Localization from Semantic Observations via the Matrix Permanent

Nikolay Atanasov*  Menglong Zhu  Kostas Daniilidis  George J. Pappas

GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA

Abstract
Most approaches to robot localization rely on low-level geometric features such as points, lines, and planes. In this paper, we use object recognition to obtain semantic information from the robot’s sensors and consider the task of localizing the robot within a prior map of landmarks, which are annotated with semantic labels. As object recognition algorithms miss detections and produce false alarms, correct data association between the detections and the landmarks on the map is central to the semantic localization problem. Instead of the traditional vector-based representation, we propose a sensor model, which encodes the semantic observations via random finite sets and enables a unified treatment of missed detections, false alarms, and data association. Our second contribution is to reduce the problem of computing the likelihood of a set-valued observation to the problem of computing a matrix permanent. It is this crucial transformation that allows us to solve the semantic localization problem with a polynomial-time approximation to the set-based Bayes filter. Finally, we address the active semantic localization problem, in which the observer’s trajectory is planned in order to improve the accuracy and efficiency of the localization process. The performance of our approach is demonstrated in simulation and in real environments using deformable-part-model-based object detectors. Robust global localization from semantic observations is demonstrated for a mobile robot, for the Project Tango phone, and on the KITTI visual odometry dataset. Comparisons are made with the traditional lidar-based geometric Monte-Carlo localization.

Keywords
active semantic localization; monte-carlo localization; mobile robot localization; matrix permanent; random finite set; particle filter; conditional entropy; object recognition; deformable part model; Project Tango

1. Introduction
Localization, the problem of estimating the pose of a mobile robot from sensor data given a prior map, is fundamental in the field of robotics. Reliable navigation, object manipulation, mapping, and many other tasks require accurate knowledge of the robot’s pose. Most existing approaches to localization and the related simultaneous localization and mapping (SLAM) rely on low-level geometric features such as points, lines, and planes. In contrast, we propose to use the recent advances in object recognition to obtain semantic information from the robot’s sensors and localize the robot within a prior map.
of landmarks, which are annotated with semantic labels. Our approach is not meant to replace, but rather enhance, the existing localization and SLAM solutions. It offers several benefits. Localizing against semantically-meaningful landmarks is less ambiguous and can be used for global localization\(^1\) and loop-closure. Also, high-precision sensors such as laser rangefinders and 3-D lidars are not crucial for accurate localization and can be replaced by regular cameras. Finally, semantically-annotated maps can be constructed for GPS-denied environments via the mapping approaches that received significant attention in recent years (Civera et al. 2011, Galindo et al. 2005, Kostavelis & Gasteratos 2013, Nüchter & Hertzb erg 2008, Pronobis 2011).

A preliminary version of this paper appeared at the 2014 Robotics: Science and Systems Conference (Atanasov et al. 2014). This version extends and clarifies the theoretical results regarding semantic localization, addresses the active semantic localization problem, and provides additional real-world experiments, which demonstrate global localization for the Project Tango phone (Google ATAP group 2014) and on the KITTI visual odometry dataset (Geiger et al. 2013).

Related Work. Monte-Carlo localization based on geometric features was proposed by Dellaert et al. (1999). The knowledge about the robot pose is represented by a weighted set of samples (particles) and is updated over time as the robot moves and senses the environment. This and other traditional localization methods use vectors to represent the map and the sensor measurements. Bayesian filtering in the resulting vector space relies on the assumption that the data association, i.e., the correspondence between the sensor observations and the features on the map, is known. While this might not be an issue for scan matching in occupancy-grid maps, the assumption is violated for landmark-based maps. Existing landmark-based localization and SLAM techniques require external solutions to the problems of data association and clutter rejection (Bailey 2002, Montemerlo & Thrun 2003). Moreover, state-of-the-art approaches nowadays are based on factor graphs (Kaess et al. 2008, Kummerle et al. 2011) and rely heavily on continuous Gaussian random variables. The introduction of semantic labels for the landmarks is not addressed in existing work and requires handling discrete (non-Gaussian) variables in the estimation.

There is a line of work addressing visual localization, which matches observed image features to an image database, whose images correspond to the nodes of a topological map (Angeli et al. 2009, Košćeká & Li 2004, Mariottini & Roumeliotis 2011, Se et al. 2005, Wang et al. 2006, Wolf et al. 2005). Wang et al. (2006) represent each location in the topological map by a set of interest points that can be reliably detected in images and use nearest neighbor search to match observed SIFT features to the database. Košćeká & Li (2004) also characterize scale-invariant key points by the SIFT descriptor and find nodes in the topological map, whose features match the observed ones the best. The drawback of this most likely data association approach is that when it is wrong it quickly causes the estimation procedure to diverge. Hesch et al. (2013) study the effects of unobservable directions on the estimator consistency in vision-aided inertial navigation systems. In the SLAM context, bad data association can be mitigated by a two-stage approach, in which the back-end (e.g. factor graph) optimizer is allowed to reject or alter associations proposed by the front-end (e.g. appearance-based place recognition) ( Sünderhauf & Protzel 2011). As object recognition algorithms miss detections and produce false alarms, correct data association is crucial for semantic localization and semantic world modeling too (Wong et al. 2013).

Instead of the traditional vector-based representation, we use random finite sets to model the semantic information obtained from object recognition. This allows us to explicitly incorporate missed detections, false alarms, and data association in the sensor model. In recent years, random-finite-set-based solutions to SLAM have gained popularity due to their unified treatment of filtering and data association. Mahler (2007) derived the Bayesian recursion with random-finite-set-valued observations and proposed a first-moment approximation, called the probability hypothesis density (PHD) filter. The PHD filter has been successfully applied to SLAM by Kalyan et al. (2010), Lee et al. (2013), and Mullane et al. (2011). In these works, the vehicle trajectory is tracked by a particle filter and the first moment of a trajectory-conditioned map for each particle is propagated via a Gaussian-mixture PHD filter. Bishop & Jensfelt (2010) address global geometric

\(^1\) Visual-odometry and SLAM techniques typically do not use an absolute reference to the world and do not provide global localization. The initial robot pose is chosen as the map origin and is simply tracked over time.
Atanasov et al.

localization by formulating hypotheses about the robot state and tracking them with the PHD filter. Zhang et al. (2012) propose an approach for visual odometry using a PHD filter to track SIFT features extracted from observed images. Most of the random-set approaches rely on a first-moment approximation via the PHD filter. Only few deal with the full observation model (Dames et al. 2013, Ma et al. 2006, Sidenbladh & Wirkander 2003) and none have applied the model in a semantic setting or studied its computational complexity.

There are several semantic localization approaches that do not rely on a random-finite-set model. Anati et al. (2012) match histogram-of-gradient-energies and quantized-colors features to the expected features from a prior semantic map. Yi et al. (2009) and Ko et al. (2013) use semantic descriptions of distance and bearing in a contextual map for active semantic localization. Bao & Savarese (2011) propose a maximum-likelihood-estimation formulation for semantic structure from motion. In addition to recovering camera parameters (motion) and 3-D locations of image features (structure), the authors recover the 3-D locations, orientations, and categories of objects in the scene. A Markov-chain-Monte-Carlo algorithm is used to solve a batch estimation problem by sampling from the data likelihood of the observed measurements.

Summary of contributions.

• We represent the semantic information obtained from object recognition with random finite sets. This allows a unified treatment of missed detections, false alarms, and data association in the sensor model.

• We prove that obtaining the likelihood of a set-valued observation is equivalent to a matrix permanent computation. It is this crucial transformation that enables an efficient polynomial-time approximation to Bayesian filtering with set-valued observations.

• We address the active semantic localization problem, in which the observer’s trajectory is planned in order to minimize the entropy of the pose distribution, conditioned on the future measurements.

Connections between the matrix permanent and data association have been identified in the target tracking community (Collins & Uhlmann (1992), Morelande (2009), Oh et al. (2009), Pasula et al. (1999), Liggins et al. (2008, Ch.11)) but this is the first connection with the random-finite-set observation model.

Paper Organization. In Sec. 2 we formulate the semantic localization problem precisely. In Sec. 3 we provide a probabilistic model, which quantifies the likelihood of a random finite set of semantic observations and captures false positives, missed detections, and unknown data association. The key relationship between filtering with the random-finite-set observation model and the matrix permanent is derived in Sec. 4. In Sec. 5, we introduce the active semantic localization problem and discuss the efficient minimization of the observer’s pose entropy conditioned on the future semantic measurements. Finally, in Sec. 6, we present results from simulations and real-world experiments and discuss the performance of our approach.

2. Problem Formulation

Consider a mobile robot, whose dynamics are governed by the motion model \( x_{t+1} = f(x_t, u_t, v_t) \), where \( x_t := (x_{tp}^t, x_{tr}^t, x_{ta}^t) \) is the robot state, containing its position \( x_{tp}^t \), orientation \( x_{tr}^t \), and other variables \( x_{ta}^t \), such as velocity and acceleration, \( u_t \) is the control input, and \( v_t \) is the motion noise. Alternatively, the model can be specified by the probability density function (pdf) of \( x_{t+1} \) conditioned on \( x_t \) and \( u_t \):

\[
p_f(\cdot \mid x_t, u_t).
\]  

The robot has access to a semantic map of the environment, containing \( n \) objects with known poses and classes. Let the set \( Y = \{y_1, \ldots, y_L\} \) represent the map, where \( y_i := (y_{pi}^i, y_{ri}^i, y_{ci}^i) \) consists of the position \( y_{pi}^i \), orientation \( y_{ri}^i \), and class \( y_{ci}^i \) of the \( i \)-th object. Depending on the application, the object state \( y_i \) may capture other observable properties of interest, such as shape priors (Dame et al. 2013).

At each time \( t \), the robot receives data from its sensors and runs an object recognition algorithm, capable of detecting instances from the set of object classes \( C \), present in \( Y \). If some object \( y \in Y \) is visible and detected from the current
Fig. 1. A mobile robot (left) localizes itself within a semantic map of the environment by detecting chairs and doors in images (top middle), obtained from its surroundings. A semantic observation received by the robot (top right) consists of a detected class, a detection (confidence) score, and a bearing angle to the detected bounding box. Due to the fact that object recognition misses detections (only one of the two visible chairs is detected) and produces false positives (there is an incorrect door detection), it is appropriate to model the collection of semantic observations via a set with randomly-varying cardinality. Finally, correct data association between the object detections (top right) and the landmarks on the prior map (bottom right) plays a key role in the robot’s ability to estimate its location.

robot pose $x_t$, then a semantic measurement $z_t$ is obtained. In the remainder, we assume that a semantic measurement, $z_t := (c_t, s_t, b_t)$, consists of a detected class $c_t \in C$, a detection score $s_t \in S$, and an estimate $b_t \in B$ of the bearing from the sensor to the detected object, where $S$ is the range of possible scores and $B$ is the range of bearings, usually specified by the sensor’s field of view (e.g. a camera with $B = [-47^\circ, 47^\circ]$ was used in our experiments). Depending on the sensors and the visual processing, $z_t$ could also contain bounding box, range, color, or other information about the detected object. Detections might also be generated by clutter, which includes the background and any objects not captured on the map $Y$. Fig. 1 illustrates the object recognition process and the challenges associated with it. Due to false alarms and missed detections, a randomly-sized collection of measurements is received at time $t$. Instead of the traditional vector representation, it is more appropriate to model the collection of semantic observations via a random finite set $Z_t$. For any $t$, denote the pdf of robot state $x_t$ conditioned on the map $Y$, the past semantic observations $Z_{0:t}$, and the control history $u_{0:t-1}$ by $p_t|t$ and that of $x_{t+1}$ | $Y$, $Z_{0:t}$, $u_{0:t}$ by $p_{t+1}|t$.

**Problem 1** (Semantic Localization). Suppose that control $u_t$ is applied at time $t \geq 0$ and, after moving, the robot obtains a random finite set $Z_{t+1}$ of semantic observations. Given a prior pdf $p_{t|t}$ and the semantic map $Y$, compute the posterior pdf $p_{t+1|t+1}$ which takes $Z_{t+1}$ and $u_t$ into account.

3. Semantic Observation Model

It is natural to approach the semantic localization problem via recursive Bayesian estimation. This, however, requires a probabilistic model, which quantifies the likelihood of a random set $Z_{t+1}$ of semantic observations, conditioned on the set of objects $Y$ and the robot state $x_{t+1}$. Really, the first challenge of the semantic localization problem is a modeling one. In Sec. 3.1, we model the likelihood of an observation received from a single object in the environment. Then, in Sec. 3.2, we combine the single-object observation models into an observation model for multiple objects, which captures data association, missed detections, and false alarms.

---

2 See Mahler (2007, Appendix F) for a formal definition of a random finite set.
3.1. Observation Model for a Single Object

The probabilistic model of a semantic observation obtained from a single object consists of three ingredients: a detection model, an observation likelihood, and a clutter model.

