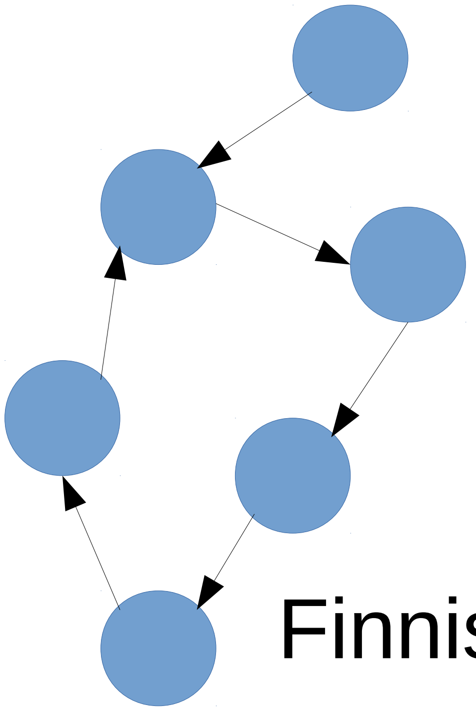
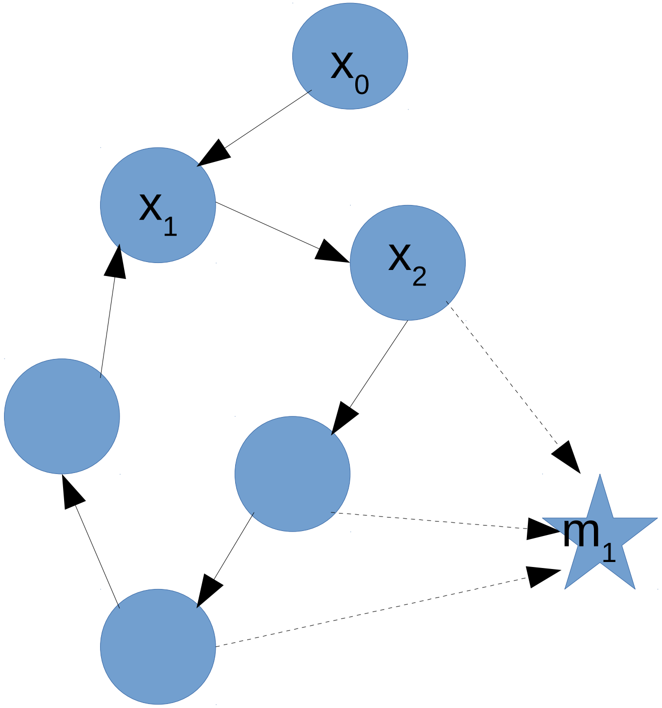


Graph SLAM

Ville Lehtola, Dr.Sc.
Senior Scientist
Finnish Geospatial Research Institute
12.3.2018

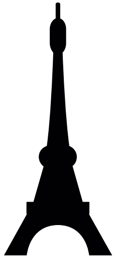
Formalism adapted to the book
“Probabilistic robotics”, Thrun, Burgard, Fox (2005)

ville.lehtola@nls.fi



Graph

- Directed or undirected
- Nodes are robot poses
- Links are either
  - consecutive poses OR
  - features sensed through measurement



Icons made by Freepik
www.flaticon.com CC 3.0 BY

ville.lehtola@nls.fi

Learning goal

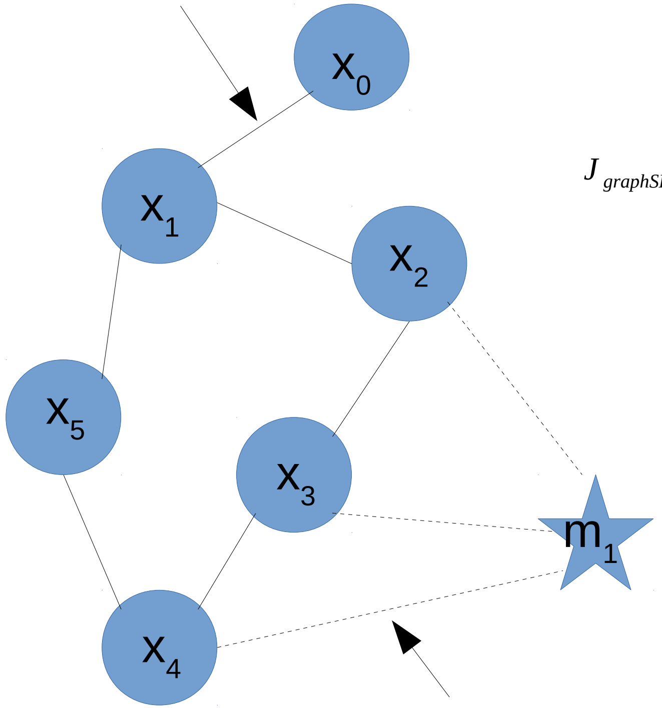
$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)] Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

“Mt. Everest”

ville.lehtola@nls.fi

Graph SLAM

$$[x_1 - g(u_1, x_0)]R^{-1}[x_1 - g(u_1, x_0)]$$



$$[z_4 - h(m_{1,} x_4)]Q^{-1}[z_4 - h(m_{1,} x_4)]$$

- Learning goals

$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t [x_t - g(u_t, x_{t-1})]R^{-1}[x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)]Q^{-1}[z_t - h(m_{c_t}, x_t)]$$

  - Graph construction and optimization
  - Case forest SLAM
  - Advanced topics

ville.lehtola@nls.fi

Why Graph SLAM?

- The use and formulation of constraints
  - Measurement constraints integrate the meas. model
  - Motion constraints integrate the motion model
  - e.g. post-processing noisy data in laboratory
- Ability to build large scale global maps
  - Sparse graph, motion constraints build linearly in time
  - Eased loop closure
  - Amount of landmarks may be very large, > 1000

$$J_{graphSLAM} = \boxed{x_0^T \Omega_0 x_0^T} + \boxed{\sum_t [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})]} + \boxed{\sum_t [z_t - h(m_{c_t}, x_t)] Q^{-1} [z_t - h(m_{c_t}, x_t)]}$$

Initial or anchor constraint

ville.lehtola@nls.fi

Graph SLAM vs other SLAM

- GraphSLAM is in post-processing phase
  - After the data is captured, offline / batch algorithm
  - Full access to all data
  - Solve the full SLAM problem
- Other SLAM
  - Online / real-time / continuous-time
    - Need fast processing, cannot access all data
  - Extended Kalman Filter, EKF SLAM

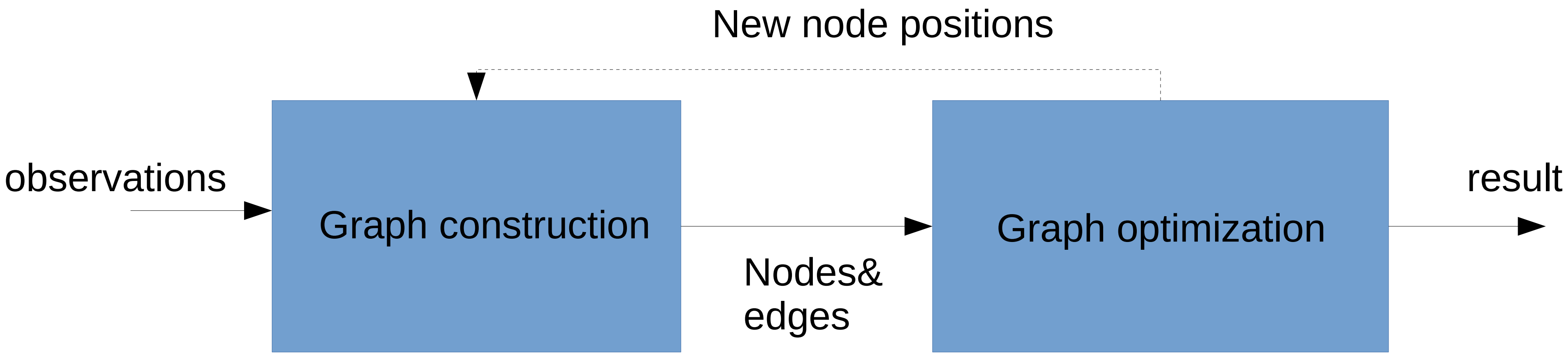
Over all time steps

$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \boxed{\sum_t} [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})] + \boxed{\sum_t} [z_t - h(m_{c_t}, x_t)] Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

NLS
FINNISH GEOSPATIAL
RESEARCH INSTITUTE
FGI

ville.lehtola@nls.fi

Graph-SLAM idea (1)

