Semantic Localization Via the Matrix Permanent

Nikolay Atanasov, Menglong Zhu, Kostas Daniilidis, and George J. Pappas
GRASP Laboratory, University of Pennsylvania
Philadelphia, PA 19104, USA
{atanasov,menglong,kostas,pappasg}@seas.upenn.edu

***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 representations, we use random finite sets to represent the object detections. This allows us to explicitly incorporate missed detections, false alarms, and data association in the sensor model. 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 enables us to solve the semantic localization problem with a polynomial-time approximation to the set-based Bayes filter. The performance of our approach is demonstrated in simulation and in a real environment using a deformable-part-model-based object detector. Comparisons are made with the traditional lidar-based geometric Monte-Carlo localization.**

I. 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 and on precise metric maps, which store the geometric information. 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. This has several benefits. Localizing against semantically meaningful landmarks is less ambiguous and helps with global localization and loop-closure. Also, high-precision sensors such as laser range finders and 3-D lidars are not crucial for accurate localization and can be replaced by regular cameras. Finally, there is an abundance of maps for GPS-denied environments which are semantically annotated, perhaps even sketched by hand, and could be used for the task. Maps can also be constructed via the semantic mapping approaches which received significant attention in recent years [18, 12, 32, 29, 7].

Monte-Carlo localization based on geometric features was introduced by Dellaert et al. [10]. The knowledge about the robot's 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 [3, 25].

There is a line of work addressing visual localization, which matches observed image features to a database, whose images correspond to the nodes of a topological map, e.g. [40, 35, 2, 39, 24, 17]. Wang et al. [39] represent each location in a topological map by a set of interest points that can be reliably detected in images. A nearest neighbor search is used to match observed SIFT features to the database. Košecká and Li [17] 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 maximum likelihood data association is that when it is wrong it quickly causes the localization filter to diverge. Hesch et al. [13] study the effects of unobservable directions on the estimator consistency in vision-aided inertial navigation systems. As object recognition algorithms miss detections and produce false alarms, correct data association is crucial for semantic localization and semantic world modeling too [41].

Instead of the traditional vector-based representations, we use random finite sets (RFS) to represent the semantic information obtained from performing object recognition on the robot's observations. This allows us to explicitly incorporate missed detections, false alarms, and data association in the sensor model. In recent years, RFS-based solutions to SLAM have gained popularity due to their unified treatment of filtering and data association. Mahler [23] derived the Bayesian recursion with RFS-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. [15], Lee et al. [20], and Mullane et al. [27]. 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 and Jensfelt [6] address geometric localization by formulating hypotheses about the robot's state and tracking them with the PHD filter. Zhang et al. [43] propose an approach for visual odometry using a PHD filter to track SIFT

features extracted from observed images. None of these RFS-based approaches have been applied in a semantic setting and all rely on a first-moment approximation via the PHD filter. In addition to modeling semantic information, we carry out filtering with the full RFS observation model. Very few works deal with the full model [9, 22, 36] and none have applied it to semantic localization or studied its computational complexity.

There are several related semantic localization approaches which do not rely on an RFS model and do not explicitly handle data association problems. Anati et al. [1] match histogram of gradient energies and quantized colors features to expected features from the prior semantic map. Yi et al. [42, 16] use semantic descriptions of distance and bearing in a contextual map for active semantic localization. Bao and Savarese [4] propose a maximum likelihood estimation formulation for Semantic Structure from Motion. In addition to recovering the camera parameters (motion) and the 3-D location 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 us to incorporate missed detections and data association with the landmarks on the map in the sensor model.
- We prove that obtaining the likelihood of a RFS-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.

Connections between the matrix permanent and data association have been identified in the target tracking community [30, 8, 31, 26], [21, Ch.11] but this is the first connection with the random-finite-set observation model.

Paper Organization: In Sec. II we formulate the semantic localization problem precisely. In Sec. III we provide a probabilistic model, which quantifies the likelihood of a random finite set of object detections and captures false positives, missed detections, and unknown data association. The key relationship between filtering with the RFS observation model and the matrix permanent is derived in Sec. IV. Finally, in Sec. V, we present results from simulations and real-world experiments and discuss the performance of our approach.

II. Problem Formulation

Consider a mobile robot, whose dynamics are governed by the *motion model* $x_{t+1} = f(x_t, u_t, w_t)$, where $x_t := (x_t^p, x_t^r, x_t^a)$ is the robot state, containing its position x_t^p, orientation x_t^r, and other variables x_t^a such as velocity and acceleration, u_t is the control input, and w_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:

$$p_f(\cdot \mid x_t, u_t). \tag{1}$$

The robot has access to a semantic map of the environment containing n objects with known poses and classes. Let the set $Y = \{y_1, \ldots, y_n\}$ represent the map, where $y_i := (y_i^p, y_i^r, y_i^c)$ consists of the position y_i^p, orientation y_i^r, and class y_i^c of the i-th object. Depending on the application, the object state y_i may capture other observable properties of interest.

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 $\mathcal{C}$ present in Y. If some object $y \in Y$ is visible and detected from the current robot pose x_t, then the algorithm returns a detection z_t. In the remainder, we assume that a detection, $z_t := (c_t, s_t, b_t)$, consists of a detected class $c_t \in \mathcal{C}$, a detection score $s_t \in \mathcal{S}$, and an estimate $b_t \in \mathcal{B}$ of the bearing from the sensor to the detected object, where $\mathcal{S}$ is the range of possible scores and $\mathcal{B}$ is the range of bearings, usually specified by the sensor's field of view (e.g. a camera with $\mathcal{B} = [-47°, 47°]$ was used in our experiments). Depending on the sensors and the visual processing, z_t could also contain a 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. Due to false alarms and misses, a randomly-sized collection of detections is returned by the object recognition algorithm at time t and is best represented by a random finite set Z_t. For any t, denote the pdf of robot state x_t conditioned on the map Y, the past detections $Z_{0:t}$, and the control history $u_{0:t-1}$ by $p_{t|t}$ and that of $x_t \mid Y, Z_{0:t}, u_{0:t}$ by $p_{t+1|t}$.

