NA 568 - Winter 2022

# Simultaneous Localization and Mapping

**Maani Ghaffari**

**Slides: Courtesy of Ryan Eustice**

March 29, 2022

UNIVERSITY OF MICHIGAN

# Three Main SLAM Paradigms

Kalman filter

Particle filter

**Graph-based**

**least squares approach to SLAM**

Courtesy: C. Stachniss

# Least Squares in General

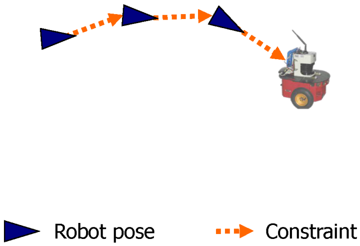
- Approach for computing a solution for an **overdetermined system**
- "More equations than unknowns"
- Minimizes the **sum of the squared errors** in the equations
- Standard approach to a large set of problems

**Today: Application to SLAM**

Courtesy: C. Stachniss

# Graph-Based SLAM

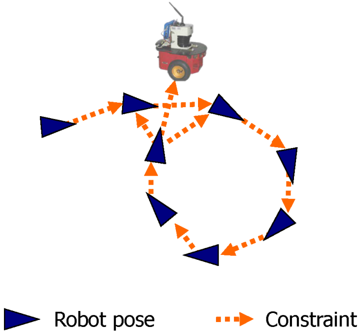
- Constraints connect the poses of the robot while it is moving
- Constraints are inherently uncertain




Courtesy: C. Stachniss

# Graph-Based SLAM

- Observing previously seen areas generates constraints between non-successive poses




Courtesy: C. Stachniss

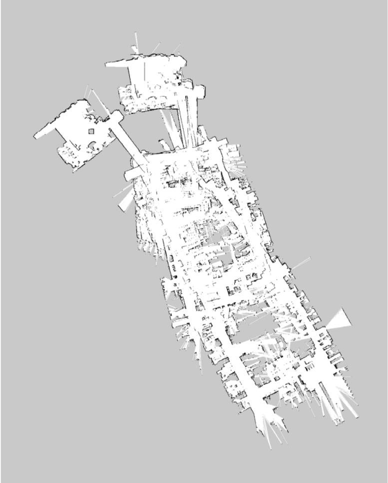
# Idea of Graph-Based SLAM

- Use a **graph** to represent the problem
- Every **node** in the graph corresponds to a pose of the robot during mapping
- Every **edge** between two nodes corresponds to a spatial constraint between them
- **Graph-Based SLAM:** Build the graph and find a node configuration that minimizes the error introduced by the constraints

Courtesy: C. Stachniss

# Graph-Based SLAM in a Nutshell

- Every node in the graph corresponds to a robot position and a laser measurement
- An edge between two nodes represents a spatial constraint between the nodes

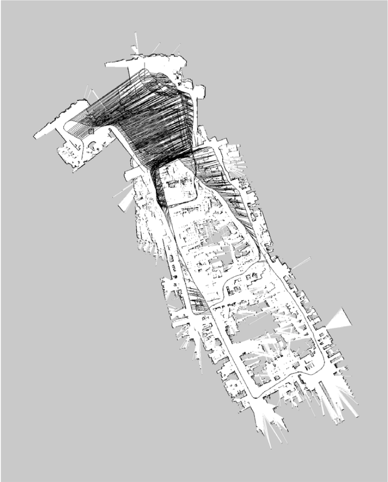


KUKA Halle 22, courtesy of P. Pfaff

Courtesy: C. Stachniss

# Graph-Based SLAM in a Nutshell

- Every node in the graph corresponds to a robot position and a laser measurement
- An edge between two nodes represents a spatial constraint between the nodes

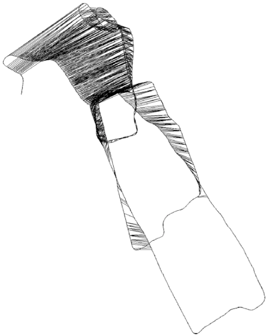


KUKA Halle 22, courtesy of P. Pfaff

Courtesy: C. Stachniss

# Graph-Based SLAM in a Nutshell

- Once we have the graph, we determine the most likely map by correcting the nodes

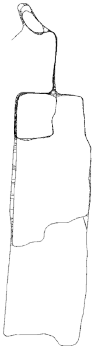


Courtesy: C. Stachniss

# Graph-Based SLAM in a Nutshell

- Once we have the graph, we determine the most likely map by correcting the nodes

