Technion – Israel Institute of Technology



Probabilistic Data Association for Semantic SLAM

Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, George J. Papas

ICRA 2017

|  |  |  |
| --- | --- | --- |
| Alon Spinner | 305184335 | alonspinner@gmail.com |
| Sher Hazan | 308026467 | Sherhazan1115@gmail.com |

February 1, 2022

# Introduction

In the past..

Recently…

Now..

A picture containing shape

Description automatically generated

# Problem Formulation and Main Contribution

Graphical user interface

Description automatically generated

Graphical user interface, diagram

Description automatically generated

# Method

To solve the data association in the back end, a two-step optimization process was devised, one where the data association computed by marginalization.

## Step1 – Initialize state

A picture containing company name

Description automatically generated



Text, letter

Description automatically generated

Chart, scatter chart

Description automatically generated

Diagram

Description automatically generated

## Step2.a - Cluster objects

Text, letter

Description automatically generatedText

Description automatically generated

Chart, histogram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Chart, histogram

Description automatically generated Chart, histogram

Description automatically generated

Chart, radar chart

Description automatically generated

## Step2.b - Compute posterior class distribution of clusters





Diagram

Description automatically generated

Diagram, venn diagram

Description automatically generated

Chart, line chart

Description automatically generated

## Step3 – Solve for camera poses and object locations given association

Diagram

Description automatically generated

## Step 4 – Remove false positives

Chart

Description automatically generated with low confidence

# Simulation

In the simulation, we simulated 15 objects are randomly generated in a 2D plane. The objects are assigned equally into 5 different object classes ( 3 objects on each class). The robot trajectory is manually designed and passes through the environment several times. As shown in figure 1.

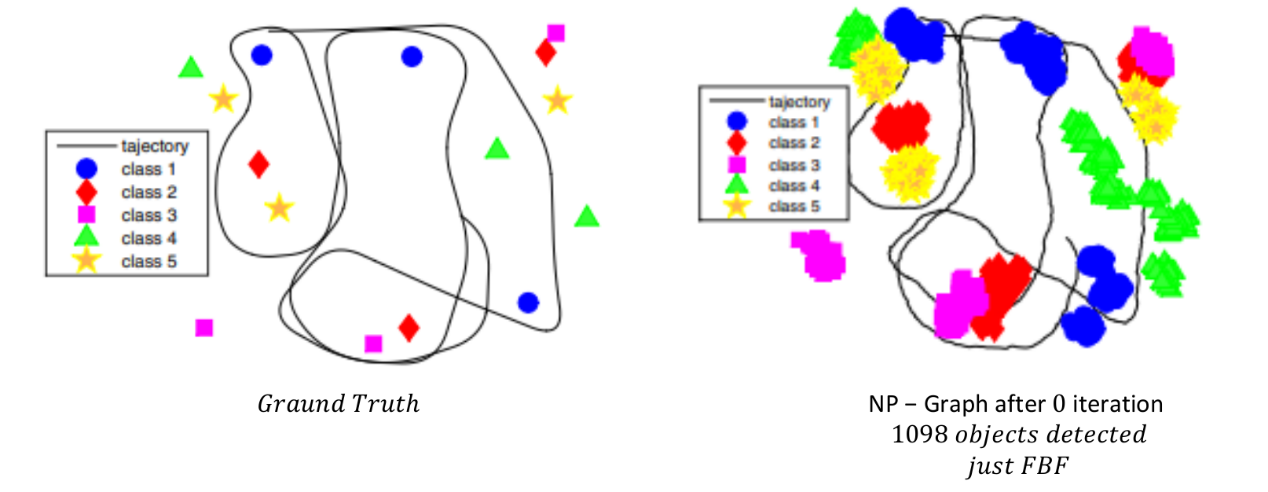


Figure 1:the gtaund truth of the simulmulated Dataset

After we generated the dataset, we calculated the trajectory and the object classes and their location by using Frame by frame detection (FbF) method. In this method each object in each frame is taken as new, and there are neither SLAM nor data association  
As we can see in figure 2 after using FbF method we observed 1098 different objects, In addition, the trajectory aren't fit as well, As a result of the lack of SLAM.

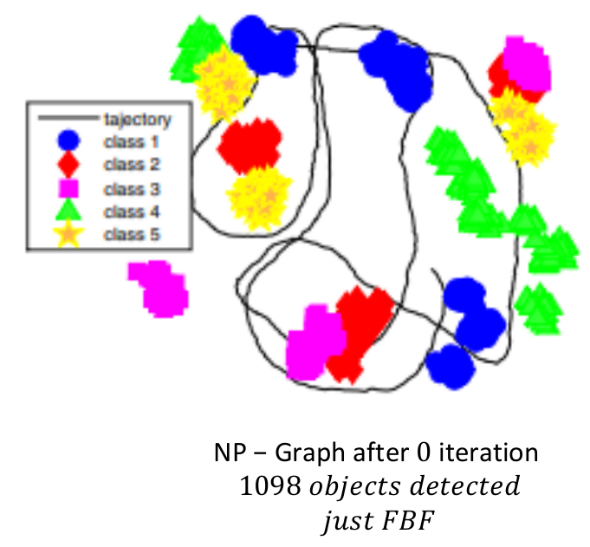


Figure 2: Initialize by FbF method

Then, we run our algorithm. After the first iteration the nonparametric pose graph clusters the measurements and uses it to correct robot poses. The total number of objects is reduced to 33. After the second iterations the algorithm further reduces the total number of objects to 20. and after the third iteration the algorithm converges to the true underlying number of objects, which is 15.