Problem (Semantic Localization)**.** *Suppose that the control u_t is applied to the robot at time $t \geq 0$ and, after moving, the robot obtains a random finite set Z_{t+1} of detections. 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.*

It is natural to approach the semantic localization problem using recursive Bayesian estimation. This, however, requires a probabilistic model of the semantic observations, which quantifies the likelihood of the random set Z_{t+1} of detections conditioned on the set of objects Y and the robot state x_{t+1}.

III. Semantic Observation Model

A. Observation Model for a Single Object Detection

We begin by constructing a probabilistic model of the semantic observations obtained from a single object in the environment. It consists of two ingredients: a *detection model* and an *observation 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 from the robot's sensor to the object y in the sensor frame[1]. Let the field of view of the sensor[2] be described by the set $FoV(x)$. Objects outside

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

[2]The field of view of a camera in 2-D, assuming its frame coincides with the robot's, can be represented by $\{w \in \mathbb{R}^2 \mid \|x^p - w\|_2 \leq r_d, \beta(x, w) \leq \alpha_d\}$, where α_d is the angle of view and r_d is the maximum range at which an object can be detected.

the field of view cannot be detected. For the ones within, we use a distance-decaying probability of detection:

$$p_d(y \mid x) = \begin{cases} p_{d,0} e^{-\|y^p - x^p\|_2 / \sigma_d^2} & \text{if } y^p \in FoV(x), \\ 0 & \text{else,} \end{cases} \quad (2)$$

where $p_{d,0}$ and σ_d^2 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 is also possible.

Supposing that an object $y \in Y$ is detected, the observation model quantifies the likelihood of the resulting detection $z = (c, s, b)$ conditioned on the true object state y and the robot state x. Assuming that conditioned on y, the bearing measurement b is independent of the detected class c and the detection score s, it is appropriate to model its conditional pdf $p_\beta(\cdot \mid y, x)$ as that of a Gaussian distribution with mean $\beta(x, y)$ and covariance Σ_β. 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 detected class and score are independent of the robot state x. Then, the observation model of the semantic measurement z can be decomposed as:

$$p_z(z \mid y, x) = p_c(c \mid y^c) p_s(s \mid c, y^c) p_\beta(b \mid y, x), \quad (3)$$

where $p_c(c \mid y^c)$ is the confusion matrix of the object detector and $p_s(s \mid c, y^c)$ is the detection score likelihood. The latter can be learned for example by recording the detection scores from the detected positive examples in a training set and using kernel density estimation (see Fig. 7). Finally, a model of the pdf, $\kappa(z)$, of a false positive detection generated by clutter is needed. While it can also be learned from data, it is realistic to assume that clutter detections are uniformly distributed and independent of the robot's state:

$$\kappa(z) = \frac{1}{|\mathcal{C}|} \frac{1}{|\mathcal{S}|} \frac{1}{|\mathcal{B}|}. \quad (4)$$

B. Observation Model for a Random Number of Detections

In this section we use Mahler's finite set statistics [23] to model the pdf of a random finite set $Z = \{z_1, \ldots, z_m\}$ of object detections. The following assumptions are necessary:

- (A1) No detection is generated by more than one object
- (A2) An object $y \in Y$ generates either a single detection with probability $p_d(y \mid x)$ or a missed detection with probability $1 - p_d(y \mid x)$
- (A3) The clutter process is Poisson-distributed in time with expected value λ and distributed in space according to the pdf $\kappa(z)$ in (4)
- (A4) The clutter process and the object-detection process are statistically independent and all detections are conditionally independent given the object states
- (A5) Any two detections in Z are independent conditioned on the map Y and the robot state x

Let $Y_d(x) := \{y \in Y \mid p_d(y \mid x) > 0\}$ be the set of detectable objects given a robot state x. We specify the pdf of Z for a series of increasingly more complex cases.

1) All measurements are clutter: If there are no objects in proximity to the sensor, *i.e.*, $Y_d(x) = \emptyset$, then any generated detections would be from clutter. The correct observation model in this case is due to the Poisson clutter process:

$$p(Z \mid \emptyset, x) = e^{-\lambda} \left(\prod_{z \in Z} \lambda \kappa(z) \right). \quad (5)$$

This integrates to 1 if the set integral definition in [23, Ch.11.3.3] is used.

2) No missed detections and no clutter: This is the case of "perfect vision", when every detectable object generates a detection, i.e. $p_d(y \mid x) \equiv 1$ for any $y \in FoV(x)$, and no detections arise in any other way, i.e. $\lambda = 0$. If the number of detections m is not equal to the cardinality $|Y_d(x)|$ of the set of detectable objects, then $p(Z \mid Y_d(x), x) = 0$ and otherwise:

$$p(Z \mid Y_d(x), x) = \sum_\pi \prod_{i=1}^{m} p_z(z_{\pi(i)} \mid y_i, x), \quad (6)$$

where the sum is over all permutations π of the set $\{1, \ldots, m\}$ and $\{y_1, \ldots, y_m\}$ is an enumeration of $Y_d(x)$. For a derivation see [23, Ch.12.3]. In general, it is not clear which of the detectable objects on the map produced which of the detections. A permutation π specifies a particular correspondence between the m detectable objects and the m received detections. Intuitively, all associations are plausible and we can think of (6) as quantifying the likelihood of Z by averaging the likelihoods of the individual detections over all data associations.