  … like this

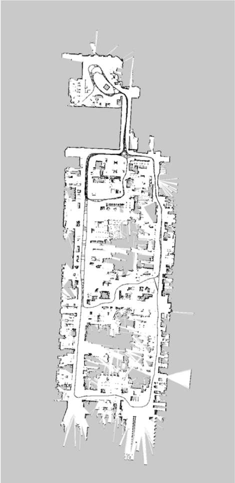


Courtesy: C. Stachniss

# Graph-Based SLAM in a Nutshell

- Once we have the graph, we determine the most likely map by correcting the nodes

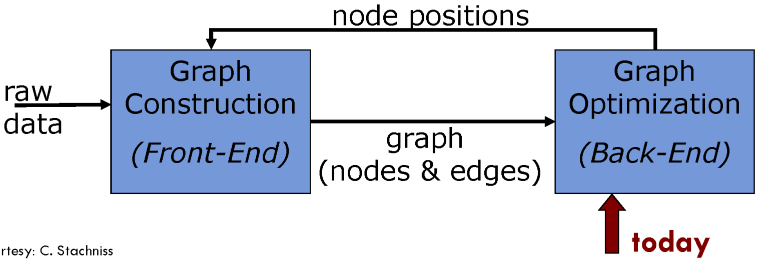
  … like this
- Then, we can render a map based on the known poses



Courtesy: C. Stachniss

# The Overall SLAM System

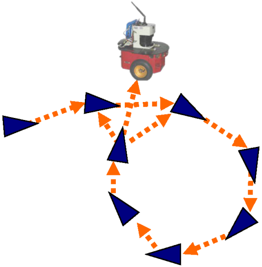
- Interplay of front-end and back-end
- Map helps to determine constraints by reducing the search space
- Topic today: optimization




Courtesy: C. Stachniss

# The Graph

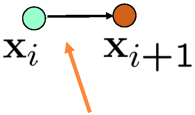
- It consists of $n$ nodes $\mathbf{x} = \mathbf{x}_{1:n}$
- Each $\mathbf{x}_i$ is a 2D or 3D transformation (the pose of the robot at time $t_i$ )
- A constraint/edge exists between the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ if…



Courtesy: C. Stachniss

# Create an Edge If… (1)

- …the robot moves from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$
- Edge corresponds to odometry

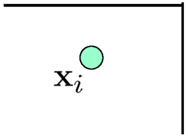



The edge represents the **odometry** measurement

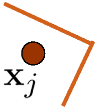
Courtesy: C. Stachniss

# Create an Edge If… (2)

- …the robot observes the same part of the environment from $\mathbf{x}_i$ and from $\mathbf{x}_j$




Measurement from $\mathbf{x}_i$

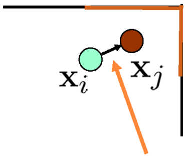



Measurement from $\mathbf{x}_j$

Courtesy: C. Stachniss

# Create an Edge If… (2)

- …the robot observes the same part of the environment from $\mathbf{x}_i$ and from $\mathbf{x}_j$
- Construct a **virtual measurement** about the position of $\mathbf{x}_j$ seen from $\mathbf{x}_i$




Edge represents the position of $\mathbf{x}_j$ seen from $\mathbf{x}_i$ based on the **observation**

Courtesy: C. Stachniss

# Transformations

- Transformations can be expressed using **homogenous coordinates**
- Odometry-Based edge

$$(\mathbf{X}_i^{-1}\mathbf{X}_{i+1})$$

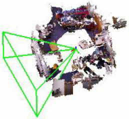
- Observation-Based edge

$$(\mathbf{X}_i^{-1}\mathbf{X}_j)$$

How node $i$ sees node $j$

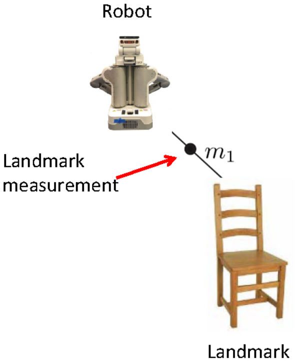
Courtesy: C. Stachniss

# Visual SLAM



Courtesy: M. Kaess

# The SLAM Problem (t=0)

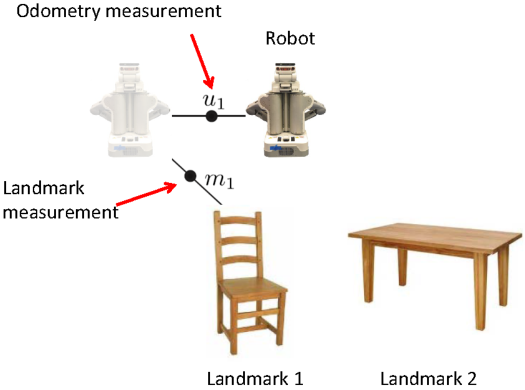



Onboard sensors:

- Wheel odometry
- Inertial measurement unit (gyro, accelerometer)
- Sonar
- Laser range finder
- Camera
- RGB-D sensors

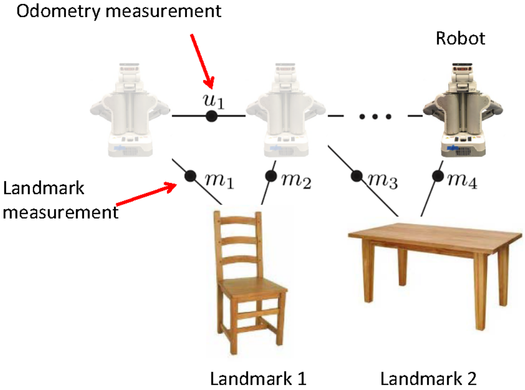
Courtesy: M. Kaess

# The SLAM Problem (t=1)



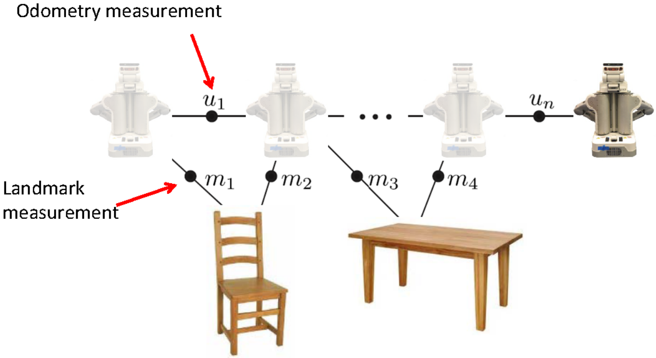

Courtesy: M. Kaess

# The SLAM Problem (t=n-1)



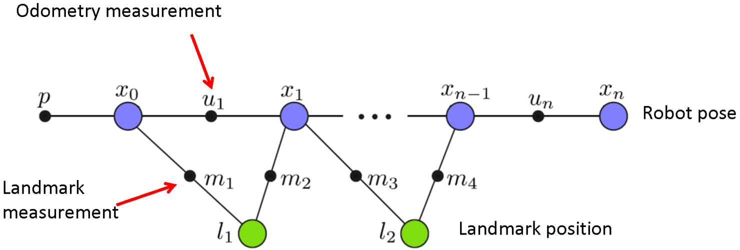

Courtesy: M. Kaess

# The SLAM Problem (t=n)




Courtesy: M. Kaess

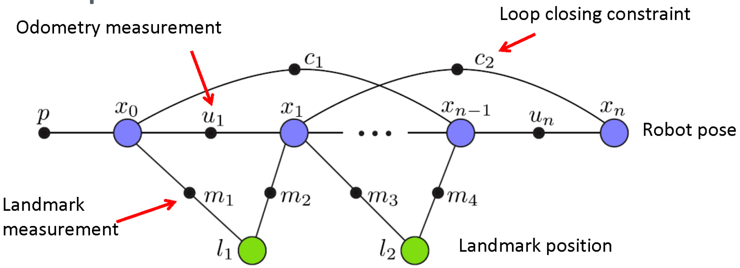
# Factor Graph Representation




Bipartite graph with ***variable nodes*** and ***factor nodes***

Courtesy: M. Kaess

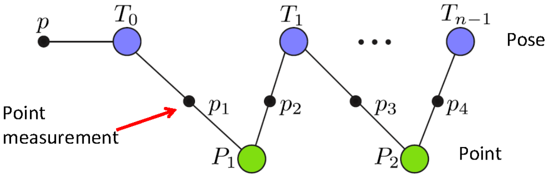
# Factor Graph Representation: Pose Graph




Bipartite graph with ***variable nodes*** and ***factor nodes***

Courtesy: M. Kaess

# Factor Graph Representation: Bundle Adjust.

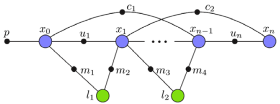



Bipartite graph with ***variable nodes*** and ***factor nodes***

Courtesy: M. Kaess

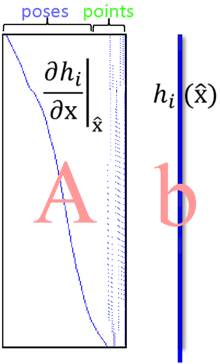
[Dellaert and Kaess, IJRR 06]

# Nonlinear Least-Squares




Gaussian noise

$$\operatorname{argmin}_{\mathrm{x}} \sum_i \|h_i(\mathrm{x})\|_{\Xi}^{2}$$




Repeatedly solve linearized system (GN)

