SLAM: Sequential Estimation

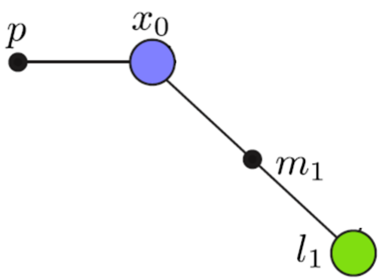
Robot Localization and Mapping
16-833

Michael Kaess

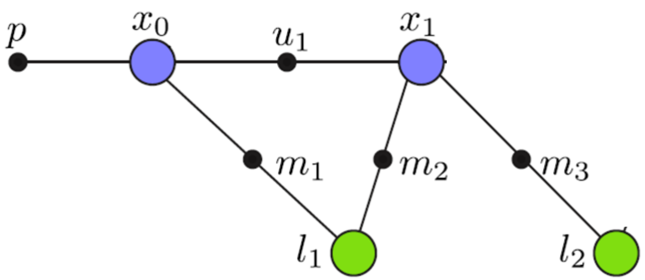
November 4, 2024

SLAM is a Sequential Estimation Problem!

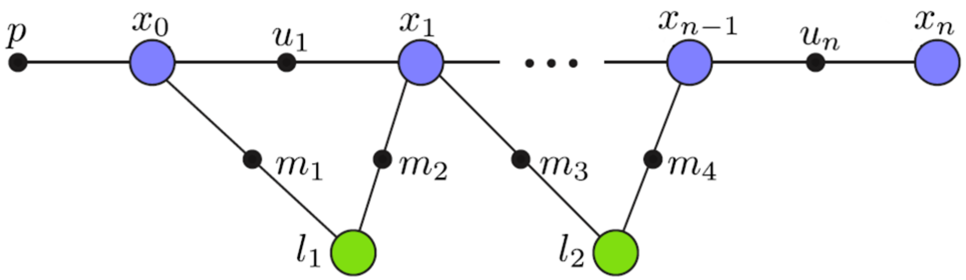
t=0



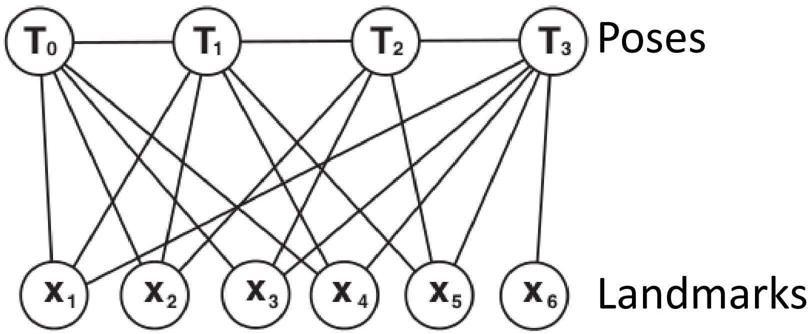
t=1



t=n-1



Full SLAM (Computer Vision: Bundle Adjustment)

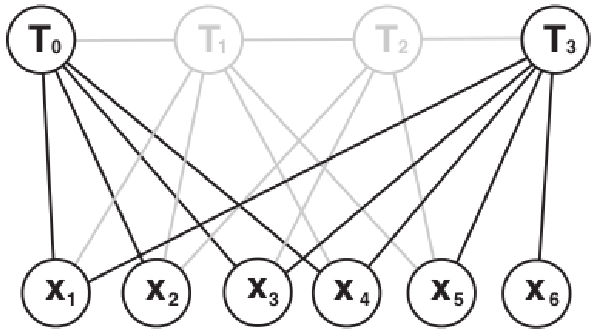


From Strasdat et al, 2011 IVC
“Visual SLAM: Why filter?”

- Graph grows with time:
  - Have to solve a sequence of increasingly larger problems
  - Will become too expensive even for sparse Cholesky

F. Dellaert and M. Kaess, “Square Root SAM: Simultaneous localization and mapping via square root information smoothing,” IJRR 2006

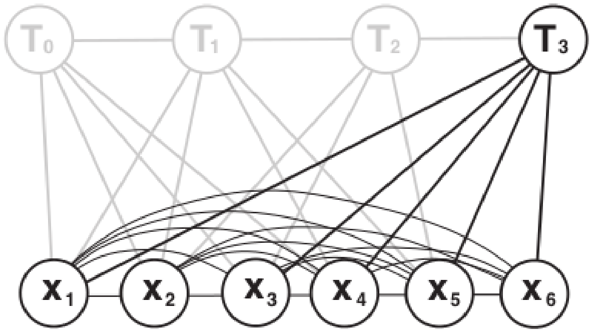
Keyframe SLAM



- Drop subset of poses to reduce density/complexity
- Only retain “keyframes” necessary for good map

