Towards a Reliable SLAM Back-End

Gibson Hu, Kasra Khosoussi, Shoudong Huang

Centre for Autonomous Systems (CAS)
University of Technology Sydney

IROS 2013
November 4, 2013
Motivation

SLAM is now becoming a technology

- Mature Theory
- Efficient Algorithms
- Optimized (FOSS) Software
- Computing Power
## Motivation

### SLAM is now becoming a technology

- Mature Theory
- Efficient Algorithms
- Optimized (FOSS) Software
- Computing Power

### Reliability

- Outliers
- Consistency
- Local Optima and Divergence
Motivation

SLAM is now becoming a technology
- Mature Theory
- Efficient Algorithms
- Optimized (FOSS) Software
- Computing Power

Reliability
- Outliers
- Consistency
- Local Optima and Divergence
Problem Statement

- Under the assumption of Gaussian noise, the Maximum Likelihood (ML) estimate is the solution of a non-linear least squares problem (NLLS):

$$x_{ML}^* = \arg \min_x \sum_{(i,j) \in E} f_{ij}(x), \text{ in which } f_{ij}(x) \triangleq \|z_{ij} - h_{ij}(x_i, x_j)\|_{\Omega_{ij}}^2$$
Problem Statement

- Under the assumption of Gaussian noise, the Maximum Likelihood (ML) estimate is the solution of a non-linear least squares problem (NLLS):

\[ x_{\text{ML}}^* = \arg\min_x \sum_{(i,j) \in E} f_{ij}(x), \quad \text{in which} \quad f_{ij}(x) \triangleq \|z_{ij} - h_{ij}(x_i, x_j)\|^2_{\Omega_{ij}} \]

- Iterative Newton-based methods: Gauss-Newton, Levenberg-Marquardt
Problem Statement

- Under the assumption of Gaussian noise, the Maximum Likelihood (ML) estimate is the solution of a non-linear least squares problem (NLLS):

\[ x_{\text{ML}}^* = \arg \min_x \sum_{(i,j) \in E} f_{ij}(x), \quad \text{in which} \quad f_{ij}(x) \triangleq \| z_{ij} - h_{ij}(x_i, x_j) \|_{\Omega_{ij}}^2 \]

- Iterative Newton-based methods: Gauss-Newton, Levenberg-Marquardt
- They may converge to a local minimum, or even fail to converge at all
Problem Statement

• Under the assumption of Gaussian noise, the Maximum Likelihood (ML) estimate is the solution of a non-linear least squares problem (NLLS):

\[ x_{ML}^* = \arg \min_x \sum_{(i,j) \in E} f_{ij}(x), \quad \text{in which} \quad f_{ij}(x) \triangleq \|z_{ij} - h_{ij}(x_i, x_j)\|^2_{\Omega_{ij}} \]

• Iterative Newton-based methods: Gauss-Newton, Levenberg-Marquardt
• They may converge to a local minimum, or even fail to converge at all

A local minimum and the ML estimate (City 10000)
Problem Statement

- Under the assumption of Gaussian noise, the Maximum Likelihood (ML) estimate is the solution of a non-linear least squares problem (NLLS):

\[ x^*_{\text{ML}} = \arg \min_x \sum_{(i,j) \in E} f_{ij}(x), \quad \text{in which} \quad f_{ij}(x) \triangleq \| z_{ij} - h_{ij}(x_i, x_j) \|^2_{\Omega_{ij}} \]

- Iterative Newton-based methods: Gauss-Newton, Levenberg-Marquardt
- They may converge to a local minimum, or even fail to converge at all

A local minimum and the ML estimate (City 10000)

- Initial guess is usually obtained from the odometry data (dead reckoning)
Problem Statement

- Under the assumption of Gaussian noise, the Maximum Likelihood (ML) estimate is the solution of a non-linear least squares problem (NLLS):

\[ x^{\star}_{ML} = \arg \min_x \sum_{(i,j) \in E} f_{ij}(x), \text{ in which } f_{ij}(x) \triangleq \| z_{ij} - h_{ij}(x_i, x_j) \|_{\Omega_{ij}}^{2} \]

- Iterative Newton-based methods: Gauss-Newton, Levenberg-Marquardt
- They may converge to a local minimum, or even fail to converge at all

A local minimum and the ML estimate (City 10000)

- Initial guess is usually obtained from the odometry data (dead reckoning)

Initial guess must be sufficiently close to the global minimum
Bootstrapping Gauss-Newton

Don’t trust dead reckoning

In reality, even for a fixed low noise level, as the length of the traversed trajectory increases, \( \mathbb{E}[\|x^* - x_{odo}\|] \) becomes larger and larger.
**Bootstrapping Gauss-Newton**

