2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

SLAM with Objects using a Nonparametric Pose Graph

Beipeng Mu[1], Shih-Yuan Liu[1], Liam Paull[2], John Leonard[2], and Jonathan P. How[1]

[1]Laboratory for Information and Decision Systems
[2]Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, {mubp, syliu, lpaull, jhow, jleonard}@mit.edu

***Abstract*— Mapping and self-localization in unknown environments are fundamental capabilities in many robotic applications. These tasks typically involve the identification of objects as unique features or landmarks, which requires the objects both to be detected and then assigned a unique identifier that can be maintained when viewed from different perspectives and in different images. The *data association* and *simultaneous localization and mapping* (SLAM) problems are, individually, well-studied in the literature. But these two problems are inherently tightly coupled, and that has not been well-addressed. Without accurate SLAM, possible data associations are combinatorial and become intractable easily. Without accurate data association, the error of SLAM algorithms diverge easily. This paper proposes a novel nonparametric pose graph that models data association and SLAM in a single framework. An algorithm is further introduced to alternate between inferring data association and performing SLAM. Experimental results show that our approach has the new capability of associating object detections and localizing objects at the same time, leading to significantly better performance on both the data association and SLAM problems than achieved by considering only one and ignoring imperfections in the other.**

I. Introduction

In many robotics applications, such as disaster relief, planetary exploration, and surveillance, robots are required to autonomously explore unknown spaces without an accurate prior map or a global position reference (e.g. GPS). A fundamental challenge faced by the robot is to effectively localize itself using only the information extracted from the environment. For example, the capability of recognizing instances of objects and associating them with unique identifiers will enable the robot to build maps of the environment and localize itself within. The problem of constructing a global map and localizing the robot within is referred as simultaneously localization and mapping (SLAM).

SLAM with various representations of the world and different sensors has been thoroughly studied in the literature. Occupancy grid map with LiDAR or laser range finders is among the early successes that dates back to the 1980s [1–4]. In occupancy based approaches, the world is represented by 2D/3D grids composed of free spaces and occupied spaces. New scans from the LiDAR or laser range finders are compared and matched with previous scans to incrementally build such maps. In recent years, SLAM with 3D dense mapping and RGB-D cameras has become more and more popular [5–7]. Incoming depth and color images are converted into volumes or deformation surfaces [5], then matched with previously constructed volumes or surfaces to incrementally build the map.

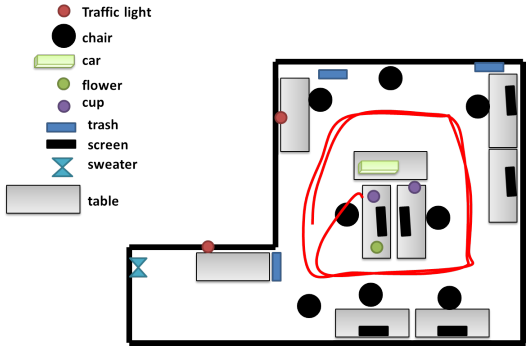


Fig. 1: In object SLAM, each object class has multiple instances, data association (associate detect objects to unique object idnetifiers) is ambiguous. Data association and SLAM (localize objects) are inherently coupled: good data association guarantees the convergence of SLAM, and good SLAM solution gives good initialization of data association.

A factor graph is a different representation of the SLAM problem [8–11]. Instead of using small units, such as grids, volumes, or surfaces, to represent the space, a factor graph encodes the poses of the robot and the observed landmarks along the trajectory. In a factor graph, each factor represents a constraint on the relative poses either between two consecutive robot poses or between a robot pose and a landmark. Factor graph SLAM scales much better than SLAM with occupancy grid maps or 3D dense maps. However, the convergence of factor graph SLAM algorithms relies heavily on correct data association of the landmarks.

The focus on of this work is on SLAM in unknown environment by recognizing objects and utilizing their positions (object SLAM). A factor graph is the natural representation, as objects can be easily represented as landmarks. Object SLAM requires the robot to be able to detect objects, generate measurements, and associate these measurements to unique identifiers. Some recent work on Region-based Convolutional Neural Networks [12, 13] gained significant success on training deep learning models to detect multiple objects instances within a single image. However, object detections only suggest the existence of objects of certain predefined object classes in an image, but provide no data association between images. This is problematic for SLAM especially when there are multiple objects of the same object class in an environment. How reliably SLAM can be achieved using only these ambiguous object detections remains an open question. As illustrated in Fig. 1, there are multiple instances of the same object class, such as chairs. The robot would need to establish the data association of

978-1-5090-3762-9/16/$31.00 ©2016 IEEE

object detections across images from different views.

This paper proposes a novel world representation, the nonparametric pose graph, to jointly perform data association and SLAM. The inference of the data associations and the optimization of the the robot and object poses are performed alternatively in this algorithm. This coupled framework achieves better performance for both data association and SLAM.

II. Related Work

Data association of objects and SLAM are typically solved as decoupled problems in the literature. Pillai and Leonard [14] showed that when the SLAM solution is known, and thus there is no uncertainty in robot poses, robot poses provide good prior information about object locations and can achieve better recalls than frame by frame detections. Song et al. [15] used a SLAM solver to build a 3D map of a room, and then fixed the map and manually labeled objects in the room. On the other hand, object detection can improve localization as well. Atanasov et al. [16] pre-mapped doors and chairs as landmarks. During the navigation stage, these pre-mapped objects are detected online and their location information is used to localize the robot.