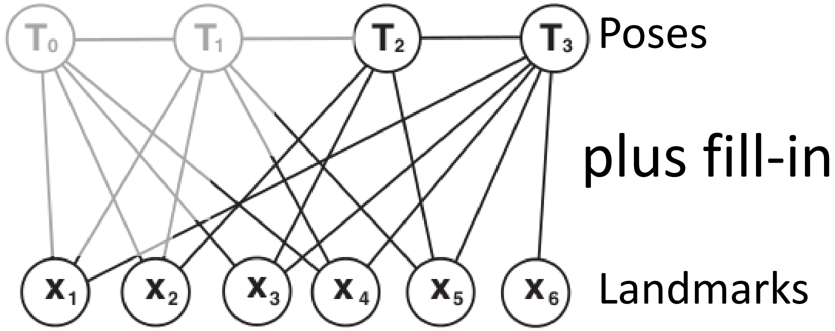
- Complexity still grows with time, just slower

Filter



- Keyframe idea not applicable: map would fall apart
- Instead, marginalize out previous poses
  - Extended Kalman Filter (EKF)
- Problems when used for SLAM:
  - All landmarks become fully connected -> **expensive**
  - Relinearization not possible -> **inconsistent**

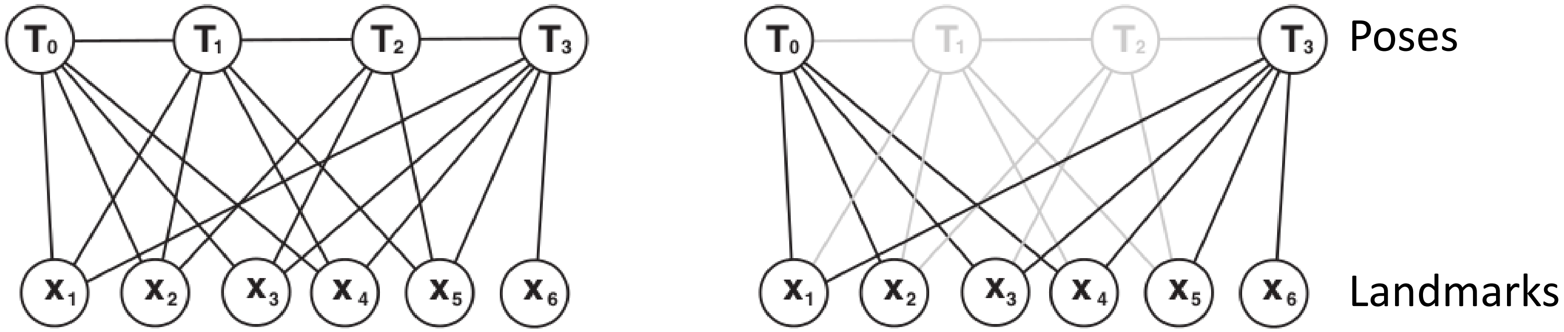
Fixed-lag Smoothing



- Marginalize out all but last n poses and connected landmarks
  - Relinearization possible
- Linear case
- Nonlinear (with some restrictions)

Is Cheap and Exact Achievable?

- Back to full BA and keyframes:



- New information is added to the graph
- Older information does not change
- Can be exploited to obtain an efficient solution!

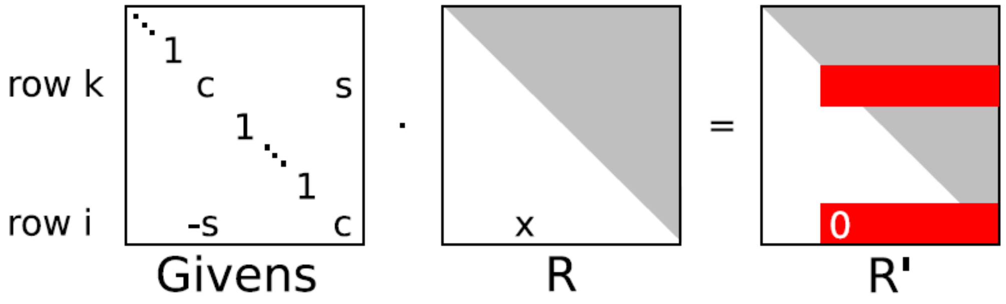
Incremental Smoothing and Mapping (iSAM)

Solving a growing system:

