Technion – Israel Institute of Technology

SLAM with Objects using a Nonparametric Pose Graph

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

IROS 2016

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

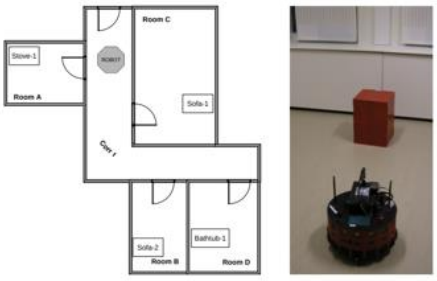
February 2, 2022

Introduction and Motivation

In a talk given by J. Leonard recently for the Tartan SLAM series [1], he shows cases a video from the 1970th where H. Moravec utilizes the 'Stanford Cart', equipped with a KL10 processor and a television camera, to navigate between objects - to do Vision Aided Navigation [2]. Given the lack of technological means to process rich sensors efficiently at the time, the theme of object-based navigation was necessary. Yet, as the robotics field grew, object labels, or semantics, had seemed to take a back-seat in the problem formulation.

In 2005, approximately 10 years after the term 'SLAM' had been coined [3], C, Galindo et al. wrote that most progress had been based on metric and/or topological representations of maps and works to fix this discrepancy [4]. In his work, he identifies colored boxes in rooms, representing semantic objects (sofa, stove…) to unify geometric maps of occupancy grids, utilizing the topological graph of the house rooms. They showed that known semantics are required to achieve *advanced* reasoning about the map's structure and utility.

Two years later, in 2007, S. Vasudevan et al. expanded on previous works by creating and updating a map with objects detected and associated via SIFT detectors-descriptors [5]. In this work they begin to tackle the hardships of mapping with semantics, most of which can be pointed towards data association. They also harvest the benefits that meaningful sparse landmarks provide, mostly in computationally, as solving the SLAM problem means solving for a large joint state of robot poses and landmarks.



2005: Multi-Hierarchical Semantic Maps for Mobile Robotics
C. Galindo et al.
(IROS)

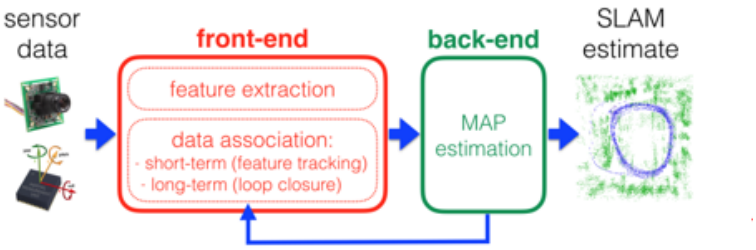
2007: Cognitive maps for mobile robots—an object based approach
Shrihari Vasudevan et al.
Robotics and Autonomous Systems 55

Figure 1 Early Works on Semantic SLAM

On the same year, 2007, a highly influential work done by George Klein and David Murray, PTAM, solved the computational problem differently achieving real-time implementation by splitting the SLAM problem to what we now call "front-end" and "back-end" [6]. In their work's front-end, robust data association of FAST image features were done via RANSAC, and the back end is comprised of *"an almost textbook implementation of Levenberg-Marquardt bundle adjustment"*. In the following years, factor graph optimizers that solve for the back end of the SLAM problem were popularized, the famous of which are still frequently used today [7], [8].

A decade after PTAM was released, their splitting solution was engrained into the SLAM community. In a survey paper by C.Cadena and L.Carlone

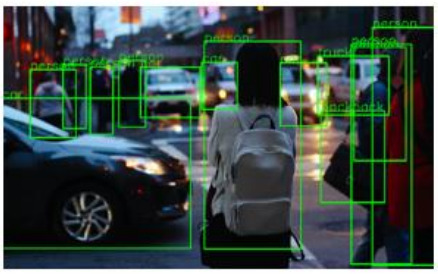
published in 2016, they write *“The architecture of a SLAM system includes two main com- ponents: the front end and the back end ”*[9].
In the same paper, the authors ask if geometric SLAM is a solved problem, and note that semantic SLAM is in its infancy, and still lacks cohesive formulation.



