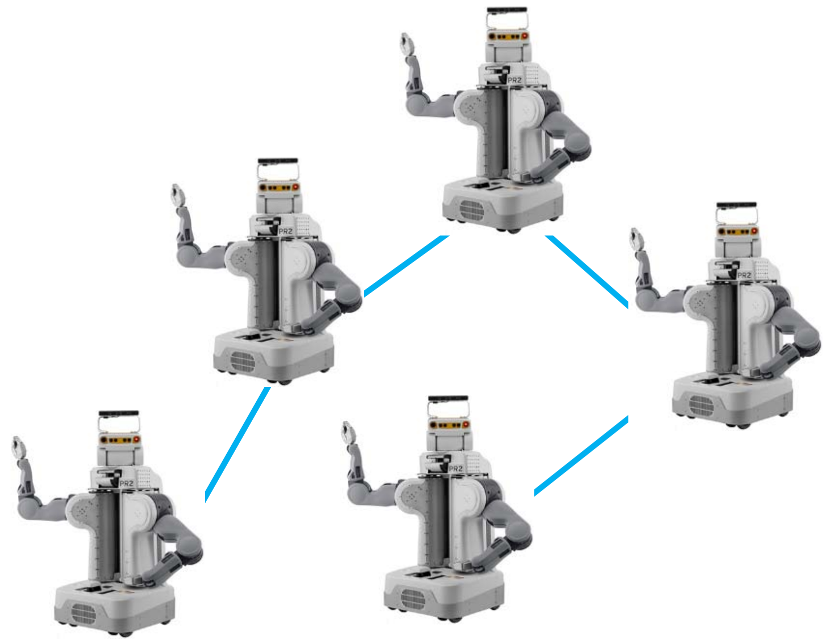
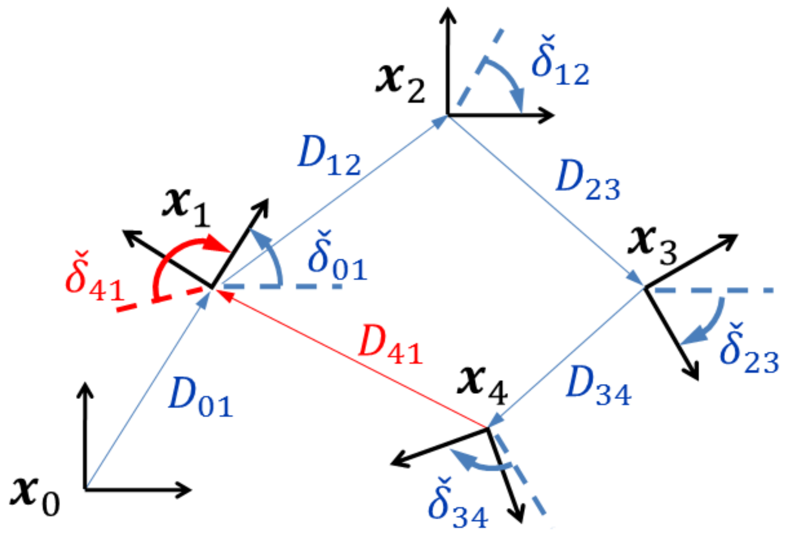
Pose graph optimization

The objective of pose graph optimization is to estimate robot trajectory (collection of poses) from relative pose measurements





- **Odometric constraints**
- **Loop closing contraints**

- Relative pose measurements obtained from:
  - Self-motion (wheel odometry, IMU)
  - Scan matching
  - Visual feature registration (sparse features)
  - ICP & variants (dense point clouds)
  - ...

