses without prun-

ing (however in the context of multi-target tracking), therefore being computationally slow. Wong, Kaelbling, and Lozano-Pérez [131] used a Dirichlet Process Mixture Model (DPMM) for data association for a partially observed environment. Sünderhauf and Protzel [119] proposed an approach to detect faulty loop closures that lead to erroneous data association in back-end optimization. Olson and Agarwal [91] proposed a robust approach that uses max-mixture models. Carlone, Censi, and Dellaert [13] classified measurements as coherent or not, thus predicting if they will result in erroneous data association. Indelman et al. [47] addressed data association using a combination of EM and model selection based on Chinese Restaurant Process.

Related work that considered data association in the context of semantics are the approaches of Mu et al. [87] and Bowman et al. [10], which considered data association as part of a joint semantic and geometric formulation. However, both works propose an iterative optimization algorithm which may converge to a local minimum. In contrast, the robust inference approach in Pathak, Thomas, and Indelman [95] maintains the relevant components of the mixture posterior, until full disambiguation is possible, however, not considering semantic information. Another approach to tackle the data association problem is a convex re-formulation that is robust to outliers by Carlone and Calafiore [14], however robustness is empirical, and probability of wrong matches is not directly maintained.

Milan et al. [82] presented a method based on LSTM neural network for data association, training it on the MOTChallenge dataset. Farazi and Behnke [27] expended on the above work to visually track and associate between identical robots using an LSTM based approach. Milan et al. [82] and Farazi and Behnke [27] are both deep learning based approaches, where a key question is how far the deployment scenario is from the training set, i.e. covariate shift.

More recently, Lajoie et al. [67] performed inference on a hybrid factor graph ("discrete-continuous graphical model" DC-GM) producing near-optimal estimates via convex relaxation. Hsiao and Kaess [45] developed a general framework extending that of Kaess et al. [50] for multi-hypothesis inference using a hypothesis tree, pruning and calculation reuse across tree levels.

2.6 Inference with Learned Models, Learned Semantic Representations

The expressiveness of Deep Learning Models has allowed their use as measurement models learned from data, for cases where an analytical model is complicated or does not exist. Dosovitskiy et al. [23] learned a viewpoint-dependent model that given viewpoint and object class generates an image of the object from that viewpoint, demonstrating some generalization. Baikovitz et al. [4] learn a model to map measurements of ground-penetrating radar to ego-motion (pose difference between the corresponding

time steps), and similarly do Sodhi et al. [113] for tactile measurements. In both cases the learnt model defines a factor subsequently used in factor-graph inference. Kopitkov and Indelman [59] does likewise (however only for inference of localization). Sodhi et al. [112] formulates a general framework for learning an observation model *as part of inference*, i.e. the learning objective being precise inference using the model (as opposed to supervised training of the model outside of inference otherwise).

Other recent approaches simultaneously learn a discriminative model and a latent space the model is conditioned on. Bloesch et al. [9], Czarnowski et al. [18], and Matsuki et al. [78] represent per-frame depth images with latent variables, then jointly refine depth for all frames using a learned depth measurement model over the latent space in (or in conjuction with Matsuki et al. [78]) a SLAM formulation. Using a similar idea for semantics, Zhi et al. [136] construct a dense semantic map by estimating then fusing latent per-timestep "scene codes" using a learned decoder as measurements model. Yu and Lee [133] and Sucar, Wada, and Davison [118] learn a latent representation of object shape and use the corresponding model to perform inference (i.e. the learned latent space corresponds to per-object shape descriptors). Yu, Moon, and Lee [134] use a similarly learned model to perform joint inference over object shape and discrete category, in a EM formulation.

The latter are reminiscent of generative models (Kingma and Welling [56] and Goodfellow et al. [40]) and in fact build upon the vast research in that area. Closely related are conditional GANs (Mirza and Osindero [86]) however without the randomization in the generator and conditional VAEs (Sohn, Lee, and Yan [115]). In contrast to common generative models however, the goal is commonly not necessarily to generate an interpretable element in the measurement modality, but rather use the decoder, or generator to optimize the latent space variable (as well as additional conditioning variables) as part of SLAM inference.

Another field related to learning latent descriptors is that of unsupervised representation learning. Pirk et al. [100] learn object representations which emergently capture semantics, in an unsupervised manner, for associating detections in consecutive frames. Le-Khac, Healy, and Smeaton [55] review and describe a general framework for contrastive representation learning. However that field is not concerned with models for inference.

Chapter 3

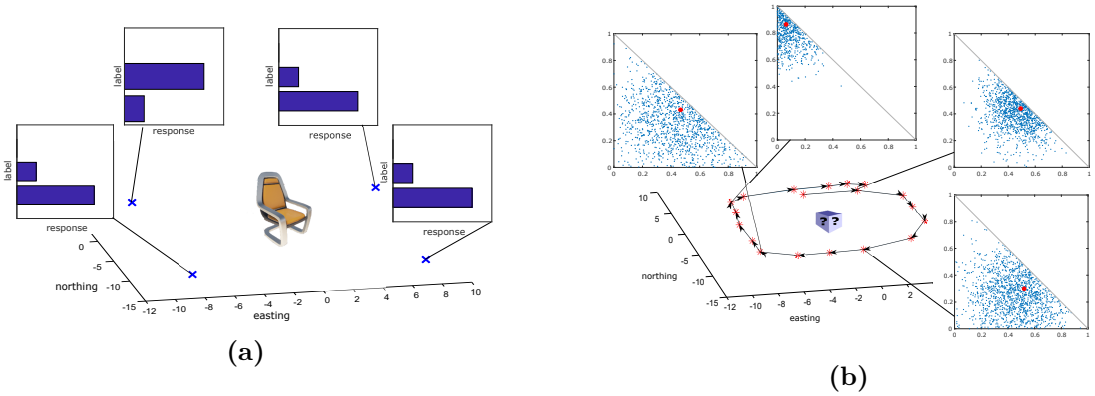
Spatially-dependent Bayesian Semantic Perception under Model and Localization Uncertainty

3.1 Introduction

In this chapter we develop a method for fusion of classifier measurements focusing on classification of a single object, viewed by a moving robot under localization and model uncertainty. We make the following contributions: First, we show how viewpoint - dependent class models capturing both viewpoint—dependent variation and spatial correlation in classifier scores can be applied under uncertain localization with continuous pose. Second, we incorporate semantic measurements carrying information of model uncertainty in inference, allowing classification to be robust to dataset shift, by reflecting the associated uncertainty in the posterior. Finally, our method uses the entire classification vector/semantic measurement output by the classifier, making it more general. We validate the approach in MATLAB simulation, in a 3D simulated environment using the UnrealEngine game engine, and finally with real images from the BigBIRD (Singh et al. [111]) and AVD (Ammirato et al. [1]) datasets.

3.2 Problem Definition and Notations

Consider a robot traversing an unknown environment, taking observations of different scenes. The robot's motion between times k and $k+1$ is initiated by a control input u_k, that may originate from a human user, or be determined by a motion planning algorithm. We denote the robot's pose at time instant k by x_k, and by $\mathcal{X}_{0:k} = \{x_0, \ldots, x_k\}$ the sequence of poses up to that time. Let $\mathcal{H}_k = \{\mathcal{U}_{0:k-1}, \mathcal{Z}_{0:k}\}$ represent the history,



(a) (b)

Figure 3.1: Left: Viewpoint-dependent variations in classifier outputs are modeled as noise if viewpoint-dependence is ignored. The GP model for an object class (Sec. 3.2.2) is created from classifier output samples (entire classification vectors) at known relative locations around object. In our simulation scenarios, samples and their locations were specified manually for each of the classes. Right: The robot acquires observations along a track in the vicinity of the object of interest. At each time step, the classifier outputs a cloud of classification vectors reflecting the model uncertainty, unlike a single vector measurement (red dot) or a component thereof in other approaches.

comprising observations $\mathcal{Z}_{0:k} = \{z_0, \ldots, z_k\}$ and controls $\mathcal{U}_{0:k-1} = \{u_0, \ldots, u_{k-1}\}$ up until time k. We focus on the task of classification of a single object belonging to one of N_c known classes, denoted by indexes $\mathcal{C} = \{1, \ldots, N_c\}$.

Our goal is to maintain the classification posterior, or *belief*, at time instant k:

$$b[c_k] \doteq \mathbb{P}(c \mid \mathcal{H}_k). \tag{3.1}$$

The classification posterior is the probability of the object in question to belong to class $c \in \mathcal{C}$, given all measurements and user controls up to time k. In calculating this posterior we want to take into account spatial correlation among measurements, model uncertainty, as well as uncertainty in pose from which these measurements are taken (localization uncertainty).

3.2.1 Classifier Measurements with Uncertainty

In the Bayesian approach, given an observation z_k (in our context, an image) we are interested in obtaining the class (categorical) posterior

$$\mathbb{P}(s^{(i)} = c \mid z_k, \mathcal{D}), \tag{3.2}$$

where $s^{(i)}$ is the i'th component of a categorical vector s (i.e. $\sum_i s^{(i)} = 1$) and $\mathcal{D}$ is training data. For a given classifier unit, the vector θ of its parameters either determines its output as function of input, or otherwise can be regarded as a sufficient statistic.

In either case, we can rewrite Eq. (3.2) as

$$\mathbb{P}(s^{(i)} = c \mid z_k, \mathcal{D}) = \int_\theta \mathbb{P}(s^{(i)} = c \mid z_k, \theta) \cdot \underbrace{\mathbb{P}(\theta \mid \mathcal{D})}_{\substack{\text{Model} \\ \text{Uncertainty}}} d\theta. \tag{3.3}$$

The second term above is referred to as *Model Uncertainty* or *Epistemic Uncertainty* (Gal [34] and Kendall and Gal [53]), capturing uncertainty in classifier parameters given our training data. Uncertainty in θ induces a distribution over the values of $\mathbb{P}(s^{(i)} = c \mid z, \theta)$, a *distribution over distributions* (Malinin and Gales [75]). The latter should intuitively be relatively concentrated near training data and spread out away from training data, reflecting the level of uncertainty in the classifier output.

We approximate the model uncertainty of the neural network classifier using MC-Dropout proposed by Gal and Ghahramani [35], although this is not required by our approach and any other technique providing (approximate) samples from $\mathbb{P}(\theta \mid \mathcal{D})$ may be equally used. We chose this method for its simplicity, although Myshkov and Julier [88] show that it may underestimate the model uncertainty in some cases. In short, for every classifier input we perform several forward passes applying random dropouts at each pass, to obtain a set of classification vectors, characterizing the uncertainty. Fig. 3.1b illustrates classifier measurements obtained over a track in the vicinity of an object of interest, when using MC-Dropout to approximate model uncertainty. Each measurement is a set of classification vectors (in the illustration - a point in the 3-d simplex corresponds to the case of 3 candidate classes) characterizing the model uncertainty, which generally varies among different viewpoints. Model uncertainty is complementary to the "classification confidence" which corresponds to the distance of the mean classification vector (red dot inside the simplex) from the corners.

Formally, we assume that the robot has at its disposal an object classifier unit, which, given observation z_k (e.g., an image), calculates a set of outputs $\mathcal{S}_k \triangleq \{s_k\}$, where each output $s_k \in \mathbb{R}^{N_c \times 1}$ represents a categorical belief over the class of the observed object, i.e. $\sum_{i=1}^{N_c} s_k^{(i)} = 1$.

The set S_k can be interpreted as an approximation to the predictive distribution from Eq. (3.2),

$$\forall s \in \mathcal{S}_k \quad s \sim \mathbb{P}(\cdot \mid z_k, \mathcal{D}), \tag{3.4}$$

carrying information of the classifier's model uncertainty for the given input z_k.

3.2.2 Viewpoint-Dependent Class Model

For the class likelihood we use a model similar to the one proposed by Teacy et al. [126]. For a single classifier measurement s (categorical vector) captured from relative

pose $x^{(rel)}$, the class likelihood is a probabilistic model

$$\mathbb{P}(s \mid c, x_k^{(rel)}), \tag{3.5}$$

where $c \in \mathcal{C}$ is the object class, and the k subscript denotes time index. Denoting object pose in global frame as o we can explicitly write

$$x_k^{(rel)} \doteq x_k \ominus o. \tag{3.6}$$

The dependence of the model in Eq. (3.5) on viewpoint naturally captures view-dependent variations in object appearance and as a result in classifier responses as illustrated in Fig. 3.1a. Further, to incorporate the notion that similar views tend to yield similar classifier responses and in particular, are not independent, we consider the joint distribution

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, \mathcal{X}_{0:k}^{(rel)}), \tag{3.7}$$

characterizing the classification unit's outputs $\mathcal{S}_{0:k} \doteq \{S_0, \ldots, S_k\}$ when viewing an object of class c from a sequence of relative poses $\mathcal{X}_{0:k}^{(rel)} \doteq \{x_0^{(rel)}, \ldots, x_k^{(rel)}\}$. Similar to Teacy et al. [126] and Velez et al. [129], we represent this joint distribution with a Gaussian Process, learned using the classifier unit. Explicitly, we model training set classifier response when viewing object of class c from relative pose $\mathcal{X}^{(rel)}$ as

$$s^{(i)} = f_{i|c}(\mathcal{X}^{(rel)}) + \varepsilon, \tag{3.8}$$

where the i index denotes component i of classification vector s, $\varepsilon \sim N(0, \sigma_n^2)$ i.i.d. noise, and (dropping the (rel) superscript for clarity)

$$f_{i|c}(x) \sim \mathcal{GP}\left(\mu_{i|c}(x), k_{i|c}(x, x)\right), \tag{3.9}$$

where $\mu_{i|c}$ and $k_{i|c}$ are the mean and covariance functions defining the GP

$$\mu_{i|c}(x) = \mathbb{E}\{s^{(i)} \mid c, x\} \tag{3.10}$$

$$k_{i|c}(x, x') = \mathbb{E}\{(f_{i|c}(x) - \mu_{i|c}(x))(f_{i|c}(x') - \mu_{i|c}(x'))\} \tag{3.11}$$

We thus model the classification vector for each class c with independent, per-component GP's. Note also the Gaussian approximation of the distribution of the classification vector, which resides in the simplex (other representations exist, which however are not readily interpreted as a spatial model).

For the covariance we use the squared exponential function:

$$k_{i|c}(x, x') = \sigma_{i|c}^2 \exp(-\frac{1}{2}(x - x')^T L_{i|c}^{-1}(x - x')), \tag{3.12}$$

where $\sigma^2_{i|c}$ is the variance, and $L_{i|c}$ is the length scale matrix (of the dimension of x), determining the rate of the covariance decay with distance. These parameters can be learned from training data (Rasmussen and Williams [104]).

Denote the training set for class c as $\{S^c_T, X^c_T\}$, with S^c_T classifier measurements, and X^c_T the corresponding poses, and denote (test-time) measurements as $S = \mathcal{S}_{0:k}$ and $X = \mathcal{X}^{(rel)}_{0:k}$. Note that training set classifier measurements are obtained without dropout i.e. they do not carry model uncertainty information. Further, the following equations Eqs. (3.13-3.16) all hold per vector-component (joined in Eq. (3.17)), i.e. for simplifying notation we drop the i index in $\mathcal{S}^{(i)}$, $\mathcal{S}^{(i)}_T$ and $k_{i|c}$.

We follow Rasmussen and Williams [104] and model the joint distribution of classifier measurements (per-component) for object of class c as

$$\mathbb{P}(S^c_T, S \mid c, X^c_T, X) = N\left(0, \begin{bmatrix} K_c(X^c_T, X^c_T) + \sigma^2_n I & K_c(X^c_T, X) \\ K_c(X, X^c_T) & K_c(X, X) \end{bmatrix}\right), \tag{3.13}$$

where S^c_T are the classifier measurements used for training, not carying model uncertainty information, $S = \mathcal{S}_{0:k}$ are the test-time measurements, such that $\mathcal{S}_j \sim \mathbb{P}(\cdot \mid z_j)$ as in Eq. (3.4), and K_c is the matrix produced by application of kernel k_c on all pairs of input vectors. We thus obtain the conditional distribution for classifier measurements of object of class c

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, X^c_T, S^c_T, \mathcal{X}^{(rel)}_{0:k}) = N(\mu, \Sigma), \tag{3.14}$$

with

$$\mu = K_c(X, X^c_T) \cdot H \cdot S \tag{3.15}$$

$$\Sigma = K_c(X, X) - K_c(X, X^c_T) \cdot H \cdot K_c(X^c_T, X), \tag{3.16}$$

and where $H \doteq \left(K_c(X^c_T, X^c_T) + \sigma^2_n I\right)^{-1}$.

Going back to the time notation, we finalize by combining the per-component models into a joint class likelihood as

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, X^c_T, S^c_T, \mathcal{X}^{(rel)}_{0:k}) = \prod_i \mathbb{P}(\mathcal{S}^{(i)}_{0:k} \mid c, X^c_T, S^{c,(i)}_T, \mathcal{X}^{(rel)}_{0:k}), \tag{3.17}$$

where the superscript (i) refers to vector component, as above. Note that this formulation somewhat differs from Teacy et al. [126], where inference from training data is done by offline learning of GP mean and covariance functions rather than using a joint distribution as in Eq. (3.13). Also note that while for simplicity we defined the model in Eq. (3.13) per-component (considering components as independent), subsequent development also holds for a joint model, i.e. jointly referring to all components of classifier measurements in Eqs. (3.13-3.16), with full covariance in Eq. (3.14) rather than block-diagonal as induced by Eq. (3.17).

3.2.3 Assumptions

We briefly review basic assumptions in the context of the formulation above. We assume that we have access to a **Bayesian classifier**, which given raw measurement z provides samples from a probability distribution over classification vectors characterizing the predictive uncertainty $\mathbb{P}(s \mid z, \mathcal{D})$, see Eq. (3.4). We assume that **viewpoint-dependent class models** are available, modeling the spatial responses $\mathcal{S}_{0:k}$ of that classifier given poses relative to object $\mathcal{X}_{0:k}^{(rel)}$, in a joint Gaussian distribution $\mathbb{P}(\mathcal{S}_{0:k} \mid c, \mathcal{X}_{0:k}^{(rel)})$ for objects of class c, where the Gaussian form is required for closed-form calculations in the following. As described in the previous section, Gaussian Process regressors can be learned to implement such models. Existence of pre-trained models for known classes implies **closed set** conditions, i.e. the possible object classes are known in advance. Objects of novel classes are assumed to trigger high predictive uncertainty in the classifier output (in itself an assumption, since predictive uncertainty depends on the classifier training set), reflected in uncertainty in the posterior. Further, the formulation focuses on classification of a single, static object. In general scenes comprising multiple objects this formulation can be applied for each object separately, under the assumption of **known data association**, and the additional underlying assumption of negligibility of the effect of occlusions. Finally, in this work we assume that the **orientation** of the object relative to the robot **is known** (perhaps, detected from observations). The need for this assumption becomes apparent and is briefly discussed in Sec. 3.3.2. Under these assumptions, in the following section we detail our approach to the classification problem formulated above.

3.3 Approach

To account for both localization and model uncertainty we rewrite Eq. (3.1) as marginalization over latent robot and object poses, and over classifier outputs. We start by marginalizing over robot pose history and object pose

$$b[c_k] = \mathbb{P}(c \mid \mathcal{H}_k) = \int_{\mathcal{X}_{0:k}, o} \mathbb{P}(c, \mathcal{X}_{0:k}, o \mid \mathcal{H}_k)\ d\mathcal{X}_{0:k} do, \tag{3.18}$$

which, using chain rule, can be written as

$$b[c_k] = \int_{\mathcal{X}_{0:k}, o} \underbrace{\mathbb{P}(c \mid \mathcal{X}_{0:k}, o, \mathcal{H}_k)}_{(a)} \underbrace{\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)}_{(b)}\ d\mathcal{X}_{0:k} do. \tag{3.19}$$

Term (a) above is the classification belief given relative poses $\mathcal{X}_{0:k}^{(rel)}$ which are calculated from $\mathcal{X}_{0:k}$ and o via Eq. (3.6). Term (b) represents the posterior over $\mathcal{X}_{0:k}$ and o given observations and controls thus far. As such, this term can be obtained from existing

SLAM approaches. One can further rewrite the above equation as

$$b[c_k] = \mathop{\mathbb{E}}_{\mathcal{X}_{0:k},o}\{\mathbb{P}(c \mid \mathcal{X}_{0:k}^{(rel)}, \mathcal{H}_k)\}, \tag{3.20}$$

where the expectation is taken with respect to the posterior $p\left(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k\right)$ from term (b). In practice, this posterior can be computed using SLAM methods (see Section 3.3.2), which commonly model it with a Gaussian distribution. We then use the obtained distribution to approximate the expectation in Eq. (3.20) using sampling.

In the following we detail the computation of the terms (a) and (b) of Eq. (3.19).

3.3.1 Classification Under Known Localization

In this section we develop the update of classification belief given known pose history, term (a) in Eq. (3.19), when receiving new measurements at time step k, while accounting for correlations with previous measurements and model uncertainty.

To simplify notation, we shall denote history of observations, controls and (known) relative poses as

$$H_k \doteq \mathcal{H}_k \cup \mathcal{X}_{0:k}^{(rel)} \equiv \{\mathcal{U}_{0:k-1}, \mathcal{Z}_{0:k}, \mathcal{X}_{0:k}^{(rel)}\}. \tag{3.21}$$

We start by marginalizing term (a) over model uncertainty in the classifier output at time k

$$\mathbb{P}(c \mid H_k) = \int_{s_k} \mathbb{P}(c \mid s_k, H_k) \cdot \mathbb{P}(s_k \mid H_k)\ ds_k. \tag{3.22}$$

Assuming s_k carries the class information from measurement z_k, and that $s_k \sim \mathbb{P}(s_k \mid z_k)$ we can rewrite this as

$$\mathbb{P}(c \mid H_k) = \int_{s_k} \mathbb{P}(c \mid s_k, H_k \setminus \{z_k\}) \cdot \mathbb{P}(s_k \mid z_k)\, ds_k. \tag{3.23}$$

In our case, $\{s_k\}$ are samples from $\mathbb{P}(s_k \mid z_k)$, so we can approximate the integral as

$$\mathbb{P}(c \mid H_k) \approx \frac{1}{n_k} \sum_{s_k \in \mathcal{S}_k} \mathbb{P}(c \mid s_k, H_k \setminus \{z_k\}). \tag{3.24}$$

To calculate the summand, we apply Bayes' law and then smoothing over class in the denominator

$$\mathbb{P}(c \mid H_k) = \sum_{s_k} \frac{\eta(s_k)}{n_k} \cdot \mathbb{P}(s_k \mid c, H_k \setminus \{z_k\}) \cdot \mathbb{P}(c \mid H_k \setminus \{z_k\}) \tag{3.25}$$

with

$$\eta(s_k) \doteq 1/\sum_{c\in\mathcal{C}} \mathbb{P}(s_k \mid c, H_k \setminus \{z_k\})\mathbb{P}(c \mid H_k \setminus \{z_k\}). \tag{3.26}$$

Note that the denominator in $\eta(s_k)$ is a sum of numerator (summand) terms in Eq. (3.25) for the different classes and can be computed efficiently (but cannot be discarded altogether due to the dependence on s_k). Further, note that

$$\mathbb{P}(c \mid H_k \setminus \{z_k\}) = \mathbb{P}(c \mid \mathcal{X}_{0:k}^{(rel)}, \mathcal{Z}_{0:k-1}) = \mathbb{P}(c \mid \mathcal{X}_{0:k-1}^{(rel)}, \mathcal{Z}_{0:k-1}) = \mathbb{P}(c \mid H_{k-1}). \tag{3.27}$$

As $\mathbb{P}(c \mid H_{k-1})$ has been computed in the previous step, we are left to compute the class likelihood term $\mathbb{P}(s_k \mid c, H_k \setminus \{z_k\})$. This term involves past observations $\mathcal{Z}_{0:k-1}$ but not classifier outputs $\mathcal{S}_{0:k-1}$, which need to be introduced to account for spatial correlation with s_k using Eq. (3.7). Marginalizing over $\mathcal{S}_{0:k-1}$ (recall that in our notation $\mathcal{S}_{0:k-1} \cup \{s_k\} = \mathcal{S}_{0:k}$) yields

$$\begin{aligned}\mathbb{P}(s_k \mid c, H_k \setminus \{z_k\}) &= \int_{\mathcal{S}_{0:k-1}} \mathbb{P}(\mathcal{S}_{0:k} \mid c, H_k \setminus \{z_k\})\; d\mathcal{S}_{0:k-1} \\ = \int_{\mathcal{S}_{0:k-1}} \mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, H_k \setminus \{z_k\}) \cdot\; &\mathbb{P}(\mathcal{S}_{0:k-1} \mid c, H_k \setminus \{z_k\})\; d\mathcal{S}_{0:k-1},\end{aligned} \tag{3.28}$$

where we applied smoothing to separate between past classifier outputs $\mathcal{S}_{0:k-1}$ for which observations $\mathcal{Z}_{0:k-1}$ are given and the current output s_k. The first term in the product reduces to $\mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)})$, a conditioned form of the class model Eq. (3.14) (and thus Gaussian, which we treat explicitly later in Eq. (3.31) and on). This term represents the probability to obtain classification s_k when observing an object of class c from relative pose $\mathcal{X}_k^{(rel)}$ given previous classification results and relative poses. The second term in Eq. (3.28) can be approximated using Eq. (3.4) for the individual observations z_i, i.e.

$$\mathbb{P}(\mathcal{S}_{0:k-1} \mid c, H_k \setminus \{z_k\}) = \mathbb{P}(\mathcal{S}_{0:k-1} \mid \mathcal{Z}_{0:k-1}) \approx \prod_{i=0}^{k-1} \mathbb{P}(s_i \,|\, z_i)$$

Note that class c and poses $\mathcal{X}_{0:k-1}^{(rel)}$, both members of H_k can be omitted since conditioning on observations determines classifier outputs up to uncertainty due to classifier intrinsics (model uncertainty). The approximation is in the last equality, since in general classifier outputs $s_0, \ldots, s_{k-1}$ are interdependent through classifier parameters. We can now rewrite $\mathbb{P}(s_k \mid c, H_k \setminus \{z_k\})$ from Eq. (3.28) as

$$\int_{\mathcal{S}_{0:k-1}} \mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}) \cdot \prod_{i=0}^{k-1} \mathbb{P}(s_i \mid z_i)\; d\mathcal{S}_{0:k-1}. \tag{3.29}$$

Assuming classifier output Eq. (3.4) is Gaussian, we denote

$$\mathbb{P}(s_i \mid z_i) = N(\mu_{z_i}, \Sigma_{z_i}), \tag{3.30}$$

where μ_{z_i} and Σ_{z_i} are estimated from $\mathcal{S}_i$. Since class model is Gaussian, see Eq. (3.14), the first term in the integrand in Eq. (3.29) is a Gaussian that we denote as

$$\mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}) = N(\mu_{k|0:k-1}, \Sigma_{k|0:k-1}) \tag{3.31}$$

where, utilizing standard Gaussian Process equations (Rasmussen and Williams [104]),

$$\mu_{k|0:k-1} = \mu_k + \Omega \cdot (\mathcal{S}_{0:k-1} - \mu_{0:k-1}) \tag{3.32}$$

$$\Sigma_{k|0:k-1} = K(x_k, x_k) - \Omega \cdot K(\mathcal{X}_{0:k-1}, x_k) \tag{3.33}$$

with $\Omega \doteq K(x_k, \mathcal{X}_{0:k-1}) K\left(\mathcal{X}_{0:k-1}, \mathcal{X}_{0:k-1}\right)^{-1}$.

Using these relations, the integrand from Eq. (3.29) is a Gaussian distribution over $\mathcal{S}_{0:k}$, that can be inferred as follows.

$$\mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}) \cdot \prod_{i=0}^{k-1} \mathbb{P}(s_i \mid z_i) = \tag{3.34}$$

$$\eta \exp\left\{-\frac{1}{2}\left(\|s_k - \mu_{k|0:k-1}\|^2_{\Sigma_{k|0:k-1}} + \sum_{i=0}^{k-1} \|s_i - \mu_{z_i}\|^2_{\Sigma_{z_i}}\right)\right\},$$

where η only depends on $\mathcal{X}_{0:k}^{(rel)}$. Using Eq. (3.32) we can write

$$s_k - \mu_{k|0:k-1} = s_k - \mu_k - \Omega \cdot (\mathcal{S}_{0:k-1} - \mu_{0:k-1}) = \begin{bmatrix} -\Omega & I \end{bmatrix} (\mathcal{S}_{0:k} - \mu_{0:k}) \tag{3.35}$$

We have that the integrand Eq. (3.34) from Eq. (3.29) is proportional to a joint Gaussian distribution $N\left(\mu_J, \Sigma_J\right)$ with

$$\Sigma_J = \left(\Sigma_s^{-1} + \Sigma_z^{-1}\right)^{-1} \tag{3.36}$$

$$\mu_J = \Sigma_J \cdot \left(\Sigma_s^{-1}\mu_s + \Sigma_z^{-1}\mu_z\right), \tag{3.37}$$

where

$$\mu_s = \begin{pmatrix} \mu_0 \\ \vdots \\ \mu_{k-1} \\ \mu_k \end{pmatrix} \quad \mu_z = \begin{pmatrix} \mu_{z_0} \\ \vdots \\ \mu_{z_{k-1}} \\ 0 \end{pmatrix}, \tag{3.38}$$

and

$$\Sigma_s^{-1} = \begin{bmatrix} -\Omega & I \end{bmatrix}^T \Sigma_{k|0:k-1}^{-1} \begin{bmatrix} -\Omega & I \end{bmatrix} \tag{3.39}$$

$$\Sigma_z^{-1} = \begin{pmatrix} \Sigma_{z_0}^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \Sigma_{z_{k-1}}^{-1} & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix} \tag{3.40}$$

Finally, the class likelihood from Eq. (3.28) is the marginal distribution of the above. Specifically, the integral is directly calculated by evaluation at s_k of a Gaussian PDF with the components corresponding to s_k from μ_J and Σ_J as mean and covariance.

So far, we have described how to update the class belief given known localization, term (a) of Eq. (3.19), upon arrival of new measurements. We now proceed to describe how the localization belief, term (b), is computed.

3.3.2 Localization Inference

In this work we assume that the object orientation relative to the robot is known (leaving o with 3 degrees of freedom), and so term (b) of Eq. (3.19) is essentially a SLAM problem with robot pose history $\mathcal{X}_{0:k}$ and one landmark, the target object localization o, to be inferred. Specifically, we can express the target distribution as marginalization over all landmarks $\mathcal{L}$, except for the object of interest

$$\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k) = \int_{\mathcal{L}} \mathbb{P}(\mathcal{X}_{0:k}, o, \mathcal{L} \mid \mathcal{H}_k)\ d\mathcal{L}. \tag{3.41}$$

This can be computed using state of the art methods, e.g. if a Gaussian posterior is assumed, using an efficient solver such as iSAM2 (Kaess et al. [49]). Note that the above assumption of Gaussian posterior is not a requirement of the formulation in Eq. (3.19) - indeed, using sampling allows the localization posterior to be of any distribution provided it can be sampled from.