---

**Don’t trust dead reckoning**

In reality, even for a fixed low noise level, as the length of the traversed trajectory increases, $\mathbb{E}[\|x^* - x_{odo}\|]$ becomes larger and larger

---

**Bootstrapping:**

finding an initial estimate for, e.g., Gauss-Newton, inside the basin of attraction of ML estimate
Bootstrapping Gauss-Newton

Don’t trust dead reckoning

In reality, even for a fixed low noise level, as the length of the traversed trajectory increases, $\mathbb{E}[||x^* - x_{odo}||]$ becomes larger and larger

Bootstrapping:

finding an initial estimate for, e.g., Gauss-Newton, inside the basin of attraction of ML estimate

- Linear Approximation Graph Optimization (LAGO): Carlone et al. [2011] (2D pose-graphs)
- Spanning Trees: Konolige et al. [2010] (pose-graphs)
- TORO*: Grisetti et al. [2007] (pose-graphs)
Bootstrapping Gauss-Newton

Don’t trust dead reckoning

In reality, even for a fixed low noise level, as the length of the traversed trajectory increases, $\mathbb{E}[\|x^* - x_{odo}\|]$ becomes larger and larger

Bootstrapping:

finding an initial estimate for, e.g., Gauss-Newton, inside the basin of attraction of ML estimate

- Linear Approximation Graph Optimization (LAGO): Carlone et al. [2011] (2D pose-graphs)
- Spanning Trees: Konolige et al. [2010] (pose-graphs)
- TORO*: Grisetti et al. [2007] (pose-graphs)

Fill the gap between $x_{odo}$ and $x^*$

Idea: A Sequence of Intermediate Optimization Problems $(\mathcal{P}_i)_{i=1}^{N}$

Find and solve $(\mathcal{P}_i)_{i=1}^{N}$ such that:

C1. $x_{odo}$ is within the basin of attraction of the solution of $\mathcal{P}_1$, i.e., $x^*_1$
C2. $x^*_i$ is within the basin of attraction of the solution of $\mathcal{P}_{i+1}$
C3. $x^*_N$ is within the basin of attraction of the original NLLS
Realizations

Option #1 (e.g., Incremental and Submmapping Approaches)

\[
x^*_k = \arg \min_x \sum_{(i,j) \in E_k} f_{ij}(x) \quad \text{for some} \quad E_k \subseteq E_{k+1} \subseteq E \quad (P_k)
\]
Realizations

Option #1 (e.g., Incremental and Submmapping Approaches)

\[ x^*_k = \arg \min_x \sum_{(i,j) \in E_k} f_{ij}(x) \quad \text{for some} \quad E_k \subseteq E_{k+1} \subseteq E \quad (P_k) \]

Option #2

\[ x^*_k = \arg \min_x \sum_{(i,j) \in E} w_{ij}^{(k)} f_{ij}(x) \quad \text{for some} \quad w_{ij}^{(k)} > 0 \quad (P_k) \]

- **Intuition:** Assign larger weights to those measurements that their corresponding residual at the current initial guess is smaller
- **Intuition:** Gradually incorporating new measurements into the process of optimization in order to control their sudden influence
Realizations

**Option #1 (e.g., Incremental and Submmapping Approaches)**

\[ x^*_k = \arg \min_x \sum_{(i,j) \in E_k} f_{ij}(x) \quad \text{for some} \quad E_k \subseteq E_{k+1} \subseteq E \quad (P_k) \]

**Option #2**

\[ x^*_k = \arg \min_x \sum_{(i,j) \in E} w_{ij}^{(k)} f_{ij}(x) \quad \text{for some} \quad w_{ij}^{(k)} > 0 \quad (P_k) \]

- **Intuition:** Assign larger weights to those measurements that their corresponding residual at the current initial guess is smaller

- **Intuition:** Gradually incorporating new measurements into the process of optimization in order to control their sudden influence
Realizations

Option #1 (e.g., Incremental and Submapping Approaches)

\[ x^{(k)} = \arg \min_x \sum_{(i,j) \in E_k} f_{ij}(x) \quad \text{for some} \quad E_k \subseteq E_{k+1} \subseteq E \quad (P_k) \]

Option #2

\[ x^{(k)} = \arg \min_x \sum_{(i,j) \in E} w^{(k)}_{ij} f_{ij}(x) \quad \text{for some} \quad w^{(k)}_{ij} > 0 \quad (P_k) \]

- **Intuition**: Assign larger weights to those measurements that their corresponding residual at the current initial guess is smaller

