

Experience-Based Prediction of Unknown Environments for Enhanced Belief Space Planning

Omri Asraf

Experience-Based Prediction of Unknown Environments for Enhanced Belief Space Planning

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

Omri Asraf

Submitted to the Senate
of the Technion — Israel Institute of Technology
Av 5780 Haifa August 2020

This research was carried out under the supervision of Associate Prof. Vadim Indelman, in the Faculty of Aerospace Engineering.

Publications:

O. Asraf and V. Indelman. Experience-based prediction of unknown environments for enhanced belief space planning. In IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2020.

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Vadim Indelman for hours of brain storming, the invested mentoring and collaborative work from which I learned a lot. Your guidance helped me gain knowledge and research skills that will remain with me for years to come. I would also like to thank my parents for the care and support all over the years. Finally, I thank to my wife Noga Shaked for always believing in me, helping with language editing and the moral support.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Tables

Abstract	1
Abbreviations and Notations	3
1 Introduction	5
1.1 Related Work	6
1.1.1 Simultaneous Localization And Mapping	6
1.1.2 BSP in unknown environments	7
1.1.3 Experience-based planning	7
1.2 Contributions	8
1.3 Organization	8
2 Problem Formulation	9
2.1 SLAM	9
2.2 Belief Space Planning (BSP)	10
2.3 Problem Statement	11
3 Approach	13
3.1 Incorporating Experience within BSP	13
3.2 Experience-Based Prediction of Future Observations	14
3.3 Offline Training Phase	15
3.4 Online Deployment	16
3.5 Novelty Detection	18
4 Results	21
4.1 Map Distribution Prediction With CVAE	21
4.2 BSP in Unknown Environments Simulation Results	23
5 Conclusions and Future Work	35
5.1 Future Work	35

List of Figures

3.1	Training and Deployment.	17
4.1	Process of organization the dataset.	21
4.2	Examples of the prediction algorithm. (a) Inputs - sub-map and action; (b) Outputs - two samples of prediction with reconstruction error (RE) and prediction error (PE). On the left worst prediction result and on the right best prediction result; (c) Ground truth.	22
4.3	Reconstruction error histogram: (a) trainset; (b) testset. Prediction error histogram: (c) trainset; (d) testset.	24
4.4	(a) 3D Gazebo simulation world; (b) Planning session with three candidate actions; (c) Occupancy grid of the partial map that was used for prediction. . . .	24
4.5	Map prediction results for three actions from Figure 4.4b. Ground truth maps in relevant regions are shown in (a) for action 1, in (b) for action 2, and in (c) for action 3. Figures (d), (e) and (f) show five samples of map predictions for each of the actions. Conditional map in light green, predicted map in dark green. . . .	25
4.6	Comparison between three methods of uncertainty evolution on three different actions from Figure 4.4b: Action 1 - top row; action 2 - middle row; action 3 - bottom row. Uncertainty evolution using (a) ground truth map (GT-map), (b) using one sample of prediction map (our approach), and (c) using motion model only (baseline). Column (d) presents a comparison between all three methods. In column (b), results are shown for one sample from Figure 4.5 for each action: action 1 - first row in 4.5d, action 2 - fifth row in 4.5e, action 3 - forth row in 4.5f. Results for all samples are summarized in Figure 4.7a. For convenient visualization covariance resolution was multiplied by 100.	27
4.7	(a) Cost function results of all the samples from Fig. 4.5 compared to cost function calculated with the GT-map method. (b) Objective function (4.1) values for the three methods. In contrast to baseline BSP, our approach preserves action-ordering with respect to GT-map.	27
4.8	Light green - conditional map, yellow - three candidate actions. Figures (a)-(o) according to planning scenarios 1-15 from Table 4.1.	29

4.9	Objective function values for the three BSP methods. baseline in black, our approach in red, GT-map in blue. Figures (a)-(o) according to planning scenarios 1-15 from Table 4.1.	30
4.10	(a) Planning session of scenario 2 with three candidate actions; (b) Occupancy grid of the partial map that was used for prediction.	31
4.11	Comparison between three methods of uncertainty evolution on three different actions from scenario 2: (a) Action 1; (b) action 2; (c) action 3. In blue - uncertainty evolution using ground truth map (GT-map); In black - using motion model only (baseline); and in red - using samples of prediction map (our approach) . For convenient visualization covariance resolution was multiplied by 100.	32
4.12	(a) Cost function results of all the samples from scenario 2 compared to cost function calculated with the GT-map method. (b) Objective function values for the three methods in scenario 2.	33
4.13	(a) Cost function results of all the samples from scenario 6 compared to cost function calculated with the GT-map method. (b) Objective function values for the three methods in scenario 6.	33

List of Tables

Abstract

Autonomous navigation missions require online decision making abilities, in order to choose from a given set of candidate actions an action that will lead to the best outcome. In a partially observable setting, decision making under uncertainty, also known as belief space planning (BSP), involves reasoning about belief evolution considering realizations of future observations. Yet, when candidate actions lead the robot to an unknown environment the decision making mission becomes a very challenging problem since without a map it is hard to foresee future observations. The main question investigated in this thesis is how to utilize information collected in the current navigation missions by SLAM and the experience from other environments for planning tasks in unknown environment. We develop a data-driven approach for predicting a distribution over an unexplored map, generating future observations, and combining these observations within BSP. However, previous experience based on a given dataset may not assist in some cases, so we suggest a novelty detection method. Using our novelty detection method, we identify online cases where the given map constructed up to current time is very different from the train-set maps. In these cases, standard BSP is preferred and used instead. In the results, we showed the performance of the map prediction algorithm on test-set and discussed success and failure cases. Additionally, We examine our approach and compared it to existing BSP methods in a Gazebo simulation. The results show that our approach often yields improved performance over standard BSP, leading to fewer mistakes in decision making and preferable in the uncertainty error for most scenarios. We also present a case where the navigation takes place in an unfamiliar environment, that is very different from the training-set, and show how our proposed novelty detection method is capable of successfully identifying this case.

Abbreviations and Notations

BSP	:	Belief Space Planning
SLAM	:	Simultaneous Localization and Mapping
DL	:	Deep Learning
NN	:	Neural Network
GAN	:	Generative Adversarial Network
VAE	:	Variational Autoencoders
MAP	:	Maximum A Posteriori (estimation)
POMDP	:	Partially Observable Markov Decision Process
RL	:	Reinforcement Learning
ICP	:	Iterative Closest Point
VO	:	Visual Odometry
iSAM	:	Incremental Smoothing and Mapping
KL	:	Kullback-Leiber divergence
ELBO	:	Evidence Lower Bound
RE	:	Reconstruction Error
PE	:	Prediction Error
GTSAM	:	Georgia Tech-Smoothing and Mapping
GT-map	:	BSP with access to Ground Truth map
Baseline	:	standard BSP method (into consideration only the motion model)
$x_{1:k}$:	robot's states until current time
M_k	:	map observed up time k
m_i	:	sub-map that is within the field of view of the sensor located at state (pose) x_i
$y_{1:k}$:	measurement up time k
$a_{1:k}$:	actions up time k
y_{ij}^{rel}	:	relative pose measurement between two robot poses x_i and x_j
b_k	:	belief of state vector $x_{1:k}$, its probability density function

Chapter 1

Introduction

Autonomous navigation in an unknown environment is a challenging problem in robotics. In this situation the agent starts from a point where it has no information about the environment, and its mission is to reach a given goal. One of the main approaches to address this challenge is simultaneous localization and mapping (SLAM). Using SLAM, an agent deals with two missions at the same time, first it perceives the surrounding environment using its onboard sensors (e.g. cameras and laser scans) and creates a representation of the map. Second, the agent estimates its pose relative to this map [2].

Another task in autonomous navigation is decision making. The agent generates candidate actions (e.g. by PRM, RRT [22], [27]) and it needs to choose which action will lead to the best outcome. One method to address this problem is belief space planning (BSP) [16, 39]. Using this method, the agent performs belief propagation and evaluates the objective function for each candidate action given a history of measurements and actions that the agent has performed up to current time, and determines the best action as the one that leads to the highest value of the objective function.

However, despite the recent progress, state of the art BSP approaches that address autonomous operation in unknown environments have some limitations. First, these approaches consider areas not yet mapped to be obstacle-free, causing some of the generated candidate actions to be infeasible in practice. Second, existing approaches perform belief propagation within unexplored areas by only considering uncertainty due to motion model and without explicitly modeling the expected sensor observations in those areas.

In contrast, when thinking about a human navigating in an unknown environment, he/she most likely does not rely solely on sensory inputs, but also on past knowledge and experience. In particular, using past experience in similar areas, and based on only partial sensory information obtained thus far, one is able to envision the expected map in unexplored nearby areas. For example, when seeing 3 walls connected to each other, as in a room entry, we can envision and complete the shape of the room based our knowledge that rooms are usually rectangular. Similarly, we are able to leverage experience to predict high-level semantics (e.g. doors, elevator) in unexplored nearby environments.

One method by which a robot can learn from experience is deep learning (DL). Specifically

relevant to this map prediction task is the inpainting problem - the task of completing partial images [31]. In this method, a neural network (NN) is trained with a dataset of partial images. In the training part the NN has access to the ground truth, and so it can improve itself by comparing the prediction's result to the given data. Afterwards there is a deployment stage, where the NN does not have access to the original pictures, and it completes an unfamiliar set of partial images based on the experience learned in the training stage. The DL architectures popular to solve the inpainting problem are generative adversarial network (GAN) [17] and variational autoencoders (VAE) [7].

In this work, we develop an approach to incorporate relevant prior experience to predict distribution over unexplored environments within belief space planning framework. We focus specifically on unexplored areas as belief propagation and reward calculation can be readily performed for candidate actions that pass through already mapped environments. In contrast, our data-driven approach approximately predicts a distribution over unexplored areas along each candidate action using a conditional generative model. This distribution enables to perform belief propagation while accounting for future sensor observations, and we show empirically our approach predicts posterior uncertainty over robot trajectory that is typically close to the actual uncertainty that will be obtained upon mapping the corresponding environment. This, in turn, enables to choose most informative actions by evaluating information-theoretic rewards.

1.1 Related Work

1.1.1 Simultaneous Localization And Mapping

In this work, we consider the inference stage as SLAM setting. SLAM research involves many different areas. At the lower level (front-end) SLAM intersects other research fields such as computer vision and signal processing; at the higher level (back-end), SLAM is an appealing mix of geometry, graph theory, optimization, and probabilistic estimation. SLAM has two main formulations for the estimation of robot and map states. The first is full-SLAM, wherein the robot and map states are estimated together, and the second is pose-SLAM, in which they are estimated separately.

Another division of SLAM is by two approaches of estimation: filtering-based and optimization-based. The most popular filtering-based approach is EKF-SLAM [13, 29] which works by two main steps. First, it predicts how the platform moved and then updates the current state with new measurements. Yet, in visual SLAM where the state dimension grows over time (e.g. map states represented by landmarks), EKF-SLAM approaches have high computational complexity. In contrast, optimization-based approaches like bundle adjustment estimate in each step the current and past states (in full-SLAM also the map states). These approaches perform calculation in the information space and can therefore exploit the inherent sparsity of the system. One of the successful approaches is Smoothing and Mapping (SAM) [5], which formulates the optimization problem by a factor graph and exploits sparsity in the (square root) information form, making calculations very efficient. In an advanced version, known as ISAM2 [19], there has been an

improvement to incremental computational scheme, with new reordering methods that were computationally preferred.

1.1.2 BSP in unknown environments

BSP is an approach developed following efforts of merging SLAM and path planning into an integrated framework. Prentice and Roy [33] suggested considering the state's uncertainty in the decision making processes in a known map setting. Decision making approaches in an unknown environment and Partially Observable Markov Decision Process (POMDP) setting can involve a variety of information-theoretic rewards and tasks. Stachniss et al. [37] develop an active SLAM approach that compares between two utilities associated with the action of exploration and revisiting. Kim et al. [23] integrated navigation algorithm that automatically balances between exploration and revisiting using a BSP reward framework that considers the visual perception measurement likelihood. Indelman et al. [16] developed general and continuous frameworks of the BSP approach for both families: robot and environment states. All of these three works suggested to consider the landmarks/map that was seen up to current time by predicting future loop closure, but also assume that the other area (unexplored area) is free. [15] proposed a cooperative multi-robot BSP in which future information sharing is considered in the planning phase, similar to loop closure but now with other agents. In practice, when the action leads to an unexplored area and the setting of the sensors is environment dependent (e.g. camera or laser scan), the future information-theoretic reward should be dependent on the unexplored area.