The detection model quantifies the probability of detecting an object \( y \in Y \) from a given robot state \( x \). Let \( \beta(x, y) \) be the true bearing angle from the robot’s sensor to the object \( y \) in the sensor frame\(^3\). Let the field of view of the sensor\(^4\) be described by the set \( \text{FoV}(x) \). Objects outside the field of view cannot be detected. For the ones within, we use a distance-decaying probability of detection:

\[
p_d(y, x) := \begin{cases} 
    p_0 \exp\left\{ -\frac{|m_0 - \|y^p - x^p\|_2|}{v_0} \right\} & \text{if } y^p \in \text{FoV}(x), \\
    0 & \text{else},
\end{cases}
\]

where \( p_0, m_0, v_0 \) are constants specifying the dependence of the detection probability on distance and are typically learned from training data. The constants might depend on the object’s class \( y^c \) but this is not explicit to simplify notation. A more complex model which depends on the relative orientation between \( x \) and \( y \) or uses a different function of the distance is also possible. Fig. 2 illustrates the detection model. If visibility information is available from the prior map, it should also be considered when calculating the probability of detection.

When an object \( y \in Y \) is detected, the probability of the resulting measurement \( z = (c, s, b) \) is quantified by the observation likelihood. Assuming that conditioned on the true object state \( y \), the bearing measurement \( b \) is independent of the class \( c \) and score \( s \) measurements, it is appropriate to model its conditional pdf \( p_\beta(b \mid y, x) \) as that of a truncated Gaussian distribution over the bearing range \( B \) with mean \( \beta(x, y) \) and covariance \( \Sigma_\beta \). The covariance can be learned from training data and can be class dependent. Since object recognition algorithms aim to be scale- and orientation-invariant, we can also assume that the class and score measurements are independent of the robot state \( x \). Then, the observation likelihood of a semantic measurement \( z \) can be decomposed as:

\[
p_z(z \mid y, x) := p_c(c \mid y^c)p_s(s \mid c, y^c)p_\beta(b \mid y, x),
\]

where \( p_c(c \mid y^c) \) is the confusion matrix of the object detector and \( p_s(s \mid c, y^c) \) is a score likelihood. The latter can be learned, for example, by recording the scores from the detected positive examples in a training set and using kernel density estimation (see Fig. 16). A more complicated generative model can be used to approximate the observation likelihood \( p_z \).

---

\(^3\) For example, in 2-D, assuming the robot and the sensor frames coincide, \( \beta(x, y) := |\tan^{-1}((x^p(2) - y^p(2)) / (x^p(1) - y^p(1))) - \pi| \).

\(^4\) The field of view of a camera in 2-D, assuming its frame coincides with the robot’s, can be represented by \( \left\{ w \in \mathbb{R}^2 \mid \|x^p - w\|_2 \leq r_d, \beta(x, w) \leq \alpha_d \right\} \), where \( \alpha_d \) is the angle of view and \( r_d \) is the maximum range at which an object can be detected.
For example, FAB-Map (Cummins & Newman 2008) uses a Chow Liu tree to approximate a probability distribution over visual words learned from SURF descriptors.

Finally, a model of the pdf, \( p_n(z) \), of a false positive measurement generated by clutter is needed. For example, FAB-Map (Cummins & Newman 2008) models the probability that an observation is generated from a place not in the prior map. In our case, \( p_n(z) \) is a product of three terms as in (3) but it is realistic to assume that the bearing measurement is independent of the robot state and uniformly distributed, i.e., with pdf \( 1/|B| \). The class and score likelihoods should be learned from data.

3.2. Observation Model for Multiple Objects

In this section, we combine the single-object observation models into a model of the likelihood of a set \( Z = \{z_1, \ldots, z_m\} \) of semantic observations. Given a robot pose \( x \), let \( Y_d(x) := \{y \in Y \mid p_d(y, x) > 0\} \) be the set of objects, detectable from \( x \). In the reminder, we denote the cardinality of \( Z \) by \( m \) and that of \( Y_d(x) \) by \( n \). As mentioned earlier, the data association \( \pi \) between the semantic observations in \( Z \) and the visible objects in \( Y_d(x) \) is important for constructing the multi-object observation model. The following assumptions are necessary:

(A1) No measurement is generated by more than one object

(A2) An object \( y \in Y \) generates either a single detection with probability \( p_d(y, x) \) in (2) or a missed detection with probability \( 1 - p_d(y, x) \)

(A3) The process of receiving false-positive measurements is Poisson-distributed in time with expected value \( \lambda \) and distributed in space according to the pdf \( p_n(z) \)

(A4) The false-positive process and the object-detection process are independent and all detections are conditionally independent given the robot and object states

(A5) Any two measurements in \( Z \) are independent conditioned on the robot state \( x \), the detectable objects \( Y_d(x) \), and the data association \( \pi \)

We specify the pdf of \( Z \), conditioned on \( x \) and \( Y_d(x) \), in five steps of increasing complexity.

All measurements are false positive  The simplest case is when there are no objects in proximity to the sensor, i.e., \( Y_d(x) = \emptyset \). Then, any generated measurements would be from clutter. The correct observation model in this case is due to assumption (A3) of a Poisson false-positive process:

\[
p(Z \mid \emptyset, x) = \frac{e^{-\lambda |Z|}}{|Z|!} \prod_{z \in Z} p_n(z).
\] (4)

This integrates to 1 if we use the set integral definition in Mahler (2007, Ch.11.3.3):

\[
\int p(Z) dZ := \sum_{m=0}^{\infty} \int p(\{z_1, \ldots, z_m\}) dz_1 \ldots dz_m.
\]

No missed detections and no false positives  Next, suppose that there are detectable objects in proximity to the sensor but let the detection be perfect. In other words, assume that every detectable object generates a measurement, i.e., \( p_d(y, x) = 1 \) for any \( y \in FoV(x) \), and no measurements arise in any other way, i.e., \( \lambda = 0 \). If the number of measurements \( m \) is not equal to the cardinality \( n \) of the set of detectable objects \( Y_d(x) \), then \( p(Z \mid Y_d(x), x) = 0 \). Otherwise, the main challenge in this “perfect detection” case is identifying the correct data association \( \pi \). In other words, it is not clear which of the detectable objects \( Y_d(x) \) on the map produced which of the measurements in \( Z \).

More formally, let \( \Pi_{n,m} \) be the set of one-to-one functions \( g : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\} \) with \( n \leq m \). Due to the “perfect detection” assumption \( m = n \) and a particular data association can be represented by a mapping \( \pi \in \Pi_{n,n} \) from the
set of detectable objects to the set of measurements. In this case, the data association \( \pi \) is just a permutation of \( \{1, \ldots, n\} \) but it is not clear which of the possible \( |\Pi_{n,n}| = n! \) associations is the correct one.

If a particular data association \( \pi \) is chosen, it is straightforward to combine the single-object observation likelihoods in (3) via the independence assumptions (A4), (A5) to obtain the pdf of \( Z \):

\[
p(Z \mid Y_d(x), x, \pi) = \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_{i}, x),
\]

where \( \{y_1, \ldots, y_n\} \) is an enumeration of \( Y_d(x) \). Assuming a uniform prior\(^5\) on the possible data associations:

\[
p(\pi \mid Y_d(x), x) = \frac{1}{n!}, \quad \pi \in \Pi_{n,n},
\]

existing work (e.g. FastSLAM by Montemerlo & Thrun (2003)) resorts to maximum-likelihood estimation and computes the likelihood of \( Z \) as follows:

\[
p(Z \mid Y_d(x), x) \geq \frac{1}{n!} \max_{\pi \in \Pi_{n,n}} \left( \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_{i}, x) \right).
\]

The above equality, however, disagrees with the law of total probability, which states that the data association should be marginalized. The observation model in the “perfect detection” case is:

\[
p(Z \mid Y_d(x), x) = \sum_{\pi \in \Pi_{n,n}} p(Z \mid Y_d(x), x, \pi) p(\pi \mid Y_d(x), x) = \frac{1}{n!} \sum_{\pi \in \Pi_{n,n}} \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_{i}, x). \tag{5}
\]

Intuitively, all associations are plausible and (5) is quantifying the likelihood of \( Z \) by averaging the likelihoods of the individual measurements over all possible data associations. The reason, why existing work avoids this marginalization, is that the summation over all \( n! \) data associations is computationally demanding. However, in Sec. 4, we will present an efficient method for computing (5). Before that, we relax the perfect-detection assumption.

---

\(^5\) It is possible to track the data association distribution over time (see Bar-Shalom et al. (2009)).
See Appendix B.2 for a verification that \( p(\pi \mid Y_d(x), x) \) is a valid pdf. As before, we can derive the likelihood of \( Z \) by marginalizing the data association:

\[
p(Z \mid Y_d(x), x) = \sum_{\pi \in \Pi_{n,m}} p(Z \mid Y_d(x), x, \pi)p(\pi \mid Y_d(x), x)
= \sum_{\pi \in \Pi_{n,m}} \prod_{j=1}^{m} p_z(z_j \mid y_{\pi(j)}, x) \prod_{i \in \mathcal{M}(\pi)} \left[ \frac{1}{m!} \prod_{j=1}^{m} p_d(y_{\pi(j)}, x) \prod_{i \in \mathcal{M}(\pi)} (1 - p_d(y_i, x)) \right] 
= \frac{1}{m!} \prod_{i=1}^{n} (1 - p_d(y_i, x)) \sum_{\pi \in \Pi_{n,m}} \prod_{j=1}^{m} p_d(y_{\pi(j)}, x) p_z(z_j \mid y_{\pi(j)}, x) \prod_{i \in \mathcal{M}(\pi)} (1 - p_d(y_i, x)) 
\]

(6)

The observation model is similar to the “perfect detection” case in (5) but the single-object-measurement likelihoods need to be scaled by the probabilities of detection. If no measurements are received but \( Y_d(x) \neq \emptyset \), the above simplifies to:

\[
p(\emptyset \mid Y_d(x), x) = \prod_{i=1}^{n} (1 - p_d(y_i, x)) 
\]

(7)

**No missed detections but false positives are possible**  In this case, \( n \leq m \) (otherwise \( p(Z \mid Y_d(x), x) = 0 \) and \( \pi \in \Pi_{n,m} \)). Again, let \( \mathcal{A}(\pi) \) be the event that the true-positive detections, \( Y_d(x) \) in this case, are assigned to the measurements in \( Z \) in the particular way specified by \( \pi \). The likelihood of \( \pi \) is:

\[
p(\pi \mid Y_d(x), x) = \mathbb{P}(\mathcal{A}(\pi)) \times \mathbb{P}(\{n \text{ true positives}\}) \times \mathbb{P}(\{m - n \text{ false positives}\}) 
= \frac{1}{m!} \times 1 \times \frac{e^{-\lambda} \lambda^{m-n}}{(m-n)!},
\]

which is a valid pdf (see Appendix B.3). The likelihood of \( Z \) is obtained by marginalizing the data association:

\[
p(Z \mid Y_d(x), x) = \sum_{\pi \in \Pi_{n,m}} p(Z \mid Y_d(x), x, \pi)p(\pi \mid Y_d(x), x)
= \sum_{\pi \in \Pi_{n,m}} \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_i, x) \prod_{j \in \{1, \ldots, m\} \setminus \cup_{i=1}^{n} \{\pi(i)\}} p_k(z_j) e^{-\lambda} \frac{\lambda^{m-n}}{m!} 
= \frac{e^{-\lambda} \lambda^{m}}{m!} \prod_{j=1}^{m} p_k(z_j) \sum_{\pi \in \Pi_{n,m}} \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_i, x) \prod_{j \in \{1, \ldots, m\} \setminus \cup_{i=1}^{n} \{\pi(i)\}} \lambda p_k(z_j) 
\]

(8)