Also note that the history $\mathcal{H}_k$ contains both geometric and semantic measurements, allowing related work to leverage the viewpoint-dependent model in inference as a semantic-geometric factor (Kopitkov and Indelman [59] and Tchuiev, Feldman, and Indelman [124]). Our case is slightly different however, as the formulation accounts for model uncertainty and correlation among viewpoints. In this work we limit ourselves to usage of semantic factors for semantic (term (a) of Eq. (3.19)) but not geometric inference (geometry helps semantics Cadena et al. [11]), requiring relative object orientation to be externally known.

Algorithm 3.1 Localization and model uncertainty aware classification.

This describes computations for time step k.

Require: Localization posterior $\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$, new observations $\mathcal{Z}_k$

1: $\mathcal{S}_k = \{s_k^{(i)}\}_{i=1}^{n_k} \leftarrow$ ClassifyWithDropout($\mathcal{Z}_k$), pre-calculate Σ_z^{-1}, μ_z (Eq. (3.40), Eq. (3.38))
2: Sample $\left\{\mathcal{X}_{0:k}^{(i)}, o^{(i)}\right\}_{i=1}^{n_x} \sim \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$
3: **for** $\mathcal{X}_{0:k}, o \in \{\mathcal{X}_{0:k}, o\}$ **do**
4: $\quad \forall c \in \mathcal{C}$ calculate Σ_J, μ_J (Eq. (3.36), Eq. (3.37)) ▷ These are per-class through GP model
5: $\quad$ **for** $s_k \in \mathcal{S}_k$ **do**
6: $\quad\quad$ **for** $c \in \mathcal{C}$ **do**
7: $\quad\quad\quad \mathbb{P}(s_k \mid c, H_k \setminus \{\mathcal{Z}_k\}) \leftarrow N(s_k; \mu_J^{(k)}, \Sigma_J^{(k,k)})$ ▷ Likelihood given past observations Eq. (3.29)
8: $\quad\quad\quad \tilde{h}^c \leftarrow \mathbb{P}(s_k \mid c, H_k \setminus \{\mathcal{Z}_k\}) \cdot \mathbb{P}(c \mid H_{k-1})$ ▷ Unnormalized class likelihood, Eq. (3.25)
9: $\quad\quad$ **end for**
10: $\quad\quad \mathbb{P}(c \mid s_k, H_k) \leftarrow \tilde{h}^c / \eta(s_k), \quad \eta(s_k) = \sum_c \tilde{h}^c$ ▷ Normalize class likelihood, Eq. (3.25)
11: $\quad$ **end for**
12: $\quad \mathbb{P}(c \mid H_k)^{(i)} \leftarrow \sum_{s_k} \mathbb{P}(c \mid s_k, H_k)/n_k$ ▷ Classification given localization Eq. (3.24)
13: **end for**
14: $\mathbb{P}(c \mid \mathcal{H}_k) \leftarrow \sum_i \mathbb{P}(c \mid H_k)^{(i)}/n_x$

Note: Lines 10, 12, 14 apply per-class, $\forall c \in \mathcal{C}$. In step 7, superscripts select elements corresponding to time k in the joint mean vector and covariance matrix.

3.3.3 Overall Algorithm

Alg. 3.1 summarizes computations for time step k in the localization and model uncertainty aware classification approach described above. Observations $\mathcal{Z}_k$ are passed through the classifier unit to obtain semantic measurements and model uncertainty, at which point summary statistics can be computed, for use in Eq. (3.29). To account for localization uncertainty, robot and object poses are sampled from the current (updated) posterior. For each sample, class model predictions (mean and covariance matrices) are computed for all classes. To account for model uncertainty, measurement likelihood is computed for each $s_k^{(i)} \in \mathcal{S}_k$, then averaged. Note that formulation using the conditional distribution of s_k given past measurements in Eq. (3.29) allows efficient marginalization in Eq. (3.25) and computation of normalizing constant $\eta(s_k)$ in Eq. (3.26), which would otherwise require intractable summation over combinations of individual semantic measurements. Similarly, using summary statistics (fitting a Gaussian per time-step) over past semantic measurements allows the marginalization in Eq. (3.29) to be computed in closed form. Classifications thus computed per localization sample are averaged to yield the final output. In the next section we analyze

the computational complexity of Alg. 3.1, then present the experimental evaluation and results.

3.3.4 Computational Aspects

While using samples allows in principle to approximate arbitrary non-tractable or unknown probability distributions, on the downside it involves computations repeated per-sample, inducing a sharp precision vs. runtime trade-off on the number of samples, especially so as state dimension grows over time. Additionally, in classification, most computations are repeated per class. Finally, complexity depends on the number of time steps k, which affects both the overall number of observations to be processed, and the number of state variables in the smoothing formulation. Note that k value relevant to the classification of an object is the number of time steps it has been observed, as other poses can be dropped from the marginalization in Eq. (3.19), since measurement model and thus classification is conditionally independent of them.

In this section we analyze the theoretical complexity of the proposed method, then present an approach for incremental sampling of robot and object poses allowing to reduce repeated calculations, and discussion.

3.3.4.1 Theoretical Complexity

We denote the number of model uncertainty samples (per time step) in step 1 of Alg. 3.1 as N_s, number of localization samples in step 2 as N_x and number of classes as N_c, and use them to express complexity. We assume these are constant throughout the run. We further denote as N_f the dimension of the semantic feature vector, i.e. $\forall k, i\ s_k^{(i)} \in \mathbb{R}^{N_f}$. In our implementation, $N_f \equiv N_c$ as we use the raw classification vectors as features.

Step 1 depends on the classification and model uncertainty computation method. In the case of a Neural Network classifier with a fixed architecture using MC-dropout, we assume each forward pass to take constant time, making complexity linear in the number of forward passes $O(N_s)$. In the same step, computation of μ_z takes $O(N_f N_s)$ and of Σ_z^{-1} takes $O(N_s N_f^2 + N_f^3)$, which is also the overall complexity for this step. Step 2 involves sampling an entire trajectory (MCMC) at each time step i.e. $O(k \cdot N_x)$. Step 4 involves (1) prediction of μ_s (2) computation of $\Sigma_{k|0:k-1}$ and of Σ_s^{-1} (see Eq. (3.33) and Eq. (3.39)), and (3) Computations of Σ_J and μ_J (see Eq. (3.36) and Eq. (3.37)). These values depend on the robot trajectory sampled in step 2 and on the class models. Complexity is dominated by the inversion in Eq. (3.36) at $O(N_f\, k^3)$ when done separately for each semantic feature. Calculations in this step can be reduced if trajectory samples from previous time steps are re-used, as discussed in Sec. 3.3.4.2. In all, complexity for step 4 is $O(N_c\, N_f\, k^3)$ as computation is done for each class. Since Σ_J, μ_J are pre-computed in step 4, calculation of likelihood in step 7 takes $O(N_f)$ (covariance-form marginalization, followed by evaluation of 1-dimensional Gaussian PDFs for N_f features). Step 8 incurs no additional cost (i.e., $O(1)$) when $\mathbb{P}(c \mid H_{k-1})$

can be assumed pre-computed, which corresponds to re-use of localization samples from previous time indexes (see Sec. 3.3.4.2). In this case complexity of inner loop steps 5-12 is $O(k^3\, N_c\, N_s\, N_f)$. Otherwise, if past localization samples are not re-used, computation from scratch is required, which can be done by repeating steps 4-12 to incrementally compute $\mathbb{P}(c \mid H_1), \mathbb{P}(c \mid H_2), \ldots, \mathbb{P}(c \mid H_{k-1})$, which also requires that semantic measurements from all previous time steps be kept. Incremental computation for time indexes $0, \ldots, k$ results in complexity $O(k^4\, N_c\, N_s\, N_f)$. Overall, total time complexity for time step k stands at $O(k^4\, N_c\, N_s\, N_f\, N_x + N_s\, N_f^2 + N_f^3)$ if class posterior in step 8 needs to be recomputed, or $O(k^3\, N_c\, N_s\, N_f\, N_x + N_s\, N_f^2 + N_f^3)$ if it does not, where k is the number of viewpoints from which the object was observed.

3.3.4.2 Incremental Sampling

Step 2 in Alg. 3.1 involves sampling of robot trajectory, a state vector which grows over time, in order to compute the marginalization in Eq. (3.19), accounting for localization uncertainty. This marginalization must involve a history of robot poses because of the measurement model (GP), Eq. (3.7), which captures correlations among present and past views.

While, as noted in Sec. 3.3.4, this sampling can be limited to poses at time indices corresponding to views of the object, a further runtime improvement can be achieved if trajectory samples from past time indices can be re-used in samples from the updated posterior.

Specifically, sample reuse would benefit step 4, where some values can be updated instead of being re-calculated, and especially step 8 where values from past time steps can be re-used eliminating the required calculations to $O(1)$, in addition to directly reducing sampling in step 2.

Formally, assuming at time step k we can use samples $\{\mathcal{X}_{0:k}^{(i)}, o^{(i)}\}_{i=1}^{N_s} \sim \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$ to approximate integral Eq. (3.19) with sum

$$\begin{aligned}\mathbb{P}(c \mid \mathcal{H}_k) &= \int_{\mathcal{X}_{0:k}, o} \mathbb{P}(c \mid \mathcal{X}_{0:k}, o, \mathcal{H}_k) \cdot \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)\; d\mathcal{X}_{0:k} do \\ &\approx \sum_i \mathbb{P}(c \mid \mathcal{X}_{0:k}^{(i)}, o^{(i)}, \mathcal{H}_k) \cdot \mathbb{P}(\mathcal{X}_{0:k}^{(i)}, o^{(i)} \mid \mathcal{H}_k)\; d\mathcal{X}_{0:k} do, \end{aligned} \tag{3.42}$$

at time step $k+1$ we would like to approximate

$$\mathbb{P}(\mathcal{X}_{0:k+1}, o \mid \mathcal{H}_{k+1}) = \mathbb{P}(x_{k+1} \mid \mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1}) \cdot \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1}) \tag{3.43}$$

without sampling the variables from scratch, ideally keeping samples of $\mathcal{X}_{0:k}, o$ from time step k, and only sampling incrementally $x_{k+1}^{(i)} \sim \mathbb{P}(\cdot \mid \mathcal{X}_{0:k}^{(i)}, o^{(i)}, \mathcal{H}_{k+1})$. Since generally $\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1}) \neq \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$ a possible approach to sample reuse is to weight incremental samples of trajectory $\mathcal{X}_{0:k+1}$ by the ratio $\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)/\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1})$ when approximating the integral from Eq. (3.42) at step $k+1$, an idea known as

importance sampling. Under this approach, full posterior samples only need to be generated to replace incremental samples with low weights, allowing reuse of samples and calculations, see e.g. Farhi and Indelman [28].

3.3.4.3 Discussion (Computational Aspects)

The method as presented in the previous section allows efficient computation with respect to model uncertainty offering a recursive formulation for model uncertainty samples, with marginalization at the current time step approximated with samples Eq. (3.24), and past time steps marginalized out analytically Eq. (3.40). However, formulation for localization uncertainty is not recursive due to the batch sampling of entire robot trajectory and object pose in Eq. (3.20) required there to account for correlations with past time steps. In the previous clause Sec. 3.3.4.2 we mentioned a possible approach to allow recursive computation with respect to localization samples as well, rendering the method fully incremental, which could be addressed in future research. In practice (and even for fully incremental computations) there is a trade-off between runtime and accurate treatment of uncertainty. One approach is to not address uncertainty at all, equivalent to using a single sample (e.g. max-likelihood). Another possibility is to neglect correlations to some past time steps. Both can be considered heuristics in the general scheme described above.

3.4 Results

We evaluate our method in a MATLAB simulation using synthetically generated classifier measurements and in a simulated 3D environment using Unreal Engine (UE) with UnrealCV (Qiu et al. [102]), with semantic measurements obtained from a CaffeNet classifier (Jia et al. [48]). In both settings, we split our evaluation into scenarios of model uncertainty and of localization uncertainty, comparing performance to baseline methods, as detailed in the next section. To validate our contributions in a realistic setting, we additionally present an evaluation using a real-world dataset, Active Vision Dataset (Ammirato et al. [1]), using AlexNet (Krizhevsky, Sutskever, and Hinton [63]) for semantic measurements.

In all settings, GP models (using Scikit-Learn (Pedregosa et al. [98])) are fit offline for each of the candidate classes. Following Eq. (3.17), components of the classification vector output by the classifier are considered independent, amounting to a separate GP per component. Another simplifying assumption is that object orientation relative to robot is known, following Sec. 3.2.3.

3.4.1 Compared Approaches and Performance Metrics

We compare the results of three methods. Ours we denote `Model with Uncertainty`, which takes into account spatial correlations, as well as uncertainty in pose and classifier

model uncertainty. The second is `Model Based`, similar to the method described by Teacy et al. [126] but with GP defined as in Eq. (3.13) (and Rasmussen and Williams [104]), which takes into account spatial correlation, but not localization nor model uncertainty The third is `Simple Bayes`, which directly uses the classifier scores and assumes spatial independence between observations, as in e.g. Patten et al. [96].

We compare the methods above with relation to the following metrics: (i) probability of ground-truth class; (ii) mean squared detection error; and (iii) most-likely-to-ground-truth ratio. The mean squared detection error (MSDE) is defined as

$$MSDE \doteq \frac{1}{N_c} \sum_{c' \in \mathcal{C}} \left(\mathbb{1}_c(c') - \mathbb{P}(c' \mid \mathcal{H}) \right)^2 \tag{3.44}$$

Here c is the ground truth class and $\mathbb{1}_c(c')$ is 1 if $c = c'$ and 0 otherwise. This measure was also used in Teacy et al. [126].

The most-likely-to-ground-truth ratio (MGR) is defined as

$$MGR \doteq \frac{\arg\max_{c'} \ \mathbb{P}(c' \mid \mathcal{H})}{\mathbb{P}(c \mid \mathcal{H})} \tag{3.45}$$

for ground truth class c. Roughly, this measure penalizes high confidence in the wrong class. In a way it "demands" ground truth class to be most (possibly, equally) likely. Finally, as a sanity check, we also present values of the "correct ratio" (CR), defined as

$$CR \doteq \mathbb{E}_{\mathcal{H}} \left\{ \mathbb{1}_c \left(\arg\max_{c'} \mathbb{P}(c' \mid \mathcal{H}) \right) \right\} \approx \frac{1}{N} \sum_{\mathcal{H}} \mathbb{1}_c \left(\arg\max_{c'} \mathbb{P}(c' \mid \mathcal{H}) \right), \tag{3.46}$$

with $\mathbb{1}$ the indicator function as above, N the number of averaged terms and with the proposal distribution of $\mathcal{H}$ defined according to reported context. This defines the CR as the ratio of correct classifications, considering a classification result vector "correct" if it assigns the highest probability to the ground truth class, i.e. if the ground truth class is the most likely. Below we report *final CR*, i.e. where $\mathcal{H}$ includes all measurements from the entire trajectory, and *overall CR*, where we also average over measurement history at intermediate time steps along trajectories, to capture the method's behavior over time.

We now proceed to detail the experiments and the results.

3.4.2 MATLAB simulation results

We present experimental results for a MATLAB simulation, in which synthetic classifier measurements are generated using the GP model of the ground truth class, along a predetermined track. Synthetic measurements are generated according to the procedure detailed in Alg. 3.2. The class inference algorithm needs to fuse these measurements into a posterior over classes, essentially identifying which of the known GP models is the more likely origin of the measurements. We study robustness of our algorithm to

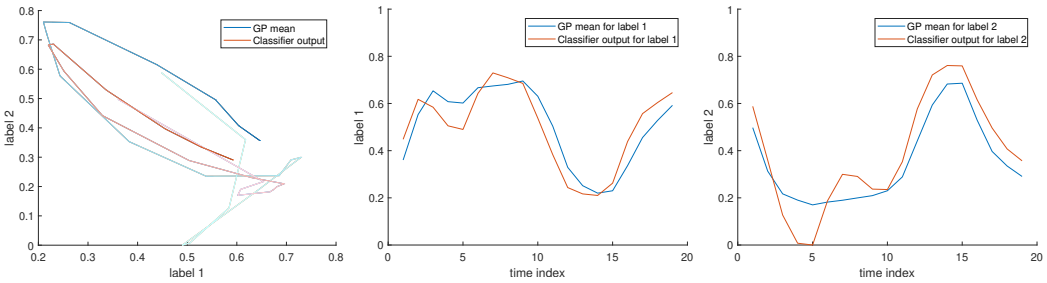


Figure 3.2: Scenario with model uncertainty (no localization errors). Standard deviation for the noised observation was chosen in $[0, 0.3]$. Left: the ground truth class GP model mean and simulated (noised) classifier measurement over robot trajectory, plots of response for 1st label against response for 2nd label. More intense color corresponds to later time index. Center and right: first and second components over time indices, respectively.

model and localization uncertainty, and compare it to the state of the art.

3.4.3 Simulation Experiments

Statistics for the three algorithms have been collected for several scenarios over realizations of simulated classification. In each scenario, GP models were created for three classes, by manually specifying the classifier response for chosen relative locations around the origin (i.e. locations assumed to be in object-centered coordinates) in the 2D plane, see Fig. 3.1a. Note that GP model for a class describes classifier responses for *all* classes, (see Eq. (3.17) and Section 3.2.2).

During simulation, the robot moves along a pre-specified trajectory and observes a single object from different viewpoints, see Fig. 3.1b for an example trajectory. At each time step the algorithm receives new classifier measurements and updated pose belief (simulating localization obtained from a SLAM solution). Classifier measurements are generated using the GP model of a "ground truth" class (the simulation of measurements is detailed in the next subsections), which needs to be inferred by the algorithm using the measurements.

We next present results on two scenarios highlighting our main contributions.

3.4.3.1 Model Uncertainty Scenario

Model uncertainty expresses the reliability of the classifier output. High model uncertainty corresponds to situations where classifier input that is far from the training data, often due to an unfamiliar scene, object or viewpoint pictured, causes output that may be arbitrary. We simulate this with two steps, performed at each time instant: first, a nominal "true" measurement $s^{nominal}$ is generated from a GP model of the ground truth class. The level of model uncertainty σ_u^2 is selected at each time step uniformly between 0 and σ_{max}^2 (a parameter). It is then used as standard deviation of a Gaussian centered at the true measurement to generate a simulated noised measurement

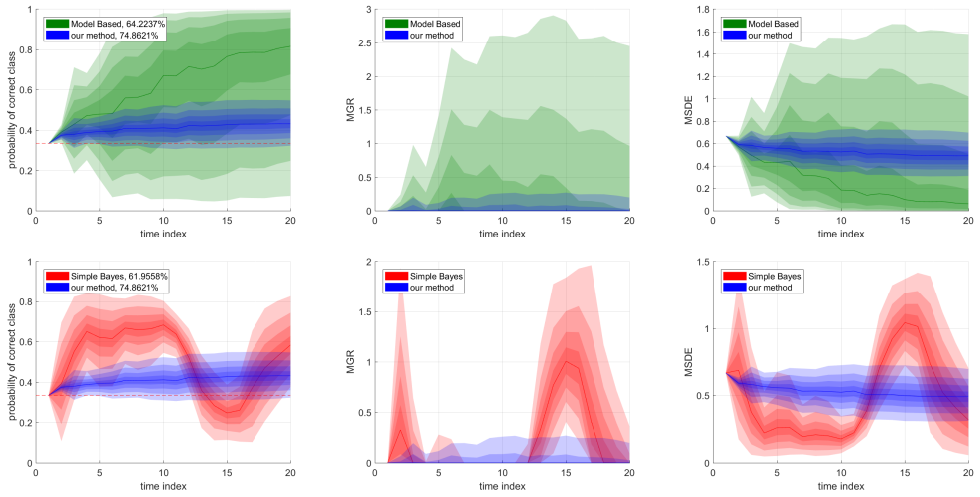


Figure 3.3: Model Uncertainty synthetic scenario. Left column: probability of correct class, middle: MGR, right: MSDE. Legend in the leftmost column shows percentage of time steps where most likely class was the correct one. Color patches denote percentiles of the respective methods, one step in lightness denotes 10% percentile step and median is plotted darkest, so that the patch around the median comprises values between the 40th and the 60th percentile, the next darkest between the 30th and 70th and so on. Dashed line denotes the uninformative prior (probability of 1/3 for label) Top row: comparison of our method to Model Based, bottom row: to Simple Bayes. While probability of correct class in our method rises slowly, it fluctuates significantly less over realizations, and the correct class is chosen more often. In both plots, Simple Bayes method performs poorly where an "inverse" measurement (see Section 3.4.3.1) exists in the model, around time index 15.

Algorithm 3.2 MATLAB simulation: procedure for simulating classifier outputs at step k

Require: $\mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}, \sigma_{max}^2, N_{samples}$
1: $s^{nominal} \sim \mathbb{P}(s \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)})$ See Eq. (3.31)
2: $\sigma_u^2 \sim Uni(0, \sigma_{max}^2)$ ▷ Choose uncertainty level
3: $s^{noised} \sim N(s^{nominal}, \sigma_u^2 I)$ ▷ Uncertain classification
4: $samples \leftarrow \emptyset$
5: **for** $N_{samples}$ times **do** ▷ Simulating dropout
6: $s \sim N(s^{noised}, \sigma_u^2 I)$
7: $samples \leftarrow samples \cup \{s\}$
8: **end for**
9: **return** $s^{nominal}, s^{noised}, samples$

s^{noised}. The `Model Based` and `Simple Bayes` algorithms receive s^{noised} as classification measurement and are not aware of the uncertainty. Our method receives samples (simulating outputs of several forward passes applying dropouts) drawn from a Gaussian distribution centered at s^{noised} with standard deviation σ_u^2. Alg. 3.2 summarizes this process.

First scenario shows the effects of considerable model uncertainty, with no localization errors (perfect localization). Fig. 3.2 shows plots of GP model of ground truth class and simulated classifier measurements (s^{noised}) over robot track (left) and per-component as a function of time (right). Fig. 3.3 shows the statistics described above (probability assigned to ground truth class and Eqs. (3.44-3.45)) along with percentiles (over scenario realizations) as patches of varying saturation, with a 10% step: median is plotted darkest, the patch around it contains the runs between 40th and 60th percentile, the next one between 30th and 70th, etc. The area above and below the plots contains the top and bottom 10% of the runs respectively. Top row shows comparison of our method (blue) to `Model Based` (green), bottom - to `Simple Bayes` (in red).

An immediate observation in comparison to `Model Based` (first row) is that our percentiles are more concentrated, which means that method results are more stable. For example, in more than 20% of the runs (bottom lightest patch and below), probability of correct class (left column) for `Model Based` in time step 15 is less than 0.2 (compared to more than 0.33 for ours). Indeed, in more than 20% of the runs the MGR (middle column) for `Model Based` at iteration 15 is higher than 1, which means that a wrong (most likely) class was assigned probability more than twice higher than the correct one, i.e. wrong class was chosen with high confidence. The MSDE plot displays similar behavior. In the bottom row, drop of accuracy of `Simple Bayes` around time step 15 is the result of an "inverse" measurement in the model, meaning that from a certain angle, classifier response suggests a different class (see for example in Fig. 3.1a). This illustrates well the difference from our method, which matches the entire sequence of measurements against a model, and thus can use also "inverse" measurements to classify correctly (on the downside, requiring a class model).

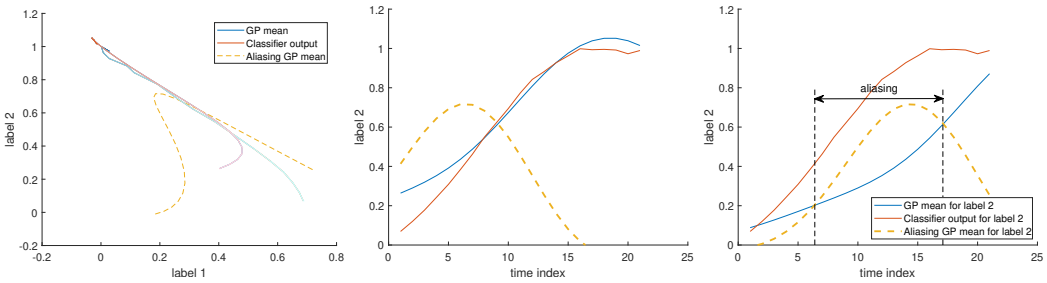


Figure 3.4: Scenario with localization bias in the x axis. Time index corresponds to x coordinate (robot motion is a straight line, in the direction of the x axis). Left: as before, simulated classifier output generated from GP of ground truth class. Center: we concentrate on responses for class 2 (2nd component of classification vectors). Classifier output (red) matches GP of ground truth class (in blue) at true position. Right: bias in the x axis means that classifier output is effectively compared to a shifted model, better matching GP of a wrong class (yellow). This leads to classification errors unless accounting for localization uncertainty.

3.4.3.2 Localization Uncertainty Scenario

In methods making use of spatial class models, localization errors may cause classification aliasing when acquired measurements correspond to the model of a wrong class, because of the spatial shift in the query. To exemplify this, in this scenario, we introduced (a constant) bias in easting coordinate (the robot moves eastward in a straight line), causing aliasing between models (with no model uncertainty). Consider Fig. 3.4. The left plot as before shows GP mean of the correct class model (blue) and classifier output over robot track (red). It also shows the GP mean of the model of a wrong class (yellow). In the center plot, classifier outputs for label 2 (red) compared without localization bias against the corresponding GP component of the ground truth class model (blue) show a clear match. After introducing a bias of -8 units in easting (right plot) classifier responses (red) are matched against shifted spatial models, making the wrong class (yellow) a more likely match until around time step 16, after which the the blue line can be matched correctly in spite of the shift. The effects of this on performance are shown in Fig. 3.5. While our method, aware of the localization uncertainty (standard deviation) accumulates classification evidence gracefully, the `Model Based` method infers the wrong class with high confidence (as can be seen in the MGR plot, center) peaking at around time step 15, after which disambiguating measurements start to arrive. In the bottom row of the same figure, `Simple Bayes` method performs well (closely following the line from Fig. 3.4), since classifier measurements are stable and not ambiguous (the aliasing happens when trying to match against the different models).

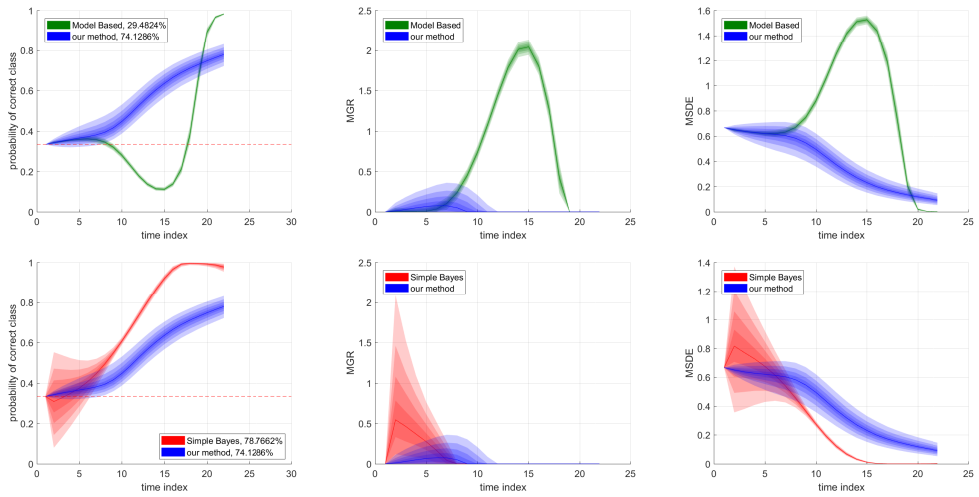


Figure 3.5: Localization uncertainty synthetic scenario. Left column: probability of correct class, middle: MGR, right: MSDE. Localization bias of -8 units in the x axis causes severe aliasing in Model Based method resulting in a wrong class being inferred with high confidence. Our method is aware of localization uncertainty of standard deviation 16, and is able to recover. Simple Bayes method does not experience aliasing, as it uses the raw measurements directly, rather than matching them to a model.

3.4.4 Evaluation in synthetic 3D environments

To study our approach in a more realistic setting when using a DL classifier, we created a set of 3D simulated environments using UnrealEngine. In this setting, classifier measurements are obtained by feeding rendered images of an object to a neural network classifier (CaffeNet), in contrast to the basic MATLAB simulation, where they were generated from a manually specified GP model.

For simplicity, we limit ourselves to planar environments, containing a single object, corresponding to solved data association. Camera moves at a constant height of 160cm, in object proximity - corresponding to object detection, in a way that the object occupies most of the frame - and facing it. We selected a set of objects for which 3D models were available, considering them as representative of the corresponding classes. For each object/class we fit a spatial (2D) GP model to classifier outputs for rendered viewpoints sampled over a grid. For simplicity, each view is directed towards the object.

Fig. 3.6 shows the GP models fit for each of the classes (chest, crate, desk), sampled over a 2D spatial grid centered on the object, as well as example images for corresponding viewpoints and raw classifications output by the classifier unit (as used for learning the GP models). The color of each pixel in the raster maps is calculated as the sum of class colors (blue for chest, orange for crate, green for desk), weighted by the classification vector predicted for the (object-centered) corresponding view.

We used classifier scores for the selected classes as features, although other choices are in principle possible, such as using scores for additional or fewer classes, or in fact any spatially varying feature.

We next show experiments using the above GP models for classification under model uncertainty, Sec. 3.4.4.1 and localization uncertainty, Sec. 3.4.4.2.

3.4.4.1 Model Uncertainty Experiments

In order to induce model uncertainty, we modified the crate 3D model by filling one of its sides with plain color. Note that while both the original and the modified crate were not part of classifier training set, we added color to induce an irregular appearance and elevated model uncertainty.