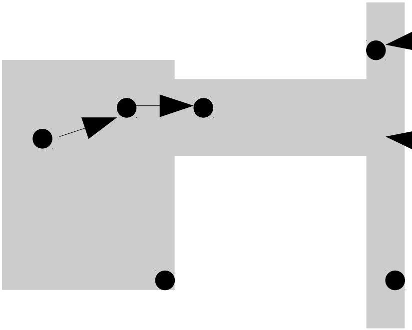


Graph-SLAM idea (2)

- Build 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
- Once we have the graph, we optimize the most likely map of the environment by correcting the nodes
  - minimize the error introduced by the constraints
    - Motion, measurement

State space S

- Robot state: position and orientation $x_t \in S$
- Cf. Physical particle (state: position and momentum)
- Thought experiment: Robot so small it would be an aerosol, a particle in the air, scout the space with diffusion
- How to discretize S so that the problem is computationally tractable?
  - Reasonable size of $\{x_t\}$



One robot state $x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$

$\theta \in [0, 2\pi]$

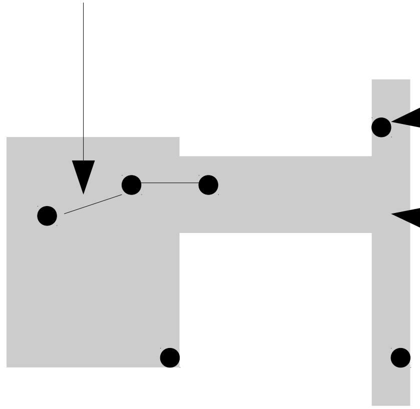
Position space P

$(x, y) \in P$

NLS
FINNISH GEOSPATIAL RESEARCH INSTITUTE
FGI

Motion constraint (1)

- We control the robot
- Discretize the state space w.r.t. time Δt, or by saying that there can be only one state per traveled distance D_0
  - e.g. $D_0 = 1$ m



One robot state $x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$

$\theta \in [0, 2\pi]$

Position space P

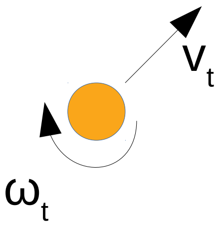
$(x, y) \in P$

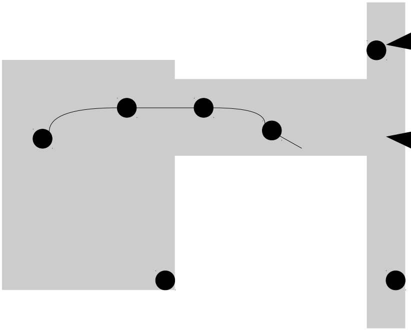
Motion constraint (2)

- Used controls tell us how the robot should move
  - Translational velocity v_t
  - Rotational velocity ω_t
  - Turn radius $r=|v_t / \omega_t|$

Indexing notation different here:

$$\begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ \theta_{i-1} \end{pmatrix} + \begin{pmatrix} \frac{-v_i}{\omega_i}\sin\theta_{i-1} + \frac{v_i}{\omega_i}\sin(\theta_{i-1}+\omega_i \Delta t) \\ \frac{v_i}{\omega_i}\cos\theta_{i-1} - \frac{v_i}{\omega_i}\cos(\theta_{i-1}+\omega_i \Delta t) \\ \omega_i \Delta t \end{pmatrix} \rightarrow g(u_i, x_{i_1})$$





One robot state $x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$

$\theta \in [0, 2\pi]$

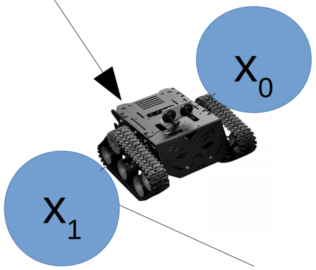
Position space P

$(x, y) \in P$

ville.lehtola@nls.fi

Motion constraint (3)

$$[x_1 - g(u_1, x_0)]\, R^{-1}\, [x_1 - g(u_1, x_0)]$$



Current state estimate y_t

Jacobian of g

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1})$$

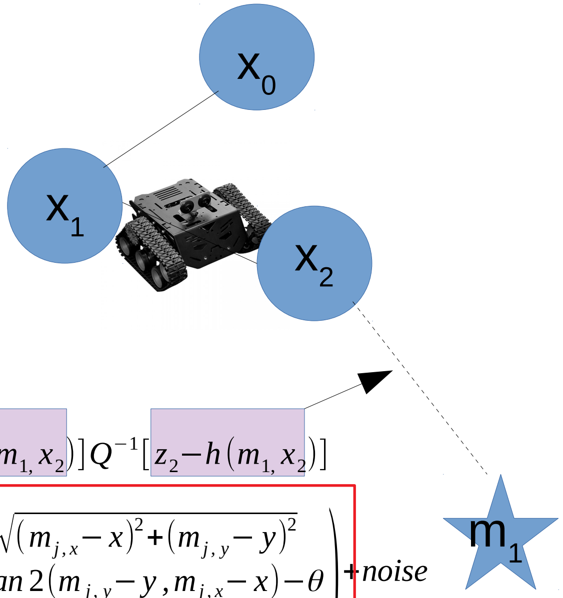
$$x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} \qquad x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

Initial conditions (anchor)

- When the robot changes its state from x_0 to x_1
- Does the state change satisfy the motion constraint?
- Motion model: g
  - Models updates to the state vector
  - Translational velocity v_1
  - Angular velocity ω_1
  - Linearized with Taylor expansion so that current state estimate μ_t may be used

ville.lehtola@nls.fi

Measurement constraint (1)



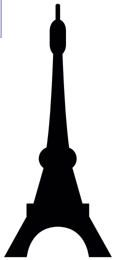
$$[z_2 - h(m_1, x_2)]\,Q^{-1}\,[z_2 - h(m_1, x_2)]$$

$$z_t = \begin{pmatrix} r_t \\ \phi_t \\ s_t \end{pmatrix} \simeq \boxed{\begin{pmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ atan2(m_{j,y}-y, m_{j,x}-x)-\theta \\ s_j \end{pmatrix}} + noise$$

h

$$x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} \qquad x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

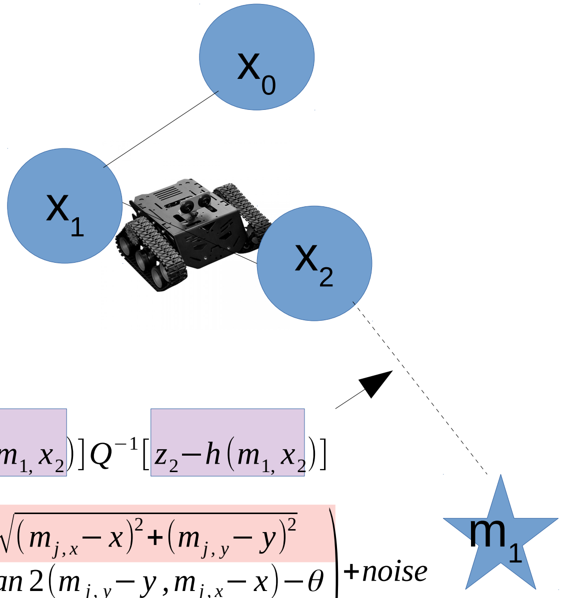
- When the robot sees the landmark m_1 from x_2
  - Measurement model: h
    - Landmark m_1 with signature s_1
    - Observer position x_2
- Compare the measurement z_2 against the previously known position of m_1
  - Is m_1 at the same range r and at the same viewing angle Φ?
  - But is it the same landmark than before?
    - correspondence



NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Measurement constraint (2)

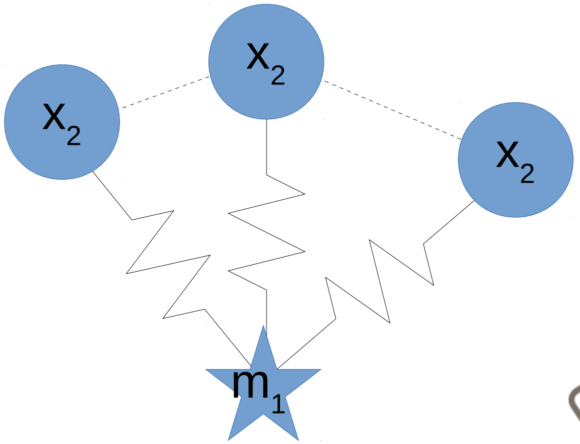


$$[z_2 - h(m_1, x_2)]\,Q^{-1}\,[z_2 - h(m_1, x_2)]$$

$$z_t = \begin{pmatrix} r_t \\ \phi_t \\ s_t \end{pmatrix} \simeq \begin{pmatrix} \sqrt{(m_{j,x}-x)^2 + (m_{j,y}-y)^2} \\ atan2(m_{j,y}-y, m_{j,x}-x) - \theta \\ s_j \end{pmatrix} + noise$$

$$x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} \qquad x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

