INTRODUCTORY TUTORIAL OF SLAM

Presenter: Zili Wang

Date: July 22nd, 2020

SIMULTANEOUS LOCALIZATION AND MAPPING

SURVEYING THE SURVEYS AND TUTORIALS

Year	Topic	Reference
2006	Probabilistic approaches and data association	Durrant–Whyte and Bailey [8], [70]
2008	Filtering approaches	Aulinas *et al.* [7]
2011	SLAM back end	Grisetti *et al.* [98]
2011	Observability, consistency and convergence	Dissanayake *et al.* [65]
2012	Visual odometry	Scaramuzza and Fraundofer [86], [218]
2016	Multi robot SLAM	Saeedi *et al.* [216]
2016	Visual place recognition	Lowry *et al.* [160]
2016	SLAM in the Handbook of Robotics	Stachniss *et al.* [234, Ch. 46]
2016	Theoretical aspects	Huang and Dissanayake [110]

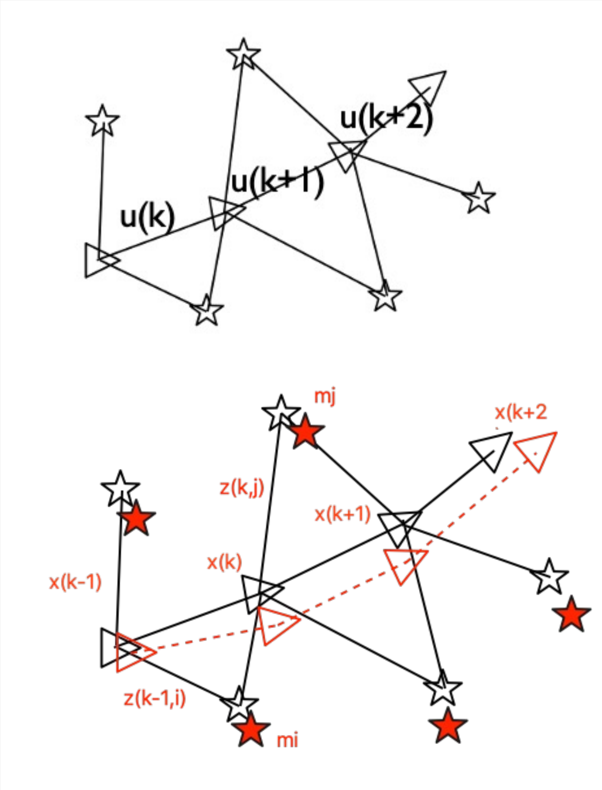
Surveys and Tutorials in SLAM [1]

(1986-2004) Classical age:
Main probabilistic formulations
- ❑ Extended Kalman Filters (EKF)
- ❑ Rao-Blackwellized Particle Filters (RBPF)
- ❑ Maximum Likelihood Estimation

(2004-2015) Algorithmic-analysis age:
Fundamental properties studies
- ❑ Observability
- ❑ Convergence
- ❑ Consistency etc

(2015-Present) Robust-perception age
- ❑ Robust performance
- ❑ High-level understanding
- ❑ Resource awareness
- ❑ Task-driven perception

SLAM PROBLEM DEFINITION



Given

- Robot controls $u_{1:T} = \{u_1, u_2, u_3, \ldots, u_T\}$
- Observations $z_{1:T} = \{z_1, z_2, z_3, \ldots, z_T\}$

Want

- Map of the environment m (landmarks, occupancy grids, surface maps, point clouds)
- Robot path $\mathrm{x}_{1:\mathrm{T}} = \{x_0, x_1, x_2, \ldots, x_T\}$

Full SLAM estimates the entire path

$$p\left(x_{0:T}, m \middle| z_{1:T,}, u_{1:T}\right)$$

Online SLAM estimated the most recent pose

$$p\left(x_t, m \middle| z_{1:t,}, u_{1:t}\right)$$

FOUNDATIONS IN SLAM

- Kalman Filter based SLAM
- Particle Filter based SLAM
- Graph based SLAM

FOUNDATIONS IN SLAM

Kalman Filter based SLAM

Particle Filter based SLAM

Graph based SLAM

BAYES FILTER

$$
\begin{aligned}
bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) & \\
= \;& \eta \, p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \, p(x_t \mid z_{1:t-1}, u_{1:t}) && \text{Bayes' rule} \\
= \;& \eta \, p(z_t \mid x_t) \, p(x_t \mid z_{1:t-1}, u_{1:t}) && \text{Markov assumption} \\
= \;& \eta \, p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) && \text{Law of total probability} \\
& \qquad p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1} \\
= \;& \eta \, p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) \, p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \, dx_{t-1} && \text{Markov assumption} \\
= \;& \eta \, p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) \, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}) \, dx_{t-1} && \text{Markov assumption} \\
= \;& \eta \, p(z_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_t) \, \underline{bel(x_{t-1})} \, dx_{t-1}
\end{aligned}
$$