Another important aspect in BSP is computational complexity. Standard BSP methods usually perform posterior belief for each candidate action from scratch, which are very expensive calculations. Kopitkov et al. [24, 25] suggested to resort to matrix determinant lemma and calculation re-use techniques for information-theoretic decision making. Farhi et al. [10] introduced an approach for incremental expectation belief space planning that avoids the common assumption of maximum likelihood observations. Elimelech et al. [8] suggested to perform a precursory variable reordering procedure on the belief, in order to optimize the number of variables to update in planning.

1.1.3 Experience-based planning

Experience-based navigation or planning is usually linked to reinforcement learning (RL). RL and BSP both share the same goal of finding the optimal action, but there are differences in approaches. In RL the policy is mostly learned offline based on experience from similar missions; numerous model-free and model-based approaches have been developed in recent years due to rise of DL. However, the vast majority of these approaches consider a Markov decision process (MDP) problem, i.e. the state is fully observable. Contrarily, BSP is calculated online based on the history of information in the current mission and it is an instantiation of a partially observable MDP (POMDP) problem.

Recently several works addressed (deep) RL under POMDP setting, considering different levels of end-to-end planning under uncertainty in order to deal with the active localization

problem [4, 20, 26], assuming environment is known. More closely related to us are learning approaches that consider autonomous navigation in unknown environments [34, 38]. These approaches focused on the goal to find the shortest feasible action in unknown environments without considering the uncertainty that propagated due to future observations.

However, end-to-end approaches have limitations in terms of interpretability. Therefore, another approach is a hybrid between classic planning methods and experience based methods. In other words, most of the planning process is based on models and incorporates experience just for specific hard problems when there is no sufficient model. For example, Richter et al. [35] added a NN to visual navigation for predicting future collisions in an unknown environment, and Katyal et al. [21] used a NN for map prediction that assisted an efficient exploration of unknown environments. Our approach belongs to this category as well, as we utilize classic model based BSP while reasoning about previously mapped environments and incorporate experience-based map prediction only for the unexplored environments.

1.2 Contributions

We contribute an approach that appropriately incorporates experience within belief space planning, considering operation in unknown environments. In particular, we (i) develop an algorithm to calculate a predicted distribution over an unexplored area; (ii) we leverage this distribution to predict future observations and incorporate these within BSP; (iii) we suggest an online novelty detection method to avoid using irrelevant experience; (iv) we study our approach in a realistic simulation in Gazebo.

1.3 Organization

This thesis is organized as follows.

1. Chapter 2 introduces the concepts of pose-SLAM, BSP, and gives a formal statement of the problem.
2. Chapter 3 describes our approach divided to five steps.
3. Chapter 4 presents experimental results.
4. Conclusions and future work are described in Chapter 5.

Chapter 2

Problem Formulation

2.1 SLAM

The SLAM problem involves inferring states of the robot (e.g. pose and velocity) and of the map (e.g. occupancy grid, landmarks). Let $x_{1:k} \doteq \{x_i\}_{i=1}^k$ denote the robot's states until current time, which includes in this work the poses (position and orientation) of the robot. We denote by M_k the map observed by time k , and by $m_i \subseteq M_i$ the sub-map that is within the field of view of the sensor located at state (pose) x_i . Let $y_{1:k} \doteq \{y_1, \dots, y_k\}$ and $a_{0:k-1} \doteq \{a_0, \dots, a_{k-1}\}$ denote, respectively, the obtained measurements and the actions up to time k . In the current case we will be basing on motion and observation models with additive Gaussian noise. The motion model for a given action a_{i-1} and robot state x_{i-1} is

$$x_i = f(x_{i-1}, a_{i-1}) + w_i, \quad w_i \sim \mathcal{N}(0, \Sigma_w). \quad (2.1)$$

The formulation of an observation model is dependent on the kind of sensor and measurement algorithm we are using. Some sensors measure directly the robot state, e.g. GPS, while others like cameras or laser sensors measure the robot state in relation to the environment. For the latter case, the generative/measurement model for the raw measurements (e.g images, pointclouds) thus depends both on the robot state and the environment, i.e.

$$y_i = g(x_i, m_i) + u_i, \quad u_i \sim \mathcal{N}(0, \Sigma_u), \quad (2.2)$$

Where Σ_w and Σ_u are the process and measurement noise covariance matrices of the motion and observation models, which are assumed to be given.

Using two raw measurements, y_i and y_j , we can create a relative pose measurement y_{ij}^{rel} between two robot poses x_i and x_j . When the raw measurements are pointclouds, the algorithm could be e.g. ICP, and with raw images measurement it could be Visual Odometry (VO). In these algorithms, the relative-pose observation model is often modeled, for simplicity, by a Gaussian distribution, i.e. measurement noise is modeled as Gaussian (see e.g. [3, 28]),

$$y_{ij}^{rel}(y_i, y_j) = h(x_i, x_j) + v_{ij}, \quad v_{ij} \sim \mathcal{N}(0, \Sigma_v(y_i, y_j)). \quad (2.3)$$

Note that the relative-pose measurement y_{ij}^{rel} depends on the raw measurements y_i and y_j , which, according to the generative model (2.2) depends also on the environment map. Importantly, while the covariance Σ_v is assumed constant in most previous works, in practice, similarly to the measurements y_{ij}^{rel} , it is a function of the raw measurements and thus the map [3, 28].

SLAM has two main formulations. The first is full-SLAM where the robot and map states are estimated together. The second is pose-SLAM, in which the robot states are estimated separately and independently of the map estimation. In the current work we consider a pose-SLAM framework, hence the conditional probability density function (pdf) over the joint states, the *belief*, at time instant k is $b_k \doteq \mathbb{P}(x_{1:k}|H_k)$, where $H_k \doteq \{y_{1:k}, a_{0:k-1}\}$ denotes measurements and actions history. Using Bayes rule and standard assumptions it can be rewritten as

$$b_k = \eta \mathbb{P}(x_0) \prod_{i=1}^k [\mathbb{P}(x_i|x_{i-1}, a_{i-1}) \prod_j \mathbb{P}(y_{ij}^{rel}|x_i, x_j)], \quad (2.4)$$

where η includes all terms that do not involve the states, $\mathbb{P}(x_0)$ is the prior on x_0 , $\mathbb{P}(x_i|x_{i-1}, a_{i-1})$ and $\mathbb{P}(y_{ij}^{rel}|x_i, x_j)$ denote, respectively, the motion model (2.1) and measurement likelihood (2.3) terms. The *belief* is represented by a Gaussian distribution $b_k = \mathcal{N}(\hat{x}_{1:k}, \Sigma_k)$, parametrized by mean $\hat{x}_{1:k}$, and covariance Σ_k . Computationally efficient inference approaches, such as ISAM2 [19], can be used to calculate these parameters.

The posterior over the map M_k , an occupancy map in our case, can be calculated via marginalization over robot states, as in

$$\mathbb{P}(M_k|H_k) = \int \mathbb{P}(M_k|x_{1:k}, H_k) \mathbb{P}(x_{1:k}|H_k) dx_{1:k}. \quad (2.5)$$

However, in practice, these calculations are computationally expensive, and therefore, a common alternative in pose-SLAM is to approximate it using only the mean of the belief:

$$\mathbb{P}(M_k|H_k) \approx \mathbb{P}(M_k|\hat{x}_{1:k}, y_{1:k}). \quad (2.6)$$

2.2 Belief Space Planning (BSP)

In the planning stage the robot gets a set of (non-myopic) actions $A \doteq \{a_{k:k+L}^i\}_{i=1}^n$ and has to choose the best action according to an objective function. Here, we consider a discrete action space. Generally, each action could have a different planning horizon. The objective function is given by

$$J(b_k, a_{k:k+L-1}) = \sum_{l=1}^L \mathbb{E}_{y_{k+1:k+l}} \{c(b_{k+l}, a_{k+l-1})\}, \quad (2.7)$$

where the cost $c(\cdot)$ is a function of the posterior future belief

$$b_{k+l} \doteq \mathbb{P}(x_{1:k+l} | H_k, a_{k:k+l-1}, y_{k+1:k+l}), \quad (2.8)$$

which by itself depends on future observations. The expectation operator accounts for all possible realizations of these future observations. Generally, each action could have a different planning horizon.

The expectation can be explicitly written as

$$J(b_k, a_{k:k+L-1}) = \sum_{l=1}^L \int \mathbb{P}(y_{k+1:k+l} | H_k, a_{k:k+l-1}) c(b_{k+l}, a_{k+l-1}) dy_{k+1:k+l}. \quad (2.9)$$

The optimal (non-myopic) action is defined as

$$a_{k:k+L-1}^* = \arg \min_{a_{k:k+L-1}} J(b_k, a_{k:k+L-1}). \quad (2.10)$$

To calculate optimal action, one has to evaluate Eq. (2.7) for each candidate action. To do so, we should be able to reliably predict future observations, and appropriately approximate the expectation in (2.7), typically via sampling.

2.3 Problem Statement

In this work we will develop a method to predict the distribution of the future measurements. In order to predict this distribution, based on Eq. (2.2), the future map and robot states are required. If the action leads the robot to an environment that it mapped before (e.g. for myopic case, $m_{k+1} \subseteq M_{1:k}$) then we could generate future measurements based only on history. In our case, we focus on a situation where the goal is outside the map which the robot has seen and therefore there is no access to future measurements. Hence, given that action a_k yields x_{k+1} outside of the current map M_k , without any additional or prior knowledge, the distribution of m_{k+1} is uninformative, i.e. uniform.

Current BSP methods lack the information necessary to predict future measurements in unknown environments. To address this problem we suggest to incorporate experience within BSP, aiming to improve prediction of future observations as part of the expectation calculations in Eq. (2.7).

Chapter 3

Approach

3.1 Incorporating Experience within BSP

We propose to appropriately incorporate experience within the BSP framework, aiming to improve prediction of future observations as part of the expectation calculations in Eq. (2.7). In this section we define the available experience to include any data that could contribute to the prediction of future observations task. Further in section 3.3, we will define the experience more specifically as appropriate to the current work. Denoting the available experience by D , we re-write Eq. (2.9) as

$$J(b_k, a_{k:k+L-1}) = \sum_{l=1}^L \int \mathbb{P}(y_{k+1:k+l} | H_k, a_{k:k+l-1}, D) c(b_{k+l}, a_{k+l-1}) dy_{k+1:k+l} \quad (3.1)$$

For simplicity, in this section we present formulation for a myopic setting ($L = 1$), and extend later the discussion to the more general, non-myopic case ($L > 1$). Denoting $H_{k+1}^- \doteq \{H_k, a_k\}$, we re-write Eq. (2.7) as

$$J(b_k, a_k) = \int \mathbb{P}(y_{k+1} | H_{k+1}^-, D) c(b_{k+1}, a_k) dy_{k+1}. \quad (3.2)$$

Further, we empirically approximate the expectation via sampling. Considering N samples, we get

$$J(b_k, a_k) \approx \frac{1}{N} \sum_{y_{k+1} \sim \mathbb{P}(y_{k+1} | H_k, a_k, D)} c(b_{k+1}, a_k). \quad (3.3)$$

To sample future observations y_{k+1} , we first marginalize over the robot state x_{k+1} , and recalling the generative model (2.2), also over the corresponding sub-map/scene m_{k+1} .

$$\mathbb{P}(y_{k+1} | H_{k+1}^-, D) = \int_{x_{k+1}} \int_{m_{k+1}} \mathbb{P}(y_{k+1}, x_{k+1}, m_{k+1} | H_{k+1}^-, D) dm_{k+1} dx_{k+1}. \quad (3.4)$$

Note in cases the map m is a discrete variable (e.g occupancy grid), the integral over m_{k+1} should be replaced by a sum operator. In general, the map could be a continuous variable. For example, the map representation could include the color of the environment, on top of the geometry (see

e.g [9]). Applying the chain rule and recalling the Markov assumption yields

$$\begin{aligned} \mathbb{P}(y_{k+1}|H_{k+1}^-, D) &= \int_{x_{k+1}, m_{k+1}} \mathbb{P}(y_{k+1} | x_{k+1}, m_{k+1}) \cdot \\ &\quad \mathbb{P}(x_{k+1} | m_{k+1}, H_{k+1}^-, D) \mathbb{P}(m_{k+1} | H_{k+1}^-, D) dm_{k+1} dx_{k+1}. \end{aligned} \quad (3.5)$$

Recalling the pose-SLAM framework we, omit in the second term above the dependency of the belief on the robot state on the map, and furthermore, we omit the conditioning on the experience since the probabilistic models are considered given, i.e. $\mathbb{P}(x_{k+1} | m_{k+1}, H_{k+1}^-, D) = \mathbb{P}(x_{k+1} | H_{k+1}^-)$, which can be calculated by propagating the belief from the previous time instant and marginalization: $\mathbb{P}(x_{k+1} | H_k, a_k) = \int_{x_{1:k}} \mathbb{P}(x_{1:k} | H_k) \mathbb{P}(x_{k+1} | x_k, a_k) dx_{1:k}$. We note as the considered pdfs are Gaussian, this marginalization can be performed analytically and computationally efficiently, yielding $\mathbb{P}(x_{k+1} | H_k, a_k) = N(\hat{x}_{k+1}^-, \Sigma_{k+1}^-)$.