The spatial uncertainty map in Fig. 3.7a shows the standard deviation of classifier vectors obtained from MC dropout for the corresponding views, summed over components (corresponding to the classes of interest). For all views camera is at a constant height, at each point oriented towards the object center. Data at the center of the map is missing since views too close to the object become uninformative. As expected, the plot generally exhibits elevated uncertainty at points corresponding to views of the colored crate side (right side of the map). Uncertainty values also tend to be higher around the map edges (especially left center and corners), possibly related to larger portions of the crate's environment, which is different from training (natural) images, in the frame. Relatively low uncertainty values in the center right area are due to low

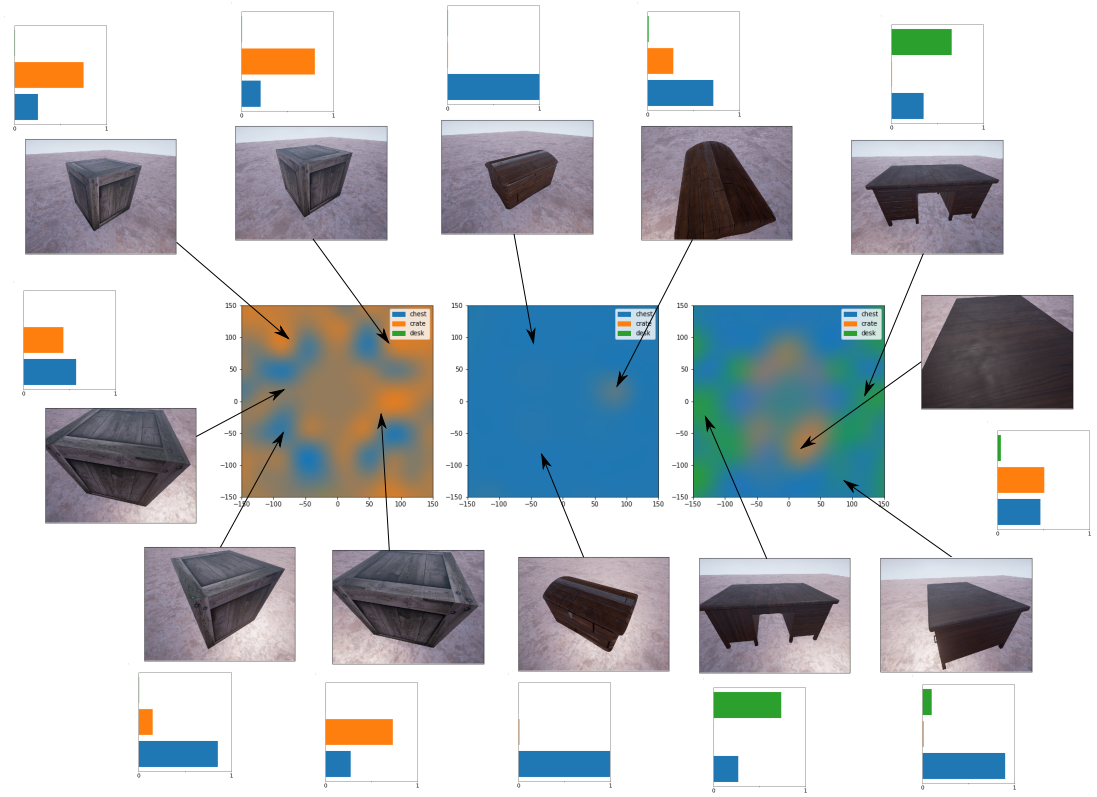
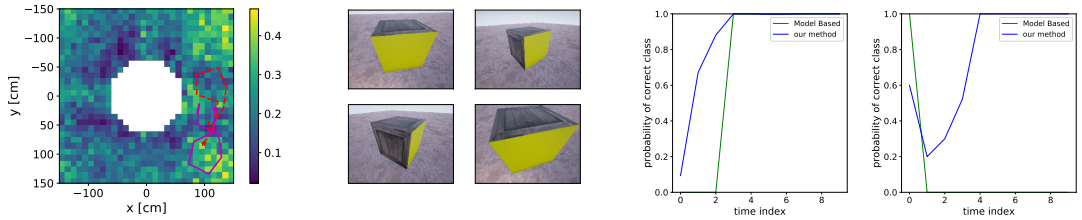


Figure 3.6: Spatial GP models learned for the classes of interest and example views. Raster maps show spatial predicted classifications output by the black box classifier unit when observing an object of the corresponding class. Left raster shows predictions of GP model learned for a crate object, center - model for chest, right - model for desk. Each pixel corresponds to a classification vector predicted by the GP model. Class GP models are fit to "raw" classifier measurements for varying relative viewpoints. Thus, a strong orange component indicates that the element corresponding to the crate class in the classification vector output by the classifier is close to 1 for that viewpoint. For each model representative views are displayed along with classification vectors output by the classifier unit (bar plots). Note the difference between "raw" chest, crate and desk classifications output by the classifier unit, to chest, crate and desk class models which in general need not be related (i.e. to construct class models features other than explicit class predictions can be used). Learned models indicate that chosen crate object is for some viewpoints mis-classified as chest by the classifier unit, chosen chest object is generally classified correctly from all viewpoints, chosen desk object may be classified as each of the classes, depending on viewpoint, motivating the use of class models over raw classification scores.



(a) Model uncertainty map

(b) Example frames for the easy track

(c) Classification of easy track - ma- genta

(d) Classification of hard track - red

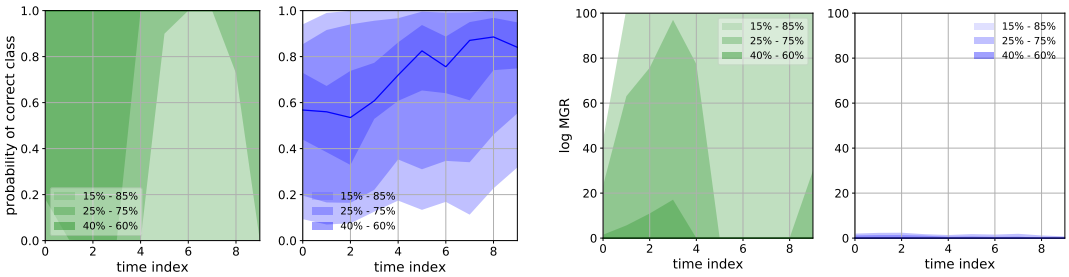
Figure 3.7: Setting of the model uncertainty experiments. (a) Views of the colored side exhibit higher model uncertainty. Map shows standard deviation of classifier vectors obtained from MC dropout, summed over components. Two Example random tracks are plotted in red "hard" and magenta "easy". (b) Example frames along the easy track. (c) and (d) Probability of correct class over time steps along the track for easy and hard track respectively, using Model Based (green) and our method (blue).

values of components of the classification vector corresponding to classes of interest, which result in low variance.

We generated 300 random tracks in the area exhibiting elevated uncertainty values (right side of the map), rendering 10 viewpoints along each track. Two example tracks are shown in Fig. 3.7a, example viewpoints are shown in Fig. 3.7b. As before, viewpoints are centered on the object. For each frame, we estimated model uncertainty using 10 forward passes with MC dropout, although any other method can be used in the context of our Bayesian fusion scheme. In all experiments, class models, localization and classification measurements input to the compared methods are the same, but our method is aware of model uncertainty by using all MC-Dropout samples, while compared method is input only the samples average. Fig. 3.7c and Fig. 3.7d compare classification of the two example tracks using `Model Based` and our method showing the evolution of score assigned to ground truth class (crate) over the time steps / track points. In the "hard" track (shown in red in Fig. 3.7a) the `ModelBased` method fails while ours is able to recover.

Fig. 3.8 provides a statistical comparison of Bayesian classification using our method (in blue) against the `Model Based` method (in green). As before, color patches denote respective percentiles per-timestep over the random tracks. Plots show the evolution over time steps of probability assigned to ground truth class by each of the methods (left column) and the MGR (right column).

For both methods results vary over tracks. For the `ModelBased` method, median score for ground truth class was 1. However, for every time step in 20% of the tracks the ground truth class score is exactly 0 (threshold = 10^{-10}). For around 25% of the tracks, ground truth class score remains zero after measurements from the entire trajectory have been incorporated, compared to none for our method. Conversely, in our method, percentiles are rising steadily, corresponding to gradual accumulation of



(a) (b)

Figure 3.8: Model uncertainty UE experiments. Blue: our method. Green: Model Based. Left (Fig. 3.8a): Probability assigned to ground truth class. Right (Fig. 3.8b): MGR, Eq. (3.45).

information, and are less spread out, in particular - not approaching score 0, which corresponds to a confident misclassification. This can be equally seen in the (log) MGR plot Fig. 3.8b, which for the `Model Based` method in up to 40% of the tracks reaches around 20 (meaning e^{20} times higher probability assigned to wrong class), and remains so at the end of 25% of the tracks (i.e. after incorporating data from the entire track), while for our method final MGR is 0 at 75% and 5.05 at 99%, implying reduced confident mis-classifications and thus relative resilience to model uncertainty.

Table 3.1 summarizes MGR (75th and 95th percentile) and correct most likely class rate, that is - ratio of tracks for which the ground truth class was the most likely. "Final" column presents statistics over tracks at the final time step, that is - for each track, after information from the entire track has been incorporated. "Overall" column presents statistics over all time steps from all tracks. Correct rates for both methods are similar, while MGR is considerably lower for our method, consistent with the plots.

This behavior also corresponds to results presented in Fig. 3.3 for the MATLAB simulation. While ground truth class scores for our method are generally lower than with `Model Based` method, reflecting the classification uncertainty, they are more stable under model uncertainty, and information is accumulated gracefully, as opposed to making overconfident mistakes.

Table 3.1: Model uncertainty experiment. Correct rate is ratio of classifications where the ground truth class was the most likely. Statistics are over tracks given measurements from the entire track ("final" columns, ratio of tracks) and given measurements up to every time step ("overall" columns, ratio of time steps from all tracks). Our method achieves roughly the same correct classification rates, but makes less confident mis-classifications, as reflected by the significantly smaller MGR.

	Correct rate - final	Correct rate - overall	MGR - final		MGR - overall	
			75%	95%	75%	95%
Model Based	0.74	**0.67**	29.98	629.2	44.67	519.30
our method	**0.76**	0.65	**0**	**2.46**	**0.85**	**2.99**

3.4.4.2 Localization Uncertainty Experiments

To assess performance under a noisy relative pose estimate returned by SLAM / perception algorithm, we manually specified test tracks of varying classification difficulty (described below). Test tracks were then corrupted by perturbing each of the viewpoints along the track with an i.i.d. (both among tracks and among each track's viewpoints) Gaussian noise $N(0, \sigma)$, coordinate-wise. Each corrupted track thus obtained corresponds to one realization of a noisy robot trajectory and object localization estimate (while the ground truth remains the original test track before application of noise). Corrupted tracks were fed to the compared algorithms as the position estimate, our method additionally input σ as the estimate's uncertainty. In our experiments we collected results over realizations of corrupting noise, statistically exploring the effect of perception errors on classification of the chosen tracks.

Fig. 3.9a shows the three test (ground truth) tracks over which statistics are presented in the below sections. The background shows the "aliasing map" in classification of a desk object under a (one particular) localization bias of +10cm in the x axis, using the `Model Based` method (i.e., not uncertainty-aware). Each pixel is obtained by weighting the colors corresponding to the different classes (blue for chest, orange for crate, green for desk) with the classification vector obtained for the given viewpoint (as before, the object is at the coordinate origin). Thus, pixels with a strong orange component correspond to views for which the desk object is incorrectly classified as crate under the given localization bias, due to the raw classifier measurements for this view better matching the crate model.

Thus, the hard track (in red) passes (exclusively) through viewpoints which are incorrectly classified by the `Model Based` algorithm for this particular localization bias. The moderate track (black) passes through viewpoints that are classified correctly, but other points in their vicinity are not. The easy track (magenta) passes away from incorrectly classified viewpoints. Note that while we selected and described the ground truth tracks with respect to their difficulty under one particular value of localization bias, it turns out that they also exhibit the described behavior for sampled random values of localization bias, generally different ones for each viewpoint along the track, as can be seen in statistical results described below.

The situation leading to classification aliasing is exemplified in Fig. 3.9b and Fig. 3.9c for classification of viewpoints along the hard track (red). To reduce clutter here we focus on the "chest" component of the raw classification measurement vector (dashed line), although in practice behavior is determined by all vector components at once, as shown in Fig. 3.10. In Fig. 3.9b, the "chest" component of the raw classification measurement vector (dashed) reasonably matches the desk class model predictions (plain line). While crate model appears to match the measurements better, in practice the other vector components provide disambiguation, as described in detail in Fig. 3.10. In Fig. 3.9c the same measurements (again, dashed) are compared with class models at

a wrong location, due to the localization error, matching better the crate class model than the desk model, causing confident erroneous classification, for this particular bias, if not accounting for uncertainty. Fig. 3.10 visualizes the entire classification vectors at each time step, providing insight into why there is disambiguation in the former (no localization error) but not the latter (with localization error) case.

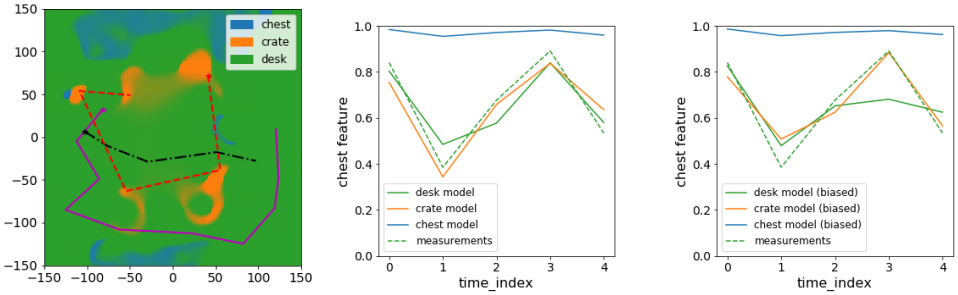
Note that this effect closely corresponds to the MATLAB simulation, in particular what is presented in Fig. 3.4. Note however, that the experiments presented in the current section are different from the synthetic localization uncertainty experiments in Sec. 3.4.3.2, where tracks, localization bias and class models were engineered to illustrate contributions, and presented statistics were over realizations of the synthetic measurement models (GPs). In contrast, here classifier measurements both for training of class models and at test time are obtained from a Deep Neural Network classifier operating on rendered images, and statistics are presented over errors in perceived localization w.r.t. given ground-truth tracks.

3.4.4.2.1 Given Level of Uncertainty As described in Sec. 3.4.3.2, and Sec. 3.4.4.2, localization error causes class aliasing when the sequence of class measurements obtained over a track better fits a model other than the ground truth class when matched at the erroneous estimate. The effect of marginalization over pose uncertainty in Eq. (3.19) is averaging term (a) over the belief, term (b), i.e. matching obtained measurements against the models as above at different values of shift w.r.t. the localization estimate, according to the belief posterior distribution.

This averaging may mitigate class aliasing if the ground truth class is matched for at least a portion of the shift values under the belief. On the other hand, this averaging may reduce confidence of classification even when there is no aliasing due to localization errors, if aliasing happens for some portion of averaged shift values.

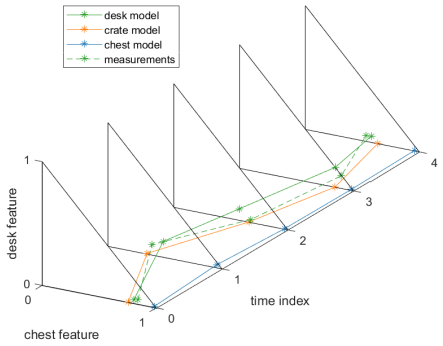
In Fig. 3.11 we explore statistics for a given level of uncertainty $\sigma = 10\ cm$ over random additive noise as described above (i.e. additive localization errors drawn from $N(0, \sigma)$ for each viewpoint) for the tracks from Fig. 3.9a, which turn out to exemplify both cases. The figure shows percentiles of probability assigned to ground truth class by the two methods (left two columns) and the MGR (right two columns) for each time step along the track for the hard (top row), moderate (middle row) and easy (bottom row) ground truth tracks, statistics taken over additive localization noise as described above.

In the hard (Fig. 3.11a, Fig. 3.11b) and moderate (Fig. 3.11c and Fig. 3.11d) cases, our method mitigates aliasing, significantly reducing classification variability. The MGR plots indicate that mis-classifications, when they occur, are accompanied by uncertainty in the output classification, i.e. the correct class being assigned a non-negligible weight - note that the lightest patch indicates maximum values (i.e. 100th percentile), with lower percentiles much smaller. In contrast, confident mis-classifications occur in significant portions of the cases for the `Model Based` method,

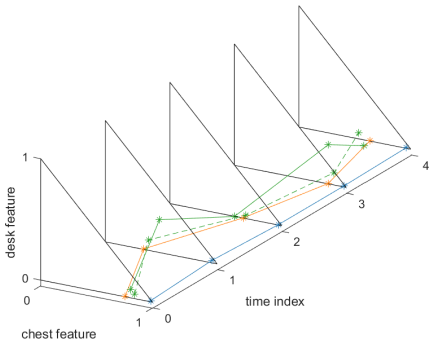


(a) **(b)** **(c)**

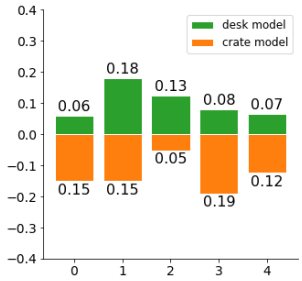
Figure 3.9: **Left**: tracks of varying classification complexity, used for statistics presented in Fig. 3.11 against aliasing map for desk class, for bias of +10 cm in the x axis. The aliasing map is obtained by comparing per-pixel (every $\sim$ 5 cm) predictions from the desk GP model against shifted class models (see Fig. 3.6 and Fig. 3.4), corresponding to a localization bias, or a single realization in the localization uncertainty experiments. Magenta track (Fig. 3.11e and Fig. 3.11f, "easy") steers clear of aliasing areas making correct classification easy. Red track (Fig. 3.11a and Fig. 3.11b, "hard") passes through aliasing areas, causing classification errors in the Model Based method and high uncertainty in our method. Black track (Fig. 3.11c and Fig. 3.11d, "moderate") does not pass through aliasing areas for the particular bias pictured above, but does for other bias realizations, as can be seen in the classification statistics. **Center and Right**: class aliasing along the red track when viewing a "desk" object, equivalent of Fig. 3.4 for measurements and class models based on DL classifier output. **Center** plot shows the evolution of "chest" feature component in raw classification vectors (dashed line) against model predictions when localization estimate is correct (i.e. unlike the aliasing map shown in the left plot). Measurements curve appears to match both crate and desk class models (orange and green respectively), in practice inferred classification is mostly correct ("desk") as disambiguation is provided by other components (not shown in the plot). Fig. 3.10 details the classification process. **Right** plot shows the same measurements curve (dashed) against models sampled with an error of +10 cm in the x axis (corresponding to the aliasing map in the left plot). Measurements match the crate model, leading to inference of wrong classification (see details in Fig. 3.10).



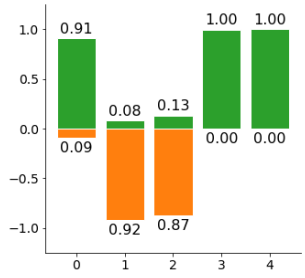
(a) Measurements and model predictions for correct localization



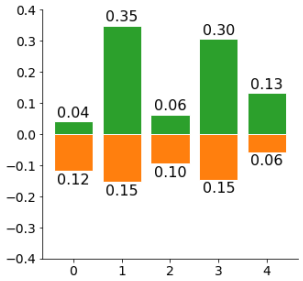
(b) Measurements and model predictions under localization error



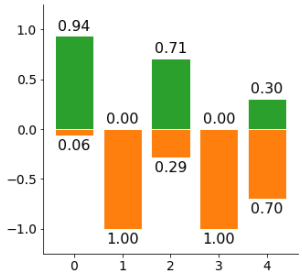
(c) Difference norm



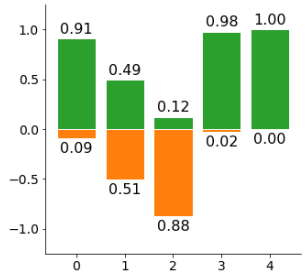
(d) Classification (single measurement)



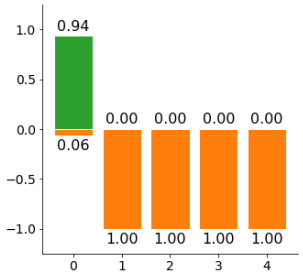
(e) Difference norm



(f) Classification (single measurement)



(g) Classification posterior (joint)



(h) Classification posterior (joint)

Figure 3.10: Classification of a desk object over the red track from Fig. 3.9 under correct localization (left column) and with localization bias of +10 cm in the x axis (right column), corresponding to the scenario from Fig. 3.9. Top row: obtained measurements and class models samples over the track time steps. For every time index, raw classification vector (vertex on the dashed line) and model prediction mean for each candidate class are shown as points in the 2D simplex, corresponding to 3-coordinate classification vectors. Classification inference roughly corresponds to selecting the model curve which best fits the measurements. Fig. 3.10c and Fig. 3.10e show the Euclidean distance of measurement vector from prediction mean for desk and crate models, for each time index. The smaller the distance of measurement to model prediction, the more likely is the corresponding class. Fig. 3.10d and Fig. 3.10f show the corresponding class probability (i.e. normalized class likelihood) using a single measurement Eq. (3.5), whereas Fig. 3.10g and Fig. 3.10h show classification at each time index using all measurements obtained up to that time Eq. (3.7). While correct model (desk) does not perfectly match all obtained measurements (particularly at time indexes 1 and 2), eventually disambiguating measurements arrive and correct class is inferred in the left column. Introducing a localization error (right column) further brings measurements relatively closer to wrong model, causing the wrong class to be inferred Fig. 3.10h.

	Easy				Moderate				Hard			
	Correct rate		MGR (95%)		Correct rate		MGR (95%)		Correct rate		MGR (95%)	
	final	overall	final	overall	final	overall	final	overall	final	overall	final	overall
Model Based	1.0	0.97	0	0	0.958	0.90	0	8.8	0.66	0.64	48.6	29.0
our method	1.0	**0.974**	0	0	**1.0**	**0.946**	0	**0.007**	**0.72**	**0.70**	**1.1**	**1.26**

Table 3.2: Localization uncertainty scenarios - ratio of correct classifications, i.e. classifications where the ground truth class was the most likely given measurements from the entire track ("final" columns, ratio of tracks) and given measurements up to every time step ("overall" columns, ratio of time steps from all tracks). As in the model-uncertainty experiments, uncertainty-awareness results in correct classification rates similar to the non-aware case, but significantly reducing rate of confident mis-classifications, see MGR in Fig. 3.11.

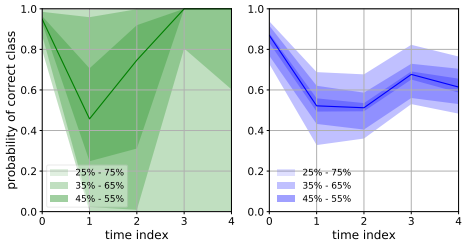
with the median MGR higher than 0 for the hard track. These results are equally supported by Table 3.2, which lists correct most likely class rates over all time steps ("Overall") and last time step ("Final") and 95th percentile of MGR for statistics with each of the test tracks. Our method reaches a similar, if slightly higher, correct rates as `Model Based`, with a much lower MGR.

In the easy case (Fig. 3.11e and Fig. 3.11f) our method more hesitantly reaches the same correct classification assigning lower confidence to the measurements due to uncertainty, with an identical rate of correct classifications overall.

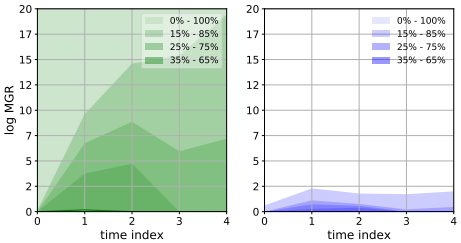
3.4.4.2.2 Sensitivity Experiments We explore the sensitivity of localization uncertainty aware classification to the level of localization uncertainty. To this end, we repeat the experiment from Sec. 3.4.4.2.1 for values of σ between 1 cm and 20 cm. Note that as the actual localization errors are drawn from a Gaussian distribution centered at 0, a higher value of σ does not imply that localization errors are necessarily higher, although higher errors do become more likely. As before, in each experiment our method is input σ in addition to the noisy localization estimate as measure of the localization uncertainty.

Fig. 3.12 shows percentiles of probability assigned to ground truth class at the end of each track ("final") - left two columns, and at each time step using measurements up to that time step ("overall") - right two columns, both function of uncertainty level, for the hard track (top row), moderate track (middle row) and easy track (bottom row). Fig. 3.13 shows corresponding percentiles of the MGR.

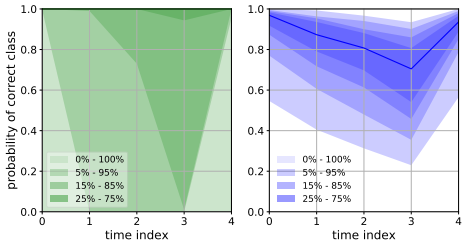
The plots generally exhibit behavior similar to the one observed in Sec. 3.4.4.2.1 for $\sigma = 10$ cm, this time for a range of σ values. As generally confidence of classification (probability assigned to ground truth class) is lower for our method, its percentiles are more concentrated and so results are more stable across the runs. Our method is also significantly less likely to assign a high probability to a wrong class, as demonstrated by the MGR plots Fig. 3.13.



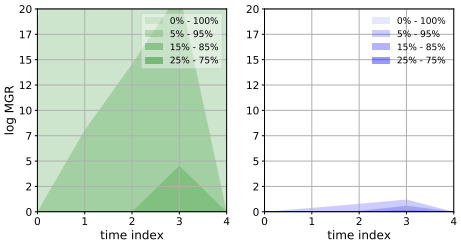
(a) Ground truth probability (hard case).



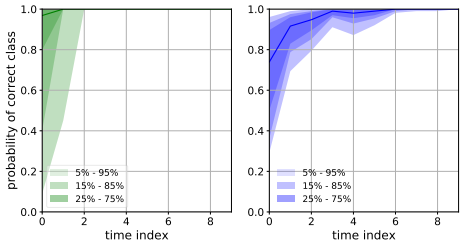
(b) Log MGR (hard case).



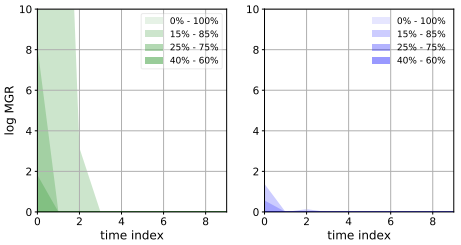
(c) Ground truth probability (moderate case).



(d) Log MGR (moderate case).



(e) Ground truth probability (easy case).



(f) Log MGR (easy case).

Figure 3.11: Localization uncertainty UE experiments, for set level of uncertainty $\sigma = 10cm$. Localization estimate for both methods produced by corrupting test tracks with with an i.i.d. (both among tracks and among each track's viewpoints) Gaussian noise $N(0, \sigma)$, coordinate-wise. Our method is additionally input σ as the localization uncertainty measure. Blue: our method. Green: Model Based. Left column: Probability assigned to ground truth class. Right column: MGR, Eq. (3.45). Top row: hard track (Fig. 3.9), middle row: moderate track, bottom row: easy track.

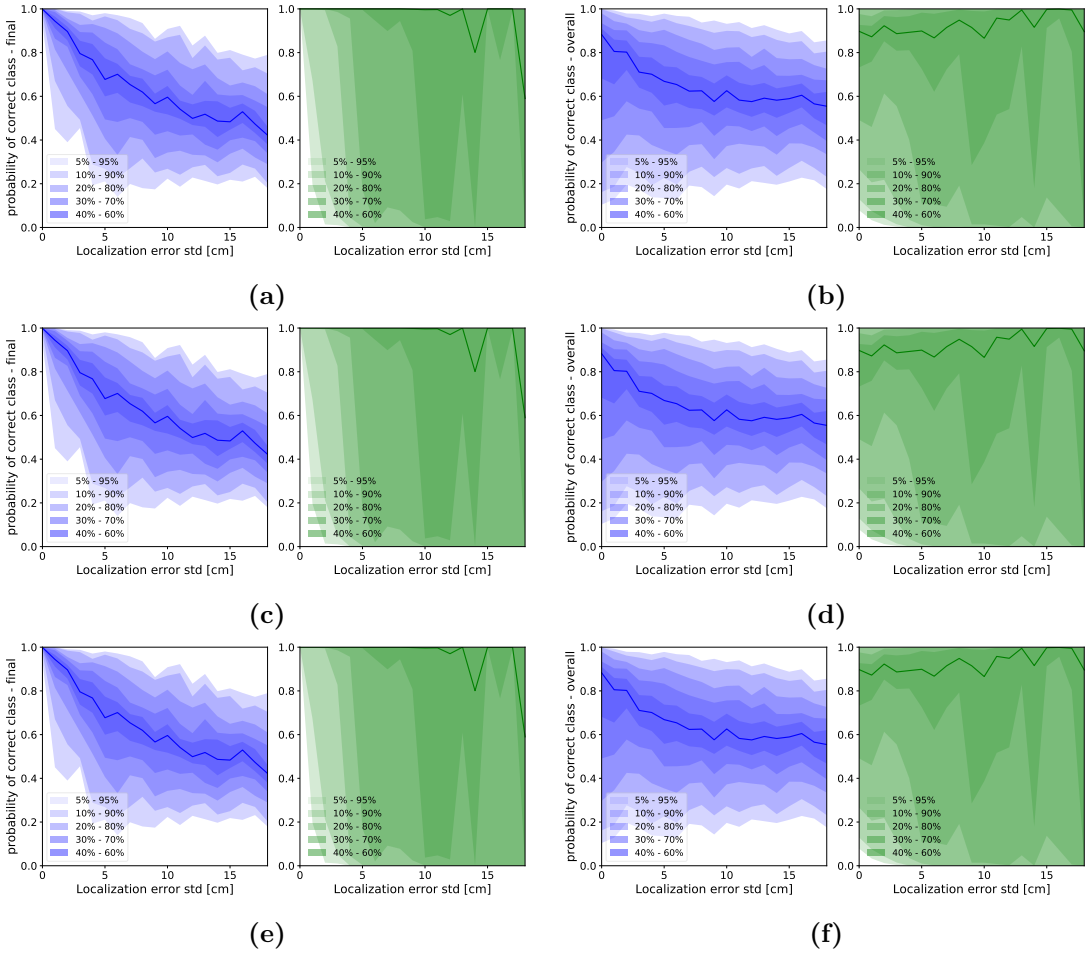


Figure 3.12: Localization uncertainty sensitivity UE experiments, results for varying levels of localization error (σ). Probability assigned to ground truth class. Blue: our method. Green: Model Based. Left column: statistics over final time index. Right column: statistics over all time steps (see Table 3.2 and Eq. (3.46)). Top row: hard track, middle row: moderate track, bottom row: easy track.