- Range: Landmark observation is a bit like a 1D spring in a spring-mass model

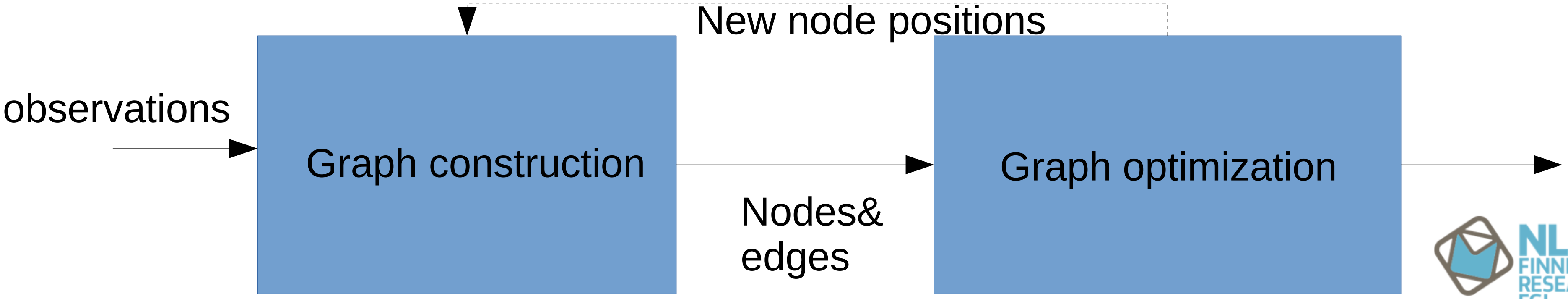


ville.lehtola@nls.fi

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

Uncertainties

- At beginning, state uncertainties are at the highest level
- Uncertainties are reduced through constraints, and iteration
- Graph building, trade-off
  - If the state space is sufficiently dense, the motion model linearization is more likely to work
  - If the state space is sufficiently dense, the measurement model is more likely to map, and correspond, the landmarks correctly
  - If the state space is too dense, CPUtime cost becomes high



Bayesian scheme (1)

- Revision of basics
- Bayes' theorem
  - P(A|B) is a conditional probability, given B
    - The next state
  - P(A) is the prior (marginal probability)
    - The previous state
  - P(B|A) is the likelihood given A
    - Constraints
  - P(B) is the normalization factor
    - constant

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

Bayesian scheme (2)

- Updating the previous states 1:t-2 with measurement and motion constraints gives the next states 1:t-1
- Obtain states 1:t as a posterior distribution

Next states 0:t

Previous states 1:t-1

$$p(y_{0:t}|z_{1:t},u_{1:t},c_{1:t}) = \eta\, p(z_t|y_t,c_t)\, p(x_t|x_{t-1},u_t)\, p(y_{1:t-1}|z_{1:t-1},u_{1:t-1},c_{1:t-1})$$

$$p(y_{0:t}|z_{1:t},u_{1:t},c_{1:t}) = \eta\, p(z_t|y_t,c_t)\, p(x_t|x_{t-1},u_t)\, \eta_{t-1}\, p(z_{t-1}|y_{t-1},c_{t-1})\, p(x_{t-1}|x_{t-2},u_{t-1})\, p(y_{1:t-2}|z_{1:t-2},u_{1:t-2},c_{1:t-2})$$

…. (recursion) ….

$$p(y_{0:t}|z_{1:t},u_{1:t},c_{1:t}) = \eta\, p(y_0) \prod_t p(x_t|x_{t-1},u_t)\, p(z_t|y_t,c_t)$$

States 1:t-2

$$y_{0:t} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \ldots \\ x_t \\ m_1 \\ \ldots \\ m_n \end{pmatrix}$$

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Bayesian scheme (3)

- Obtain the final states 1:t as a recursive posterior of all the previous states 1,2,3… t-1
- Stack all measurement and motion constraints

$$p(y_{0:t}|z_{1:t},u_{1:t},c_{1:t}) = \eta\, p(y_0) \prod_t \boxed{p(x_t|x_{t-1},u_t)} \boxed{\prod_i p(z_t^i|y_t,c_t^i)} \longrightarrow$$ Also Gaussian

$$\boxed{p(x_t|x_{t-1},u_t) = \eta \exp\left(\frac{-1}{2}(x_t - g(u_t,x_{t-1}))^T R_t^{-1}(x_t - g(u_t,x_{t-1}))\right)}$$

- Use convex log() function to get rid of Gaussian exp()
  - (Also change signs, since log(p) < 0 , because 0 < p < 1)

$$\log p(y_{0:t}|z_{1:t},u_{1:t},c_{1:t}) = const. + \log p(y_0) + \sum_t \boxed{\log p(x_t|x_{t-1},u_t)} + \sum_i \log p(z_t^i|y_t,c_t^i)$$

…

$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t \boxed{[x_t - g(u_t,x_{t-1})] R^{-1} [x_t - g(u_t,x_{t-1})]} + \sum_t [z_t - h(m_{c_t},x_t)] Q^{-1} [z_t - h(m_{c_t},x_t)]$$

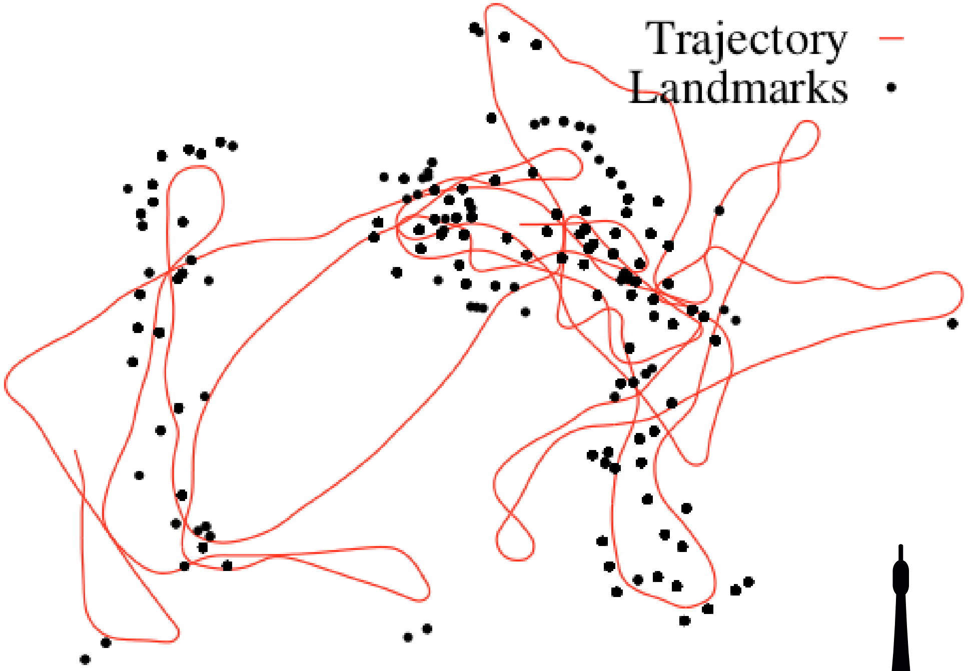
Our goal

$$y_{0:t} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_t \\ m_1 \\ \dots \\ m_n \end{pmatrix}$$

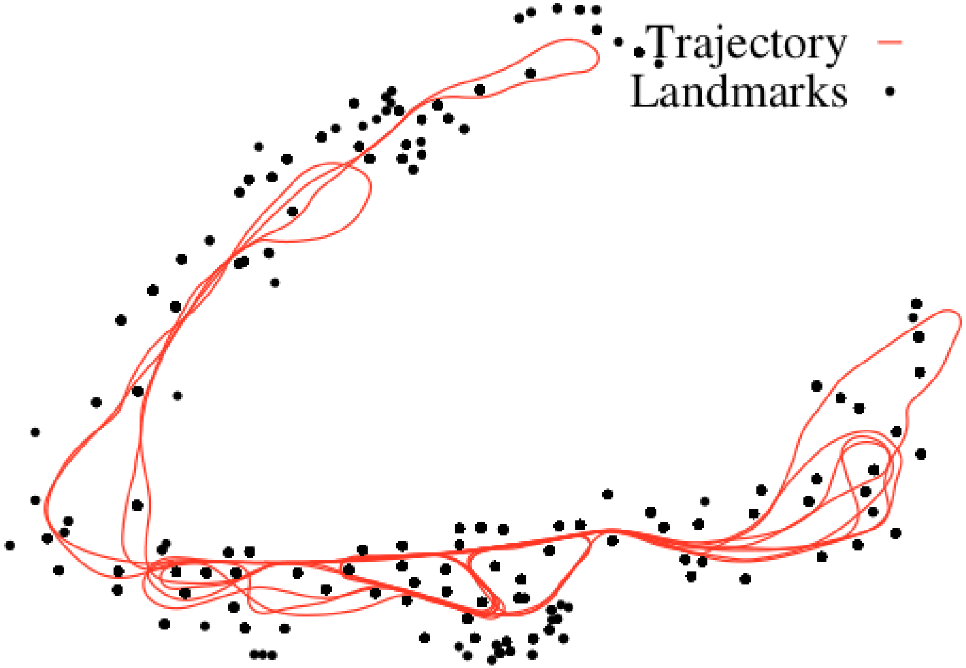
ville.lehtola@nls.fi

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

2D odometry & 2D landmarks



Before



After

"Victoria Park, Fig 1"
Kümmerle, Rainer, et al. "g 2 o: A general framework for graph optimization." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