Pose graph optimization

Related work

- Lu and Milios [Autonomous Robots 1997]: first formulation
- Frese et al. [TRO 2005]: multi-level relaxation
- Thrun and Montemerlo [IJRR 2006]: conjugate gradient-based optimization
- Olson et al. [ICRA 2006]: new parametrization with improved convergence
- Grisetti et al. [ITS 2009]: stochastic gradient on a tree parametrization (TORO)
- Kuemmerle et al. [ICRA 2011]: Gauss-Newton on manifold (g2o)
- Kaess et al. [IJRR 2012]: incremental algorithm (iSAM2)
- Olson and Agarwal [RSS 2012]: estimation robustness
- Sünderhauf and Protzel [ICRA 2012, IROS 2012]: estimation robustness
- Huang et al. [ICRA 2010, RSS 2012]: convergence properties

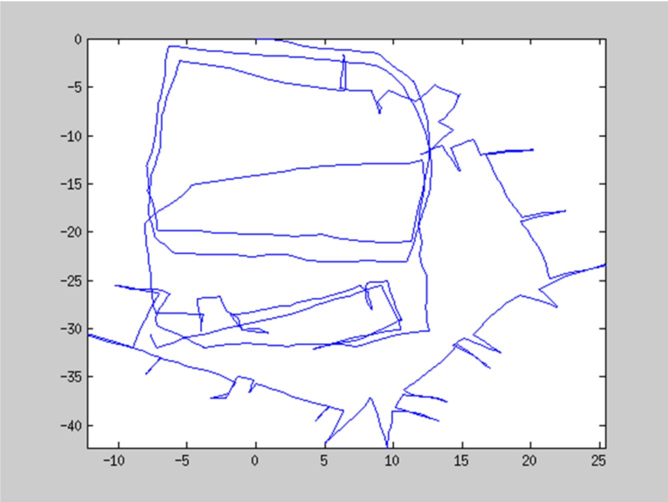
Pose graph optimization

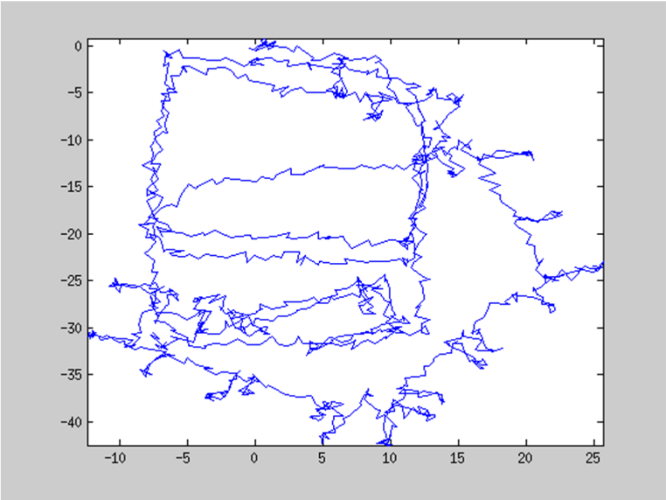
Related work

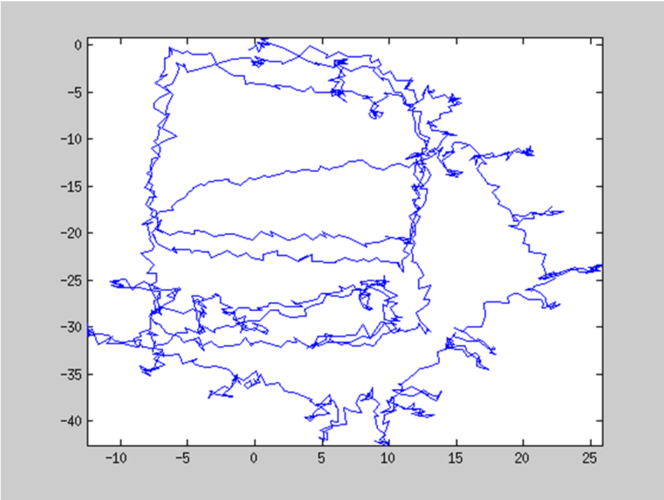
- Pose graph optimization leads to a hard non-convex optimization problem
- State-of-the-art approaches apply iterative optimization techniques:

1. start from an initial guess
2. approximate the original problem with a convex problem
3. solve the convex problem, obtaining a new initial guess
4. repeat from 2. until convergence

ISSUES:
- Local minima
- No uncertainty quantification







Pose graph optimization

- We now show that in planar pose graph optimization:

1. If the orientations of the robot are known, than pose graph optimization is a convex problem

2. Estimating robot orientations is hard (angles on manifold, multiple local minima)

3. A smart reparametrization of the problem allows to compute a global solution of the orientation estimation problem

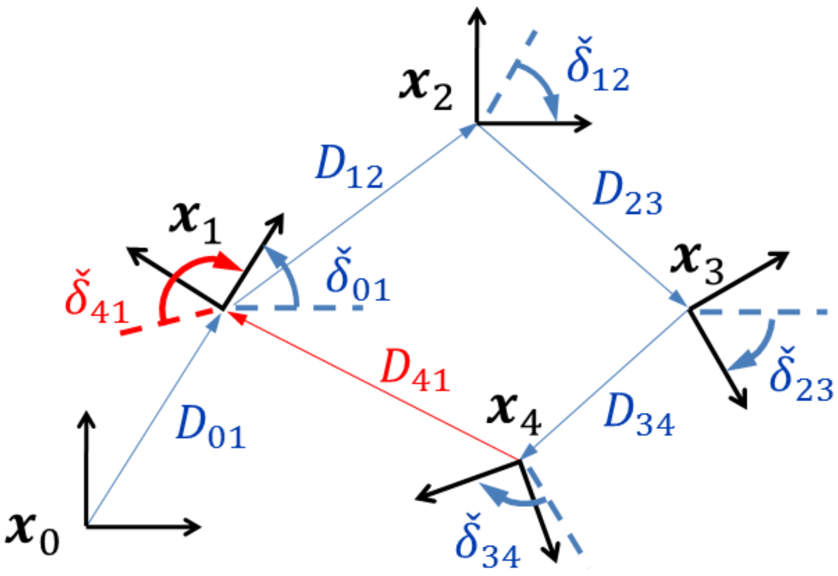
4. The global solution is unique with probability one

5. We can quantify how far is the estimate from the real robot orientations

- But we need some more notation...

Pose graph optimization

$$\boldsymbol{x}_i \begin{cases} \boldsymbol{p}_i \in \mathbb{R}^2 & \text{position of node } i \\ \check{\theta}_i \in (-\pi, +\pi] & \text{orientation of node } i \end{cases}$$



- **Input:**
  m relative pose measurements for all $(i,j) \in \mathcal{E}$

relative position measurements

$$\boldsymbol{D}_{ij} = \boldsymbol{R}(\check{\theta}_i)^{\mathsf{T}}(\boldsymbol{p}_j - \boldsymbol{p}_i) + \text{noise}$$

relative orientation measurements

$$\check{\delta}_{ij} = \langle \check{\theta}_j - \check{\theta}_i + \text{noise} \rangle_{2\pi}$$

- **Output:**
  Maximum likelihood estimate of the poses $\boldsymbol{x}^\star = (\boldsymbol{p}_0^\star \;\; \boldsymbol{p}_1^\star \;\; \cdots \;\; \boldsymbol{p}_n^\star \;\; \check{\theta}_0^\star \;\; \check{\theta}_1^\star \;\; \cdots \;\; \check{\theta}_n^\star)^{\mathsf{T}}$