**Both missed detections and false positives are possible**  Finally, consider the most general case that captures all artifacts of object recognition: missed detections, false positives, and unknown data association. If there are no detectable objects close by, i.e. \( n = 0 \), then the pdf of \( Z \) is given by (4). If no measurements are received, i.e. \( m = 0 \), then the pdf of \( Z \) is given by (7). Otherwise, \( \pi \in \Pi_{n,m} \), where \( \Pi_{n,m} \) is the set of functions \( g : \{1, \ldots, n\} \to \{0, 1, \ldots, m\} \) with the property: \( g(i) = g(i') > 0 \Rightarrow i = i' \), which ensures that (A1) is satisfied. The index ‘0’ in the range of \( g \) represents the case of missing a detectable object. For example, it allows for the possibility that all detectable objects are missed (associated with ‘0’), in which case we obtain the term in (7). The number of possible data associations in this case is

\[
|\Pi_{n,m}| = \sum_{k=0}^{\min(n,m)} \binom{n}{k} m^k
\]
where the index $k$ indicates the number of true-positive assignments made by a particular data association $p_i$. The likelihood of $\pi \in \Pi_{n,m}$ with $k$ true-positive assignments is:

$$p(\pi \mid Y_d(x), x) = \mathbb{P}(A(\pi)) \times \mathbb{P}(\{y_i \mid \pi(i) > 0\ \text{are detected}\}) \times \mathbb{P}(\{y_i \mid \pi(i) = 0\ \text{are missed}\}) \times \mathbb{P}(\{m - k \text{ false positives}\})$$

$$= \frac{1}{m!} \prod_{i: \pi(i) > 0} p_d(y_i, x) \prod_{i: \pi(i) = 0} (1 - p_d(y_i, x)) e^{-\lambda(m-k)} (m-k)!,$$

where, as before, $A(\pi)$ is the event that the $k$ true-positive detections are assigned to the measurements in $Z$ in the particular way specified by $\pi$. See Appendix B.4 for a verification that $p(\pi \mid Y_d(x), x)$ is a valid pdf. As before, we can derive the likelihood of $Z$ by marginalizing the data association:

$$p(Z \mid Y_d(x), x) = \sum_{\pi \in \Pi_{n,m}} p(Z \mid Y_d(x), x, \pi)p(\pi \mid Y_d(x), x)$$

$$= \sum_{\pi \in \Pi_{n,m}} \left[ \prod_{i: \pi(i) > 0} p_s(z_{\pi(i)} \mid y_i, x) \right] \prod_{j \in \{1, \ldots, m\} \setminus \cup_{i=1}^{\pi(i)} (\pi(i))} p_e(z_j) \prod_{i: \pi(i) > 0} p_d(y_i, x) p_e(z_{\pi(i)} \mid y_i, x)$$

$$= p(Z \mid \emptyset, x)p(\emptyset \mid Y_d(x), x) \sum_{\pi \in \Pi_{n,m}} \prod_{i: \pi(i) > 0} p_d(y_i, x) p_e(z_{\pi(i)} \mid y_i, x) \left(1 - p_d(y_i, x)\right) \lambda p_e(z_{\pi(i)}) \tag{9}$$

To gain intuition about the semantic observation model in this most general case, refer to Fig. 3. With this model in hand, we can state the Bayesian filtering equations needed for semantic localization.

**Proposition 1.** The Bayesian recursion which solves the Semantic Localization problem is:

**Predict:**

$$p_{t+1|t}(x) = \int p_f(x \mid x', u_t)p_{t|t}(x')dx' \tag{10}$$

**Update:**

$$p_{t+1|t+1}(x) = \eta_{t+1} p(Z_{t+1} \mid Y_d(x), x)p_{t+1|t}(x),$$

where $p(Z_{t+1} \mid Y_d(x), x)$ is the random finite set observation model in (9) and $\eta_{t+1}$ is a normalization constant.
4. Approximating the Set-based Bayes Filter

While the Bayesian recursion with set-valued observations in Prop. 1 is theoretically appealing, like its vector-based counterpart, it is intractable. An accurate and efficient approximation to the set-based Bayes filter is, therefore, the subject of this section. The particle filter (Thrun et al. 2005, Ch.4) is an approximation to the Bayes filter with vector-valued observations, which has been very successful in practice for geometric localization. Since the robot state is still vector-valued, we represent its pdf \( p_t(x) \) at time \( t \) with a set of particles \( \{w^k_{t|t}, x^k_{t|t}\}_{k=1}^N \):

\[
p_{t|t}(x) \approx \sum_{k=1}^N w^k_{t|t} \delta(x - x^k_{t|t}),
\]

where \( \delta(\cdot) \) is a Dirac delta function. The particle-filter implementation of (10), with the motion model (1) as a proposal distribution, is summarized in Alg. 1.

Algorithm 1 Set-based Particle Filter

1: **Input**: Particle set \( \{w^k_{t|t}, x^k_{t|t}\}_{k=1}^N \), motion model pdf \( p_f \), observation model pdf \( p_o \), semantic map \( M \), control input \( u_t \), detection set \( Z_{t+1} \)
2: **Output**: Particle set \( \{w^k_{t+1|t+1}, x^k_{t+1|t+1}\}_{k=1}^N \)
3: for \( k = 1, \ldots, N \) do
4:   **Predict**: Draw \( x^k_{t+1|t} \) from pdf \( p_f(\cdot | x^k_{t|t}, u_t) \)
5:   \( w^k_{t+1|t} \leftarrow w^k_{t|t} \)
6:   **Update**: \( w^k_{t+1|t+1} \leftarrow p(Z_{t+1} | Y_d(x^k_{t+1|t}), x^k_{t+1|t}) w^k_{t+1|t} \)
7:   \( x^k_{t+1|t+1} \leftarrow x^k_{t+1|t} \)
8:   Normalize the weights \( \{w^k_{t+1|t+1}\}_{k=1}^N \) and re-sample if necessary

It appears standard with the exception that, instead of the conventional vector-based measurement update, line 6 requires computing the likelihood of the random set \( Z_{t+1} \) according to (9). As mentioned earlier, it is not apparent how to efficiently compute the sum over all data associations \( \pi \). To gain intuition we begin with the simpler case of “perfect detection” in (5).

Fix a robot state \( x \) and consider the nontrivial case when the received measurements \( Z \) and the detectable landmarks \( Y_d(x) \) have the same cardinality \( m \). We can think about the data association between \( Z \) and \( Y_d(x) \) from a graph-theoretic perspective. Represent the sets \( Y_d(x) \) and \( Z \) by the vertices of a complete balanced bipartite graph. In detail, let \( V_1 := Y_d(x) \) and \( V_2 := Z \) be the vertices and \( E \) be the complete set of edges. Associate the weight \( w_e := p_e(x | y, x) \) with every edge \( e := (z, y) \in E \) and consider the weighted bipartite graph \( G := (V_1, V_2, E, w) \). The data associations \( \pi \), between the objects \( V_1 \) and the measurements \( V_2 \), in the “perfect detection” case (5), in fact, correspond to perfect matchings\(^6\) in \( G \). Given a perfect matching \( \pi \), the associated product term inside the sum in (5) corresponds to its weight. Then, the sum over all \( \pi \) corresponds to the sum of the weights of all perfect matchings in \( G \), which notably is equal to the permanent of the adjacency matrix of \( G \).

**Definition 1** (Permanent). The permanent of an \( n \times m \) matrix \( A = [A(i,j)] \) with \( n \leq m \) is defined as:

\[
\text{per}(A) := \sum_{\pi} \prod_{i=1}^{n} A(i, \pi(i)),
\]

where the sum is over all one-to-one functions \( \pi : \{1, \ldots, n \} \to \{1, \ldots, m \} \). If \( n > m \), then \( \text{per}(A) := \text{per}(A^T) \).

It is now clear that the likelihood of a set of semantic observations in the “perfect detection” case can be obtained by computing the permanent of a matrix.

---

\(^6\) A matching in graph \( G \) is a subgraph of \( G \) in which no two edges share a common vertex. The weight of a matching is the product of all its edge weights. A matching is perfect if it contains all of \( G \)’s vertices.
Proposition 2. The likelihood in (5) of a random finite set of semantic observations, \( Z = \{ z_1, \ldots, z_m \} \), in the case of no false positives and no missed detections, with \( n := |Y_d(x)| = m \) detectable objects, satisfies:

\[
p(Z \mid Y_d(x), x) = \frac{1}{n!} \per(Q),
\]

where \( Q \) is a \( n \times n \) matrix with \( Q(i,j) := p_d(z_j \mid y_i, x) \) and \( \{y_1, \ldots, y_n\} \) is an enumeration of the set \( Y_d(x) \) of detectable objects.

The general case in (9), where both false positives and missed detections are possible can be analyzed using the same graph-matching intuition. The following is our main result and its proof appears in Appendix C.

Theorem 1. Given a robot state \( x \) and set of detectable objects \( Y_d(x) \) with \( n := |Y_d(x)| > 0 \), the likelihood of a random finite set \( Z = \{ z_1, \ldots, z_m \} \) of semantic observations, with \( m > 0 \), when both false-positive and missed detections are possible satisfies:

\[
p(Z \mid Y_d(x), x) = \frac{e^{-\lambda} \lambda^m}{m!} \prod_{z \in Z} p_n(z) \prod_{y \in Y_d(x)} (1 - p_d(y, x))
\]

\[
\times \begin{cases}
\frac{1}{n!} \per\left( \begin{bmatrix} Q & I_n \\ 1_{m,m} & 1_{m,n} \end{bmatrix} \right), & n \leq m \\
\frac{1}{n!} \per\left( \begin{bmatrix} Q^T & I_m \\ 1_{n,n} & 1_{n,m} \end{bmatrix} \right), & m \leq n
\end{cases}
\]

(11)

where \( p_d(y, x) \) is the probability of detecting object \( y \in Y_d(x) \), \( \lambda \) is the expected number of false-positive detections with spatial pdf \( p_n(\cdot) \), \( 1_{n,m} \) is a \( n \times m \) matrix of all ones, and \( Q \) is a matrix with elements:

\[
Q(i,j) := \frac{p_d(y_i, x)p_d(z_j \mid y_i, x)}{(1 - p_d(y_i, x))}\lambda p_n(z_j), \quad i = 1, \ldots, n, \\
j = 1, \ldots, m.
\]

Theorem 1 maps the problem of determining the pdf of \( Z \) in the general case in (9) to the problem of finding the permanent of a \( (m+n) \times (m+n) \) square matrix. The problem is still computationally challenging because computing the permanent of a matrix is \#P-complete\(^7\) (Valiant 1979). However, Theorem 1 allows us to leverage the extensive literature on approximation algorithms for computing the matrix permanent. The proof of Theorem 1 reduces the problem of summing the weights of all matchings in an unbalanced bipartite graph to the problem of summing the weights of all perfect matchings in an unbalanced bipartite graph and then to the problem of summing the weights of all perfect matchings in a balanced bipartite graph. We could stop at the first reduction, which would require calculating the permanent of a rectangular matrix. The reason for the second reduction is that existing permanent-approximation algorithms are much better for square than for rectangular matrices.

An exact method for computing the permanent of a \( d \times d \) matrix, proposed by Ryser (1963) and later improved by Nijenhuis & Wilf (1978, Ch.23), is summarized in Alg. 2. Its time complexity is \( \Theta(d2^{d-1}) \). The dimension of the matrix in (11) is equal to the number of detections returned by the vision algorithm plus the number of detectable objects within the sensor field of view, which in some cases is often small enough to enable a real-time implementation of Alg. 2. Otherwise, there are a number of polynomial-time arbitrarily-close approximations to the permanent computation. For example, Jerrum

\(^7\)A \#P-complete problem is equivalent to computing the number of accepting paths of a polynomial-time nondeterministic Turing machine and \#P contains NP.
The running time was later improved by Bezáková et al. (2006) to $O(\gamma d)$ such that all row and column sums are at least $\gamma d$ for $\gamma \in (0.6, 1]$, Law (2009, Ch.2.2) provides an algorithm with expected running time $O(d^2 \log d + \epsilon^{-2} \log \delta^{-1}))$.

**Proposition 3.** Given $m$ object detections and a semantic map with $L$ objects, the time complexity of Alg. 1 for semantic localization with $N$ particles is $O(N(m + L)^2 (m + L))$, if the measurement update is computed exactly with Alg. 2, and $O(N(m + L)^2 (\log(m + L))^4)$ in expectation, if computed approximately via the randomized method of Bezáková et al. (2006).

While the time complexity in Prop. 3 is in terms of $L$, the number of objects in the environment, the running time of the filter updates actually depends on $n := |Y_d(x)|$, the number of detectable objects within the field of view of each particle, which is typically much smaller than $L$. This, of course, critically depends on the assumption in (2) that the probability of detection is 0 outside of the sensor field of view. Thus, the dimension of the matrix, whose permanent needs to be calculated in (11), scales with the density of the detectable objects within the sensor field of view, rather than the environment size.

Using the idea of Theorem 1, similar results can be obtained for the simpler cases with no false-positives or no missed detections. Appendix D shows the link between the likelihood of a set of semantic observations and the matrix permanent for all cases.