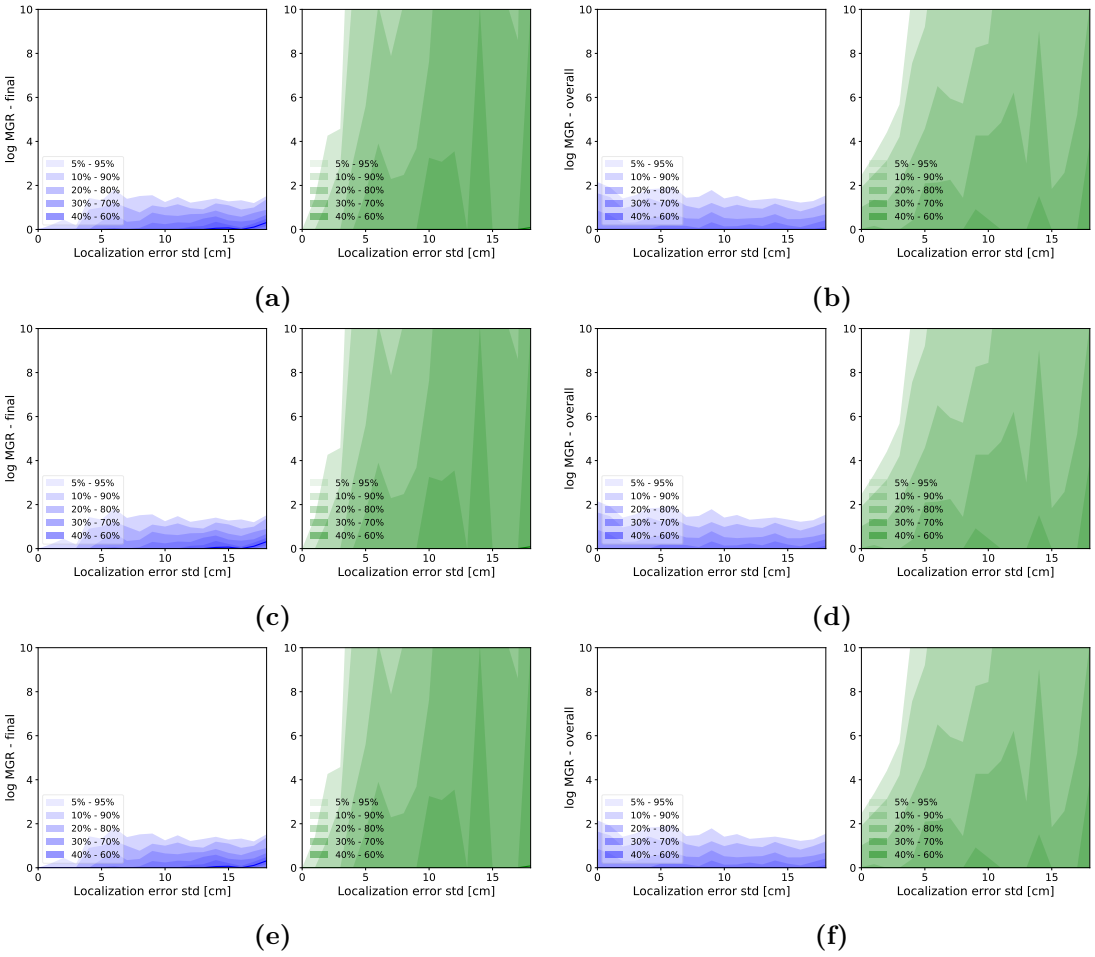
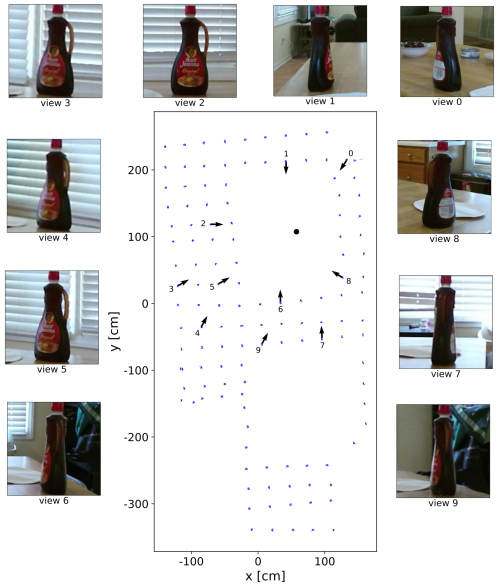


Figure 3.13: Localization uncertainty sensitivity UE experiments, results for varying levels of localization error (σ). MGR, Eq. (3.45). Blue: our method. Green: Model Based. Left column: statistics over final time index. Right column: statistics over all time steps (see Table 3.2 and Eq. (3.46)). Top row: hard track, middle row: moderate track, bottom row: easy track.



(a) Layout of AVD scene Home_001_1 with all scene viewpoints shown in blue, and chosen views numbered. Around appear cropped object detections for the corresponding views. Black dot denotes (estimated) object location. Full frames for views 0 and 4 are given in Fig. 3.14b and Fig. 3.14c respectively.

(b) Frame at index 0 (from Fig. 3.14a) with bounding box corresponding to object of interest

(c) Frame at index 4 (from Fig. 3.14a) with bounding box corresponding to object of interest (view opposite to Fig. 3.14b

Figure 3.14: AVD evaluation setting. Fig. 3.14a shows the scene layout, viewpoints chosen for evaluation and corresponding object detections. Fig. 3.14b and Fig. 3.14c show two example views of the scene.

3.4.5 Evaluation with real-world imagery

We present results validating our contributions w.r.t. localization uncertainty in a real-world environment with the Active Vision Dataset (AVD) (Ammirato et al. [1]), using data from the Berkeley Instance Recognition Dataset (BigBIRD, Singh et al. [111]) to learn GP models. Similarly to before, raw classifier measurements are provided by an AlexNet neural network classifier (Krizhevsky, Sutskever, and Hinton [63]) implementation in PyTorch (Paszke et al. [94]).

The AVD comprises a set of indoor scenes, each one with images taken over a grid of spatial locations and camera orientations, allowing to simulate real-world trajectories. Each scene contains a set of BigBIRD object instances, with bounding boxes provided for each object in the set for each containing frame.

For our initial evaluation, we focus on classification of an "aunt jemima original syrup" instance in scene `Home_001_1`. Fig. 3.14 shows the scene layout. On the left (Fig. 3.14a) all viewpoints in the scene are denoted as blue dots. A subset of viewpoints chosen to demonstrate contributions is numbered with time step indexes as appear in later plots, arrows showing the ground truth camera orientations, pointed roughly

towards the object, which is denoted by a large black dot. On the right, the frame at time 0 (top Fig. 3.14b) and time 4 (bottom Fig. 3.14c) are overlayed with object bounding boxes as provided in the dataset.

3.4.5.1 Learning spatial GP Models with BigBIRD Data

The BigBIRD dataset (Singh et al. [111]) provides images of a set of "object instances" taken from varied angles. Images were taken using a set of static cameras with imaged objects placed on a rotating table Fig. 3.15. Ground truth poses for the cameras and the rotating table are provided, as well as a binary segmentation mask for each frame marking pixels belonging to the object.

We learn a spatial GP model for each object in the BigBIRD dataset, treating it as a separate class and using the provided segmentation masks to compute bounding boxes (Fig. 3.16 top row). These are then cropped and fed to an AlexNet to produce a set of classification / feature vectors, associated to corresponding camera poses relative to the rotating table, which can be directly computed from ground truth data. For simplicity, we limit ourselves to 2D models, implying classification of upright objects - which is enough for this evaluation. Accordingly, we only use features and corresponding relative poses computed for images from the bottom ring of relative camera poses in Fig. 3.15c. We fit GP models (Eq. (3.7)) to capture the spatial variation of chosen (elaborated below) classification vector components.

In Fig. 3.16 the second row from the top shows the GP model mean learned for each instance, as sampled at training set points, for a chosen subset of 4 classification vector components. As before, each component is assigned a color (as listed in the legend, along with component indexes), the color of a pixel in the raster is calculated by weighting the component colors with the corresponding values predicted by the model. Similar colors in raster plots correspond to similar raw classification vectors (components), thus indicating aliased (single) views and motivating both the accounting for (relative) localization uncertainty and the fusion of measurements from multiple viewpoints for disambiguation.

The third row from the top shows rasters of the same GP models, this time over the unit square, similar to Fig. 3.6. The bottom row shows rasters of GP models learned for a different set of components (again, elaborated below). BigBIRD object instances generally do not have an adequate corresponding ImageNet "ground truth" class, yet - as demonstrated by our evaluation - scores of an ImageNet-trained classifier can be directly re-used by our method as features to correctly classify an object fusing measurements from multiple views.

3.4.5.1.1 Choice of features: The AlexNet classifier outputs scores for 1000 ImageNet (Russakovsky et al. [108]) categories. While using the entire classification vectors to learn GP models is possible (especially under the independent components approx-

imation Sec. 3.4), it is inefficient Sec. 3.3.4.1, and more so as the vast majority of components are negligible for most objects. We thus choose to model 4 raw classification components with high (in particular, not negligible) scores across views for the object instances we used as classification candidates. The learned GP models are shown in the second and third rows of Fig. 3.16. Other choices are possible (e.g. bottom row of Fig. 3.16). The choice of good features is a research topic in its own right and is beyond the scope of this work.

3.4.5.1.2 GP parametrization / coordinates: As BigBIRD data is only available on a ring (semi-sphere in 3D), images taken from front object views, we parametrize the GP models Eq. (3.7) with (uncertain) 2D location relative to the object, projecting the (relative) coordinates to the unit circle, that is, for the sake of this explanation denoting robot 2D relative pose to object o as $\mathcal{X}_i^{(rel)} \doteq (t_{o\to i}, R_i)$ (i.e. $t^o_{o\to i}$ is robot 2D location w.r.t. the object), we approximate

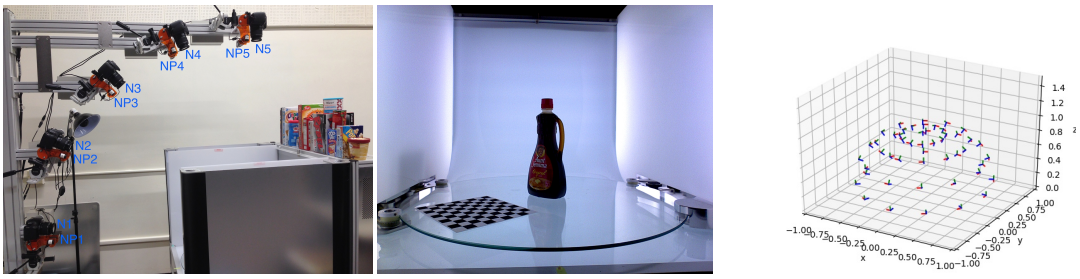
$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, \mathcal{X}^{(rel)}_{0:k}) \approx \mathbb{P}(\mathcal{S}_{0:k} \mid c, t^o_{o\to i}/\|t^o_{o\to i}\|). \tag{3.47}$$

Note that the above is an approximation, since in general classifier scores may be affected by distance from the object impacting e.g. the ability to distinguish detail due to finite resolution of input images. Additionally, an object viewed from farther away is more likely to be partially occluded and affected by limitations of the object detector, all of which could in principle be incorporated into the class model (but are out of scope of the present work). While distance from object could be simulated by purportedly blurring and sub-sampling input images, we did not go as far in the present initial evaluation.

Finally, it makes sense to consider parametrizing the class models with polar coordinates, which could allow to better capture different co-variation levels in the radial and tangential directions (partly visible in Fig. 3.6), to reduce calculations (e.g. in Eq. (3.15), Eq. (3.16)) by reducing the dimensionality of GP inputs, and possibly to improve precision with a given amount of samples. On the downside, polar parametrization would entail domain considerations, such as ensuring continuity at boundaries. As using GPs for spatial modeling of classifier / detector scores (e.g. Teacy et al. [126] and Velez et al. [129]) is not a contribution of this research (rather, the extension of the approach to handle localization and model uncertainty) we did not explore the various parametrization options in depth.

3.4.5.2 Evaluation in classification under localization uncertainty

In Fig. 3.17a we present the evolution of classification over the set of views treated as a track, and the object instance shown in Fig. 3.14 in a setting similar to Sec. 3.4.3.2, i.e. we introduce a bias corresponding to erroneous localization estimate to each pose along the track, producing an erroneous estimate of pose relative to the object. For



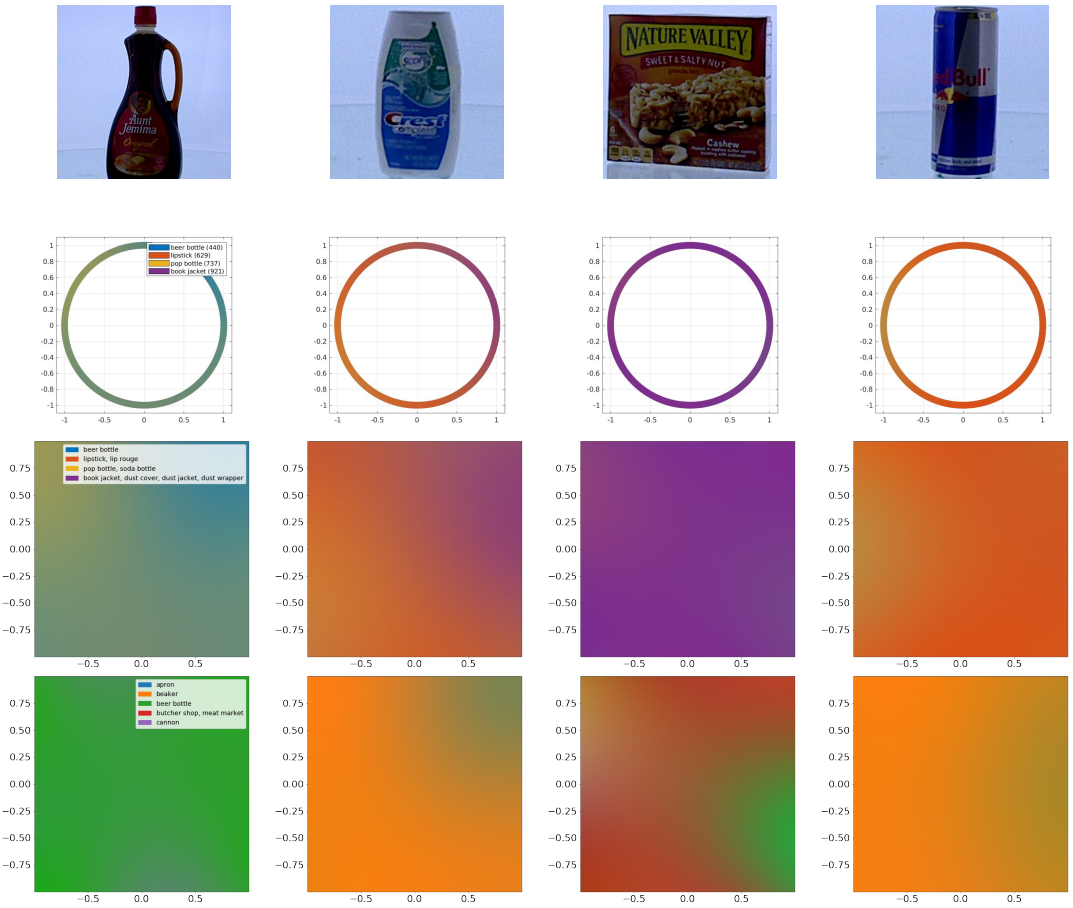
(a) **(b)** **(c)**

Figure 3.15: BigBIRD dataset setup (Fig. 3.15a, Fig. 3.15b taken from the dataset). Left: data collection setup, cameras facing the rotating table. Center: an example dataset image of the target instance on the rotating table. Right: Subset of camera poses relative to the rotating table for the different rotating table states. In all, images for 120 poses of the rotating table are provided, for each of the 5 cameras. We only use images from NP1 (i.e. the bottom row of relative poses in Fig. 3.15c) to learn 2D GP models, accordingly limiting ourselves (for simplicity) to classification of objects that are upright (w.r.t. BigBIRD coordinates).

classification candidates, we randomly chose a subset of 3 object instances (in addition to the ground truth class "aunt jemima original syrup"), shown in Fig. 3.16. The corresponding models display relatively little aliasing with the ground truth model (mostly in the second quadrant, with "crest complete minty fresh" and "red bull").

Both `Model Based` method and ours are fed with classification vectors from the true poses along with the erroneous pose estimate, our method in addition being input a standard deviation of $30cm$ for the 2D localization estimate (on each axis). The `Model Based` method is first affected before recovering, while our method, accounting for the pose uncertainty, is able to correctly accumulate information. To statistically examine this behavior, we randomize the localization estimate in the vicinity of the above biased estimate, adding to it a vector drawn from a zero centered Gaussian with standard deviation of $30cm$ along each axis. As before, our method receives a constant standard deviation of $30cm$ as the estimate uncertainty. The center and right columns of Fig. 3.17 show percentiles of the correct class/instance ("aunt jemima original syrup") probability output by each method (Fig. 3.17b) as well as the log MGR Eq. (3.45) (Fig. 3.17c) over the track / time indexes. Similarly to before, while in most cases both methods eventually deduce the correct class, the `Model Based` method experiences hard failures in around 10% of the runs outputting high probability for an incorrect class (around 10 times the score of the correct class) at the last step, while our method is able to classify correctly.

The reason for this behavior can be seen by considering variations in classification as function of localization bias, i.e. - given a raw classifier response - what is the GP model / class likelihood for various deviations of the localization estimate from ground truth. Computing this likelihood is equivalent to classification using the `Model Based` method (i.e., not considering localization uncertainty). Fig. 3.18 shows the



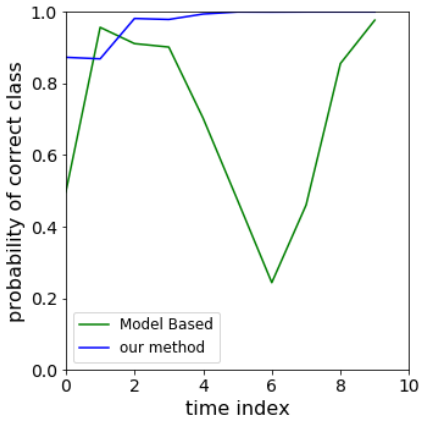
(a) "aunt jemima original syrup"

(b) "crest complete minty fresh"

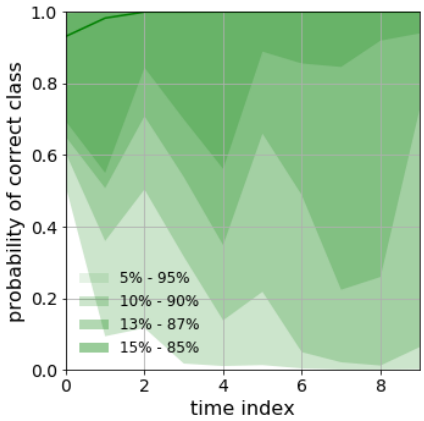
(c) "nature valley sweet and salty nut cashew"

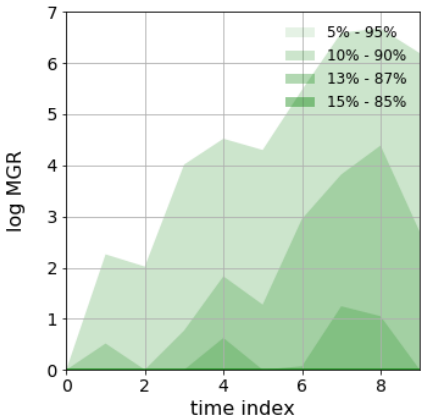
(d) "red bull"

Figure 3.16: Object instances and corresponding GP models. Top row: BigBIRD instances, cropped using provided bounding boxes. Second from top: GP models at training points. Third row: GP models raster. Bottom row: GP models for a different set of features (chosen as ones with highest significance with xgboost (‘)Chen16kdd). Instance names are listed in captions as they appear in the dataset.

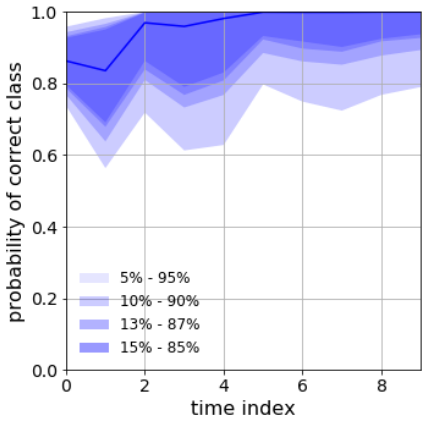


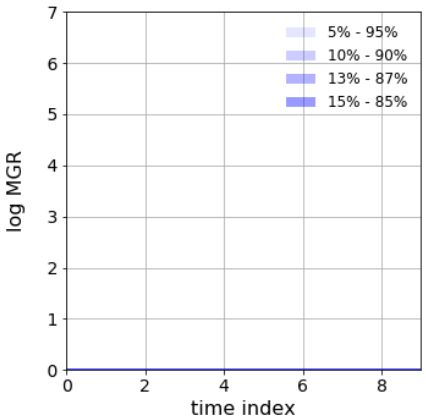
(a)





(b)



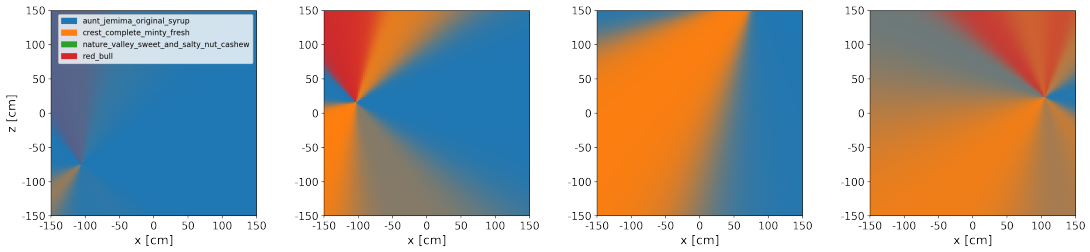


(c)

Figure 3.17: Top left: classification results (probability of ground truth class "aunt jemima original syrup") over the track from Fig. 3.14a with erroneous localization estimate for the `Model Based` and our method. Bottom left and right: statistical results when randomizing the error in localization estimate in the vicinity of the error from before. As before, color patch denotes the respective percentile range. Top row: probability of correct class, bottom row: log MGR Eq. (3.45). As the `Model Based` method classifies based on a single localization sample, its results vary, while our method is more resilient to the noisy estimate, consistent with the behavior observed in simulation.

classification obtained for a range of bias in 2D localization at different steps along the track from Fig. 3.14a. For each plot, the axes denote the amount of localization error in centimeters, the value at $(0, 0)$ corresponding to classification obtained with ground truth localization. The color at each pixel is obtained as the sum of colors corresponding to the object classes (shown in the legend), weighted by their likelihood given the localization error, e.g. pixel value at $(-30, 20)$ corresponds to classification obtained (by `Model Based`) given that localization estimate is off by $-30cm$ in the x axis, and $+20cm$ in the z axis, the correct class, "`aunt_jemima_original_syrup`", corresponding to blue color. For example, in view 0 the classification is correct for the above bias, whereas in views 3 and possibly also 6 classification is incorrect, which can be seen as the color departs from blue.

In the plot, colors of pixels indicate specific values of localization error causing the `Model Based` method to fail - the real-world equivalent of Fig. 3.4. Assuming model uncertainty negligible, the output of our method is obtained according to Eq. (3.42), roughly by averaging the output of `Model Based` over localization samples, drawn from a Gaussian centered at the erroneous estimate. Localization-uncertainty-aware classification score (i.e. using our method) thus depends on the proportion of samples of the correct color (the "average color") in the vicinity of each pixel. The wider spread of percentiles in Fig. 3.17 of the `Model Based` method is likewise explained by that its classification decisions are based on the value at a single localization sample, whereby bad estimates directly lead to erroneous classification ouputs. Averaging over a belief rather than using a single estimate allows us to mitigate the localization uncertainty-induced aliasing where enough localization samples produce the correct classification, and in any case - produces smoother classification decisions. On the downside, this sampling / averaging may reduce the confidence in the correct class where some of the samples produce incorrect classifications, as in view 1 in Fig. 3.17a.



(a) view 0 **(b)** view 1 **(c)** view 3 **(d)** view 6

Figure 3.18: Variations in classification function of localization bias for single views of the track from Fig. 3.14a. Each plot is obtained for a given semantic observation obtained from ground truth pose / viewpoint. The axes denote the amount of bias in centimeters, the color at each pixel is obtained as a (weighted) sum of colors corresponding to object classes (shown in the legend), weighted by their likelihood for the given localization error, i.e. the pixel at $(0, 0)$ shows classification with localization estimate equal to ground truth position (Note the difference from the raster plots of 3.16, where colors correspond to raw classification vector components).

Chapter 4

Data Association-aware semantic SLAM via Viewpoint-Dependent Classifier Models

4.1 Introduction

In this chapter we address perception for an autonomous robot traversing an environment with multiple objects, which it measures with a classifier unit. Robot and object localization is uncertain, although priors are available. Data association is not assumed given. We utilize a viewpoint-dependent model capturing spatial variation of the classifier response as function of viewpoint relative to the object. We show that we can address data association by maintaining multiple association hypotheses and their corresponding weights, while pruning hypotheses with weights that fall below a threshold, using the viewpoint semantic measurement model for disambiguation. We verify in simulation that the viewpoint-dependent model results in stronger disambiguation keeping the problem tractable.

4.2 Problem Formulation

Consider a robot operating in a partially known environment containing different, possibly perceptually similar or identical, objects. The robot aims to localize itself, and map the environment geometrically and semantically while reasoning about ambiguous data association (DA). We consider a closed-set setting where each object is assumed to be one of M classes. Moreover, in this work we consider the number of objects in the environment is known. The objects are assumed to be stationary.

Let x_k denote the robot's camera pose at time k; let o_n and c_n represent the n-th object pose and class, respectively. We denote the set of all object poses and classes by $\mathcal{O} \doteq \{o_1, ..., o_N\}$ and $\mathcal{C} \doteq \{c_1, ..., c_n\}$. To shorten notations, denote $\mathcal{X}_k \doteq \{x_{0:k}, \mathcal{O}\}$.

Further, we denote the data association realization at time k as β_k: given n_k object observations at time k, $\beta_k \in \mathbb{N}^{n_k}$; each element in β_k corresponds to an object observation, and is equal to an object's identity label. For example, if at time k the camera observes 2 objects with hypothesized identity labels 4 in observation 1 and 6 in observation 2, then $\beta_k = [\beta_{k,1}, \beta_{k,2}]^T \in \mathbb{N}^2$, and $\beta_{k,1} = 4$ and $\beta_{k,2} = 6$. Denote $\mathcal{Z}_k \doteq \{z_{k,1}, ..., z_{k,n_k}\}$ as the set of n_k measurements at time k, and u_k as the robot's action at time k.

Each measurement $z_{k,i} \in \mathcal{Z}_k$ consists of two parts: a geometric part $z^{geo}_{k,i}$, e.g. range or bearing measurements to an object, and a semantic part $z^{sem}_{k,i}$. The set of all geometric measurements for time k is denoted $\mathcal{Z}^g_k$, and similarly for semantic measurements $\mathcal{Z}^s_k$, such that $\mathcal{Z}_k = \mathcal{Z}^g_k \cup \mathcal{Z}^s_k$. We assume the geometric and semantic measurements are independent from each other. In addition, we assume independence between measurements at different time steps.

We consider standard motion and geometric observation Gaussian models, such that $\mathbb{P}(x_{k+1}|x_k, u_k) = \mathcal{N}(f(x_k, u_k), \Sigma_w)$ and $\mathbb{P}(z^{geo}_k|x_k, o) = \mathcal{N}(h^{geo}(x_k, o), \Sigma^{geo}_v)$. The process and geometric measurement covariance matrices, Σ_w and Σ^{geo}_v, as well as the functions $f(.), h^{geo}(.)$ are assumed to be known.

For the semantic measurements, we utilize a (deep learning) classifier that provides a vector of class probabilities where $z^{sem}_{k,i} \doteq \mathbb{P}(c_i|I_{k,i})$ given sensor raw observation $I_{k,i}$, e.g. an image cropped from a bounding box of a larger image taken by the camera of object i at time k. To simplify notations we drop index i, as the measurements, both semantic and geometric, apply to each bounding box. Thus, $z^{sem}_k \in \mathbb{R}^M$ with

$$z^{sem}_k \doteq \begin{bmatrix} \mathbb{P}(c = 1|I_k) & \cdots & \mathbb{P}(c = M|I_k) \end{bmatrix}^T. \tag{4.1}$$

A crucial observation, following Feldman and Indelman [29], is that z^{sem}_k is dependent on the camera's pose relative to the object (see Fig. 4.1). In this work we contribute an approach that leverages this coupling to assist in inference and data association disambiguation.

Specifically, we model this dependency via a classifier model $\mathbb{P}(z^{sem}_k|c = m, x_k, o)$. The classifier model represents the distribution over classifier output, i.e. class probability vector z^{sem}_k, when an object with a class hypothesis m is observed from relative pose $o \ominus x_k$. Note that for M classes we require M classifier models, one for each class. The model can be represented with a Gaussian Process (Feldman and Indelman [29] and Teacy et al. [126]) or a deep neural network (Kopitkov and Indelman [59]). In this work, we use a Gaussian classifier model, given by

$$\mathbb{P}(z^{sem}_k \mid c, x_k, o) = \mathcal{N}(h_c(x_k, o), \Sigma_c(x_k, o)), \tag{4.2}$$

where the viewpoint-dependent functions $h_c(x_k, o)$ and $\Sigma_c(x_k, o)$ are learned offline. Note that unlike Teacy et al. [126] and Feldman and Indelman [29] we do not model

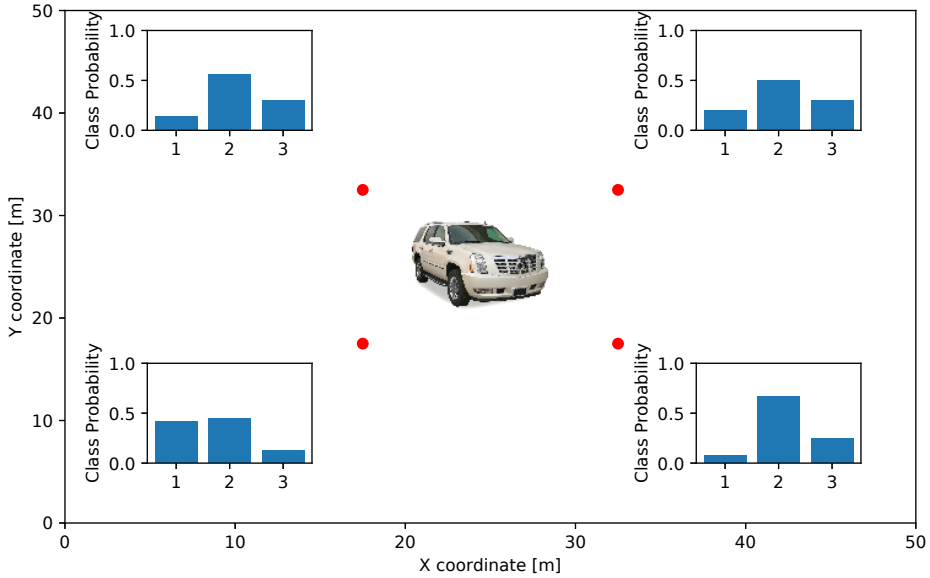


Figure 4.1: A classifier observing an object from multiple viewpoints will produce different classification scores for each viewpoint.

correlations in classifier scores among viewpoints. Conversely, we do not assume data association is known.

