Optimization in Factor Graphs for Fast and Scalable 3D Reconstruction and Mapping
Frank Dellaert, Robotics & Intelligent Machines @ Georgia Tech
Monte Carlo Localization

Dellaert, Fox, Burgard & Thrun, ICRA 1999

In the Smithsonian Institution's National Museum of American History and ON THIS WEB SITE!
SLAM = Simultaneous Localization and Mapping

FastSLAM: Particle Filter on Trajectories

Montemerlo, Thrun, Koller, & Wegbreit, AAAI 2002
2013: Full 3D LIDAR Mapping

Data/Movie by Nick Carlevaris-Bianco and Ryan Eustice (U. Michigan)
Large-Scale Structure from Motion

Photo Tourism
[Snavely et al. SIGGRAPH’06]

Photosynth
[Microsoft]

Iconic Scene Graphs
[Li et al. ECCV’08]

Build Rome in a Day
[Agarwal et al. ICCV’09]

Build Rome on a Cloudless Day
[Frahm et al. ECCV’10]

Discrete-Continuous Optim.
[Crandall et al. CVPR’11]
3D Models from Community Databases

- E.g., Google image search on “Dubrovnik”
3D Models from Community Databases

Agarwal, Snavely et al., University of Washington
http://grail.cs.washington.edu/rome/

5K images, 3.5M points, >10M factors
4D Reconstruction

Historical Image Collection

Grant Schindler
4D Reconstruction

Historical Image Collection

4D Cities: 3D + Time

Grant Schindler

trimensional
3D Scanner for iPhone
3D Reconstruction

4D Structure over Time
Outline

Intro

Factor Graphs

On Graphs & Matrices

Graph-Inspired Methods

Support Subgraphs

Future Directions

End
Acknowledgements

- Provided Materials: Michael Kaess & John Leonard (MIT), Ryan Eustice & Ed Olson (Michigan), Tim Barfoot (Toronto), Pratik Agarwal (Freiburg)
- Sponsors: NSF, DARPA, ONR, ARO/ARL, Microsoft Research, Samsung, Intel
- Noah Snavely, Sameer Agarwal, and Steve Seitz for cool data/videos
Boolean Satisfiability

$$(\neg M \lor I) \land (M \lor \neg I) \land (M \lor A) \land (\neg I \lor H) \land (\neg A \lor H) \land (\neg H \lor G)$$
Boolean Satisfiability

\((\neg M \lor I) \land (M \lor \neg I) \land (M \lor A) \land (\neg I \lor H) \land (\neg A \lor H) \land (\neg H \lor G)\)
Boolean Satisfiability

\[ (\neg M \lor I) \land (M \lor \neg I) \land (M \lor A) \land (\neg I \lor H) \land (\neg A \lor H) \land (\neg H \lor G) \]
Boolean Satisfiability

\[ (\neg M \lor I) \land (M \lor \neg I) \land (M \lor A) \land (\neg I \lor H) \land (\neg A \lor H) \land (\neg H \lor G) \]
Boolean Satisfiability

\[(\neg M \lor I) \land (M \lor \neg I) \land (M \lor A) \land (\neg I \lor H) \land (\neg A \lor H) \land (\neg H \lor G)\]
Constraint Satisfaction Problems
Graphical Models

Example 11. A Bayesian network (Pearl, 1988) or

4.1.1 Bayesian Networks

4.1 Representation

Figure 4.2: Bayes nets for example in Figure 4.1. The observed variables are shown as square nodes.

Figure 4.3: Factor graph for examples in Figure 4.2.
Continuous Probability Densities

\[ P(X,M) = k^* P(x_0) \prod_{i=1}^{M} P(x_i | x_{i-1}, u_i) \times \prod_{k=1}^{K} P(z_k | x_{i_k}, l_{j_k}) \]
Factor Graph -> Smoothing and Mapping (SAM)!
Factor Graph -> Smoothing and Mapping (SAM)!

