From Square Root SAM to GTSAM: Factor Graphs in Robotics

Frank Dellaert, Georgia Institute of Technology
Michael Kaess, Carnegie Mellon University
5 years ago I did a sabbatical at a startup to help them build the most advanced flying AI on the planet: the Skydio drone.
Earlier this year the Skydio 2 was released, which innovates in both 360 perception and superior autonomy.
Skydio 2
To deliver value, the autonomy stack has to support superior navigation, tracking, and motion planning at very low power.
Many of these at their core are optimization problems with locality, which is captured well by *factor graphs*. 

Diagram: 
- Robot 
- Landmark \( m_1 \) 
- Landmark measurement
Many of these at their core are optimization problems with locality, which is captured well by factor graphs.

Odometry measurement

Robot

Landmark measurement

Landmark 1

Landmark 2
Many of these at their core are optimization problems with locality, which is captured well by factor graphs.
Factor graphs can represent many robotics problems, from tracking to optimal control to sophisticated 3D mapping.
Factor graphs expose opportunities for raw speed because of the deep connection with sparse linear algebra.

- Ordering heuristics
- Nested Dissection
- Sparsification
- Pre-integration
- Iterative Solvers
- Incremental Inference and the Bayes tree
Factor graphs are beneficial in designing and thinking about your problem, even aside from performance.
Outline:
From SAM to GTSAM
Navigation and Mapping
Pushing the Boundaries
New Frontiers
Outlook
In SAM we are interested in inferring the trajectory of the robot and a map of the unknown environment.
The factor graph associated with a small SAM problem instantaneously shows the structure of the problem.
In practice, Square Root SAM is implemented using sparse matrix factorization, which is a computation on a graph.

QR Factorization on Factor Graph

\[
A \xrightarrow{QR} R
\]
Visual SLAM in 2005 might have looked a bit cheesy, but we already did 8-camera visual SLAM back then.

We used non-vacuum iRobots.
The key points from the Square Root SAM papers have stood the test of time, but we know so much more now

- Key points:
  - Matrices ↔ Graphs
  - Factorization ↔ Variable Elimination
  - Improving Performance ↔ Variable Ordering

- What we know now:
  - Factor graphs can represent many robotics problems
  - Factor graphs expose opportunities to improve computational performance
  - Factor graphs are beneficial in designing and thinking about your problem, even aside from performance
GTSAM embodies many of the ideas we and others have developed around factor graphs since then:

- C++ library: gtsam.org
- Python & Matlab wrappers
- Open-source, BSD-licensed
- Optimization on Manifolds and Lie groups
- Reverse AD Expression Language
Outline:
From SAM to GTSAM
Navigation and Mapping
Pushing the Boundaries
New Frontiers
Outlook
Square Root SAM on real sequence, the Sydney Victoria Park dataset, shows how sparsity is key to performance.
While finding an optimal ordering is NP complete, heuristics coupled with domain knowledge can do wonders.
Domain knowledge often shows how to break up graphs, a generalization of “nested dissection” [Kai Ni et al. IROS ’10]

Now CEO of Beijing autonomous driving startup
Hyper-SFM applies hierarchical nested dissection to the structure from motion problem [Ni et al. 3DIMPVT’12]
Breaking up graphs can lead to powerful new paradigms for distributed mapping [Alex Cunningham et al, ICRA ’13]
The Bayes tree is a powerful graphical model that enables incremental Smoothing and Mapping (iSAM) [IJRR ‘12]

Exploit the fact that the square root information matrix can be understood as a directed junction tree: the Bayes tree
iSAM edits a Bayes tree as new measurements arrive.
iSAM2 at work on a synthetic sequence really shows off the reduction in amortized costs afforded by the Bayes tree
iSAM has been applied in many applications, from mapping aircraft carriers [Kim et al 2013] to experiments on the ISS.
Pre-integrating IMU measurements yields state of the art visual-inertial navigation [Forster et al. TRO’17]

- VIO pre-integrated IMU
- Integrates IMU measurements between poses, subtracting gravity
- Efficient and accurate!
Sparsification in visual-inertial navigation strikes a perfect balance between efficiency and accuracy.

1. Extract Markov Blanket
2. Marginalize
3. Sparsify
4. Reinsert
Outline:

From SAM to GTSAM

Navigation and Mapping

Pushing the Boundaries

New Frontiers

Outlook
MIT’s Kimera is a state of the art metric-semantic SLAM built upon factor graphs and GTSAM [Rosinol et al. ICRA ‘20]

- Four modules:
  – VIO pre-integrated IMU
  – Robust factor-graph-based pose graph
  – Real-time meshing module
  – Semantics module fuses semantic 2D information into the 3D mesh representation.
Rosinol et al. will present the impressive Dynamic Scene Graphs here at RSS, which builds upon Kimera
Czarnowski et al. [RAL ’19] integrated deep VAEs into factor graphs to build a real-time, dense SLAM system.