Spherical environment constraint (1)

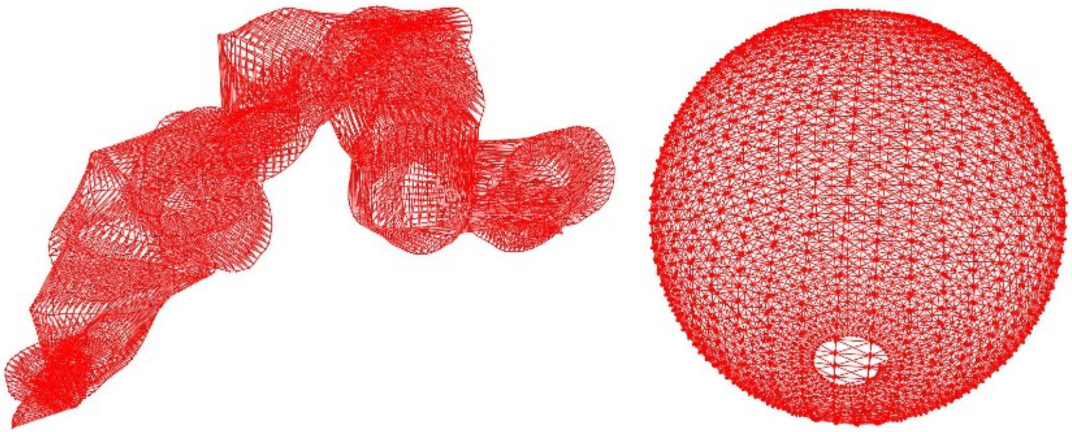


Fig. 11. Pose-graph obtained by simulating a robot moving on a sphere. Left: Initial configuration. Right: After optimizing the pose graph the sphere has accurately been recovered by Algorithm 2.

Image: A Tutorial on Graph-Based SLAM, Grisetti et al.

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Spherical environment constraint (2)

- Simulated SLAM
- Measurement model:
  - Laser observations form a spherical surface

ville.lehtola@nls.fi

Graph SLAM with forest data (1)

- Problem: Tree foliage causes GNSS errors
  - Acquired trajectory & 3D point cloud is noisy
- Solution: GraphSLAM
  - Correct trajectory

Kukko, A., Kaijaluoto, R., Kaartinen, H., Lehtola, V. V., Jaakkola, A., & Hyyppä, J. (2017). Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. ISPRS Journal of Photogrammetry and Remote Sensing, 132, 199-209.

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Graph SLAM with forest data (2)

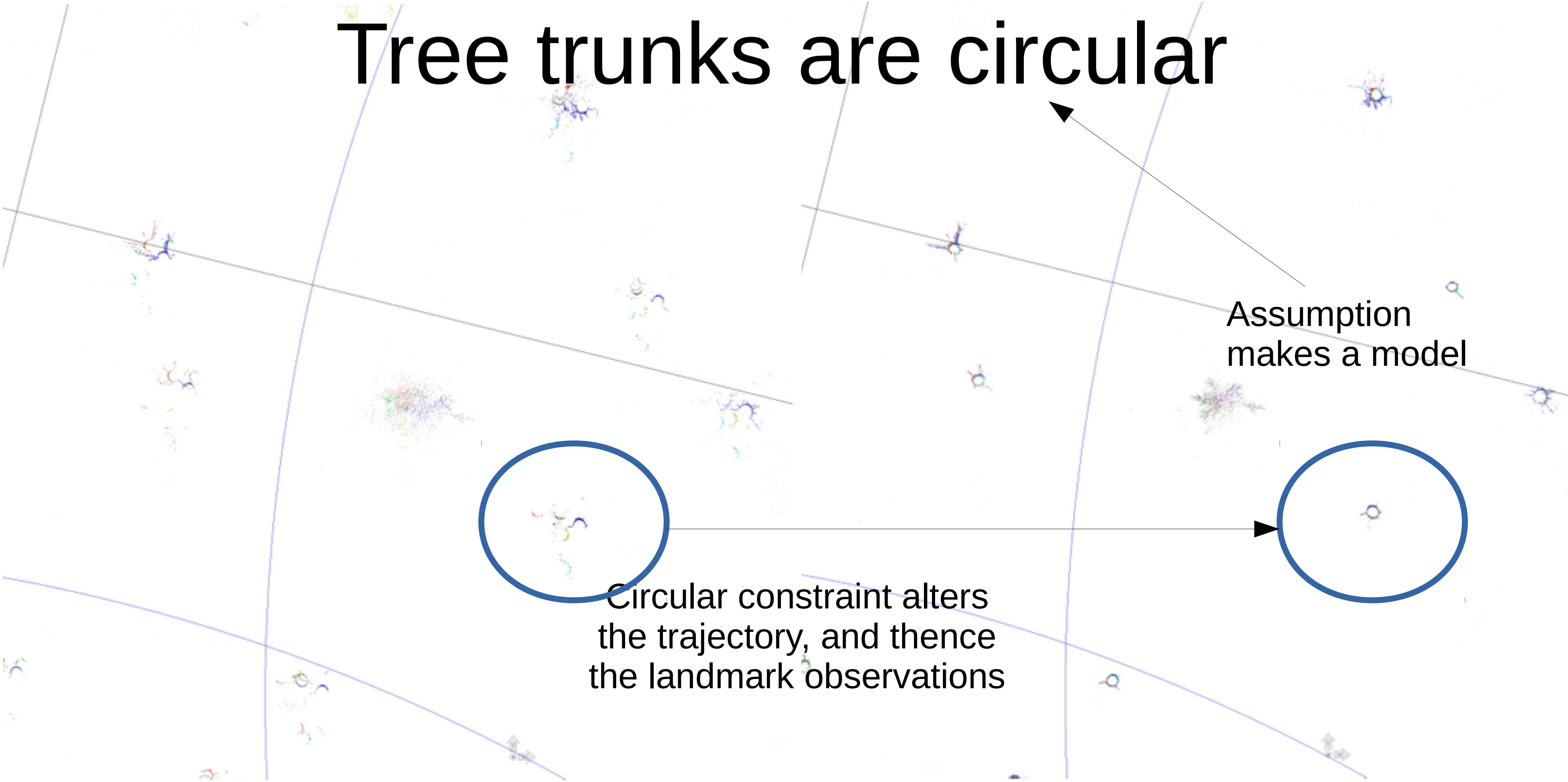
- Nodes: the trajectory is formulated as a graph where the poses at consecutive timestamps (200 Hz in our case) and the detected features at certain time instance (captured as a mean of the feature points’ timestamps) form the nodes
- Edges (or constraints) between the nodes are formed from the measured relative transformations between them.