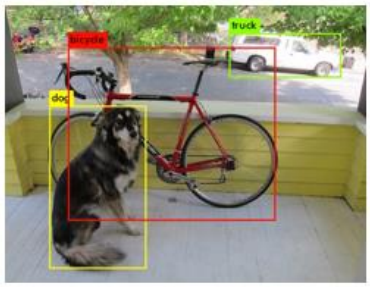
2016: Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age
(DOI: 10.1109/TRO.2016.2624754)

Figure 2 Front-End/Back-End Split

As such, a reasonable hypothesis would be that the front-end/back-end split does not suite the SLAM solution when semantics are included. In their work, which we present here, Beipeng Mu et al. utilizes the new technology of RGB-D cameras, and Deep Neural Networks for object detection and labeling to solve for robust data association as part of the back-end optimization [10][11],[12].



2014: R-CNN
Ross Girshick, Jeff Donahue, Trevor Darrel, Jitendra Malik

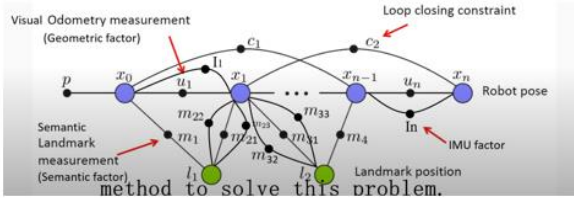


2015: You Only Look Once (YOLO)
Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

Figure 3 Deep Neural Network For Object Detection

A year later, Sean L. Bowman et al. showed how semantic factors can be incorporated into existing factor graph solvers [13]. It was this method that proved most successful and eventually adopted by the SLAM community.

2017: Probabilistic Data Association for Semantic SLAM
Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, George J. Papas



https://www.youtube.com/watch?v=ssupiMpUU20&t=40s&ab_channel=YidongDu

Figure 4 Semantics in factor graph

Problem Formulation

In this work the authors solve an object-based SLAM problem solving the following optimization problem:

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

Where camera poses are denoted X, object locations L, object class posteriors π. As for measurements, at each time step t, an odometry measurement o_t. as-well as relative object-camera pose z_t^k, and object classes, u_t^k, measurements. The k superscript indicates the k^{th} object measurement at time t. The object measured is also associated with an index of the object y_t^k.
Finally, we would like to note that the measurement z_t^k is of the object's bounding box geometrical center.

$$o_t = X_t \ominus X_{t-1} + v \qquad v \sim \mathcal{N}(0, Q)$$
$$z_t^k = L_{y_t^k} \ominus X_t + w \qquad w \sim \mathcal{N}(0, R)$$
$$u_t^k = argmax(RCNN\ Class\ Posterior)$$

Method

As mentioned beforehand, the authors aim to solve an object-based SLAM. They do this by solving the data association inside a two-step incremental solver. The first step solves the data association by marginalization and the second step solves the regular SLAM problem with a standard factor graph solver.

In this section we provide a walk-through of the proposed algorithm which we split to 4 steps: Initialization, clustering (data association), SLAM solver and false positive removals.

We would like to note that this is different from a front-end,back-end approach as steps 2-3 are iterative In the optimizer.

Additionally, while utilizing the syntax from the article, we do not go over each equation as it was written there for pedagogical reasons. For the complete picture, which includes the pseudo-code you, the reader, is referenced there.

Step1 – Initialization via Frame-by-Frame

In the algorithm, the authors initialize their solver with a Frame-by-Frame (FbF) method, in which the odometry is integrated in an open-loop manner, and each object viewed in a frame is assumed to be unique.