- **Intuition**: Gradually incorporating new measurements into the process of optimization in order to control their sudden influence

**Key Insight**

This situation is similar to M-estimators (ML-type estimators) when implemented as iterative re-weighted least squares (IRLS).
M-estimators

\[ \psi_{\alpha}(r) \triangleq \frac{r}{(1 + r^2)^{\alpha}} \quad , \quad w_{\alpha}(r) \triangleq \frac{\psi_{\alpha}(r)}{r} = \frac{1}{(1 + r^2)^{\alpha}} \]

- \( \alpha = 1 \): Cauchy
- \( \alpha = 2 \): Geman-McClure
- \( \alpha = \frac{1}{2} \): Huber
- \( \alpha = 0 \): Least Squares

Influence functions of different M-estimators
M-estimators

\[ \psi_\alpha(r) \triangleq \frac{r}{(1 + r^2)^\alpha} \], \quad w_\alpha(r) \triangleq \frac{\psi_\alpha(r)}{r} = \frac{1}{(1 + r^2)^\alpha} \]

- \( \alpha = 1 \): Cauchy
- \( \alpha = 2 \): Geman-McClure
- \( \alpha = \frac{1}{2} \): Huber
- \( \alpha = 0 \): Least Squares

Influence functions of different M-estimators
Algorithm 1: Cauchy Bootstrapper

**Input:** initial estimate from odometry: $x_{odo}$

**Output:** bootstrapper’s initial estimate: $x^{(0)}$

$\alpha \leftarrow 1$ // Cauchy M-estimator

$k \leftarrow 1$

$x^* (0) \leftarrow x_{odo}$

**repeat**

**foreach** edge $(i, j) \in E$ **do**

- Compute $r_{ij}^{(k-1)}$ using $x^*_{(k-1)}$

- $w_{ij}^{(k)} = w_\alpha (r_{ij}^{(k-1)})$

**end**

$x^*_{(k)} = \text{GaussNewton} \left( \sum_{(i, j) \in E} w_{ij}^{(k)} f_{ij}(x), x^*_{(k-1)} \right)$

$k \leftarrow k + 1$

**until** $\|w^{(k)} - w^{(k-1)}\| \leq \epsilon$

$x^{(0)}_{\text{Cauchy}} \leftarrow x^*_{(k)}$

**return** $x^{(0)}_{\text{Cauchy}}$
Evaluation

**Success Rate**

How many times in a series of Monte Carlo simulations bootstrapper+GN has converged to the optimal estimate?

- Compare our estimate with the GroundTruth+GN
- Look at the value of reduced (normalized) $\chi^2$ (for sufficiently large number of measurements it must be close to 1)
## Results