Variables are multivariate! Factors are non-linear!
Variables are multivariate! Factors are non-linear!

Factors induce probability densities.

\[-\log P(x) = |E(x)|^2\]
Structure from Motion (Chicago, movie by Yong Dian Jian)

180 cameras, 88723 points
458642 projections
active camera: 4
Outline

Intro
Factor Graphs
On Graphs & Matrices
Graph-Inspired Methods
Support Subgraphs
Future Directions
End
Gaussian Factor Graph == $m \times n$ Matrix
Linear Least Squares

1129x334, 5917 nnz
Linear Least Squares

\[ A^T A \theta^* = A^T b \]
Linear Least Squares

\[ A^T A \theta^* = A^T b \]

\[ I \Delta A^T A = R^T R \]

334x334, 7992 nnz

1129x334, 5917 nnz

334x334, 9606 nnz
Square Root Factorization

\[
\begin{align*}
\text{ATA} & \rightarrow \text{Cholesky} \rightarrow R \\
A & \rightarrow \text{QR} \rightarrow R
\end{align*}
\]
Intel Dataset

(b) Final trajectory and evidence grid map. (c) Final R factor with side length 2730.

910 poses, 4453 constraints, 45s or 49ms/step
(Olson, RSS 07: iterative equation solver, same ballpark)
MIT Killian Court Dataset

(b) Final trajectory and evidence grid map.

(c) Final R factor with side length 5823.

1941 poses, 2190 constraints, 14.2s or 7.3ms/step
Variable Elimination

- Choose Ordering
- Eliminate one node at a time

Express node as function of adjacent nodes
Variable Elimination

• Choose Ordering
• Eliminate one node at a time

\[ P(l_1, x_1, x_2) = P(l_1|x_1, x_2)P(x_1, x_2) \]

Basis = Chain Rule! e.g. \[ P(l_1, x_1, x_2) = P(l_1|x_1, x_2)P(x_1, x_2) \]
• Choose Ordering
• Eliminate one node at a time

• Express node as function of adjacent nodes

Basis = Chain Rule! e.g. $P(l_1,x_1,x_2) = P(l_1|x_1,x_2)P(x_1,x_2)$
• Choose Ordering
• Eliminate one node at a time

• Express node as function of adjacent nodes
• Choose Ordering
• Eliminate one node at a time

• Express node as function of adjacent nodes
• Choose Ordering
• Eliminate one node at a time
• Express node as function of adjacent nodes
Choose Ordering

Eliminate one node at a time

Express node as function of adjacent nodes
Small Example

• End-Result = **Bayes Net**!
Elimination at work

100 vertices, 100 current edges, 0 final edges
Elimination at work

100 vertices, 180 edges
GTSAM 2: http://tinyurl.com/gtsam

- C++ library
- Open, BSD-licensed
- based entirely on Factor Graphs
- Optimization on Lie groups

```
NonlinearFactorGraph graph;
noiseModel::Diagonal::shared_ptr priorNoise =
  noiseModel::Diagonal::Sigmas(Vector_(3, 0.3, 0.3, 0.1));
graph.add(PriorFactor<Pose2>(1, Pose2(0, 0, 0), priorNoise));

// Add odometry factors
noiseModel::Diagonal::shared_ptr model =
  noiseModel::Diagonal::Sigmas(Vector_(3, 0.2, 0.2, 0.1));
graph.add(BetweenFactor<Pose2>(1, 2, Pose2(2, 0, 0), model));
graph.add(BetweenFactor<Pose2>(2, 3, Pose2(2, 0, M_PI_2), model));
graph.add(BetweenFactor<Pose2>(3, 4, Pose2(2, 0, M_PI_2), model));
graph.add(BetweenFactor<Pose2>(4, 5, Pose2(2, 0, M_PI_2), model));

// Add pose constraint
graph.add(BetweenFactor<Pose2>(5, 2, Pose2(2, 0, M_PI_2), model));
```
Elimination order is again CRUCIAL