BAYES FILTER

$$bel(x_t) = \eta\, p(z_t|x_t) \int_{x_{t-1}} p(x_t|x_{t-1}, u_t)\, bel(x_{t-1})\, dx_{t-1}$$

- Assumption

I. Observations are conditionally independent given the map and current robot pose

II. State transition is a Markov process

- Prediction step

$$\overline{bel}(x_t) = \int \underbrace{p(x_t|x_{t-1}, u_t)}_{motion\ model}\, bel(x_{t-1})\, dx_{t-1}$$

- Correction step

$$bel(x_t) = \eta \underbrace{p(z_t|x_t)}_{sensor\ model}\, \overline{bel}(x_t)$$

EKF-SLAM

- Assume $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are additive, zero mean uncorrelated Gaussian noise with covariance $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ (linear Gaussian)
- Motion model: $\mathrm{p}(x_t|x_{t-1},u_t) \Leftrightarrow \boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1},\boldsymbol{u}_t) + \boldsymbol{w}_t$
- Observation model: $\mathrm{p}(z_t|x_t) \Leftrightarrow \boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{x}_t, m) + \boldsymbol{v}_t$
- Algorithm ($\widehat{\boldsymbol{x}}_{t|t}, \widehat{\boldsymbol{m}}_t$, and $\boldsymbol{P}_{t|t}$: mean and covariance of joint posterior distribution)

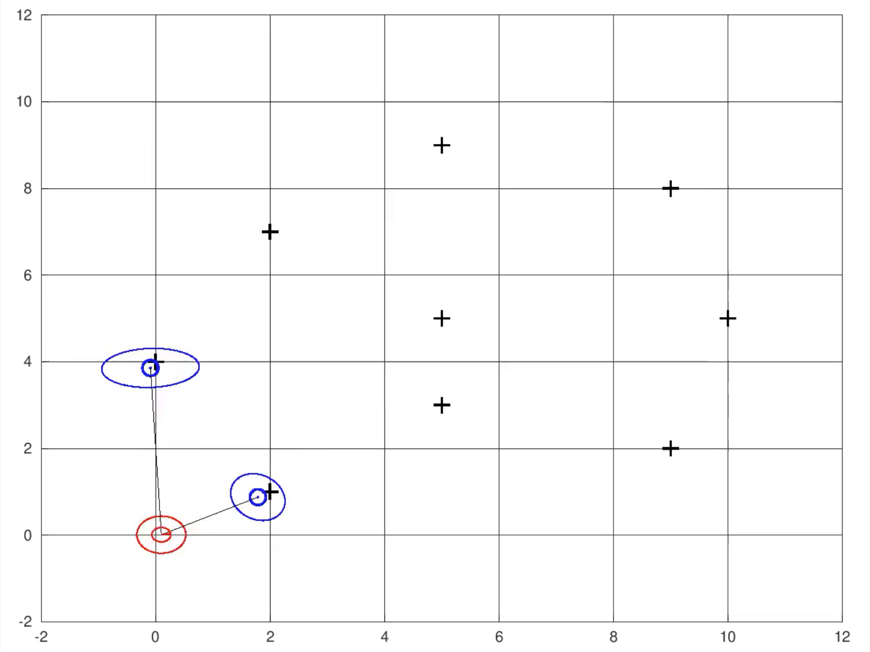
i. Prediction update

$$\widehat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{f}(\hat{x}_{t-1|t-1}, \boldsymbol{u}_k)$$
$$\boldsymbol{P}_{xx,t|t-1} = \nabla \boldsymbol{f}\, \boldsymbol{P}_{xx,t-1|t-1}\, \nabla \boldsymbol{f}^T + \boldsymbol{Q}_t$$

i. Correction update

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\, \nabla \boldsymbol{h}^T \left(\nabla \boldsymbol{h}\, \boldsymbol{P}_{t|t-1} \nabla \boldsymbol{h}^T + \boldsymbol{R}_t\right)^{-1}$$
$$\begin{bmatrix} \widehat{\boldsymbol{x}}_{t|t} \\ \widehat{\boldsymbol{m}}_t \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{x}}_{t|t-1} \\ \widehat{\boldsymbol{m}}_{t-1} \end{bmatrix} + \boldsymbol{K}_t\, (\boldsymbol{z}(t) - \boldsymbol{h}(\widehat{\boldsymbol{x}}_{t|t-1}, \widehat{\boldsymbol{m}}_{t-1}))$$
$$\boldsymbol{P}_{t|t} = (I - \boldsymbol{K}_t \nabla \boldsymbol{h}) \boldsymbol{P}_{,t|t-1}$$