However, in the scenario considered here, neither data association of objects nor robot poses are perfectly known. Algorithms that solve object detection and SLAM jointly can be categorized into front-end approaches and back-end approaches.

A. Front-end Data Association

In front-end approaches, objects detected in new images are compared with previous images. If matches between new and old images are found, then corresponding objects are associated to the same unique identifier. These data associations by front-end procedures are taken as reliable and true, and then passed to a SLAM solver [17, 18].

In this work, instead of creating exact templates for objects, deep learning is used to detect objects in the environment. Deep learning generalizes much better than template-based approaches. However, the detections have significant ratio of false positives and partial occlusions, thus are very challenging for front-end algorithms to produce reliable data associations.

B. Back-end Robust SLAM

Robust SLAM is a line of research that explicitly use back-end approaches to deal with outliers in the data [19, 9, 20]. In robust SLAM, when some measurement is incorrectly associated, it will be inconsistent with other object measurements of the same identifier. Robust SLAM instead maximizes a set of measurements that are consistent with each other in both identifiers and predicted locations. Only the consistent measurements are plugged into a SLAM solver to recover the robot poses and landmark locations.

By nature robust SLAM relies on the assumption that inlier measurements with unique identifier associations are the majority compared to outlier measurements. Under this assumption, eliminating outliers can still give good SLAM results. However, in object SLAM, it is often the case that there are multiple instances of the same object class. If all object measurements with same class are associated to the same identifier, different object instances will always give inconsistent measurements. If only one set of consistent measurements for each object class is kept, the algorithm will eliminate the majority of the data and fail to identify any repetitive instances of the same class.

The algorithm presented in this paper is a back-end approach where there are multiple instances of the same object class. The data association of object measurements to unique identifiers are considered unknown and must be established while doing SLAM. We exploit the coupling between data association and SLAM, jointly optimize both, and achieve better performance on both.

III. Object Measurements via Deep Learning

This section sets up the approach to generate object measurements via deep learning.

A. Deep Learning Based Object Detection

Object detection refers to the problem of identifying the existence of objects of certain classes and find bounding boxes for them in single images.

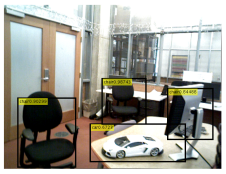
Region-based convolutional neural network (R-CNN) [12] was among the first works on object detection using a CNN. This algorithm uses the selective search [21] algorithm to generate bounding box proposals, and then crops an image patch using each proposal. The image patches are subsequently scaled and run through a CNN model for object detection. R-CNN is extremely slow (13 seconds per image) because all patches need to run through the CNN individually.

In Faster R-CNN [13], Ren et. al. ran the full image through the CNN only once, and they only use features in top layer in each bounding box patch for object detection. They further proposed a region proposal network (RPN) that learns how to generate bounding box proposals by looking at the top layer features. This new algorithm achieves an average speed of 100 milliseconds per image.

Our work trained a faster R-CNN model on the ImageNet 2014 dataset [22], which contain categories that are more relevant to indoor/urban settings, including *cars, motorcycles, bicycles, traffic lights, televisions, chairs, flowerpots, cups, and keyboards*. Note that this framework can be easily modified to parse out any other subset of classes from the ImageNet dataset that are relevant to the specific applications.

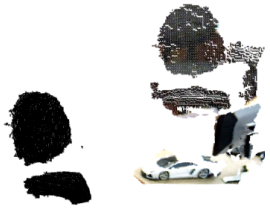
B. Object Measurements

An object measurement refers to a labeled 3D location with respect to the robot pose. To generate such measurements, location information relative to the robot is required in addition to object detection. In this paper, this is done by inquiring the corresponding pixels in the depth images: (1) Crop bounding boxes in the depth image in correspondence with the RGB bounding box. (2) Filter out background pixels that are too far away. (3) Generate point cloud from RGB and depth pairs. (4) Compute the centroid of the point cloud as center of the object.



(a) Object detection with RGB image.

(b) The depth image corresponding to the RGB image.



(c) Object localization in 3D space

Fig. 2: Deep learning based object detection

Fig. 2a shows the detected object with faster R-CNN from a single image of an office environment. Fig. 2b shows the corresponding depth image, and Fig. 2c shows the four point clouds for the four detected objects in 3D space.

It is clear from Fig. 2a that object SLAM with deep learning object detection has two major challenges. First, there are multiple instances of the object class, such as "chair" in Fig. 2a. Without correct data association, it is hard to distinguish different object instances. Standard pose-graph SLAM algorithms can only optimize poses with exact data association, such as g2o[23], isam[10], gtsam[24]. The second challenge is high false positive rates. As the `chair` detected in Fig. 2a, deep learning algorithms report objects now and then when there are actually none. Blindly using these unfiltered detections in standard SLAM algorithms will lead to "non-exist" nodes and cause loop closure failures.

Notice that the centroid is used as the center of objects in this case. When objects are looked at from different views, and be partially occluded, centroids would not be a consistent measure of the object locations. In our experience, the error could be 10-20cm. However, we will show that in office settings, our algorithm still converges even under significant occlusion and view point noise.