In practice, we approximate the expectation over x_{k+1} with a single sample from $\mathbb{P}(x_{k+1} | H_{k+1}^-)$, the maximum likelihood estimate \hat{x}_{k+1}^- (see line 15 in Algorithm 3.1). Thus, we re-write (3.5) as

$$\mathbb{P}(y_{k+1}|H_k, a_k, D) \approx \int_{m_{k+1}} \mathbb{P}(y_{k+1}|\hat{x}_{k+1}^-, m_{k+1}) \mathbb{P}(m_{k+1}|H_{k+1}^-, D) dm_{k+1}. \quad (3.6)$$

As seen, we get an intuitive result: to generate future observations we need a distribution over sub-maps/scenes. This is where we propose to leverage experience.

3.2 Experience-Based Prediction of Future Observations

In this section we consider the problem of approximately representing the distribution over future maps, i.e. $\mathbb{P}(m_{k+1}|H_{k+1}^-, D)$, utilizing available experience. While there are different approaches to address this problem, herein, we present our proposed method which uses generative models within a deep learning framework.

We start by marginalizing over the current map M_k ,

$$\begin{aligned} \mathbb{P}(m_{k+1}|H_{k+1}^-, D) &= \int_{M_k} \mathbb{P}(m_{k+1}, M_k|H_{k+1}^-, D) dM_k \\ &= \int_{M_k} \mathbb{P}(m_{k+1}|M_k, a_k, D) \mathbb{P}(M_k|H_k) dM_k, \end{aligned} \quad (3.7)$$

where the posterior over the map, $\mathbb{P}(M_k|H_k)$, is already available (see Eq. (2.6)).

Further, we approximate (3.7) with a single sample from $\mathbb{P}(M_k|H_k)$, the maximum likelihood estimate \hat{M}_k (see line 9 in Algorithm 3.1). With this approximation the future map distribution becomes

$$\mathbb{P}(m_{k+1}|H_{k+1}^-, D) \approx \mathbb{P}(m_{k+1}|\hat{M}_k, a_k, D). \quad (3.8)$$

As mentioned above, in SLAM setting, the constructed map M_k grows over time. Yet, memory-free DL architectures expect input from a pre-defined dimensionality. For this reason, in this

work we introduce another approximation, and use only the current sub-map $\hat{m}_k \subseteq \hat{M}_k$, i.e.

$$\mathbb{P}(m_{k+1} | \hat{M}_k, a_k, D) \approx \mathbb{P}(m_{k+1} | \hat{m}_k, a_k, D). \quad (3.9)$$

As seen from (3.9), our task now is to utilize experience D to approximate the predictive distribution over future sub-map m_{k+1} , given current (estimated) sub-map \hat{m}_k and action a_k . Recalling that we focus on a setting where the action takes us to an *unobserved area*, our hypothesis is that experience-based map predictions can particularly be beneficial. Thus, we aim to learn *offline* the conditional distribution $\mathbb{P}(m_{k+1}|C, D)$ given training data D and conditional C , and query it *online* with the actual conditioned data $\{\hat{m}_k, a_k\}$.

3.3 Offline Training Phase

Given offline available environment maps $\mathcal{M}_D \doteq \{M^i\}$ and action space $\mathcal{A} \doteq \{a\}$, and recalling Eq. (3.9), we define experience as

$$D \doteq \{(m, a, m') | m, m' \in M^i, \forall M^i \in \mathcal{M}_D, a \in \mathcal{A}\}, \quad (3.10)$$

where, each tuple (m, a, m') corresponds to a submap m observed from some robot pose, an executed action a which transitions the robot to a different pose, from which submap m' is observed. The experience D is thus constructed by randomizing these for different environment maps in \mathcal{M}_D . Finally, we are interested in learning a function that maps from the conditioning $C \doteq (m, a)$ to a distribution over m' , i.e. $\mathbb{P}(m'|C, D)$, considering different realizations of C from the dataset D . To reduce clutter, in the following we shall omit the explicit conditioning on D .

In our work we use a Conditional Variational Autoencoders (CVAE) to approximate the distribution $\mathbb{P}(m'|C)$. We now briefly present this formulation for self-containment (see, e.g., [36]). First, we marginalize over a latent variable z and re-write $\mathbb{P}(m'|C)$ as

$$\mathbb{P}(m'|C) = \int_z \mathbb{P}(m'|z, C) \mathbb{P}(z|C) dz. \quad (3.11)$$

Then, as standard in CVAE, we set $\mathbb{P}(m'|z, C; \theta) = \mathcal{N}(f(z, C; \theta), \sigma^2 * I)$, and refer to it as the *decoder*, where $f(\cdot; \theta)$ is a deterministic function parametrized by θ and σ is a typically small fixed hyperparameter. The integral (3.11) can be approximately calculated by sampling z from $\mathbb{P}(z|C)$, which is typically a very simple distribution (e.g. a Gaussian). Yet, in practice, for many such samples of z , $\mathbb{P}(m'|z, C; \theta)$ will be practically zero. Hence, the key idea in VAE is to sample z from another distribution, $\mathbb{Q}(z|m', C; \phi)$, such that these samples are more likely to generate m' . This latter distribution, denoted as the *encoder*, is modeled in practice as a Gaussian $\mathcal{N}(\mu(m', C; \phi), \Sigma(m', C; \phi))$ where $\mu(\cdot)$ and $\Sigma(\cdot)$ are deterministic functions parameterized by ϕ . Based on the definition of Kullback-Leiber divergence (KL) between $\mathbb{Q}_z \doteq \mathbb{Q}(z | m', C; \phi)$ and

$\mathbb{P}(z | m', C)$ (see more details in [7]), we get the main equation of VAE:

$$\begin{aligned} \log \mathbb{P}(m' | C) - \text{KL}[\mathbb{Q}(z | m', C; \phi) \| \mathbb{P}(z | m', C)] = \\ \mathbb{E}_{z \sim \mathbb{Q}_z} \{\log \mathbb{P}(m' | z, C; \theta)\} - \text{KL}[\mathbb{Q}(z | m', C; \phi) \| \mathbb{P}(z | C)] \end{aligned}$$

In order to minimize $\text{KL}[\mathbb{Q}(z | m', C) \| \mathbb{P}(z | m', C)]$, since it is non-negative by definition and $\log \mathbb{P}(m' | C)$ is independent of θ and ϕ parameters, we can maximize just the right hand side, which is known as the Evidence Lower Bound (ELBO). A common assumption in VAE is that the approximation of the expectation over $z \sim \mathbb{Q}(z | m', C; \phi)$ is done only by a single sample. Also, to make the above equation tractable via backpropagation, the re-parameterization trick is used [7]. Next, we will isolate the ELBO and substitute the explicit Gaussian models in place of $\mathbb{P}(m' | z, C; \theta)$ and $\mathbb{Q}(z | m', C; \phi)$. Maximizing the ELBO is equivalent to minimizing the loss function for a given sampled tuple $(m', C) \sim D$, defined as

$$\begin{aligned} l(\theta, \phi; m', C) \doteq \|m' - f(z, C; \theta)\|^2 \\ + \text{KL}[\mathcal{N}(\mu(m', C; \phi), \Sigma(m', C; \phi)) \| \mathcal{N}(0, I)]. \end{aligned} \quad (3.12)$$

The first term is a measure of the "reconstruction error", the error between the decoder output and the ground truth. The second term is KL divergence, an operator that evaluates the distance between the encoder output and the target distribution. In our case the target distribution $\mathbb{P}(z | C)$ is a normal distribution $\mathcal{N}(0, I)$ to allow easy sampling in the online stage. The loss (3.12), considering the entire dataset D , can be written as

$$Loss(\theta, \phi; D) = \sum_{m', C \sim D} l(\theta, \phi; m', C). \quad (3.13)$$

Finally, the encoder and decoder weights θ and ϕ , are optimized as

$$\phi^*, \theta^* = \arg \min_{\phi, \theta} Loss(\theta, \phi; D). \quad (3.14)$$

In practice, this optimization is done via standard stochastic gradient methods, i.e over mini-batches which are subsets of D . The CVAE flow chart in the training and deployment stages is shown in Fig. 3.1.

Remark: We note that whereas in this work, ideal submaps for training are considered (see Eq. (3.10)), i.e. $m, m' \in M^i$, one could instead use the estimated maps \hat{m}, \hat{m}' that would be obtained by performing SLAM offline. We leave this aspect to future work.

3.4 Online Deployment

Having described the offline learning of the conditional distribution $\mathbb{P}(m_{k+1} | C, D)$, we now focus on the deployment stage, considering a planning session at time instant k . Recalling Eq. (3.9), and the sub-map estimate \hat{m}_k , we resort to a sampling-based approximate representation of the

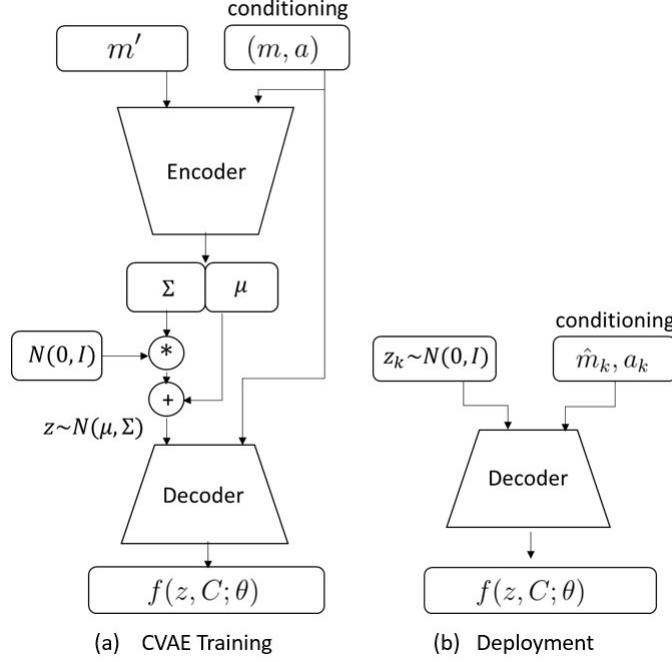


Figure 3.1: Training and Deployment.

distribution $\mathbb{P}(m_{k+1}|\hat{m}_k, a_k, D)$ for different candidate actions a_k . Observe that this distribution is conditioned on data \hat{m}_k, a_k that generally is different from the conditioning C considered in the offline training phase. We shall come back to this key point later on.

The conditional distribution can be expressed as

$$\mathbb{P}(m_{k+1} | \hat{m}_k, a_k, D) = \int_z \mathbb{P}(m_{k+1} | z, \hat{m}_k, a_k, D) \mathbb{P}(z | \hat{m}_k, a_k, D) dz,$$

which can be approximated via sampling as

$$\mathbb{P}(m_{k+1} | \hat{m}_k, a_k, D) \approx \frac{1}{n_z} \sum_{z \sim \mathbb{P}_z} \mathbb{P}(m_{k+1} | z, \hat{m}_k, a_k, D),$$

where n_z is the number of samples, and $\mathbb{P}_z \doteq \mathbb{P}(z | \hat{m}_k, a_k, D)$.

Given a trained decoder $\mathbb{P}(m'|z, C; \theta) = \mathcal{N}(f(z, C; \theta), \sigma^2 * I)$, see Section 3.3, the distribution $\mathbb{P}(m_{k+1}|\hat{m}_k, a_k, D)$ can be approximately represented by a Gaussian mixture model

$$\mathbb{P}(m_{k+1} | \hat{m}_k, a_k, D) \approx \frac{1}{n_z} \sum_{z \sim \mathcal{N}(0, I)} \mathcal{N}(f(z, \hat{m}_k, a_k; \theta^*), \sigma^2 * I), \quad (3.15)$$

where, as standard in VAE, we consider $\mathbb{P}(z | \hat{m}_k, a_k, D) = \mathbb{P}(z) = \mathcal{N}(0, I)$. As the decoder is learned, i.e. function $f(\cdot; \theta^*)$ is deterministic at this point, sampling z corresponds to choosing one of the components in the GMM representation (3.15), from which we can easily sample a realization of the future sub-map i.e. $m_{k+1} \sim \mathcal{N}(f(z, \hat{m}_k, a_k; \theta^*), \sigma^2 * I)$. See illustration of this process in Fig. 3.1(b). The obtained predicted maps m_{k+1} are then used for generating future *raw* observations y_{k+1} , see Eq. (3.6). These raw measurements, along with an appropriate

measurement likelihood model are used to update the future posterior belief b_{k+1} , followed by calculation of the cost function. In the non-myopic case, we repeat for L look-ahead steps the above mentioned process, when the map prediction from previous time, m_{k+j-1} is used as conditioning for the next prediction m_{k+j} . Recalling the empirical expectation from Eq. (3.3), this entire process is repeated N times for each candidate action sequence, each time with a different sampled sequence of future raw observations. The entire process is summarized in Algorithm 3.1.