3) No clutter but missed detections are possible: If $m > |Y_d(x)|$, then $p(Z \mid Y_d(x), x) = 0$. If $m = 0$, then:

$$p(\emptyset \mid Y_d(x), x) = \prod_{i=1}^{|Y_d(x)|} (1 - p_d(y_i \mid x)) \quad (7)$$

and if $m \in \{1, \ldots, |Y_d(x)|\}$, then

$$p(Z \mid Y_d(x), x) = p(\emptyset \mid Y_d(x), x) \sum_\pi \prod_{i \mid \pi(i) > 0} \frac{p_d(y_i \mid x) p_z(z_{\pi(i)} \mid y_i, x)}{(1 - p_d(y_i \mid x))},$$

where the sum is over all functions $\pi : \{1, \ldots, |Y_d(x)|\} \to \{0, 1, \ldots, m\}$ with the property: $\pi(i) = \pi(i') > 0 \Rightarrow i = i'$, which ensures that (A1) is satisfied. The index '0' in the range of π represents the case when a detectable object was missed. For example, it allows for the possibility that all detectable objects are missed (associated with '0') and all m detections are due to clutter.

4) No missed detections but clutter is possible: If $m < |Y_d(x)|$, then $p(Z \mid Y_d(x), x) = 0$; otherwise

$$p(Z \mid Y_d(x), x) = p(Z \mid \emptyset, x) \sum_\pi \prod_{i=1}^{|Y_d(x)|} \frac{p_z(z_{\pi(i)} \mid y_i, x)}{\lambda \kappa(z_{\pi(i)})},$$

where the summation is over all one-to-one functions $\pi : \{1, \ldots, |Y_d(x)|\} \to \{1, \ldots, m\}$.

5) Both missed detections and clutter are possible: This is the most general model and captures all artifacts of object recognition: missed detections, false positives, and unknown data association. If $|Y_d(x)| = 0$, then the pdf is given by (5). If $m = 0$, then the pdf is given by (7). Otherwise:

$$p(Z \mid Y_d(x), x) = p(Z \mid \emptyset, x)p(\emptyset \mid Y_d(x), x) \times \sum_{\pi} \prod_{i|\pi(i)>0} \frac{p_d(y_i \mid x)p_z(z_{\pi(i)} \mid y_i, x)}{(1 - p_d(y_i \mid x))\lambda\kappa(z_{\pi(i)})}, \tag{8}$$

where the sum is over all functions $\pi : \{1, \ldots, |Y_d(x)|\} \to \{0, 1, \ldots, m\}$ with the property: $\pi(i) = \pi(i') > 0 \Rightarrow i = i'$.

Having derived a general observation model for a random number of object detections, we can now state the Bayesian filtering equations needed for semantic localization.

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

$$\textit{Predict:} \quad p_{t+1|t}(x) = \int p_f(x \mid x', u_t)p_{t|t}(x')dx' \tag{9}$$

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

where $p(Z_{t+1} \mid Y_d(x), x)$ is the random finite set observation model in (8) and η_{t+1} is a normalization constant.

IV. 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 [37, Ch.4] is an approximation to the Bayes filter with vector-valued observations, which has been very successful for geometric localization. Since the robot state is still vector-valued, we represent its pdf $p_{t|t}$ 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 (9) with the motion model p_f as a proposal distribution, is summarized in Alg. 1. 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 (8). In particular, it is not apparent how to efficiently compute the sum over all associations π. To gain intuition we begin with the simpler case of "perfect vision" in (6).

Fix a robot state x and consider the non-trivial case when the received measurements Z and the detectable landmarks $Y_d(x)$ are of the same cardinality m. 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 $w_e := p_z(z \mid y, x)$ with every edge $e := (z, y) \in E$ and consider the weighted bipartite graph $G := (V_1, V_2, E, w)$. The permutations π in (6) correspond to different associations

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, semantic map Y, 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 $p_f(\cdot \mid x^k_{t|t}, u_t)$
5: $w^k_{t+1|t} \leftarrow w^k_{t|t}$
6: *Update*: $w^k_{t+1|t+1} \leftarrow p(Z_{t+1} \mid Y_d(x^k_{t+1|t}), x^k_{t+1|t})w^k_{t+1|t}$
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

between the objects V_1 and the measurements V_2 or in other words to perfect matchings[3] in G. Given a perfect matching π, its associated product term inside the sum in (6) corresponds to its weight. Then, the sum over all π corresponds to the sum of the weights of all perfect matchings in G, which notably is equivalent 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:*

$$\mathbf{per}(A) := \sum_{\pi} \prod_{i=1}^{n} A(i, \pi(i)),$$

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

It is now clear that the detection likelihood in the case of no false positives and no missed detections can be obtained by computing the permanent of a matrix.

Proposition 2. *The likelihood in (6) of a random finite set of detections $Z = \{z_1, \ldots, z_m\}$, in the case of no false positives and no missed detections, with $|Y_d(x)| = m$ satisfies:*

$$p(Z \mid Y_d(x), x) = \mathbf{per}(P),$$

where P is a $m \times m$ matrix with $P(i, j) := p_z(z_j \mid y_i, x)$.

The general case in (8), 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 the Appendix.