5. Active Semantic Localization

The previous sections described the sensing model and the computational aspects of implementing a particle filter for localization using semantic observations. In this section, we emphasize that the observer can choose motion trajectories actively in order to improve the performance of the semantic localization. As before, let the pdf $p_{t|t}$ of the pose at time $t$ be represented by the particle set $\{u^{k}_{t|t}, x^{k}_{t|t}\}_{k=1}^{N}$. The main idea is to choose a sequence $\sigma := u_{t}, \ldots, u_{t+T-1}$ of control inputs for the next $T$ time steps in order to minimize some measure of uncertainty in the pose. To simplify the notation going forward, assume, without loss of generality, that the current time is $t = 0$. We choose to use the entropy $\mathbb{H}(x_{0:T} \mid Z_{1:T})$ of the current and future poses $x_{0:T}$ conditioned on the future semantic observations $Z_{1:T}$ as the uncertainty criterion. The conditional entropy is an appropriate objective because it quantifies the amount of information needed to describe the outcome of a random quantity (the future poses) given the value of another random quantity (the semantic observations) (Cover & Thomas 2006). Conditional entropy and mutual information, a closely-related measure of information, have been successfully applied to several active perception problems (Charrow et al. 2013, Karasev et al. 2012). Here, we consider the following problem.

---

**Algorithm 2** Permanent (Nijenhuis & Wilf 1978, Ch.23)

```plaintext
1: Input: $d \times d$ matrix $A$  
2: for $i = 1, \ldots, d$ do  
3: \hspace{1cm} $x(i) \leftarrow A(i, d) - \frac{1}{2} \sum_{j=1}^{d} A(i, j)$  
4: \hspace{1cm} $s \leftarrow -1$, $g \leftarrow \text{false}(d, 1)$, \hspace{1cm} $p \leftarrow \prod_{i=1}^{d} x(i)$  
5: for $k = 2, \ldots, 2^{d-1}$ do  
6: \hspace{1cm} if $k$ is even then $j \leftarrow 1$ \hspace{1cm} $\triangleright$ Obtain next gray code subset  
7: \hspace{1cm} else $j \leftarrow 2$  
8: \hspace{1cm} while $g_{j-1}$ is false do  
9: \hspace{2cm} $j \leftarrow j + 1$  
10: \hspace{1cm} end while  
11: for $i = 1, \ldots, d$ do  
12: \hspace{1cm} $x(i) \leftarrow x(i) + z A(i, j)$  
13: \hspace{1cm} $g_{j} \leftarrow \text{not} g_{j}$  
14: end for  
15: end for  
16: return $2(-1)^{d}p$
```

---
Problem 2 (Active Semantic Localization). Given a prior pdf of the pose $x_0$, the semantic map $Y$, and a space $\Sigma := \{\sigma^i \mid \sigma^i := u_0^i, \ldots, u_{T−1}^i\}_{i=1}^M$ of possible control sequences of length $T$, choose the sequence $\sigma^*$, which minimizes the entropy of the current and future poses $x_{0:T}$, conditioned on the future semantic observations $Z_{1:T}$:

$$\sigma^* \in \arg \min_{\sigma \in \Sigma} \mathbb{H}(x_{0:T} \mid Z_{1:T})$$

$$\text{subj. to } x_{s+1} \sim p_f(\cdot \mid x_s, \sigma_s), \ s = 0, \ldots, T−1$$

$$Z_s \sim p(\cdot \mid Y_d(x_s), x_s), \ s = 1, \ldots, T$$

(12)

where $p_f$ is the motion model (1) and $p(\cdot \mid Y_d(x), x)$ is the semantic observation model (11).

Note that the active semantic localization problem as stated in (12) is much simpler than a problem which asks for a closed-loop control policy, minimizing the conditional entropy. Reinforcement learning or dynamic programming approaches can be used to learn such a control policy but the complexity would be much worse than that of the solution we offer here. In (12), the number of states, at which the cost function needs to be evaluated, scales linearly with the number of motion primitives in $\Sigma$ and the number of particles in the prior pose distribution since only the reachable set of pose probability density functions need to be considered. In contrast, an approach for computing a control policy would need to consider the space of all possible probability density functions and evaluate the cost function over it. As we show below, evaluating the cost function in (12) even a single time is computationally very challenging.

An important research question, which we do not address here, is: “How should the set $\Sigma$ of control sequence in (12) be chosen?” Instead, we focus on evaluating the entropy criterion in (12) for a given sequence $\sigma \in \Sigma$ efficiently. We assume that $\Sigma$ has been designed offline and consists of motion primitives, each with $T$ sampling poses, that can provide reasonable coverage of the robot’s surroundings. Sixty locations with outward facing orientations were chosen on the perimeter of a circle of radius 10 m. A differential-drive controller was used to to generate a control sequence of length $T = 5$, which would lead a robot at the origin to each of the selected locations. Fig. 4 shows the resulting set of motion primitives. They are wavy because the controller tries to follow a straight using a discrete set of velocity and angular-velocity input. The waviness provides measurement points with different orientations which is advantageous for localization because the measurements, collected along the way, would be diverse.

5.1. Minimizing Conditional Entropy

While accuracy is of utmost importance for the inference (filtering) process, speed is crucial for the planning process in (12). At each time step, the inference process needs to be carried out for a single set of observations (the actual one), but the planning process needs to envision various measurement realizations and various robot responses. Since it is desirable to plan with many control sequences and for long time horizons, in essence, the inference process needs to be carried out many many times at each planning step. If there is to be any hope for real-time planning, computing the objective function in (12) needs to be extremely efficient.

Given a control sequence $u_{0:T−1}$, the conditional entropy is defined as follows:

$$\mathbb{H}(x_{0:T} \mid Z_{1:T}) := \int \left[ \int -g(x_{0:T}, Z_{1:T}, u_{0:T−1}) \log g(x_{0:T}, Z_{1:T}, u_{0:T−1}) dx_{0:T} \right] p(Z_{1:T}) dZ_{1:T},$$

(13)

where $p(Z_{1:T})$ is the (unconditional) pdf of the semantic observations and we have defined:

$$g(x_{0:T}, Z_{1:T}, u_{0:T−1}) := \frac{p_{u_0}(x_0) \prod_{s=1}^{T} p_f(x_s \mid x_{s−1}, u_{s−1}) \prod_{s=1}^{T} p(Z_s \mid Y_d(x_s), x_s)}{p(Z_{1:T})},$$

(14)
using the assumption that the sets $Z_{1:T}$ are conditionally independent, given the set of detectable objects $\bigcup_{s=1}^{T} Y_d(x_s)$ and the trajectory $x_{1:T}$. This definition makes efficient computation seem hopeless. Even if the measurement sets $Z_{1:T}$ were given, the inside integral in (13) would need to be evaluated for each of the $M$ control sequences with $N^{T+1}$ future particle evolutions, each requiring $T$ evaluations (permanent computations) of the semantic observation likelihood. Assuming exact permanent computations, this makes the complexity of obtaining just the inside integral: $O(M N^{T+1} \sum_{s=1}^{T} |Y_s| + |Z_s|) 2^{(|Y_s| + |Z_s|)}$! In order to address the complexity of this planning problem we will use several approximations.

**Maximum likelihood data association** First, during the planning process, there is no hope for data association via the permanent. We resort to maximum likelihood data association. Given a set $Z$ of semantic observations with $m := |Z|$, for each particle $x$, we construct an association function $\pi : \{1, \ldots, m\} \rightarrow \{0, 1, \ldots, |Y_d(x)|\}$ by processing the measurements $z_j$, $j = 1, \ldots, m$ sequentially. For $z_j$, we compute:

$$\max \left\{ \max_{i \in \{1, \ldots, |Y_d(x)|\}} p_d(y_i, x) p_z(z_j \mid y_i, x), \frac{\lambda}{|Z|} q p_e(z_j) \right\},$$

where $q$ is the number of measurements already assigned to clutter, and let $\pi(j) = i$, if the max is achieved at a detectable object $y_i \in Y_d(x)$, and $\pi(j) = 0$, otherwise. Then, the likelihood of $Z$ is:

$$p(Z \mid Y_d(x), x, \pi) = e^{-\lambda} \prod_{j: \pi(j) = 0} \lambda p_e(z_j) \prod_{y \in D} (1 - p_d(y, x)) \prod_{j: \pi(j) > 0} p_d(y_{\pi(j)}, x) p_z(z_j \mid y_{\pi(j)}, x),$$

where $D$ is the set of unassigned detectable objects. The use of maximum likelihood data association in (14) replaces the $O(\sum_{s=1}^{T} |Y_s| + |Z_s|) 2^{(|Y_s| + |Z_s|)}$ cost of permanent computations by $O(\sum_{s=1}^{T} |Z_s| |Y_s|)$.

**Noiseless motion** Another problematic term in the computational-complexity characterization of the inner integral in (13) is $N^{T+1}$. It is due to the evolution of the set of $N$ particles over the planning horizon $T$. The integral can be simplified significantly by neglecting the noise in the motion model (1). In other words, the robot can be optimistic and plan its future
trajectories with a “perfect motion” assumption (albeit not satisfied in reality):

\[ p_f(x_{s+1} \mid x_s, u_s) = \delta(x_{s+1} - f(x_s, u_s, 0)). \]  

(15)

For the given control sequence \( u_{0:T-1} \), let the (now) deterministic evolution of the particles in the initial particle set \( \{w_{0|0}^k, x_{0|0}^k\} \) over the time horizon \( s = 0, \ldots, T - 1 \) be \( x_{s+1|s+1}^k := f(x_{s|s}^k, u_s, 0) \). Then, the “perfect motion” assumption implies that

\[ g(x_{0:T}, z_{1:T}, u_{0:T-1}) = \sum_{k=1}^{N} w_{0|0}^k \prod_{s=1}^{T} p(Z_s \mid Y_d(x_{s|s}^k), x_{s|s}^k) \prod_{s=0}^{T} \delta(x_{s|s}^k - x_s), \]

which in turn reduces the integral in (13) to:

\[ H(x_{0:T} \mid z_{1:T}) = \int \left[ - \sum_{k=1}^{N} \tilde{w}^k(z_{1:T}) \log \tilde{w}^k(z_{1:T}) \right] p(z_{1:T}) dZ_{1:T}, \]  

(16)

where

\[ \tilde{w}^k(z_{1:T}) := \frac{w_{0|0}^k \prod_{s=1}^{T} p(Z_s \mid Y_d(x_{s|s}^k), x_{s|s}^k)}{p(z_{1:T})} \]

(17)

are the normalized weights of the (updated) particle set at time \( T \). Note that, above, \( p(z_{1:T}) \) is a normalization factor and does not need to be computed explicitly. Combining the result in (16) with maximum likelihood data association, reduces the computational complexity of the inner integral (now a sum) in (13) from \( O(MN(T+1)\sum_{s=1}^{T}(|Y_s|+|Z_s|)2^{Y_s}|Z_s|) \) to \( O(M|Y|\sum_{s=1}^{T} |Z_s|) \). Most importantly, the new complexity does not have an exponential dependence on the problem parameters.

**Particle set summarization** A final reduction in complexity can be obtained by decreasing the number of particles that represent the prior pose distribution. Intuitively, for planning purposes, it is not crucial to represent the distribution accurately but rather to ensure that it contains the competing hypotheses. Charrow et al. (2013) proposed replacing subsets of similar particles with their average in the context of target tracking with range-only sensing. The authors prove that for Gaussian measurement noise, the approximation introduces a bounded error in the mutual information between the observations and the target state. We adopt the same idea here, despite that the measurement noise (except the bearing noise) is not Gaussian. Specifically, we partition the robot state space with a regular square grid and replace the particles, contained in the same cell, with their weighted average. Depending on the size of grid cells, this approximation can reduce the number of particles significantly (see Fig. 5). We emphasize that all these approximations (particle summarization, noiseless motion, maximum likelihood data association) are used only in the planning process. The inference process still uses the full particle set, the full semantic observation model in (11), and considers motion noise.

Now, that the evaluation of the inner integral in (13) has been simplified significantly, we consider the outside integration over the set-valued variables \( z_{1:T} \). Since not even the cardinality of the measurement sets is known, an exact computation would be hopeless. However, given a robot trajectory, the semantic map \( Y \) and the semantic observation model (11) can be used to simulate measurements from the detectable objects and, in turn, obtain a Monte Carlo approximation to (13).