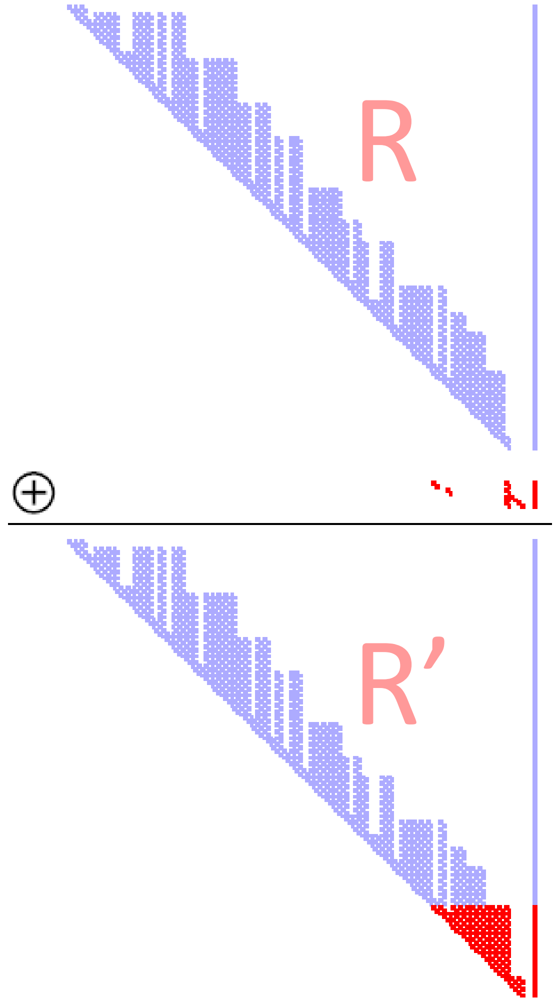
– R factor from previous step
– How do we add new measurements?

Key idea:

– Append to existing matrix factorization
– "Repair" using Givens rotations



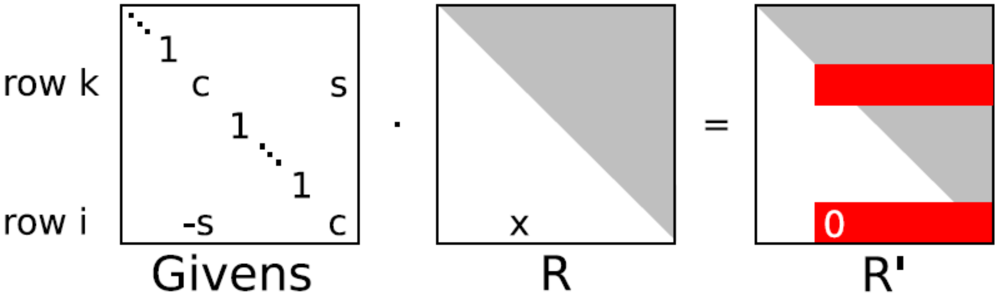
New measurements ->



QR Factorization: Householder Reflections

- On the board

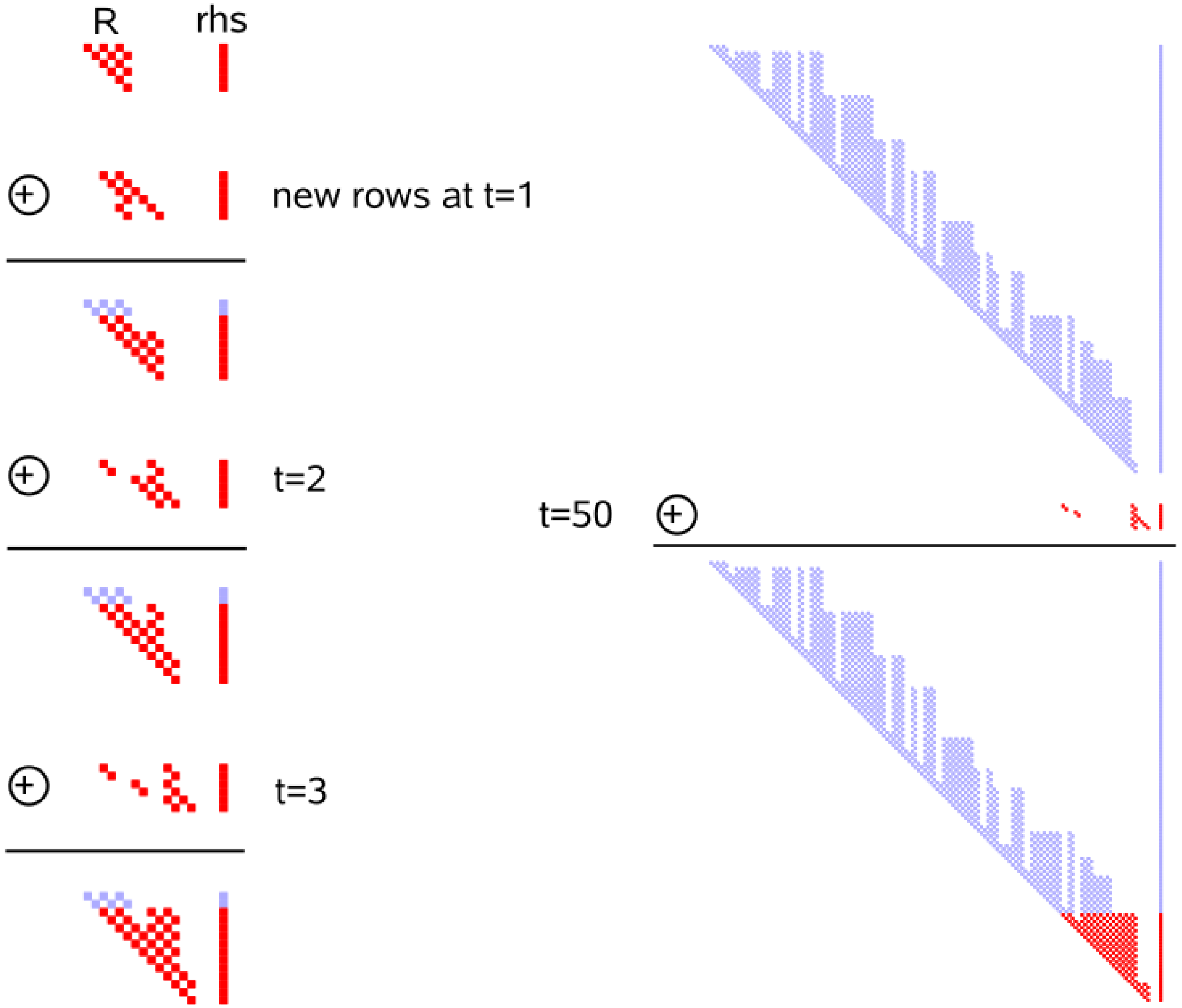
Givens Rotations



$$
(\cos\phi, \sin\phi) = \begin{cases} (1,0) & \text{if } \beta = 0 \\ \left(\frac{-\alpha}{\beta\sqrt{1+\left(\frac{\alpha}{\beta}\right)^2}}, \frac{1}{\sqrt{1+\left(\frac{\alpha}{\beta}\right)^2}} \right) & \text{if } |\beta| > |\alpha| \\ \left(\frac{1}{\sqrt{1+\left(\frac{\beta}{\alpha}\right)^2}}, \frac{-\beta}{\alpha\sqrt{1+\left(\frac{\beta}{\alpha}\right)^2}} \right) & \text{otherwise} \end{cases}
$$

where $\alpha := a_{kk}$ and $\beta := a_{ik}$.

iSAM Updates



[Kaess et al., TRO 08]

Incremental Smoothing and Mapping (iSAM)

Update and solution are O(1)

Are we done?

SLAM is nonlinear…

iSAM requires periodic batch factorization to relinearize

Also: loop closures cause fill-in!