We assume a prior on initial camera and object poses, x_0 and X^o respectively, and class realization probability $\mathbb{P}(C)$. For simplicity, we assume independent variable priors (although this assumption is not required by our approach, and is not true in general, as e.g. some objects are more likely to appear together than others), thus we can write the prior as follows:

$$\mathbb{P}(x_0, \mathcal{X}^o, C) = \mathbb{P}(x_0) \prod_{i=1}^{N} \mathbb{P}(o_i)\mathbb{P}(c_i). \tag{4.3}$$

In this paper we use a Gaussian prior for the continuous variables, and uninformative (uniform) prior for the object classes.

Problem formulation: We aim to efficiently maintain the following hybrid belief

$$\mathbb{P}(\mathcal{X}_k, \mathcal{C}, \beta_{1:k} \mid \mathcal{H}_k), \tag{4.4}$$

with history $\mathcal{H}_k \doteq \{\mathcal{Z}_{1:k}, \mathcal{U}_{0:k-1}\}$. The belief Eq. (4.4) is both over continuous variables, i.e. robot and object poses $\mathcal{X}_k$ (continuous variables), and over discrete variables, i.e. object classes $\mathcal{C}$ and data association hypotheses thus far, $\beta_{1:k}$. In the following, we incorporate a viewpoint-dependent classifier model and develop a recursive formulation to update that hybrid belief with incoming information captured by the robot as it moves in the environment.

4.3 Approach

In this section we develop a recursive scheme to compute and maintain the hybrid belief from Eq. (4.4). We start by factorizing using the chain rule as

$$\mathbb{P}(\mathcal{X}_k, C, \beta_{1:k}|\mathcal{H}_k) = \underbrace{\mathbb{P}(\mathcal{X}_k|C, \beta_{1:k}, \mathcal{H}_k)}_{b[\mathcal{X}_k]^C_{\beta_{1:k}}} \underbrace{\mathbb{P}(C, \beta_{1:k}|\mathcal{H}_k)}_{w^C_{\beta_{1:k}}}, \tag{4.5}$$

where $b[\mathcal{X}_k]^C_{\beta_{1:k}} \doteq \mathbb{P}(\mathcal{X}_k \mid C, \beta_{1:k}, \mathcal{H}_k)$ is the conditional belief over the continuous variables, and $w^C_{\beta_{1:k}} \doteq \mathbb{P}(C, \beta_{1:k} \mid \mathcal{H}_k)$ is the marginal belief over the discrete variables, and can be considered as the conditional belief weight. Thus, each realization of the discrete variables, i.e. data association and class hypotheses, has its own probability (weight) and gives rise to a different belief over the continuous variables.

Moreover, the factorization (4.5) facilitates computation of marginal distributions that are of interest in practice. In particular, the posterior over robot and object poses can be calculated via

$$\mathbb{P}(\mathcal{X}_k \mid \mathcal{H}_k) = \sum_{\beta_{1:k}} \sum_{C} w^C_{\beta_{1:k}} b[\mathcal{X}_k]^C_{\beta_{1:k}}, \tag{4.6}$$

while the marginal distributions over object classes and data association hypotheses are given by

$$\mathbb{P}(C \mid \mathcal{H}_k) = \sum_{\beta_{1:k}} w^C_{\beta_{1:k}}, \tag{4.7}$$

$$\mathbb{P}(\beta_{1:k} \mid \mathcal{H}_k) = \sum_{C} w^C_{\beta_{1:k}}. \tag{4.8}$$

The posterior $\mathbb{P}(\mathcal{X}_k \mid \mathcal{H}_k)$ in Eq. (4.6) is a mixture belief that accounts for all hypotheses regarding data association and classification. Without semantic observations, our approach degenerates to passive DA-BSP. The term $\mathbb{P}(\mathcal{C} \mid \mathcal{H}_k)$ is the distribution over classes of all objects while accounting for both localization uncertainty and ambiguous data association. As such, it is important for robust semantic perception. Finally, the posterior over data association hypotheses, $\mathbb{P}(\beta_{1:k} \mid \mathcal{H}_k)$ accounts for all class realizations for all objects.

Next, we derive a recursive formulation for calculating the continuous and marginal distributions in the factorization (4.5). As will be seen, semantic observations along with the viewpoint-dependent classifier model (4.2) impact both of the terms in the factorization (4.5), and as a result assist in inference of robot and objects poses (via Eq. (4.6)) and helps in disambiguation between data association realizations (via Eq. (4.8)). Furthermore, as discussed in Sec. 3.D, while the number of objects' classes and data association hypotheses (number of weights $w^{\mathcal{C}}_{\beta_{1:k}}$) is intractable, in practice

many of these are negligible and can be pruned.

4.3.1 Conditional Belief Over Continuous Variables: $b[\mathcal{X}_k]^C_{\beta_{1:k}}$

Using Bayes law we get the following expression:

$$\begin{aligned} b[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k}} \equiv \mathbb{P}(\mathcal{X}_k \mid \mathcal{C}, \beta_{1:k}, \mathcal{H}_k) \propto \\ \mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, \mathcal{C}, \beta_k) \cdot \mathbb{P}(\mathcal{X}_k \mid \mathcal{C}, \beta_{1:k-1}, \mathcal{H}_k^-), \end{aligned} \tag{4.9}$$

where $\mathcal{H}_k^- \doteq \{\mathcal{Z}_{1:k-1}, \mathcal{U}_{0:k-1}\}$, the normalization constant is omitted as it does not depend on $\mathcal{X}_k$, and β_k is dropped in the second term because it refers to association of $\mathcal{Z}_k$ which is not present.

The expression $\mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, \mathcal{C}, \beta_k)$ in Eq. (4.9) is the joint measurement likelihood for all geometric and semantic observations obtained at time k. Given classifications, associations and robot pose at time k, history $\mathcal{H}_k^-$ and past associations $\beta_{1:k-1}$ can be omitted. The joint measurement likelihood can be explicitly written as

$$\begin{aligned} \mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, \mathcal{C}, \beta_k) = \\ \prod_{i=1}^{n_k} \mathbb{P}(z^{geo}_{k,i} \mid x_k, o_{\beta_{k,i}}) \cdot \mathbb{P}(z^{sem}_{k,i} \mid x_k, o_{\beta_{k,i}}, c_{\beta_{k,i}}), \end{aligned} \tag{4.10}$$

where $o_{\beta_{k,i}}$ and $c_{\beta_{k,i}}$ are the object pose and class corresponding to the measurement respectively, given DA realization β_k and n_k the number of measurements obtained at time k as before. Note that the viewpoint-dependent semantic measurement term above $\mathbb{P}(z^{sem}_{k,i} \mid x_k, o_{\beta_{k,i}}, c_{\beta_{k,i}})$ couples between semantic measurement and robot pose relative to object, making it useful for inference of both.

The term $b^-[\mathcal{X}_k]^C_{\beta_{1:k-1}} \doteq \mathbb{P}(\mathcal{X}_k \mid C, \beta_{1:k-1}, \mathcal{H}_k^-)$ in Eq. (4.9) is the propagated belief over continuous variables, which, using chain rule, can be written as

$$b^-[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k-1}} = \mathbb{P}(x_k | x_{k-1}, u_{k-1}) b[\mathcal{X}_{k-1}]^{\mathcal{C}}_{\beta_{1:k-1}}. \tag{4.11}$$

Overall, the conditional belief (4.9) can be represented as a factor graph (Kschischang, Frey, and Loeliger [64]). Note that each realization of $\beta_{1:k}$ has a different factor graph topology (observation factors are affected, motion model factors are not). For a fixed $\beta_{1:k}$ with different class assignments $\mathcal{C}$ the corresponding conditional belief factor graph topology remains the same (geometric and semantic observation factors connect the same nodes), but semantic factors change, according to class models.

Fig. 4.2 presents an example for 2 factor graphs, in which $k = 2$, $N = 2$, and the DA hypothesis is that at time $k = 1$ the camera observes object 1 for the first graph and 2 for the second, at time $k = 2$ the camera observes objects 1 and 2 for both graphs. To efficiently infer $\mathcal{X}_k$ for every realization of C and $\beta_{1:k}$, state of the art incremental inference approaches, such as iSAM2 (Kaess et al. [49]) can be used. The joint posterior from Eq. (4.4) can thus be maintained following Eq. (4.5) as a set of continuous beliefs

Figure 4.2: A toy example for two factor graphs in our approach, each for a different data association realization. The edges that connect between camera poses correspond to motion model $\mathbb{P}(x_k|x_{k-1}, u_{k-1})$. The edges that connect directly between camera and object poses correspond to the measurement model Eq. (4.10) for both semantic and geometric measurements. Thus a viewpoint-dependent semantic measurement model results in geometric constraints on robot-to-object relative pose.

$b[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k}}$ conditioned on the discrete variables $\beta_{1:k}$ and $\mathcal{C}$ each represented with a factor graph, along with their corresponding component weights $w^{\mathcal{C}}_{\beta_{1:k}}$, describing the marginal belief over discrete variables. In the next section, we describe how the latter can be calculated.

4.3.2 Marginal Belief Over Discrete Variables: $w^{C}_{\beta_{1:k}}$

To compute the DA and class realization weight $w^{\mathcal{C}}_{\beta_{1:k}}$ we marginalize over all continuous variables:

$$w^{C}_{\beta_{1:k}} \equiv \mathbb{P}(C, \beta_{1:k} \mid \mathcal{H}_k) = \int_{\mathcal{X}_k} \mathbb{P}(\mathcal{X}_k, C, \beta_{1:k}|\mathcal{H}_k) d\mathcal{X}_k. \tag{4.12}$$

Using Bayes law, we can expand the above as follows:

$$\mathbb{P}(\mathcal{X}_k, \mathcal{C}, \beta_{1:k} \mid \mathcal{H}_k) = \tag{4.13}$$

$$\eta \cdot \mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, \mathcal{C}, \beta_{1:k}) \cdot \mathbb{P}(\mathcal{X}_k, \mathcal{C}, \beta_{1:k}|\mathcal{H}_k^-) \tag{4.14}$$

where $\eta = \mathbb{P}(\mathcal{Z}_k \mid \mathcal{H}_k^-)^{-1}$ is a normalization constant and the joint measurement likelihood $\mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, C, \beta_{1:k})$ can be explicitly written as in Eq. (4.10).

We further expand $\mathbb{P}(\mathcal{X}_k, \mathcal{C}, \beta_{1:k}|\mathcal{H}_k^-)$ using chain rule:

$$\mathbb{P}(\mathcal{X}_k, \mathcal{C}, \beta_{1:k}|\mathcal{H}_k^-) = \mathbb{P}(\beta_k|\beta_{1:k-1}, \mathcal{X}_k, \mathcal{C}, \mathcal{H}_k^-) \cdot \tag{4.15}$$

$$\cdot \mathbb{P}(x_k|x_{k-1}, u_{k-1}) \cdot \mathbb{P}(\mathcal{X}_{k-1}, \mathcal{C}, \beta_{1:k-1} \mid \mathcal{H}_{k-1}), \tag{4.16}$$

where:

$$\mathbb{P}(\mathcal{X}_{k-1}, \mathcal{C}, \beta_{1:k-1}|\mathcal{H}_{k-1}) = w^{\mathcal{C}}_{\beta_{1:k-1}} \; b[\mathcal{X}_{k-1}]^{\mathcal{C}}_{\beta_{1:k-1}}. \tag{4.17}$$

Eq. (4.17) is the prior belief calculated at time $k-1$ and represented as a continuous belief component along with corresponding weight as described above. The term

$\mathbb{P}(\beta_k|\beta_{1:k-1}, \mathcal{X}_k, \mathcal{C}, H_k^-)$ from Eq. (4.16) is the object observation model that represents the probability of observing a scene given a hypothesis of camera and object poses. In this paper we use a simple model that depends only on camera and object poses at current time step, thus it can be written as $\mathbb{P}(\beta_k \mid x_k, \mathcal{O}_{\beta_k})$, where $\mathcal{O}_{\beta_k} \doteq \{o_{\beta_k,i}\}_{i=1}^{n_k}$. If the model predicts observation of all objects corresponding to β_k then $\mathbb{P}(\beta_k \mid x_k, \mathcal{O}_{\beta_k})$ is equal to a constant, otherwise it is zero.

Plugging the above into Eq. (4.12) yields a recursive rule for calculating component weights at time k

$$w^{\mathcal{C}}_{\beta_{1:k}} = \eta \cdot \int_{\mathcal{X}_k} \mathbb{P}(\mathcal{Z}_k|\mathcal{X}_k, \mathcal{C}, \beta_k) \cdot \mathbb{P}(\beta_k|x_k, \mathcal{O}_{\beta_k}) \cdot \tag{4.18}$$

$$\cdot b^-[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k-1}} w^{\mathcal{C}}_{1:k-1} \, d\mathcal{X}_k. \tag{4.19}$$

The normalization constant η (from Eq. (4.14)) does not depend on variables and cancels out when weights are normalized to sum to 1. It is therefore dropped out in subsequent calculations. Note that the realization weight from the previous time step $w^C_{\beta_{1:k-1}}$ is independent from $\mathcal{X}_k$, and thus can be taken out of the integral. Recalling Eq. (4.10), the continuous variables participating in $\mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, \mathcal{C}, \beta_{1:k})$ are x_k and $\mathcal{O}_{\beta_k}$. Those variables are participating also in $\mathbb{P}(\beta_k \mid x_k, o_k)$. As $b^-[\mathcal{X}_k]^C_{\beta_{1:k-1}}$ is Gaussian, all other continuous variables can be marginalized easily. On the other hand, x_k and $\mathcal{O}_{\beta_k}$ must be sampled because of the object observation model $\mathbb{P}(\beta_k \mid x_k, \mathcal{O}_k)$, which is commonly not Gaussian. If the observation model predicts that the objects will not be observed for most of the samples, then $w^C_{\beta_{1:k}}$ will be small and likely to be pruned. We can express the realization weight as follows:

$$\begin{aligned} w^{\mathcal{C}}_{\beta_{1:k}} \propto w^{\mathcal{C}}_{\beta_{1:k-1}} \iint_{x_k, \mathcal{O}_{\beta_k}} & \mathbb{P}(\mathcal{Z}_k \mid x_k, \mathcal{O}_{\beta_k}, \mathcal{C}, \beta_{1:k}) \cdot \\ & \cdot \mathbb{P}(\beta_k \mid x_k, \mathcal{O}_{\beta_k}) \; b^-[x_k, \mathcal{O}_{\beta_k}]^{\mathcal{C}}_{\beta_{1:k-1}} \, dx_k \, d\mathcal{O}_{\beta_k}, \end{aligned} \tag{4.20}$$

where:

$$\begin{aligned} b^-[x_k, \mathcal{O}_{\beta_k}]^{\mathcal{C}}_{\beta_{1:k-1}} \doteq \mathbb{P}(x_k, \mathcal{O}_{\beta_k}|C, \beta_{1:k-1}, \mathcal{H}_k^-) = \\ \int_{\mathcal{X}_k^- \setminus \{x_k, \mathcal{O}_{\beta_k}\}} b^-[\mathcal{X}_k]^C_{\beta_{1:k-1}} \, d\left\{\mathcal{X}_k^- \setminus \{x_k, \mathcal{O}_{\beta_k}\}\right\}. \end{aligned} \tag{4.21}$$

The viewpoint dependent classifier model contributes to data association disambiguation by acting as reinforcement or contradiction to the geometric model. If both 'agree' on the poses' hypothesis, $w^C_{\beta_{1:k}}$ will be large relative to cases where both 'disagree'.

Next, we provide an overview of the inference scheme, then address computational aspects.

4.3.3 Overall Algorithm

The proposed scheme is outlined in Alg. 4.1. For every time step we are input the prior belief $\mathbb{P}(\mathcal{X}_{k-1}, \mathcal{C}, \beta_{1:k-1} \mid \mathcal{H}_{k-1})$ represented following Eq. (4.5) as a set of weights $w^{\mathcal{C}}_{\beta_{1:k-1}} \doteq \mathbb{P}(\mathcal{C}, \beta_{1:k-1} \mid \mathcal{H}_{k-1})$ and corresponding continuous (Gaussian) belief components $b[\mathcal{X}_{k-1}]^{\mathcal{C}}_{\beta_{1:k-1}} \doteq \mathbb{P}(\mathcal{X}_{k-1} \mid \mathcal{C}, \beta_{1:k-1}, \mathcal{H}_{k-1})$. In our implementation we maintain a separate factor graph for each such component. We also obtain an action u_{k-1} and observations $\mathcal{Z}_k$, separated into geometric $\mathcal{Z}^g_k$, and semantic $\mathcal{Z}^s_k$. We propagate each prior belief component using the motion model $\mathbb{P}(x_k \mid x_{k-1}, u_{k-1})$ (step 3). Each component then splits to a number of subcomponents, one for each possible assignments of data associations β_k at current time (generally a vector of length n_k). Procedure `PropWeights` at step 5 computes the normalized weight of each subcomponent via Eq. (4.20) as a product of the component (prior) weight $w^{\mathcal{C}}_{\beta_{1:k-1}}$ with an update term comprising the measurement likelihood $\mathbb{P}(\mathcal{Z}_k \mid x_k, \mathcal{O}_{\beta_k}, \mathcal{C}, \beta_{1:k})$ (both geometric and semantic, see Eq. (4.10)) and object observation model $\mathbb{P}(\beta_k \mid x_k, \mathcal{O}_{\beta_k})$, averaged over the propagated belief $b^-[x_k, \mathcal{O}_{\beta_k}]^{\mathcal{C}}_{\beta_{1:k-1}}$ from Eq. (4.21). In step 7 we prune low-weight subcomponents by setting their weights to 0 and re-normalizing remaining weights to 1, in an approximation to true posterior (other pruning strategies are equally possible). In step 11 we update the posterior for non-zero weight subcomponents using current measurements. Finally, we return posterior as a set of Gaussian components and corresponding weights.

We next address aspects of computational tractability of the scheme.

3.D Computational Complexity and Tractability

With M candidate classes, and N objects, the number of possible class realizations, and consequently initial number of belief components, is M^N. At time step k each prior component splits into up to N^{n_k} subcomponents as each measurement can in general be associated to any scene object. The maximum number of components at time k is thus $M^N \cdot \prod_{j=1}^{k} N^{n_j} = O\left(M^N \cdot N^{\psi \cdot k}\right)$ if ψ is an upper bound on n_k, making the approach computationally intractable in theory without pruning. In practice, as observed by Pathak, Thomas, and Indelman [95], the number of components that need to be accounted for is limited by the belief, and is much smaller than the theoretical maximum, with the rest getting negligible weights that can be safely pruned under any scheme. Further, our empirical results suggest that semantic information added through the viewpoint-dependent factors leads to even stronger disambiguation than observed in DA-BSP (which uses only geometric information), both in data association and localization, resulting in smaller number of non-negligible weights.

Additionally, we hypothesize that in practice classification uncertainty is usually limited to only a few classes, and thus would not cause a computational bottleneck even with numerous candidate classes. Further, we avoid explicitly maintaining the initially exponential number of components (M^N) by noting that the classes of objects

that were not observed yet under an association hypothesis $\beta_{1:k}$ do not participate in the inference process for that belief component, and thus do not need to be maintained separately. That is, for two class realizations $\mathcal{C}$ and $\mathcal{C}'$, if $\mathcal{C}_{\beta_{1:k}} = \mathcal{C}'_{\beta_{1:k}}$ with $\mathcal{C}_{\beta_{1:k}} \doteq \{\forall_{1 \leq j \leq k}, i \;\; c_{\beta_{j,i}}\}$ (i.e. classifications for all associated objects are the same) and $\mathcal{C}_{\neg\beta_{1:k}} \neq \mathcal{C}'_{\neg\beta_{1:k}}$ (i.e. realizations differ on classifications for objects that do not participate in $\beta_{1:k}$), then $w^{\mathcal{C}'}_{\beta_{1:k}} = w^{\mathcal{C}}_{\beta_{1:k}}$ (assuming uninformative prior on classes) and $b[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k}} = b[\mathcal{X}_k]^{\mathcal{C}'}_{\beta_{1:k}}$ (always), without need to compute or maintain those separately.

Finally we note that parts of Alg. 4.1 can be readily parallelized ('embarrassingly parallel'), thanks to computations being independent across components and wide availability of massively parallel processors (e.g. GPUs), contributing to its practical applicability.

4.4 Experimental Results

In this section we evaluate the performance of our approach in a 2D simulation and demonstrate the advantage of using a viewpoint dependent classifier model for disambiguating between DA realizations and improving inference accuracy. Our implementation uses the GTSAM library (Dellaert [21]) with a Python wrapper; all experiments were run on an Intel i7-7700 CPU running at 2800 GHz and with 16GB RAM.

We consider a scenario where the robot navigates in an uncertain perceptually aliased environment represented by a set of scattered objects of the same class, i.e. objects differ in their position and orientation. In this scenario $M = 2$ and $N = 6$, thus the number of possible class realizations is $M^N = 64$. Fig. 4.3a shows the ground truth object poses and robot trajectory.

The prior (4.3) comprises a highly uncertain initial robot pose, and an uninformative prior on object classes. Object poses are assumed to be known up to a certain accuracy (i.e. uncertain map). The prior covariance of the objects is $\Sigma_o = diag(0.05, 0.05, 0.5 \cdot 10^{-3})$, and initial robot pose is $\Sigma_p = diag(100, 100, 0.04)$. The process and geometric measurement covariance matrices are $\Sigma_w = diag(0.75 \cdot 10^{-3}, 0.75 \cdot 10^{-3}, 0.25 \cdot 10^{-3})$ (corresponds to spatial coordinates and orientation), and $\Sigma_v^{geo} = diag(0.1, 0.05)$ (corresponds to range and bearing).

The semantic measurement model (4.2) is defined as:

$$h_c(c = 1, \theta) = \begin{bmatrix} \alpha \sin^2(\theta/2) + (1 - \alpha) \\ \alpha - \alpha \sin^2(\theta/2) \end{bmatrix} \tag{4.22}$$

where θ is the relative angle from the object to camera, calculated from the relative pose $x_k^{rel} \doteq x_k \ominus o$. This chosen model represents a mirror symmetrical object (e.g. a car) with a parameter α that corresponds to the viewpoint dependency 'strength', i.e. $\frac{\partial h_c}{\partial \theta}$ values are larger when α increases (for computation details, see Kopitkov and Indelman [59]). We assume the classifier scores are independent from camera-

to-object range as the observations are cropped from bounding boxes, and unless the camera is very close to the object the perspective distortion is negligible. In practice, the classifier model can be learned from images of an object from different viewpoints with corresponding classifier outputs via a neural network or GP for example. The measurement covariance matrix $\Sigma_c \doteq (R^T R)^{-1}$ is defined as $R = K \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \end{bmatrix}$. We note that in general, also Σ_c can be viewpoint-dependent (Feldman and Indelman [29] and Kopitkov and Indelman [59]). The parameters α and K are constants and take the values $\alpha = 0.25$ and $K = 15$ by default. We sample measurements from our motion, geometric, and semantic models.

Further, we sample 1000 sets of x_k and o_{β_k} for each computation of $w^C_{\beta_{1:k}}$, see procedure PropWeights in Alg. 4.1, and compute them as shown in Eq. (4.20). At each time k we prune components with weight w below threshold $\frac{\{w_k\}_{max}}{150} \leq w$, where $\{w_k\}_{max}$ is the highest weight component at time k.

We compare performance of our approach that utilizes semantic observations along with a viewpoint-dependent classifier model against an alternative that does not use this information, with the latter roughly corresponding to the passive instance of DA-BSP (Pathak, Thomas, and Indelman [95]). To quantify performance as a function of α we compare between the following metrics:

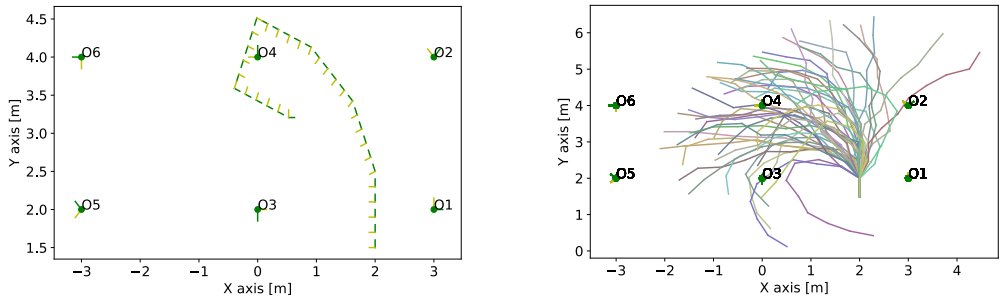
1. Entropy over data association weights: for N_k non-pruned weights $\{w_i\}_{i=1}^{N_k}$ we compute the entropy $H(w)$ with $H(w) \doteq -\sum_{i=1}^{N_k} w_i \log(w_i)$.

2. Determinant of position covariance $det(\Sigma)$ of x_k for the highest weight realization at each time k.

3. Estimation error $\tilde{x}^{w_{max}}$, which is the Euclidean distance from ground truth to highest weight estimation for the last pose.

4. Estimation error $\tilde{x}^{w-avg}$, which is the weighted average of all estimation errors for the last pose.

Fig. 4.4 shows results of an example scenario for different time steps, comparing between using the viewpoint-dependent classifier model (middle row) and without semantic information (upper row), essentially utilizing passive DA-BSP (Pathak, Thomas, and Indelman [95]). At each time k, the plots show the mixture posterior $\mathbb{P}(x_k|\mathcal{H}_k)$ over camera pose x_k, calculated from (4.6), where each component is a Gaussian, thus represented by mean and covariance. Estimated camera poses are shown in red and blue lines, where the blue line represents the camera orientation. Components with higher weight are shown with thicker covariance ellipse lines. To reduce clutter, the posterior over the rest of the continuous variables, i.e. object poses and past robot poses, is not shown. Additionally, the plots show the ground truth trajectory (from Fig. 4.3a) of the robot. The bottom row reports the probabilities of DA hypotheses



(a) **(b)**

Figure 4.3: **(a)** An example scenario with ground truth camera trajectory, represented in terms of camera poses (green line is the camera heading) and objects $O1$ to $O6$ (green dots indicate position, orientation is indicated by green lines from the dots). **(b)** Multiple sampled paths for the statistical study, each path realization is presented with different color.

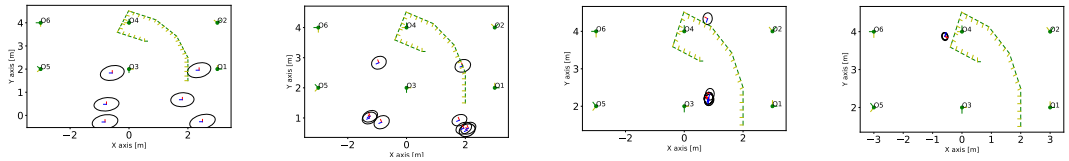
from (4.8) for different time instances for both compared cases. The correct association is marked with a green circle.

As seen from the upper row of Fig. 4.4, inference without incorporating viewpoint-dependent semantic information (i.e. omitting the semantic model in Eq. (4.10), which results in the form from Pathak, Thomas, and Indelman [95]) results in the first time steps in multiple DA realizations with similar weights. The reason is that given only geometric range and bearing measurements without observing all the objects, inference results can be interpreted in multiple ways, i.e. perceptually aliased (see Fig. 4.4i). Only at time $k = 25$ the DA was disambiguated once the camera observed objects $O1$ and $O2$.

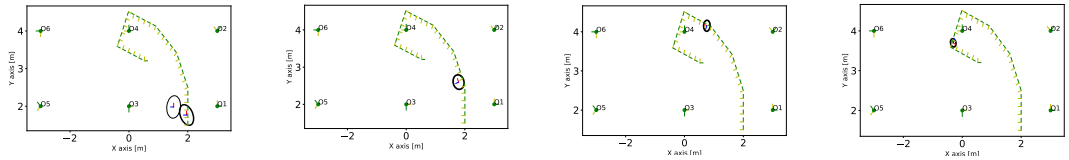
In contrast, utilizing a viewpoint-dependent classifier model admits faster DA disambiguation, as shown in the bottom row of Fig. 4.4. In particular, already at time $k = 1$ the posterior $\mathbb{P}(x_k|\mathcal{H}_k)$ has only two non-negligible components, while at time $k = 5$ there is a single DA realization with significant weight. This shows an improvement over Fig. 4.4b where there are multiple DA realizations with significant weight when not using the classifier model.

The bottom row in Fig. 4.4 presents the realization weights for the times $k = 1, 5, 15, 25$, and compare between weights without and with classifier model. For each realization $\beta_{1:k}$, we present $\mathbb{P}(\beta_{1:k} \mid \mathcal{H}_k)$ after pruning without classifier model as a blue bar, and with as a red bar. If the bar is missing, then $\mathbb{P}(\beta_{1:k} \mid \mathcal{H}_k) = 0$. In all sub-figures the classifier model reduces the number of non pruned DA realizations, and for time $k = 15$ and $k = 25$ the DA is disambiguated with the classifier model. We observe more DA realizations when the classifier model is not used, and at time $k = 15$ the DA with a classifier model fully disambiguated.