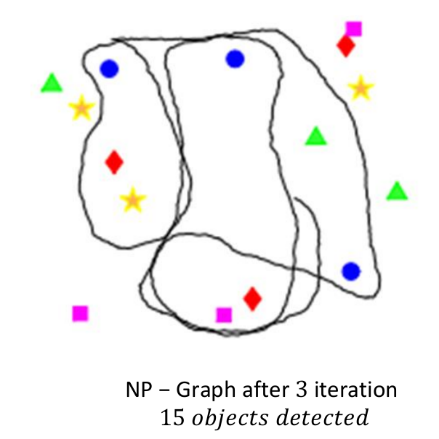
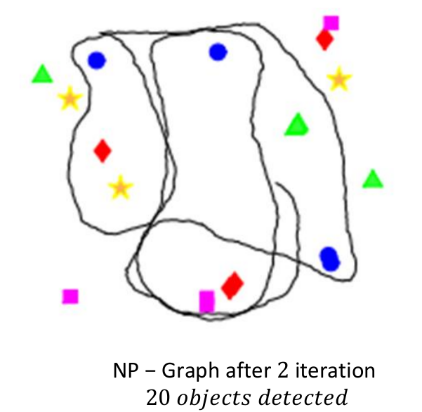
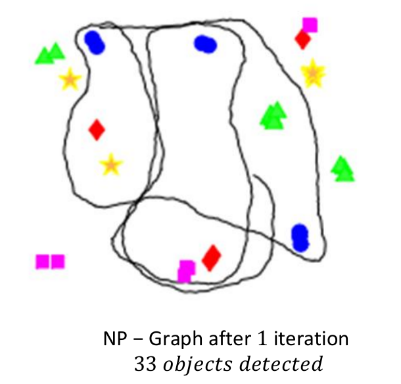


Figure 3: Result of nonparametric pose graph at different iterations.

Compare the SLAM performance results of different algorithms.  
Open loop (OL) method purely rely on odometry and do not correct robot poses, therefore have the biggest error. Furthermore, OL does not do any data association, thus significantly overestimate the number of objects.  
R-SLAM uses a subset of object measurements to close loops on robot poses, thus the error is smaller. But this method 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.  
Our method (NP-graph) make use of all the object measurements, thus has the smallest error on both robot poses and object positions. In addition, our method 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.  
In the following figures we shown the results of each method.

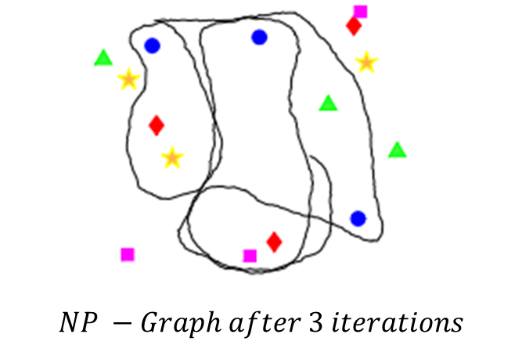
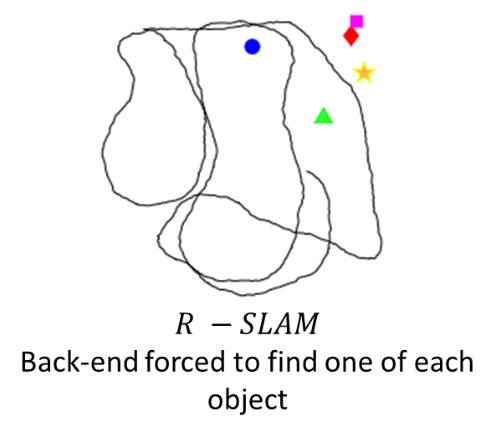


Figure 4: Compare the SLAM performance results of different methods.

# Experiment

To test the performance in real-world scenarios, we collected a dataset of an office environment and used R-CNN to detect objects, such as chair, screen, cups etc.  
Because 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.

The following figure shows a few examples of the detected and well associated objects, which includes chair, screen, keyboard and toy car. These figures are extracted from point cloud of a single bounding box that is associated to the corresponding object, the algorithm uses the centroid of these point clouds as object measurements.

In the experiment, we obtained similar results to the results obtained from the simulation as expected.



Figure 5: Compare the SLAM performance results of different method in real-world scenarios.