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

$$p(Z \mid Y_d(x), x) = e^{-\lambda}\left(\prod_{z \in Z} \lambda\kappa(z)\right) \prod_{y \in Y_d(x)} (1 - p_d(y \mid x)) \times \frac{1}{m!}\,\mathbf{per}\left(\begin{bmatrix} Q & I_{|Y_d(x)|} \\ 1_{m,m} & 1_{m,|Y_d(x)|} \end{bmatrix}\right), \tag{10}$$

where λ is the expected number of clutter detections, $\kappa(\cdot)$ is the spatial pdf of the clutter, $p_d(y \mid x)$ is the probability of

[3] 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.

detecting object $y \in Y_d(x)$, $1_{n,m}$ is a $n \times m$ matrix of all ones, and Q is a matrix with elements:

$$Q(i,j) := \frac{p_d(y_i \mid x) p_z(z_j \mid y_i, x)}{(1 - p_d(y_i \mid x))\lambda\kappa(z_j)}, \qquad \begin{matrix} i = 1, \ldots, |Y_d(x)|, \\ j = 1, \ldots, m, \end{matrix}$$

where without loss of generality it is assumed that $|Y_d(x)| \leq m$; otherwise re-label the sets Z and Y_d.

Theorem 1 maps the problem of determining the pdf of Z in the general case in (8) to the problem of finding the permanent of a $(m+|Y_d(x)|)\times(m+|Y_d(x)|)$ square matrix. The problem is still computationally challenging because computing the permanent of a matrix is #P-complete[4] [38]. However, the main advantage of Theorem 1 is that it allows us to leverage the extensive literature on approximation algorithms for computing the matrix permanent.

An exact method for computing the permanent of a $n \times n$ matrix, proposed by Ryser [34] and later improved by Nijenhuis and Wilf [28, Ch.23], is summarized in Alg. 2. Its time complexity is $\Theta(n2^{n-1})$. The dimension of the matrix in (10) 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 practice 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 et al. [14] show that for any $\epsilon \in (0, 1]$ and $\delta > 0$, there exists a randomized algorithm whose output comes within a factor $(1 \pm \epsilon)$ of $\mathbf{per}(A)$ with probability at least $1 - \delta$ with a random running time T such that $\mathbb{E}(T) = O(n^{10}(\log n)^3)$. The running time was later improved by Bezáková et al. [5] to $O(n^7(\log n)^4)$. Also, when $A \in [0, 1]^{n\times n}$ is a matrix such that all row and column sums are at least γn for $\gamma \in (0.6, 1]$, Law [19, Ch.2.2] provides an algorithm with expected running time $O(n^4(\log n + \epsilon^{-2} \log \delta^{-1}))$.

Proposition 3. *Given m object detections and a semantic map with n objects, the time complexity of Alg. 1 for semantic localization with N particles is $O(N(m + n)2^{(m+n)})$ if the measurement update is computed exactly with Alg. 2 and $O(N(n+m)^7(\log(m+n))^4)$ if computed approximately with the method of Bezáková et al. [5].*

A final note on the computation of (10) is that the scaling of the numbers can be improved by using that for a matrix $A \in \mathbb{R}^{n\times m}$ and a constant γ, $\mathbf{per}(\gamma A) = \gamma^{\min(n,m)} \mathbf{per}(A)$. In particular, some of the terms $p(Z \mid \emptyset, x)$, $p(\emptyset \mid Y_d(x), x)$, or $1/m!$ can be included within the permanent calculation.

V. Performance Evaluation

A. 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). In all experiments, the semantic localization was achieved using Alg. 1 with measurement updates obtained with the exact permanent algorithm (Alg. 2). Only the RGB images were used for the semantic measurement updates. The depth was not used, while the lidar was used to provide ground truth poses in the real-world experiments via geometric Monte-Carlo localization. The performance of our approach is demonstrated for *global localization*, which means that the robot has absolutely no information about its starting pose.

[4]A #P-complete problem is equivalent to computing the number of accepting paths of a polynomial-time nondeterministic Turing machine and #P contains NP.

Algorithm 2 Permanent (Nijenhuis and Wilf [28, Ch.23])

Input: $n \times n$ matrix A **Output**: $per(A)$
for $i = 1, \ldots, n$ **do**
  $x(i) \leftarrow A(i, n) - \frac{1}{2}\sum_{j=1}^{n} A(i, j)$
$s \leftarrow -1$, $g \leftarrow$ **false**$(n, 1)$, $p \leftarrow s\prod_{i=1}^{n} x(i)$
for $k = 2, \ldots, 2^{n-1}$ **do**
  if k is even **then** $j \leftarrow 1$ ▷ Obtain next gray code subset
  else { $j \leftarrow 2$
    while g_{j-1} is false **do**
      $j \leftarrow j + 1$ }
  $s \leftarrow -s$, $z \leftarrow 1 - 2g_j$, $g_j \leftarrow$ **not** g_j
  for $i = 1, \ldots, n$ **do**
    $x(i) \leftarrow x(i) + zA(i, j)$
  $p \leftarrow p + s\prod_{i=1}^{n} x(i)$
return $2(-1)^n p$

B. Observation Model

The state-of-the-art performance in single-image object detection is obtained by star-structured models such as deformable part models (DPM) [11]. Deformable part models were constructed for two object classes: $\mathcal{C} := \{\text{door, chair}\}$ (Fig. 2). A DPM-based detector was used to process the RGB images obtained by the robot as follows. 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. Detectors for the different classes in $\mathcal{C}$ are applied sequentially to the image, in a sliding-window fashion, and output detection scores at each pixel and scale of the pyramid. Detection scores above a certain threshold are returned along with bounding box and bearing information. The collection of all such detections at time t forms the random finite set Z_t. The detection model $p_d(y \mid x)$ and the observation model $p_z(z \mid y, x)$ were obtained from training data as discussed in Sec. III-A. The angle of view of the wide-angle lens was $94°$, the detection range - 10 meters, and the following constants were learned: $p_{d,0} = 0.92, \sigma_d = 4.53, \Sigma_\beta = 4°$. 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. 7.

