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

Omri Asraf and Vadim Indelman

***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. In this paper we develop a data-driven approach for predicting a distribution over an unexplored map, generating future observations, and combining these observations within BSP. We examine our approach and compare it to existing BSP methods in a Gazebo simulation, and demonstrate it often yields improved performance.**

I. 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 [1].

Another task in autonomous navigation is decision making. The agent generates candidate actions (e.g. by PRM, RRT [2], [3]) and it needs to choose which action will lead to the best outcome. One method to address this problem is belief space planning (BSP) [4], [5]. 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 [6]. The DL architectures popular to solve the inpainting problem are generative adversarial network (GAN) [7] and variational autoencoders (VAE) [8].

In this work we develop an approach to incorporate relevant prior experience to predict a distribution over unexplored environments within belief space planning framework. Our data-driven approach approximately predicts a distribution over unexplored areas along with 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.

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

The authors are with the Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel, asomri@campus.technion.ac.il, vadim.indelman@technion.ac.il.

problem [9]–[11], assuming environment is known. More closely related to us are learning approaches that consider autonomous navigation in unknown environments [12], [13]. 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. [14] added a neural network (NN) to visual navigation for predicting future collisions in an unknown environment, and Katyal et al. [15] 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.

Contributions: Our contribution is the improvement of existing BSP approaches by the incorporation of relevant aspects from experience. 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.

II. Notations and Problem Formulation

A. 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. We denote by M_k the map observed by time k, any 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 $y_{1:k} \doteq \{y_1, .., y_k\}$ and $a_{0:k-1} \doteq \{a_0, ..., a_{k-1}\}$ 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

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

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, point clouds) thus depends both on the robot state and the environment, i.e.

$$y_i = g(x_i, m_i) + u_i, \qquad u_i \sim \mathcal{N}(0, \Sigma_u). \tag{2}$$

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). The observation model of a relative-pose measurement is (see e.g. [16], [17])

$$y_{ij}^{rel}(y_i, y_j) = h(x_i, x_j) + v_{ij}, \quad v_{ij} \sim \mathcal{N}(0, \Sigma_v(y_i, y_j)). \tag{3}$$

Note that the measurements y_{ij}^{rel} depend on the raw measurements y_i and y_j, and therefore also on the map (see Eq. (2)). 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 [16], [17].

In the current work we consider a pose-SLAM framework, hence the conditional probability density function (pdf) over the 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 history. Using Bayes rule and standard assumptions it can be rewritten as

$$b_k = \eta \mathbb{P}(x_0) \prod_{i=1}^{k} [\mathbb{P}(x_i|x_{i-1}, a_{i-1}) \prod_{j} \mathbb{P}(y_{ij}^{rel}|x_i, x_j)], \tag{4}$$

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 (1) and measurement likelihood (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 [18], 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 $\mathbb{P}(M_k|H_k) = \int \mathbb{P}(M_k|x_{1:k}, H_k)\mathbb{P}(x_{1:k}|H_k)dx_{1:k}$. 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:

$$\mathbb{P}(M_k|H_k) \approx \mathbb{P}(M_k|\hat{x}_{1:k}, y_{1:k}). \tag{5}$$

B. 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 given by.

$$J(b_k, a_{k:k+L-1}) = \sum_{l=1}^{L} \mathop{\mathbb{E}}_{y_{k+1:k+l}} \{c(b_{k+l}, a_{k+l-1})\}, \tag{6}$$

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

$$b_{k+l} \doteq \mathbb{P}(x_{1:k+l} \mid H_k, a_{k:k+l-1}, y_{k+1:k+l}), \tag{7}$$

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 optimal (non-myopic) action is defined as

$$a_{k:k+L-1}^{*} = \mathop{\arg\min}_{a_{k:k+L-1}} J(b_k, a_{k:k+L-1}). \tag{8}$$

To calculate optimal action, one has to evaluate Eq. (6) for each candidate action. To do so, we should be able to reliably predict future observations, and appropriately approximate the expectation in (6), typically via sampling.

C. 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), 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 incorporating experience within BSP, aiming to improve prediction of future observations as part of the expectation calculations in Eq. (6).