### Table: Convergence Rate (%) for 50 MC Simulations (Average of the obtained Reduced $\chi^2$)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Noise $(\sigma_x, \sigma_y, \sigma_\theta)$</th>
<th>CAUCHY+GN</th>
<th>LAGO+GN</th>
<th>TORO+GN</th>
<th>SpanningTree+GN</th>
<th>Odometry+GN</th>
<th>GT+GN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manhattan3500</td>
<td>0.05,0.05,0.05</td>
<td>100 (1.002)</td>
<td>50 (6.64)</td>
<td>100 (1.002)</td>
<td>100 (1.002)</td>
<td>50 (5.63)</td>
<td>(1.002)</td>
</tr>
<tr>
<td></td>
<td>0.1,0.1,0.1</td>
<td>100 (1.001)</td>
<td>2 (1.29e+4)</td>
<td>100 (1.001)</td>
<td>90 (1.073)</td>
<td>2 (5.04e+5)</td>
<td>(1.001)</td>
</tr>
<tr>
<td></td>
<td>0.2,0.2,0.2</td>
<td>98 (0.99)</td>
<td>0 (1.28e+4)</td>
<td>70 (1.14)</td>
<td>10 (2.11e+5)</td>
<td>0 (2.05e+3)</td>
<td>(0.99)</td>
</tr>
<tr>
<td></td>
<td>0.3,0.3,0.3</td>
<td>80 (90.03)</td>
<td>0 (4.60e+3)</td>
<td>40 (2.0e+2)</td>
<td>0 (1.05e+4)</td>
<td>0 (1.07e+3)</td>
<td>(0.99)</td>
</tr>
<tr>
<td></td>
<td>0.05,0.05,0.2</td>
<td>96 (1.046)</td>
<td>0 (1.16e+4)</td>
<td>74 (1.33e+2)</td>
<td>28 (3.21e+5)</td>
<td>0 (6.32e+5)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.2,0.2,0.05</td>
<td>100 (1.002)</td>
<td>46 (1.62e+3)</td>
<td>100 (1.002)</td>
<td>100 (1.002)</td>
<td>0 (3.81e+2)</td>
<td>(1.002)</td>
</tr>
<tr>
<td></td>
<td>0.1,0.1,0.1 (correlated)</td>
<td>86 (1.15e+3)</td>
<td>4 (1.63e+6)</td>
<td>0 (3.80e+3)</td>
<td>90 (1.03)</td>
<td>0 (8.28e+6)</td>
<td>(1.002)</td>
</tr>
<tr>
<td></td>
<td>0.2,0.2,0.2 (correlated)</td>
<td>78 (1.008)</td>
<td>0 (1.814e+4)</td>
<td>0 (2.15e+3)</td>
<td>12 (6.51e+5)</td>
<td>0 (1.26e+6)</td>
<td>(0.99)</td>
</tr>
<tr>
<td>City10000</td>
<td>0.05,0.05,0.05</td>
<td>96 (1.04)</td>
<td>2 (54.74)</td>
<td>94 (1.11)</td>
<td>100 (1.00)</td>
<td>0 (2.78e+2)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.1,0.1,0.1</td>
<td>100 (1.00)</td>
<td>0 (22.16)</td>
<td>82 (1.069)</td>
<td>94 (1.01)</td>
<td>0 (68.52)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.2,0.2,0.2</td>
<td>98 (1.00)</td>
<td>0 (7.50)</td>
<td>8 (1.23)</td>
<td>0 (1.29)</td>
<td>0 (19.3)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.3,0.3,0.3</td>
<td>92 (1.00)</td>
<td>0 (4.49)</td>
<td>0 (1.30)</td>
<td>0 (1.57)</td>
<td>0 (8.82)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.05,0.05,0.2</td>
<td>70 (1.31)</td>
<td>0 (16.44)</td>
<td>20 (8.1737)</td>
<td>4 (2.59)</td>
<td>0 (50.86)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.2,0.2,0.05</td>
<td>100 (1.00)</td>
<td>0 (96.90)</td>
<td>94 (1.026)</td>
<td>100 (1.00)</td>
<td>0 (361.32)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.1,0.1,0.1 (correlated)</td>
<td>92 (1.03)</td>
<td>0 (32.05)</td>
<td>0 (43.99)</td>
<td>96 (1.01)</td>
<td>0 (112.65)</td>
<td>(1.00)</td>
</tr>
<tr>
<td></td>
<td>0.2,0.2,0.2 (correlated)</td>
<td>90 (1.00)</td>
<td>0 (8.32)</td>
<td>0 (40.95)</td>
<td>0 (1.43)</td>
<td>0 (24.60)</td>
<td>(1.00)</td>
</tr>
</tbody>
</table>
Conclusion

- Odometry is not a reliable initial guess in many scenarios.
- Most of the existing bootstrapping techniques are limited to pose-graphs while our approach can handle 2D/3D pose/feature-graphs and bundle adjustment (look at the paper for some results).
- M-estimators, even the in the absence of outliers, may be useful as a bootstrapper for Gauss-Newton.
- Our algorithm improves the reliability of the state-of-the-art SLAM algorithms significantly (especially under high noise conditions) without changing the computational complexity.


Thank you for your attention
Which M-estimator?

**Table:** Success Rate (%) of $\psi_{\alpha}(\cdot)$ in 100 Monte Carlo Simulations for Different $\alpha$ Values in Manhatten Dataset

<table>
<thead>
<tr>
<th>Noise $(\sigma_x, \sigma_y, \sigma_\theta)$</th>
<th>$\alpha = 0.5$</th>
<th>$\alpha = 0.75$</th>
<th>$\alpha = 1$</th>
<th>$\alpha = 1.25$</th>
<th>$\alpha = 1.5$</th>
<th>$\alpha = 1.75$</th>
<th>$\alpha = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1,0.1,0.1)</td>
<td>54%</td>
<td>94%</td>
<td>100%</td>
<td>98%</td>
<td>78%</td>
<td>14%</td>
<td>4%</td>
</tr>
<tr>
<td>(0.2,0.2,0.2)</td>
<td>16%</td>
<td>94%</td>
<td>98%</td>
<td>88%</td>
<td>22%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>(0.3,0.3,0.3)</td>
<td>0%</td>
<td>58%</td>
<td>74%</td>
<td>60%</td>
<td>2%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Results Cont’d.

(a) MIT Killian Court (pose-graph)

(b) Victoria Park (feature-based)

(c) Malaga (bundle adjustment): GPS (green), Our method (blue)

Results of Cauchy+GN for different variants of SLAM