C. Simulation Results

The performance of the semantic localization algorithm was evaluated in a simulated environment of size 25×25 m^2,

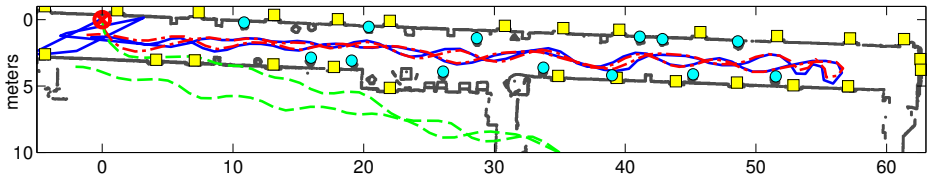

Fig. 1: Trajectories estimated by lidar-based geometric localization (red), image-based semantic localization (blue), and odometry (green) from a real experiment. The starting position, the door locations, and the chair locations are denoted by the red cross, the yellow squares, and the blue circles, respectively. See the attached video or http://www.seas.upenn.edu/~atanasov/vid/RSS14_SemanticLocalization.mp4 for more details.

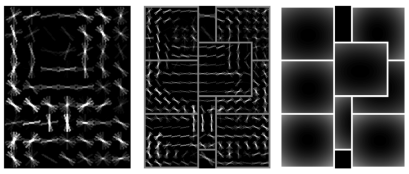

Fig. 2: A component of the deformable part model of a chair

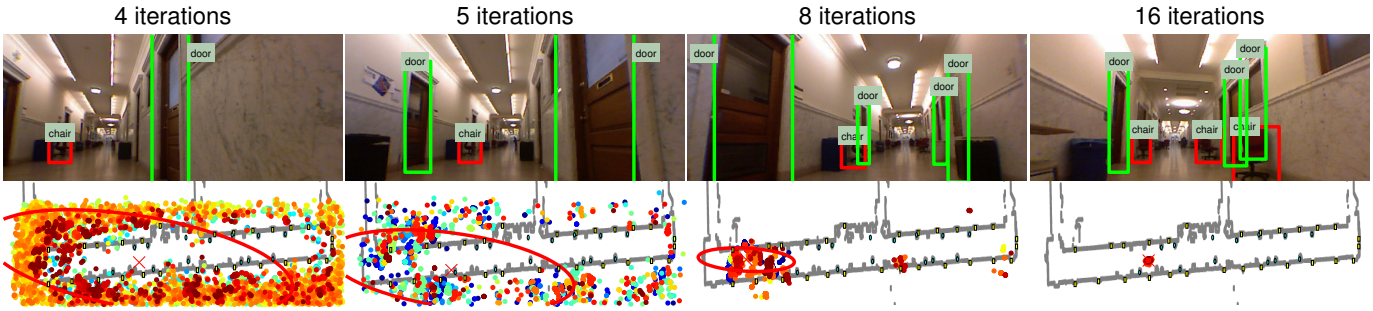

Fig. 3: Particle filter evolution (bottom) and object detections (top) during a real semantic localization experiment

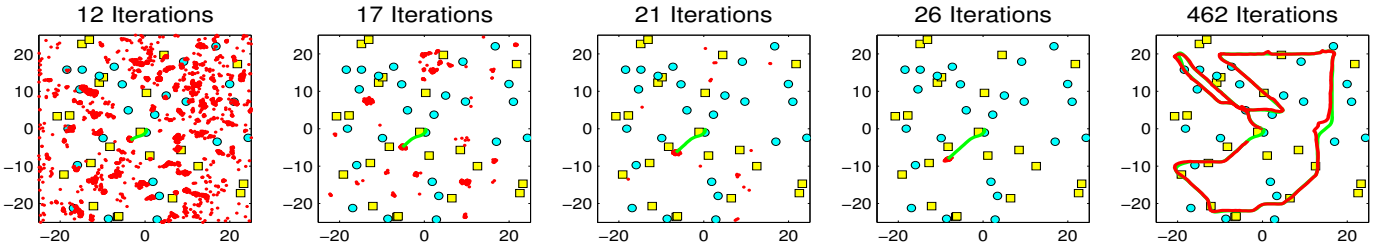

Fig. 4: A simulated environment with 45 objects from two classes (yellow squares, blue circles). The plots show the evolution of the particles (red dots), the ground truth trajectory (green), and the estimated trajectory (red). The expected number of clutter detections was set to $\lambda = 2$.