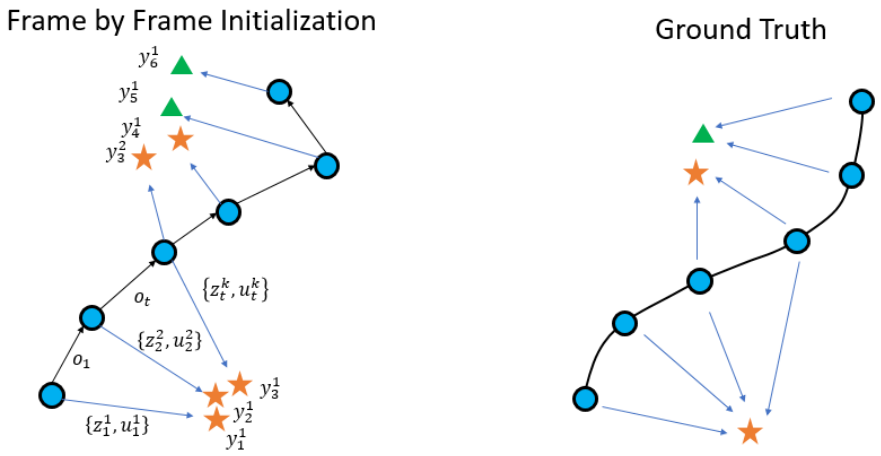


Figure 5 Frame by Frame Initialization

Step2.a - Cluster objects

After the FbF initialization, the authors turn to object clustering as a means for data association. They decide to cluster objects as function of their classes, and the distance between them.

In the figure below we changed the y_t^k notation to Obj_i. as y_t^k is an index pointing to Obj_i.

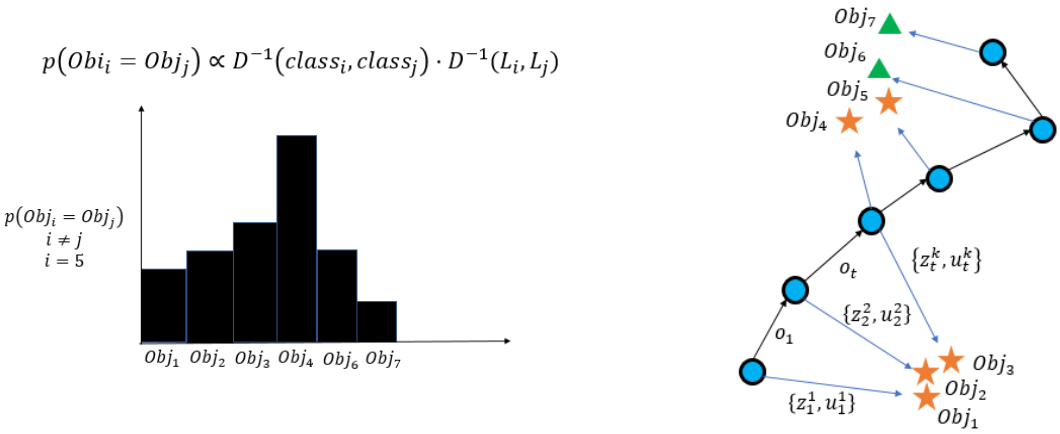


Figure 6 data association probability

To correct for the probability that Obj_i may not be associated with any of the other objects, but is a true positive, they process the probability distribution through a Dirichlet process. In this process, a probability distribution is manipulated by the following equation, to create a new one:

$$p(i=j) = DP(i) = \begin{cases} \dfrac{m_i}{\Sigma_i m_i + \alpha} & 1 \le i \le m \\ \dfrac{\alpha}{\Sigma_i m_i + \alpha} & i = M+1 \end{cases}$$

In the new distribution, the probability that Obj_i is unique, and a true positive is α. The whole Dirichlet process is demonstrated below for Obj_5 in our toy example.

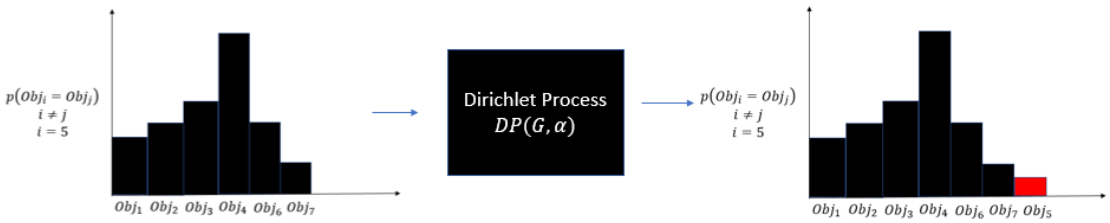


Figure 7 Dirichlet Process

To finalize the data association by clustering with a value, the authors decided on a threshold value. If $\max\left(p\left(Obj_i = Obj_j\right)\right) > threshold$ than Obj_i is associated with $\underset{j}{\operatorname{argmax}}\left(p\left(Obj_i = Obj_j\right)\right)$, else, Obj_i is determined to be unique.

