Robot Mapping

TORO – Gradient Descent for SLAM

Gian Diego Tipaldi, Wolfram Burgard
Stochastic Gradient Descent

- Minimize the error individually for each measurement (decomposition of the problem into sub-problems)
- Solve one step of each sub-problem
- Solutions might be contradictory
- The magnitude of the correction decreases with each iteration
- Learning rate to achieve convergence

[First used in the SLAM community by Olson et al., ’06]
Stochastic Gradient Descent

- Minimize the error individually for each measurement (decomposition of the problem into sub-problems)
- Solve one step of each sub-problem
- Solutions might be contradictory
- The magnitude of the correction decreases with each iteration
- Learning rate to achieve convergence

[First used in the SLAM community by Olson et al., ’06]
Stochastic Gradient Descent

- Minimize the error individually for each measurement (decomposition of the problem into sub-problems)
- Solve one step of each sub-problem
- Solutions might be contradictory
- The magnitude of the correction decreases with each iteration
- Learning rate to achieve convergence

[First used in the SLAM community by Olson et al., ’06]
Preconditioned SGD

- Minimize the error individually for each measurement
- Solve one step of each sub-problem
- A solution is found when an equilibrium is reached
- Update rule for a single measurement:

\[
x^{t+1} = x^t + \lambda H^{-1} J^T \Omega r_{ij}
\]
Node Parameterization

- How to represent the nodes in the graph?
- Impacts which parts need to be updated for a single measurement update
- Transform the problem into a different space so that:
  - the structure of the problem is exploited
  - the calculations become fast and easy

\[ x = g(p) \iff p = g^{-1}(x) \]

\[ x^* = \arg\min_x \sum_{i,j} e'_{ij}(x)^T \Omega_{ij} e'_{ij}(x) \]

Mapping function

Transformed problem
Parameterization of Olson

- Incremental parameterization:

\[ x_i = p_i - p_{i-1} \]

- Directly related to the trajectory

- **Problem:** to optimize a measurement between the nodes \( i \) and \( k \), one needs to update the nodes \( i, ..., k \) ignoring the topology of the environment
Alternative Parameterization

- Exploit the topology of the space to compute the parameterization
- Idea: “Loops should be one sub-problem”
- Such a parameterization can be extracted from the graph topology itself
Tree Parameterization

- How should such a problem decomposition look like?
Tree Parameterization

- Use a spanning tree!
Tree Parameterization

- Construct a spanning tree from the graph
- Mapping between poses and parameters

\[ X_i = P_{\text{parent}(i)}^{-1} P_i \]

- Error of a measurement in the new parameterization

\[ E_{ij} = \Delta_{ij}^{-1} \text{UpChain}^{-1} \text{DownChain} \]

Only variables along the path of a measurement are involved in the update
Stochastic Gradient Descent With The Tree Parameterization

- The tree parameterization leads to several smaller problems which are either:
  - measurements on the tree (“open loop”)
  - measurements not in the tree (“a loop closure”)
- Each SGD equation independently solves one sub-problem at a time
- The solutions are integrated via the learning rate
Computation of the Update Step

- 3D rotations are non-linear
- Update according to the SGD equation may lead to poor convergence
- SGD update:

\[ \Delta x = \lambda \mathbf{H}^{-1} \mathbf{J}_{ij}^T \Omega_{ij} \mathbf{r}_{ij} \]

- Idea: distribute a fraction of the residual along the parameters so that the error of that measurement is reduced
Computation of the Update Step

Alternative update in the “spirit” of the SGD: Smoothly deform the path along the measurements so that the error is reduced.

$P_i \triangleq_{ij} P_j$

Distribute the rotational error

Distribute the translational error
Rotational Error

- In 3D, the rotational error cannot be simply added to the parameters because the rotations are not commutative.
- Find a set of **incremental** rotations so that the following equality holds:

\[ R_1 R_2 \cdots R_n B = R'_1 R'_2 \cdots R'_n \]
Rotational Residual

- Let the first node be the reference frame
- We want a correcting rotation around a single axis
- Let $A_i$ be the orientation of the i-th node in the global reference frame

$$A'_n = A_n B$$
Rotational Residual

- Written as a rotation in global frame
  \[ A'_n = A_n B = QA_n \]
- with a decomposition of the rotational residual into a chain of incremental rotations obtained by spherical linear interpolation (slerp)
  \[ Q = Q_1 Q_2 \cdots Q_n \]
  \[ Q_k = \text{slerp}(Q, u_{k-1})^T \text{slerp}(Q, u_k) \quad u \in [0 \ldots \lambda] \]
- Slerp designed for 3d animations: constant speed motion along a circle
What is the SLERP?

- Spherical LinEar inteRPolation
- Introduced by Ken Shoemake for interpolations in 3D animations
- Constant speed motion along a circle arc with unit radius
- Properties:

\[
\begin{align*}
\mathcal{R}' &:= \text{slerp}(\mathcal{R}, u) \\
\text{axisOf}(\mathcal{R}') &= \text{axisOf}(\mathcal{R}) \\
\text{angleOf}(\mathcal{R}') &= u \text{ angleOf}(\mathcal{R})
\end{align*}
\]
Rotational Residual

- Given the $Q_k$, we obtain
  \[ A'_k = Q_1 \cdots Q_k A_k = Q_1:k A_k \]

- as well as
  \[ R'_k = A'_{k-1}^T A'_k \]

- and can then solve:
  \[
  \begin{align*}
    R'_1 &= Q_1 R_1 \\
    R'_2 &= (Q_1 R_1)^T Q_1:2 R_1:2 = R_1^T Q_1^T Q_1 Q_2 R_1 R_2 \\
    &\vdots \\
    R'_k &= [(R_{1:k-1})^T Q_k R_{1:k-1}] R_k
  \end{align*}
  \]
Rotational Residual

- Resulting update rule
  \[ R'_k = (R_{1:k-1})^T Q_k R_{1:k} \]

- It can be shown that the change in each rotational residual is bounded by
  \[ \Delta r'_{k,k-1} \leq |\text{angleOf}(Q_k)| \]

- This bounds a potentially introduced error at node k when correcting a chain of poses including k
How to Determine $u_k$?