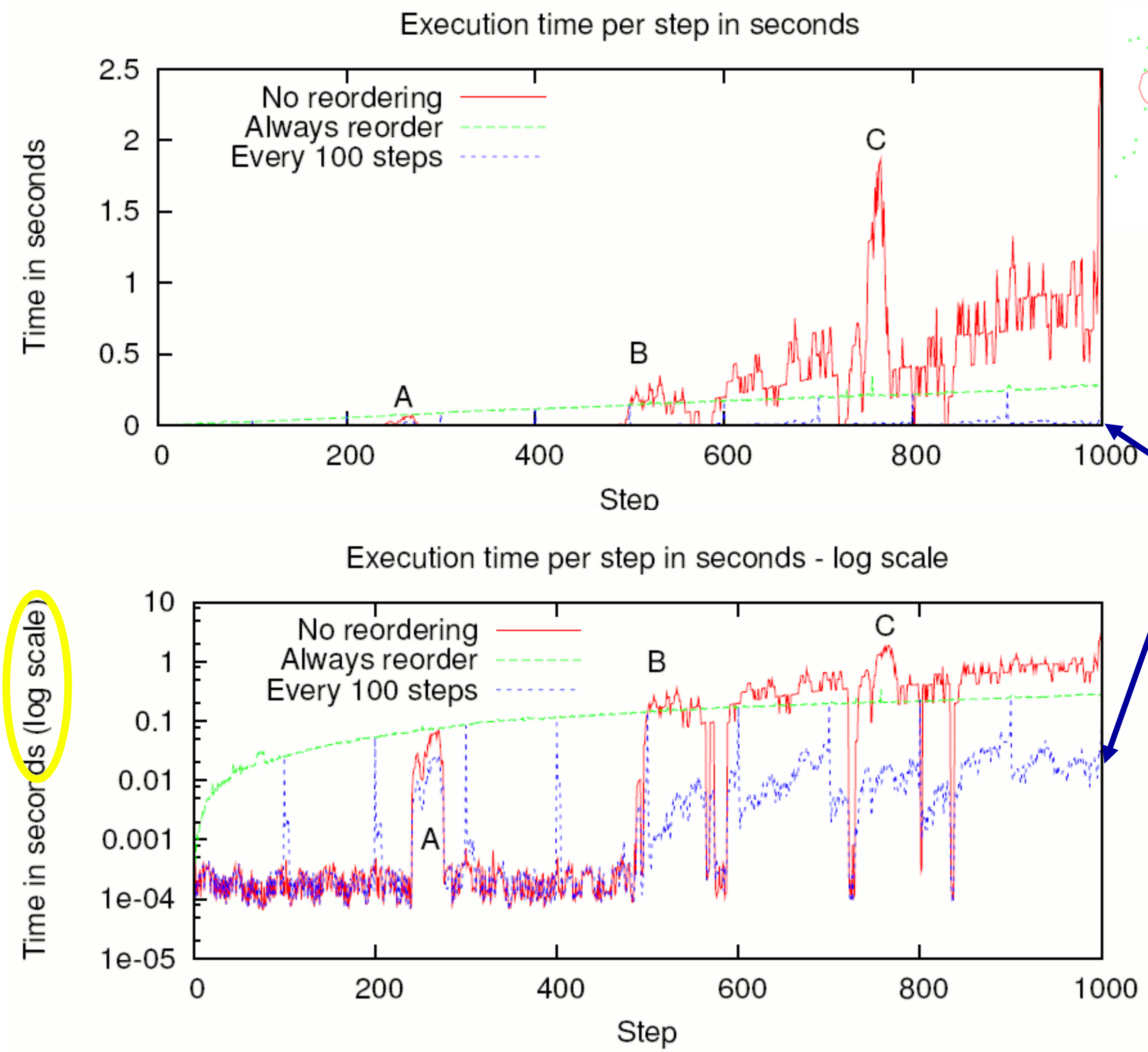
Loops and Periodic Variable Reordering

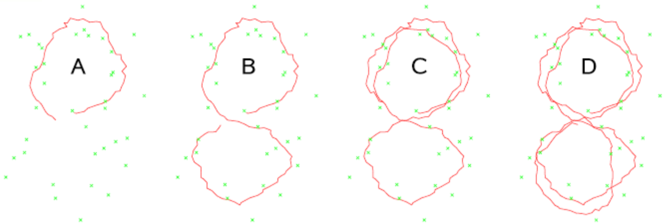
Factor R for 500 frames (n=1579)

No variable reordering

Variable reordering COLAMD

Periodic Variable Reordering – Timing





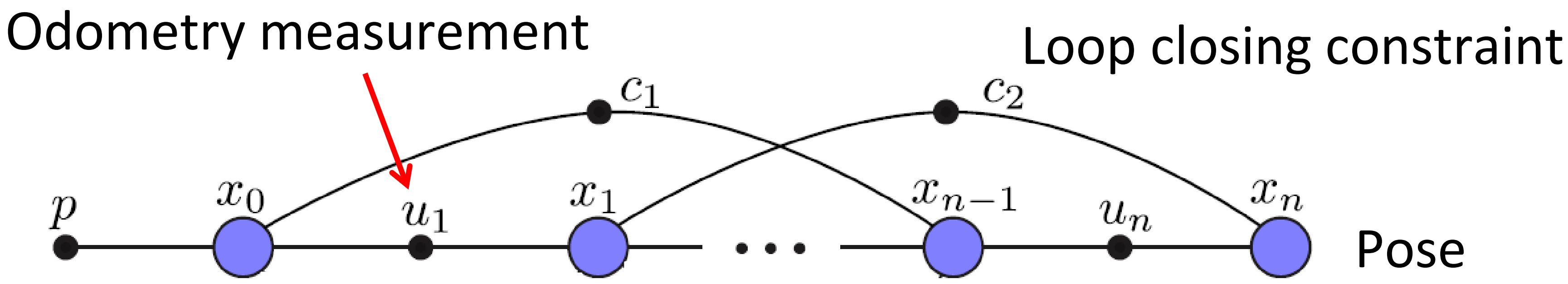
Combines advantages of incremental SAM and batch reordering

Live iSAM Example

Open-source code at: https://people.csail.mit.edu/kaess/isam/

Newer version (iSAM2, based on graphical models, discussed in next lectures) is part of GTSAM library: https://github.com/borglab/gtsam

Pose Graph SLAM - Scalability



Smoothing: Grows unboundedly in time

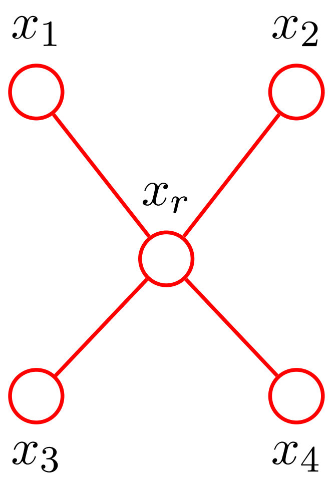
Should only depend on explored space

Solution: Reduced Pose Graph

Johannsson, Kaess, Fallon, Leonard (ICRA 13)

