University of Passau
Lehrstuhl für Informatik mit Schwerpunkt Sensorik

Switchable Constraints For Robust Pose Graph

Master seminar talk

Abdelaziz Ben Othman

August 21, 2018

Plan

Motivation

Problematic

SLAM as a Non-Linear Squares Optimization Problem

Evaluation

Results and discussion

Summary

Motivation
State Of The Art Of SLAM

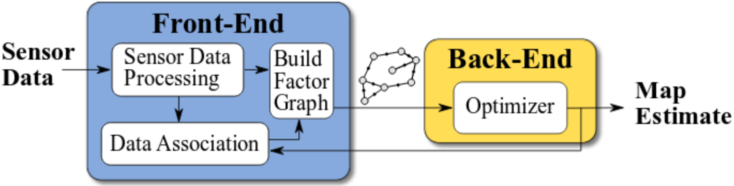


Figure: State of the Art Of SLAM

Source: [SP12]

Problematic

- Why SLAM is hard to solve?
  - Failed place recognition due to strong similarities
  - Problem of data association
  - Errors in sensor measurements occur

Figure: Different places with high similarities

Source: [SL04]

Problematic

- Data Association and their effects on the map
  - Introduction of false loop closure
  - Back-end performance degrades intensively
  - Slam fails

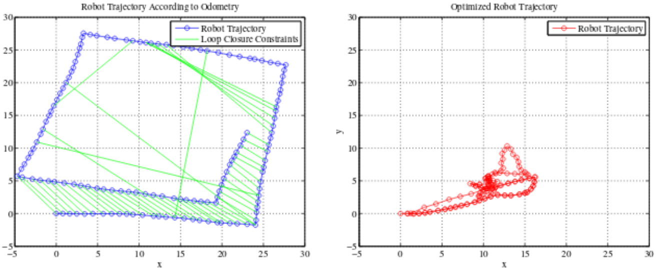


Figure: Failure when loop closures occur

Source: [Sün12]

SLAM as a Non-Linear Squares Optimization Problem

- We propose a robust back-end capable of detecting and eliminating the false loop-closure.
- The robust back-end will optimize the graph topology while solving it.
- The proposed robust back-end can cope with large number of false loop-closure in a variety of scenarios.

SLAM as a Non-Linear Squares Optimization Problem

Relation between successive robot poses

$$x_{i+1} = f(x_i, u_i) + w_i \tag{1}$$

where:

f $=$ a non-linear motion model
u_i $=$ the odometry measurement between the poses
w_i $=$ the noise of the odometry sensor system, $w_i \sim \mathcal{N}(0, \sum_i)$

$$x_{i+1} \sim \mathcal{N}(f(x_i, u_i), \textstyle\sum_i) \tag{2}$$

SLAM as a Non-Linear Squares Optimization Problem

To perform loop closing, robot has to recognize places it has visited before

$$x_j = f(x_i, u_{ij}) + \lambda_{ij} \tag{3}$$

where:

$$\begin{aligned} f &= \text{Sensor model function} \\ u_{ij} &= \text{odometry measurement between poses } x_i \text{ and } x_j \\ \lambda_{ij} &= \text{the noise of the odometry sensor system, } \lambda_{ij} \sim \mathcal{N}(0, \Lambda_{ij}) \end{aligned}$$

$$x_j \sim \mathcal{N}(f(x_i, u_{ij}), \Lambda_{ij}) \tag{4}$$

SLAM as a Non-Linear Squares Optimization Problem

Our main goal is to find the trajectory X of the robot given the odometry and the loop closure constraints $u_i, u_{ij} \in U$.

$$X^* = \underset{X}{\operatorname{argmax}} P(X|U) \tag{5}$$

We can factor the joint probability $P(X|U)$.

$$P(X|U) \propto \underbrace{\prod_i P(x_{i+1}|x_i, u_i)}_{\text{Odometry Constraints}} \cdot \underbrace{\prod_{ij} P(x_j|x_i, u_{ij})}_{\text{Loop Closure Constraints}} \tag{6}$$

SLAM as a Non-Linear Squares Optimization Problem

Under the assumptions of equation(2) we can write

$$P(x_{i+1}|x_i, u_i) = \frac{1}{\sqrt{2\pi|\sum_i|}} exp(-\frac{1}{2}(f(x_i, u_i) - x_{i+1})^T \sum_i^{-1}(f(x_i, u_i) - x_{i+1})) \tag{7}$$

Applying now the squared Mahalanobis distance definition

$$P(x_{i+1}|x_i, u_i) = \eta exp(-\tfrac{1}{2}\|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i}) \tag{8}$$

The same way, we can write the loop constraints closure gain

$$P(x_j|x_i, u_{ij}) = \eta exp(-\tfrac{1}{2}\|f(x_i, u_{ij}) - x_j\|^2_{\lambda_{ij}}) \tag{9}$$