IV. Pose Graph Background

This section sets up the background on graphical models used in SLAM problems. The next section will extend the current pose graph to a novel nonparametric pose graph and introduce an algorithm to perform inference on it.

A factor graph is a graphical model widely used in SLAM problems. Denote $\mathbf{X} = \{X_1, \cdots, X_n\}$ as the random variables. Denote $\psi_a(X_{\{a\}})$ as a factor among random variable in set $\{a\}$.

The log likelihood, $\log p(x)$, can be written in a sum of factors: $\log p(x) \propto \sum_{a \in \mathcal{A}} \phi_a \left(x_{\{a\}}\right)$, where $\mathcal{A}$ is the set of all factors. Each factor $\phi_a(x_{\{a\}})$ maps the values of random variables to a strictly positive real number representing the log likelihood of the variables.

Fist assume that there exist static landmarks that the robot can identify to localize itself.

Assumption 1: There exists a library of static landmarks to localize the robot in the environment. The number and locations of these landmarks is not known a priori.

With the landmark assumption, when moving in the environment, the robot can obtain measurements of these landmarks. Given a dataset, the robot trajectory is represented as a discrete sequence of poses. Denote T as the total number of time steps, and denote $\mathbf{X}_{0:T} = \{X_0, \cdots, X_T\}$ as the robot's trajectory from the start to the end. Each robot pose consists of a position and an orientation. Denote $SE(2)$ as the space of 2D poses and $SE(3)$ as the space of 3D poses. Then $X_t \in SE(2)$ for 2D cases and $X_t \in SE(3)$ in 3D cases. In GPS-denied environments these poses are not directly observable. However, the robot can always measure the incremental change between two sequential poses via an IMU or wheel encoder, which is referred to as odometry. Denote o_t as the odometry measurement between pose x_t and pose x_{t-1}. Under the standard assumption that o_t is corrupted by additive Gaussian noise, the odometry measurement at time t can be represented as:

$$o_t = X_t \ominus X_{t-1} + v, \quad v \sim \mathcal{N}(0, Q), \tag{1}$$

where $\ominus$ represents an operator that takes two pose and return the relative pose between them in $SE(2)$ or $SE(3)$, and Q is the odometry noise covariance matrix.

During navigation, the robot also observes landmarks from the environment. Assuming that there exist M landmarks in the environment, which might be unknown ahead of time. The positions of the landmarks are denoted as $\mathbf{L} = \{L_1, \cdots, L_N\}$. In the 2D case $L_i \in \mathbb{R}^2$, and in the 3D case $L_i \in \mathbb{R}^3$. At time t, the robot obtains K_t landmark measurements, denoted as $\mathbf{z}_t = \{z_t^1, z_t^2, \cdots, z_t^{K_t}\}$. Each measurement is associated to a unique landmark identifiers, the associations are denoted as $\mathbf{y}_t = \{y_t^1, y_t^2, \cdots y_t^{K_t}\}$, where $y_t^i \in \{1, \cdots, M\}$. For example, at time 0, the robot obtained two measurements, $z_0 = \{z_0^1, z_0^2\}$. And these 2 measurements are from landmark 5 and 7, then $y_0 = \{y_0^1, y_0^2\} = \{5, 7\}$.

Using the standard model that object measurements z_t^k are corrupted by additive Gaussian noise:

$$z_t^k = L_{y_t^k} \ominus X_t + w, \quad w \sim \mathcal{N}(0, R) \tag{2}$$

where R is the measurement noise matrix.

The pose graph SLAM problem optimizes robot poses $\mathbf{X}_{0:T}$ and object locations $\mathbf{L}$ such that the log likelihood is maximized:

$$\max_{\mathbf{X}_{0:T}, \mathbf{L}} \log p(\mathbf{o}_{1:T}, \mathbf{z}_{0:T}; \mathbf{X}_{0:T}, \mathbf{L}). \tag{3}$$

Note that the log likelihood is nonlinear in X_t and L_t as $\ominus$ is a nonlinear operation in (1) and (2).

A factor graph representation for SLAM is also referred to as a pose graph.

V. Nonparametric Pose Graph

This section sets up the joint data association and SLAM problem by extending the current pose graph to a novel non-parametric pose graph that tightly couples object association with robot poses. A new algorithm is also introduced to

jointly infer the data association and perform SLAM with this new model.

A. Factor Graph with Multi-class Objects

Before we move into nonparametric factor graph for imperfect data association, first notice in object SLAM, except for measuring the 3D location of objects, we also observe an object class. The observed object class is not always reliable, thus we first establish the probabilistic model for object classes. Assume there are N object classes in total. For object i, denote u as an observation of the object class. The likelihood of u is modeled with a Categorical distribution:

$$p(u = j) = \pi_i(j), \quad j = 1, \cdots, N \tag{4}$$

Denote $\pi_i = \{\pi_i(0), \cdots, \pi_i(N)\}$, $\sum_{n=0}^{N} \pi_i(n) = 1$. And if the true object class is j, we have $\pi(j) \gg \pi(k)$ for $k \neq j$.

Notice $1 \leq u \leq N$, but we especially design $\pi_i(0)$ to represent the probability of false positives. This design would help the algorithm to filter non-exist object detections in real-world experiments.