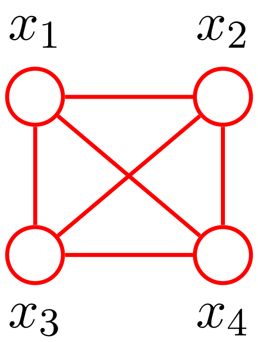
Pose Graph Reduction

Reduction by marginalization



Remove x_r

$$p(x_1, x_2, x_3, x_4) = \int p(x_1, x_2, x_3, x_4, x_r) dx_r$$

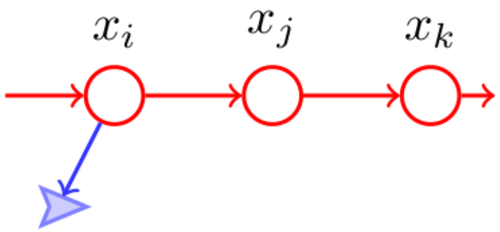


Avoiding dense graphs:

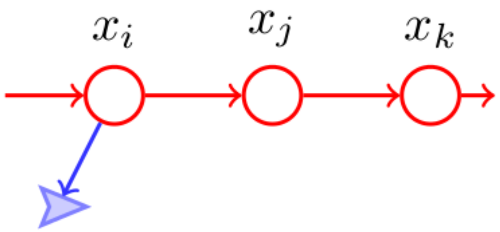
- Kretzschmar et al. (IROS 11): approximate marginal using Chow-Liu tree
- Eade et al. (IROS 10): limit degree of nodes and remove edges
- Carlevaris-Bianco, Kaess, Eustice (TRO 14): consistent sparsification
- Mazuran et al. (IJRR 15): nonlinear factor recovery
- **Our approach:** keeping the graph simple during construction

Reduced Pose Graph (step n)

In general, not revisiting exactly same poses

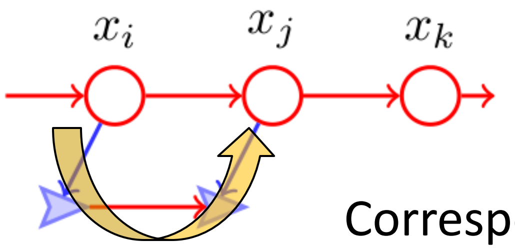


Standard pose graph:



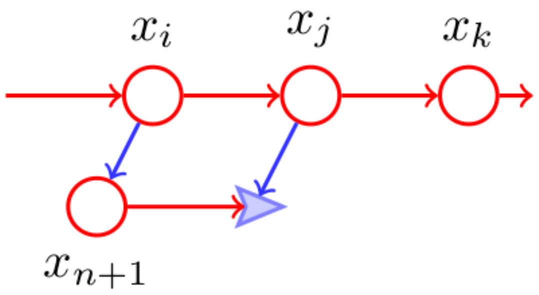
Reduced Pose Graph (step n+1)

In general, not revisiting exactly same poses



Corresponds to a constraint between x_i and x_j

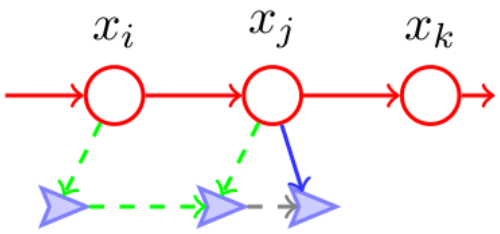
Standard pose graph:



New pose is added

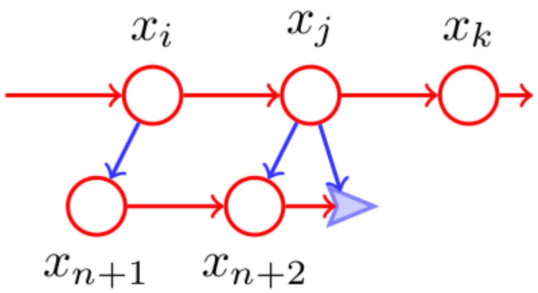
Reduced Pose Graph (step n+2)

Avoiding inconsistency



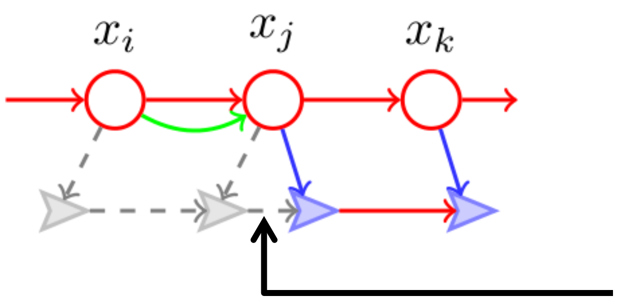
Second loop closure to x_j to avoid double use of constraint