- The $u_k$ describe the distribution of the error:
  \[ Q_k = \text{slerp}(Q, u_{k-1})^T \text{slerp}(Q, u_k) \quad u \in [0 \ldots \lambda] \]

- Consider the uncertainty of the measurements:
  \[ u_k = \min \left( 1, \lambda |P_{ij}| \right) \left[ \sum_{m \in P_{ij} \land m \leq k} d_m^{-1} \right]^{-1} \left[ \sum_{m \in P_{ij}} d_m^{-1} \right]^{-1} \]

  \[ d_m = \sum_{\langle l,m \rangle} \min \left[ \text{eigen}(\Omega_{lm}) \right] \]

  all measurements connecting $m$

- This assumes roughly spherical covariances!
Distributing the Translational Error

- That is trivial
- Just scale the x, y, z movements
Summary of the Algorithm

- Decompose the problem according to the tree parameterization

- Loop:
  - Select a measurement
    - Randomly or sample inverse proportional to the number of nodes involved in the update
  - Compute the nodes involved in update
    - Nodes according to the parameterization tree
  - Reduce the error for this sub-problem
    - Reduce the rotational error (slerp)
    - Reduce the translational error
Complexity

- In each iteration, the approach handles all measurements.
- Each measurement optimization requires to update a set of nodes (on average: the average path length according to the tree).

\[ \mathcal{O}(Ml) \]

# measurements  \[ \text{avg. path length} \]
(\text{parameterization tree})
Cost of a Measurement Update

\[ \approx \mathcal{O}(M \log(N)) \]
Node Reduction

- Complexity grows with the length of the trajectory
- Combine measurements between nodes if the robot is well-localized

\[ \Omega_{ij} = \Omega^{(1)}_{ij} + \Omega^{(2)}_{ij} \]

\[ z_{ij} = \Omega^{-1}_{ij} \left( \Omega^{(1)}_{ij} z^{(1)}_{ij} + \Omega^{(2)}_{ij} z^{(2)}_{ij} \right) \]

- Similar to adding rigid measurements
- Then, complexity depends on the size of the environment (not trajectory)
Simulated Experiment

- Highly connected graph
- Poor initial guess
- 2200 nodes
- 8600 measurements
Spheres with Different Noise

initialization

10 iterations

50 iterations

300 iterations

error/constraint

0 50 100 150 200 250 300

iteration

error/constraint

0 50 100 150 200 250 300

iteration

error/constraint

0 50 100 150 200 250 300

iteration
Mapping the EPFL Campus

- 10km long trajectory with 3D laser scans
Mapping the EPFL Campus
TORO vs. Olson’s Approach

Olson’s approach

1 iteration

10 iterations

50 iterations

100 iterations

300 iterations

TORO
TORO vs. Olson’s Approach

Graphs showing the comparison between different approaches:
- Olson’s approach
- Tree approach + node reduction
- Tree approach

Error per constraint vs. iteration for different noise levels:
- Olson’s approach (big noise)
- Tree approach (big noise)
- Olson’s approach (small noise)
- Tree approach (small noise)
Time Comparison

![Bar chart showing execution time per iteration for different algorithms and constraint counts.](image)

- Olson’s algorithm
- Olson’s algorithm, spheric covariances
- MLR
- Our approach
- Our approach with node reduction

**Axes:**
- Y-axis: Execution time per iteration [s]
- X-axis: Number of constraints

**Legend:**
- 3.7k
- 30k
- 64k
- 360k
- 720k
- 1.9M
Robust to the Initial Guess

- Random initial guess
- Intel dataset as the basis for 16 floors distributed over 4 towers

initial configuration  intermediate result  final result (50 iterations)
Drawbacks of TORO

- The slerp-based update rule optimizes rotations and translations separately
- It assumes \textit{roughly spherical covariance} ellipses
- Slow convergence speed close to minimum
- No covariance estimates
Conclusions

- TORO - Efficient maximum likelihood estimate for 2D and 3D pose graphs
- Robust to bad initial configurations
- Efficient technique for ML map estimation (or to initialize GN/LM)
- Works in 2D and 3D
- Scales up to millions of measurements
- Available at OpenSLAM.org
  http://www.openslam.org/toro.html
Literature

SLAM with Stochastic Gradient Descent

- Olson, Leonard, Teller: “Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates”
- Grisetti, Stachniss, Burgard: “Non-linear Constraint Network Optimization for Efficient Map Learning”
Slide Information

- These slides have been created by Cyrill Stachniss as part of the robot mapping course taught in 2012/13 and 2013/14. I created this set of slides partially extending existing material of Giorgio Grisetti, Wolfram Burgard, and myself.
- I tried to acknowledge all people that contributed image or video material. In case I missed something, please let me know. If you adapt this course material, please make sure you keep the acknowledgements.
- Feel free to use and change the slides. If you use them, I would appreciate an acknowledgement as well. To satisfy my own curiosity, I appreciate a short email notice in case you use the material in your course.
- My video recordings are available through YouTube: http://www.youtube.com/playlist?list=PLgnQpQtFTOGQrZ4O5Qzb1Hgl3b1JHimN&_feature=g-list

Cyrill Stachniss, 2014
cyrill.stachniss@igg.uni-bonn.de