Square Root SAM: Simultaneous Location and Mapping via Square Root Information Smoothing, Frank Dellaert, Robotics: Science and Systems, 2005
Exploiting Locality by Nested Dissection For Square Root Smoothing and Mapping, Peter Krauthausen, Frank Dellaert, and Alexander Kipp, Robotics: Science and Systems, 2006

• Order with least fill-in NP-complete to find
Approximate Minimum Degree

400 vertices, 1583 current edges, 0 final edges
Divide and Conquer

400 vertices, 1583 edges
St. Peter’s Basilica, Rome

Data Courtesy Microsoft Research

Nested Dissection (but good separators are hard)
GTSAM in Robotics

Tim Barfoot
University of Toronto
(in Oxford now!)

Frank Dellaert: Optimization in Factor Graphs for Fast and Scalable 3D Reconstruction and Mapping

Tim Barfoot
University of Toronto
(in Oxford now!)

Wednesday, September 11, 13
Example from real sequence:
Square root inf. matrix
Side length: 21000 variables
Dense: 1.7GB, sparse: 1MB

Victoria Park in Sydney, Australia
4 km trajectory
6968 frames, 140 landmarks

233499 non-zeros
≈ 0.1% density
≈ 11/column
iSAM [Kaess et al., TRO 08]

Michael Kaess

**Incremental Smoothing and Mapping (iSAM)**
- Uses incremental matrix factorization
- Based on “Givens rotations”
- periodic Relinearization, re-ordering
- Then: Victoria Park comfortably real-time
Georgia Tech & MIT
Kaess et al., IJRR 12
iSAM2: Bayes Tree for Manhattan Sequence

Georgia Tech & MIT
Kaess et al., IJRR 12
Notable Applications of iSAM

LG Electronics LSM-100
Scanner Mouse
Notable Applications of iSAM

LG Electronics LSM-100 Scanner Mouse

Spheres miniature satellites onboard ISS
Notable Applications of iSAM

LG Electronics LSM-100 Scanner Mouse

Spheres miniature satellites onboard ISS

Ship Hull Inspection / U.S. Navy
August 2013, Work by Ryan Eustice et al., University of Michigan & MIT
Long-term Visual Mapping

Reduced pose graph [Johannsson, Kaess, Fallon, Leonard, ICRA 13]
– Reuses existing poses
– Grows with explored space, not time

MIT Stata Center Dataset (publicly available)
– Duration: 6 months
– Operation time: 9 hours
– Distance travelled: 11 km (about 7 miles)
– VO keyframes: 630K
Reduced Pose Graph – 10 Floors
Reduced Pose Graph – 10 Floors

iSAM optimizes reduced pose graph, Video by Michael Kaess
Reduced Pose Graph – 10 Floors

iSAM optimizes reduced pose graph, Video by Michael Kaess
Reduced Pose Graph

Map of 10 floors

- Accelerometer used to detect elevator transitions
- iSAM optimizes RPG to achieve real-time
Very long-term Mapping

Long-Term Simultaneous Localization and Mapping with Generic Linear Constraint Node Removal
Nick Carlevaris-Bianco and Ryan Eustice (U. Michigan), IROS 2013
Of Eiffel Towers....
Of Eiffel Towers....
Iterative Methods

- Better than direct methods for large problems because they
  - Perform only simple operations and no variable elimination
  - Require constant (linear) memory

- But they may converge slowly when the problem is ill-conditioned

- The conjugate gradient method: The convergence speed is related to the condition number $\kappa(A^tA) = \lambda_{max}/\lambda_{min}$
Main Idea (with Viorela Ila and Doru Balcan)

The whole problem  =  The easy part (subgraph)  +  The hard part (loop closures)
Linear Reparametrization is Preconditioning

![Original Problem]

\[ Ax = b \]

- \( y = Rx \)

### New Problem

\[ AR^{-1}y = b \]