Standard pose graph:



Reduced Pose Graph (step n+3)

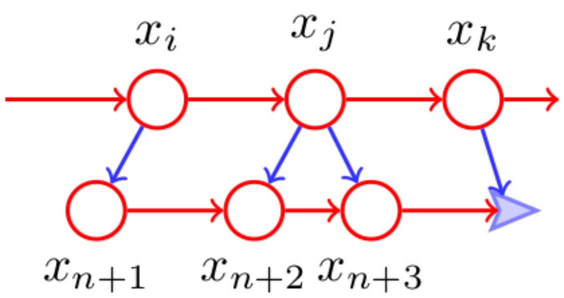
Avoiding inconsistency



Constraint between x_i and x_j added

Omitting short odometry links
similar to ESEIF by Walter 07

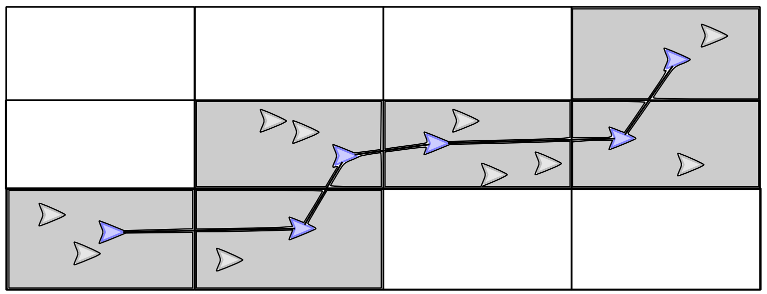
Standard pose graph:



Marginalization instead would lead to fully correlated pose graph !!

Partitioning

- How to know when to add a new pose?
- Partitioning schemes
  - Regular grid (x, y, heading)
  - Based on visibility (view frustum)
  - Based on feature overlap (typically done for keyframes)
- Choice of scheme depends on the sensors and motion



MIT Stata Center Data Set

Publically available: http://projects.csail.mit.edu/stata/

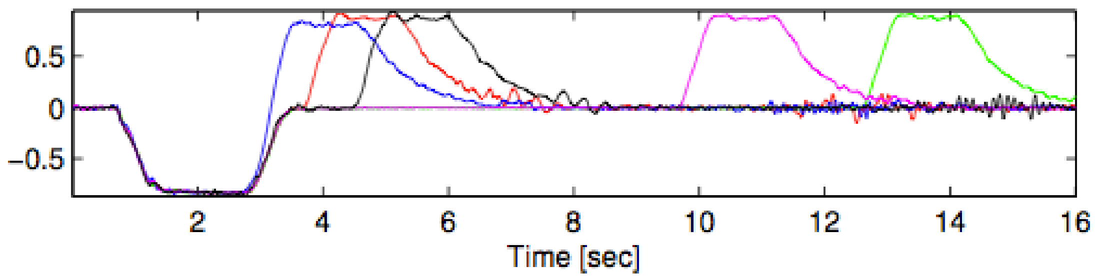
- IJRR data paper (Fallon, Johannsson, Kaess, Leonard)
- Duration: 18 months
- Operation time: 38 hours
- Distance travelled: 42 km (26 miles)
- Size: 2.3TB
- **Ground truth** by aligning laser scans with floor plans

Reduced Pose Graph – Second Floor

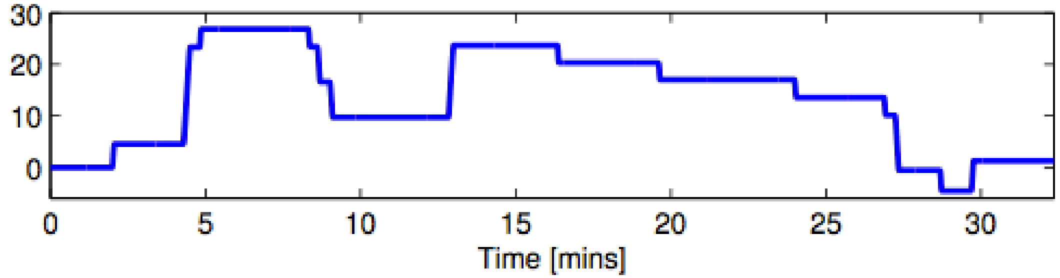
Multiple Floors – Elevator Transitions

- Accelerometer sufficient to determine floor

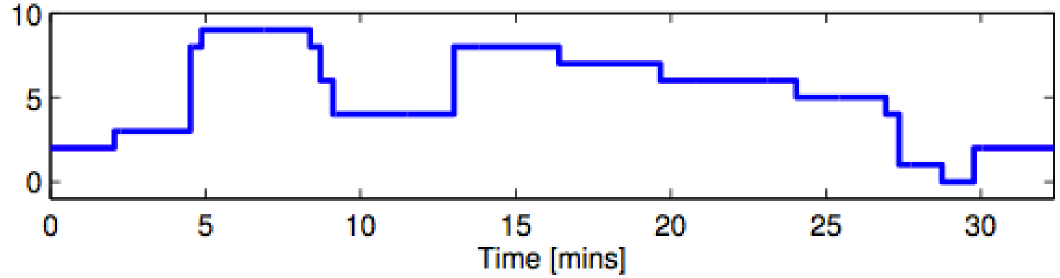
Filtered vertical acceleration
during elevator ride