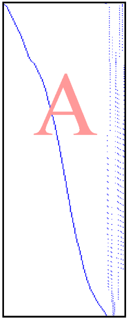
$$\operatorname{argmin}_x \|Ax - b\|^2$$

$$A = \begin{bmatrix} F_{11} & & G_{11} & & \\ F_{12} & & & G_{12} & \\ F_{13} & & & & G_{13} \\ & F_{21} & G_{21} & & \\ & F_{22} & & G_{22} & \\ & F_{23} & & & G_{23} \end{bmatrix}, \quad x = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}, \quad b = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \\ b_{15} \\ b_{16} \end{bmatrix}$$

Courtesy: M. Kaess

# Solving the Linear Least-Squares System

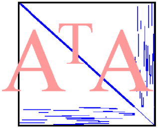
Solve: $$\operatorname{argmin}_x \|Ax - b\|^2$$




Measurement Jacobian

Normal equations

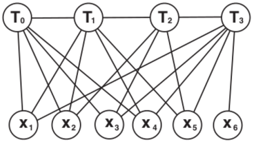
$$A^T A x = A^T b$$




Information matrix

Courtesy: M. Kaess

# Full Bundle Adjustment



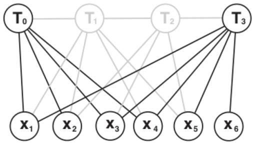

From Strasdat et al, 2011 IVC "Visual SLAM: Why filter?"

- Graph grows with time:
  - Have to solve a sequence of increasingly larger BA 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

Courtesy: M. Kaess

# Keyframe Bundle Adjustment



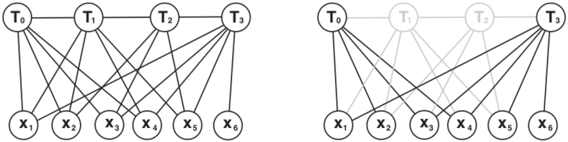

- Drop subset of poses to reduce density/complexity
- Only retain "keyframes" necessary for good map

- Complexity still grows with time, just slower

Courtesy: M. Kaess

# Incremental Solver

- 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!

Courtesy: M. Kaess

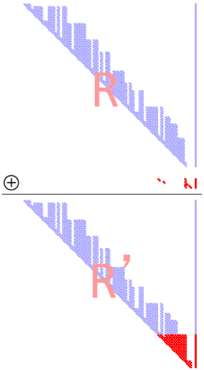
[Kaess et al., TRO 08]

# iSAM

Solving a growing system:

- Exact/batch (quickly gets expensive)
- Approximations
- Incremental Smoothing and Mapping (iSAM)

New measurements ->




Key idea:

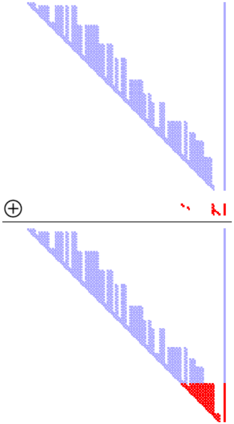
- Append to existing matrix factorization
- "Repair" using Givens rotations

Periodic batch steps for

- Relinearization
- Variable reordering (to keep sparsity)

Courtesy: M. Kaess

# Factor Updates with Givens Rotations



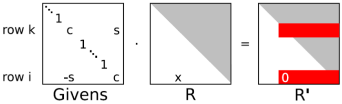
Old $R$ factor

New rows

New $R$ factor:
Triangulated using
Givens Rotations

Constant time!

Zeroing entries with Givens Rotations:




**Numerically stable !**

Courtesy: M. Kaess

[Kaess et al., TRO 08]

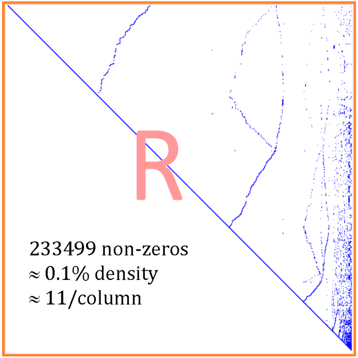
# iSAM

Example from real sequence:

Square root inf. matrix

Side length: 21000 variables

Dense: 1.7GB, sparse: 1MB




Courtesy: M. Kaess

# Readings & References


- Dellaert, F. and Kaess, M., 2006. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. The International Journal of Robotics Research, 25(12), pp.1181-1203.
- Kaess, M., Ranganathan, A. and Dellaert, F., 2008. iSAM: Incremental smoothing and mapping. IEEE Transactions on Robotics, 24(6), pp.1365-1378.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. and Leonard, J.J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics, 32(6), pp.1309-1332.
- State Estimation for Robotics: Ch. 8 and 9