EKF SLAM



Properties

- Convergence::
  the standard deviation of a landmark decreases monotonically towards a lower bound
- Computational effort:
  Computation grows quadratically with the number of landmarks
- Data association:
  Fragile to incorrect association of observations to landmarks
- Non-linearity:
  Local linearization may deviate from consistency and convergence

Improvements [3]

FOUNDATIONS IN SLAM

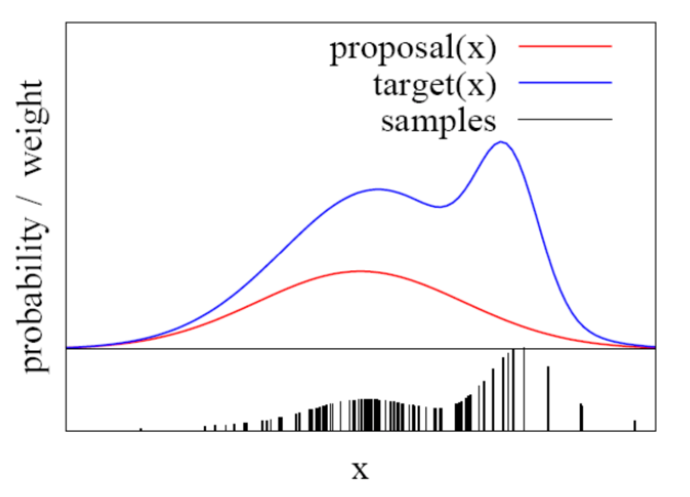
Kalman Filter based SLAM

Particle Filter based SLAM

Graph based SLAM

RBPF SLAM

- Based on Monte Carlo sampling/ particle filtering to directly represent non-linear process model and non-Gaussian pose distribution
- Particle Filter and Importance sampling principle



- Particle Filter algorithm

For N particles:

i. Sample the particles using the proposal distribution $x_t^{[j]} = \pi(x_t)$

ii. Compute the importance weights $w_t^{[j]} = \dfrac{target\left(x_t^{[j]}\right)}{proposal\ \left(x_t^{[j]}\right)}$

Resampling: draw samples i with probability $w_t^{[j]}$ and repeat N times

RBPF SLAM

- Rao-Blackwellisation: joint state factorization

$$p(x_1, x_2) = \underbrace{p(x_2|x_1)}_{\substack{can\ be\ represented \\ analytically}} \quad \underbrace{p(x_1)}_{\substack{need\ be \\ sampled}\ x_1^{(i)} \sim p(x_1)}$$

$$\Leftrightarrow \{x_1^{(i)}, p\left(x_2 \middle| x_1^{(i)}\right)\}_i^N$$

- Rao-Blackwellisation: SLAM

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) = \underbrace{p(x_{1:t}|z_{1:t}, u_{1:t})}_{\substack{path\ posterior \\ particle\ filter}} \underbrace{p(m|x_{1:t}, z_{1:t})}_{\substack{map\ posterior \\ ananytic}}$$

- Importance weights

i. Target distribution: $p(x_{1:t}|z_{1:t}, u_{1:t})$

ii. Proposal distribution: $\pi(x_{1:t}|z_{1:t-1}, u_{1:t}) = \pi(x_t|x_{t-1}, u_t)\pi(x_{1:t-1}|z_{1:t-1}, u_{1:t-1})$ gives FastSLAM 1.0

$\pi(x_{1:t}|z_{1:t}, u_{1:t}) = \pi(x_t|x_{t-1}, z_{1:t}, u_t)\pi(x_{1:t-1}|z_{1:t-1}, u_{1:t-1})$ gives FastSLAM 2.0

RBPF SLAM

i. Sampling based on proposal distribution π
ii. Importance weighting
iii. Resampling such that all particles have same weight
iv. Map estimation based on robot trajectory and observations $\underbrace{(p(m|x_{1:t}, z_{1:t}))}$