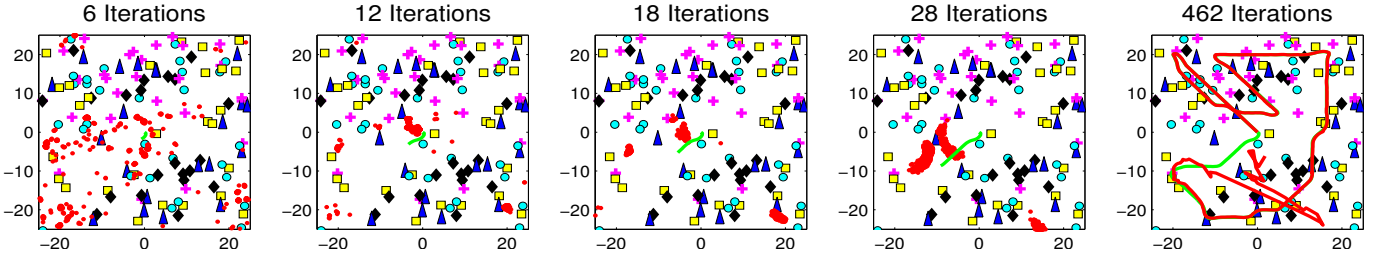

Fig. 5: A simulated environment with 150 objects from 5 classes (circles, squares, triangles, crosses, and diamonds) in a $25 \times 25\ m^2$ area. The plots show the particles (red dots), the ground truth trajectory (green), and the estimated trajectory (red) for clutter rate $\lambda = 4$.

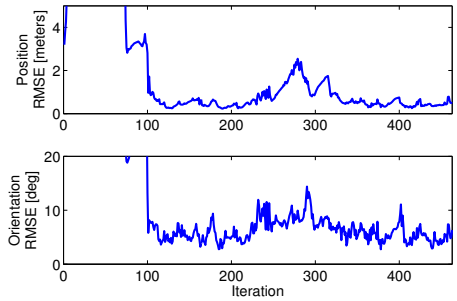

Fig. 6: Root mean squared error (RMSE) in the pose estimates obtained from the semantic localization algorithm after 50 simulated runs of the scenario in Fig. 4

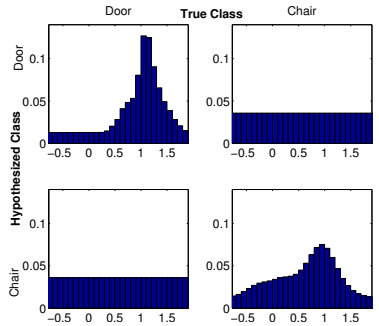

Fig. 7: Detection score likelihoods learned via kernel density estimation

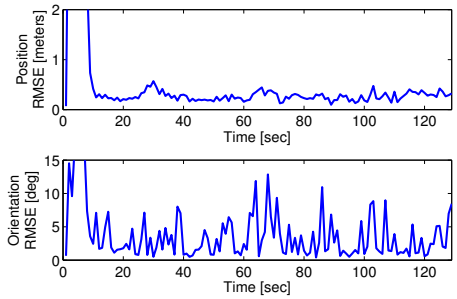

Fig. 8: Root mean squared error (RMSE) between the pose estimates from semantic localization and from lidar-based geometric localization obtained from four real experiments

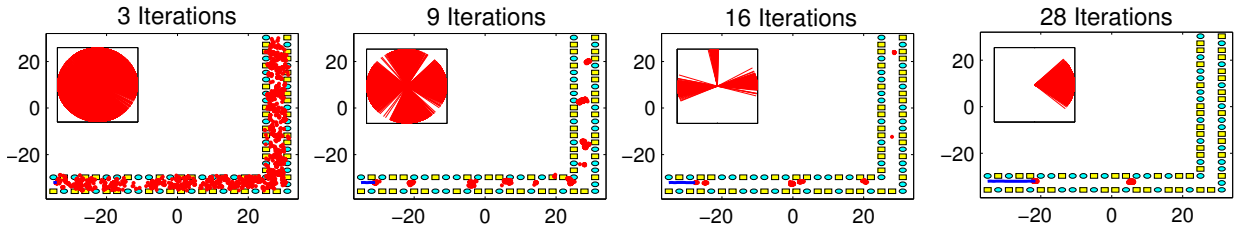

Fig. 9: A simulated example of semantic localization in the presence of severe perceptual aliasing. The ground truth trajectory (blue) and the evolution of the particle positions (red points) and orientations (red lines, top left) are shown.

populated by objects with randomly-chosen positions and classes (see Fig. 4). The error in the estimates averaged over 50 repetitions with different randomly-generated scenes is presented in Fig. 6. Since the localization starts with a uniform prior over the whole environment the error is large in the initial iterations. Multiple hypotheses are present until the robot obtains enough detections to uniquely localize itself. The performance in a challenging scenario with a lot of ambiguity is presented in Fig. 9. The reason for using only two classes is to increase the ambiguity in the data association. Our approach can certainly handle more classes and a higher object density. While the complexity in Prop. 3 is in terms of the total number of objects in the environment, the filter update times actually scale with the density of the detectable objects. Fig. 5 shows a simulation with clutter rate $\lambda = 4$ and 150 objects from 5 classes in a $25 \times 25\ m^2$ area. Scenarios with such high object density necessitate the use of an approximate, rather than the exact, permanent algorithm for real-time operation.

D. Experimental Results

In the 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 amcl package in ROS [33] and the results were used as the ground truth. The lidar and semantic estimates of the robot trajectory are shown in Fig. 1. The error between the two, averaged over the 4 runs, is presented in Fig. 8. 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, the robot is able achieve average errors in the position and orientation estimates of less than 35 cm and 10°, respectively. The particle filter evolution is illustrated in Fig. 3 along with some object detections.

In our experiments, it was sufficient to capture only class and position information in the object state because the orientation and appearance variations were handled well by the DPM. We emphasize that our model can incorporate richer object representations by extending the state y and training an appropriate observation model. This is likely to make the data association more unimodal. As permanent approximation methods rely on Monte-Carlo sampling from the data associations, fewer samples can be used in this case to speed up the computations. Our reduction to the permanent incorporates this naturally and leverages state of the art algorithms.