Algorithm 3.1 BSP with Experience-Based Prediction

```

1: Inputs:
2:  $b_k$ : state belief at current time
3:  $\mathbb{P}(M_k|H_k)$ : map belief at current time
4:  $a_{k:k+L-1}$ : a candidate  $L$  look-ahead steps action sequence
5:  $f(\cdot; \theta^*)$ : trained decoder
6: Outputs:
7:  $J(b_k, a_{k:k+L-1})$ : computed objective function for a given action sequence
8:
9:  $\hat{M}_k \leftarrow \mathbb{P}(M_k|H_k)$                                 ▷ Get maximum likelihood estimate of map belief
10:  $\hat{m}_k \subseteq \hat{M}_k$                                 ▷ Get current sub-map estimate from  $\hat{M}_k$  (Eq. (3.9))
11: for  $i = 1 : N$  do
12:    $m_k^i = \hat{m}_k$ 
13:   for  $j = 1 : L$  do
14:     ▷ Get ML estimate without future observations
15:      $\hat{x}_{k+j}^- \leftarrow \mathbb{P}(x_{k+j}|H_k, a_{k:k+j-1})$ 
16:      $z^j \sim \mathcal{N}(0, I)$ 
17:     ▷ Predict sub-map (Eq. (3.15))
18:      $m_{k+j}^i \sim \mathcal{N}(f(z^j, m_{k+j-1}^i, a_{k+j-1}; \theta^*), \sigma^2 * I)$ 
19:     ▷ Generate future observation (Eq. (3.6))
20:      $y_{k+j}^i \sim \mathbb{P}(y_{k+j} | m_{k+j}^i, \hat{x}_{k+j}^-)$ 
21:     Calculate  $b_{k+j}^i$                                 ▷ Calculate future belief using  $y_{k+j}^i$  (Eq. (2.8))
22:     Calculate cost/reward  $c(b_{k+j}^i)$ 
23:     ▷ Accumulate costs
24:      $J(b_k, a_{k:k+L-1}) = J(b_k, a_{k:k+L-1}) + c(b_{k+j}^i)$ 
25:   end for
26: end for
27: ▷ Normalize to get empirical expectation
28:  $J(b_k, a_{k:k+L-1}) = \frac{1}{N} J(b_k, a_{k:k+L-1})$ 
29: return  $J(b_k, a_{k:k+L-1})$ 

```

3.5 Novelty Detection

It is common to assume that the environments considered in the offline training should be representative and similar to the environment where the robot is actually deployed. More specifically, these environments can be implicitly characterized by a distribution $\mathbb{P}(M)$, that could correspond to, e.g., typical apartments, office environments, underground mines, etc. In practice, however, we do not have access to such a distribution; instead, the environment maps $\mathcal{M}_D \doteq \{M^i\}$ used for training should (at least) approximately represent $\mathbb{P}(M)$. The actual environment map M^{online} is assumed to be of a similar nature, i.e. $M^{\text{online}} \sim \mathbb{P}(M)$. Note that this assumption does *not* imply $M^{\text{online}} \in \mathcal{M}_D$. For example, this would correspond to the setting where the robot is deployed in a previously unseen office environment, while training data captures typical office environments.

Moreover, in the considered SLAM setting, $M \doteq M^{\text{online}}$ is not available to the robot; rather,

at time k , the robot observed with its (noisy) sensors only part of the environment, $M_k \subseteq M$, and maintains a belief over it, i.e. $\mathbb{P}(M_k | H_k)$. Therefore, when using algorithms based on experience we need to determine how much the experience is relevant and reliable for the current task. While this is a very active research area on its own (see e.g. [11, 30, 32]) and is outside the scope of this paper, in this work we consider the simpler problem of novelty detection, aiming to decide whether to use the experience-based map prediction or not.

One option for novelty detection is the autoencoder-based approach that was first proposed by Japkowicz et al. [18], and used recently in the navigation domain by Richter et al. [35]. With this approach we would need to train a separate autoencoder using the same training set D , and calculate the reconstruction error $RE(m) = \|m - Dec(Enc(m))\|^2$ for $m \in M \in \mathcal{M}_D$. At the deployment stage we would calculate $RE(m_k)$ based on the actual sub-map m_k we have at the current time k and compare it to the typical RE obtained with the training set. In case $RE(m_k)$ is significantly higher (e.g. compared to a manually determined threshold), we would decide the available experience is less relevant for the current planning session and use a standard BSP method instead (i.e. without predicting unexplored maps). See more details, e.g. in [35].

Noting the above approach requires training separate NNs for map prediction and novelty detection, we suggest a simple method that executes these missions concurrently. In our method, we assume there is an overlap area (OA) between the conditional input m_k and the map prediction $f(z, m_k, a_k; \theta^*)$, which the NN should learn to copy shifted according to the action. We therefore, suggest to measure the error in the copy operation of this overlapping area, i.e. for sub-map m_k calculate $RE(m_k) = \|\{m_k\}^{OA} - \{f(z, m_k, a_k; \theta^*)\}^{OA}\|^2$, where $\{m\}^{OA}$ denotes the corresponding overlapping area in sub-map m . Finally, we can use a threshold over this re-defined RE , similarly to the above-mentioned autoencoder-based novelty detection approach.

Chapter 4

Results

We evaluate the performance of our approach in two steps. First, we examine the map prediction algorithm using a dataset of real floor-plans. Then, we study the performance of our experience-enhanced BSP in a realistic simulation of autonomous navigation in an unknown environment.

4.1 Map Distribution Prediction With CVAE

In this section we show the implementation for map distribution prediction as described in Section 3.3, while discussing success and failure cases.

The dataset that was used is the KTH dataset [1] which includes 182 floor-plans and nearly 38,000 real-world rooms. The XML files of floor-plans were converted to 2D occupancy grid maps with fixed scale (four pixels for one meter). Next, each map was cut into sub-maps with fixed size (32/32 pixels). Finally, we created N tuples of sub-maps, action (direction and length of stride) and the ground truth map post action i.e (m, a, m') (see Fig. 4.1). The actions between the sub-maps were defined by one-hot vector that represents the four possible stride directions (up, down, left, right). In the training stage we used the KTH dataset (593,264 tuples of (m, a, m') in indoor environment) and separated this data to 80% train set and 20% test set.

We examined the prediction performance on testset maps that were not part of the training process. Fig. 4.2 demonstrates four examples of prediction results. For each prediction we calculate the reconstruction error (RE) on the overlap area by the novelty detection method that was presented in Section 3.5, and additionally the prediction error (PE) on the unknown area against the ground truth. Examples in (I) show successful prediction results even in the

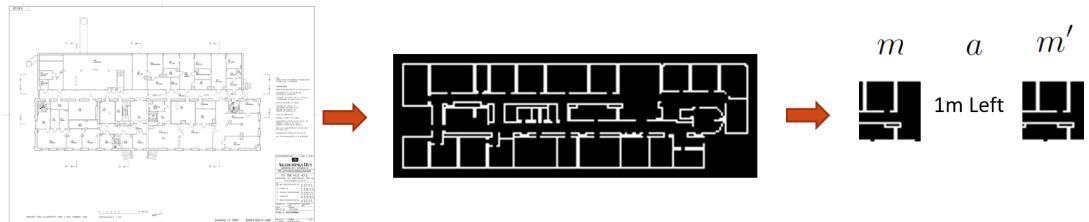


Figure 4.1: Process of organization the dataset.

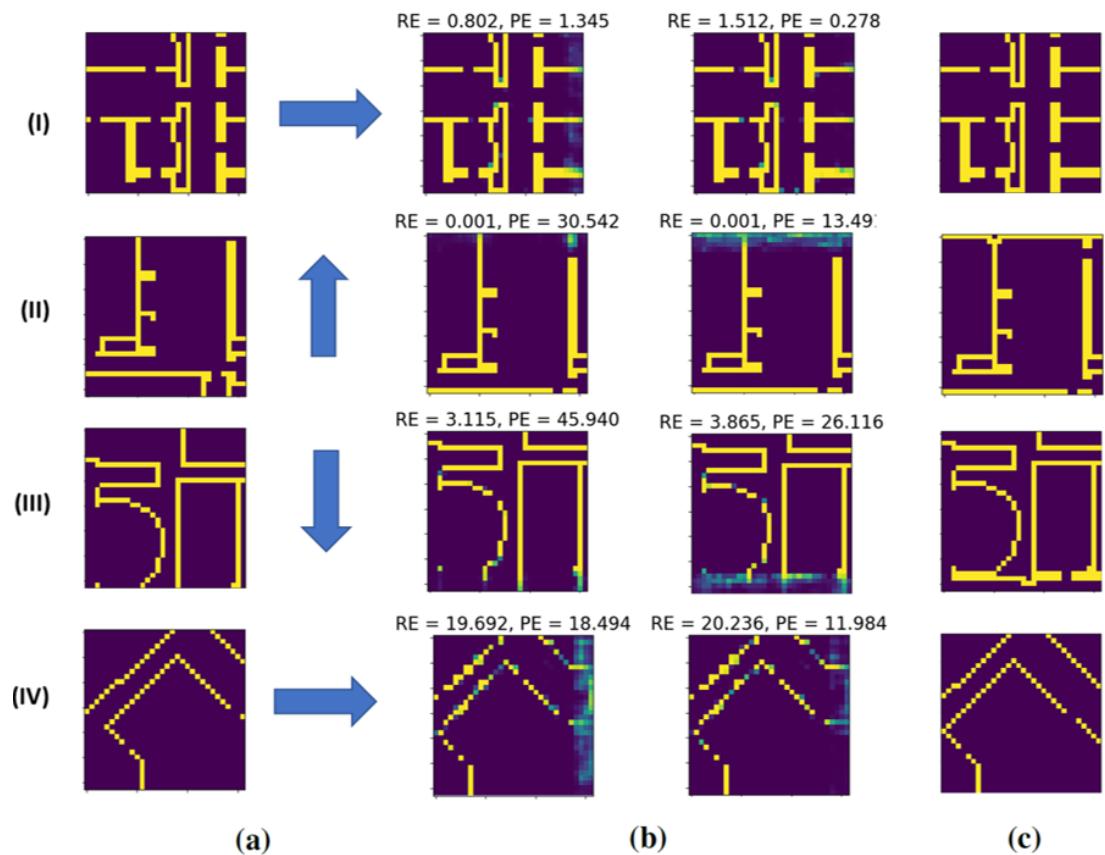


Figure 4.2: Examples of the prediction algorithm. (a) Inputs - sub-map and action; (b) Outputs - two samples of prediction with reconstruction error (RE) and prediction error (PE). On the left worst prediction result and on the right best prediction result; (c) Ground truth.

worst case, while in example (II) most predictions had mistakes, since they mostly did not predict the wall that closed the room. It is possible that since the trivial solution is that the same wall continues further as in Fig. 4.2c.I, and does not close into a room as in Fig. 4.2c.II, the NN predictions will mostly present the solution that appeared a greater amount of times in the training dataset. However, our results are probabilistic, and therefore, the non-trivial cases (right sample in example II) are still represented. Examples (III) and (IV) show a conditioning with an uncommon wall shape (circular or diagonal). In this case, most predictions have failed, since the training dataset included only very few to none cases of such walls.

These examples represent three families of experience-based prediction results: (I) most predictions are correct (low PE); (II) most predictions are wrong because of uncommon ground truth map (high PE and low RE); (III) most predictions are wrong because of an unfamiliar input (high PE and high RE). Using the novelty detection method from Section 3.5 we can identify online cases with unfamiliar input, and hence, avoid using DL-based predictions. In contrast, the second family type is problematic for our method, as identifying it online is impossible; thus, in such setting, wrong DL-based predictions could disturb rather than improve the decision making. Nevertheless, statistically we assume that in most cases the online map will be of a similar nature as the dataset maps (as discussed in Section 3.5), and therefore, most cases will be from the first family above and, thus, improve the decision making process.

For more statistical results we calculate the RE and PE for entire KTH dataset. Figs. 4.3.a-b show the RE histogram of the train set and the test set: we can see very similar statistics in both of the sets, since both are of the same environment type (KTH dataset). Based on these results, we can define the novelty detection threshold to be 5 (See section 3.5). In addition, in Fig. 4.3.c-d we show the PE histogram of the train set and the test set. We can see most of the results are with prediction error less on 5.