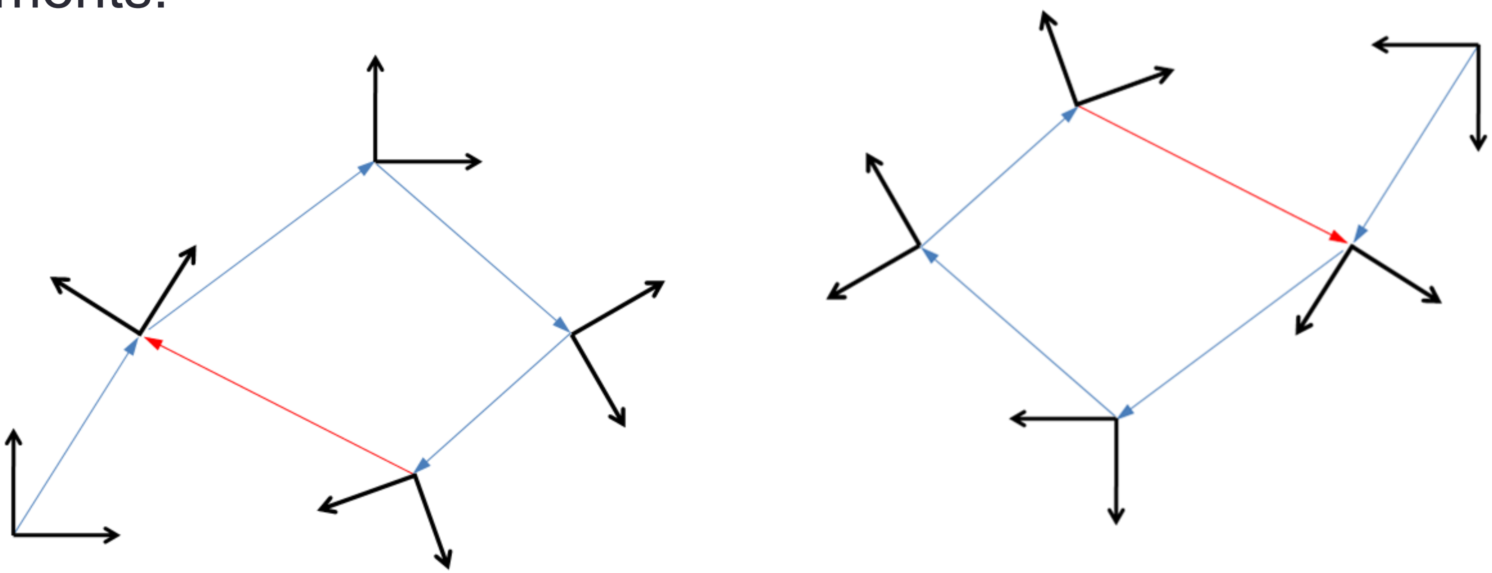
modulus 2π

$$\langle \cdot \rangle_{2\pi} : \mathbb{R} \to (-\pi, +\pi]$$

$$\langle \omega \rangle_{2\pi} \doteq \omega + 2\pi \left\lfloor \frac{\pi - \omega}{2\pi} \right\rfloor$$

Pose graph optimization

- Several (infinite) poses may produce the same relative pose measurements:



- Absolute poses are not observable from <u>relative</u> pose measurements
- We assume that the first pose is the origin of the reference frame:

$$\boldsymbol{p}_0 = \mathbf{0}_2 \qquad \check{\theta}_0 = 0$$

- Therefore, the maximum likelihood estimate becomes:

$$\boldsymbol{x}^\star = (\boldsymbol{p}_1^\star \ \cdots \ \boldsymbol{p}_n^\star \ \ \check{\theta}_1^\star \ \cdots \ \check{\theta}_n^\star)^\mathsf{T} = (\boldsymbol{p}^\star, \check{\boldsymbol{\theta}}^\star)$$

Pose graph optimization

- Assuming Gaussian noise, the maximum likelihood estimate can be obtained by minimizing the weighted sum of the residual errors:

$$(\boldsymbol{p}^\star, \breve{\boldsymbol{\theta}}^\star) = \underset{\boldsymbol{p}\in\mathbb{R}^{2n}, \breve{\boldsymbol{\theta}}\in(-\pi,+\pi]^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \underbrace{\left\| \boldsymbol{R}_i^\top (\boldsymbol{p}_j - \boldsymbol{p}_i) - \boldsymbol{D}_{ij} \right\|^2_{\boldsymbol{P}^{-1}_{\boldsymbol{D}_{ij}}}}_{\text{Position estimation subproblem}} + \underbrace{\left\| \langle \breve{\theta}_j - \breve{\theta}_i - \breve{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}}_{\text{Orientation estimation subproblem}}$$