In order to have closed form updates, we apply Dirichlet prior to π_i for object i:

$$\pi_i \sim \text{Dir}(\beta_i). \tag{5}$$

when there is an observation of class j, $u = j$, the posterior distribution of π_i is:

$$\pi_i | u \sim \text{Dir}(\beta_i + e_j). \tag{6}$$

where e_j represents a unit vector with jth element to be 1.

Notice $\beta_i(0)$ represents the initial likelihood of object i to be a false positive. Since observations cannot be 0, when there are more and more observations of object i being obtained, the posterior $\beta_i(0)$ will monotonically decrease. This is consistent with the intuition that if repeated observations are obtained from some object, then it has lower chance to be a false positive.

Combine the multi-class probabilistic setting with the original SLAM problem: each object measurement would be a pair $\{z_t^k, u_t^k\}$, where continuous variable z_t^k represents the 3D location measurement, and discrete variable u_t^k represents the observed object class. Recall that $y_t^k = i \in \{1, \cdots, M\}$ represents that the k-th measurement at time t is from object i. Then u_t^k is a sample from the posterior distribution $\pi_{y_t^k}$.

$$p(u_t^k = j) = \pi_{y_t^k}(j), \quad j \in \{1, \cdots, N\} \tag{7}$$

The joint log likelihood becomes:

$$\begin{aligned}
&\log p(\mathbf{o}_{1:T}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T}; \mathbf{X}_{0:T}, \mathbf{L}) \\
=& \sum_{t=1}^{T} p(o_t; X_{t-1}, X_t) + \\
& \sum_{t=0}^{T} \sum_{k=1}^{K_t} \left(p(z_t^k; X_t, L_{y_t^k}) + \log \pi_{y_t^k}(u_t^k) \right) \\
=& \sum_{t=1}^{T} p(o_t; X_{t-1}, X_t) + \sum_{t=0}^{T} \sum_{k=1}^{K_t} p(z_t^k; X_t, L_{y_t^k}) \\
& + \sum_{t=0}^{T} \sum_{k=1}^{K_t} \log \pi_{y_t^k}(u_t^k)
\end{aligned} \tag{8}$$

The new optimization problem is then

$$\max_{\mathbf{X}_{0:T}, \mathbf{L}, \pi} \log p(\mathbf{o}_{1:T}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T}; \mathbf{X}_{0:T}, \mathbf{L}, \pi). \tag{9}$$

Compared to (3), the observed data in problem (9) further includes object class observations $\mathbf{u}_{0:T}$, and the variables to be estimated further include the class of objects π. From (8), given data association $\mathbf{y}_{0:T}$, the joint likelihood can be factorized into the sum of likelihood of $\mathbf{z}_{0:T}$ and $\mathbf{o}_{0:T}$, and the likelihood of $\mathbf{u}_{0:T}$. Therefore, the class classes $\pi_{0:T}$ is independent of the robot poses $X_{0:T}$ and object positions $\mathbf{L}$. Optimizing (9) is equivalent to solving problem (3) and computing the object class posterior π independently.

B. Nonparametric Pose Graph

Now we move to the case that the data association y_t^k is unknown and must be established. Because of the ambiguous data association, the total number of objects M is unknown ahead of time, and needs to be established as well. Dirichlet Process (DP) is such a nonparametric stochastic process that models discrete distributions but with flexible parameter size. It can be taken as the generalization of a Dirichlet distribution with infinite dimension. Same as Dirichlet distribution is the conjugate prior for a categorical distribution, DP can be viewed as the conjugate prior for infinite, nonparametric discrete distributions [25]. In this work, we use a Dirichlet Process (DP) as the prior for data associations y_t^k. In particular, assume at any point, there are M objects being detected in total, the probability of y_t^k belongs to object i:

$$p(y_t^k = i) = \text{DP}(i) = \begin{cases} \frac{m_i}{\Sigma_i m_i + \alpha} & 1 \leq i \leq M, \\ \frac{\alpha}{\Sigma_i m_i + \alpha} & i = M + 1. \end{cases} \tag{10}$$

where m_i is the number of measurements of object i, and α is the concentration parameter of DP prior that determines how likely it is to create a new object. The intuition behind this model is that the probability y_t^k is from some existing object $i \leq M$ is proportional to the number of measurements of object i, and the probability y_t^k is from a new object $M+1$ is proportional to α.

The new optimization problem is then over The joint log likelihood of odometry $\mathbf{o}_{0:T}$, object measurement $\mathbf{z}_{0:T}$ and object classes $\mathbf{u}_{0:T}$ given data association $\mathbf{y}_{0:T}$:

$$\max_{\mathbf{X}_{0:T}, \mathbf{L}, \mathbf{y}_{0:T}, \pi} \log p(\mathbf{o}_{1:T}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T}; \mathbf{X}_{0:T}, \mathbf{L}, \mathbf{y}_{0:T}, \pi). \tag{11}$$

Compared with Equation (3), the new optimization problem Equation (11) is more challenging in that data associations $\mathbf{y}_{0:T}$ are unknown. As a result, log probabilities of object measurements no longer have a simple form, and the problem Equation (11) becomes a mixed integer nonlinear problem. Secondly, the number of true objects in the environment M is not necessarily known a priori, problem Equation (11) must infer M at the same time.