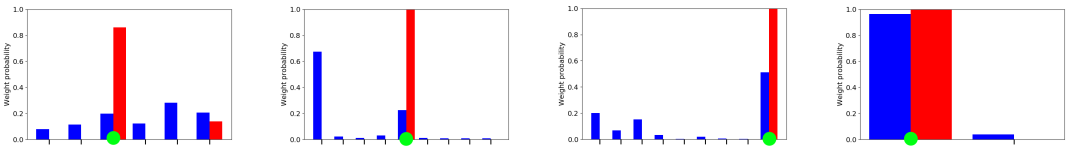
Further, we quantify the performance improvement due to the viewpoint-dependent classifier model in a statistical study by sampling multiple ground truth tracks in the



(a) $k = 1$, no classifier **(b)** $k = 5$, no classifier **(c)** $k = 15$, no classifier **(d)** $k = 25$, no classifier



(e) $k = 5$, with classifier **(f)** $k = 15$, with classifier **(g)** $k = 15$, with classifier **(h)** $k = 25$, with classifier

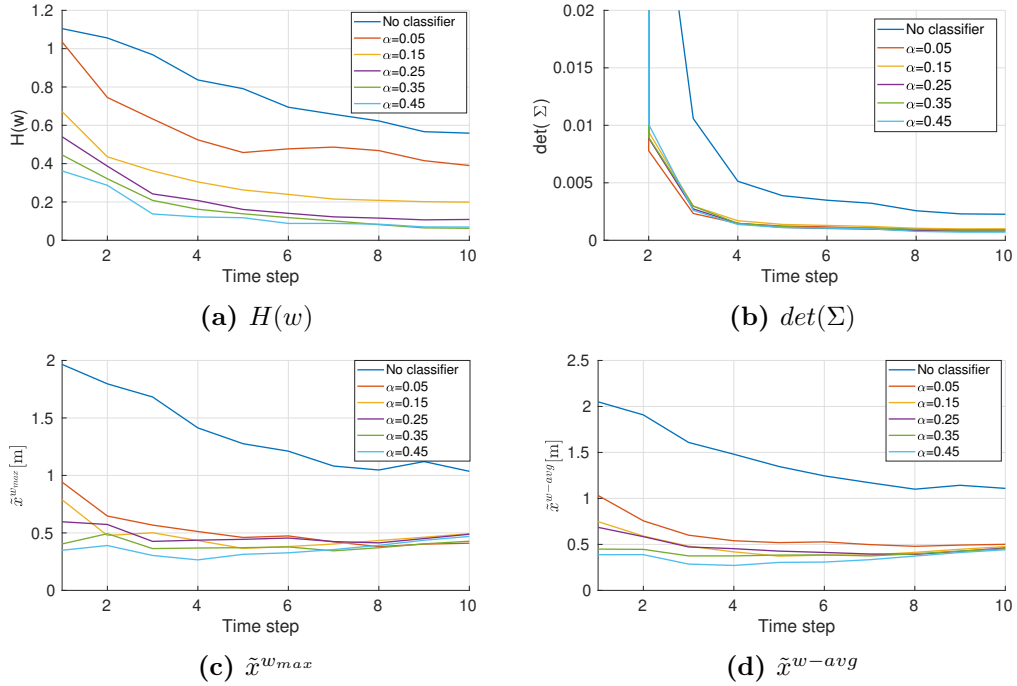


(i) $k = 1$ **(j)** $k = 5$ **(k)** $k = 15$ **(l)** $k = 25$

Figure 4.4: **(a)** - **(h)**: Posterior over robot poses of all non-pruned realizations for times $k = 1, 5, 15, 25$, without (first row) and with a classifier model (second row). **Bolder** lines correspond to higher weights. Ground truth trajectory is shown in each of the plots (in terms of camera poses). **(i)**-**(l)**: Corresponding posterior over data association hypotheses, $\mathbb{P}(\beta_{1:k} \mid \mathcal{H}_k)$, at each time. **Blue** bars are without classifier model, **red** bars are with. **Green** circles represent ground truth data associations.

scenario, while keeping the same landmarks. The sampled tracks are shown in Fig. 4.3b. For this study, we sampled 50 different tracks with 10 time steps of path length, and performed a statistical analysis on the performance parameters. In all paths, the starting position is identical.

The results of this study are shown in Fig. 4.5, which shows average over each of the mentioned metrics ($H(w)$, $\det(\Sigma)$, $\tilde{x}^{w_{max}}$, $\tilde{x}^{w\text{-avrg}}$). In that figure we also study sensitivity to α, which controls the level of viewpoint-dependency in the considered classifier model (4.22). The plots show a significant improvement of utilizing a classifier model, both for DA disambiguation and inference where the estimation error (Fig. 4.5c, 4.5d) and uncertainty (Fig. 4.5b) are lower when the model is utilized. From all the plots, the most notable performance increase occurs for DA disambiguation (Fig. 4.5a), where stronger viewpoint dependence assists more significantly; Overall, Fig. 4.5 presents a strong advantage for utilizing a viewpoint classifier model in the presented scenario.



(a) $H(w)$ **(b)** $det(\Sigma)$

(c) $\tilde{x}^{w_{max}}$ **(d)** $\tilde{x}^{w-avg}$

Figure 4.5: Effects of different α values on DA disambiguation ability, estimation uncertainty and accuracy in terms of the metrics ($H(w)$, $det(\Sigma)$, $\tilde{x}^{w_{max}}$, and $\tilde{x}^{w-avg}$), averaged over 50 sampled tracks (see Fig. 4.3b).

Algorithm 4.1 Data Association-Aware Mapping and Localization. Inference at time k

Require: Prior belief $\mathbb{P}(\mathcal{X}_{k-1}, \mathcal{C}, \beta_{1:k-1} \mid \mathcal{H}_{k-1})$, observations $\mathcal{Z}_k = (\mathcal{Z}^g, \mathcal{Z}^s)$, action $\mathcal{U}_{k-1}$

for every component $\beta_{1:k-1}, \mathcal{C}$ **s.t.** $w^{\mathcal{C}}_{\beta_{1:k-1}} > 0$ **do**
  ▷ Propagate component according to motion model
  $b[\mathcal{X}_k^-]^{\mathcal{C}}_{\beta_{1:k-1}} \leftarrow \mathbb{P}(x_k | x_{k-1}, \mathcal{U}_{k-1}) \cdot b[\mathcal{X}_{k-1}]^{\mathcal{C}}_{\beta_{1:k-1}}$
  ▷ Propagate weights Eq. (4.18), Eq. (4.20)
  $w^{\mathcal{C}}_{\beta_{1:k}} \leftarrow \textsc{PropW}.\left(b[\mathcal{X}_k^-]^{\mathcal{C}}_{\beta_{1:k-1}}, w^{\mathcal{C}}_{\beta_{1:k-1}}, \mathcal{Z}_k\right)$
  ▷ Prune low-probability components
  $w^{\mathcal{C}}_{\beta_{1:k}} \leftarrow \textsc{PruneAndNormalize}(w^{\mathcal{C}}_{\beta_{1:k}})$
  ▷ Propagate non-zero weight components
  for $\beta_{1:k}, \mathcal{C}$ **s.t.** $w^{\mathcal{C}}_{\beta_{1:k}} > 0$ **do**
    ▷ Add observation factors, Eqs. (4.9), and (4.10)
    $b[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k}} \leftarrow b[\mathcal{X}_k^-]^{\mathcal{C}}_{\beta_{1:k-1}} \cdot \mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_k, \mathcal{C}, \beta_k)$
  end for
end for
return $\mathbb{P}(\mathcal{X}_k, \mathcal{C}, \beta_{1:k} \mid \mathcal{H}_k) \equiv \{(b[\mathcal{X}_k]^{\mathcal{C}}_{\beta_{1:k}}, w^{\mathcal{C}}_{\beta_{1:k}})\}$

procedure $\textsc{PropWeights}(b[\mathcal{X}_k^-]^{\mathcal{C}}_{\beta_{1:k-1}}, w^{\mathcal{C}}_{\beta_{1:k-1}}, \mathcal{Z}_k)$
  for every possible assignment of β_k **do**
    ▷ Sample current poses by Eq. (4.21)
    Sample $\{x_k^{(i)}, \mathcal{O}_{\beta_k}{}^{(i)}\}_{i=1}^{n_s} \sim b^-[x_k, \mathcal{O}_{\beta_k}]^{\mathcal{C}}_{\beta_{1:k-1}}$
    ▷ Calculate update factor and propagate Eq. (4.20)
    $\psi \leftarrow (1/n_s) \cdot \sum_{i=1}^{n_s} \mathbb{P}(\mathcal{Z}_k, \beta_k \,|\, x_k^{(i)}, \mathcal{O}_{\beta_k}{}^{(i)}, \mathcal{C}, \beta_{1:k-1})$
    $\widetilde{w}^{\mathcal{C}}_{\beta_{1:k}} \leftarrow w^{\mathcal{C}}_{\beta_{1:k-1}} \cdot \psi$
  end for
  ▷ Normalize weights and return
  return $w^{\mathcal{C}}_{\beta_{1:k}} \leftarrow \widetilde{w}^{\mathcal{C}}_{\beta_{1:k}} / \sum_{\beta_k} \widetilde{w}^{\mathcal{C}}_{\beta_{1:k}}$
end procedure

procedure $\textsc{PruneAndNormalize}(w^{\mathcal{C}}_{\beta_{1:k}})$
  for $\beta_{1:k}, \mathcal{C}$ **s.t.** $w^{\mathcal{C}}_{\beta_{1:k}} <$ THRESHOLD **do**
    $\widetilde{w}^{\mathcal{C}}_{\beta_{1:k}} \leftarrow 0$
  end for
  return $w^{\mathcal{C}}_{\beta_{1:k}} \leftarrow \widetilde{w}^{\mathcal{C}}_{\beta_{1:k}} / \sum_{\beta_k} \widetilde{w}^{\mathcal{C}}_{\beta_{1:k}}$
end procedure

Chapter 5

Continuous Learned Representation for Semantic SLAM through a Viewpoint-Dependent Observation Model

5.1 Introduction

In this chapter we propose a novel formulation for object SLAM through continuous (as opposed to mixed - continuous and discrete) inference - by means of inference in a learned latent semantic representation space. The semantic representation space is induced by a viewpoint-dependent model modeling variation in object appearance. We propose approaches to learning such a model, as well as an alternative formulation of the model over *changes* in relative viewpoint, which eliminates the conditioning on object pose, rendering the model more general (e.g. applicable for objects for which pose cannot be adequately defined), but on the downside no more allowing object pose inference. We present experiments, demonstrating the learned representations and the use of the learned models for inference in simulation, using images from the BigBIRD dataset (Singh et al. [111]).

5.2 Problem Definition and Notations

Consider a robot traversing an unknown environment, taking observations of different scenes. Robot motion between times t_k and t_{k+1} is initiated by a control input $\mathcal{U}_k$, that may originate from a human user, or be determined by a motion planning algorithm. We denote the robot pose at time instant k by x_k, and by $\mathcal{X}_{0:k} = \{\mathcal{X}_0, \ldots, \mathcal{X}_k\}$ the se-

quence of poses up to that time. We denote by $\mathcal{Z}_k = \{z_{k,(i)}\}$ all observations obtained at time k. We denote with $\mathcal{Z}_k^g \subseteq \mathcal{Z}_k$ (correspondingly, $\mathcal{Z}_k^g = \{z_{k,(i)}^g\}$) geometric detections, including detections of landmarks, object bounding boxes (image coordinates) and centroids. We denote with $\mathcal{Z}_k^s \subseteq \mathcal{Z}_k$ (correspondingly, $\mathcal{Z}_k^s = \{z_{k,(i)}^s\}$) the semantic measurements, i.e. bounding box RGB images. We further denote the observation and user control history up until time k as $\mathcal{H}_k = \{\mathcal{U}_{0:k-1}, \mathcal{Z}_{0:k}\}$.

We are interested in maintaining a posterior

$$\mathbb{P}(\mathcal{X}_{0:k}, \mathcal{L}, \mathcal{O}, \mathcal{E} \mid \mathcal{H}_k), \tag{5.1}$$

over robot track, geometric landmarks $\mathcal{L}$ (e.g. tracked SIFT, ORB, geometric features such as planes or lines), object geometric information $\mathcal{O} = \{o_i\}$ (centroids in the simplest case, other representation possibilities exist including ellipsoids (Nicholson, Milford, and Sünderhauf [90]) or cubes (Yang and Scherer [132]) - all of these are tied to the per-object semantic representations), and per-object continuous variables $\mathcal{E} = \{e_i\}$ capturing object semantic information. Fig. 5.1b shows a factor graph corresponding to this approach.

In the following sections, we develop the framework for learning the viewpoint-dependent model from data Sec. 5.5 and for inference using it in Sec. 5.3, we address aspects of the practical applicability of the approach in Sec. 5.6, we then present experiments exploring our approach.

5.3 Incremental Inference

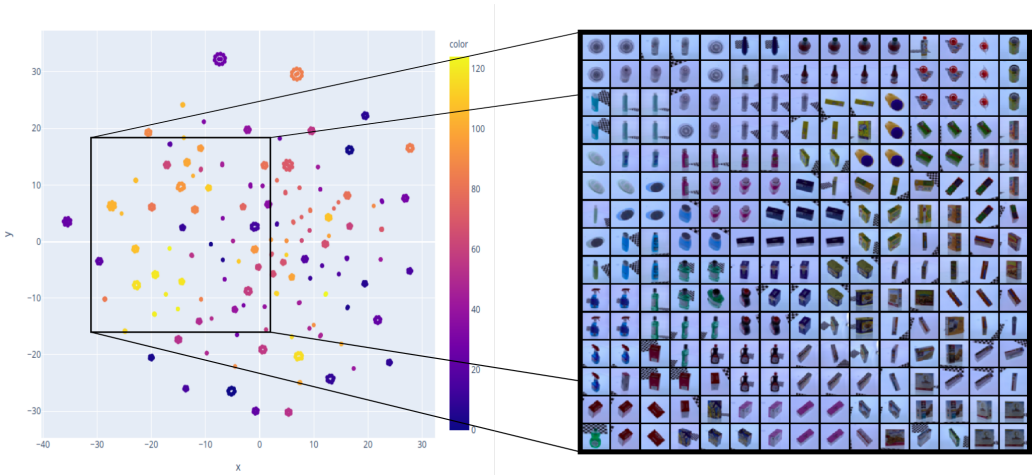
The update equation for the target posterior at time k can be generally written by applying Bayes rule as

$$\begin{aligned} &\mathbb{P}(\mathcal{X}_{0:k}, \mathcal{L}, \mathcal{O}, \mathcal{E} \mid \mathcal{H}_k) = \\ &\eta \cdot \mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_{0:k}, \mathcal{L}, \mathcal{O}, \mathcal{E}, \mathcal{H}_k^-) \cdot \mathbb{P}(\mathcal{X}_k \mid \mathcal{X}_{k-1}, \mathcal{U}_{k-1}) \\ &\cdot \mathbb{P}(\mathcal{X}_{0:k-1}, \mathcal{L}, \mathcal{O}, \mathcal{E} \mid \mathcal{H}_{k-1}), \end{aligned} \tag{5.2}$$

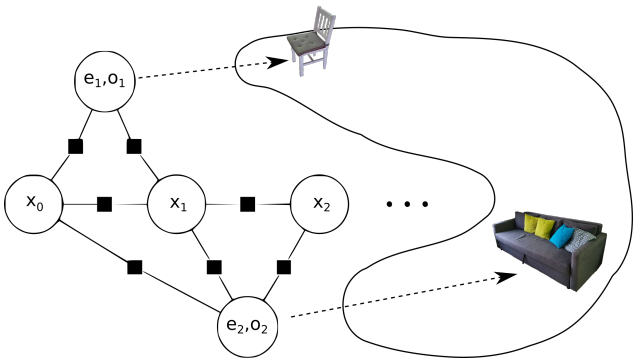
where $\mathcal{H}_k^- \doteq \mathcal{H}_k \setminus \{\mathcal{Z}_k\}$, and η is a constant normalization factor. In the above, the motion model $\mathbb{P}(\mathcal{X}_k \mid \mathcal{X}_{k-1}, \mathcal{U}_{k-1})$ is assumed known, and priors for the map variables $\mathcal{L}, \mathcal{O}, \mathcal{E}$ may be present or assumed uninformative. We further split the measurement update term into a "geometric" and a "semantic" part

$$\begin{aligned} &\mathbb{P}(\mathcal{Z}_k \mid \mathcal{X}_{0:k}, \mathcal{L}, \mathcal{O}, \mathcal{E}, \mathcal{H}_k^-) = \\ &\mathbb{P}(\mathcal{Z}_k^g \mid \mathcal{X}_k, \mathcal{L}, \mathcal{O}) \cdot \mathbb{P}(\mathcal{Z}_k^s \mid \mathcal{X}_{0:k}, \mathcal{O}, \mathcal{E}, \mathcal{H}_k^-). \end{aligned} \tag{5.3}$$

The former (geometric) term is comprised of geometric models e.g. projection factors (and thus given variables it does not depend on measurement history). The latter



(a) Embedding space induced by the Viewpoint-Dependent model fit on the BigBIRD dataset.



(b) Example factor graph.

Figure 5.1: Top: Learned latent space visualization using tSNE (Maaten and Hinton [74]). Latent vectors (encoder output) on the left and (roughly) corresponding input images from the BigBIRD dataset (Singh et al. [111]) on the right, while training the Viewpoint-Dependent model from Sec. 5.4. Bottom: An example factor graph with two objects described by latent semantic representations e_1 and e_2 and geometric representations o_1 and o_2, observed from consecutive poses $x_{0:2}$. Factors connecting robot poses and semantic representatinons correspond to the learned semantic observation model Eq. (5.6).

term is the semantic observation model (which in general may depend on measurement history (Teacy et al. [126] and Feldman and Indelman [30])). Neglecting interactions among object detections (e.g. occlusions) and assuming that data association is known we can split the semantic term into per-detection components

$$\mathbb{P}(\mathcal{Z}_k^s \mid \mathcal{X}_{0:k}, \mathcal{O}, \mathcal{E}, \mathcal{H}_k^-) = \prod_i \mathbb{P}(z_{k,(o_i)}^s \mid \mathcal{X}_{0:k}, o_i, e_i, \mathcal{H}_k^-), \tag{5.4}$$

where by a slight abuse of notation $z_{k,(o_i)}^s$ is an observation corresponding to object o_i. In a factor graph representation, the above components correspond to semantic factors, involving a semantic measurement, variables describing the measured object geometry (o_i) and semantics (e_i) and robot poses when measuring the object.

To control the size of these factors (which can be prohibitively large, e.g. when an entire image, or even classification vector is used as raw semantic measurement), we utilize a feature vector $f_\psi(z_{k,(o_i)}^s)$ of chosen dimension, computed using a feature extractor f_ψ in-lieu of $z_{k,(o_i)}^s$. The feature extractor will be learned simultaneously with the viewpoint-dependent model, as we describe next.

In the following, we define the viewpoint-dependent semantic measurement factor from Eq. (5.4) in Sec. 5.4, then describe how it can be learned in Sec. 5.5.

5.4 Viewpoint-Dependent Model

Assuming that measurements are independent (which generally is an approximation Feldman and Indelman [30]) we can drop the dependence on previous poses and history

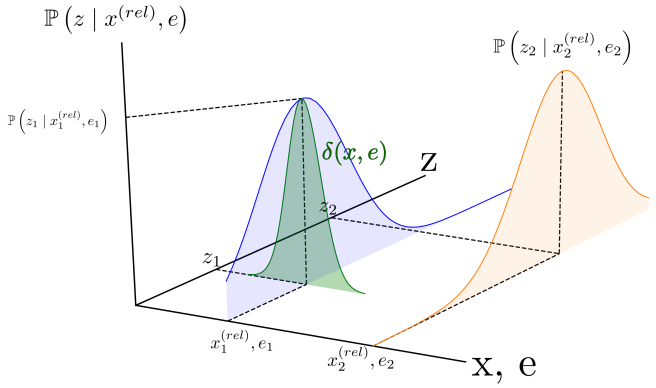
$$\mathbb{P}(z_{k,(o_i)}^s \mid \mathcal{X}_{0:k}, o_i, e_i, \mathcal{H}_k^-) = \mathbb{P}(z_{k,(o_i)}^s \mid \mathcal{X}_k, o_i, e_i). \tag{5.5}$$

We further assume that the semantic measurement only depends on $\mathcal{X}_k^{(rel)}$ - the pose of the robot relative to the object at time k and the semantic description vector e_i, and that any relevant information pertaining to the measurement can be captured with a feature extractor f_ψ from a parametric family, with ψ the set of parameters, acting on the raw measurement - so that we can write

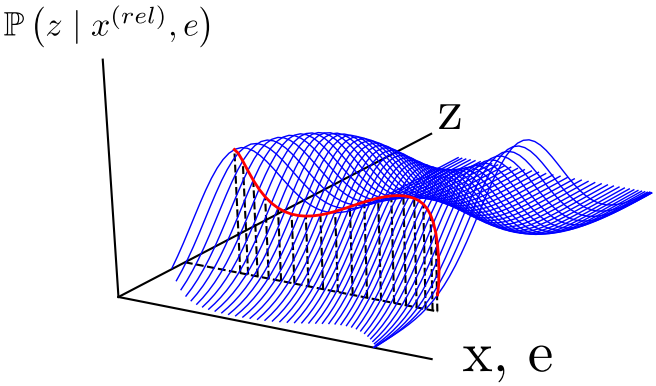
$$\mathbb{P}(z_{k,(o_i)}^s \mid \mathcal{X}_k, o_i, e_i) = \mathbb{P}\left(f_\psi(z_{k,(o_i)}^s) \mid \mathcal{X}_k^{(rel)}, e_i\right). \tag{5.6}$$

While similarly to a viewpoint-dependent model conditioned on discrete class Eq. (3.5) under this model the relative pose is required to be known at training time, the requirement for ground truth classification can be relaxed to a requirement of data association among measurements of the same instance.

The resultant factor contributes a Jacobian of $\dim(f_\psi(z_{k,(o_i)}^s))$ rows, which can be controlled in particular to be less than the $\dim(z_{k,(o_i)}^s)$ which would be contributed by



(a)



(b)

Figure 5.2: Top: $\mathbb{P}(\cdot \mid \mathcal{X}^{(rel)}, e)$ is by-definition a Gaussian over $\mathcal{Z}$ space, with peak at $\mathcal{Z}_i$ for $\mathbb{P}(\cdot \mid \mathcal{X}_i^{(rel)}, e_i)$. We aim to constrain the likelihood surface in the $\mathcal{X}^{(rel)}, e$ directions to be amenable to inference. Bottom: while likelihood is a Gaussian in the direction of the conditioning variables, in the direction of the input (i.e. for given values of conditionals) it is a general function, unconstrained away from data samples, and thus amenable to shaping.

a model over the raw semantic observation $z^s_{k,(o_i)}$.

5.5 Fitting the Model

We developed two approaches for fitting the described models, one is a maximum a-posteriori formulation, the other an energy minimization framework, aiming to solve limitations of the former, which we discuss below. We now describe and present experimental results for the two approaches. In this section we assume a training dataset comprised of measurements (RGB bounding boxes) $\mathcal{Z}^s_{0:k}$. To simplify notations, we drop the superscript and assume a single measurement per time index, i.e. $\mathcal{Z}^s_k \equiv \mathcal{Z}_k = \{z_k\}$ (and thus, use $\mathcal{Z}_k$ and z_k interchangeably). We assume that we are given the corresponding poses relative to the object instance $\mathcal{X}^{(rel)}_{0:k}$ viewed at each time step (might be a different one at each time index k), and the observed object instance identifiers in discrete variables $\beta_{0:k}$.

5.A Joint Posterior Maximization

We formulate the learning of a model of the form Eq. (5.6) as maximizing the joint posterior

$$\underset{\theta,\mathcal{E}_{1:n}}{\arg\max}\ \mathbb{P}_\theta(\mathcal{Z}_{0:k}, \mathcal{E}_{1:n} \mid \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}), \tag{5.7}$$

over object detections (RGB bounding boxes) $\mathcal{Z}_{0:k}$ and the latent representations of the observed object instances $\mathcal{E}_{1:n}$, given poses relative to the object instance $\mathcal{X}_{0:k}^{(rel)}$ viewed at each time step (might be a different one at each time index k), and the observed object instance identifiers as discrete variables $\beta_{0:k}$ (data association). In contrary to the initial derivation from Feldman and Indelman [32] which was based on a VAE (Kingma and Welling [56]) formulation, here we explicitly estimate the latent representation corresponding to each object instance, thus constraining the model to capture viewpoint-dependent variations only through $\mathcal{X}^{(rel)}$.

The posterior in Eq. (5.7) can be developed by chain rule as

$$\begin{aligned}\mathbb{P}(\mathcal{Z}_{0:k}, \mathcal{E}_{1:n} \mid \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}) = \\ \mathbb{P}(\mathcal{Z}_{0:k} \mid \mathcal{E}_{1:n}, \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}) \cdot \mathbb{P}(\mathcal{E}_{1:n} \mid \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}).\end{aligned} \tag{5.8}$$

Here, the first term can be decomposed into a product of semantic observation factors assuming measurement independence (which generally is an approximation Feldman and Indelman [30])

$$\mathbb{P}(\mathcal{Z}_{0:k} \mid \mathcal{E}_{1:n}, \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}) = \prod_i \mathbb{P}_\theta(\mathcal{Z}_i \mid e_{\beta_i}, \mathcal{X}_i^{(rel)}). \tag{5.9}$$

In practice, we learn to predict a Gaussian given the object latent representation vector and relative pose, i.e.

$$\mathbb{P}_\theta(\mathcal{Z}_i \mid e_{\beta_i}, \mathcal{X}_i^{(rel)}) = \mathcal{N}\left(\mu_\theta(e_{\beta_i}, \mathcal{X}_i^{(rel)}), \Sigma_\theta(e_{\beta_i}, \mathcal{X}_i^{(rel)})\right), \tag{5.10}$$

with diagonal covariance Σ_θ. We believe that a unimodal distribution is adequate in this case since the model describes the measurement of a particular object instance from a given relative pose (disregarding occlusions).

The second term of Eq. (5.A) in absence of measurements degenerates into a prior over the latent representations

$$\mathbb{P}(\mathcal{E}_{1:n} \mid \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}) = \mathbb{P}(\mathcal{E}_{1:n}), \tag{5.11}$$

which in our experiments we assume to be proportional to either Contrastive (Chopra, Hadsell, and LeCun [15]) or N-Pairs loss (Sohn [114] and Pirk et al. [100]) to facilitate learning.

Finally, to initialize the latent representation variables during inference, we learn a model predicting the latent representation given an object detection:

$$\mathbb{P}_\phi\left(e \mid \mathcal{Z}_k\right) = \mathcal{N}\left(\mu_\phi(\mathcal{Z}_k), \Sigma_\phi(\mathcal{Z}_k)\right), \tag{5.12}$$

again with diagonal covariance. This model corresponds to the encoder in the formulation of Feldman and Indelman [32]. In the current formulation this model is learned separately as a post-processing step, after the model Eq. (5.10) is fit. This model is learned so as to maximize the likelihood of the object representations learned in Eq. (5.7):

$$\phi^* \leftarrow \arg\max_\phi \ \mathbb{P}(\mathcal{Z}_{0:k} \mid \mathcal{E}_{1:n}) = \arg\max_\phi \ \prod_i \mathbb{P}_\phi\left(e_i \mid \mathcal{Z}_i\right), \tag{5.13}$$

the latter equality true assuming independence of measurements and uninformative prior $\mathbb{P}(e_i)$, for each individual e_i.

5.A.1 Experimental Results

We performed experiments using the BigBIRD dataset Singh et al. [111] to validate the approach described above to learning a viewpoint-dependent model over a latent semantic space that would enable continuous joint semantic and geometric inference as in Eq. (5.2). In Sec. 5.B we investigate the learned models, we present results of inference over object location and robot track (localization only for both). We further present and discuss results of sensitivity of the estimation process to the initialization for individual views in Sec. 5.D.

5.B Learned Models

We fit a viewpoint-dependent observation model following Sec. 5.4 to images and ground truth relative poses of objects from the BigBIRD dataset Singh et al. [111]. Each image is cropped using the provided object mask, and resized to 32x32x3. Fig. 5.1a visualizes the resultant latent space using tSNE Maaten and Hinton [74]. On the left, latent vectors (encoder output) for the various viewpoints for each object show as well-localized clusters of the same color, a single cluster per object - although distinct clusters of very similar colors are present, in practice they correspond to different objects. Right: input images corresponding to the sub-region of the tSNE space highlighted on the right. The images are nearest neighbors to points of an axis-aligned grid in the tSNE space. The same clustered structure of the embedding space shows, with all viewpoints of an object appearing grouped. The observed structure of the embedded space is segregated among clusters corresponding to object instances and is expected to allow the envisioned reasoning using the learned model.

Fig. 5.4c shows mean images predicted by the model on validation data. Fig. 5.4a

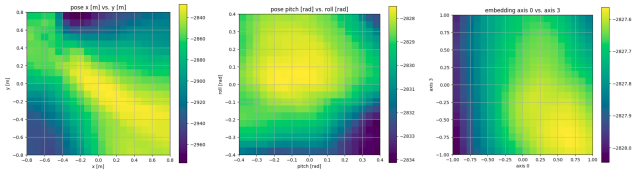


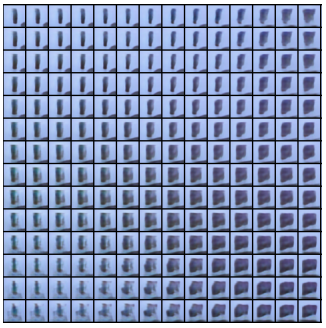
Figure 5.3: Log likelihood over pairs of axes: location (left), orientation (center), semantic representation (right). Coordinates correspond to offset w.r.t. ground truth, i.e. the value at the origin is the likelihood of a ground truth sample, while values away from the origin correspond to values of the likelihood function at locations with varying error in the inputs w.r.t. ground truth.

and Fig. 5.4b show mean predictions when locally varying the latent representation (while keeping $\mathcal{X}^{(rel)}$ constant) and the relative pose (while keeping the latent representation constant) respectively, corresponding to a validation example and the representation learned for the corresponding object. In Fig. 5.4b, images predicted with values of $\mathcal{X}^{(rel)}$ varying over a grid while keeping the latent representation constant exhibit strong regular viewpoint changes, unlike the predictions in Fig. 5.4a when varying the latent representation which appear relatively random - implying that the learned model disentangles viewpoint dependence from object "semantic" representation. Fig. 5.3 again demonstrates the viewpoint dependence for an example view. The plots show the log values of the learned likelihood model for a given example view $\mathcal{Z}$ with relative pose $\mathcal{X}^{(rel)}$ and the corresponding learned representation e, while varying $\mathcal{X}^{(rel)}$ on a grid around the ground truth value along combinations of axes. A peak around the origin implies that given an erroneous initial guess, inference with the learned model would recover the correct values. The peaks are close but not exactly at the origin, indicating that the learned model might be improved.

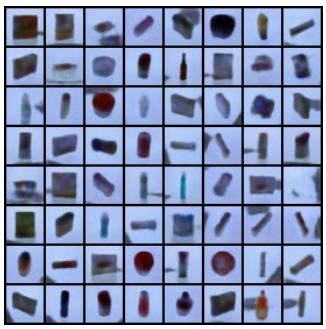
5.C Inference

We conducted initial simulation experiments with BigBIRD imaging data towards evaluating our approach in inference. In our experiments, a robot moves along a simulated (2D) track around a BigBIRD object (Fig. 5.5a, ground truth track in plain black) obtaining corresponding detection images (Fig. 5.5b) of the object (located at the origin in the plot). In this initial experiment, the goal is to infer the correct robot track - i.e. we focus on localization only, with fixed orientation and semantic representation - given a noisy initial robot track (dashed grey in the plot) and object localization (not shown) estimate, as well as a weak and noisy odometry (dead reckoning track shown in dashed red) and priors on the object and robot initial locations. Further, only odometry and the learned semantic observation model are used. In this experiment, the inference is done in "batch mode", simultaneously for the entire robot track and object location. The track estimate after incorporating the observed images using the learned viewpoint-dependent semantic observation model (blue dashed line) appears closer to the ground truth track. Note that the semantic factors are in general complementary

(a) Mean predictions generated for random values of the latent representation while keeping the relative pose constant.