III. Approach

A. Incorporating Experience within BSP

For simplicity, in this section we present formulation for a myopic setting, and extend later the discussion to the more general, non-myopic case. Denoting the available experience by D and $H^-_{k+1} \doteq \{H_k, a_k\}$, we re-write Eq. (6) as

$$J(b_k, a_k) = \int \mathbb{P}(y_{k+1}|H^-_{k+1}, D)c(b_{k+1}, a_k)dy_{k+1}. \quad (9)$$

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

$$J(b_k, a_k) \approx \frac{1}{N} \sum_{y_{k+1}\sim\mathbb{P}(y_{k+1}|H_k,a_k,D)} c(b_{k+1}, a_k). \quad (10)$$

To sample future observations y_{k+1}, we first marginalize over the robot state x_{k+1}, and recalling the generative model (2), also over the corresponding sub-map/scene m_{k+1}. Applying chain rule and recalling Markov assumption yields

$$\mathbb{P}(y_{k+1}|H^-_{k+1}, D) = \int_{x_{k+1},m_{k+1}} \mathbb{P}(y_{k+l} \mid x_{k+1}, m_{k+1}) \cdot$$
$$\mathbb{P}(x_{k+1} \mid m_{k+1}, H^-_{k+1}, D)\mathbb{P}(m_{k+1} \mid H^-_{k+1}, D)dm_{k+1}dx_{k+1}.$$

Recalling 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. $\mathbb{P}(x_{k+1} \mid m_{k+1}, H^-_{k+1}, D) = \mathbb{P}(x_{k+1} \mid H^-_{k+1})$, which can be calculated by propagating the belief from the previous time instant and marginalization: $\mathbb{P}(x_{k+1} \mid H_k, a_k) = \int_{x_{1:k}} \mathbb{P}(x_{1:k} \mid H_k)\mathbb{P}(x_{k+1} \mid x_k, a_k)dx_{1:k}$. We note as the considered pdfs are Gaussian, this marginalziation can be performed analytically and computationally efficiently, yielding $\mathbb{P}(x_{k+1} \mid H_k, a_k) = N(\hat{x}^-_{k+1}, \Sigma^-_{k+1})$.

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

$$\int_{m_{k+1}} \mathbb{P}(y_{k+1}|\hat{x}^-_{k+1}, m_{k+1})\mathbb{P}(m_{k+1}|H^-_{k+1}, D)dm_{k+1}. \quad (11)$$

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.

B. 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,

$$\mathbb{P}(m_{k+1}|H^-_{k+1}, D) = \int_{M_k} \mathbb{P}(m_{k+1}|M_k, a_k, D)\mathbb{P}(M_k|H_k)dM_k,$$

where the posterior over the map, $\mathbb{P}(M_k|H_k)$, is already available (see Eq. (5)).

Further, we approximate the above equation with a single sample from $\mathbb{P}(M_k|H_k)$, the maximum likelihood estimate $\hat{M}_k$ (see line 9 in Algorithm 1). With this approximation the future map distribution becomes $\mathbb{P}(m_{k+1}|H_k, a_k, D) \approx \mathbb{P}(m_{k+1}|\hat{M}_k, a_k, D)$. 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.

$$\mathbb{P}(m_{k+1} \mid \hat{M}_k, a_k, D) \approx \mathbb{P}(m_{k+1} \mid \hat{m}_k, a_k, D). \quad (12)$$

As seen from (12), 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 $\{\hat{m}_k, a_k\}$.

C. Offline Training Phase

Given offline available environment maps $\mathcal{M}_D \doteq \{M^i\}$ and action space $\mathcal{A} \doteq \{a\}$, and recalling Eq. (12), we define experience as

$$D \doteq \{(m, a, m')|m, m' \in M^i, \forall M^i \in \mathcal{M}_D, a \in \mathcal{A}\}, \quad (13)$$

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 CVAE to approximate the distribution $\mathbb{P}(m'|C)$. We now briefly present this formulation for self-containment (see, e.g., [19]). First, we marginalize over a latent variable z and re-write $\mathbb{P}(m'|C)$ as

$$\mathbb{P}(m'|C) = \int_z \mathbb{P}(m'|z, C)\mathbb{P}(z|C)dz. \tag{14}$$

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 $f(.; \theta)$ is a deterministic function parametrized by θ and σ is a typically small fixed hyperparameter. The integral (14) 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 $\mu(.)$ and $\Sigma(.)$ are deterministic functions parameterized by ϕ. Based on the definition of Kullback-Leiber divergence (KL) between $\mathbb{Q}_z \doteq \mathbb{Q}(z \mid m', C; \phi)$ and $\mathbb{P}(z \mid m', C)$ (see more details in [8]), we get the main equation of VAE:

$$\log \mathbb{P}(m' \mid C) - \mathtt{KL}[\mathbb{Q}(z \mid m', C; \phi) \mid\mid \mathbb{P}(z \mid m', C)] =$$
$$\mathop{\mathbb{E}}_{z\sim\mathbb{Q}_z}\{\log \mathbb{P}(m' \mid z, C; \theta)\} - \mathtt{KL}[\mathbb{Q}(z \mid m', C; \phi) \mid\mid \mathbb{P}(z \mid C)]$$