4.2 BSP in Unknown Environments Simulation Results

In this section we examine our experience-enhanced BSP approach and compare it to an existing BSP approach, in a realistic Gazebo simulation, considering autonomous navigation in an unknown environment. The simulation setting is a Pioneer robot that navigates autonomously in a 3D Gazebo world (see Fig. 4.4a) using odometry and Lidar sensors. The odometry sensor provides relative measurements with a constant motion model. The Lidar sensor provides laser scans that are used to build a map, and for relative measurements via ICP with an environment-dependent model [3]. The ICP measurement model function needs as input raw (LIDAR) measurements and provides covariance that represents the uncertainty of the ICP measurement (calculated relative pose based on two laser scans). In our setting the ICP measurements are more accurate than the odometry, and therefore, preferable by default; but when ICP fails (un-matching two scans) the odometry will be used instead. Our pose-SLAM implementation uses GTSAM [6] and the mapping process uses OctoMap [14].

In the planning stage, the robot starts from a defined point and gets a set of actions randomly by the PRM method or manually by the user. The robot's mission is to choose the best action by

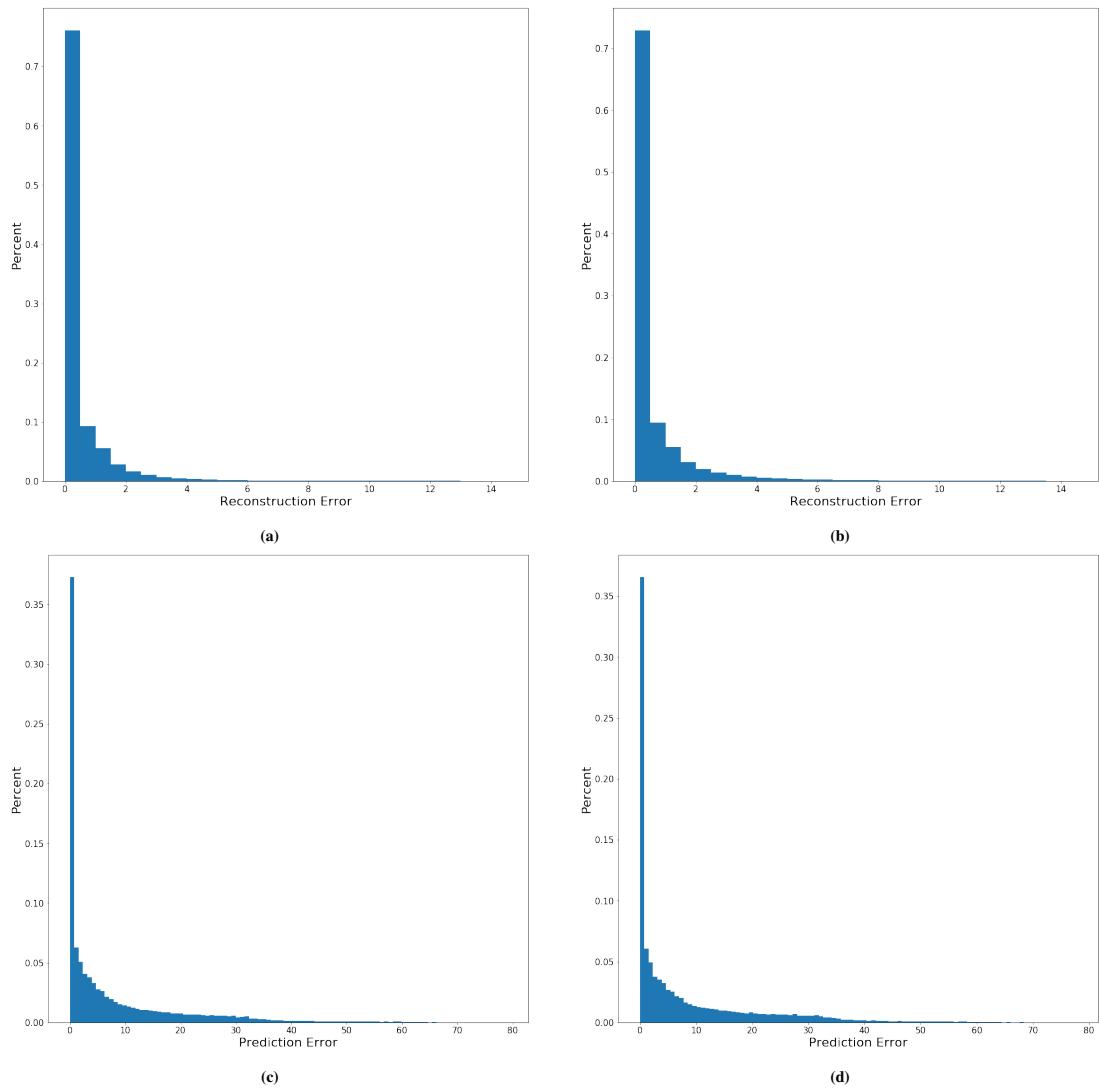


Figure 4.3: Reconstruction error histogram: (a) trainset; (b) testset. Prediction error histogram: (c) trainset; (d) testset.

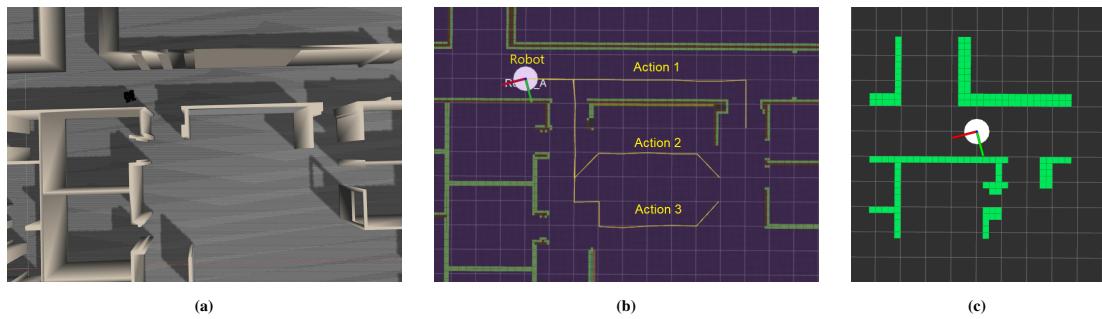


Figure 4.4: (a) 3D Gazebo simulation world; (b) Planning session with three candidate actions; (c) Occupancy grid of the partial map that was used for prediction.

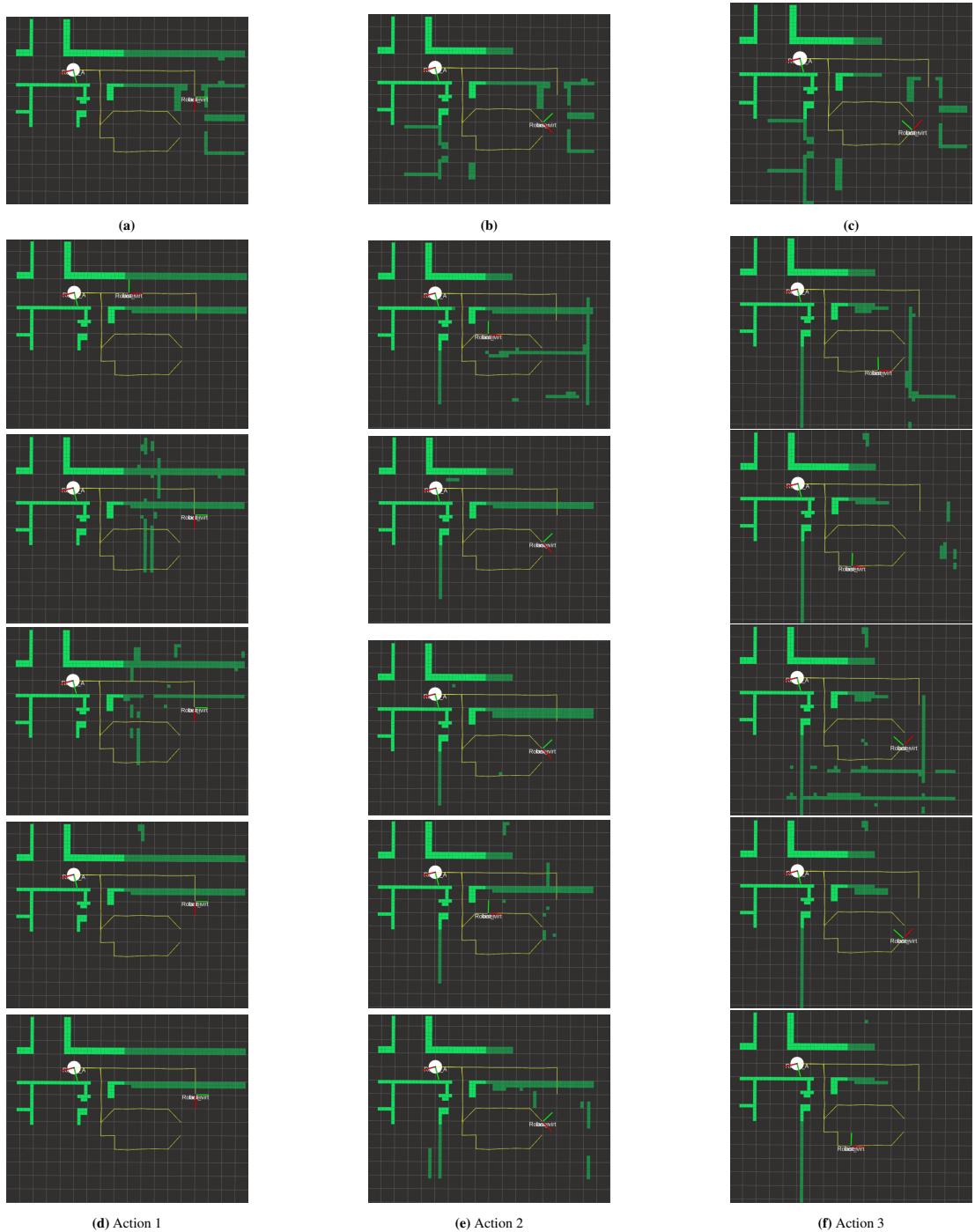


Figure 4.5: Map prediction results for three actions from Figure 4.4b. Ground truth maps in relevant regions are shown in (a) for action 1, in (b) for action 2, and in (c) for action 3. Figures (d), (e) and (f) show five samples of map predictions for each of the actions. Conditional map in light green, predicted map in dark green.

calculating the objective function for each action. The main question in this work is how we can do this calculation when the action places the robot in an unknown area. The solution by a standard BSP method, denoted **baseline**, is to ignore the unknown future measurements (in our case point clouds) and take into consideration only the motion model. The non-realistic solution, denoted **GT-map**, is to use the ground truth maps to generate the expected future measurements and take them into consideration in the objective function calculation. Our approach suggests to leverage experience to predict the unknown area around the candidate actions given the partial map observed in the inference stage. In our evaluation below, we compare our approach to the **baseline** BSP method and investigate which one is closer to the **GT-map** BSP method.

We implemented our approach based on Algorithm 3.1, and used the DL-based prediction from section 4.1, i.e. using the KTH dataset for training while being deployed in a previously unseen Gazebo environment. In order to get a binary map from the DL-based prediction function we used a constant threshold of 0.3 that was determined offline. As described, the prediction was done for an action sequence of L steps, where each step contains one sample. In this example this prediction was done five times ($N = 5$). For each map prediction we generated a laser scan that was used to generate relative pose measurements via ICP. Also leveraging the approach from [3] we used two laser scans to get the measurement likelihood model, i.e the measurement uncertainty covariance. These relative measurements, along with an appropriate measurement likelihood model are used to perform belief propagation and calculate the cost function.

The cost function that we defined in our simulation is a function of the uncertainty at the end of the trajectory, i.e. $\sqrt{\text{Trace}(\Sigma_{k+L})}$. For objective function calculation we average the cost function of all samples, i.e

$$J(b_k, a_{k:k+L-1}) \doteq \frac{1}{N} \sum_{i=1}^N \sqrt{\text{Trace}(\Sigma_{k+L}^i)}. \quad (4.1)$$

Since the measurement uncertainty covariance is environment-dependent, the decision making process in our setting depends on the DL-predicted map.

We present an example of a planning mission when there are three candidate actions (see Fig. 4.4b) and a partial map presented by an occupancy grid with equal scale of the dataset (Fig. 4.4c). All three candidate trajectories (non-myopic actions) lead to the pre-defined goal. Note that the partial map used is a ground truth sub-map and not from the belief. This was done for reasons of simplification since in the current work our focus was the planning stage.

In Fig. 4.5 we show for each action the ground truth map (top row) against five samples of map predictions (map predictions are shown by dark green color). Fig. 4.5d shows five predictions for action 1; all samples predicted a long corridor similar to the ground truth (Fig. 4.5a). On the other hand neither of the samples predicted the opening on the right. Figures 4.5e and 4.5f show prediction results for action 2 and 3, where we can see more open space and kind of rooms similar to the real environment around these actions.