- The preconditioned conjugate gradient (PCG) method will converge faster if the new problem becomes well-conditioned
\[ \lambda_{\text{max}}(\Sigma) = 1.6e+01, \kappa(\Sigma) = 1.6e+06 \]
Preconditioner

\[ \lambda_{max}(\Sigma) = 4.3 \times 10^2, \quad \kappa(\Sigma) = 5.3 \times 10^6 \]
Preconditioned, max

\[
\lambda_{\text{max}}(\Sigma) = 1.0e+00, \ k(\Sigma) = 2.7e+04
\]
Illustration with Real Data

- Solution to a sparse subgraph
  - An approximated solution
  - Efficient to compute

- Solution to the entire graph
  - The optimal solution
  - Expensive to compute directly

Yong-Dian Jian, Doru C. Balcan and Frank Dellaert

*Generalized Subgraph Preconditioners for Large-Scale Bundle Adjustment*
Proceedings of 13th International Conference on Computer Vision (ICCV), Barcelona, 2011
Notre Dame

Solution from Subgraph

Solution from the Original Graph

Data from Noah Snavely
Piazza San Marco

Solution from Subgraph

Solution from the Original Graph

Data from Sameer Agarwal
The condition numbers of the preconditioned systems

<table>
<thead>
<tr>
<th>source</th>
<th>Original</th>
<th>SPCG</th>
<th>Jacobi</th>
<th>GSP-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-15</td>
<td>5.58e+21</td>
<td>1.87e+06</td>
<td>5.94e+04</td>
<td>4.36e+03</td>
</tr>
<tr>
<td>V-02</td>
<td>6.54e+21</td>
<td>6.46e+09</td>
<td>6.35e+05</td>
<td>1.38e+05</td>
</tr>
<tr>
<td>F-01</td>
<td>3.68e+11</td>
<td>1.92e+08</td>
<td>7.54e+06</td>
<td>8.71e+05</td>
</tr>
</tbody>
</table>

The original condition numbers are very large

GSP-3 has consistently smaller condition numbers
Intro
Factor Graphs
On Graphs & Matrices
Graph-Inspired Methods
Support Subgraphs
Future Directions
End
• Brought to you by DARPA
• Fully multi-threaded

Robust Vision-Aided Navigation Using Sliding-Window Factor Graphs, Han-Pang Chiu, Stephen Williams, Frank Dellaert, Supun Samarasekera, Rakesh Kumar, ICRA ’13

Information Fusion in Navigation Systems via Factor Graph Based Incremental Smoothing, Vadim Indelman, Stephen Williams, Michael Kaess, and Frank Dellaert, RAS, 2013

Frank Dellaert: Optimization in Factor Graphs for Fast and Scalable 3D Reconstruction and Mapping
Backbone Methods

• Long-term, GPS-denied (Vision Aided) Navigation
  – long chains, removed from start
  – loss of numerical accuracy -> blow-up

• Solution
  – revive Ed Olson’s idea: backbone
  – Now backbone is 100Hz IMU stream

• Variants:
  – Backbone preconditioning
  – Backbone iSAM
  – Backbone SGD (Olson !)
Who needs correspondence...

• When you have robust error metrics?

Proposed by Ed Olson and Pratik Agarwal at the 1st Workshop on Robust and Multimodal Inference in Factor Graphs, ICRA 2013
Distributed SLAM (and SFM ?)

GTSAM: SAM + MPC

• SAM and Model-Predictive Control, side by side in GTSAM
• Fully Lie-algebra based, collaboration w. Marin Kobilarov at JHU
GTSAM: SAM + MPC

• SAM and Model-Predictive Control, side by side in GTSAM
• Fully Lie-algebra based, collaboration w. Marin Kobilarov at JHU
GTSAM: SAM + MPC

• SAM and Model-Predictive Control, side by side in GTSAM
• Fully Lie-algebra based, collaboration w. Marin Kobilarov at JHU