**Monte Carlo integration** The key, to a fast and accurate Monte Carlo approximation of (13), is to simulate measurement sets from \( p(z_{1:T}) \) in a way that the samples are concentrated in regions that make large contributions to the integral. Observe that, due to the particle set approximation of the prior \( p_{0|0} \) and the “perfect motion” assumption, \( p(z_{1:T}) \) is a finite mixture
model:
\[
p(Z_{1:T}) = \int p(Z_{1:T}, x_{0:T}) dx_{0:T}
\]
\[
= \prod_{s=1}^{T} p(Z_s \mid Y_d(x^k_s|s), x^k_s|s) \delta(x_s|s - x_s)p_{0:0}(x_0) dx_{0:T}
\]
\[
= \sum_{k=1}^{N} w^k_{0:0} \prod_{s=1}^{T} p(Z_s \mid Y_d(x^k_s|s), x^k_s|s).
\]
An efficient way to sample from the mixture model \( p(Z_{1:T}) \) is to first sample the mixture component according to the weight \( w^k \) and then sample each \( Z_s \) from the conditional densities. This has the additional benefit of sampling observation sets that are likely to be encountered by the robot and should provide a large contribution to the integral. Thus, for a given control sequence \( u_{0:T-1} \), we follow the following steps to estimate \( \mathbb{H}(x_{0:T} \mid Z_{1:T}) \):

1. Sample a particle \( x^l_{0:0} \) from the prior particle set according to the weights \( w^k_{0:0}, k = 1, \ldots, N \)
2. Compute the particle trajectory \( x^l_{s+1|s+1} := f(x^l_s|s, u_s, 0) \) for \( s = 0, \ldots, T - 1 \)
3. Sample \( Z^l_s \) from the semantic observation model \( p(Z^l_s \mid Y_d(x^l_s|s), x^l_s|s) \) for \( s = 1, \ldots, T \)
4. Compute the normalized updated particle weights \( \tilde{w}^k(Z^l_{1:T}) \) via (17) for \( k = 1, \ldots, N \)
5. Evaluate the inner sum: \( H_l := -\sum_{k=1}^{N} \tilde{w}^k(Z^l_{1:T}) \log \tilde{w}^k(Z^l_{1:T}) \)
6. Repeat the above steps \( N_z \) times to obtain the Monte Carlo approximation: \( \mathbb{H}(x_{0:T} \mid Z_{1:T}) \approx \frac{1}{N_z} \sum_{l=1}^{N_z} H_l \)

Fig. 5 shows a Monte Carlo approximation of the entropy of the robot pose, conditioned on a single future observation set, in a simulated 2-D environment. The results hint at several important considerations regarding active localization. In particular, there is a correlation, among the landmark distribution in the environment, the sensing range and field of view of the robot, and the length of the planning horizon \( T \), which affects the performance. On one hand, if the sensing range and the field of view are unrestricted, there would be no need to for active localization. The filtering process alone will be able to uniquely identify the robot pose. On the other hand, since the planning process is inherently local, if the horizon \( T \) is not long enough to reach perceptually-distinct areas in the environment, the robot can get stuck in a local maximum (the flat red region in Fig. 5) of the entropy surface. Then, all considered motions will have the same cost and no progress will be made. Active localization becomes particularly attractive when the sensing range and the field of view are limited but the environment contains distinct landmarks within the reachable (in \( T \) steps) sensing perimeter. In such scenarios, the planning process can improve both the efficiency and accuracy of the localization filter.

5.2. Localization as a Secondary Objective

Often times, localization is a requisite but secondary objective for a mobile robot. In addition, a robot typically needs to avoid collisions and reach a primary objective, such as a goal pose in the environment. In this section, we discuss how to combine the active semantic localization with collision avoidance and path-planning to a goal pose. First, as discussed by Fox et al. (1998), an additional term can be added to the cost function in order to minimize the probability that a control sequence leads to a collision. Dealing with obstacles in the environment correctly also requires that object visibility is accounted for both in the probability of detection \( p_d(y, x) \) and in the sampling of measurement sets for the Monte Carlo evaluation of the conditional entropy. Second, once the robot is localized well, it can plan a global path \( P = \{ \rho \} \), consisting
of a sequence of poses $\rho$, which leads the robot to its ultimate goal. Along the way, if re-localization is necessary the robot should not deviate significantly from the intended path $P$. Thus, we consider the following three-fold objective:

$$\sigma^* \in \operatorname{arg min}_{\sigma \in \Sigma} \alpha_1 \mathbb{H}(x_{0:T} \mid Z_{1:T}) + \alpha_2 \mathbb{E} \left[ \min_{\rho \in \mathcal{P}} d(x_{T}, \rho) \right] + \alpha_3 \max_{s=1}^{T} \mathbb{P}(x_s \in \text{Collision})$$

subject to $x_{s+1} \sim p_f(\cdot \mid x_s, \sigma_s), \ s = 0, \ldots, T - 1$

$Z_s \sim p(\cdot \mid Y_d(x_s), x_s), \ s = 1, \ldots, T$

where $\mathbb{E} \left[ \min_{\rho \in \mathcal{P}} d(x_T, \rho) \right]$ is the expected minimum deviation of the final motion sequence pose $X_T$ from the global path $\mathcal{P}$ and $\max_{s=1}^{T} \mathbb{P}(x_s \in \text{Collision})$ is the maximum probability of collision along the chosen trajectory. The constants $\alpha_1, \alpha_2, \alpha_3$ specify the relative importance of the three objectives. Due to the “perfect motion” assumption, the last two terms in the cost function can be computed as follows:

$$\max_{s=1}^{T} \mathbb{P}(x_s \in \text{Collision}) = \max_{s=1}^{T} \left( \sum_{k=1}^{N} \mathbb{I}\{x_{s|s}^k \in \text{Collision}\} w_{0|0}^k \right)$$

$$\mathbb{E} \left[ \min_{\rho \in \mathcal{P}} d(x_T, \rho) \right] = \sum_{k=1}^{N} \left( \min_{\rho \in \mathcal{P}} d(x_{T|T}^k, \rho) \right) w_{0|0}^k,$$

where $\mathbb{I}\{x_{s|s}^k \in \text{Collision}\}$ is the indicator of the set $\{x_{s|s}^k \in \text{Collision}\}$.

The performance of the active semantic localization approach is demonstrated in simulation with a differential-drive robot in Fig. 6 and Fig. 7. The task of the robot is to localize itself globally and autonomously and subsequently reach a goal pose specified on the prior map. The initial particle set is uniformly distributed over the whole environment. In the early iterations, minimizing the entropy will be expensive and of little value. In our experiments, the robot either acquires several observation sets without moving (as in Fig. 6 (A)) or chooses motion primitives which minimize the collision probability only (by setting $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$). Once the summarized particle set contains less hypotheses, both the entropy and the probability-of-collision criteria can be enabled to select informative trajectories (see Fig. 6 (B) for details). We used $\alpha_1 = 0.55, \alpha_2 = 0, \alpha_3 = 0.45$ before the first time the robot is localized well (the covariance of the particle set is small). Once well-localized, the robot can plan a path from the mode of the distribution to the goal pose. In our experiments, we used $A^*$ with a cost map that rewards landmark visibility. If along the way to the goal the uncertainty in the robot pose starts to increase due to the motion noise, the robot can carry out the minimization in (18) with all three terms enabled. We used $\alpha_1 = 0.5, \alpha_2 = 0.05, \alpha_3 = 0.45$ in this case (see Fig. 7 (B) for details). The experiments demonstrate that the robot achieves global localization autonomously, avoids collisions in the environment, re-localizes itself if necessary, and reaches the goal pose successfully. Additional simulations, which compare our approach to other active localization techniques, are presented in Sec. 6.4.
Fig. 6. A simulation of a differential-drive robot employing our active semantic localization approach to reach a goal. The environment contains objects from three classes (square, circle, and triangle) in six areas, divided by the black obstacles. The task of the robot is to localize itself (position and orientation) and reach pose 5, indicated by the green arrow on subplot (b). It has a field of view of 94° and a sensing range of 12.5m. The other parameters of the observation model were $p_0 = 0.73, m_0 = 2.7, v_0 = 35, \Sigma_\beta = 5^\circ, \lambda = 0.5$. The robot had no prior information about its initial pose (subplot (A)). The particle distribution converges to 5 ambiguous locations after several semantic observations because a yellow square and a cyan circle are detected repeatedly (subplot (B)). The robot plans its motion (using the motion primitives in Fig. 4) to minimize the probability of collision and the entropy of its pose, conditioned on 5 future sets of semantic observations (subplot (C)). The description continues in Fig. 7.
Fig. 7. Continuation of the active semantic localization simulation from Fig. 6. The robot recognizes correctly that the best way to disambiguate its pose is to visit the bottom-right area (subplot (A)). At this point, there are only two remaining hypotheses and more weight is starting to concentrate around the true pose. Once the robot considers itself localized (the covariance of the particle set is small), it plans a path to the goal in the top-right area. As there are no landmarks along the hallway, the motion noise causes the uncertainty in the robot pose to increase. Using the entropy-minimization criterion, the robot recognizes that it needs to deviate from its intended path and visit an area with landmarks in order to re-localize (subplot (B)). The robot reaches the goal successfully (subplot (C)).
6. Performance Evaluation

This section evaluates the performance of our approach in simulation and in two real-world scenarios. Global localization from semantic observations is demonstrated for a differential-drive robot in Sec. 6.1 and for a Tango phone (Google ATAP group 2014) in Sec. 6.2.

Semantic information was obtained using deformable part models (DPM, Felzenszwalb et al. 2010), which achieve state-of-the-art performance in single-image object recognition. Given an input image, an image pyramid is obtained via repeated smoothing and subsampling. Histograms of oriented gradients are computed on a dense grid at each level of the pyramid. For each object class (in the set $C$), a detector is applied in a sliding-window fashion to the image pyramid, in order to obtain detection scores at each pixel and scale (see Fig. 8). The detection scores above a certain threshold are returned, along with bounding box, class, and bearing information. The collection of all such measurements at time $t$ forms the random finite set $Z_t$. A significant increase in detection speed is obtained via the active approach of Zhu et al. (2014), which optimizes the order of filter evaluations and the time at which to stop and make a decision.

In our experiments, it was sufficient to represent the state of an object $y$ with its position and class because orientation and appearance variations are captured well by a DPM-based detector. If necessary, our model can incorporate rich appearance and shape signatures by extending the object state $y$ and training an appropriate observation model. This is likely to make the data association more unimodal (i.e., make the terms in the sum in (9) dominated by the weight of a single data association), in which case the maximum likelihood data association approach (Sec. 5.1) would perform well. We emphasize that the permanent approach can handle this scenario efficiently too. As permanent approximation methods rely on Monte-Carlo sampling from the data associations, fewer samples can be used to speed up the computations. The connection between the observation model and the permanent incorporates this naturally and leverages state-of-the-art algorithms.

6.1. Mobile Robot Semantic Localization

Robot Platform We carried out simulations and real-world experiments in an indoor environment using a differential-drive robot equipped with an inertial measurement unit (IMU), magnetic wheel encoders, a Kinect RGB-D camera with Nyko Zoom wide-angle lens, and a Hokuyo UTM-30LX 2D laser range finder. The IMU and the encoders were integrated using a differential-drive model and Gaussian noise was added to obtain the motion model in (1). Only the RGB images were used for semantic observations. The depth was not used, while the lidar was used to provide a ground-truth trajectory in the real-world experiments, via geometric Monte-Carlo localization. Alg. 1 was used for semantic localization. The measurement updates were performed using the exact permanent algorithm (Alg. 2). The performance is demonstrated for global localization, which means that the robot had no prior information about its initial pose.

The detection model $p_d(y, x)$, the measurement likelihood $p_z(z \mid y, x)$, and the clutter model $p_c(z)$ were obtained from training data as discussed in Sec. 3.1. The angle of view of the wide-angle lens was 94°, the detection range - 10 meters, and the following parameters were learned: $p_0 = 0.92$, $m_0 = 3.5$, $v_0 = 20.52$, $\Sigma_{\beta} = 4°$. The semantic map in the real-world experiment contained doors (class 1) and chairs (class 2). The confusion matrix was:

$$p_c(c \mid y^c) = \begin{bmatrix} 0.94 & 0.08 \\ 0.06 & 0.92 \end{bmatrix}$$

while the detection score likelihood is shown in Fig. 16.

Simulation Results A simulated environment of size 25 x 25 m$^2$ was populated by objects with randomly-chosen positions and classes (see Fig. 12). The robot motion was simulated along a pre-specified trajectory. Semantic observations were simulated using the learned detection, clutter, and measurement likelihood models. The error in the estimates, averaged over 50 repetitions with different randomly-generated scenes, is presented in Fig. 15. Since the robot starts with a uniform
Table 1: Comparison of maximum likelihood data association (MLD) and our permanent-based data association approach (PER) with the exact permanent computation (Alg. 2) on the four robot datasets (Fig. 9) and the simulations in Fig. 12 and Fig. 13. Two types of initializations were used: local (L), for which the initial particle set had errors of up to 1 m and 30°, and global (G), for which the initial particle set was uniformly distributed over the whole environment. Number of particles (NP) in thousands, position error (PE), orientation error (OE), and filter update time (UT), averaged over time, are presented. The first MLD(G) column uses the same number of particles as PER(G), while the second uses a large number in an attempt to improve the performance.