In Fig. 4.6 we show uncertainty evolution for the three actions and three BSP methods. For action 1 even though the map prediction is not perfectly accurate, our approach predicts the

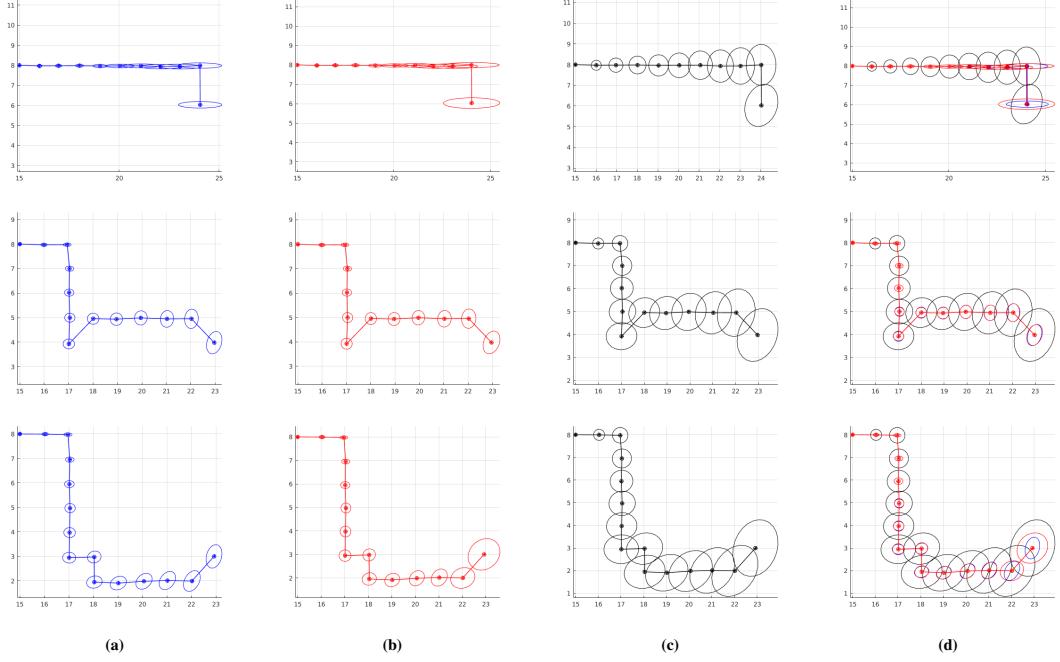


Figure 4.6: Comparison between three methods of uncertainty evolution on three different actions from Figure 4.4b: Action 1 - top row; action 2 - middle row; action 3 - bottom row. Uncertainty evolution using (a) ground truth map (GT-map), (b) using one sample of prediction map (our approach), and (c) using motion model only (baseline). Column (d) presents a comparison between all three methods. In column (b), results are shown for one sample from Figure 4.5 for each action: action 1 - first row in 4.5d, action 2 - fifth row in 4.5e, action 3 - forth row in 4.5f. Results for all samples are summarized in Figure 4.7a. For convenient visualization covariance resolution was multiplied by 100.

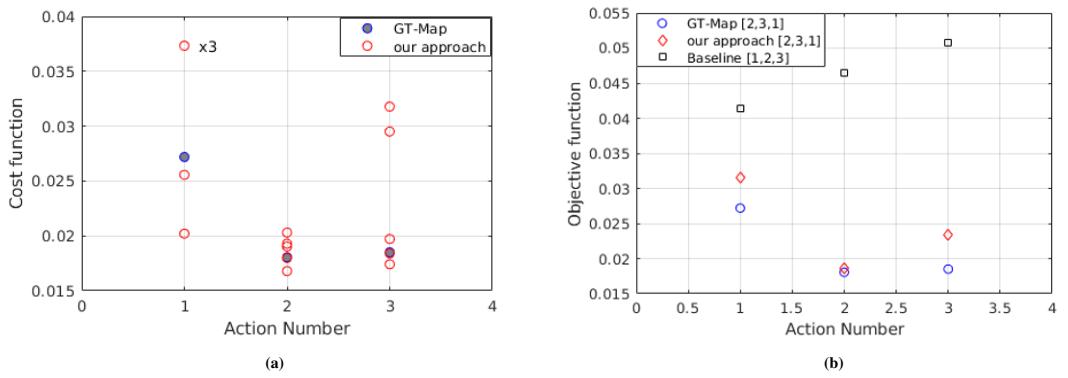


Figure 4.7: (a) Cost function results of all the samples from Fig. 4.5 compared to cost function calculated with the GT-map method. (b) Objective function (4.1) values for the three methods. In contrast to baseline BSP, our approach preserves action-ordering with respect to GT-map.

uncertainty shape better than the **baseline** method. In a corridor environment we expect to get laser scans without corners and correspondingly high uncertainty in the corridor direction. Additionally, for action 2 the uncertainty evolution by our approach was very similar to the GT-map method. In contrast, action 3, for this particular map prediction sample, predicts an open space at the end of the path, causing the generated laser scans to be with insufficient points for ICP matching; thus, in this part, our approach fallbacks to the **baseline** method.

Fig. 4.7a summarizes the cost function results of all the samples compared to the GT-map cost function calculation. For action 1 the samples are spread, while in action 2 all five samples were very close to the GT-map cost. In action 3, three samples (see rows 1,2,3 in Fig. 4.5f) were close and two other predicted higher uncertainty compared to the GT-map cost (see rows 4,5 in Fig. 4.5f). The density of the samples could be used as an indication of the prediction confidence.

Fig. 4.7b shows the calculated objective function (4.1) for the BSP methods, considering the three candidate actions and $N = 5$ samples. The **baseline** approach, which only considers a motion model, yields action ordering [1,2,3] i.e., action 1 is chosen. However, in our approach, action ordering is [2,3,1] which is the same as the approach that has access to the ground truth map, i.e., action 2 is chosen since our algorithm predicted this action will give future measurements that are more informative than action 1 and 3. Thus, in this scenario our approach had no ordering mistakes, while the **baseline** had two mistakes (without double counting).

Finally, fifteen scenarios of planning sessions are tested and summarized in Table 4.1. Each scenario includes a different environment and three actions that lead to an unknown area, where the first fourteen scenarios are from indoor environments and scenario 15 is very different from the dataset (see in Fig. 4.8 the planning sessions and in Fig. 4.9 the objective function comparison of these scenarios). We calculated the reconstruction error for all scenarios and showed that we recognized unfamiliar environments and avoided using our approach in these cases. We can see, that using the **baseline** BSP method in an unknown environment is insufficient and could cause a lot of decision mistakes compared to GT-map. On the other hand, our approach showed a significant performance improvement, in nine out of fifteen scenarios yielded fewer mistakes against only one case when the **baseline** method was preferable. Moreover we qualitatively compared the error that represents the uncertainty cost of making mistakes in action ordering, i.e., $100\% \cdot (J^{GT}(b, a') - J^{GT}(b, a^*) / J^{GT}(b, a^*)$. Here, a^* denotes the optimal action by GT-map and a' the chosen action by each approach. We can see that our approach yields an improved uncertainty cost error.

Extended Results - Bad Scenario Discussion

In this section, we focus on scenario 2 from Table 4.1) which was the only one where the **baseline** was preferable. Fig. 4.10a shows the planning session of scenario 2 where the robot mission was to choose the best action to execute from three candidate actions. Fig. 4.10b shows the partial map used as a condition for the map distribution prediction task. The main question here is: why in this case the map prediction failed? And perhaps it is more important to ask: can

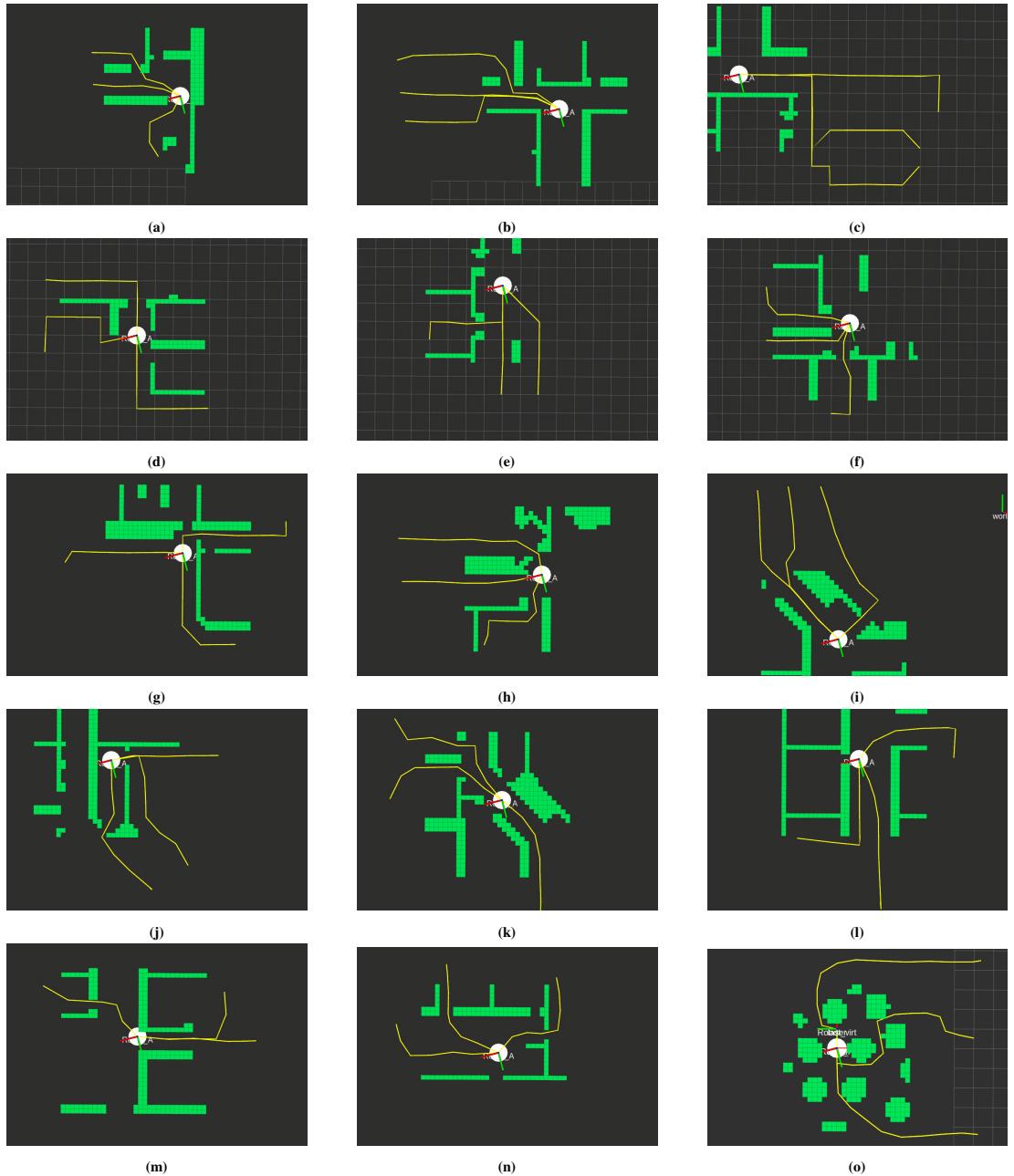


Figure 4.8: Light green - coditional map, yellow - three candidate actions. Figures (a)-(o) according to planning scenarios 1-15 from Table 4.1.

Table 4.1: Performance of our and baseline approaches in 15 different scenarios. Each scenario includes a different environment and three actions. Several examples of planning settings are shown at the bottom. The table reports for each method the number of action ordering mistakes with respect to BSP with ground truth map (GT-map), and the uncertainty cost error. We also show the reconstruction error (RE) calculated by our novelty detection approach.