ville.lehtola@nls.fi

Collect data
↓
Georeference measurements with GNSS/IMU based trajectory
↓
Segment ground points
↓
Extract horizontal slice of point cloud at constant offset from ground
↓
Detect tree trunk by looking for arcs of points
↓
Associate tree trunk detections representing same tree with each other
↓
Create a graph represention of the initial trajectory and tree trunk detections and optimize it
↓
Georeference the measurements again with corrected trajectory

ville.lehtola@nls.fi

Tree trunks are circular



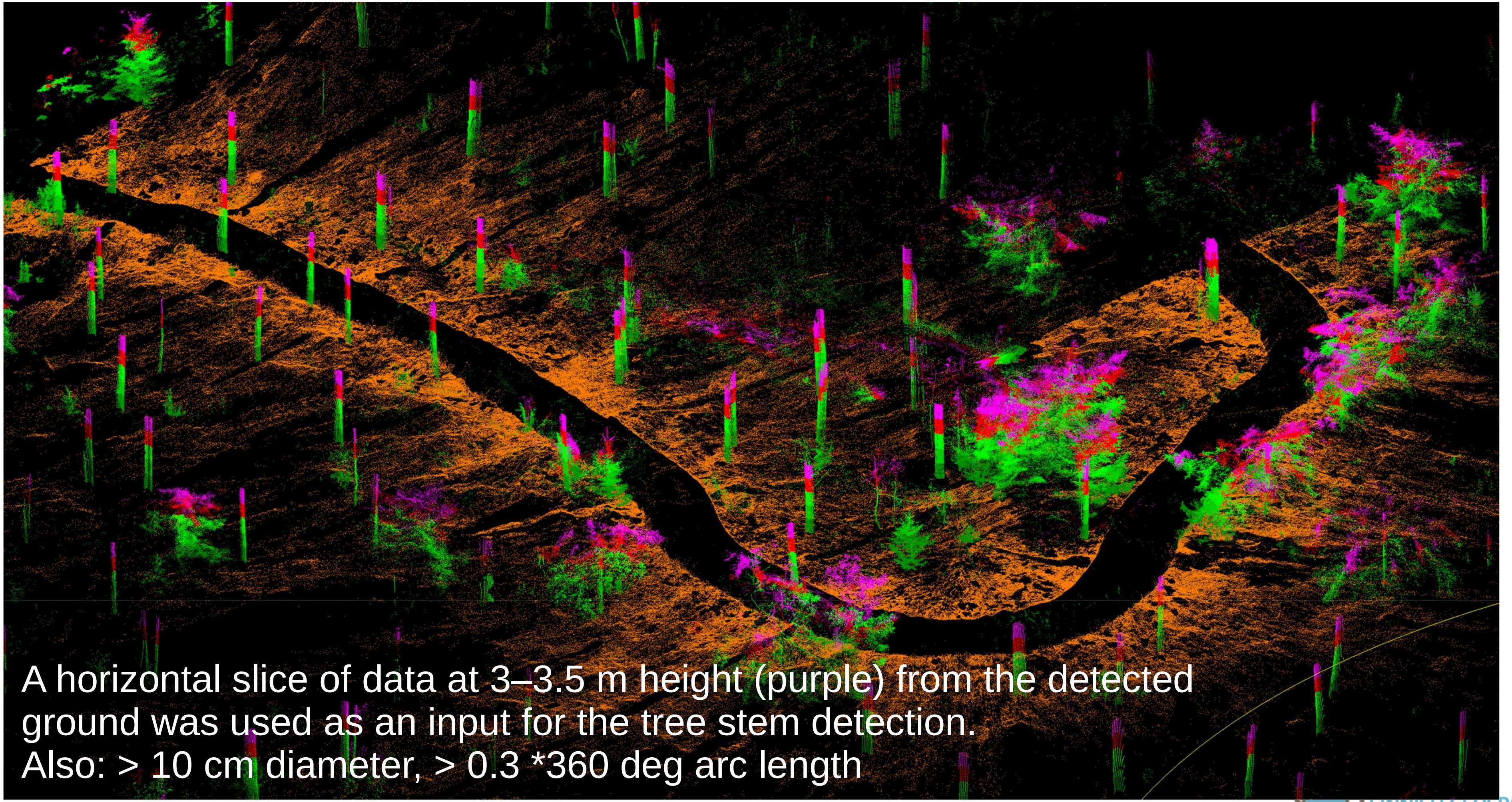
a b

Graph SLAM with forest data (3)

- Construct the graph: the trajectory is expressed as poses.
  - Optimize the graph, i.e. correct the trajectory, iterate until convergence
  - Measurement model constraint: See if tree stems appear as circles from above
  - Motion model constraint: consecutive timestamps (200 Hz), no control signals
  - Result: 6 cm mean error in absolute tree stem locations

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

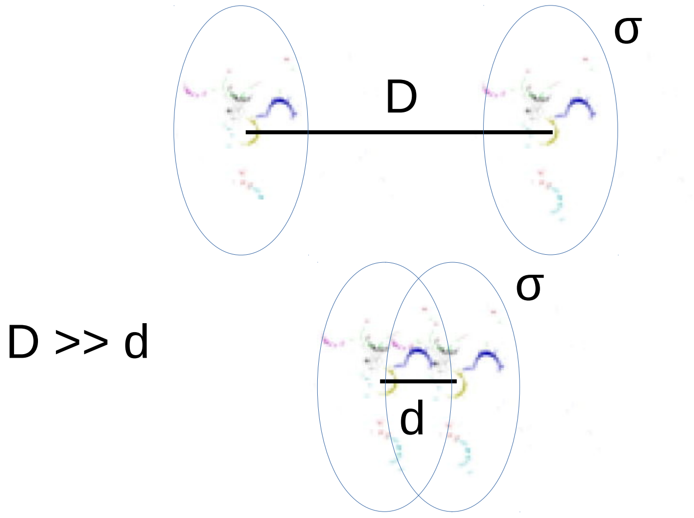
ville.lehtola@nls.fi



FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Graph SLAM with forest data (4)



- Method limitations
- Correspondence of landmarks
  - Which points belong to which trees?
  - What happens
    - if D → d?

ville.lehtola@nls.fi

Graph SLAM with forest data (5)

- Method limitations:
  - observations from different trees need to be separable
  - i.e. initial correspondences must be good
  - Ergo: method fails in a dense forest where the distances between trees are smaller and the trajectory noise is larger

Information matrix Ω (1)

- Inverse of the covariance matrix

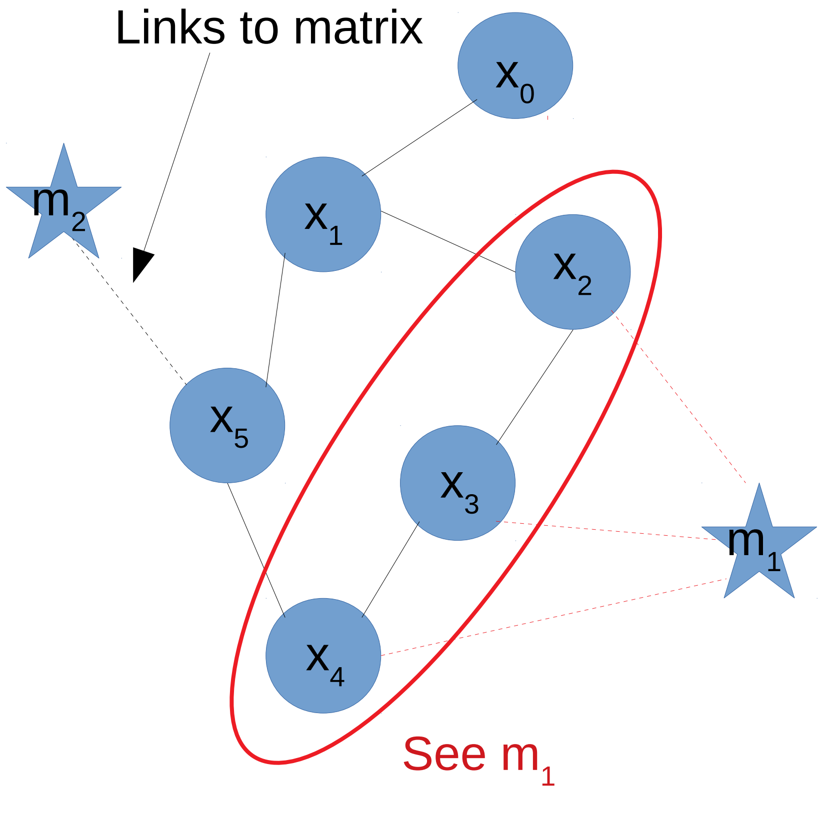
$$\Omega = \Sigma^{-1}$$

- Connected to the information vector

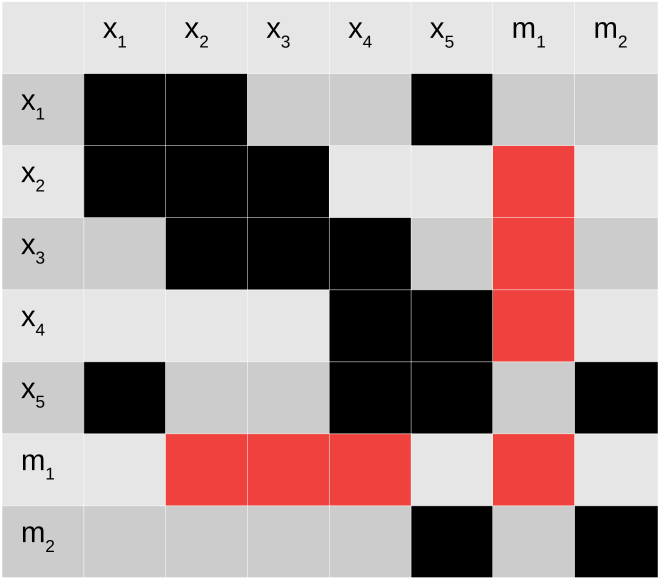
$$\xi = \Sigma^{-1} \mu = \Omega \mu$$

ville.lehtola@nls.fi

Information matrix Ω (2)



- Graph links are represented in matrix



Ω

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Information matrix Ω (3)

Landmark correspondencies

- Linearize:
- $J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(y_t, c_t^i)] Q^{-1} [z_t - h(y_t, c_t^i)]$

Quadratic and linear in x_t

Quadratic and linear in y_t

- Collect quadratic terms to Ω and linear to ξ

$$J_{graphSLAM} = const - \frac{1}{2} y_{0:t}^T \Omega \, y_{0:t} + y_{0:t}^T \xi$$

Voilà!