SLAM as a Non-Linear Squares Optimization Problem

Having now equation (6),(8) and (9) we write

$$P(X|U) \propto \prod_i exp(-\frac{1}{2}\|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i}). \prod_{ij} exp(-\frac{1}{2}\|f(x_i, u_{ij}) - x_j\|^2_{\lambda_{ij}}) \tag{10}$$

We transform the products into sum by taking the negative logarithm:

$$-\log P(X|U) \propto \sum_i \|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i} + \sum_{ij} \|f(x_i, u_{ij}) - x_j\|^2_{\lambda_{ij}} \tag{11}$$

As consequence,

$$X^* = \operatorname*{argmin}_X - \log P(X|U) = \operatorname*{argmin}_X \sum_i \|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i} + \sum_{ij} \|f(x_i, u_{ij}) - x_j\|^2_{\lambda_{ij}} \tag{12}$$

SLAM as a Non-Linear Squares Optimization Problem

How to deal with suspicious loop closure?

- Idea: Remove edges corresponding to false loop closure
- Mathematically: introduce a binary weight w_{ij} that allows us to do so.

$$X^* = \operatorname*{argmin}_X \underbrace{\sum_i \| f(x_i, u_i) - x_{i+1} \|^2_{\sum_i}}_{\text{odometry Constraints}} + \underbrace{\sum_{ij} \| w_{ij}.f(x_i, u_{ij}) - x_j \|^2_{\lambda_{ij}}}_{\text{Loop Closure Constraints}} \quad (13)$$

SLAM as a Non-Linear Squares Optimization Problem

Binary weights w_{ij} themselves will be subject to optimization.
In this step we introduce, the switch variables s_{ij} and the switch function such that:

$$w_{i,j} = \Psi(s_{ij}) : \mathbb{R} \rightarrow \{0, 1\} \tag{14}$$

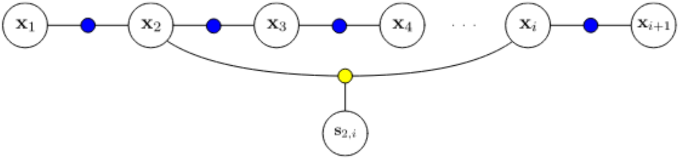


Figure: Introduction of the switch variables

Source: [Sün12]

Notice how the switch variables s_{2i} is governing the loop closure.

SLAM as a Non-Linear Squares Optimization Problem

After introducing the switch variables, the optimization problem now is augmented and can be written as:

$$S^*, X^* = \underset{X,S}{\operatorname{argmin}} \underbrace{\sum_{ij} \|\mathbf{f}(\mathbf{x_i}, \mathbf{u_i}) - \mathbf{x_{i+1}}\|^2_{\sum_i}}_{\text{Odometry Constraints}} + \underbrace{\sum_{ij} \|\psi(\mathbf{s_{ij}})(\mathbf{f}(\mathbf{x_i}, \mathbf{u_{ij}}) - \mathbf{x_j})\|^2_{\Lambda_{ij}}}_{\text{Switched Loop Closure Constraints}} \tag{15}$$

SLAM as a Non-Linear Squares Optimization Problem

Introducing the Switch Priors

- Idea: The switch variables must be initialized before starting the optimization.
- These initial values are the switch priors γ_{ij}
- Initially, accept all loop closure constraint, and let $\Psi(\gamma_{ij}) \approx 1$

$$s_{ij} \approx \mathcal{N}(\gamma_{ij}, \Xi_{ij}) \tag{16}$$

And Like we demonstrate previously, we can write

$$S^{*} = \underset{S}{\operatorname{argmin}} \sum_{ij} \|\gamma_{ij} - s_{ij}\|^{2}_{\Xi_{i}} \tag{17}$$

SLAM as a Non-Linear Squares Optimization Problem

Putting all together now we have the final optimization equation for the SLAM pose graph

$$S^*, X^* = \underset{X,S}{\operatorname{argmin}} \sum_{ij} \|\mathbf{f}(\mathbf{x_i}, \mathbf{u_i}) - \mathbf{x_{i+1}}\|^2_{\sum_i} + \sum_{ij} \|\psi(\mathbf{s_{ij}})(\mathbf{f}(\mathbf{x_i}, \mathbf{u_{ij}}) - \mathbf{x_j})\|^2_{\Lambda_{ij}} + \sum_{ij} \|\gamma_{ij} - s_{ij}\|^2_{\Xi_{ij}} \quad (18)$$

The evaluation part will answer these questions

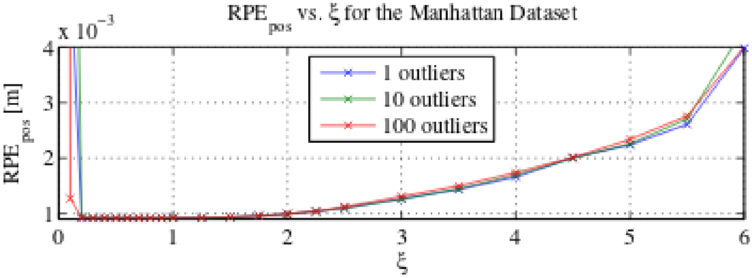
- How robust is the robust back-end?
- How much false loop closure can the robust back-end discard?
- What is the impact of the new implementation on the runtime?