Scenarios	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Avg
Our approach																
Mistakes	0	2	0	1	0	0	0	0	0	2	0	0	1	0	-	0.43
Error[%]	0	40	0	0	0	0	0	0	0	4	0	0	0	0	-	2.9
RE	0.3	1.1	1.4	0.6	0.8	0.1	1.1	2.2	2	1.6	3.8	1.4	1	0	10.4	-
Baseline																
Mistakes	0	1	2	2	1	2	0	1	1	2	1	1	1	1	0	1.07
Error[%]	0	1	49	6	10	41	0	20	0	12	0	0	16	0	0	10.3

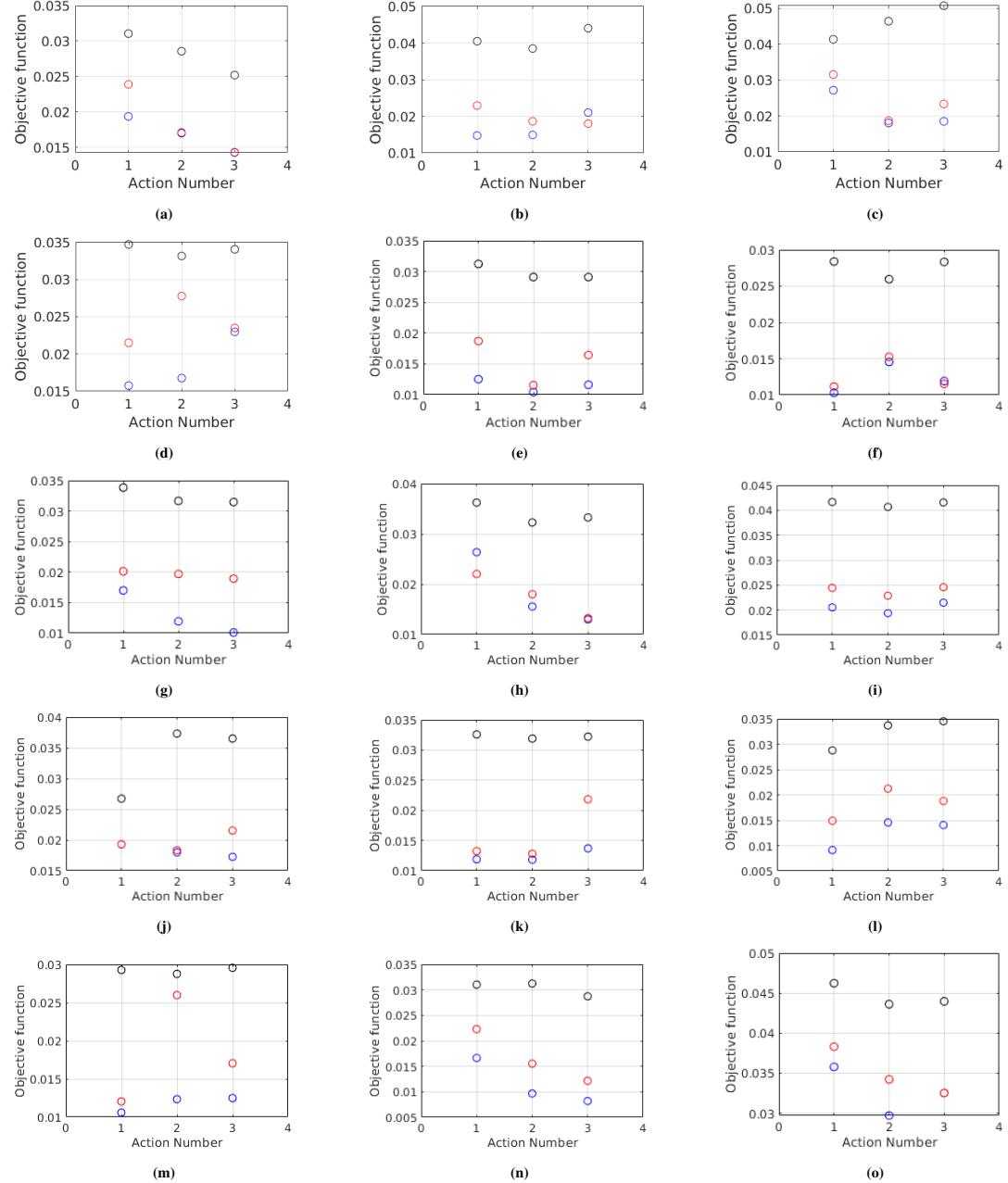


Figure 4.9: Objective function values for the three BSP methods. baseline in black, our approach in red, GT-map in blue. Figures (a)-(o) according to planning scenarios 1-15 from Table 4.1.

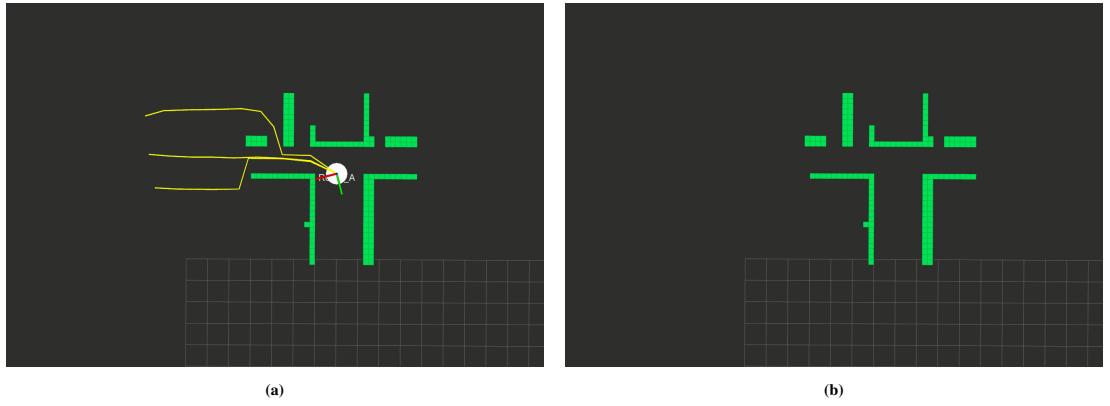


Figure 4.10: (a) Planning session of scenario 2 with three candidate actions; (b) Occupancy grid of the partial map that was used for prediction.

we recognize it online and avoid using the prediction algorithm in these cases?

First of all, we can see that this scenario was not detected as unfamiliar input by the novelty algorithm, since the reconstruction error was low. In section 4.1 bad predictions with low RE associate to the second prediction family: most predictions are wrong because of the uncommon ground truth map. We show that statistically, in most cases the online map will be of a similar nature as the dataset maps, and therefore, most cases will be from the first family above and, thus, improve the decision making process.

As seen in Fig. 4.10a, there are many possible predictions of the environment along the candidate actions. This variety is also expressed by the map prediction algorithm. Fig 4.11 shows the uncertainty evolution of three different actions from scenario 2. In red, we can see three of the five samples from the prediction map for each candidate action. The uncertainty evolution predictions are very different from each sample. The variance between the samples can be seen more clearly in Fig. 4.12.a. For reference, Fig. 4.13.a shows the variance between the samples of successful scenarios.

We can see from this case that the variance between the samples could be connected to the prediction confidence. Therefore, we suggest for future work to calculate this variance at each planning session and utilize it using one of these two methods:

- Define a threshold of confidence, similarly to the novelty detection approach, and in case we surpass that threshold use the baseline method instead.
- Weigh the confidence in calculating a weighted objective function between the baseline and map prediction approaches.

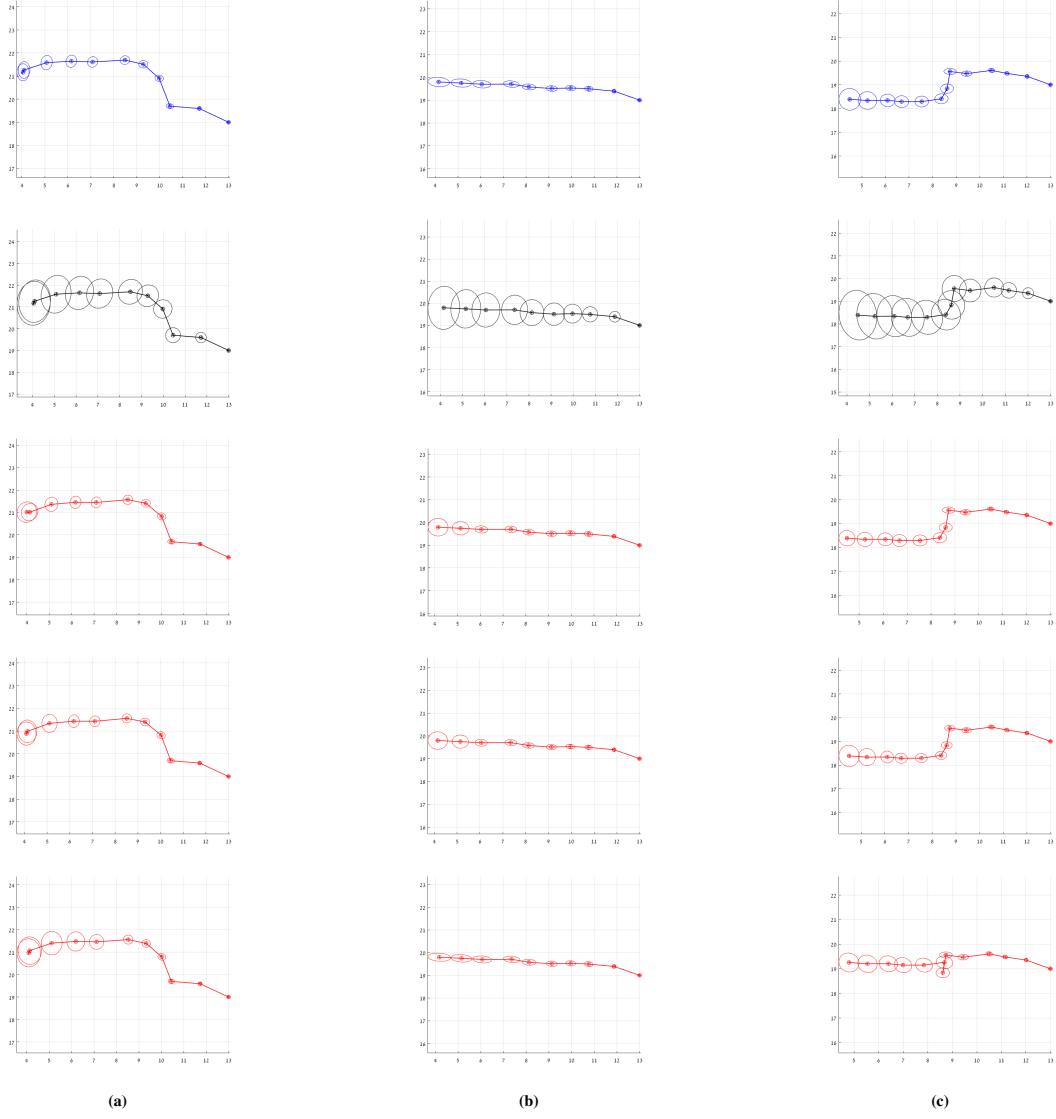


Figure 4.11: Comparison between three methods of uncertainty evolution on three different actions from scenario 2: (a) Action 1; (b) action 2; (c) action 3. In blue - uncertainty evolution using ground truth map (GT-map); In black - using motion model only (baseline); and in red - using samples of prediction map (our approach). For convenient visualization covariance resolution was multiplied by 100.

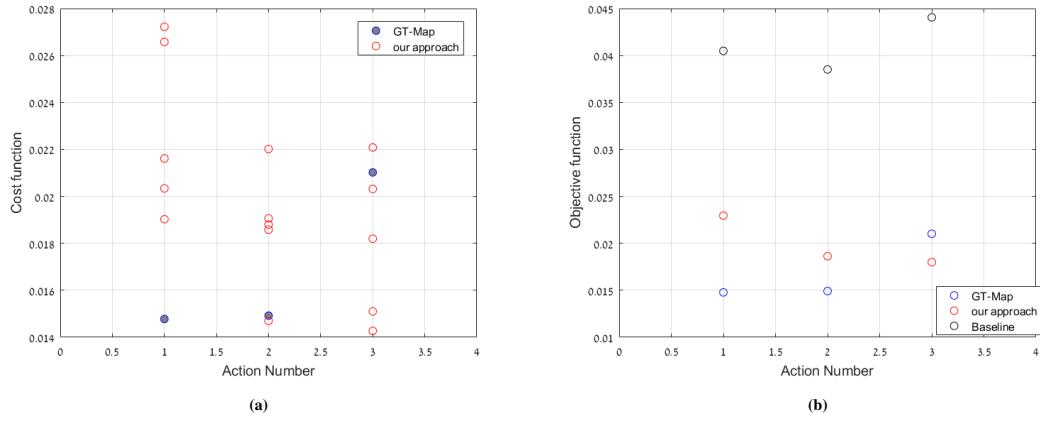


Figure 4.12: (a) Cost function results of all the samples from scenario 2 compared to cost function calculated with the GT-map method. (b) Objective function values for the three methods in scenario 2.