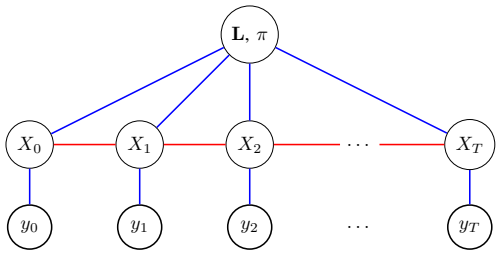


Fig. 3: Factor graph for SLAM with imperfect data association. y_t represents the data association: the measurement at time t is from object y_t. In SLAM with imperfect data association, y_t is unknown and must be established at the same time.

C. Nonparametric SLAM

From the last section, for $t = 1, \cdots, T, k = 1, \cdots, K_t$, the generative model for our problem is

$$y_t^k \sim \text{DP}(\alpha), \tag{12a}$$
$$\pi_{y_t^k} \sim \text{Dir}(\beta_{y_t^k}), \tag{12b}$$
$$o_t \sim \mathcal{N}(X_t \ominus X_{t-1}, Q), \tag{12c}$$
$$u_t^k \sim \text{Cat}(\pi_{y_t^k}), \tag{12d}$$
$$z_t^k \sim \mathcal{N}(L_{y_t^k} \ominus X_t, R), \tag{12e}$$

where α, β, Q, and R are given parameters. Robot poses $\mathbf{X}_{0:T}$, landmark locations $\mathbf{L}$, object class distributions $\pi_{1:M}$ and object associations $\mathbf{y}_{0:T}$ are variables to be estimated. The odometry $\mathbf{o}_{1:T}$ and object measurements $\mathbf{z}_{0:T}, \mathbf{u}_{0:T}$ are observed data.

Different from a canonical DP mixture model, the observed data $\mathbf{z}_{0:T}$, $\mathbf{u}_{0:T}$, and $\mathbf{o}_{0:T}$ are not independent samples given variables $\mathbf{X}_{0:T}$, $\mathbf{L}$, and π, but are correlated through the factor graph. Therefore, the inference involves computing maximum likelihood over factor graphs. When both associations and variables are to be established, standard approaches alternate between assigning data and optimizing variables. In the case of known object number M, K-means has a deterministic data association, while expectation-maximization associates data in a probabilistic way [26]. When the number of objects is not known a priori and DP is used as prior, Markov Chain Monte Carlo methods (e.g. Gibbs sampling) or variational inference algorithms are widely used [26]. However, in these algorithms, the likelihood of each label y_t^k to be any underlying object $\mathbf{L}$ needs to be computed and tracked all the time. The algorithm will need to go through all of the data multiple times to converge to a steady state distribution. The large scale and strong dependence of data in our problem make such approaches inappropriate.

It is shown in [27] that under the small variance assumptions, Gibbs sampling can be simplified to DPmeans. Instead of sampling the posterior distribution, y_t is assigned to be the maximum likelihood object if the likelihood is within some certain threshold, otherwise it is assigned to a new object. Intuitively, in this case, small variance means that the noise in odometry, object measurement and object class is relative small, so that the posterior distribution of y_t is peaky.

Assumption 2: Variance in odometry, object measurement and object class is small, so that the posterior distribution of data association has small variance and a unique maximal likelihood value.

The DPmeans algorithm alternates between two steps: maximize likelihood on variables $\mathbf{X}_{0:T}, \mathbf{L}, \pi$, and assign data association $\mathbf{y}_{0:T}$ to their maximum likelihood objects. Algorithm 1 shows the overall flow of the approach. And the following explains the algorithm step by step.

a) Initialization (line 1): In initialization, all y_t^k are set to be an object by its own. Robot poses $\mathbf{X}_{0:T}$ and object locations $\mathbf{L}$ are initialized by their open loop estimation. The Dirichlet distribution prior for object class are set to be some initial value β_0.

b) Optimizing data association (line 3): While executing the main loop, the algorithm alternates between optimizing associations $\mathbf{y}_{0:T}$, and variables $\mathbf{X}_{0:T}$, $\mathbf{L}$, and β. When optimizing object association, fix $\mathbf{X}_{0:T}$, $\mathbf{L}$ and β, and compute the posterior of y_t^k as the product of its DP prior (10) and likelihood of measurements (u_t^k, z_t^k) (see (1) and (7)).

$$p_i \propto \text{DP}(i) p(u_t^k; \pi_i) p(z_t^k; X_t, L_i). \tag{13}$$

Then y_t^k is assigned to the maximum likelihood object

$$y_t^k = \arg\max_i p_i. \tag{14}$$

c) Optimizing poses (line 10): When optimizing poses, object associations y_t^k are fixed. The posterior parameters for the Dirichlet distribution of object class can be updated with

$$\beta_i(j) \leftarrow \beta_0(j) + \sum_{t,k} \mathbb{I}_{y_t^k = i} \mathbb{I}_{u_t^k = j}, \tag{15}$$

where β_i is the hyper parameter for the Dirichlet prior on π_i. Notation $\mathbb{I}_{a=b}$ represents indicators whether quantity a equals quantity b. Then $\sum_{k,t} \mathbb{I}_{y_t^k = i}$ is the total number of object detections assigned to object i, and $\sum_{k,t} \mathbb{I}_{y_t^k = i} \mathbb{I}_{u_t^k = j}$ represents from the detections of object i, how many are class j. With Dirichlet prior $\text{Dir}(\beta_i)$, the maximum likelihood(ML) of each object class i is proportional to parameters β_i:

$$\pi_i = \text{ML}(\text{Dir}(\beta_i)). \tag{16}$$

The maximum likelihood value of robot poses $\mathbf{X}_{0:T}$ and object locations $\mathbf{L}$ can then be obtained by standard SLAM solvers (see (3)).