- We use the standard notation for the Mahalanobis distance:

$$\|\boldsymbol{y}\|^2_{\boldsymbol{P}} = \boldsymbol{y}^\top \boldsymbol{P} \, \boldsymbol{y}$$

Pose graph optimization

- Assuming Gaussian noise, the maximum likelihood estimate can be obtained by minimizing the weighted sum of the residual errors:

$$(\boldsymbol{p}^\star, \breve{\boldsymbol{\theta}}^\star) = \underset{\boldsymbol{p}\in\mathbb{R}^{2n}, \breve{\boldsymbol{\theta}}\in(-\pi,+\pi]^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \boldsymbol{R}_i^\mathsf{T}(\boldsymbol{p}_j - \boldsymbol{p}_i) - \boldsymbol{D}_{ij} \right\|^2_{\boldsymbol{P}^{-1}_{\boldsymbol{D}_{ij}}} + \left\| \langle \breve{\theta}_j - \breve{\theta}_i - \breve{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

1. If the orientations of the robot are known, than pose graph optimization is a convex problem

$$(\boldsymbol{p}^\star, \breve{\boldsymbol{\theta}}^\star) = \underset{\boldsymbol{p}\in\mathbb{R}^{2n}, \breve{\boldsymbol{\theta}}\in(-\pi,+\pi]^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \boldsymbol{R}_i^\mathsf{T}(\boldsymbol{p}_j - \boldsymbol{p}_i) - \boldsymbol{D}_{ij} \right\|^2_{\boldsymbol{P}^{-1}_{\boldsymbol{D}_{ij}}} + \cancel{\left\| \langle \breve{\theta}_j - \breve{\theta}_i - \breve{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}}$$

known!

known!

- Position estimation becomes a standard least squares problem

$$\min_{\boldsymbol{p}} \left\| \boldsymbol{X}^\mathsf{T}\boldsymbol{p} - \boldsymbol{y} \right\|^2$$

- It can be solved in closed form

$$\boldsymbol{p} = (\boldsymbol{X}\boldsymbol{X}^\mathsf{T})^{-1}\boldsymbol{X}\boldsymbol{y}$$

Analysis of the distributed and centralized setup by Barooah and Hespanha [2009] and Russell et al. [2011]

Global orientation estimation

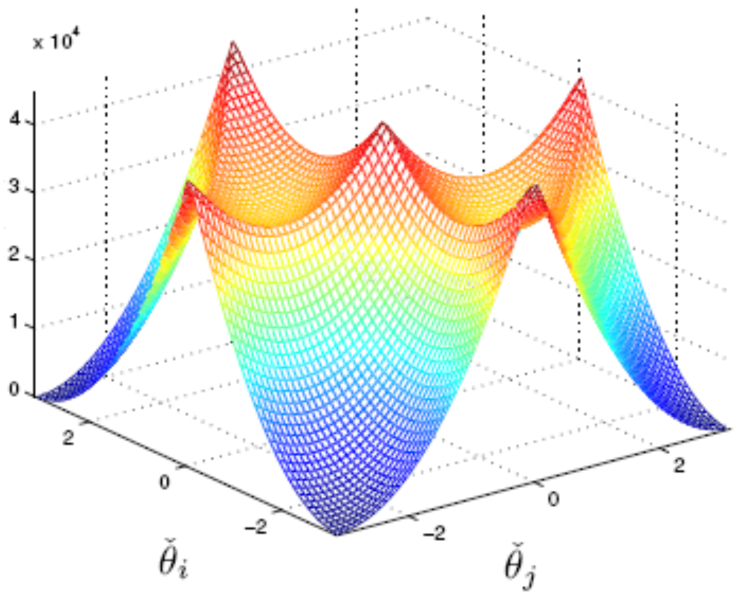
- Now the idea is to solve first the orientation estimation subproblem, and then use the resulting estimate for solving the original problem

$$S^1 = \arg\min_{\check{\boldsymbol{\theta}} \in (-\pi, +\pi]^n} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \check{\theta}_j - \check{\theta}_i - \check{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