<table>
<thead>
<tr>
<th></th>
<th>Fig. 9</th>
<th>MLD(L)</th>
<th>MLD(G)</th>
<th>MLD(G)</th>
<th>PER(L)</th>
<th>PER(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP [K]</td>
<td>0.50</td>
<td>3.00</td>
<td>40.0</td>
<td>0.50</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>PE [m]</td>
<td>0.26</td>
<td>22.9</td>
<td>0.31</td>
<td>0.26</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>OE [deg]</td>
<td>2.54</td>
<td>107</td>
<td>2.75</td>
<td>2.67</td>
<td>2.69</td>
<td></td>
</tr>
<tr>
<td>UT [sec]</td>
<td>0.023</td>
<td>0.060</td>
<td>0.600</td>
<td>0.065</td>
<td>0.320</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fig. 12</th>
<th>MLD(L)</th>
<th>MLD(G)</th>
<th>MLD(G)</th>
<th>PER(L)</th>
<th>PER(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP [K]</td>
<td>0.50</td>
<td>5.00</td>
<td>10.0</td>
<td>0.50</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>PE [m]</td>
<td>15.3</td>
<td>24.9</td>
<td>17.3</td>
<td>0.32</td>
<td>0.72</td>
<td></td>
</tr>
<tr>
<td>OE [deg]</td>
<td>67.0</td>
<td>68.8</td>
<td>72.8</td>
<td>4.58</td>
<td>9.17</td>
<td></td>
</tr>
<tr>
<td>UT [sec]</td>
<td>0.012</td>
<td>0.062</td>
<td>1.100</td>
<td>0.042</td>
<td>0.400</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fig. 13</th>
<th>MLD(L)</th>
<th>MLD(G)</th>
<th>MLD(G)</th>
<th>PER(L)</th>
<th>PER(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP [K]</td>
<td>0.50</td>
<td>24.0</td>
<td>10.0</td>
<td>0.50</td>
<td>24.0</td>
<td></td>
</tr>
<tr>
<td>PE [m]</td>
<td>26.9</td>
<td>48.8</td>
<td>26.9</td>
<td>0.11</td>
<td>2.35</td>
<td></td>
</tr>
<tr>
<td>OE [deg]</td>
<td>74.9</td>
<td>112</td>
<td>74.9</td>
<td>2.08</td>
<td>4.10</td>
<td></td>
</tr>
<tr>
<td>UT [sec]</td>
<td>0.027</td>
<td>0.760</td>
<td>3.340</td>
<td>0.062</td>
<td>2.620</td>
<td></td>
</tr>
</tbody>
</table>

prior over the whole environment, the error is large in the initial iterations. Multiple hypotheses are present until enough semantic measurements are obtained to localize the robot uniquely. The performance is also demonstrated in a challenging scenario with a lot of ambiguity in Fig. 13. The reason for using only two classes in the experiments was to increase the ambiguity in the data association. Our approach can certainly handle more classes and a higher object density. Fig. 14 shows a simulation with clutter rate $\lambda = 4$ and 150 objects from 5 classes in a $25 \times 25$ m² area. Scenarios with such high object density necessitate the use of an approximate permanent algorithm for real-time operation.

Real Experiments The robot was driven through a long hallway containing doors and chairs. Four data sets were collected from the IMU (at 100 Hz), the encoders (at 40 Hz), the lidar (at 40 Hz), and the RGB camera (at 1 Hz). Lidar-based geometric localization was performed via the ROS amcl package (Howard & Gerkey 2002) and the results were used as the ground truth. Extension 1 contains a video of the experiment. The lidar and semantic estimates of the robot trajectory are shown in Fig. 9. The error between the two, averaged over the 4 runs, is presented in Fig. 17. The error is large initially because, unlike the lidar-based localization, our method was started with an uninformative prior. Nevertheless, after the initial global localization stage, our approach achieves average errors in the position and orientation estimates of less than 35 cm and 10°, respectively. The particle filter evolution is illustrated in Fig. 11 along with some object detections.

Comparison with Maximum Likelihood Data Association We compared our permanent-based data association (PER) approach to the more traditional maximum likelihood data association (MLD) approach, used for example in FastSLAM (Montemerlo & Thrun 2003). PER is based on Alg. 1 with an exact permanent computation (Alg. 2) on line 6. MLD is based on Alg. 1 also but the set of detections on line 6 is processed sequentially as described in Sec. 5.1. The two approaches were compared on the four real datasets (Fig. 9) and on the simulations in Fig. 12 and Fig. 13. Because we assume semantically-meaningful measurements, the observation sets in our comparison had relatively low cardinalities. Of course, if there are many observations per time step (e.g. SIFT features), MLD would be significantly more efficient than the exact permanent algorithm. In future work, we plan to compare MLD to our approach with an approximate permanent computation.

The reported times are from a MATLAB implementation on a computer with i7 CPU@2.3GHz and 16GB RAM. The timing results include only the time needed to perform data association and update the weights for all particles. The time required for object recognition is not included because it is the same for both methods.
The performance is presented in Table 1 for two types of initializations: local (L), for which the initial particle set had errors of up to 1 m and 30°, and global (G), for which the set was distributed uniformly over the environment. MLD(L) performs as well as PER(L) in the real experiments and in Fig. 13. In Fig. 12, the data association is highly multi-modal and MLD(L) does not converge even with 15K particles. This is reinforced in the global initialization cases. While PER(G) performs well with 3K particles, MLD(G) needs 40K to converge consistently on the real datasets and is slower at the same level of robustness. In Fig. 12 and Fig. 13, MLD(G) does not converge even with 100K particles. We conclude that once the particles have converged correctly MLD performs as well as PER. However, with global initialization or ambiguous data association, MLD makes mistakes and can never recover, while PER is robust with a small number of particles.

6.2. Global Localization for Project Tango

The Project Tango phone (Google ATAP group 2014) is designed to track its full 3-D motion and create a geometric map of the environment. However, it does not use an absolute reference to the world and provides localization, only with respect to its initial frame of reference. We demonstrate that our semantic localization approach can provide global positioning of the Tango phone within an existing map of semantically-meaningful landmarks.

The Tango phone is equipped with an IMU, a front-facing (120° field of view) camera, a rear-facing RGB/IR narrow (68° field of view) camera, and a rear-facing fisheye (180° field of view) camera. It provides a 6-D position-quaternion estimate of its pose and associated covariance, over time, within the initial frame of reference. In our experiments, this local trajectory was used as the motion model (1) in the prediction step in Alg. 1. The update step was performed using semantic observations (class, score, and bearing) only from the narrow camera RGB images. The same hallway, as in the robot experiment (Sec. 6.1), was traversed several times with the phone. RGB images from the narrow camera (at 30 Hz) and the Tango visual odometry (at 30 Hz) were recorded. The prior semantic map of the environment (see Fig. 10) contained doors (class 1), red chairs (class 2), and brown chairs (class 3). Two of the runs were used to train the object detector and to learn the observation model parameters: sensing range 15 m, $p_0 = \begin{pmatrix} 0.71 & 0.81 & 0.82 \end{pmatrix}^T$, $v_0 = 35.4$, $m_0 = 2.7$, $\Sigma_\beta = 5^\circ$, $\lambda = 0.76$, and confusion matrix:

$$p_c(c \mid \gamma^c) = \begin{bmatrix} 0.98 & 0 & 0 \\ 0 & 0.94 & 0.08 \\ 0.02 & 0.06 & 0.92 \end{bmatrix}.$$  

Our semantic localization approach was used to recover the global Tango trajectories in the rest of the runs. Since the prior semantic map contained 2-D object positions, only the horizontal bearing angle was used to update the 2-D position and yaw angle of the phone. A good estimate of the phone’s pitch and roll angles can be obtained from the local 6-D trajectory (provided by the Tango phone). Thus, the global semantic localization was performed in 5-D (without the z-axis). This can be extended, of course, if vertical bearing measurements are used and the landmarks in the prior map are annotated with z-coordinates. The active DPM-based detector of Zhu et al. (2014) was used for object recognition. The likelihoods of the semantic observations were computed via the exact permanent algorithm (Alg. 2). Videos, from two of the experiments, are provided in Extension 2 and Extension 3. The phone trajectory, recovered from the second run (Extension 3), is shown in Fig. 10. Unfortunately, ground-truth trajectories are not available for these experiments. The videos show 9 global localization trials, in which, on average, 11 sets of semantic observations were needed to obtain an accurate estimate of the phone pose in the prior map. They demonstrate that our algorithm can repeatedly re-localize and track the phone pose within the same environment. Moreover, our semantic localization approach is very robust to perceptual aliasing and can improve the visual odometry provided by the phone in ambiguous environments and when closing loops.
Fig. 8. A component of the deformable part model of a chair (top) and scores (bottom) from its evaluation on an image (middle) containing four chairs.

Fig. 9. Robot trajectories estimated by lidar-based geometric localization (red), image-based semantic localization (blue), and odometry (green) in the real experiment described in Sec. 6.1. The starting position, the door locations, and the chair locations are denoted by the red cross, the yellow squares, and the cyan circles, respectively. See Extension 1 for more details.

Fig. 10. Tango phone trajectory (red) estimated via semantic localization in the real experiment described in Sec. 6.2. The semantic map contains doors (yellow squares), red chairs (cyan circles), and brown chairs (blue triangles). Ground-truth information is not available for this experiment. See Extension 3 for more details.

Fig. 11. Particle filter evolution (bottom) and object detections (top) during a real semantic localization experiment.

Fig. 12. A simulated environment with 45 objects from two classes (yellow squares, blue circles). The plots show the evolution of the particles (red dots), the ground truth trajectory (green), and the estimated trajectory (red). The expected number of clutter detections was set to $\lambda = 2$.

Fig. 13. A simulated example of semantic localization in the presence of severe perceptual aliasing. The ground truth trajectory (blue) and the evolution of the particle positions (red points) and orientations (red lines, top left) are shown.
6.3. Evaluation on the KITTI Dataset

This section evaluates the performance of our global semantic localization approach on the KITTI visual odometry dataset (Geiger et al. 2013). The dataset consists of 22 sequences of color stereo images (0.5 Megapixels in png format) and 3-D Velodyne point clouds (100k points per frame) recorded from a vehicle, driving through residential areas. Eleven sequences (00-10) contain ground-truth vehicle trajectories provided by a GPS/IMU system with real-time kinematic correction. Only sequences {00, 05, 06, 07, 08, 09, 10} were used in our experiments because the rest either had too few static landmarks or did not contain ground-truth information. The cars (class 1) and windows (class 2) in the RGB image sequences were labeled in order to build prior semantic maps. The Velodyne range information was mapped to the images and the ground-truth trajectories were used to project the labeled objects to the 3-D world coordinate frame. The final semantic maps are provided in Extension 13 and the map for sequence 00 is shown in Fig. 18.

The pre-trained deformable-part car models provided in the KITTI object dataset were used for car recognition. Sequence 07 was used to train a deformable-part-model-based window detector and to obtain parameters for the single-object detection model \( p_d(y,x) \), observation likelihood \( p_z(z \mid y, x) \), and clutter model \( p_c(z) \). The car detection model was nonzero for a distance range of \([3,33]\) meters and used parameters \( p_0 = 0.7, m_0 = 11.8, v_0 = 14\). The window detection model was nonzero in the range \([7,24]\) with parameters \( p_0 = 0.7, m_0 = 12.7, v_0 = 7\). The rest of the parameters were: \( p_c(c \mid y^c) = I_2 \), sensing range 32 m, field of view 80°, \( \Sigma_\beta = 5^\circ \), and \( \lambda = 0.5\).

Visual odometry via Viso2 (Geiger et al. 2011) was used for the prediction step in Alg. 1. Viso2 provides a 6-D local trajectory. As in the Tango experiments (Sec. 6.2), only the 2-D position and the yaw angle of the trajectory were updated via our method. The likelihoods of the car-window semantic observations were computed via the exact permanent algorithm (Alg. 2). Global semantic localization was carried out on sequences 00, 05-10. The vehicle trajectory, recovered from sequence 00, is shown in Fig. 18 along with some object detections. A video of the experiment is included in Extension 5. An additional experiment, in which the localization was restarted every 400 iterations, was carried out on sequence 00 and
is presented in Extension 4. The experiment demonstrates that our algorithm can repeatedly and successfully re-localize and track the vehicle pose within the same environment. Finally, the results of the global semantic localization on the rest of the sequences (05-10) are presented in Extensions 6 - 11. The localization errors with respect to the ground-truth trajectories from all experiments are presented in Fig. 19. Initially, the errors are large because our method is started with an uninformative prior (a uniform distribution in the area around the landmarks). Nevertheless, after the initial global localization stage, our approach achieves average errors in the position and orientation estimates of less than 1 m and $5^\circ$, respectively. Even though the data association obtained from permanent computations is very robust to perceptual aliasing, sometimes, the ambiguity in the environment is large enough to cause particle depletion problems. For example, if resampling is done too frequently and there is no way to detect if the system is lost (i.e., the particle distribution is never reset), the localization might fail. Such a fail case is shown in Extension 12.