TABLE I: Comparison of maximum likelihood data association (MLD) and our random finite set approach (RFS) on the 4 real datasets (Fig. 1) and the simulations in Fig. 4 and Fig. 9. 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[5](UT), averaged over time, are presented. The first MLD(G) column uses the same number of particles as RFS(G), while the second uses a large number in an attempt to improve the performance.

Fig. 1	MLD(L)	MLD(G)	MLD(G)	RFS(L)	RFS(G)
NP [K]	0.50	3.00	40.0	0.50	3.00
PE [m]	0.26	22.9	0.31	0.26	0.26
OE [deg]	2.54	107	2.75	2.67	2.69
UT [sec]	0.023	0.060	0.600	0.065	0.320
Fig. 4					
NP [K]	0.50	5.00	100	0.50	5.00
PE [m]	15.3	24.9	17.3	0.32	0.72
OE [deg]	67.0	68.8	72.8	4.58	9.17
UT [sec]	0.012	0.062	1.100	0.042	0.400
Fig. 9					
NP [K]	0.50	24.0	100	0.50	24.0
PE [m]	0.27	48.8	26.9	0.11	2.35
OE [deg]	3.68	112	74.9	2.08	4.05
UT [sec]	0.027	0.760	3.340	0.062	2.620

E. Comparison with Maximum Likelihood Data Association

We compared our random finite set (RFS) approach to the more traditional maximum likelihood data association (MLD) approach used in FastSLAM [25]. MLD is based on Alg. 1 but the set of detections on line 6 is processed sequentially. For each individual detection z, each particle x with weight w determines the most likely data association: $q := \max_{y \in Y_d(x)} p_z(z \mid y, x)$ and updates its weight: $w' = qw$. The performance is presented in Table I 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 it was uniformly distributed over the environment. MLD(L) performs as well as RFS(L) in the real experiments and in Fig. 9. In Fig. 4, the data association is highly multimodal and MLD(L) does not converge even with 15K particles. This is reinforced in the global initialization cases. While RFS(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. 4 and Fig. 9, MLD(G) does not converge even with 100K particles. We conclude that once the particles have converged correctly MLD performs as well as RFS. However, with global initialization or ambiguous data association MLD

[5]The reported times are from a MATLAB implementation on a PC with i7 CPU@2.3GHz and 16GB RAM

makes mistakes and can never recover while RFS is robust with a small number of particles.

VI. Conclusion

Modeling the semantic information obtained from object detection with random finite sets enabled a unified treatment of filtering, data association, missed detections, and false positives. The efficient implementation of the set-based Bayes filter depends critically on the connection between the matrix permanent and the RFS observation model. 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 offers superior performance in cases of global localization, loop closure, and perceptual aliasing. The real experiments demonstrated that the accuracy of the semantic localization method is comparable with the laser-based geometric approaches. Future work will focus on extensions to semantic SLAM and active semantic localization.

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.

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 π in (8) specify different associations between the objects V_1 and the detections V_2. The introduction of missed detections ('0' in the range of π) means that some detectable objects need not to be assigned to a detection in Z. As any object could be missed, the associations π correspond to matchings in the graph G. Given a matching π, the associated product term inside the sum in (8) corresponds to the weight of π. Then, the sum over all π corresponds to the sum of the weights of all matchings in G. The sum of the weights of all matchings with k edges can be computed via the k-th subpermanent sum of the adjacency matrix Q of G.

Definition 2 (Subpermanent Sum)**.** *Let A be an $n \times m$ non-negative matrix with $n \leq m$ and let $Q_{k,n}$ be the set of all subsets of cardinality k of $1, \ldots, n$. For $\alpha \in Q_{k,n}$ and $\beta \in Q_{k,m}$ let $A[\alpha, \beta] := [A(\alpha_i, \beta_j)]_{i,j=1}^{k}$ be the corresponding k-by-k submatrix of A. Define $per_0(A) := 1$ and*

$$\mathbf{per}_k(A) := \sum_{\alpha \in Q_{k,n}, \beta \in Q_{k,m}} \mathbf{per}(A[\alpha, \beta]), \quad k = 1, \ldots, n$$

The sum in (8) is then equal to the sum over all k-matchings:

$$\sum_{\pi} \prod_{i|\pi(i)>0} \frac{p_d(y_i \mid x) p_z(z_{\pi(i)} \mid y_i, x)}{(1 - p_d(y_i \mid x)) \lambda \kappa(z_{\pi(i)})} = \sum_{k=0}^{|Y_d(x)|} \mathbf{per}_k(Q), \quad (11)$$

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 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,*

$$\sum_{k=0}^{n} \mathbf{per}_k(A_{n,m}) = \mathbf{per}(\begin{bmatrix} A_{n,m} & I_n \end{bmatrix}).$$

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 corresponding 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 α and β corresponds to:

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

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

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

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,*

$$\mathbf{per}(A_{n,m}) = \frac{1}{(m-n)!} \mathbf{per}\left(\begin{bmatrix} A_{n,m} \\ 1_{m-n,m} \end{bmatrix}\right)$$

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 $B := \begin{bmatrix} A_{n,m}^T & 1_{m-n,m}^T \end{bmatrix}^T$ 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 β corresponds to:

- A n-matching between V_1 and $V_2 \setminus \beta$ of weight $\mathbf{per}(A[V_1, V_2 \setminus \beta])$
- A $(m-n)$-matching between V_3 and β of weight $(m-n)!$

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

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

where the last equality follows directly from Def. 2. ■

The proof is completed by combining the two reductions above to write the sum in (11) as:

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

References

[1] R. Anati, D. Scaramuzza, K. Derpanis, and K. Daniilidis. Robot Localization Using Soft Object Detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4992–4999, 2012.