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

ville.lehtola@nls.fi

Solving GraphSLAM

- GraphSLAM graph

$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})] + \\ + \sum_t [z_t - h(y_t, c_t^i)] Q^{-1} [z_t - h(y_t, c_t^i)]$$

- (or use conjugate gradient or gradient descent)

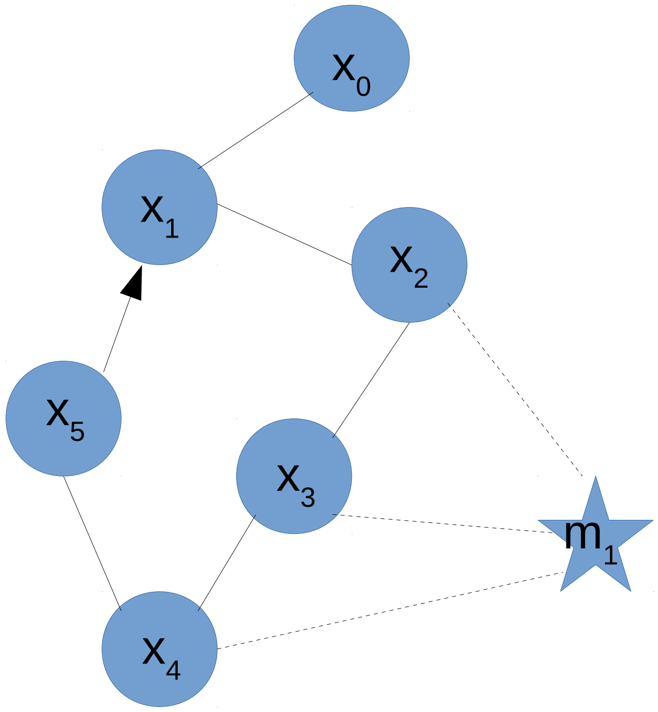
- Levenberg-Marquardt

$$F = \underset{\alpha}{argmin} \sum_{i=1}^{m} [y_i - f(x_i, \alpha)]^2$$

  - Parameter vector α
  - Damped least squares

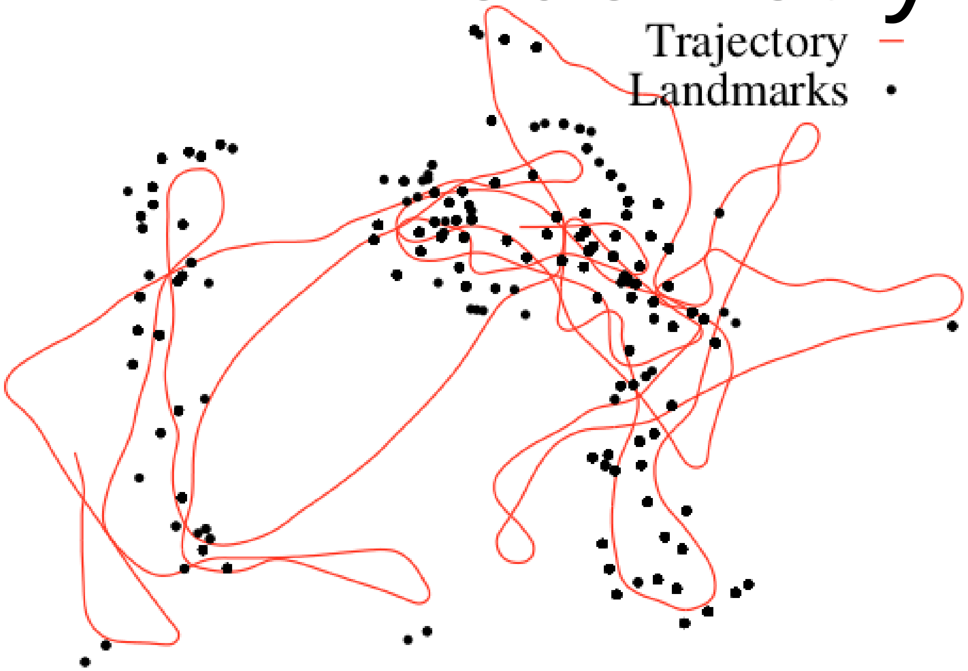
ville.lehtola@nls.fi

Loop closures (1)

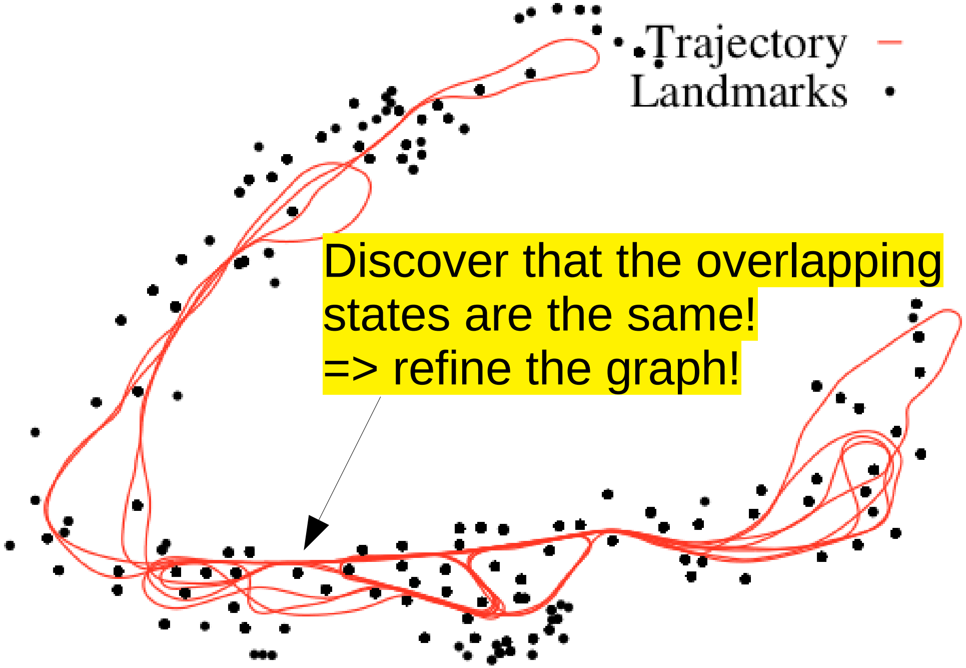


- We see that x_6=x_1 !
- Hence, there is a loop in the graph
- But how do we actually recognize that x_6=x_1 ?

Loop closures (2)
2D odometry & 2D landmarks



Before



After

“Victoria Park, Fig 1”
Kümmerle, Rainer, et al. "g 2 o: A general framework for graph optimization." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

Loop closures (3)

Note the similarity

$$X^*, S^* = \underset{X,S}{\operatorname{argmin}} \underbrace{\sum_i \|f(\mathbf{x}_i,\mathbf{u}_i) - \mathbf{x}_{i+1}\|^2_{\Sigma_i}}_{\text{Odometry Constraints}}$$
$$+ \underbrace{\sum_{ij} \|\,\mathrm{sig}(s_{ij})\cdot (f(\mathbf{x}_i,\mathbf{u}_{ij})-\mathbf{x}_j)\,\|^2_{\Lambda_{ij}}}_{\text{Switched Loop Closure Constraints}}$$
$$+ \underbrace{\sum_{i,j}\|\gamma_{ij}-s_{ij}\|^2_{\Xi_{ij}}}_{\text{Switch Prior Constraints}}$$

On/off switch [0,1]