d) Remove false positive (line 18): Recall that we set $\pi_i(0)$ to be the probability that object i is a false positive. In initialization, $\beta_i(0)$ is set to be some positive number. When new measurements of object i are obtained and accumulated, β_i gets updated such that $\beta_i(j)$, $j > 0$ becomes bigger compared to $\beta_i(0)$. As a result, $\pi_i(0)$ decrease monotonically. In the last step, we filter out false positives by simply putting a threshold ϵ on $\pi_i(0)$.

VI. Experiment

A. Simulated Dataset

In the simulation, 15 objects are randomly generated in a 2D plane. They are randomly assigned into 5 different object classes: red diamonds, blue circles, green triangles, yellow stars, and magenta squares. The robot trajectory is manually designed and passes through the environment several times. Fig. 5a shows the ground truth of the generated dataset. At each pose X_t, the robot observes the relative position o_t^k

Algorithm 1 Nonparametric SLAM

Input: Odometry measurements $o_{1:T}$, Object measurements $\mathbf{u}_{0:T}$, $\mathbf{z}_{0:T}$

Output: Poses $\mathbf{X}_{0:T}$, number of objects M, object association $\mathbf{y}_{0:T}$, object locations and classes $\mathbf{L}$, β

Initialize $\mathbf{X}_{0:T}$, $\mathbf{L}$ with open loop prediction, initialize $\beta_i = \beta_0$. Initialize each y_t^k to be an object of its own
while not converged **do**
  Fix $\mathbf{X}_{0:N}$, $\mathbf{L}$, β
  for Each measurement y_t^k **do**
    Computer posterior p_i of being object i:
      $p_i \propto \mathrm{DP}(i) p(u_t^k; \pi_i) p(z_t^k; X_t, L_i)$
    Assign y_t^k to be maximum likelihood association:
      $y_t^k = \arg\max_i p_i$
  end for
  Fix $\mathbf{y}_{0:T}$
  for each object i **do**
    update class π:
      $\beta_i(j) \leftarrow \beta_i(j) + \sum_{t,k} \mathbb{I}_{y_t^k = i} \mathbb{I}_{u_t^k = j}$
      $\pi_i = \mathrm{ML}(\mathrm{Dir}(\beta_i))$
  end for
  optimize $\mathbf{X}_{0:T}$, $\mathbf{L}$ with standard SLAM solver with (3)
end while
Remove false positive
  $\forall i$, delete object i if $\pi_i(0) > \epsilon$

and class u_t^k of the objects that are within its field of view. Gaussian noise are added to the odometry measurements as well as object measurements, see (1), and (2). The parameters of the dataset are listed in Table I.

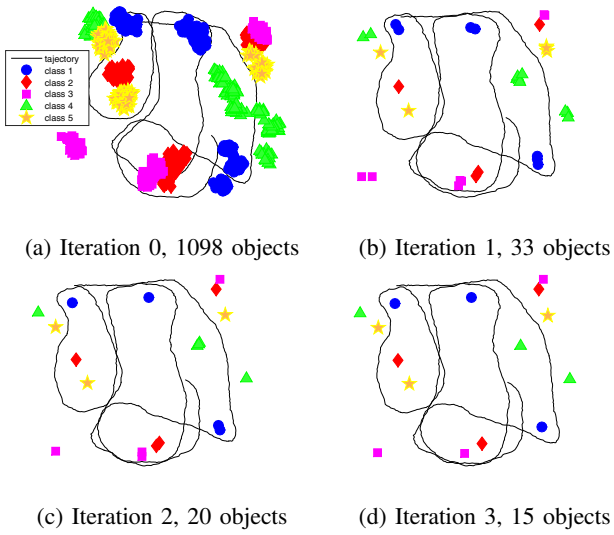
TABLE I: Simulated Dataset Overview

Distance Traveled	72.7m
field of view	4m, 120 degree
no. of odometry measurements	766
no. of object measurements	1098
odometry noise	$\mathcal{N}(0, 0.02^2)$
measurement noise	$\mathcal{N}(0, 0.1^2)$

Fig. 4a shows the object predictions based purely on open-loop odometry. There is significant amount of variance and drift in the distribution of these predicted object locations, which obscures the determination of exactly how many objects there actually are in the environment. The result after the first iteration is shown in figure 4b; the nonparametric pose graph clusters the measurements and uses it to correct robot poses. The total number of objects is reduced to 33. The result after the second iteration is shown in figure 4c; the algorithm further reduces the total number of objects to 20. After three iterations(figure 4d), the algorithm converges to the true underlying number of objects, which is 15.