In order to minimize $\mathtt{KL}[\mathbb{Q}(z \mid m', C) \mid\mid \mathbb{P}(z \mid m', C)]$, since it is non-negative by definition and $\log \mathbb{P}(m' \mid C)$ 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 $z \sim \mathbb{Q}(z \mid m', C; \phi)$ is done only by a single sample. Also, to make the above equation tractable via backpropagation, the re-parameterization trick is used [8]. 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

$$\begin{aligned} l(\theta, \phi; m', C) &\doteq \|m' - f(z, C; \theta)\|^2 \\ &+ \mathtt{KL}[\mathcal{N}(\mu(m', C; \phi), \Sigma(m', C; \phi)) \mid\mid \mathcal{N}(0, I)]. \end{aligned} \tag{15}$$

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 $\mathbb{P}(z \mid C)$ is a normal distribution $\mathcal{N}(0, I)$ to allow easy sampling in the online stage. The loss (15), considering the entire dataset D, can be written as $Loss(\theta, \phi; D) = \sum_{m',C\sim D} l(\theta, \phi; m', C)$. Finally, the encoder and decoder weights θ and ϕ, are optimized as $\phi^*, \theta^* = \arg\min_{\phi,\theta} Loss(\theta, \phi; D)$. In practice, this optimization is done via standard stochastic gradient methods, i.e over mini-batches which are subsets of D.

D. Online Deployment

Having described the offline learning of the conditional distribution $\mathbb{P}(m_{k+1} \mid C, D)$, we now focus on the deployment stage, considering a planning session at time instant k. Recalling Eq. (12), and the sub-map estimate $\hat{m}_k$, we resort to a sampling-based approximate representation of the 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 $\mathbb{P}(m_{k+1} \mid \hat{m}_k, a_k, D) = \int_z \mathbb{P}(m_{k+1} \mid z, \hat{m}_k, a_k, D)\mathbb{P}(z \mid \hat{m}_k, a_k, D)dz$, which can be approximated via sampling as $\mathbb{P}(m_{k+1} \mid \hat{m}_k, a_k, D) \approx \frac{1}{n_z}\sum_{z\sim\mathbb{P}_z} \mathbb{P}(m_{k+1} \mid z, \hat{m}_k, a_k, D)$, where n_z is the number of samples, and $\mathbb{P}_z \doteq \mathbb{P}(z \mid \hat{m}_k, a_k, D)$.

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

$$\frac{1}{n_z}\sum_{z\sim\mathcal{N}(0,I)} \mathcal{N}(f(z, \hat{m}_k, a_k; \theta^*), \sigma^2 * I), \tag{16}$$

where, as standard in VAE, we consider $\mathbb{P}(z \mid \hat{m}_k, a_k, D) = \mathbb{P}(z) = \mathcal{N}(0, I)$. As the decoder is learned, i.e. function $f(; \theta^*)$ is deterministic at this point, sampling z corresponds to choosing one of the components in the GMM representation (16), from which we can easily sample a realization of the future sub-map i.e. $m_{k+1} \sim \mathcal{N}(f(z, \hat{m}_k, a_k; \theta^*), \sigma^2 * I)$.

The obtained predicted maps m_{k+1} are then used for generating future *raw* observations y_{k+1}, see Eq. (11). 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. (10), 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 1.