2. Estimating robot orientation is <u>hard</u> (angles on manifold, multiple local minima)

- Iterative optimization techniques
  - need initial guess
  - incur in local minima

- It is not even clear if the orientation estimation problem admits a unique global minimum



Global orientation estimation

3. A smart reparametrization of the problem allows to compute a <u>global solution</u> of the orientation estimation problem

Original problem on *n* angles

$$S^1 = \arg\min_{\boxed{\check{\boldsymbol{\theta}} \in (-\pi,+\pi]^n}} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \check{\theta}_j - \check{\theta}_i - \check{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

Constrained optimization

Unconstrained optimization

Mixed-integer optimization

Integer optimization

Integer optimization (v2.0)

Final formulation on ℓ integers

$$S^5 = \arg\min_{\gamma\in\mathbb{Z}^\ell} \|\gamma - \hat{\gamma}\|^2_{\boldsymbol{P}^{-1}_{\gamma}}$$

n = number of nodes in the graph

ℓ = number of cycles in the graph

Global orientation estimation

- Constrained optimization
- Unconstrained optimization
- Mixed-integer optimization
- Integer optimization
- Integer optimization (v2.0)

- The first transformation leads to an unconstrained optimization problem

$$S^1 = \underset{\breve{\boldsymbol{\theta}} \in (-\pi, +\pi]^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \breve{\theta}_j - \breve{\theta}_i - \breve{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

$$\boxed{\boldsymbol{\theta}^\star \to \langle \boldsymbol{\theta}^\star \rangle_{2\pi}}$$

$$S^2 = \underset{\boldsymbol{\theta} \in \mathbb{R}^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \theta_j - \theta_i - \breve{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

Spoiler: removing constraints may simplify the problem

$$\min_{\boldsymbol{p}} \|\boldsymbol{X}^\mathsf{T}\boldsymbol{p} - \boldsymbol{y}\|^2$$

$$\min_{\boldsymbol{p}\in\mathrm{Set}} \|\boldsymbol{X}^\mathsf{T}\boldsymbol{p} - \boldsymbol{y}\|^2$$

The cost function is still non-convex and hard to minimize

Global orientation estimation

- Constrained optimization
- Unconstrained optimization
- Mixed-integer optimization
- Integer optimization
- Integer optimization (v2.0)

- The second transformation gets rid of the modulus 2π

$$S^2 = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \theta_j - \theta_i - \check{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

$$\boxed{(\boldsymbol{\theta}^\star, \boldsymbol{k}^\star) \rightarrow \boldsymbol{\theta}^\star}$$

$$S^3 = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^n, k_{ij} \in \mathbb{Z}} \sum_{(i,j)\in\mathcal{E}} \left\| \theta_j - \theta_i - \check{\delta}_{ij} + 2\pi k_{ij} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