To demonstrate that localization from semantic observations is complementary to existing odometry and SLAM techniques, we also carried out tracking experiments, in which the initial vehicle pose was known. In Fig. 20, the position and orientation errors obtained from visual odometry are compared to those obtained from visual odometry, combined with our semantic localization approach. Even though visual odometry provides excellent tracking results by itself, the addition of semantic observations provides a reference to the absolute (semantic map) frame and improves the results.

### 6.4. Active Semantic Localization Simulations

In this section, we evaluate the performance of the active semantic localization approach (Sec. 5.1) by comparing it to three other active localization methods in simulation. The simulations use a differential-drive robot model and the motion primitives in Fig. 4. Fifty environments of size $120 \times 120$ m$^2$ containing 300 objects from 3 classes (square, circle, triangle) were generated by sampling random points and placing one of three possible four-object structures (square-square-circle-triangle, square-circle-circle-triangle, or square-triangle-circle-triangle) in order to create perceptual ambiguity (see Fig. 21). A start and a goal pose for the robot were chosen in the top-left and bottom-right corners of each environment, respectively. The semantic localization algorithm (Alg. 1 with exact permanent computation via Alg. 2) was initialized with a uniform particle distribution over the whole environment. Semantic observations were simulated using the learned detection, clutter, and measurement likelihood models for cars and windows (from the KITTI dataset in Sec. 6.3) and red

---

8 The orientation error is the cumulative error in degrees between the ground-truth and the estimated orientations divided by path length in meters.
Fig. 19. Position (euclidean distance) and orientation errors of the vehicle trajectories recovered via global semantic localization on sequences \{00, 05, 06, 07, 08, 09, 10\} from the KITTI visual odometry dataset. The plot, titled “Sequence 00 + Restarts”, shows results from an experiment in which the localization was restarted every 400 iterations. Extensions 4 - 12 provide videos of all experiments.

Fig. 20. Average translation and orientation errors obtained from visual odometry via Viso2 (Geiger et al. 2011) and from visual odometry combined with semantic localization on sequences \{00, 05, 06, 07, 08, 09, 10\} from the KITTI visual odometry dataset. Both methods use a known starting vehicle pose, i.e., perform tracking instead of global localization.

chairs (from the Tango experiments in Sec. 6.2), corresponding to squares, circles, and triangles, respectively. For each environment, semantic observations were collected, while moving in a straight line, and the filter was updated without resampling until there were 100 effective particles (see Fig. 21 for an example of the initial particle set). Starting with this initial particle distribution, the following methods were compared to our active semantic localization (ASL) approach:

- **RND**: chooses a motion primitive (from the ones in Fig. 4) at random
- **MIN**: chooses the motion primitive, which drives the mean of the particle distribution closest to the closest landmark in the environment
- **BEM**: chooses a motion primitive by minimizing the entropy of the robot pose conditioned on the future bearing measurements only (see Appendix E for details).
Fig. 21. The left plot shows the trajectories, followed by four different active-semantic-localization approaches, which localize and lead a differential-drive robot to a goal pose in a simulated environment containing 300 objects from 3 classes (yellow square, cyan circle, blue triangle). The initial particle distribution is shown by the black dots. The four methods are: (1) ASL: active semantic localization presented in Sec. 5.1, (2) RND: chooses motion primitives at random, (3) MIN: chooses the motion primitive that drives the particle mean closest to the closest landmark, (4) BEM: bearing-only entropy minimization (see text for details). The right plot shows the particle-distribution entropies along the trajectories associated with each method.

Table 2: Comparison of the average (over 50 simulated environments) performance of the four active-semantic-localization approaches, referenced in Fig. 21. The average euclidean distance between the start and the goal positions was 251 m. If the goal was not reached in 1000 iterations, the experiment was terminated. The table presents averages of the number of iterations until termination, the euclidean distance to the goal at termination, the entropy in the particle distributions, and the position and orientation errors with respect to the ground-truth robot trajectory.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>595</td>
<td>35.02</td>
<td>3.00</td>
<td>22.17</td>
<td>12.48</td>
</tr>
<tr>
<td>MIN</td>
<td>508</td>
<td>35.58</td>
<td>2.83</td>
<td>27.58</td>
<td>11.98</td>
</tr>
<tr>
<td>BEM</td>
<td>447</td>
<td>16.22</td>
<td>3.05</td>
<td>18.49</td>
<td>11.13</td>
</tr>
<tr>
<td>ASL</td>
<td>345</td>
<td>12.09</td>
<td>2.05</td>
<td>12.59</td>
<td>6.37</td>
</tr>
</tbody>
</table>

The methods were used to choose motion primitives if the entropy in the particle distribution (computed by discretizing the robot state space into cells of size $2.5 \times 2.5 \text{ m}^2$ and $25^\circ$ and replacing the particles, contained in the same cell, with their average) increased above 2.5 bits. Otherwise, if the entropy decreased under 2.5 bits, each method planned a trajectory from the mean of the particle distribution to the goal state using A* and followed it using a deterministic controller. The trajectories, followed by the four active localization methods, and the associated particle-distribution entropies are shown for one of the simulated environments in Fig. 21. Performance statistics, averaged over the 50 environments, are presented in Tab. 2. The results show that, on average, our active-semantic-localization approach reaches the goal in less iterations, with lower particle-distribution entropy, and with lower estimation error, compared to the other three approaches. Of course the approach is also a lot more computationally demanding. As expected, the random approach performs the worst because when there no landmarks in proximity to the robot, it might spend a long time until re-localization. The BEM approach demonstrates much better performance but the main problem with it and the MIN method is that they rely on the mean of the particle distribution for planning. Sometimes, when the mean is far from the true pose, these methods choose trajectories, which do not necessarily result in improved localization accuracy.
7. Conclusion

Modeling the semantic information obtained from object recognition with random finite sets enables a unified treatment of data association, missed detections, and clutter. The efficient use of this semantic observation model for Bayesian filtering depends critically on the connection with the matrix permanent. Simulations of our approach showed precise and robust localization from semantic information in various scenarios and over many repetitions. Compared to maximum likelihood data association, our solution is more robust to perceptual aliasing and offers superior performance in cases of global localization and loop-closure. The real experiments demonstrated that the accuracy of the semantic localization method is comparable with the laser-based geometric approaches. More importantly, due to the semantically-meaningful observations, our approach is able to repeatedly solve the global localization problem in real environments. Finally, to enable autonomous localization, we addressed the active semantic localization problem, in which the observer's trajectory is selected to minimize the entropy in the pose distribution conditioned on the future semantic measurements. The simulations demonstrated that our method, although computationally demanding, outperforms simpler active localization heuristics. Future work will attempt to replace the particle filter with more sophisticated estimation techniques that handle both discrete (e.g., object classes) and continuous (e.g., object poses) measurements. Examples include inference algorithms for graphical models in the spirit of Kaess et al. (2012) or online optimization techniques such as Jadabaie et al. (2015). Extensions to semantic mapping and semantic SLAM are of great interest as well.

Acknowledgments

We gratefully acknowledge support by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA and the following grants: NSF-OIA-1028009, ARL RCTA W911NF-10-2-0016, NSF-DGE-0966142, and NSF-IIS-1317788. We thank Chris Clingerman for his help with the MAGIC robot.

Appendix A. Index to Multimedia Extensions

The multimedia extension page can be found at http://www.ijrr.org.
Table 3: Index to Multimedia Extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Media Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Video</td>
<td>Mobile robot localization from semantic observations</td>
</tr>
<tr>
<td>2</td>
<td>Video</td>
<td>Global positioning of the Tango phone</td>
</tr>
<tr>
<td>3</td>
<td>Video</td>
<td>Global positioning of the Tango phone</td>
</tr>
<tr>
<td>4</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 00 with several restarts</td>
</tr>
<tr>
<td>5</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 00</td>
</tr>
<tr>
<td>6</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 05</td>
</tr>
<tr>
<td>7</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 06</td>
</tr>
<tr>
<td>8</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 07</td>
</tr>
<tr>
<td>9</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 08</td>
</tr>
<tr>
<td>10</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 09</td>
</tr>
<tr>
<td>11</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 10</td>
</tr>
<tr>
<td>12</td>
<td>Video</td>
<td>Global semantic localization on KITTI dataset sequence 08 (fail case)</td>
</tr>
<tr>
<td>13</td>
<td>Data</td>
<td>Car and window positions used for the semantic maps in the KITTI dataset experiments</td>
</tr>
</tbody>
</table>

Appendix B. Validity of the Data Association Probability Density Functions

For simplicity, let $p_d(y_i, x) = p_d(y_j, x) \equiv p_d$ for all $i, j$ in this section. We verify that $p(\pi \mid Y_d(x), x)$ is a valid probability density function in each of the following cases.

**Appendix B.1. No missed detections and no false positives**

In this case, $\pi \in \Pi_{n,n}$ with likelihood $p(\pi \mid Y_d(x), x) = 1/n!$, which sums to one as follows:

$$\sum_{\pi \in \Pi_{n,n}} p(\pi \mid Y_d(x), x) = \frac{1}{n!} |\Pi_{n,n}| = 1.$$
Appendix B.2. No false positives but missed detections are possible

In this case, \( \pi \in \Pi_{m,n} \) with likelihood:

\[
p(\pi \mid Y_d(x), x) = \frac{1}{m!} p_d^m (1 - p_d)^{n-m},
\]

which sums to one using the Binomial theorem:

\[
\sum_{m=0}^{n} \sum_{\pi \in \Pi_{m,n}} p(\pi \mid Y_d(x), x) = \sum_{m=0}^{n} m P_m \frac{1}{m!} p_d^m (1 - p_d)^{n-m} = (p_d + (1 - p_d))^n = 1.
\]

Appendix B.3. No missed detections but false positives are possible

In this case, \( \pi \in \Pi_{n,m} \) with likelihood:

\[
p(\pi \mid Y_d(x), x) = \frac{1}{m P_n} e^{-\lambda} \lambda^{m-n} \frac{1}{(m-n)!},
\]

which sums to one as follows:

\[
\sum_{m=n}^{\infty} \sum_{\pi \in \Pi_{n,m}} p(\pi \mid Y_d(x), x) = \sum_{m=n}^{\infty} m P_n \frac{1}{m P_n} e^{-\lambda} \lambda^{m-n} \frac{1}{(m-n)!} = 1.
\]

Appendix B.4. Both missed detections and false positives are possible

The likelihood of \( \pi \in \Pi_{n,m} \) with \( k \) true positive assignments is:

\[
p(\pi \mid Y_d(x), x) = \frac{1}{m P_k} p_d^k (1 - p_d)^{n-k} e^{-\lambda} \lambda^{m-k} \frac{1}{(m-k)!},
\]

which sums to one as follows:

\[
\sum_{m=0}^{\infty} \sum_{k=0}^{\min\{m,n\}} m P_k \frac{1}{m P_k} p_d^k (1 - p_d)^{n-k} e^{-\lambda} \lambda^{m-k} \frac{1}{(m-k)!} = \sum_{m=0}^{\infty} \sum_{k=0}^{\min\{m,n\}} \frac{n!}{k! (n-k)!} p_d^k (1 - p_d)^{n-k} e^{-\lambda} \lambda^{m-k} \frac{1}{(m-k)!} = 1.
\]

Appendix C. Proof of Theorem 1

Let \( V_1 := Y_d(x) \) and \( V_2 := Z \) be the vertices of a weighted complete bipartite graph \( G := (V_1, V_2, E, w) \), where the weight \( w_e \) associated with \( e := (i, j) \in E \) is \( Q(i, j) \). The functions \( \pi \) in (9) specify different data associations between the objects \( V_1 \) and the measurements \( V_2 \). The introduction of missed detections ("0" in the range of \( \pi \)) means that some detectable objects need not to be assigned to a measurement in \( Z \). As any object could be missed, the data associations \( \pi \) correspond to matchings (not necessarily perfect as before) in the graph \( G \). Given a matching \( \pi \), the associated product term inside the sum in (9) corresponds to the weight of \( \pi \). The sum over all \( \pi \) corresponds to the sum of the weights of
Lemma 2. Let \( G \) be a weighted complete bipartite graph. Define \( \text{per}_k(A) := \sum_{\alpha \in Q_{k,n}, \beta \in Q_{k,m}} \text{per}(A[\alpha, \beta]), \quad k = 1, \ldots, n \)

Then, the sum in (9) is equal to the sum over the weights of all \( k \)-matchings:

\[
\sum \prod_{\pi, i, \pi(i) > 0} \frac{p_d(y_i, x)p_x(z_{\pi(i)} \mid y_i, x)}{(1 - p_d(y_i, x)) \lambda p_n(z_{\pi(i)\})} \right|_{Y_d(x)} = \sum_{k=0}^{\text{per}_k(Q)}
\]

where the assumption that \( |Y_d(x)| \leq m \) is used. The following two lemmas describe a reduction from the problem of summing all subpermanent sums of a rectangular matrix (or perfect matchings in an unbalanced bipartite graph) to the problem of the permanent of a rectangular matrix (or perfect matchings in an unbalanced bipartite graph) and then to the problem of the permanent of a square matrix (or perfect matchings in a balanced bipartite graph).