E. Novelty Detection

Clearly, the environments considered in the offline training should be representative and similar to the environment

Algorithm 1 BSP with Experience-Based Prediction

Inputs:
b_k: state belief at current time
$\mathbb{P}(M_k|H_k)$: map belief at current time
$a_{k:k+L-1}$: a candidate L look-ahead steps action sequence
$f(.;\theta^*)$: trained decoder
Outputs:
$J(b_k, a_{k:k+L-1})$: computed objective function for a given action sequence

$\hat{M}_k \Leftarrow \mathbb{P}(M_k|H_k)$ ▷ Get maximum likelihood estimate of map belief
$\hat{m}_k \subseteq \hat{M}_k$ ▷ Get current sub-map estimate from $\hat{M}_k$ (Eq. (12))
for $i = 1 : N$ **do**
  $m_k^i = \hat{m}_k$
  for $j = 1 : L$ **do**
    ▷ Get ML estimate without future observations
    $\hat{x}^-_{k+j} \Leftarrow \mathbb{P}(x_{k+j}|H_k, a_{k:k+j-1})$
    $z^i \sim \mathcal{N}(0, I)$
    ▷ Predict sub-map (Eq. (16))
    $m^i_{k+j} \sim \mathcal{N}(f(z^i, m^i_{k+j-1}, a_{k+j-1}; \theta^*), \sigma^2 * I)$
    ▷ Generate future observation (Eq. (11))
    $y^i_{k+j} \sim \mathbb{P}(y_{k+j} \mid m^i_{k+j}, \hat{x}^-_{k+j})$
    Calculate b^i_{k+j} ▷ Calculate future belief using y^i_{k+j} (Eq. (7))
    Calculate cost/reward $c(b^i_{k+j})$
    ▷ Accumulate costs
    $J(b_k, a_{k:k+L-1}) = J(b_k, a_{k:k+L-1}) + c(b^i_{k+j})$
  end for
end for
▷ Normalize to get empirical expectation
$J(b_k, a_{k:k+L-1}) = \frac{1}{N} J(b_k, a_{k:k+L-1})$
return $J(b_k, a_{k:k+L-1})$

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 $M^{\texttt{online}}$ is assumed to be of a similar nature, i.e. $M^{\texttt{online}} \sim \mathbb{P}(M)$. Note that this assumption does *not* imply $M^{\texttt{online}} \in \mathcal{M}_D$. 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, $M \doteq M^{\texttt{online}}$ 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. $\mathbb{P}(M_k \mid H_k)$. 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. [20]–[22]) 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. [23] and used recently in the navigation domain by Richter et al. [14]. 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 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 [14].

Noting the above approach requires training separate NNs for map prediction and novelty detection, we suggest a simple method that does 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.

IV. 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.

A. Map Distribution Prediction With CVAE

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

The dataset that was used is KTH dataset [24] 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'). The actions between the sub-maps were defined by one-hot vector that represents the four possible stride directions (up, down, left, right).

We examined the prediction performance on testset maps that were not part of the training process. Fig. 1 demonstrates three examples of prediction results (For statistical results see [25]). For each prediction we calculate the reconstruction error (RE) on the overlap area by novelty detection method that was presented in Section III-E, and additionally the prediction error (PE) on the unknown area against the ground truth. Examples in (I) show successful prediction results even in the 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. 1c.I, and does not close into a room as in Fig. 1c.II, the NN predictions will mostly present the solution that appeared a greater amount of times in the training dataset. Yet, our results are probabilistic

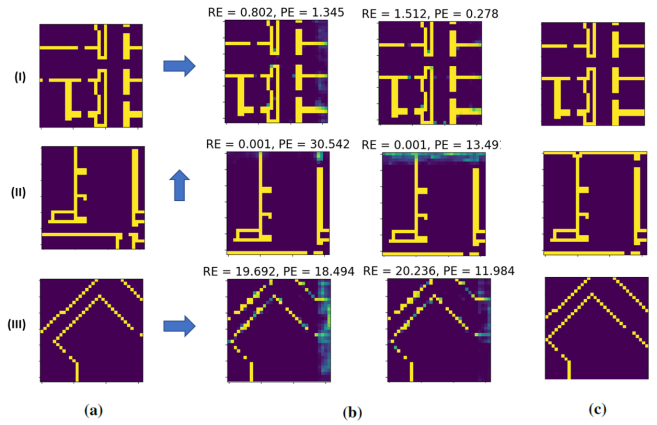


Fig. 1: Examples of the prediction algorithm. (a) Inputs – sub-map and action; (b) Outputs – two samples of prediction with reconstruction error (RE) and prediction error (PE). On the left worst prediction result and on the right best prediction result; (c) Ground truth. See statistical results in [25].