Height (m)



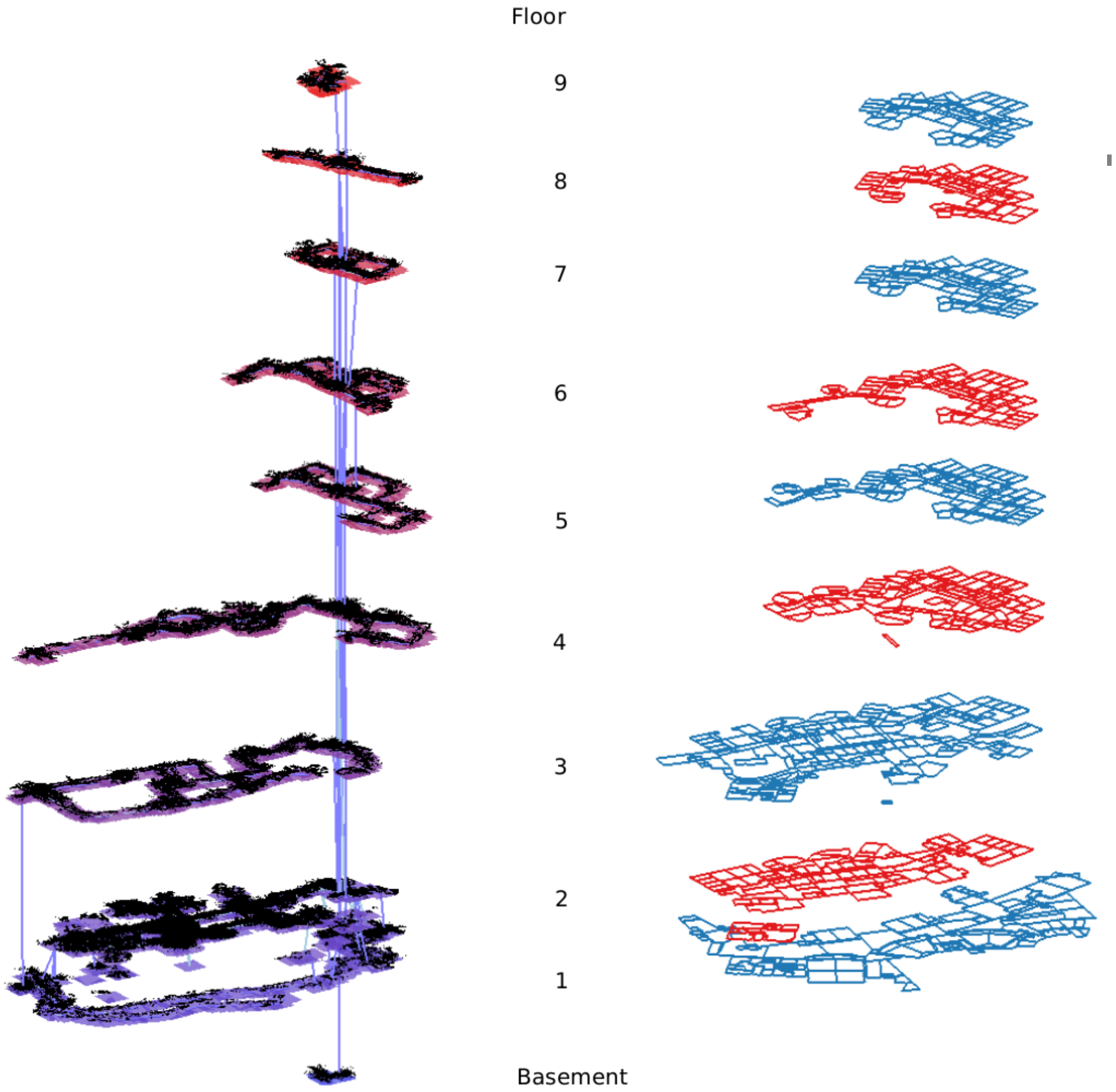
Floor Assignment



Reduced Pose Graph

Map of 10 floors

- Accelerometer used to detect elevator transitions
- iSAM optimizes RPG to achieve real-time



Reduced Pose Graph – 10 Floors