[2] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat. Visual topological SLAM and global localization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[3] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, The University of Sydney, 2002.

[4] S. Bao and S. Savarese. Semantic Structure from Motion. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[5] I. Bezáková, D. Štefankovič, V. Vazirani, and E. Vigoda. Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 900–907, 2006.

[6] A. Bishop and P. Jensfelt. Global Robot Localization with Random Finite Set Statistics. In *Int. Conf. on Information Fusion*, 2010.

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

[8] J. Collins and J. Uhlmann. Efficient Gating in Data Association with Multivariate Gaussian Distributed States. *IEEE Trans. on Aerospace and Electronic Systems*, 28 (3):909–916, 1992.

[9] P. Dames, D. Thakur, M. Schwager, and V. Kumar. Playing Fetch with Your Robot: The Ability of Robots to Locate and Interact with Objects. *IEEE Robotics and Automation Magazine*, 21(2):46–52, 2013.

[10] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo Localization for Mobile Robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, 1999.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010.

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

[13] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. Towards Consistent Vision-Aided Inertial Navigation. In *Algorithmic Foundations of Robotics X*, volume 86 of *Springer Tracts in Advanced Robotics*. 2013.

[14] M. Jerrum, A. Sinclair, and E. Vigoda. A Polynomial-time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *Journal of the ACM*, 51(4):671–697, 2004.

[15] B. Kalyan, K. Lee, and W. Wijesoma. FISST-SLAM: Finite Set Statistical Approach to Simultaneous Localization and Mapping. *The International Journal of Robotics Research*, 29(10):1251–1262, 2010.

[16] D. W. Ko, C. Yi, and I. H. Suh. Semantic Mapping and Navigation: A Bayesian Approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2630–2636, 2013.

[17] J. Košecká and F. Li. Vision Based Topological Markov Localization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1481–1486, 2004.

[18] I. Kostavelis and A. Gasteratos. Learning Spatially Semantic Representations for Cognitive Robot Navigation. *Robotics and Autonomous Systems*, 61(12):1460 – 1475, 2013.

[19] W. Law. *Approximately Counting Perfect and General Matchings in Bipartite and General Graphs*. PhD thesis, Duke University, 2009.

[20] C. Lee, D. Clark, and J. Salvi. SLAM With Dynamic Targets via Single-Cluster PHD Filtering. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):543–552, 2013.

[21] M. Liggins, D. Hall, and J. Llinas. *Handbook of Multisensor Data Fusion*. Taylor & Francis, 2008.

[22] W.-K. Ma, B.-N. Vo, S. Singh, and A. Baddeley. Tracking an Unknown Time-Varying Number of Speakers Using TDOA Measurements: A Random Finite Set Approach. *IEEE Trans. on Signal Processing*, 54(9):3291–3304, 2006.

[23] R. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[24] G. Mariottini and S. Roumeliotis. Active Vision-based Robot Localization and Navigation in a Visual Memory. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[25] M. Montemerlo and S. Thrun. Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1985–1991, 2003.

[26] M. Morelande. Joint Data Association Using Importance Sampling. In *Int. Conf. on Information Fusion*, pages 292–299, 2009.

[27] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo. *Random Finite Sets for Robot Mapping & SLAM*. Springer Tracts in Advanced Robotics. Springer, 2011.

[28] A. Nijenhuis and H. Wilf. *Combinatorial Algorithms*. Academic Press, 1978.

[29] A. Nüchter and J. Hertzberg. Towards Semantic Maps for Mobile Robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[30] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Trans. on Automatic Control*, 54(3):481–497, 2009.

[31] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking Many Objects with Many Sensors. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, volume 2, pages 1160–1167, 1999.

[32] A. Pronobis. *Semantic Mapping with Mobile Robots*.

PhD thesis, KTH Royal Institute of Technology, 2011.

[33] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *Proc. Open-Source Software Workshop Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[34] H. Ryser. *Combinatorial Mathematics*. Carus Mathematical Monographs # 14. Mathematical Association of America, 1963.

[35] S. Se, D. Lowe, and J. Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[36] H. Sidenbladh and S.-L. Wirkander. Tracking Random Sets of Vehicles in Terrain. In *Computer Vision and Pattern Recognition Workshop*, volume 9, pages 98–98, June 2003.

[37] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press Cambridge, 2005.

[38] L. Valiant. The Complexity of Computing the Permanent. *Theoretical Computer Science*, 8(2):189–1201, 1979.

[39] J. Wang, H. Zha, and R. Cipolla. Coarse-to-Fine Vision-Based Localization by Indexing Scale-Invariant Features. *IEEE Trans. on Systems, Man, and Cybernetics*, 36(2): 413–422, 2006.

[40] J. Wolf, W. Burgard, and H. Burkhardt. Robust Vision-based Localization by Combining an Image Retrieval System with Monte Carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.

[41] L. Wong, L. Kaelbling, and T. Lozano-Pérez. Data Assocation for Semantic World Modeling from Partial Views. In *International Symposium on Robotics Research (ISRR)*, 2013.

[42] C. Yi, I. H. Suh, G. H. Lim, and B.-U. Choi. Active-Semantic Localization with a Single Consumer-Grade Camera. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 2161–2166, 2009.

[43] F. Zhang, H. Stahle, A. Gaschler, C. Buckl, and A. Knoll. Single Camera Visual Odometry Based on Random Finite Set Statistics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 559–566, 2012.