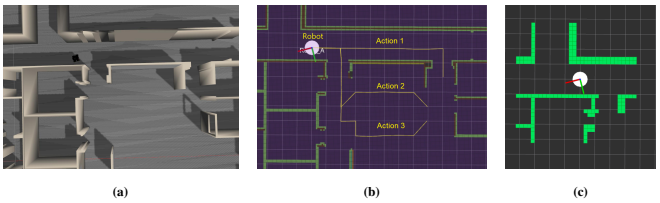


Fig. 2: (a) 3D Gazebo simulation world; (b) Planning session with three candidate actions; (c) Occupancy grid of the partial map that was used for prediction.

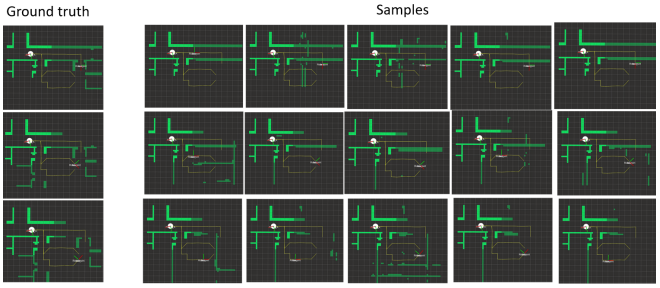


Fig. 3: Map prediction results for three actions from Figure 2b: action 1 - top row; action 2 - middle row; action 3 - bottom row. Ground truth maps in relevant regions are shown in left column. On the right we show five samples of map predictions for each of the actions. Conditional map in light green, predicted map in dark green.

and therefore the non-trivial cases (right sample in example II) are still represented. Example (III) shows 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 not at all cases of such walls.

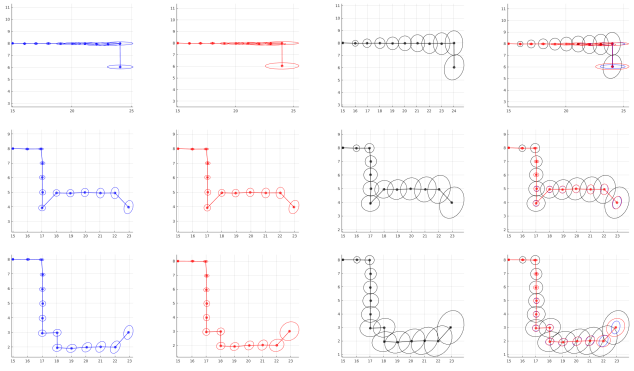
These examples represent three families of experience-based prediction results: (I) most predictions are correct (low PE); (II) most predictions are wrong because of uncommon ground truth map (high PE and low RE); (III) most predictions are wrong because of an unfamiliar input (high PE and high RE). Using the novelty detection method from Section III-E 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. Yet, statistically we assume that in most cases the online map will be of a similar nature as the dataset maps (as discussed in Section III-E), and therefore, most cases will be from the first family above and, thus, improve the decision making process.

B. 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. 2a) 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 [16]. 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 (unmatching two scans) the odometry will be used instead. Our pose-SLAM implementation uses GTSAM [26] and the mapping process uses OctoMap [27].

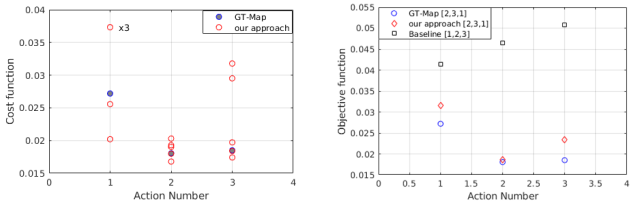
In the planning stage the robot starts from a defined point and gets a set of actions randomly by PRM method or manually by the user. The robot's mission is to choose the best action by calculating the objective function for each action. The main question in this work is how we can do this calculation 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 1, and used the DL-based prediction from section IV-A, 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 [16] we used two laser scans to get the measurement likelihood model, i.e the measurement



(a) GT-map (b) Our approach (c) Baseline (d) Comparison

Fig. 4: Comparison between three methods of uncertainty evolution on three different actions from Figure 2b: 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 3 for each action: action 1 - first sample in row 1, action 2 - fifth sample in row 2, action 3 - forth sample in row 3. Results for all samples are summarized in Figure 5a. For convenient visualization covariance resolution was multiplied by 100.



(a) (b)

Fig. 5: (a) Cost function results of all the samples from Fig. 3 compared to cost function calculated with the `GT-map` method. (b) Objective function (17) values for the three methods. In contrast to `baseline` BSP, our approach preserves action-ordering with respect to `GT-map`.