- Quantitative Metrics
  - Root Mean Square (RMSE) and Relative Pose Error Metric (RPE): How much we deviate from the true trajectory.
- Qualitative Metrics
  - Precision-Recall: How well we discard false loop closure / keep True loop Closure.

We used 6 Datasets and we added to them false loop closure using different methods:

- Random Constraints
- Local Constraints
- Randomly Grouped Constraints
- Locally Grouped Constraints

- Goal: What is the influence of the initial value of the covariance Ξ_{ij} on the results.
- Method: Set $\Xi_{ij} = \zeta$ and run the implementation.
- Result: RPE is minimal for $1.5 > \zeta > 0.3$



[SP12]

Results and discussion
Robustness in presence of outliers

- Goal: How robust is the back-end against false loop closures
- Methodology: Inject 1000 outliers in our datasets
- Result: Our back-end is indeed robust

Dataset	max outl. ratio	min RPE_{pos}	max RPE_{pos}	median RPE_{pos}	incorrect solutions	success rate
Manhattan (g^2o)	47.6%	0.0009	0.0009	0.0009	0	100%
Manhattan (orig.)	47.6%	0.0009	5.9659	0.0009	1	99.8%
City 10000	9.4%	0.0005	0.0005	0.0005	0	100%
Sphere2500	40.8%	0.0953	0.0953	0.0964	1	99.8%
Intel	111.9%	0.2122	0.2122	0.2132	0	100%

Table: overall RPE_{pos} metric for the different datasets

Results and discussion
Qualitative robustness of our back-end

- Goal: How well does our implementation discard false loop closure and keep the true one?
- Methodology: Use the precision-recall metrics
- Results: The proposed back-end reaches almost optimal results.

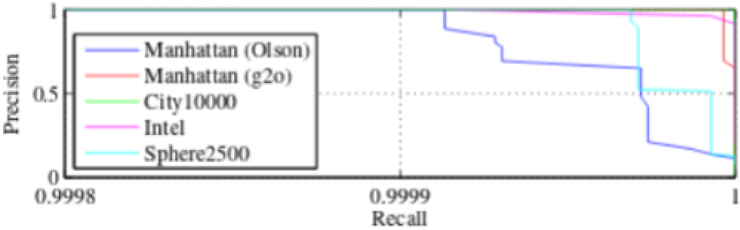


Figure: Precision-Recall for different datasets

Source: [SP12]

Results and discussion

- Goal: Does the implementation decrease the convergence time?
- Methodology: Run different implementations on a Core2-duo processor (2.4 Ghz)
- Result: The runtime depends strongly on the dataset, the policy and the number of injected outliers.

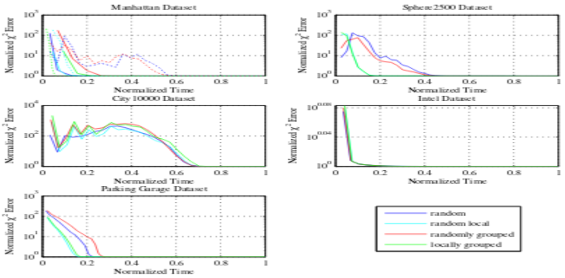


Figure: Convergence time vs χ^2 error

Source: [SP12]

- Fact: Two trials in the Sphere2500 and Manhattan(Olsons) failed.
- Reason of failure:
  - The Sphere2500 is composed by two misaligned sub-maps.
  - A false loop closure was not detected in the center in the Manhattan dataset.
- Lesson learned: Not deactivating a single loop closure leads to a global distortion of the graph.

Other failure we had using the Garage Dataset.

- Reason of failure:
  - Complex structure: Four parking decks (sparsely connected)
  - Few odometry Information
- Learned lesson: We need enough odometry information to generate a correct graph

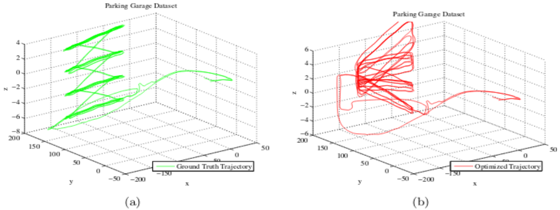


Figure: Ground truth vs obtained results

Source: [SP12]

Summary

- Explanation of the current state of the art of SLAM and its problems
- Proposition of our robust back-end solution
- Evaluation under many datasets and scenarios
- Explanation of the failed results

References I

[SL04] Computer Science and System Engineering Laboratory, *Loop Closure Detection*, 2004.

[SP12] Niko Sünderhauf and Peter Protzel, *Switchable constraints for robust pose graph slam*, Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 1879–1884.

[Sün12] Niko Sünderhauf, *Robust optimization for simultaneous localization and mapping*, Ph.D. thesis, Technischen Universitat Chemnitz, 2012.