- Real-time dense SLAM system
  - Variational auto-encoder (VAE)
  - Compact “latent” codes
  - Codes are unknowns in a iSAM-based SLAM system
Factor graphs have been used in humanoid state estimation at University of Michigan... [Hartley et al. IROS ’18] (1/2)

- Fuse inertial with visual and domain specific knowledge about legged robots.
- Forward kinematics (FK).
- Pre-integrated contact factors, which integrate foot contacts.
...and at Oxford for fusing visual odometry and quadruped state estimation [Wisth et al. ’19-20]
Factor graphs turn out to be an excellent framework in which to innovate in motion planning [Mukadam et al. IJRR ’18]

- Factors for:
  - Overall task-related objective
  - Gaussian Process motion prior factors
  - Obstacle avoidance, joint limits, etc…
- Fast incremental replanning using the Bayes Tree
We used factor-graph-based motion planning to plan artistic action such as robot calligraphy [Wang et al. IROS ’20]
We used factor graphs to encode robot dynamics and applied to kino-dynamic motion planning [Xie et al. '20]

- **Recipe:**
  - Take Lynch & Park modern dynamics formulation
  - Turn into factor graph
  - Optimize with sparse (incremental) solvers
STEAP does both: simultaneous trajectory estimation & (motion) planning [Mukadam Auro‘18]
Outline:

From SAM to GTSAM

Navigation and Mapping

Pushing the Boundaries

New Frontiers

Outlook
Hard constraints for Bayes tree significantly expand iSAM2 capabilities [Sodhi et al 2020]
Factor graphs support modeling ambiguous situations, but what about inference?

Noisy odometry: 1m

Option 1
0.5m

Option 2
1.5m

Multi-mode factor
Non-Gaussian inference using nonparametric belief propagation on the Bayes tree [Fourie et al 2016]
For Gaussian problems with ambiguity, multi-hypothesis tracking run multiple parallel instances of inference

- One Bayes tree per hypothesis
- Exponential growth: Pruning

Total $2^6 = 64$ Bayes trees
Multi-hypothesis Bayes tree saves computation by avoiding redundant computation [Hsiao et al 2019]

Each clique is associated with one Hypo-layer
Multi-hypothesis RGB-D mapping avoids wrong decisions and can provide multiple plausible solutions.
Knowledge about ambiguity is useful for planning including active SLAM [Hsiao et al 2020]
Is loopy belief propagation on factor graphs a better match to the hardware of the future [Davison et al 2020]?

- Factor graphs on a graph processor
- Loopy belief propagation
- Well suited for parallel hardware
- CVPR: 30x faster SfM!
Working on Square Root SAM 15 years ago we had no crystal ball, but we certainly imagined more robots around
The outlook for airborne autonomy is relatively positive, as the airspace environment is the easiest to conquer.

- Planning state space is just 6D
- The airspace is relatively uncluttered
- Skydio has show convincing results
- Efforts by NASA/DARPA to assure safe airspace
- Crashes of light-weight drones are probably non-lethal
The timeline for self-driving cars is less clear, because of the “long tail”, bugs, and their possibly lethal consequences.
Factor graphs and GTSAM have been used in several autonomous driving companies, e.g., Zoox, Holomatic
We started a non-profit initiative, OpenSAM.org, to advance certifiable factor graphs for embedded applications

- GTSAM for back-office and Research
- OpenSAM: reference implementation for embedded systems
  - Collaboration with Holomatic and other companies…
  - Goal: fast, certifiable code for a subset of GTSAM functionality
  - Looking for industry memberships/collaborations!
The most difficult environment to deploy robots in is the home, because of perception/manipulation/HRI.

- Perception is very challenging due to clutter, occlusion…
- Manipulation in those environments is yet unsolved
- Expectations of people are mismatched
Our new effort, SwiftFusion, is focused on combining estimation and optimal control with the data-driven revolution

- Google collaboration: SwiftFusion
  - Seamless integration with TensorFlow
  - Fast, automatically differentiated factors
  - Sparse factor graphs and dense tensor processing in one language
- Which will enable:
  - Combine probabilistic estimation and optimal control with data-driven factors
- Collaborators wanted, DM me @fdellaert

https://github.com/borglab/SwiftFusion
All of this was only possible by amazing collaborations over the years, in academia, on github.com, and industry.
References: Navigation and Mapping


Kai Ni, and Frank Dellaert, HyperSfM, 2012. IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)


References: Pushing the boundaries


References: New Frontiers


Questions?