The performance of the proposed nonparametric graph (NP-Graph) is compared to three existing methods:

1) *Frame by frame detection (FbF)*: each object in each frame is taken as new, and there are neither SLAM nor data association (see Fig. 4a).
2) *Open-loop Object Detection (OL)*: use robot odometry to perform data association across images, but do not use data association results to correct robot poses (see Fig. 5c).
3) *Robust SLAM (R-SLAM)*: back-end algorithm that finds the maximal set of consistent measurements, but eliminate inconsistent measurements (see Fig. 5b).



(a) Iteration 0, 1098 objects (b) Iteration 1, 33 objects

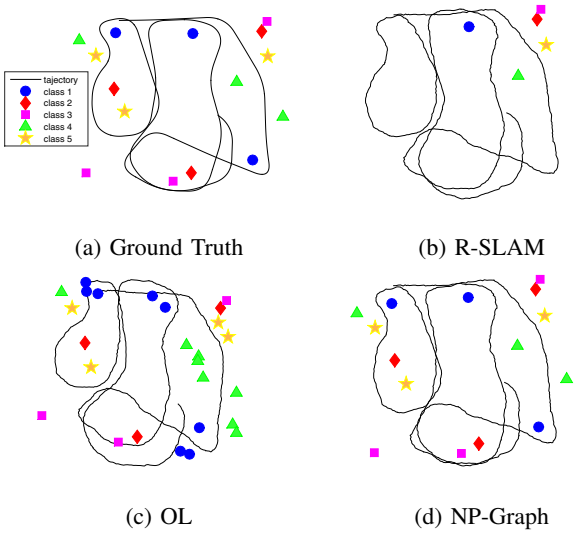
(c) Iteration 2, 20 objects (d) Iteration 3, 15 objects

Fig. 4: Result of nonparametric pose graph at different iterations. Initially there are 1098 object detections. The number reduces to 33 after the first iteration, reduces to 20 after the second iteration, and converges to the ground truth 15 after 3 iterations.

TABLE II: Performance Comparison on Simulated Dataset

	mean pose error	cumulative trajectory error	percent of measurements used	number of objects	mean object error
NP-Graph	0.07	55.1	100	15	0.05
OL	0.42	320.6	100	39	0.39
R-SLAM	0.20	150.5	20.2	5	0.20
FbF	0.42	320.6	100	1098	0.49

Fig. 5 and Table II compare the SLAM performance results of four different algorithms. FbF and OL purely rely on odometry and do not correct robot poses, therefore have the biggest error. R-SLAM uses a subset of object measurements to close loops on robot poses, thus the error is smaller. Our NP-graph based approach make use of all the object measurements, thus has the smallest error on both robot poses and object positions. FbF does not do any data association, thus significantly over estimate the number of objects. The OL approach does not optimize robot poses. When the robot comes back to a visited place, the odometry has drifted significantly thus the OL approach could not associate the objects to the same one observed before. As a result, the OL approach also over estimate the total number of objects. R-SLAM only keeps one set of consistent measurements for each object class, therefore it is only able to detect one instance for each object class, and significantly underestimate the total number of objects. NP-Graph, on the other hand, utilize all of the object measurements and jointly infers both robot poses and the data associations, thus can correctly infer the right number of objects.



(a) Ground Truth (b) R-SLAM

(c) OL (d) NP-Graph

Fig. 5: Simulation. Black line represents the robot trajectory. Each marker color/shape represent an object class. FbF does neither data associate nor SLAM. OL associate object detection across images but does not optimize robot poses. R-SLAM only uses a subset of consistent object measurements to optimize robot poses. Our approach NP-graph optimizes both robot poses and data association.

B. Office Environment

To test the performance in real-world scenarios, we collected a dataset of an office environment and used deep learning to detect objects, such as chair, screen, cups etc. The statistics about the office dataset is shown in Table III.

TABLE III: Office Dataset

image resolution	640×480
distance traveled	28.06m
during	167s
no. of odometry	696
no. of objects	30
no. of object detections	1588
odometry noise	$\mathcal{N}(0, 0.1))$
measurement noise	$\mathcal{N}(0, 0.5)$

Table IV and Fig. 6 compare the performance of FbF, R-SLAM, OL and our approach NP-Graph. While the ground truth for object positions is not available for this dataset, we compare the performance on the number of valid objects, the number of inlier measurements and the variance on object positions. An object is defined as valid when its false positive probability $\pi_i(0)$ is below a threshold ($\epsilon = 2\%$), otherwise it is marked as a false positive. A measurement is denoted as an inlier when it is associated with a valid object. The object variance is determined from the uncertainty in the predicted location of the object from its associated measurements. From Table IV, the NP-Graph has the highest percentage of inlier measurements, the closest number of objects to truth, and the smallest variance on the object locations.