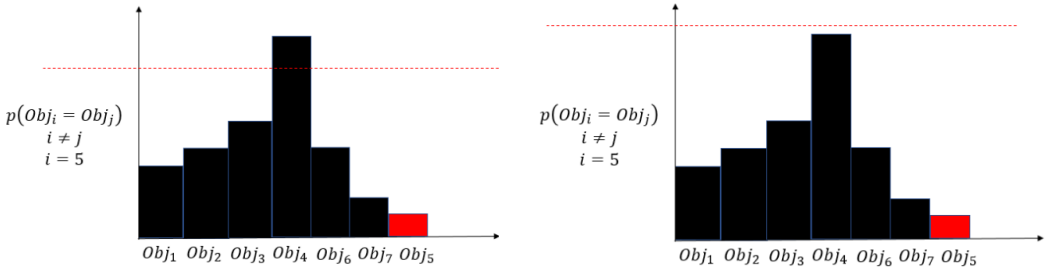


Figure 8 Thresholding pdf to determine data association or uniqueness

Step2.b - Compute posterior class distribution of clusters

After clustering object observations y_t^k, the authors turn to update object class posteriors distribution π_i of Obj_i. They do this using the amount of votes each class receives, held in a vector β_i and a Dirichlet distribution. Each observation y_t^k provides a vote according to the maximum likelihood of the posterior class distribution of the object it points to.

Figure 9 Cluster class votes example

A Dirichlet distribution is probability distribution over multiple variables, holding one parameter for each variable, where the sum of all variables is one at each point.

$$f(x_1, \dots, x_k | \alpha_1, \dots \alpha_k) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

$$\sum_{i=1}^{k} x_i = 1 \qquad x_i \geq 0$$

To calculate π_i, the posterior categorical class distribution, from the Dirichlet distribution, the authors take the ML.

$$\pi_i = ML\big(Dir(\beta_i)\big)$$

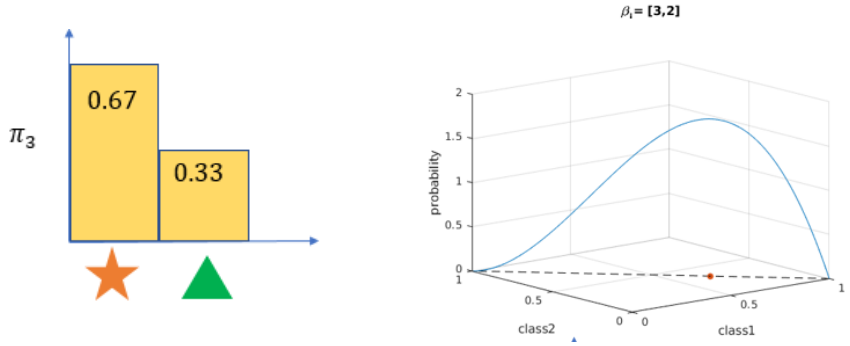


Figure 10 Dirichlet Distribution Example

To account for object detections being false positives, each object is initialized with one vote for such case in $\beta_i(0)$. for our example, the corrected voting counts β_3 is shown below

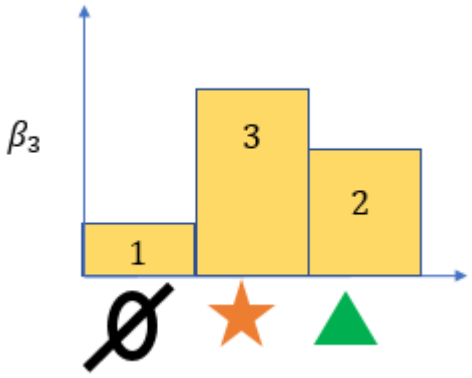


Figure 11 Object initialization with false positive

We would note that Step 2 of the algorithm compromises the “non-parametric“ part declared in the method’s name.
From Wikipedia: *“Nonparametric statistics is based on either being distribution-free or having a specified distribution but with the distribution's parameters unspecified”*

Step3 – Solve for camera poses and object locations given association

As the data association has been solved in step 2, the authors can now utilize a standard slam solver. In this case they chose isam.