**Lemma 1.** Let \( A_{n,m} \) be an \( n \times m \) matrix with \( n \leq m \). Then,

\[
\sum_{k=0}^{n} \text{per}_k(A_{n,m}) = \text{per} \left( \begin{bmatrix} A_{n,m} & I_n \end{bmatrix} \right).
\]

**Proof:** Associate \( A \) with a weighted complete bipartite graph \( G_A := (V_1 := \{1, \ldots, n\}, V_2 := \{1, \ldots, m\}, E, w_A) \), where the weights \( w_A \) correspond with the entries of \( A \). To obtain the graph \( G_B \) associated with \( B := \begin{bmatrix} A_{n,m} & I_n \end{bmatrix} \)

add \( n \) dummy nodes \( V_3 \) to \( V_2 \) and \( n \) edges of weight 1. For \( k \in \{0, \ldots, n\} \), fix subsets \( \alpha \in Q_{k,n} \) and \( \beta \in Q_{k,m} \) using the notation from Def. 2. A perfect matching in \( G_B \) associated with \( \alpha \) and \( \beta \) corresponds to:

- A \( k \)-matching between \( \alpha \in V_1 \) and \( \beta \in V_2 \) of weight \( \text{per}(A[\alpha, \beta]) \)
- A \( (n - k) \)-matching between \( V_1 \setminus \alpha \) and \( V_3 \) of weight 1

Then, \( \text{per}(B) \) is the sum of all perfect matchings in \( G_B \):

\[
\text{per}(B) = \sum_{k=0}^{n} \sum_{\alpha \in Q_{k,n}, \beta \in Q_{k,m}} \text{per}(A[\alpha, \beta]) = \sum_{k=0}^{n} \text{per}_k(A),
\]

where the last equality follows directly from Def. 2.

**Lemma 2.** Let \( A_{n,m} \) be an \( n \times m \) matrix with \( n \leq m \). Then,

\[
\text{per}(A_{n,m}) = \frac{1}{(m-n)!} \text{per} \left( \begin{bmatrix} A_{n,m} & 1_{m-n,m}^T \end{bmatrix} \right)
\]

where \( 1_{m-n,m} \) is a \( (m-n) \times m \) matrix of all ones.

**Proof:** Associate \( A \) with a weighted complete bipartite graph \( G_A := (V_1 := \{1, \ldots, n\}, V_2 := \{1, \ldots, m\}, E, w_A) \), where the weights \( w_A \) correspond with the entries of \( A \). To obtain the graph \( G_B \) associated with \( B := \begin{bmatrix} A_{n,m}^T & 1_{m-n,m}^T \end{bmatrix} \)

add
(m − n) dummy nodes $V_3$ to $V_1$ and $(m − n)m$ edges of weight 1. Fix a subset $\beta \in Q_{m−n,m}$ using the notation from Def. 2. A perfect matching in $G_B$ associated with $\beta$ corresponds to:

- A $n$-matching between $V_1$ and $V_2 \setminus \beta$ of weight $\text{per}(A[V_1, V_2 \setminus \beta])$

- A $(m − n)$-matching between $V_3$ and $\beta$ of weight $(m − n)!$

Then, $\text{per}(B)$ is the sum of all perfect matchings in $G_B$:

$$
\text{per}(B) = \sum_{\beta \in Q_{m−n,m}} (m − n)! \text{per}(A[V_1, V_2 \setminus \beta])
= (m − n)! \text{per}(A),
$$

where the last equality follows directly from Def. 2.

The proof is completed by combining the two reductions above to write the sum in (19) as:

$$
\sum_{k=0}^{\lfloor Y_d(x) \rfloor} \text{per}_k(Q) = \frac{1}{m!} \text{per} \left( \begin{bmatrix} Q & I_{\lfloor Y_d(x) \rfloor} \\ 1_{m,m} & 1_{m,\lfloor Y_d(x) \rfloor} \end{bmatrix} \right).
$$

Appendix D. Summary of the Semantic Observation Models

Table 4: No missed detections and no clutter: the likelihood $p(Z \mid Y_d(x), x)$ of a set of semantic observations $Z$ is shown for different combinations of $m := |Z|$ and $n := |Y_d(x)|$. The dependence of the likelihoods on $x$ is omitted for clarity.

<table>
<thead>
<tr>
<th>$n \neq m$</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 = $m = n$</td>
<td>1</td>
</tr>
<tr>
<td>0 &lt; $m = n$</td>
<td>[ \frac{1}{m!} \text{per} \begin{bmatrix} p_d(z_1 \mid y_1) &amp; \cdots &amp; p_d(z_m \mid y_1) \ \vdots &amp; \ddots &amp; \vdots \ p_d(z_1 \mid y_n) &amp; \cdots &amp; p_d(z_m \mid y_n) \end{bmatrix} ]</td>
</tr>
</tbody>
</table>

Table 5: No clutter but missed detections are possible: the likelihood $p(Z \mid Y_d(x), x)$ of a set of semantic observations $Z$ is shown for different combinations of $m := |Z|$ and $n := |Y_d(x)|$. The dependence of the likelihoods on $x$ is omitted for clarity.

<table>
<thead>
<tr>
<th>$n &lt; m$</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 = $m \leq n$</td>
<td>[ \prod_{i=1}^{n} (1 - p_d(y_i)) ]</td>
</tr>
<tr>
<td>0 &lt; $m \leq n$</td>
<td>[ \frac{1}{m!(m-n)!} \text{per} \begin{bmatrix} p_d(y_1)p_d(z_1 \mid y_1) &amp; \cdots &amp; p_d(y_1)p_d(z_m \mid y_1) &amp; 1 - p_d(y_1) &amp; \cdots &amp; 1 - p_d(y_1) \ \vdots &amp; \ddots &amp; \vdots &amp; \ddots &amp; \ddots &amp; \vdots \ p_d(y_n)p_d(z_1 \mid y_n) &amp; \cdots &amp; p_d(y_n)p_d(z_m \mid y_n) &amp; 1 - p_d(y_n) &amp; \cdots &amp; 1 - p_d(y_n) \end{bmatrix} ]</td>
</tr>
</tbody>
</table>
Both missed detections and clutter are possible: the likelihood \( p(Z \mid Y_d(x), x) \) of a set of semantic observations \( Z \) is shown for different combinations of \( m := |Z| \) and \( n := |Y_d(x)| \). The dependence of the likelihoods on \( x \) is omitted for clarity.

| \( m < n \) | \( e^{-\lambda} \frac{\lambda^m}{m!} \prod_{j=1}^{m} p_n(z_j) \) |
| \( 0 = n \leq m \) | \( 0 \) |
| \( 0 < n \leq m \) | \( e^{-\lambda} \frac{\lambda^m}{m!(m-n)!} \) \begin{pmatrix}
  \frac{p_d(y_1)p_e(z_1 \mid y_1)}{\lambda p_n(z_1)} & \cdots & \frac{p_d(y_1)p_e(z_m \mid y_1)}{\lambda p_n(z_m)} & 0 \\
  \vdots & & \vdots & \ddots \\
  \frac{p_d(y_n)p_e(z_1 \mid y_n)}{\lambda p_n(z_1)} & \cdots & \frac{p_d(y_n)p_e(z_m \mid y_n)}{\lambda p_n(z_m)} & 0
\end{pmatrix}

Both missed detections and clutter are possible: the likelihood \( p(Z \mid Y_d(x), x) \) of a set of semantic observations \( Z \) is shown for different combinations of \( m := |Z| \) and \( n := |Y_d(x)| \). The dependence of the likelihoods on \( x \) is omitted for clarity.

| \( n = 0 \) | \( e^{-\lambda} \frac{\lambda^m}{m!} \prod_{j=1}^{m} p_n(z_j) \) |
| \( m = 0 \) | \( \prod_{i=1}^{n} (1 - p_d(y_i)) \) |
| \( 0 < n \leq m \) | \( e^{-\lambda} \frac{\lambda^m}{m!(m-n)!} \) \begin{pmatrix}
  \frac{p_d(y_1)p_e(z_1 \mid y_1)}{\lambda p_n(z_1)} & \cdots & \frac{p_d(y_1)p_e(z_m \mid y_1)}{\lambda p_n(z_m)} & 1 - p_d(y_1) & 0 \\
  \vdots & & \vdots & \ddots & \ddots \\
  \frac{p_d(y_n)p_e(z_1 \mid y_n)}{\lambda p_n(z_1)} & \cdots & \frac{p_d(y_n)p_e(z_m \mid y_n)}{\lambda p_n(z_m)} & 0 & 1 - p_d(y_n)
\end{pmatrix}

| \( 0 < m \leq n \) | \( e^{-\lambda} \frac{\lambda^m}{m!(m-n)!} \) \begin{pmatrix}
  \frac{p_d(y_1)p_e(z_1 \mid y_1)}{\lambda p_n(z_1)} & \cdots & \frac{p_d(y_1)p_e(z_m \mid y_1)}{\lambda p_n(z_m)} & 1 - p_d(y_1) & \cdots & 1 - p_d(y_1) \\
  \vdots & & \vdots & \ddots & \ddots & \ddots \\
  \frac{p_d(y_n)p_e(z_1 \mid y_n)}{\lambda p_n(z_1)} & \cdots & \frac{p_d(y_n)p_e(z_m \mid y_n)}{\lambda p_n(z_m)} & 1 - p_d(y_n) & \cdots & 1 - p_d(y_n)
\end{pmatrix}

- \( \lambda \) is the rate of the Poisson distribution.
- \( p_n(z_j) \) is the probability of observing \( z_j \) in the current set.
- \( p_d(y_i) \) is the probability of detecting \( y_i \).
Appendix E. Active Bearing-only Localization

Finally, we present details about the bearing-only entropy minimization (BEM) method. It solves the active semantic localization problem in (12) but the pose entropy is conditioned only on the future bearing measurements:

\[
\sigma^* \in \arg \min_{\sigma \in \Sigma} \mathbb{H}(x_{0:T} \mid B_{1:T})
\]

subj. to \( x_{s+1} = f(x_s, \sigma_s, v_s), \ s = 0, \ldots, T - 1 \)

\[
B_s = \{ \beta(x_s, y) + \eta_s \mid y^p \in \text{FoV}(x_s) \}, \ s = 1, \ldots, T,
\]

where \( \eta_s \sim \mathcal{N}(0, \Sigma_{\beta}) \) is the bearing-measurement noise and \( B_s \) is the set of bearing measurements obtained at time \( s \). Since the robot motion model \( f(x, u, v) \) and the bearing measurement model \( \beta(x, y) \) are continuous functions of the robot state \( x \) and landmark states \( y \), perturbed by Gaussian noise, we can linearize them to simplify the above problem. In detail, let \( \bar{x}_s \) be the mean of the particle distribution at time \( s \) and define \( \delta x_s := x_s - \bar{x}_s \). We linearize the constraints in Problem (20) around the means \( \bar{x}_s \) and \( \beta(\bar{x}_s, y) \) to obtain:

\[
\sigma^* \in \arg \min_{\sigma \in \Sigma} \mathbb{H}(\delta x_{0:T}) - \mathbb{H}(\delta x_{0:T}; \delta B_{1:T})
\]

subj. to \( x_{s+1} = f(\bar{x}_s, \sigma_s, 0), \ s = 0, \ldots, T - 1 \)

\[
\delta x_s \approx \left[ \frac{\partial f}{\partial x}(\bar{x}_s, \sigma_s, 0) \right] \delta x_s + \left[ \frac{\partial f}{\partial v}(\bar{x}_s, \sigma_s, 0) \right] v_s
\]

\[
\delta B_s \approx \left\{ \frac{\partial \beta}{\partial x}(\bar{x}_s, y) \delta x_s + \eta_s \mid y^p \in \text{FoV}(x_s) \right\}, \ s = 1, \ldots, T.
\]

Let \( C_0 \) be the covariance of the prior particle distribution and assume that \( \delta x_0 \sim \mathcal{N}(0, C_0) \). Since the constraints in (21) are linear and the measurement noise is Gaussian, the distribution of \( \delta x_s \) remains Gaussian for \( s = 1, \ldots, T \). In particular, it can be computed via the Kalman filter. In addition, the entropy and mutual information of Gaussian random variables depend only on the associated covariance matrices (and not on the particular measurement realization \( \delta B_{1:T} \)) and can be computed in closed form. Thus, we compute the cost for each control sequence \( \sigma \in \Sigma \) and choose the sequence with the lowest cost. Refer to Atanasov et al. (2015) for more details regarding conditional entropy minimization via linearization and model predictive control.

References