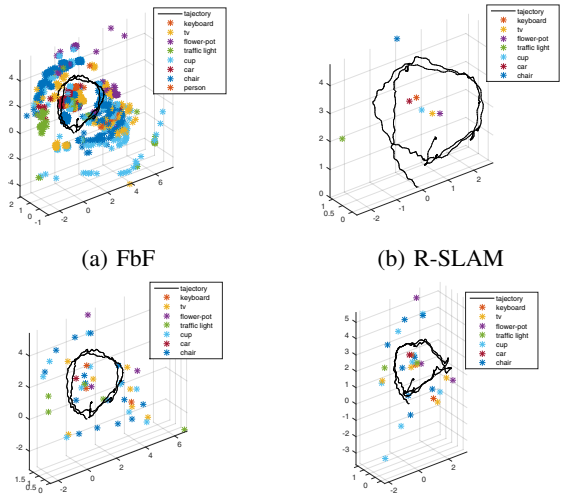
While the ground truth for robot poses is not available, either, we compare the performance qualitatively. Fig. 1 shows the floor map of the environment as well as the robot trajectory. Fig. 6 compares the results of 4 approaches. FbF and OL estimation are open-loop approaches and over estimate total number of objects. R-SLAM only uses a subset

TABLE IV: Performance Comparison on Office Dataset

	percentage of measurement inliers	number of inlier objects	number of false positive objects	variance on objects
NP-Graph	88.0	31	88	0.058
OL	82.2	36	175	0.121
R-SLAM	22.5	7	0	0.225
FbF	0	0	1588	-

of the object measurements. It can only identify one instance for each object class, and has bad estimates even it closes loops on robot poses. On the other hand, NP-Graph is able to close loops on robot poses and recover the turnings at corners. While there is no ground truth in the office dataset for computing object localization errors, it is worth noting that there is a sweater hanging on the shelf in the far bottom left corner, our algorithm is able to recover its distance while other approaches failed to.

Fig. 7 shows a few examples of the detected and well associated objects, which includes chair, screen, keyboard, toy car and the sweater hanging in the back corner. These figures are extracted from point cloud of a single bounding box that is associated to the corresponding object. Note that these point clouds are only for illustration purposes, but not maintained in the algorithm. The algorithm only uses the centroid of these point clouds as object measurements.



(a) FbF (b) R-SLAM

(c) OL (d) NP-Graph

Fig. 6: Office Dataset. Black line represent robot trajectory. Markers represent objects. Each color represent an object class. FbF approach does not do data association nor SLAM. R-SLAM does SLAM but not data association. OL approach does data association, but not SLAM. NP-Graph jointly infers data association and does SLAM. It has the least number of objects, data localize the objects, and closes loop thus has least error on robot trajectory.

VII. Conclusion

Object SLAM is challenging as data association is ambiguous and location measurements unknown. Data association and SLAM are inherently coupled problems. This work proposed a novel nonparametric pose graph that tightly couples these two problems, and developed an algorithm to alternative between inferring data association and performing SLAM. Both simulated and real-world datasets show that our

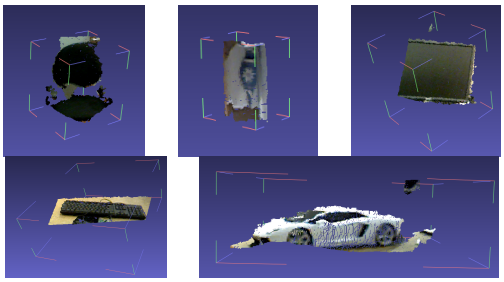


Fig. 7: Example of detected objects, plotted from a single frame point cloud. From left to right, top to down are chair, sweater in the corner, screen, keyboard and toy car.

new approach has the capability of doing data association and SLAM simultaneously, and achieves better performance on both associating object detections to unique identifiers and localizing objects.

ACKNOWLEDGMENTS

This research is supported in part by ARO MURI grant W911NF-11-1-0391, ONR grant N00014-11-1-0688 and NSF Award IIS-1318392.

REFERENCES

[1] A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Sixth Conference on Uncertainty in AI*, pages 7–24, 1990.

[2] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, june 1989.

[3] S. Thrun. Learning occupancy grids with forward sensor models. *Autonompous Robots*, 15:111–127, 2003.

[4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT press, Cambridge, Massachusetts, USA, 2005.

[5] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense slam without a pose graph. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[6] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3D Vision - 3DV 2013, 2013 International Conference on*, pages 1–8, June 2013.

[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*. IEEE, October 2011.

[8] I. Mahon, S.B. Williams, O. Pizarro, and M. Johnson-Roberson. Efficient view-based SLAM using visual loop closures. *Robotics, IEEE Transactions on*, 24(5):1002–1014, Oct. 2008.

[9] M.C. Graham, J.P. How, and D.E. Gustafson. Robust incremental slam with consistency-checking. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 117–124, Sept 2015.

[10] D.M. Rosen, M. Kaess, and J.J. Leonard. An incremental trust-region method for robust online sparse least-squares estimation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1262–1269, May 2012.

[11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587, 2014.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[14] S Pillai and J. Leonard. Monocular slam supported object recognition. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[15] S. Song, L. Zhang, and J. Xiao. Robot in a room: Toward perfect object recognition in closed environments. *CoRR*, abs/1507.02703, 2015.

[16] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Semantic localization via the matrix permanent. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

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

[18] J. Civera, D. Galvez-Lopez, L. Riazuelo, J.D. Tardos, and J. M. Montiel. Towards semantic slam using a monocular camera. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277–1284, Sept 2011.

[19] N. Sunderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1254–1261, May 2012.

[20] Y. Latif, C. Cadena, and J. Neira. Robust loop closing over time for pose graph slam. *The International Journal of Robotics Research*, 2013.

[21] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F. Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[23] g2o: A general framework for graph optimization. https://openslam.org/g2o.html.

[24] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sept 2012.

[25] T. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[26] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st edition, 2007.

[27] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 513–520, 2012.