- Feature based

$$\underbrace{\prod_{i=1}^{M} p(m_i|x_{0:t}, z_{1:t})}_{EKFs\ (conditioned\ on\ the\ whole\ trajcotyr, landmakrs\ are\ independent\ \ Gaussians)}$$

- Grid based (GMapping)

$$l(m_i \mid z_{1:t}, x_{1:t}) = \underbrace{l(m_i \mid z_t, x_t)}_{\text{inverse sensor model}} + \underbrace{l(m_i \mid z_{1:t-1}, x_{1:t-1})}_{\text{recursive term}} - \underbrace{l(m_i)}_{\text{prior}}$$

FastSLAM1.0_known_correspondence$(z_t, c_t, u_t, \mathcal{X}_{t-1})$:

for $k = 1$ to N do // loop over all particles
  Let $\left\langle x_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \right\rangle, \ldots \right\rangle$ be particle k in $\mathcal{X}_{t-1}$
  $x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ // sample pose
  $j = c_t$ // observed feature
  if feature j never seen before
    $\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ // initialize mean
    $H = h'(\mu_{j,t}^{[k]}, x_t^{[k]})$ // calculate Jacobian
    $\Sigma_{j,t}^{[k]} = H^{-1}\, Q_t\, (H^{-1})^T$ // initialize covariance
    $w^{[k]} = p_0$ // default importance weight
  else

EKF update

    $\hat{z}^{[k]} = h(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // measurement prediction
    $H = h'(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // calculate Jacobian
    $Q = H\, \Sigma_{j,t-1}^{[k]}\, H^T + Q_t$ // measurement covariance
    $K = \Sigma_{j,t-1}^{[k]}\, H^T\, Q^{-1}$ // calculate Kalman gain
    $\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]} + K(z_t - \hat{z}^{[k]})$ // update mean
    $\Sigma_{j,t}^{[k]} = (I - K\, H)\Sigma_{j,t-1}^{[k]}$ // update covariance

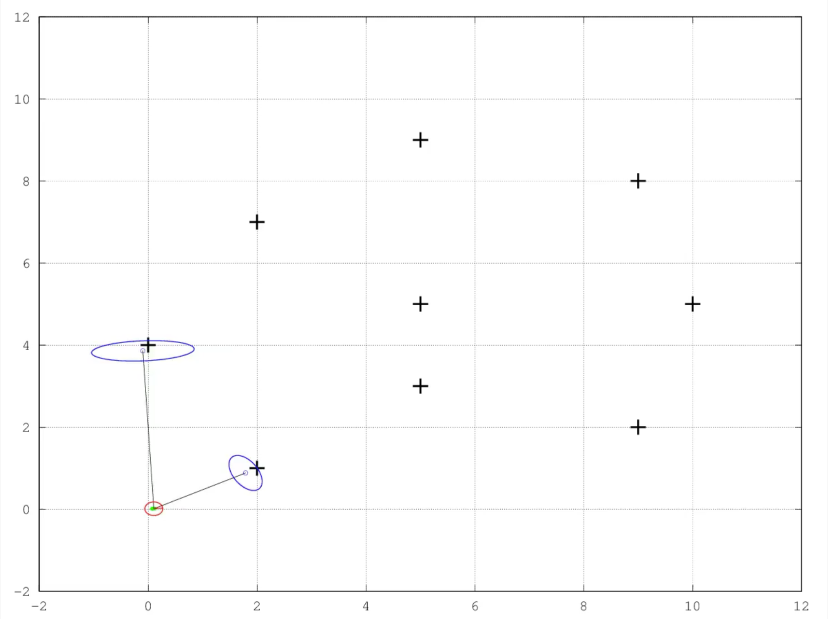
    $w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$ // importance factor
  endif
  for all unobserved features j' do
    $\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle$ // leave unchanged
  endfor
endfor
$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \ldots, w^{[k]} \right\rangle_{k=1,\ldots,N}\right)$
return $\mathcal{X}_t$

RBPF SLAM

- FastSLAM 1.0: motion model as proposal distribution (suboptimal, esp. when sensor information is much more precise than motion estimate from odometry) , importance weights are based on observation model.