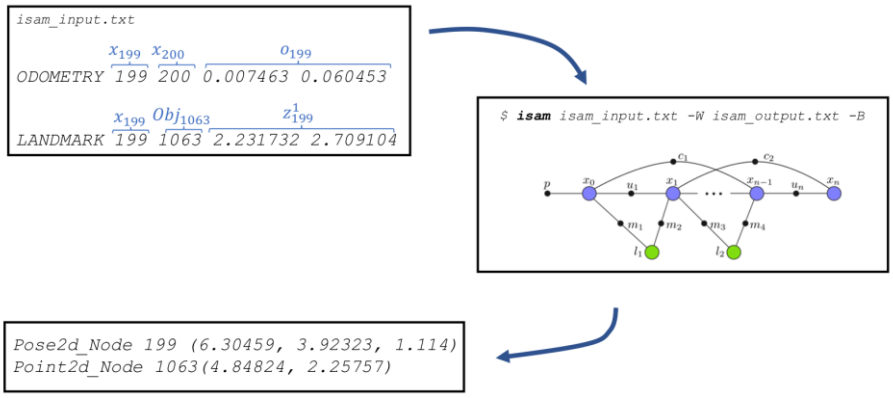


Figure 12 ISAM as SLAM factor graph solver

Step 4 – Remove false positives

After convergence of the iterative steps 2-3, the authors turn to the task of eliminating false positives. They determine an observation of an object is a false positive if its class distribution posterior $\pi_i(0)$ holds a value bigger than some predetermined ϵ. This correlates to object observations that were not clustered with enough other observations.

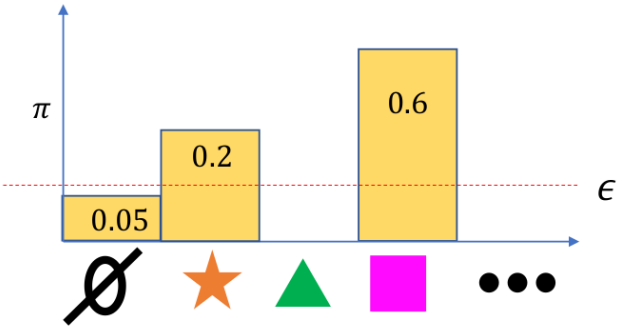


Figure 13 Removing False Positives

Simulation

In their simulation, Beipeng Mu et al. simulated 15 objects that are randomly generated on a 2D plane. The objects are assigned equally into 5 different object classes (3 for each class). The robot trajectory is manually designed and passes through the environment several times to ensure loop closures as shown below:

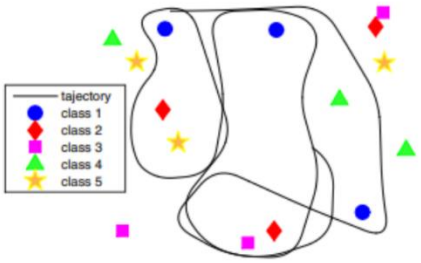


Figure 14 Simulation ground truth

After the dataset was generated, they turned to initialization of their algorithm using the FbF method which concluded with 1098 object observations.

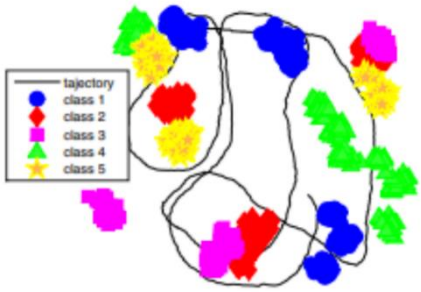


Figure 15 FbF initialization - 1098 objects detected

After the first iteration of optimizer, the total number of objects is reduced to 33. The second iterations the optimizer 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.