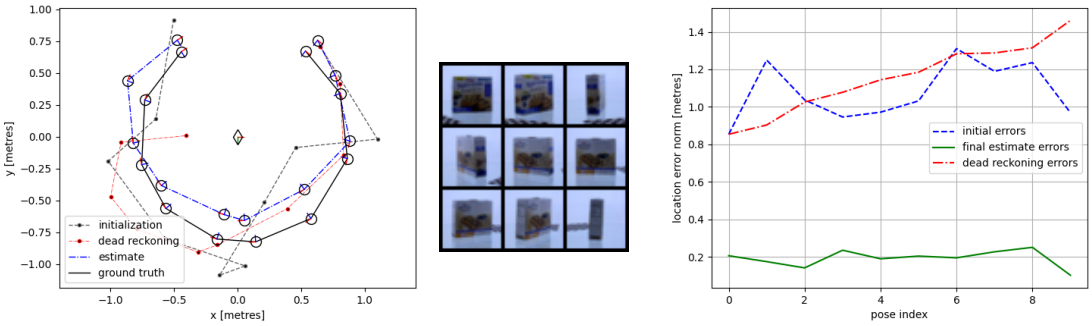
(b) Mean predictions generated on a grid of relative poses while keeping the latent representation constant.



(c) Example mean predictions generated by the learned model for object views not seen during training.

Figure 5.4: Right: predicted mean images for validation examples. Center: predictions obtained for input pose varying over a grid display structured viewpoint changes. Left: predictions obtained for random vectors in the vicinity of the representation learned for a class vary relatively randomly, implying disentanglement w.r.t. relative pose.

to standard geometric observation models (however in this experiment, no geometric observation models are used - only the semantic model and odometry). The corresponding plot on the right Fig. 5.5c further illustrates this, showing the norm of the localization errors along track indexes: both of the initial estimate (dashed blue) and the dead reckoning track (dashed red) show significantly higher error than the estimate after incorporating the learned semantic factor (plain green).

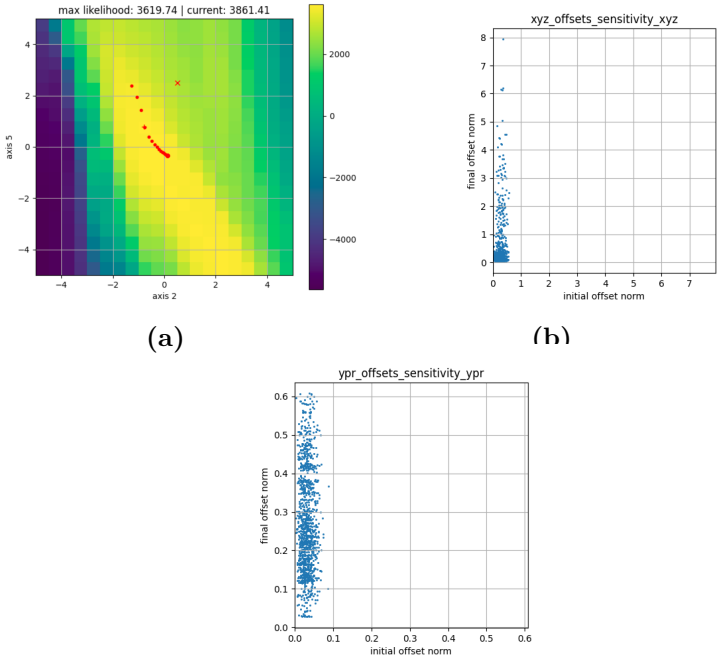


(a) **(b)** **(c)**

Figure 5.5: **Left:** Example scenario for the experiment in Sec. 5.C. The ground truth robot track around the object (of type "nutrigrain_harvest_blueberry_bliss", at the origin) is shown in plain black. The initial track estimate is shown in grey, and the "dead-reckoning" track - in red. The robot track estimated after incorporating the learned semantic observation factors (in blue) is closer to the ground truth. **Center:** Example object views along the track. **Right:** Error evolution over robot track indexes corresponding to the scenario in Fig. 5.5a. Both the initial estimate (dashed blue) and the dead reckoning track (dashed red) show significantly higher error than the estimate after incorporating the learned semantic factor (plain green).

5.D Sensitivity Experiments

To statistically assess the performance of the learned model in inference, we present results of maximum - likelihood estimation of separate parameter groups (robot relative location, orientation, embedding) starting from random initialization in the vicinity of ground truth samples (parameters that are not optimized upon are fixed to their ground truth values). Ground truth samples were taken for various objects, one of which was used for the plots in Fig. 5.5. Fig. 5.6a shows an example path taken by the likelihood optimization in the section over two axes (the origin corresponds to the ground truth / correct values). Fig. 5.6b shows sensitivity experiment results when optimizing over the relative location axes (only, i.e. the rest are fixed to the ground truth) - the x axis corresponds to the norm of the initial error, the y axis corresponds to the norm of the estimate error (we would wish all points to lie on the x axis, i.e. have zero final estimate error). Points that lie below the diagonal (x=y) are experiments where the estimation process reduced the error norm, the case of the majority of the points. Fig. 5.6c displays the analogous plot when optimizing only over the orientation



(a) **(b)**

(c)

Figure 5.6: **Left:** Example optimization track. The color levels correspond to values of the log likelihood of Eq. (5.6) in the vicinity of a ground truth example (origin) along two optimization axes. The red cross mark denotes the (biased) initialization, while a small '+' sign denotes the empirical max likelihood location - the slight offset w.r.t. the origin indicates an imperfection of the learned model. The red dots denote the track taken by the optimization along the corresponding two axes. **Center:** Sensitivity to initial errors in location axes - for a large proportion of the random initializations, optimizing over location axes reduces estimate error. **Right:** Sensitivity to initial errors in orientation axes (output once every few iterations) - the estimation is very sensitive to even small errors in orientation. In this case, we expect orientation to be inferred using geometric factors.

axes (relative to the object). The plot indicates that this optimization is extremely unstable, with small initial errors often resulting in large estimate errors. This may be related to that the current formulation does not carry significant camera orientation information as long as the object is within the camera field of view, and consequently orientation axes are extraneous degrees of freedom in the parametrization (meaning, the "ground-truth" training data lies on a manifold in this higher-dimensional space), which might result in instability outside of this manifold. Further, we currently observe somewhat similar behavior when optimizing over the semantic representation - which prompts us to attempt to reduce the dimension of both the embedding and the semantic measurements.

5.D.1 Discussion

As mentioned in the previous section, optimization results with model learned with the MAP formulation presented were unstable. This is supported by Fig. 5.3, where

for two of the three pairs of axes the maximum likelihood is not located at the origin (ground truth), but away from it. Further, it may be seen that the likelihood function is not convex, occasionally causing optimization to diverge, or become stuck in a local minimum. This behavior of the learned viewpoint-dependent model led us to refine the learning approach, in order to attempt to shape the likelihood function in a way facilitating inference - which we describe next.

5.E Energy Minimization Approach

As before we parametrize the model of Eq. (5.6) as a (spatially-varying) Gaussian, that is

$$\mathbb{P}\left(\cdot \mid \mathcal{X}_k^{(rel)}, e_i\right) = N\left(\cdot;\ \mu_\theta(\mathcal{X}_k^{(rel)}, e_i),\ \Sigma_\theta(\mathcal{X}_k^{(rel)}, e_i)\right), \tag{5.14}$$

with θ - the vector of model parameters. We set the covariance $\Sigma_\theta(\mathcal{X}_k^{(rel)}, e_i)$ to be a constant diagonal matrix. We formulate a learning objective over the parameters of the model θ and the feature extractor ψ and a set of the latent representations of observed object instances $\mathcal{E}_{1:n}$

$$J\left(\theta, \mathcal{E}_{1:n}, \psi\right) = J_d\left(\theta, \mathcal{E}_{1:n}, \psi\right) + J_r(\mathcal{E}_{1:n}) + J_f\left(\psi\right), \tag{5.15}$$

Where J_r and J_f act as priors/regularizers on the representation and the feature extractor respectively, and J_d is a data term, all of which we describe next.

Data term: for training example i consisting of raw measurement $\mathcal{Z}_i$, ground truth relative pose $\mathcal{X}_i^{(rel)}$ and instance identifier / data association β_i, we would like the likelihood function induced by the model $g(\mathcal{X}^{(rel)}, e) \doteq \mathbb{P}(z_i^s \mid \mathcal{X}^{(rel)}, e)$ to be amenable to inference, that is - we would like a log-likelihood accent method which starts at an erroneous initial estimate $\mathcal{X}_i^{(rel)} + \Delta\mathcal{X}, e_{\beta_i} + \Delta e$ and observation $\mathcal{Z}_i$ to recover the ground truth values $\mathcal{X}_i^{(rel)}, e_{\beta_i}$, at least for small $\Delta\mathcal{X}, \Delta e$. To this end we define the data term for training example i, denoted $J_d^{(i)}$, to be a weighted $\mathcal{L}_1$ distance between the model-induced likelihood $\mathbb{P}(\mathcal{Z}_i \mid \mathcal{X}_i^{(rel)} + \Delta\mathcal{X}, e_i + \Delta e)$ and a target function $\delta\left(\Delta\mathcal{X}, \Delta e\right)$ possessing the above quality. In particular, we choose δ to be a Gaussian distribution with small diagonal covariance $\delta\left(\cdot\right) = N\left(\cdot \mid 0, \varepsilon I\right)$ (resulting in a quadratic log likelihood), which allows us to write

$$\begin{aligned} J_d^{(i)}(\theta, \mathcal{E}_{1:n}, \psi) = \mathop{\mathbb{E}}_{\Delta\mathcal{X}, \Delta e \sim \delta} \Big\{ \Big| & \log \mathbb{P}(z_i^s \mid \mathcal{X}_i^{(rel)} + \Delta\mathcal{X}, e_i + \Delta e) \\ & - \log \delta\left(\Delta\mathcal{X}, \Delta e\right) \Big| \Big\}. \end{aligned} \tag{5.16}$$

Note that the above resembles (up to absolute value) a KL divergence. However, the likelihood function is only a distribution in the z_i^s argument, but not in $\mathcal{X}_i^{(rel)}, e_i$ - thus a distance between distributions cannot be used. This is illustrated in Fig. 5.2b: while for every set value of $\mathcal{X}^{(rel)}$ and e the likelihood $\mathbb{P}(\cdot \mid \mathcal{X}^{(rel)}, e)$ is a Gaussian distribution

(along the z^s axis, curves in blue), for a set z^s the likelihood (as function of $\mathcal{X}^{(rel)}$ and e, curve in red) need not be one, and in particular is not constrained to sum to 1.

The role of the δ function in Eq. (5.16) is further exemplified in Fig. 5.2a: the aim is to constrain the likelihood function (in the $\mathcal{X}^{(rel)}, e$ input space) in the vicinity of each training example. In blue and in orange are the Gaussians (in the z^s space) predicted for $\mathcal{X}_1^{(rel)}, e_1$ and $\mathcal{X}_2^{(rel)}, e_2$ respectively. We want their maximum likelihood to occur at z_1^s, z_2^s (i.e. for $\Delta\mathcal{X}^{(rel)}, \Delta e = 0$) and for the likelihood function locally to resemble δ to allow for inference - both of which are expressed in Eq. (5.16). The locality of the constraint is provided by the expectation over δ, whereby far samples make negligibly small contributions. Finally, the data loss term is obtained by averaging the per-example loss terms: $J_d = \frac{1}{N_D} J_d^{(i)}$, for N_D - the number of training examples.

Representation term: acts as a prior on the latent representations. In our experiments we assume this term to be proportional to either Contrastive (Chopra, Hadsell, and LeCun [15]) or N-Pairs loss (Sohn [114] and Pirk et al. [100]) to facilitate learning.

Feature Extractor term: without any regularization, minimization of the previous two terms admits trivial solutions, such as all raw measurements being mapped to the same feature vector (e.g., 0). To avoid such solutions, we define a regularization term encouraging the feature extractor to preserve *distance ordering* among raw measurements. That is, for every triplet of raw measurements $(\mathcal{Z}_i, \mathcal{Z}_j, \mathcal{Z}_k)$ in the training set, we demand that if $d(\mathcal{Z}_i, \mathcal{Z}_j) < d(\mathcal{Z}_i, \mathcal{Z}_k)$ then $d(z_i^s, z_j^s) < d(z_i^s, z_k^s)$ by at least some small margin (hinge loss), that is (denoting the margin as ε)

$$J_f(\psi) \propto \sum_{(\mathcal{Z}_i, \mathcal{Z}_j, \mathcal{Z}_k)} \mathbb{1}\left\{d(\mathcal{Z}_i, \mathcal{Z}_j) < d(\mathcal{Z}_i, \mathcal{Z}_k)\right\} \cdot \max\left(d(z_i^s, z_j^s) - d(z_i^s, z_k^s) + \epsilon, 0\right). \tag{5.17}$$

In our experiments we use the squared difference norm as the metric both in the measurements' space and in the feature space. While this loss term formulation is based on an observations' space metric, which does not necessarily reflect semantic relations - we believe that the constraints it imposes on the feature extractor are sufficiently weak so as not to interfere with the optimization of the data term. In practice, both the representation and the feature extractor term are small compared to the data term, but are successfully minimized concurrently to it, without need for any additional weighting / regularization.

5.E.1 Experimental Results

Similar to Fig. 5.3 we display rasters of the value of the likelihood function obtained using a model learned with the energy-based scheme Sec. 5.E. Each raster shows the variation of the likelihood value for varying offsets with respect to a ground truth sample: in particular, the value at the center of the raster corresponds to the likelihood of the ground truth (expected to be the likelihood maximum). Fig. 5.7 shows the

variation of the likelihood value as function of pose offset along the different pairs of dimensions: x,y,z, yaw, pitch, roll. Fig. 5.8 does the like for the first 6 dimensions of the latent representation vector corresponding to the ground truth sample. The maximum of the visualized likelihood appears to be at the ground truth (i.e. at the origin for all rasters), and appears to be convex - which is favorable to optimization using the factor, and qualitatively significantly better than results for model learned with MAP approach Fig. 5.3.

This is a very initial validation of the new scheme, further validation with inference experiments is required.

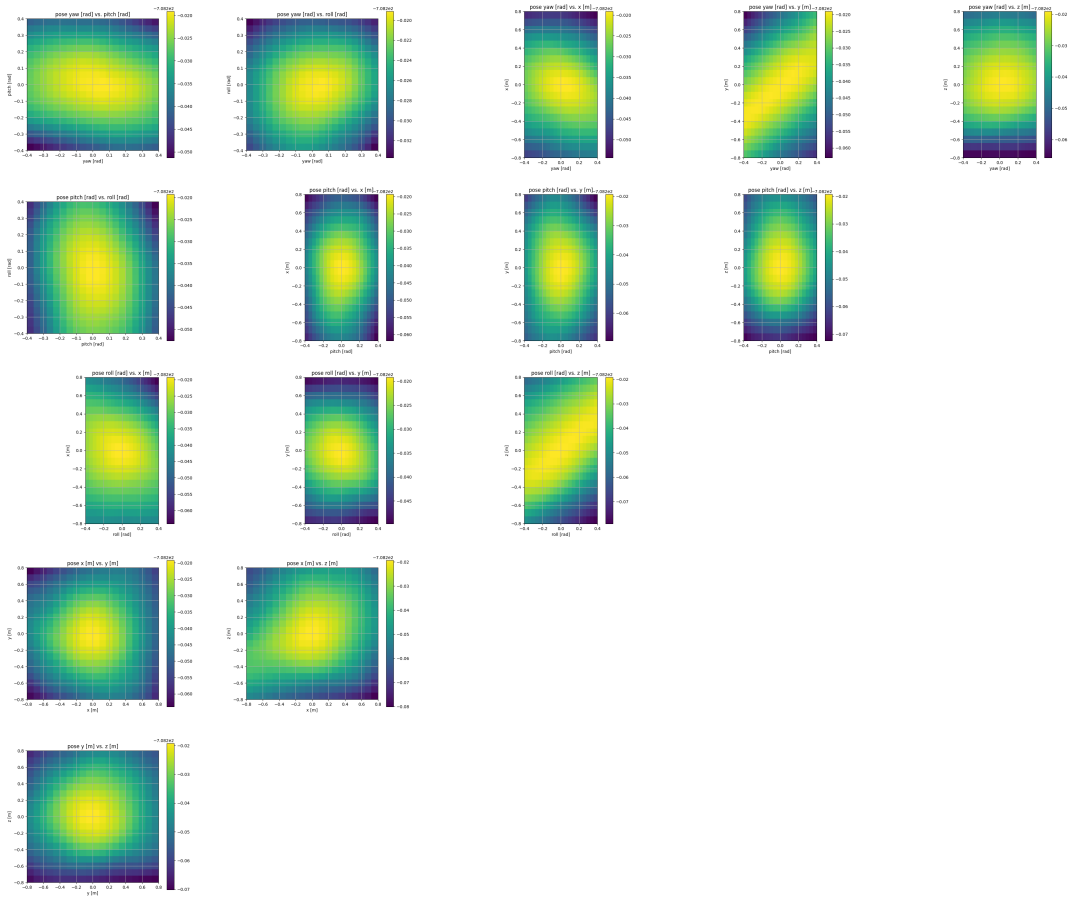


Figure 5.7: Log likelihood over pairs of pose axes for model learned using the energy-based approach.

We have thus described how the semantic observation models required for the inference in Eq. (5.2) can be learned. In the next section, we address a few questions related to the applicability of the above in online inference.

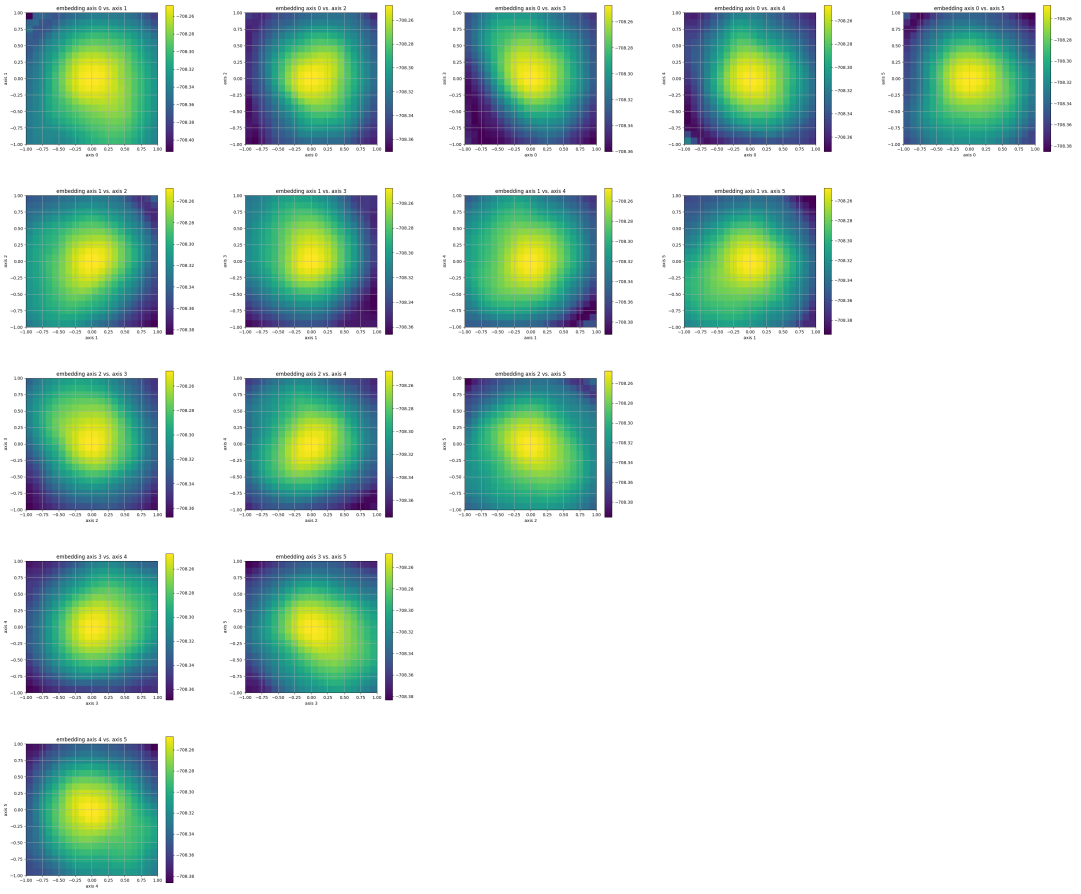


Figure 5.8: Log likelihood over pairs of latent representation axes for model learned using the energy-based approach.

5.6 Feasibility of the Approach for Online Semantic SLAM

In the following we address the applicability of our approach in the context of three specialized sub-tasks of online semantic SLAM, namely: inference of semantics (Sec. 6.A), localization and mapping (Sec. 6.B), online operation (Sec. 6.C).

6.A "Grounding" of the Semantic Representation

Our formulation thus far made no use of semantic information beyond instance-level data association, yet - the latent representation variables would clearly carry semantic information (as long as instance - level association, or object images can be inferred back from them, as in Eq. (5.10). In other words, the described approach could allow to perform semantic mapping without need for ground truth class information, provided that the learned model holds, i.e. that objects encountered at test time are not very different from those seen during training.

Still, for the map to be interpretable by a human operator, given a (task-specific

and possibly non-unique) set of candidate classes, a correspondence to the latent space needs to be established. Pirk et al. [100] report the learned latent space (with a different objective function) to reflect semantic structure, i.e. objects that are similar in appearance (and thus, in their case - in function) tend to emergently be close in the learned space (based on appearance only). In such an ideal case, manual tagging of a few example images (which through the encoder induces tagging of a region of the latent space) followed by a nearest neighbor search in the latent space could likely produce satisfactory classification results.

In the general case however, classes of interest could encompass objects with significantly varying appearance which will not be close in the latent space unless explicitly forced. In such a case we can assume a limited amount of classification-tagged instances (a single tagged image implies a tagged instance because of the known data association assumption). Denoting the available classification data collectively as $\mathcal{C}$, we can incorporate it in the training process Eq. (5.7) via conditioning:

$$\begin{aligned} &\mathbb{P}(\mathcal{Z}_{0:k}, \mathcal{E}_{1:n} \mid \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}, \mathcal{C}) = \\ &\mathbb{P}(\mathcal{Z}_{0:k} \mid \mathcal{E}_{1:n}, \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}, \mathcal{C}) \cdot \mathbb{P}(\mathcal{E}_{1:n} \mid \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}, \mathcal{C}) = \\ &\mathbb{P}(\mathcal{Z}_{0:k} \mid \mathcal{E}_{1:n}, \mathcal{X}_{0:k}^{(rel)}, \beta_{0:k}) \cdot \mathbb{P}(\mathcal{E}_{1:n} \mid \beta_{0:k}, \mathcal{C}), \end{aligned} \tag{5.18}$$

where the classification in the first term can be dropped because we model the latent representation to be a sufficient statistic for the observation, and we drop the relative poses from the second term as the object semantic representation is independent of them in absence of measurements. The second term can now be split into a product

$$\mathbb{P}(\mathcal{E}_{1:n} \mid \beta_{0:k}, \mathcal{C}) = \prod_i \mathbb{P}(e_i \mid \mathcal{C}_i), \tag{5.19}$$

where for simplicity of notation $\mathcal{C}_i = \emptyset$ when class information is unavailable for the corresponding instance. Where class information is available, the terms $\mathbb{P}(e_i \mid \mathcal{C}_i)$ can be modeled as Gaussians with learnable parameters (μ_i, Σ_i), providing a mapping from categories to the latent representation. Maximizing the joint posterior Eq. (5.7) (conditioned on classification information as in Eq. (6.A)) thus simultaneously produces this mapping. A slightly more detailed derivation, explaining how the Contrastive / N-Pairs loss can still be used (similar to Feldman and Indelman [32]) will be provided in the supplementary material.

6.B Improving Localization

A similar principle as was mentioned in the previous clause to adapt the latent space to facilitate classification, could be applied to attempt to facilitate localization. Formally, analogously to the above Sec. 6.A and in the notations of Eq. (5.1), we could set the

optimization objective to be the joint posterior

$$\mathbb{P}(\mathcal{X}_{0:k}, \mathcal{L}, \mathcal{O}, \mathcal{Z}^s_{0:k}, \mathcal{E}_{1:n} \mid \mathcal{Z}^g_{0:k}, \mathcal{U}_{0:k-1}) = \tag{5.20}$$
$$\mathbb{P}(\mathcal{X}_{0:k}, \mathcal{L}, \mathcal{O}, \mid \mathcal{H}_k) \cdot \mathbb{P}(\mathcal{Z}^s_{0:k}, \mathcal{E}_{1:n} \mid \mathcal{Z}^g_{0:k}, \mathcal{U}_{0:k-1}).$$

Here the second term in Eq. (6.B) can be developed similarly to Eq. (5.A), while the former term is the localization problem from Eq. (5.1) with a twist: here the model parameters, participating in the semantic factors are variable too, giving a scheme for concurrent learning and inference.

6.C Computational Aspects

The results in Sec. 5.5 were obtained for RGB detections of size 32x32 and embedding size 128 (a 24x reduction in dimensionality). Since those are detections rather than entire images, a reasonable amount of object appearance detail can in general be captured and consequently modeled. However, directly using the raw detections as measurements would imply a prohibitive factor size of ≈3000 for each semantic observation. One possible approach we currently are considering to tackle the dimensionality problem is learning a feature extractor simultaneously with the viewpoint-dependent model to reduce the measurement dimensionality, while fitting the viewpoint-dependent model to predict the lower-dimensional measurement. Ideally, the dimensionality-reduced measurement would still retain the semantic information relevant to the task, as well as be sensitive to viewpoint changes, to allow for precise localization inference.

Chapter 6

Conclusion and Future Directions

We developed approaches for semantic perception addressing the challenges associated with fusing semantic measurements. We utilized learned viewpoint-dependent models to capture the spatial variation of semantic measurements, and showed their benefit for disambiguation of object classification and of data association, as well an approach to learning a continuous latent object representation allowing a continuous formulation of object SLAM.

We started by formulating an approach to the classification of a single object by a moving robot using measurements from a deep learning classifier, carrying model uncertainty. We defined per-class viewpoint-dependent models as Gaussian Processes over viewpoint relative to an object of a given class, thus capturing spatial variation as well as inter-viewpoint correlations of classifier (mean) response. We then developed an inference scheme using those models, which accounts for localization uncertainty and model uncertainty, as well as inter-viewpoint correlations. We validated our approach in MATLAB simulation, in a 3D environment powered by Unreal Engine, and finally with real-world data - demonstrating graceful aggregation of information, as opposed to the expected over-confident and as a consequence frequently failing outputs of baseline methods.

We then addressed the more general case of multiple objects developing an object SLAM approach that is data-association aware, i.e. maintaining a belief joint with data association. We addressed data association of object measurements by maintaining the corresponding hypotheses, utilizing in inference a per-class viewpoint-dependent model capturing the spatial variation of a semantic measurement (classification probability vector). We demonstrated theoretically and empirically in a simulation that the viewpoint-dependent model improved data association disambiguation in a way that allowed the hypotheses to be maintained in practice (while hypotheses lower than a threshold are discarded) with their number not growing exponentially, which would be the theoretical maximum.

In the third part of the research we proposed an approach allowing to reformulate object SLAM as continuous inference over a learned latent semantic representation

space. The representation space is induced by learning a viewpoint-dependent model predicting object measurements conditioned on the latent semantic representation corresponding to the object (as well as the viewpoint relative to the object). The learned model can then be used as a factor in inference. This approach directly addresses two challenges of semantic perception: discrete representation (a continuous one is proposed) and scarcity of ground truth data - as the training of the model does not require semantic ground-truth (e.g. true object class), rather - only data association of multiple observations for each instance. We explored two ways to train such a model: first, as a MAP formulation. Second, through an energy-minimization approach, aiming to shape the model in a way that would facilitate its use for inference - with a novel objective for learning a model for inference. We presented experiments, demonstrating learned models and their use for inference in simulation, using images from a real-world dataset.

6.1 Future Directions

While multiple recent methods turn to constructing dense semantic maps (contrary to the approaches in this thesis) - arguing those contain fuller information about the environment which is needed for motion planning and manipulation - the construction of such maps is costly in computational resources, and while done in real time, it is on medium to high grade systems. At the same time, as argued in Davison [19] and Davison and Ortiz [20], use cases for spatial perception require it to operate under limiting compute time and power constraints, as part of embedded systems. In such settings sparse, topological (e.g. object-based) approaches are more likely to be feasible, and thus are still of interest.

One direction that can be explored is can the proposed semantic descriptors be used to construct a hierarchical (topological) representation, that would enable efficient inference and planning, mirroring the representation of e.g. Rosinol et al. [107], however not requiring semantic ground truth for training.

Further, an assumption underlying most past work on SLAM is that the mapped environment is static. Some methods relax this assumption by explicitly ignoring in the mapping process elements of the environment that are suspected to be in motion (e.g. when they are outliers with respect to data association or correspond to objects of a class known to be dynamic, such as a human or an animal). However, if moving objects need to be incorporated in the situational awareness, this implies primarily a difficult data-association problem, requiring tracking object instances. Unsupervised learning of object descriptors as proposed in the third part of this thesis may be beneficial and can be explored for solving data-association in this case.

Finally, a fundamental limitation of the method described for learning the semantic descriptors is the requirement for objects to be correctly detected. A possible direction can be attempting to relax this requirement.

Bibliography

[1] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Kosecka, and Alexander C. Berg. "A Dataset for Developing and Benchmarking Active Vision". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

[2] N. Atanasov, B. Sankaran, J.L. Ny, G. J. Pappas, and K. Daniilidis. "Nonmyopic View Planning for Active Object Classification and Pose Estimation". In: *IEEE Trans. Robotics* 30 (2014), pp. 1078–1090.

[3] Alper Aydemir, Adrian N Bishop, and Patric Jensfelt. "Simultaneous object class and pose estimation for mobile robotic applications with minimalistic recognition". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2010, pp. 2020–2027.

[4] Alexander Baikovitz, Paloma Sodhi, Michael Dille, and Michael Kaess. "Ground encoding: Learned factor graph-based models for localizing ground penetrating radar". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 5476–5483.

[5] Dana H Ballard. "Generalizing the Hough transform to detect arbitrary shapes". In: *Pattern Recognition* 13.2 (1981), pp. 111–122.

[6] S. Y. Bao, M. Bagra, Y-W. Chao, and S. Savarese. "Semantic structure from motion with points, regions, and objects". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2703–2710.

[7] A. Barbu and S.-C. Zhu. "Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities". In: *IEEE Trans. Pattern Anal. Machine Intell.* 27.8 (Aug. 2005), pp. 1239–1253.

[8] Israel Becerra, Luis M Valentín-Coronado, Rafael Murrieta-Cid, and Jean-Claude Latombe. "Reliable confirmation of an object identity by a mobile robot: A mixed appearance/localization-driven motion approach". In: *Intl. J. of Robotics Research* 35.10 (2016), pp. 1207–1233.

[9] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. "CodeSLAM — learning a compact, optimisable representation for dense visual SLAM". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2560–2568.

[10] S. Bowman, N. Atanasov, K. Daniilidis, and G. Pappas. “Probabilistic data association for semantic SLAM”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1722–1729.

[11] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian D Reid, and John J Leonard. “Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age”. In: *IEEE Trans. Robotics* 32.6 (2016), pp. 1309–1332.

[12] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam”. In: *IEEE Trans. Robotics* 37.6 (2021), pp. 1874–1890.