Exclude trivial solutions

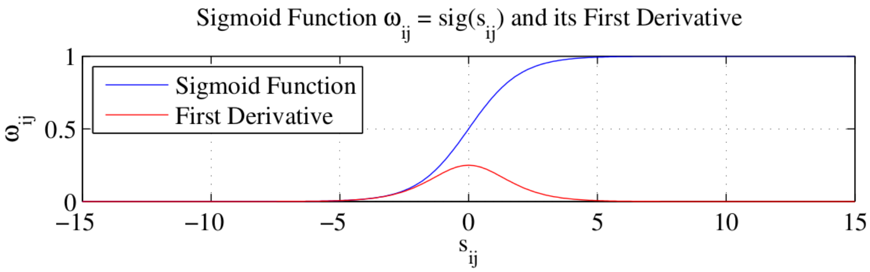
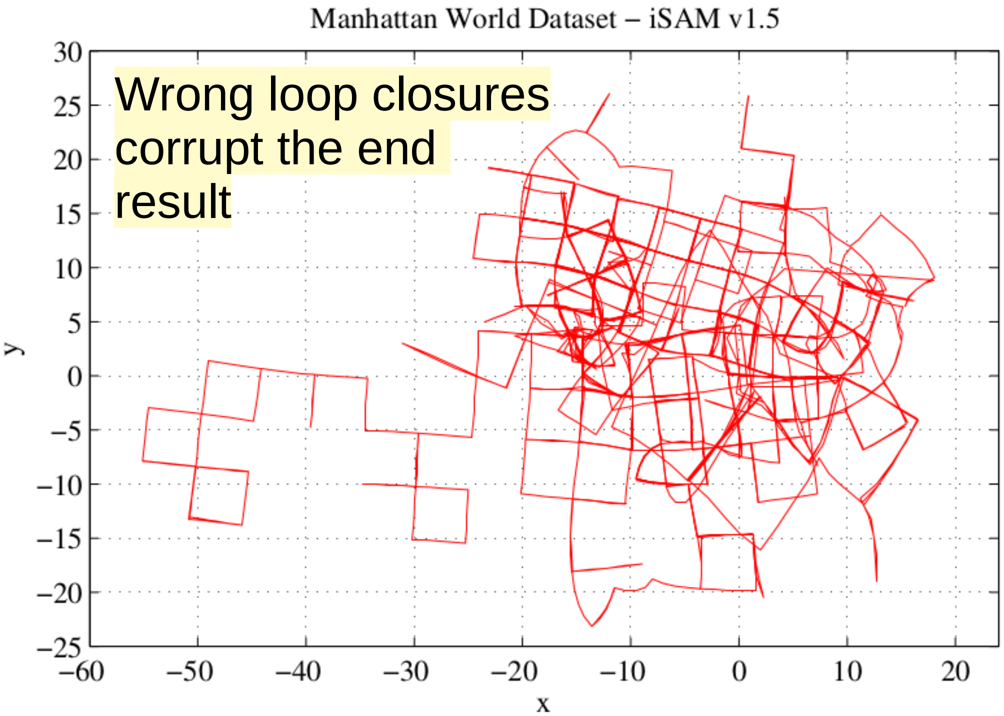
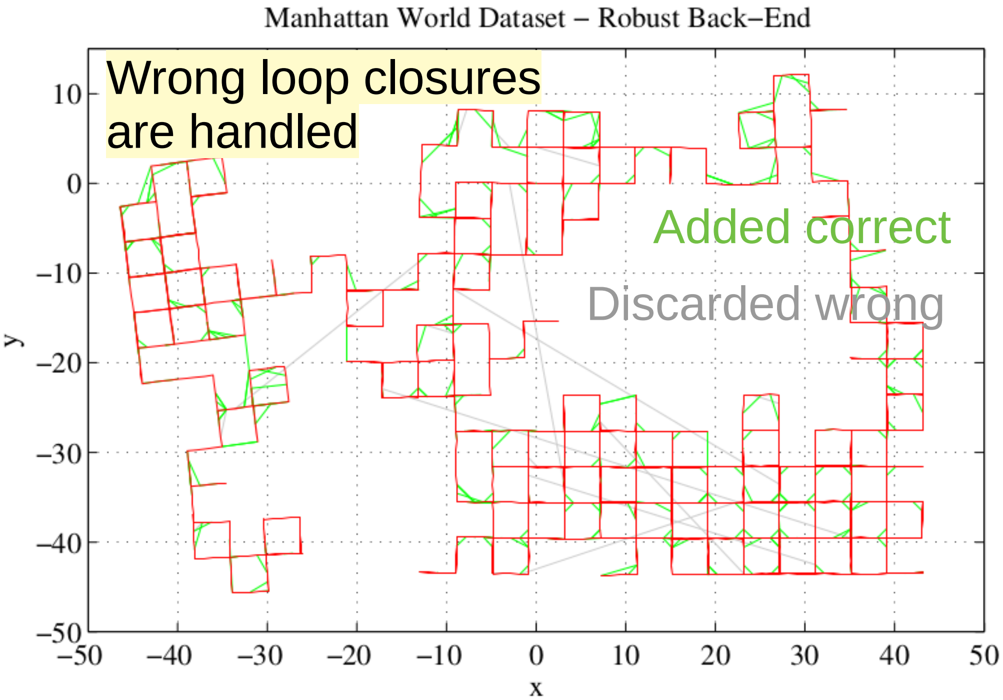


Fig. 4: The sigmoid function $\mathrm{sig}(s_{ij}) = 1/(1+e^{-s_{ij}})$ and its derivative $\mathrm{sig}'(s_{ij}) = \mathrm{sig}(s_{ij})\cdot(1-\mathrm{sig}(s_{ij}))$

Sünderhauf, Niko, and Peter Protzel. "Switchable constraints for robust pose graph SLAM." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

Loop closures (4)





Sünderhauf, Niko, and Peter Protzel. "Switchable constraints for robust pose graph SLAM." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

Pose matching

$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)] Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

$$[z_t - h(y_t, c_t^i)] \rightarrow [x_i - f(u_{ij}, x_j)]$$

We can match any 2 poses:

$$x_i = f(u_{ij}, x_j) + \lambda_{ij}$$

Gaussian noise

=> Pose match check for x_i and x_j

- We can replace the landmark function h() with a pose-match function f()
  - e.g. occupancy grid-based, vision

ville.lehtola@nls.fi

Pose matching (2)
Graph of key maps

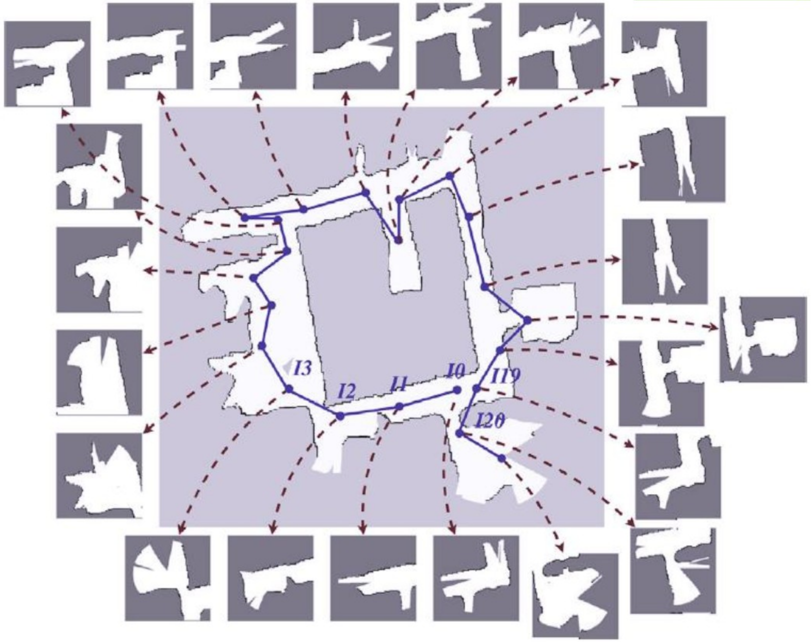


Fig. 3. Graph of key-maps along the route of the robot.

- Laser system
- Global map and local maps
- Each state is represented by a local 2D occupancy grid
- Visual feature techniques used to search for correspondences

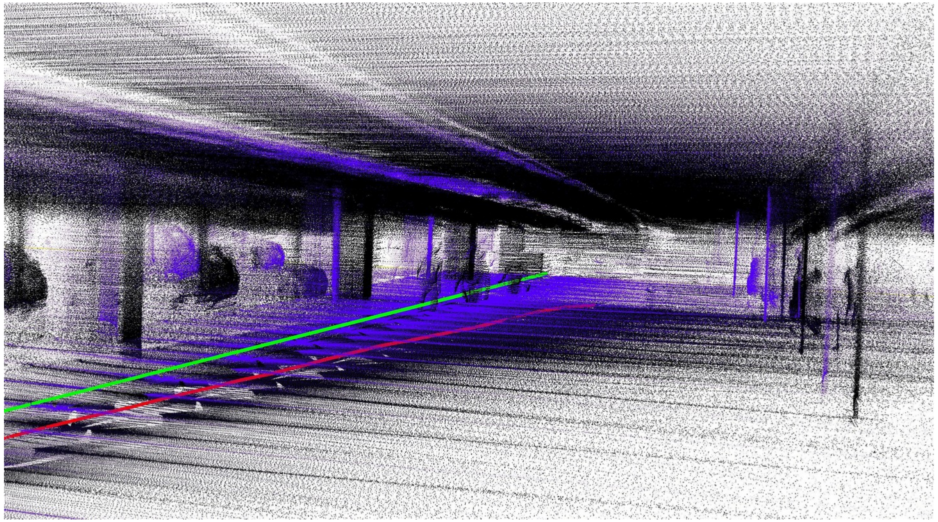
Gil, A., Juliá, M., & Reinoso, Ó. (2015). Occupancy grid based graph-SLAM using the distance transform, SURF features and SGD. Engineering Applications of Artificial Intelligence, 40, 1-10.