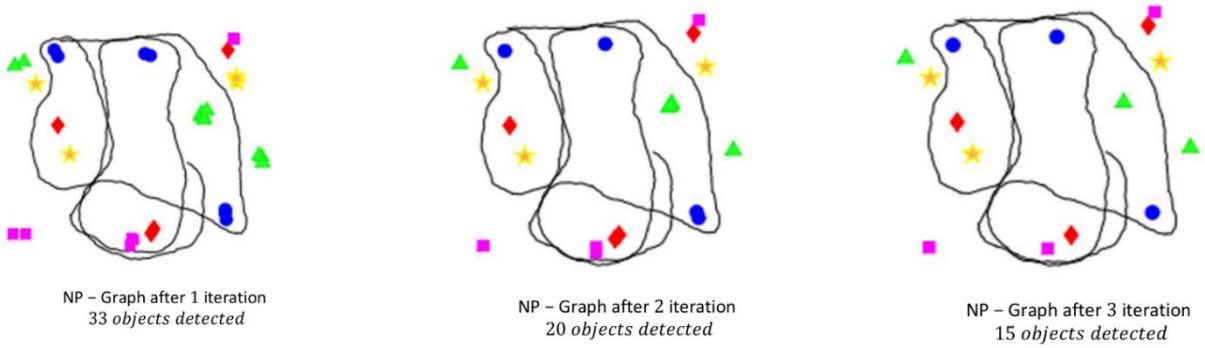


Figure 16 NP-graph optimizer Iterations

To analyze their results, the writers compare their nonparametric pose graph optimization method to other methods:

1. Open Loop (OL) – A method that purely relies on odometry integration for pose estimation and does only the data association step in the optimization process (no SLAM solver)
2. Robust SLAM(R-SLAM) – This method uses a subset of consistent object measurements to optimize robot pose in the SLAM solver, assuming one object of each class.

As the OL method purely integrates the odometry, it has the largest trajectory error. Those drifts cause wrong object associations in the optimizer iterations, which causes overestimating the number of total objects.
The R-SLAM on the other side completely underestimates the number of objects because of its pre-built assumption.

The suggested method in this article (NP-graph) makes use of all the object measurements, thus has the smallest error on both robot poses and object positions. In addition, it infers correct data associations, and thus correctly infer the right number of objects.
In the following figures we shown the results of each method.

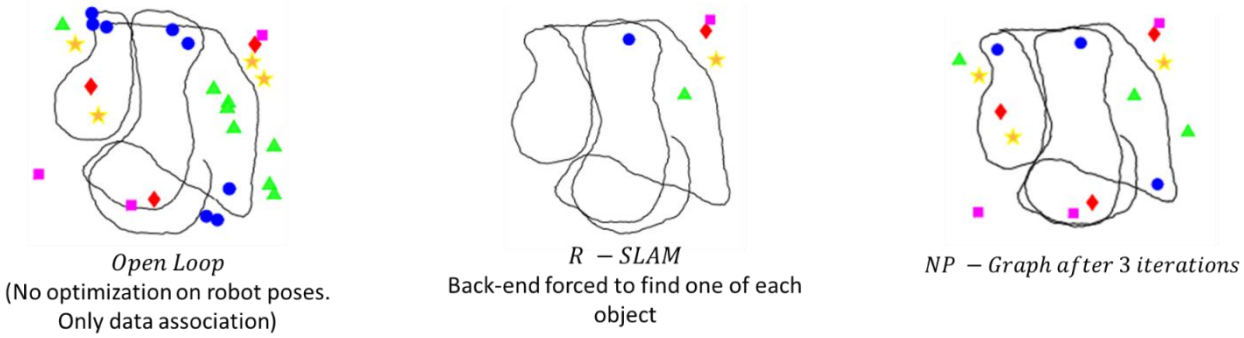


Figure 17 Simulation - method comparison

Experiment

To test the performance in real-world scenarios, Beipeng et al. collected a dataset of an office environment and used R-CNN to detect objects, such as chair, screen, cups etc. The geometrical observation z_t^k is computed as the centroid of the point cloud corresponding to the object measurement.
The class observation u_t^k is computed as the ML on the R-CNN categorical distribution output.