  FastSLAM 2.0 integrate most recent sensor observation into the proposal (optimal, gives a more peaked proposal distribution), less particles are required, more robust and accurate

- Improved techniques for Grid Mapping with RBPFs (improved proposal distribution, scan matching, and adaptive sampling) – details refer to [4]



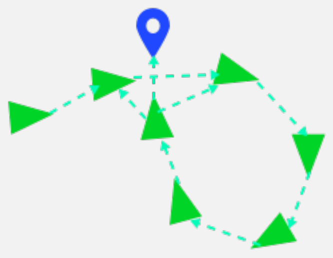
FOUNDATIONS IN SLAM

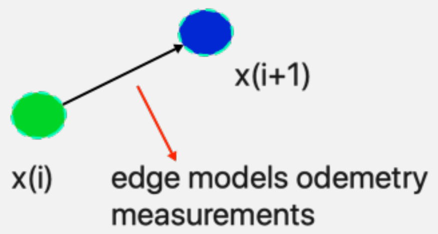
Kalman Filter based SLAM

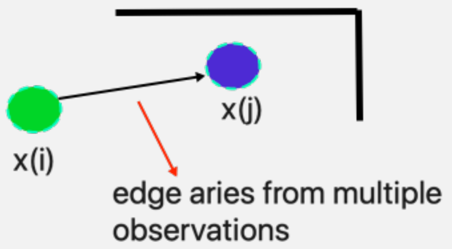
Particle Filter based SLAM

Graph based SLAM

GRAPH







- Node: a robot position and a measurement acquired at that position
- Edge: spatial constraint relating the corresponding two robot poses, a constraint consists in a probability distribution over the relative transformation between two poses ($< z_{ij}, \Omega_{ij} >$, where Ω_{ij} is the information matrix, larger value make the edge matter more)
- Graph SLAM: Build the graph and find a node configuration which minimizes the error introduced by constraints

GRAPH SLAM

Node positions

Graph Construction (Frond-end)

Raw data

- Interpret the sensor data to obtain the constraints and construct the graph
- Determine the most likely constraint from an observation (data associations)

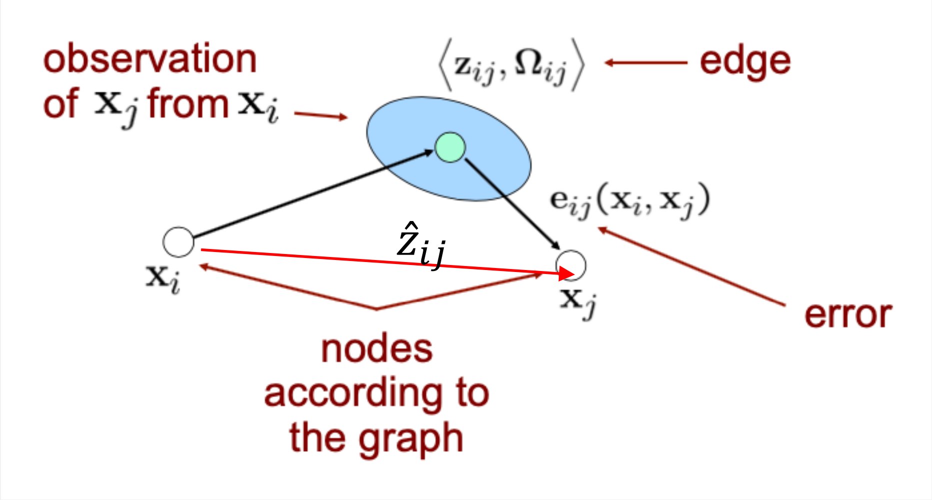
Graph (nodes and edges)

Graph Optimization (Back-end)

- Determine the most likely configuration of the poses give the edges of the graph

For correct data association, the frond-end requires consistent estimate of $p(x_{1:T}|z_{1:T}, u_{1:T})$, which is fed in from back-end

PROBLEM DEFINITION



- z_{ij} : mean of virtual measurement
- $\hat{z}_{ij}$: prediction of virtual measurement
- $e_{ij}\left(x_i, x_j\right) = z_{ij} - \hat{z}_{ij}$
- Given gaussian noise, the log-likelihood of a measurements is proportional to $e_{ij}^T\, \Omega_{ij}\, e_{ij}$
- Maximum likelihood goal is to find x^* to minimize the negative log likelihood of all the observations

$$x^* = \underset{x}{argmin}\, F(x) = \underset{x}{argmin} \sum_{(i,j)} e_{ij}^T\, \Omega_{ij}\, e_{ij}$$