3D pointclouds (1)

- "Each point is a landmark"

$$E(\mathbf{R},\mathbf{t}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})||^2$$

  - Forget correspondence, use iterative closest point (ICP)
    - rotation&translation
  - Correct the trajectory

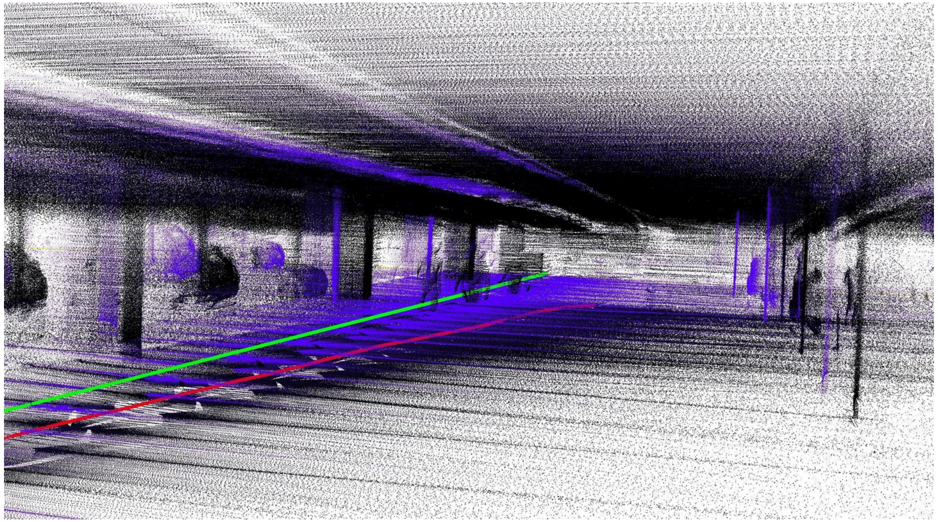


Lehtola, V. V., et al. "Localization corrections for mobile laser scanner using local support-based outlier filtering." ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 3 (2016): 81.

Nüchter, A., Lingemann, K., Hertzberg, J., & Surmann, H. (2007). 6D SLAM—3D mapping outdoor environments. Journal of Field Robotics, 24(8-9), 699-722.

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

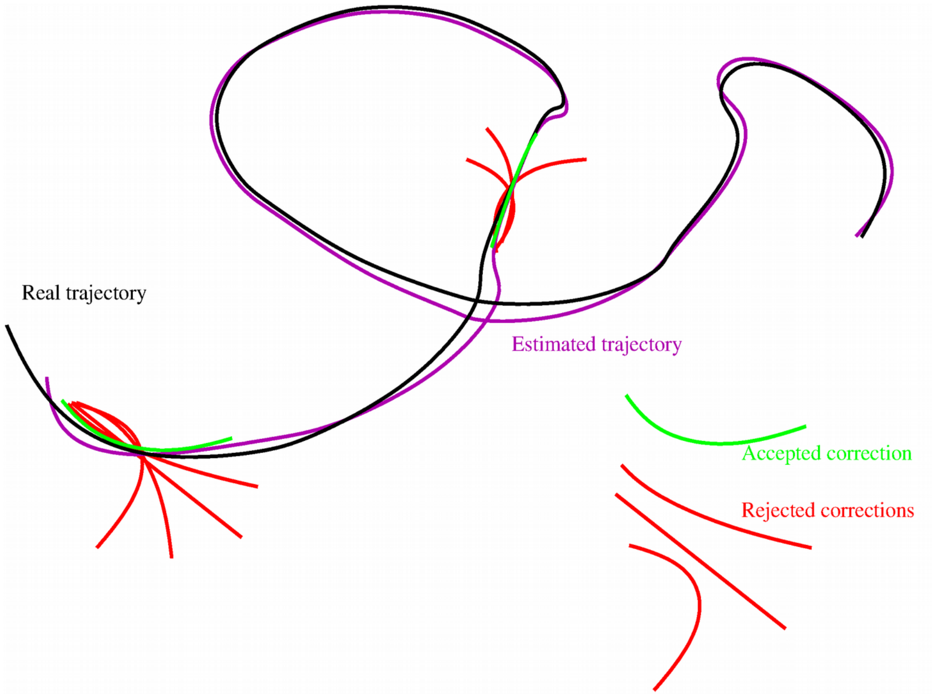
3D pointclouds (2)



- “Each point is a landmark”
  - Need to downsample!
  - But how to overcome local minima in energy function minimization?
  - ICP does not work, it is greedy

ville.lehtola@nls.fi

3D pointclouds (3)

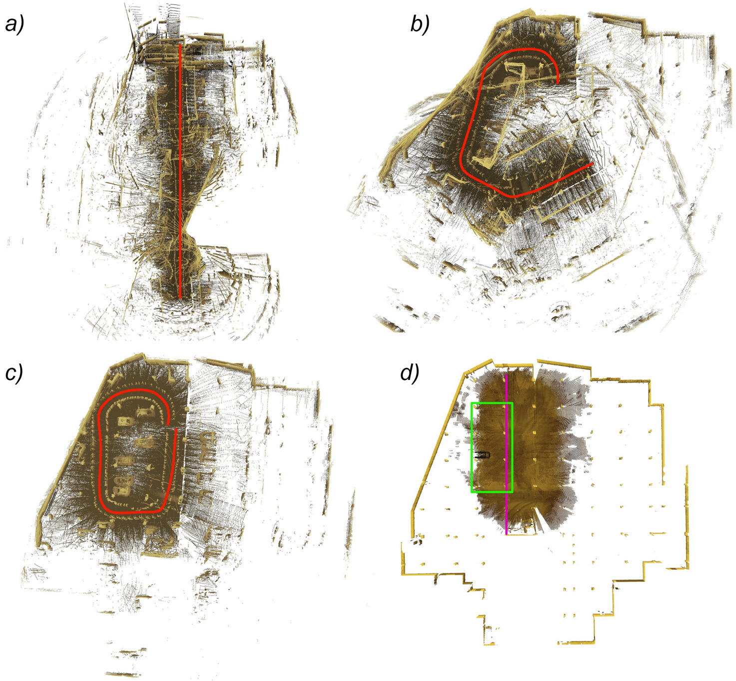


- Build an error function to evaluate different curve pieces
  - Always pick the best
- Obtain trajectory estimate for ICP!

Lehtola, Ville V., et al. "Localization of a mobile laser scanner via dimensional reduction." ISPRS Journal of Photogrammetry and Remote Sensing 121 (2016): 48-59.

ville.lehtola@nls.fi

3D pointclouds (4)



- "Each point is a landmark"
  - Bend the trajectory with curve-pieces
  - Bend again with different characteristic length
  - Apply ICP
  - Done

Lehtola, Ville V., et al. "Localization of a mobile laser scanner via dimensional reduction." ISPRS Journal of Photogrammetry and Remote Sensing 121 (2016): 48-59.

ville.lehtola@nls.fi

NLS FINNISH GEOSPATIAL RESEARCH INSTITUTE FGI

More information

- A Tutorial on Graph-Based SLAM http://www2.informatik.uni-freiburg.de/~stachnis/pdf/grisetti10titsmag.pdf
- Book: “Probabilistic robotics”, Thrun, Burgard, and Fox

Available methods

- g2o: A General Framework for Graph Optimization https://openslam.org/g2o.html
- C++ code, LGPL v3, Open source
- Others: TreeMap, TORO, sqrt(SAM), iSAM, Sparse Pose Adjustment, iSAM2

ville.lehtola@nls.fi

Graph SLAM – learning summary

- GraphSLAM solves the full SLAM problem offline
  - Measurement constraints integrate the measurement model
    - Landmarks
  - Motion constraints integrate the motion model
    - Data from controls not necessarily needed (use IMU or smoothing)
- Ability to build large scale global maps
  - Sparse graph, motion constraints build linearly in time
  - Eased loop closure
  - Amount of landmarks may be very large, > 1000

$$J_{graphSLAM} = x_0^T \Omega_0 x_0^T + \sum_t [x_t - g(u_t, x_{t-1})] R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)] Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

Initial or anchor constraint

ville.lehtola@nls.fi