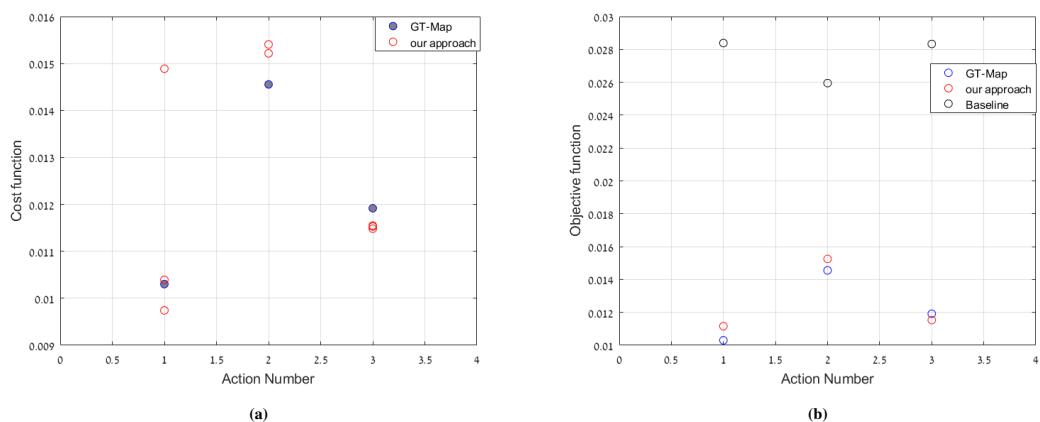


Figure 4.13: (a) Cost function results of all the samples from scenario 6 compared to cost function calculated with the GT-map method. (b) Objective function values for the three methods in scenario 6.

Chapter 5

Conclusions and Future Work

We developed a novel approach for belief space planning (BSP) in unknown environments. As a key contribution, we developed an algorithm to calculate a predicted distribution over an unexplored area using a deep learning method and incorporated this distribution within BSP. The approach has been examined in autonomous navigation scenarios in a Gazebo simulation. Simulation results demonstrated that with our approach the decision making in most cases was closer to BSP using (the unavailable) ground truth map, against an existing BSP approach. These findings indicate the potential of our approach to improve decision making in unknown environments. Furthermore, we suggested a novelty detection method to avoid using unfamiliar inputs in the prediction.

In addition, we believe a benefit of our approach is in its interpretability, since the use of experience in our method is done only at the prediction level, as opposed to end-to-end methods which involve experience all through the decision making process. Moreover, as our work focused on the uncertainty estimation of future observations, it depends mostly on the type of unexplored area (e.g. corridor or room) rather than the exact outline, and therefore is less sensitive to prediction mistakes. One could also envision utilizing a similar concept also for evaluating path feasibility; however, to this end, further work is needed to improve map prediction accuracy.

5.1 Future Work

Although a combination of learning methods and traditional algorithms in robotics tasks like planning and navigation has become very popular in the last years, we found only few works on planning under uncertainty in unknown environments in the learning area. Therefore, we believe that there is a lot of possible directions to take this research forward. In this section, our suggestions for future works are divided into 2 main topics: experience-based prediction and the reliability question of incorporating experience in the BSP algorithm.

In this work, we chose the CVAE architecture for the map prediction task. Besides great progress in generative models in last years, one possible research direction is to use state of the art of generative models and adjust them for map completion. In this work, we tried conditional

GAN [17] or combination of RNN with VAE [12] as alternative architectures, but without improvement of accuracy and did not exhaust these directions. Another direction is to generate future images measurements directly by generative models (see [9]) in a vision SLAM setting and use them for our decision-making algorithm.

Future work may extend the novelty detection solution to cases with familiar inputs that still provide wrong predictions, in order to detect which cases using experience could assist or disturb the decision making process. In section 4.2 we showed how the only failed case could have been recognized by high variance of the prediction. Yet, this solution was just introduced in this work and should be thoroughly examined in future work.

Bibliography

- [1] Alper Aydemir, Patric Jensfelt, and John Folkesson. What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4675–4682. IEEE, 2012.
- [2] 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. *IEEE Trans. Robotics*, 32(6):1309 – 1332, 2016.
- [3] Andrea Censi. An accurate closed-form estimate of icp’s covariance. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3167–3172. IEEE, 2007.
- [4] Devendra Singh Chaplot, Emilio Parisotto, and Ruslan Salakhutdinov. Active neural localization. *arXiv preprint arXiv:1801.08214*, 2018.
- [5] F. Dellaert. Square Root SAM: Simultaneous location and mapping via square root information smoothing. In *Robotics: Science and Systems (RSS)*, 2005.
- [6] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012. GTSAM.
- [7] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [8] K. Elimelech and V. Indelman. Introducing pivot: Predictive incremental variable ordering tactic for efficient belief space planning. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, October 2019.
- [9] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.
- [10] 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.

- [11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Intl. Conf. on Machine Learning (ICML)*, 2016.
- [12] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.
- [13] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *Intl. J. of Robotics Research*, 33(1):182–201, 2014.
- [14] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.
- [15] V. Indelman. Cooperative multi-robot belief space planning for autonomous navigation in unknown environments. *Autonomous Robots*, pages 1–21, 2017.
- [16] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015.
- [17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.
- [18] Nathalie Japkowicz, Catherine Myers, Mark Gluck, et al. A novelty detection approach to classification. In *IJCAI*, volume 1, pages 518–523, 1995.
- [19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.
- [20] Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4694–4704, 2017.
- [21] Kapil Katyal, Katie Popek, Chris Paxton, Phil Burlina, and Gregory D Hager. Uncertainty-aware occupancy map prediction using generative networks for robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5453–5459. IEEE, 2019.
- [22] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, 1996.
- [23] A. Kim and R. M. Eustice. Active visual SLAM for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 34(4-5):457–475, 2014.

- [24] D. Kopitkov and V. Indelman. No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, matrix determinant lemma and re-use of calculation. *Intl. J. of Robotics Research*, 36(10):1088–1130, August 2017.
- [25] D. Kopitkov and V. Indelman. General purpose incremental covariance update and efficient belief space planning via factor-graph propagation action tree. *Intl. J. of Robotics Research*, 2019.
- [26] S. Krishna, K. Seo, D. Bhatt, V. Mai, J. K. Murthy, and L. Paull. Deep active localization. *arXiv preprint, arXiv: 1903.01669*, 2019.
- [27] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *Intl. J. of Robotics Research*, 20(5):378–400, 2001.
- [28] Katherine Liu, Kyel Ok, William Vega-Brown, and Nicholas Roy. Deep inference for covariance estimation: Learning gaussian noise models for state estimation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1436–1443. IEEE, 2018.
- [29] A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3565–3572, April 2007.
- [30] Pavel Myshkov and Simon Julier. Posterior distribution analysis for bayesian inference in neural networks. In *Workshop on Bayesian Deep Learning, NIPS*, 2016.
- [31] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.
- [32] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2931–2940, 2019.
- [33] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 28(11-12):1448–1465, 2009.
- [34] Krishan Rana, Ben Talbot, Michael Milford, and Niko Sünderhauf. Residual reactive navigation: Combining classical and learned navigation strategies for deployment in unknown environments. *arXiv preprint arXiv:1909.10972*, 2019.
- [35] Charles Richter and Nicholas Roy. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems (RSS)*, 2017.

- [36] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.
- [37] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, pages 65–72, 2005.
- [38] Gregory J Stein, Christopher Bradley, and Nicholas Roy. Learning over subgoals for efficient navigation of structured, unknown environments. In *Conference on Robot Learning*, pages 213–222, 2018.
- [39] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.

תקציר

niestוט אוטונומי בסביבה לא ידועה מהויה בעיה מאוד מתוגרת ברובוטיקה. במקורה זה הרובוט מתחילה מנוקודה בה אין מידע על הסביבה בה הוא נדרש לננות.niestוט ומיפוי בו-זמןית (SLAM) מהויה את אחת הגישות הפופולריות לאתגר זה. בשימוש ב-SLAM הרובוט מתמודד עם שתי מושגים במקביל, תחילת החש את הסביבה בקרבתו בעזרת חישנים אופטיים כגון: מצלמות או חיישני ליזיר ומיצג את הסביבה באמצעות מפה. שנייה, הרובוט משערך את מיקומו ביחס למפה.

משימה נוספת בניסוט אוטונומי היא קבלת החלטות. הרובוט מייצר לו סט פעולות/מסלולים אפשריים ונדרש לבחור את הפעולה שתוביל לתוצאה הטובה ביותר. קבלת החלטות תחת אי וודאות מוכרת כתכנון למרחב ההסתברותי (Belief Space Planning, BSP), שיטה בה לוקחים בחשבון מדדיות עתידיות אפשריות להערכת התפתחות האי וודאות בכלים הסתברותיים. בשימוש BSP הרובוט מעריך לסט פעולות אפשריות את התפתחות פילוג משתנה המצב (belief) בהינתן על המדדיות והפעולות שbowtzu עד כה. לאחר מכן, בהינתן "האמונה" יחוש פונקציית המחיר שלרוב מתקשרות לאי הוודאות שצפוי להפתח לאורך כל מסלול. לבסוף יבחר הרובוט את המסלול האופטימלי לו פונקציית המחיר יהיה הקטן ביותר.

במקורה בה הפעולות האפשריות יוצאות מאזור שטוח עד כה, ככלומר מובילות לאוצר לא ידוע, קבלת ההחלטה נחפות למשימה מאוד קשה, לאחר שללא מפה, קשה להעריך מה יהיה המדדיות העתידיות. למרות ההתקדמות הרבה בשנים האחרונות, לשיטות הנוכחות של BSP אשר מתמודדות עם ניסוט אוטונומי בסביבות לא ידועות ישנים מגבלות. מגבלה ראשונה היא ששיטות אלה לרוב מניחות שהמפה הלא ידועה נקייה ממכתשיים, שנייה עד כה שיטות BSP לרוב מתחשבות באין הוודאות שצפוי להפתח בעקבות מודל התנועה בלבד ולא מתחשבות במודל המדדיות שצפוי להיות באזוריים לא ידועים.

בניגוד לכך, כשוחשבים על ניסוט של בני-אדם בסביבות לא מוכרות, הם נראה לא משתמשים על החושים בלבד אלא גם על מידע/ניסיון קודם על העולם. ככלומר, אדם יכול להשלים צורות של חללים של אזורים שאינם נמצאים בколо הראשה רק על בסיס מה שכן בколо הראשה ועל בסיס ניסיון מחללים דומים שהיה בהם בעבר. שיטה אחת בה הרובוט יכול ללמוד מניסיון זה למידה عمוקה (Deep Learning, DL).

בעבודה זאת, פיתחנו שיטה BSP מושלבת ניסיון לקבלת החלטות בסביבות לא ידועות. האלגוריתם כולל ניבוי מבוסס ניסיון לפילוג של סביבות לא ידועות בשימוש המפה שטוחה עד זמן נוכחי באמצעות SLAM. לאחר מכן, בשימוש פילוג הסביבות הלא ידועות אנו ממליצים מדדיות עתידיות אפשריות ומשלבים אותן במערכת קבלת החלטות BSP.

עם זאת, ניסיון קודם מפות נבחר איננו בהכרח רלוונטי למשימה הנוכחית, לכן אנו מציעים שיטה לזיהוי חריגות (Novelty Detection). באמצעות היישוב שגיאת השזהר באזור החופף בין המפה המnobעת למפה הנתונה אנו מזהים מצבים בהם המפה הנתונה שונה מאוד ממדונו את הרשות. כאשר מפה נתונה מזוהה כחריגה אנו נמנעים להשתמש באלגוריתם שלנו לקבלת החלטות ומשתמשים במקום ב-BSP בסיסי.

בשלב התוצאות תחילת הצגנו את ביצועי אלגוריתם ניבוי פילוג המפות על סט מבחן וdone בדוגמאות חיוביות ושליליות של תוצאות הניבוי. לבסוף, בחנו את השיטה שלנו לקבלת החלטות בסימולציה Gazebo והשוונו לשיטה BSP אשר לוקחת בחשבון רק את מודל התנועה. התוצאות הראו כי השיטה שלנו מובילה לפחות טעויות בקבלת החלטות ועדיפה גם בשגיאת השערת האי וודאות שתפתח לאורך הפעולות השונות ברוב המוחלט של המקרים. כמו כן, הצגנו מקרה בו הניסוט בסביבה שונה מאוד מסט האימון עליו אומנה הרשות והראנו כי האלגוריתם לזיהוי חריגות זיהה זאת בהצלחה.

המחקר נעשה בהנחיית פרופסור ואדים אינדלמן בפקולטה להנדסת אוירונוטיקה וחלל

תודות

אני מודה לטענין על התמיכה הכספית הנדיבת בהשתלמותי.

ניבוֹי מִבּוֹס נִיסְיוֹן שֶׁל סַבִּיבָה לֹא יִדּוּעָה לְצֹרֶךְ שִׁיפּוֹר תְּכִינּוֹן בָּמֶרֶחֶב הַסְּתְּבָרוֹתִי

חיבור על מחקר

לשם מלאוי חלקי של הדרישות לקבלת התואר מגיסטר למדעים

עומר אסף

הוגש לסנט הטכניון - מכון טכנולוגי לישראל
אב תש"פ, חיפה, אוגוסט 2020

**ניבוּי מִבּוּס נִיסְיוֹן שֶׁל סְבִיבָה לֹא
יַדוּעָה לְצָורֵךְ שִׁיפּוֹר תְּכִנוֹן
בָּמֶרֶחֶב הַסְּתָבָרוֹתִי**

עוֹמְרִי אַסְרָף