[13] L. Carlone, A. Censi, and F. Dellaert. “Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2014, pp. 2667–2674.

[14] Luca Carlone and Giuseppe C Calafiore. “Convex Relaxations for Pose Graph Optimization with Outliers”. In: *IEEE Robotics and Automation Letters (RA-L)* 3.2 (2018), pp. 1160–1167.

[15] Sumit Chopra, Raia Hadsell, and Yann LeCun. “Learning a similarity metric discriminatively, with application to face verification”. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. IEEE. 2005, pp. 539–546.

[16] S. Choudhary, A. Trevor, H. I. Christensen, and F. Dellaert. “SLAM with object discovery, modeling and mapping”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2014, pp. 1018–1025.

[17] Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Henrik I Christensen, and Frank Dellaert. “Multi Robot Object-based SLAM”. In: *Intl. Sym. on Experimental Robotics (ISER)*. 2016.

[18] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. “Deepfactors: Real-time probabilistic dense monocular slam”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 721–728.

[19] Andrew J Davison. “FutureMapping: The computational structure of spatial AI systems”. In: *arXiv preprint arXiv:1803.11288* (2018).

[20] Andrew J Davison and Joseph Ortiz. “FutureMapping 2: Gaussian belief propagation for spatial AI”. In: *arXiv preprint arXiv:1910.14139* (2019).

[21] F. Dellaert. *Factor Graphs and GTSAM: A Hands-on Introduction*. Tech. rep. GT-RIM-CP&R-2012-002. Georgia Institute of Technology, Sept. 2012.

[22] F. Dellaert and M. Kaess. “Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing”. In: *Intl. J. of Robotics Research* 25.12 (Dec. 2006), pp. 1181–1203.

[23] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. “Learning to generate chairs, tables and cars with convolutional networks”. In: *IEEE Trans. Pattern Anal. Machine Intell.* 39.4 (2016), pp. 692–705.

[24] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. “Model globally, match locally: Efficient and robust 3D object recognition”. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 998–1005.

[25] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry”. In: *IEEE Trans. Pattern Anal. Machine Intell.* 40.3 (2017), pp. 611–625.

[26] Jakob Engel, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-scale direct monocular SLAM”. In: *European Conf. on Computer Vision (ECCV)*. 2014, pp. 834–849.

[27] Hafez Farazi and Sven Behnke. “Online visual robot tracking and identification using deep LSTM networks”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 6118–6125.

[28] E. I. Farhi and V. Indelman. “iX-BSP: Belief Space Planning through Incremental Expectation”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2019.

[29] Y. Feldman and V. Indelman. “Bayesian Viewpoint-Dependent Robust Classification under Model and Localization Uncertainty”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018.

[30] Y. Feldman and V. Indelman. “Spatially-Dependent Bayesian Semantic Perception under Model and Localization Uncertainty”. In: *Autonomous Robots* (2020).

[31] Y. Feldman and V. Indelman. “Towards Robust Autonomous Semantic Perception”. In: *Workshop on Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding, in conjunction with IEEE International Conference on Robotics and Automation (ICRA)*. May 2018.

[32] Y. Feldman and V. Indelman. “Towards Self-Supervised Semantic Representation with a Viewpoint-Dependent Observation Model”. In: *Workshop on Self-Supervised Robot Learning, in conjunction with Robotics: Science and Systems (RSS)*. July 2020.

[33] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. “Multi-target tracking using joint probabilistic data association”. In: *Proc. 19th IEEE Conf. on Decision & Control*. 1980.

[34] Yarin Gal. “Uncertainty in Deep Learning”. PhD thesis. University of Cambridge, 2017.

[35] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Intl. Conf. on Machine Learning (ICML)*. 2016.

[36] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. “Deep bayesian active learning with image data”. In: *Intl. Conf. on Machine Learning (ICML)*. JMLR. org. 2017, pp. 1183–1192.

[37] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigal, and J. González. “Multi-Hierarchical Semantic Maps for Mobile Robotics”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2005, pp. 3492–3497.

[38] Sourav Garg, Niko Sünderhauf, Feras Dayoub, Douglas Morrison, Akansel Cosgun, Gustavo Carneiro, Qi Wu, Tat-Jun Chin, Ian Reid, Stephen Gould, et al. “Semantics for robotic mapping, perception and interaction: A survey”. In: *Foundations and Trends® in Robotics* 8.1–2 (2020), pp. 1–224.

[39] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 580–587.

[40] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2014, pp. 2672–2680.

[41] J.-S. Gutmann and K. Konolige. “Incremental Mapping of Large Cyclic Environments”. In: *IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*. 1999, pp. 318–325.

[42] Ali Harakeh. “Estimating and Evaluating Predictive Uncertainty in Deep Object Detectors”. PhD thesis. University of Toronto (Canada), 2021.

[43] Ali Harakeh, Michael Smart, and Steven L Waslander. “Bayesod: A bayesian approach for uncertainty estimation in deep object detectors”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 87–93.

[44] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural networks* 4.2 (1991), pp. 251–257.

[45] M. Hsiao and M. Kaess. “MH-iSAM2: Multi-hypothesis iSAM using Bayes Tree and Hypo-tree”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2019.

[46] Nathan Hughes, Yun Chang, and Luca Carlone. “Hydra: A Real-time Spatial Perception Engine for 3D Scene Graph Construction and Optimization”. In: *arXiv preprint arXiv:2201.13360* (2022).

[47] V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert. “Incremental Distributed Inference from Arbitrary Poses and Unknown Data Association: Using Collaborating Robots to Establish a Common Reference”. In: *IEEE Control Systems Magazine (CSM), Special Issue on Distributed Control and Estimation for Robotic Vehicle Networks* 36.2 (2016), pp. 41–74.

[48] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. “Caffe: Convolutional Architecture for Fast Feature Embedding”. In: *arXiv preprint arXiv:1408.5093* (2014).

[49] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree”. In: *Intl. J. of Robotics Research* 31.2 (2 Feb. 2012), pp. 217–236.

[50] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. “iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Shanghai, China, May 2011.

[51] M. Kaess, A. Ranganathan, and F. Dellaert. “iSAM: Incremental Smoothing and Mapping”. In: *IEEE Trans. Robotics* 24.6 (Dec. 2008), pp. 1365–1378.

[52] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. “Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding”. In: *arXiv preprint arXiv:1511.02680* (2015).

[53] Alex Kendall and Yarin Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in Neural Information Processing Systems (NIPS)*. 2017, pp. 5580–5590.

[54] Alex Kendall, Matthew Grimes, and Roberto Cipolla. “Posenet: Convolutional networks for real-time 6-dof camera relocalization”. In: *Intl. Conf. on Computer Vision (ICCV)*. 2015.

[55] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. “Contrastive representation learning: A framework and review”. In: *IEEE Access* 8 (2020), pp. 193907–193934.

[56] Diederik P Kingma and Max Welling. “Auto-encoding Variational Bayes”. In: *International Conference on Learning Representations (ICLR)*. 2014.

[57] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. “Optimization by simulated annealing”. In: *science* 220.4598 (1983), pp. 671–680.

[58] G. Klein and D. Murray. “Parallel Tracking and Mapping for Small AR Workspaces”. In: *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*. Nara, Japan, Nov. 2007, pp. 225–234.

[59] D. Kopitkov and V. Indelman. “Robot Localization through Information Recovered From CNN Classificators”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. Madrid, Spain, Oct. 2018.

[60] D. Kortenkamp. “Cognitive maps for mobile robots: A representation for mapping and navigation”. PhD thesis. University of Michigan, 1993.

[61] Eitan Kosman and Dotan Di Castro. “GraphVid: It Only Takes a Few Nodes to Understand a Video”. In: *European Conf. on Computer Vision (ECCV)*. 2022.

[62] Ioannis Kostavelis and Antonios Gasteratos. “Semantic mapping for mobile robotics tasks: A survey”. In: *Robotics and Autonomous Systems* (2014).

[63] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[64] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. “Factor Graphs and the Sum-Product Algorithm”. In: *IEEE Trans. Inform. Theory* 47.2 (Feb. 2001), pp. 498–519.

[65] B.J. Kuipers. “The Spatial Semantic Hierarchy”. In: *Artificial Intelligence* 119 (2000), pp. 191–233.

[66] B.J. Kuipers and Y.-T. Byun. “A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations”. In: *Robotics and Autonomous Systems* 8 (1991), pp. 47–63.

[67] Pierre-Yves Lajoie, Siyi Hu, Giovanni Beltrame, and Luca Carlone. “Modeling Perceptual Aliasing in SLAM via Discrete-Continuous Graphical Models”. In: *IEEE Robotics and Automation Letters (RA-L)* (2019).

[68] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2017, pp. 6402–6413.

[69] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324.

[70] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[71] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. “SSD: Single Shot Multibox Detector”. In: *European Conf. on Computer Vision (ECCV)*. Springer. 2016, pp. 21–37.

[72] F. Lu and E. Milios. “Globally consistent range scan alignment for environment mapping”. In: *Autonomous Robots* (Apr. 1997), pp. 333–349.

[73] Björn Lütjens, Michael Everett, and Jonathan P How. “Safe Reinforcement Learning with Model Uncertainty Estimates”. In: *arXiv preprint arXiv:1810.08700* (2018).

[74] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *J. of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.

[75] Andrey Malinin and Mark Gales. “Predictive uncertainty estimation via prior networks”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2018, pp. 7047–7058.

[76] Andrey Malinin, Anton Ragni, Kate Knill, and Mark Gales. “Incorporating uncertainty into deep learning for spoken language assessment”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2017, pp. 45–50.

[77] Joshua G Mangelson, Derrick Dominic, Ryan M Eustice, and Ram Vasudevan. “Pairwise consistent measurement set maximization for robust multi-robot map merging”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 2916–2923.

[78] Hidenobu Matsuki, Raluca Scona, Jan Czarnowski, and Andrew J Davison. “Codemapping: Real-time dense mapping for sparse slam using compact scene representations”. In: *IEEE Robotics and Automation Letters (RA-L)* 6.4 (2021), pp. 7105–7112.

[79] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Vivian Weller. “Concrete problems for autonomous vehicle safety: advantages of Bayesian deep learning”. In: *Intl. Joint Conf. on AI (IJCAI)*. 2017.

[80] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. “Fusion++: Volumetric Object-Level SLAM”. In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 32–41.

[81] 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)*. IEEE. 2017, pp. 4628–4635.

[82] A. Milan, S.H. Rezatofighi, A.R. Dick, I.D. Reid, and K. Schindler. “Online Multi-Target Tracking Using Recurrent Neural Networks.” In: *Nat. Conf. on Artificial Intelligence (AAAI)*. 2017, pp. 4225–4232.

[83] B. Mildenhall, P.P Srinivasan, M. Tancik, J.T Barron, R. Ramamoorthi, and R. Ng. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *European Conf. on Computer Vision (ECCV)*. Springer. 2020, pp. 405–421.

[84] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. “Evaluating Merging Strategies for Sampling-based Uncertainty Techniques in Object Detection”. In: *arXiv preprint arXiv:1809.06006* (2018).

[85] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. “Dropout sampling for robust object detection in open-set conditions”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–7.

[86] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[87] Beipeng Mu, Shih-Yuan Liu, Liam Paull, John Leonard, and Jonathan How. “SLAM with Objects using a Nonparametric Pose Graph”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2016.

[88] Pavel Myshkov and Simon Julier. “Posterior distribution analysis for bayesian inference in neural networks”. In: *Workshop on Bayesian Deep Learning, NIPS*. 2016.

[89] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. “DTAM: Dense Tracking and Mapping in Real-Time”. In: *Intl. Conf. on Computer Vision (ICCV)*. Barcelona, Spain, Nov. 2011, pp. 2320–2327.

[90] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. “QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented slam”. In: *IEEE Robotics and Automation Letters (RA-L)* 4.1 (2018), pp. 1–8.

[91] E. Olson and P. Agarwal. “Inference on networks of mixtures for robust robot mapping”. In: *Intl. J. of Robotics Research* 32.7 (2013), pp. 826–840.

[92] Shayegan Omidshafiei, Brett T Lopez, Jonathan P How, and John Vian. “Hierarchical Bayesian Noise Inference for Robust Real-time Probabilistic Object Classification”. In: *arXiv preprint arXiv:1605.01042* (2016).

[93] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. “Deep exploration via bootstrapped DQN”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2016, pp. 4026–4034.

[94] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. “Automatic differentiation in pytorch”. In: *Advances in Neural Information Processing Systems (NIPS)* (2017).

[95] S. Pathak, A. Thomas, and V. Indelman. “A Unified Framework for Data Association Aware Robust Belief Space Planning and Perception”. In: *Intl. J. of Robotics Research* 32.2-3 (2018), pp. 287–315.

[96] T. Patten, M. Zillich, R. Fitch, M. Vincze, and S. Sukkarieh. “Viewpoint Evaluation for Online 3-D Active Object Classification”. In: *IEEE Robotics and Automation Letters (RA-L)* 1.1 (Jan. 2016), pp. 73–81.

[97] Timothy Patten, Wolfram Martens, and Robert Fitch. “Monte Carlo planning for active object classification”. In: *Autonomous Robots* 42.2 (2018), pp. 391–421.

[98] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. “Scikit-learn: Machine learning in Python”. In: *J. of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[99] Sudeep Pillai and John Leonard. “Monocular slam supported object recognition”. In: *Robotics: Science and Systems (RSS)*. 2015.

[100] Sören Pirk, Mohi Khansari, Yunfei Bai, Corey Lynch, and Pierre Sermanet. “Online Object Representations with Contrastive Learning”. In: *arXiv preprint arXiv:1906.04312* (2019).

[101] L. Polok, V. Ila, M. Solony, P. Smrz, and P. Zemcik. “Incremental Block Cholesky Factorization for Nonlinear Least Squares in Robotics”. In: *Robotics: Science and Systems (RSS)*. 2013.

[102] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. “UnrealCV: Virtual Worlds for Computer Vision”. In: *Proceedings of the 2017 ACM on Multimedia Conference*. ACM. 2017, pp. 1221–1224.

[103] A. Ranganathan and F. Dellaert. “Semantic Modeling of Places using Objects”. In: *Robotics: Science and Systems (RSS)*. Atlanta; USA, 2007.

[104] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT press, Cambridge, MA, 2006.

[105] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You only look once: Unified, real-time object detection”. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788.

[106] 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)*. IEEE. 2020, pp. 1689–1696.

[107] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. “Kimera: From SLAM to spatial perception with 3D dynamic scene graphs”. In: *The International Journal of Robotics Research* 40.12-14 (2021), pp. 1510–1546.

[108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "Imagenet large scale visual recognition challenge". In: *Intl. J. of Computer Vision* 115.3 (2015), pp. 211–252.

[109] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. Kelly, and A. J. Davison. "Slam++: Simultaneous localisation and mapping at the level of objects". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1352–1359.

[110] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. "Slam++: Simultaneous localisation and mapping at the level of objects". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1352–1359.

[111] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. "BigBIRD: A large-scale 3d database of object instances". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 509–516.

[112] Paloma Sodhi, Eric Dexheimer, Mustafa Mukadam, Stuart Anderson, and Michael Kaess. "LEO: Learning energy-based models in graph optimization". In: *Conference on Robot Learning*. 2021.

[113] Paloma Sodhi, Michael Kaess, Mustafa Mukadam, and Stuart Anderson. "Learning tactile models for factor graph-based estimation". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2021, pp. 13686–13692.

[114] Kihyuk Sohn. "Improved deep metric learning with multi-class n-pair loss objective". In: *Advances in Neural Information Processing Systems (NIPS)*. 2016, pp. 1857–1865.

[115] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models". In: *Advances in neural information processing systems*. 2015, pp. 3483–3491.

[116] Hauke Strasdat, José MM Montiel, and Andrew J Davison. "Visual SLAM: why filter?" In: *Image and Vision Computing* 30.2 (2012), pp. 65–77.

[117] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. "iMAP: Implicit mapping and positioning in real-time". In: *Intl. Conf. on Computer Vision (ICCV)*. 2021, pp. 6229–6238.

[118] Edgar Sucar, Kentaro Wada, and Andrew Davison. "NodeSLAM: Neural object descriptors for multi-view shape reconstruction". In: *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 949–958.

[119] N. Sünderhauf and P. Protzel. “Switchable Constraints for Robust Pose Graph SLAM”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2012.

[120] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. “The limits and potentials of deep learning for robotics”. In: *Intl. J. of Robotics Research* 37.4-5 (2018), pp. 405–420.

[121] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Markus Eich, Ben Upcroft, and Michael Milford. “SLAM-Quo Vadis? In support of object oriented and semantic SLAM”. In: *Proceedings of the RSS 2015 Workshop-Problem of mobile sensors: Setting future goals and indicators of progress for SLAM*. Australian Centre for Robotic Vision. 2015, pp. 1–7.

[122] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. “Meaningful Maps with Object-Oriented Semantic Mapping”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 5079–5085.

[123] A. Tapus, N. Tomatis, and R. Siegwart. “Topological Global Localization and Mapping with Fingerprint and Uncertainty”. In: *Proceedings of the International Symposium on Experimental Robotics*. 2004.

[124] V. Tchuiev, Y. Feldman, and V. Indelman. “Data Association Aware Semantic Mapping and Localization via a Viewpoint-Dependent Classifier Model”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019.

[125] V. Tchuiev and V. Indelman. “Inference over Distribution of Posterior Class Probabilities for Reliable Bayesian Classification and Object-Level Perception”. In: *IEEE Robotics and Automation Letters (RA-L)* 3.4 (2018), pp. 4329–4336.

[126] WT Teacy, Simon J Julier, Renzo De Nardi, Alex Rogers, and Nicholas R Jennings. “Observation Modelling for Vision-Based Target Search by Unmanned Aerial Vehicles”. In: *Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*. 2015, pp. 1607–1614.

[127] E. C. Tolman. “Cognitive Maps in Rats and Man”. In: *Behavior and Psychological Man*. University of California Press, 1951.

[128] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart. “Cognitive Maps for Mobile Robots: An Object based Approach”. In: *Proceedings of the IROS Workshop From Sensors to Human Spatial Concepts (FS2HSC 2006)*. 2006.

[129] Javier Velez, Garrett Hemann, Albert S Huang, Ingmar Posner, and Nicholas Roy. “Modelling observation correlations for active exploration and robust object detection”. In: *J. of Artificial Intelligence Research* (2012).

[130] Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, and Andrew Davison. "ElasticFusion: Dense SLAM without a pose graph". In: Robotics: Science and Systems. 2015.

[131] L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. "Data association for semantic world modeling from partial views". In: *International Symposium for Robotics Research*. Intl. Foundation of Robotics Research. 2013.

[132] Shichao Yang and Sebastian Scherer. "CubeSLAM: Monocular 3-D object SLAM". In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 925–938.

[133] HW Yu and Beom Hee Lee. "A variational feature encoding method of 3D object for probabilistic semantic SLAM". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3605–3612.

[134] HW Yu, JY Moon, and BH Lee. "A variational observation model of 3d object for probabilistic semantic slam". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5866–5872.

[135] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. "Coca: Contrastive captioners are image-text foundation models". In: *arXiv preprint arXiv:2205.01917* (2022).

[136] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. "Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11776–11785.

מיקום ולמעשה מסלול) של הרובוט ומיפוי העצמים בסביבתו – תוך התחשבות באי-הודאות בשיוך המדידות. בחלק שלישי של המחקר אנו מציעים גישה חדשה לבעיית המיפוי הסמנטי מבוסס העצמים שבה המידע הסמנטי של כל עצם מיוצג על ידי וקטור רציף (בניגוד לאוסף תכונות בדידות) במרחב לטנטי שהוא מרחב קלט של מודל תלוי נקודת מבט. אנו מציגים גישות ללמידת מודל כזה, וכן תוצאות למידה והיסק תוך שימוש במודל כזה.

ועוד). בעשור האחרון פותחו שיטות מבוססות למידה עמוקה שמאפשרות לחלץ מידע סמנטי מתוך מדידות חיישנים (לדוגמא, תמונות) בצורה מהימנה יחסית, וסללו את הדרך לחישה סמנטית רובוטית – נגישות המידע הסמנטי הובילה למחקר בשיטות לשילובו בחישה רובוטית, וזהו נושא המחקר הנוכחי.

ניתן להתייחס לפלט של שיטות (מודלים) מבוססות למידה עמוקה (לדוגמא, רשת סיווג – שבהינתן תמונה של אובייקט פולטת התפלגות הסיווג של העצם – וקטור שמכיל הסתברויות השתייכות לכל אחת ממספר מחלקות מועמדות, הנסכמות ל-1) כמדידות סמנטיות מ"חיישן וירטואלי" שהוא המודל הנלמד. דהיינו מודל למידה עמוקה פועל על מדידה גולמית (לדוגמא, תמונה) והפלט שלו הוא המדידה הסמנטית. מדידות סמנטיות מתנהגות באופן שונה ממדידות גיאומטריות, ושילובן בחישה רובוטית דורשת התייחסות לשוני. ראשית, הן תלויות בסט האימון של המודל שממנו התקבלו – פלט מודל נלמד מכיל אי ודאות שנגרמת מהשוני של הקלט ביחס לסט האימון. אם הקלט שונה מהותית מסט האימון המדידה עלולה להיות חסרת משמעות. ואולם, מצופה מרובוט אוטונומי שיהיה מסוגל לפעול בסביבות מגוונות, שחלקן אולי לא מיוצגות בסט האימון – ולכן יש הכרח להתמודד עם הבעיה.

מאפיין נוסף של מידע סמנטי הוא תלות בנקודת מבט ובין נקודות מבט. ככלל ההנחה לגבי מדידות גיאומטריות היא שהן בלתי-תלויות סטטיסטית, דהיינו – שכל מדידה חדשה מוסיפה מידע ביחס למדידות הקודמות. עבור מדידות סמנטיות הנחה זו נשברת מיד – ברור כי שתי תמונות מאותה נקודת מבט או נקודות מבט קרובות ככלל נושאות את אותו המידע ולכן יובילו לפלט מסווג דומה, כלומר הנחת אי-תלות סטטיסטית בין נקודות מבט היא הנחה שגויה עבור מדידות סמטיות. ערך המדידה הסמנטית גם הוא עלול להיות תלוי נק' מבט – לדוגמא פלט של מסווג ככלל לא נשאר קבוע כתלות ברקע וזווית הסתכלות על עצם ולעיתים קרובות עלול להיות שגוי לזוויות מסוימות.

אתגר נוסף בחישה סמנטית הוא שהאופי הבדיד של מידע סמנטי מוביל לסיבוכיות קומבינטורית של ייצוג המצב. בני אדם מגדירים מרכיבים בסביבה באמצעות סט סופי של מילים שמתארות תכונות, לדוגמא צבע, או סוג עצם. תכונה בדידה מתרגמת בייצוג המצב לסט של היפותזות שגודלו המרבי כמספר הערכים האפשריים לתכונה. יותר מתכונה בודדת או יותר מרכיב אחד בסביבה שמאופיין בתכונה בדידה מובילים לגידול קומבינטורי במספר ההיפותזות האפשריות – וזוהי למעשה בעיית ייצוג חמורה.

לבסוף אתגר נוסף המשותף לשיטות עם מרכיב מבוסס למידה הוא הקושי בייצור סט אימון – שהוא חמור אף יותר בשימושי רובוטיקה. ככלל שיטות למידה עמוקה דורשות שסט האימון יכיל דוגמאות קלט ופלט תואמות ומדויקות. אולם עבור רובוט אוטונומי ראשית לעיתים קרובות קיים קושי בייצור מידע אודות המצב המדויק של הרובוט – כיוון שהמצב ידוע תמיד עד כדי אי-ודאות. שנית לעיתים נדרש לרובוט לחלץ מידע עשיר מתוך המדידה הגולמית – וזהו קושי נוסף ליצירת סט האימון.

המחקר הנוכחי מפתח שיטות לחישה סמנטית תחת אי-ודאות שמתמודדות עם האתגרים שתוארו מעלה. אבחנה מרכזית היא שבעוד שפלט של מודל למידה עמוקה משתנה בין נקודות מבט, הוא עושה זאת באופן הניתן לחיזוי ולבניית מודל סטטיסטי, תלוי נקודת-מבט, שמאפשר לנצל את השינויים תלויי נקודת-המבט להיסק בו זמני סמנטי וגיאומטרי, כאשר דיזמביגואציה סמנטית מובילה לדיזמביגואציה גיאומטרית ולהיפך. בחלק ראשון של המחקר, אנו מראים שימוש למודלים תלויי נקודת מבט לסיווג של עצם באופן שמתחשב גם בקורלציה בין נקודות מבט ובאי-הודאות במדידות הסמנטיות הנובעת מהשוני ביחס לסט האימון, בנוסף לאי-הודאות במיקום הרובוט (ולכן בנקודת המבט לעבר העצם). בחלק שני של המחקר אנו מפתחים שיטה מבוססת מודל מדידה תלוי נקודת מבט ללוקליזציה (שיערוך

תקציר

רובוט אוטונומי הוא רובוט הממלא משימות באופן עצמוני, ללא שליטת מפעיל. רובוט כזה חש את סביבתו באמצעות מדידות חיישניו (לדוגמא מצלמות, מד תאוצה, גירוסקופ ומגנומטר, מדי כיוון ומרחק) ומרכיב מתוך מדידות שאסף לאורך זמן מכלל החיישנים תמונת מצב אחודה הנדרשת לשם פעולתו – תהליך שנקרא חישה רובוטית (ובקונטקסט כללי יותר לעיתים: היתוך מידע). רובוט אוטונומי משתמש בתמונת המצב שהוא בונה ומתחזק במהלך פעולתו על מנת לתכנן פעולות עתידיות הנדרשות לביצוע המשימה שלו. תמונת המצב של רובוט אוטונומי מתאפיינת באי-ודאות שנובעת ממספר מקורות - המדידות שמתקבלות הן לרוב רועשות וחלקיות (דהיינו משקפות רק חלק נראה מהסביבה), הפעולות המתוכננות מתבצעות לרוב עם שגיאה מסוימת, וכן עלול להיות קושי לשייך בין מדידות שהתקבלו למרכיבי הסביבה שמופו (שיוך מידע). מצב מאתגר במיוחד הוא כאשר איזורים שונים בסביבה מובילים למדידות זהות אצל הרובוט (לדוגמה כאשר הסביבה מכילה רכיבים חזרתיים כפי שקורה תכופות בסביבות מעשי ידי אדם) – במצב זה, שנקרא "התחזות" aliasing, יש צורך לאסוף מדידות נוספות כדי לפתור את בעיית שיוך המידע שנוצרת ולשערך את מצב הרובוט והסביבה נכונה. על מנת לפעול בצורה מהימנה, נדרש רובוט אוטונומי להתמודד עם כלל אי-הודאות בכל מהלך פעולתו ולהתחשב בה בתהליך החישה שלו.

בעבר רובוטים שפעלו באופן עצמוני מילאו משימות חזרתיות ומובנות מאד שמצריכות תמונת מצב יחסית מאד לא מורכבת – והיו מצויים בעיקר לדוגמא בבתי חרושת. לאחר התקדמות בחומרה ובשיטות אלגוריתמיות, נכנסים רובוטים אוטונומיים לשימוש במגוון הולך וגדל של תחומים, החל מרובוטים לניקוי רצפות בתים שהפכו לנפוצים היום, המשך במכסחות דשא ומנקי בריכות אוטונומיים, כלים תת מימיים ואווירים לגילוי פגמים בספינות ובמבנים, המשך במערכות לאיתור נעדרים באיזורים מרוחקים, בתת קרקע ומתחת למים, המשך במשימות בחלל. ברוב השימושים בהם נפוצים רובוטים אוטונומיים כיום די לרובוט בעיקר בתמונת מצב גיאומטרית למילוי משימתו, דהיינו על הרובוט למפות את סביבתו מבחינה גיאומטרית – לאיזורים עבירים ובלתי-עבירים ולעקוב אחר מיקומו בתוך המפה שיצר – תהליך שידוע בשם SLAM – Simultaneous Localization and Mapping – מיפוי והסקת מיקום בו-זמני. ואולם, תחומים חדשים שבהם יש צורך ברובוטים אוטונומיים ומתאפיינים בסביבות ומשימות פחות מובנות, ולא רק תלויות גיאומטריה, פעמים רבות בפעולה בסביבת בני אדם – דורשים מהרובוט ניתוח והבנה יותר עדינים של סביבתו, וחילוץ מידע מעבר לגיאומטרי, לדוגמא לגבי סוג העצמים המקיפים אותו ותפקידם – וזהו מידע סמנטי, שדורש חישה מסוג חדש - חישה סמנטית.

מידע סמנטי מספק שפה משותפת לרובוט ולמשתמש האנושי, ומאפשר להגדיר מיקומים ומשימות. הוא גם מאפשר ייצוג יעיל וברור למשתמש של סביבת הרובוט. בנוסף, מידע סמנטי מסייע בפתרון אי-ודאות בשיוך מידע (כאשר שיוך יכול להתבצע על סמך מאפיינים סמנטיים, כגון צבע, סוג אובייקט

המחקר בוצע בהנחייתו של פרופסור משנה ואדים אינדלמן, בפקולטה למדעי המחשב

חלק מן התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי-עת במהלך תקופת מחקר הדוקטורט של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

בכתבי עת

[1] Y. Feldman and V. Indelman. "Spatially-Dependent Bayesian Semantic Perception under Model and Localization Uncertainty". In: *Autonomous Robots* (2020)

בכנסים

[1] Y. Feldman and V. Indelman. "Bayesian Viewpoint-Dependent Robust Classification under Model and Localization Uncertainty". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018

[2] V. Tchuiev, Y. Feldman, and V. Indelman. "Data Association Aware Semantic Mapping and Localization via a Viewpoint-Dependent Classifier Model". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019

חלק המחבר בפרסום Tchuiev, Feldman, and Indelman [124] היה פיתוח הגישה התיאורטית (במשותף עם המחברים האחרים) וחלק בכתיבת המאמר. מימוש השיטה והניסויים נעשו כולם על ידי ולדימיר צ'ויב השותף למאמר.

בסדנאות מקצועיות

[1] Y. Feldman and V. Indelman. "Towards Self-Supervised Semantic Representation with a Viewpoint-Dependent Observation Model". In: *Workshop on Self-Supervised Robot Learning, in conjunction with Robotics: Science and Systems (RSS)*. July 2020

[2] Y. Feldman and V. Indelman. "Towards Robust Autonomous Semantic Perception". In: *Workshop on Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding, in conjunction with IEEE International Conference on Robotics and Automation (ICRA)*. May 2018

תודות

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי, וכן על תמיכה חלקית נוספת שהוענקה ע"י המרכז הישראלי למחקר בתחבורה חכמה.

חישה סמנטית תחת אי-ודאות עם מודלים מרחביים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

יורי פלדמן

הוגש לסנט הטכניון — מכון טכנולוגי לישראל
תמוז התשפ"ב חיפה יולי 2022

חישה סמנטית תחת אי-ודאות עם מודלים מרחביים

יורי פלדמן