GAUSS-NEWTON ERROR MINIMIZATION

$$\begin{aligned}\mathbf{e}_{ij}(\breve{\mathbf{x}}_i + \mathbf{\Delta x}_i, \breve{\mathbf{x}}_j + \mathbf{\Delta x}_j) &= \mathbf{e}_{ij}(\breve{\mathbf{x}} + \mathbf{\Delta x}) \\ &\simeq \mathbf{e}_{ij} + \mathbf{J}_{ij}\mathbf{\Delta x}.\end{aligned}$$

$$\begin{aligned}\mathbf{F}(\breve{\mathbf{x}} + \mathbf{\Delta x}) &= \sum_{\langle i,j\rangle \in \mathcal{C}} \mathbf{F}_{ij}(\breve{\mathbf{x}} + \mathbf{\Delta x}) \\ &= \sum_{\langle i,j\rangle \in \mathcal{C}} \mathbf{e}_{ij}(\breve{\mathbf{x}} + \mathbf{\Delta x})^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\breve{\mathbf{x}} + \mathbf{\Delta x}) \\ &\simeq \sum_{\langle i,j\rangle \in \mathcal{C}} (\mathbf{e}_{ij} + \mathbf{J}_{ij}\mathbf{\Delta x})^T \mathbf{\Omega}_{ij} (\mathbf{e}_{ij} + \mathbf{J}_{ij}\mathbf{\Delta x}) \\ &= \sum_{\langle i,j\rangle \in \mathcal{C}} \underbrace{\mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}}_{\mathbf{c}_{ij}} + 2 \underbrace{\mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}} \mathbf{\Delta x} + \mathbf{\Delta x}^T \underbrace{\mathbf{J}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}} \mathbf{\Delta x} \\ &= \sum_{\langle i,j\rangle \in \mathcal{C}} \mathrm{c}_{ij} + 2\mathbf{b}_{ij}\mathbf{\Delta x} + \mathbf{\Delta x}^T \mathbf{H}_{ij} \mathbf{\Delta x} \\ &= \mathrm{c} + 2\mathbf{b}^T \mathbf{\Delta x} + \mathbf{\Delta x}^T \mathbf{H} \mathbf{\Delta x}\end{aligned}$$

Locally minimizing solution: $\mathrm{H}\,\Delta x^* = -b, \qquad \mathrm{x}^* = \breve{x} + \Delta x^*$

GAUSS-NEWTON ERROR MINIMIZATION

- Due to the sparsity of the Jacobian

$$J_{ij} = \begin{pmatrix} 0 & \ldots 0 & \underbrace{A_{ij}}_{\substack{derivative \\ w.r.t \\ node\ i}} & 0 & \ldots 0 & \underbrace{B_{ij}}_{\substack{derivative \\ w.r.t \\ node\ j}} & 0 & \ldots 0 \end{pmatrix}$$

H_{ij} and b_{ij} are also sparse

- At each iteration, constraint-based update of H_{ij} and b_{ij} is considered due to the sparsity, referring [5] for details in 2D laser-based mapping
- Most of the visual SLAM algorithms are in this category, referring [1] to see the extensions of Graph-based SLAM

REFERENCE

[1]A complete survey of SLAM

[2][3]Tutorial of filtering-based approach

[4]GMapping

[5]Tutorial of graph-based approach

[6]Course in robot mapping, search professor for newer release

[7]Book: old but provides all the fundamental details, errata

[8][9]Project reference, both have extensions in newer release

- [1] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," in IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309-1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006, doi: 10.1109/MRA.2006.1638022.
- [3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," in IEEE Robotics & Automation Magazine, vol. 13, no. 3, pp. 108-117, Sept. 2006, doi: 10.1109/MRA.2006.1678144.
- [4] G. Grisetti, C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," in IEEE Transactions on Robotics, vol. 23, no. 1, pp. 34-46, Feb. 2007, doi: 10.1109/TRO.2006.889486.
- [5] G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard, "A Tutorial on Graph-Based SLAM," in IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, pp. 31-43, winter 2010, doi: 10.1109/MITS.2010.939925.
- [6] C. Stachniss, "Robot Mapping - WS 2013/14", http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/
- [7] Thrun, S. (2002). "Probabilistic robotics" (PDF). *Communications of the ACM*. **45** (3). doi:10.1145/504729.504754
- [8] Ma, Fangchang, Luca Carlone, Ulas Ayaz, and Sertac Karaman. "Sparse Depth Sensing for Resource-Constrained Robots." The International Journal of Robotics Research 38, no. 8 (July 2019): 935–80. doi:10.1177/0278364919850296.
- [9] A. Caccavale and M. Schwager, "Wireframe Mapping for Resource-Constrained Robots," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1-9, doi: 10.1109/IROS.2018.8594057.