Figure 18 Experiment measurements

Because the ground truth for object positions is not available in the dataset, they compared the performance on the number of valid objects, the number of inlier measurements and the variance on object positions.

The following figure shows the final trajectory and objects for each method validating the conclusions from the simulation.
Note: Graphs are presented in 3D but are 2D.

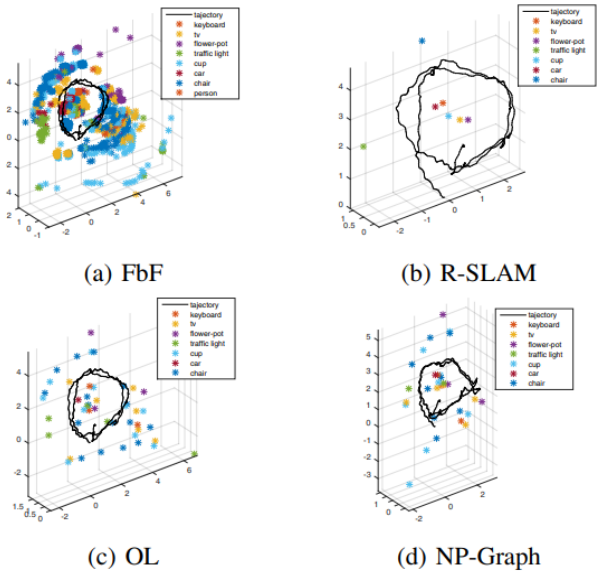


Figure 19 Compare the SLAM performance results of different method in real-world scenario

Conclusions

In this paper, Beipeng Mu et al. present one of the first simple and robust frameworks for semantic SLAM which incorporates data association as part of the solver and includes deep learning for object recognition.

Besides not including semantic factors in the SLAM graph solver, we think that the main drawback of this method is the use of object centroids instead of transforms. Giving attention to the orientation of the object observation can also aid in the data association.

Furthermore, the data association via non-parametric Bayesian inference is highly sensitive to noise and forces the authors to assume small variance on all measurements. This assumption ensures that the data association will have small variance and a unique maximal likelihood value.

Additionally, we would note that the method's implementation was done only for offline 2D SLAM.

References

[1] "The Past, Present and Future of SLAM | John Leonard | Tartan SLAM Series - YouTube." https://www.youtube.com/watch?v=FH6suW6_A5U&ab_channel=AirLab (accessed Feb. 02, 2022).

[2] les Earnest, "Stanford Cart," *https://web.stanford.edu/~learnest/sail/oldcart.html*, Dec. 2012. https://web.stanford.edu/~learnest/sail/oldcart.html (accessed Feb. 02, 2022).

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–108, 2006, doi: 10.1109/MRA.2006.1638022.

[4] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernández-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, no. September, pp. 2278–2283, 2005, doi: 10.1109/IROS.2005.1545511.

[5] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots-an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007, doi: 10.1016/j.robot.2006.12.008.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, pp. 225–234, 2007, doi: 10.1109/ISMAR.2007.4538852.

[7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011, doi: 10.1109/ICRA.2011.5979949.

[8] F. Dellaert, "Factor Graphs and {GTSAM}," *Technical Report*, no. GT-RIM-CP & R-2012-002, pp. 1–27, 2012, [Online]. Available: http://tinyurl.com/gtsam.

[9] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016, doi: 10.1109/TRO.2016.2624754.

[10] B. Mu, S. Y. Liu, L. Paull, J. Leonard, and J. P. How, "SLAM with objects using a nonparametric pose graph," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 4602–4609, 2016, doi: 10.1109/IROS.2016.7759677.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[13] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, “Probabilistic data association for semantic SLAM,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1722–1729, 2017, doi: 10.1109/ICRA.2017.7989203.