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 $\sqrt{Trace(\Sigma_{k+L})}$. For objective function calculation we average the cost function of all samples, i.e

$$J(b_k, a_{k:k+L-1}) \doteq \frac{1}{N}\sum_{i=1}^{N}\sqrt{Trace(\Sigma^i_{k+L})}. \qquad (17)$$

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 planing mission when there are three candidate actions (see Fig. 2b) and a partial map presented by an occupancy grid with equal scale of the dataset (Fig. 2c). 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. 3 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). The first row shows five predictions for action 1; all samples predicted a long corridor similar to the ground truth. On the other hand neither of the samples predicted the opening on the right. The second and third 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 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 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 th `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. 5a 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 samples 1,2,3 in third row of Fig. 3) were close and two other predicted higher uncertainty compared to the `GT-map` cost (see samples 4,5 in third row of Fig. 3). The density of the samples could be used as an indication of the prediction confidence.

Fig. 5b shows the calculated objective function (17) 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 I. 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 (for more details see [25]). 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^*

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

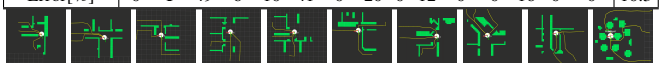


TABLE I: Performance of our and `baseline` approaches in 15 different scenarios. Each scenario includes a different environment and three actions. Several examples of planning settings are shown at the bottom. The table reports for each method the number of action ordering mistakes with respect to BSP with ground truth map (`GT-map`), and the uncertainty cost error. We also show the reconstruction error (RE) calculated by our novelty detection approach. See further details in [25].

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.

V. Conclusions

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.

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. Furthermore, we suggested a novelty detection method to avoid using unfamiliar inputs in the prediction. Future work may extend this 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.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Simultaneous localization and mapping: Present, future, and the robust-perception age," *IEEE Trans. Robotics*, vol. 32, no. 6, pp. 1309 – 1332, 2016.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, 1996.

[3] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Intl. J. of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[4] 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*, vol. 31, no. 11, pp. 1263–1278, 2012.

[5] 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*, vol. 34, no. 7, pp. 849–882, 2015.

[6] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[8] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[9] P. Karkus, D. Hsu, and W. S. Lee, "Qmdp-net: Deep learning for planning under partial observability," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 4694–4704.

[10] D. S. Chaplot, E. Parisotto, and R. Salakhutdinov, "Active neural localization," *arXiv preprint arXiv:1801.08214*, 2018.

[11] S. Krishna, K. Seo, D. Bhatt, V. Mai, J. K. Murthy, and L. Paull, "Deep active localization," *arXiv preprint, arXiv: 1903.01669*, 2019.

[12] G. J. Stein, C. Bradley, and N. Roy, "Learning over subgoals for efficient navigation of structured, unknown environments," in *Conference on Robot Learning*, 2018, pp. 213–222.

[13] K. Rana, B. Talbot, M. Milford, and N. Sünderhauf, "Residual reactive navigation: Combining classical and learned navigation strategies for deployment in unknown environments," *arXiv preprint arXiv:1909.10972*, 2019.

[14] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems (RSS)*, 2017.

[15] K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager, "Uncertainty-aware occupancy map prediction using generative networks for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5453–5459.

[16] A. Censi, "An accurate closed-form estimate of icp's covariance," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 3167–3172.

[17] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning gaussian noise models for state estimation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1436–1443.

[18] 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*, vol. 31, pp. 217–236, Feb 2012.

[19] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in neural information processing systems*, 2015, pp. 3483–3491.

[20] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Intl. Conf. on Machine Learning (ICML)*, 2016.

[21] P. Myshkov and S. Julier, "Posterior distribution analysis for bayesian inference in neural networks," in *Workshop on Bayesian Deep Learning, NIPS*, 2016.

[22] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, "Sampling-free epistemic uncertainty estimation using approximated variance propagation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2931–2940.

[23] N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *IJCAI*, vol. 1, 1995, pp. 518–523.

[24] A. Aydemir, P. Jensfelt, and J. 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*. IEEE, 2012, pp. 4675–4682.

[25] O. Asraf and V. Indelman, "Experience-based prediction of unknown environments for enhanced belief space planning - supplementary material," Technion - Israel Institute of Technology, Tech. Rep., 2020. [Online]. Available: https://indelman.github.io/ANPL-Website/Publications/Asraf20iros_supplementary.pdf

[26] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012, gTSAM.

[27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.