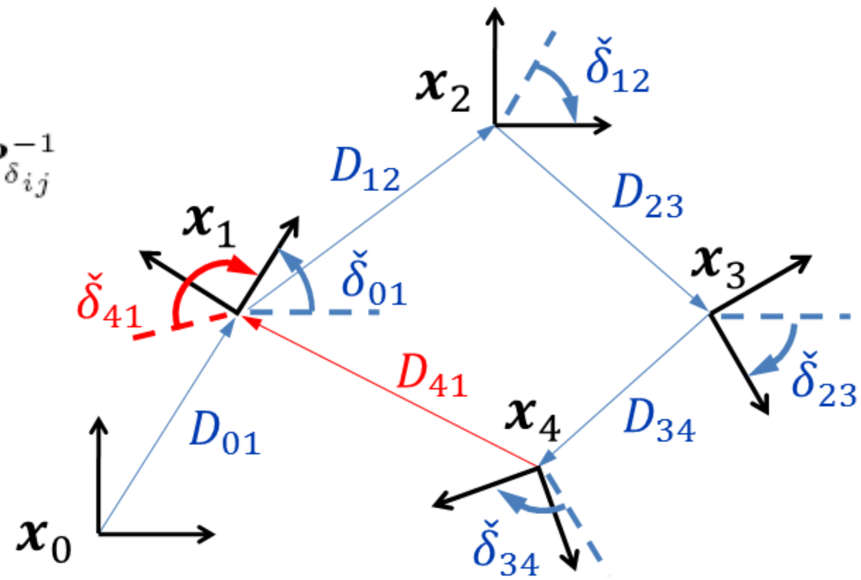
Trick: overparametrize to get rid on the non-convexity

$$| \langle \omega \rangle_{2\pi} | = \min_{k \in \mathbb{Z}} |\omega + 2\pi k|$$

The cost is now quadratic (convex) with both real-valued and integer variables (mixed-integer convex optimization problem)

Global orientation estimation

$$S^3 = \underset{\boldsymbol{\theta}\in\mathbb{R}^n, k_{ij}\in\mathbb{Z}}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \theta_j - \theta_i - \check{\delta}_{ij} + 2\pi k_{ij} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$



- Using matrix notation we obtain:

$$S^3 = \underset{\boldsymbol{\theta}\in\mathbb{R}^n, \boldsymbol{k}\in\mathbb{Z}^m}{\arg\min} \left\| \boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} - \check{\boldsymbol{\delta}} + 2\pi\boldsymbol{k} \right\|^2_{\boldsymbol{P}^{-1}_{\delta}}$$

$$\boxed{\begin{aligned} \check{\boldsymbol{\delta}} &= (\check{\delta}_1 \;\; \check{\delta}_2 \;\; \cdots \;\; \check{\delta}_m)^\mathsf{T} \\ \boldsymbol{k} &= (k_1 \;\; \cdots \;\; k_m)^\mathsf{T} \in \mathbb{Z}^m \\ \boldsymbol{P}_\delta &= \operatorname{diag}(\boldsymbol{P}_{\delta_1}, \ldots, \boldsymbol{P}_{\delta_m}) \in \mathbb{R}^{m\times m} \end{aligned}}$$

$$\overline{\boldsymbol{A}}^\mathsf{T} = \left(\begin{array}{c|cccc} -1 & +1 & 0 & 0 & 0 \\ 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & -1 & +1 & 0 \\ 0 & 0 & 0 & -1 & +1 \\ 0 & +1 & 0 & 0 & -1 \end{array} \right)$$

$\boldsymbol{A}^\mathsf{T}$

reduced incidence matrix

Global orientation estimation

- Given ***k***, the problem becomes quadratic

$$S^3 = \underset{\boldsymbol{\theta}\in\mathbb{R}^n,\ \cancel{\boldsymbol{k}\in\mathbb{Z}^m}}{\arg\min} \left\| \boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} - \check{\boldsymbol{\delta}} + \boxed{2\pi\boldsymbol{k}} \right\|^2_{\boldsymbol{P}_\delta^{-1}}$$

- and can be solved in closed form (unconstrained problem):

$$\boldsymbol{\theta}^{\star|\boldsymbol{k}} = (\boldsymbol{A}\boldsymbol{P}_\delta^{-1}\boldsymbol{A}^\mathsf{T})^{-1}\boldsymbol{A}\boldsymbol{P}_\delta^{-1}(\check{\boldsymbol{\delta}} - 2\pi\boldsymbol{k})$$

Global orientation estimation

Constrained optimization

Unconstrained optimization

Mixed-integer optimization

Integer optimization

Integer optimization (v2.0)

- The third transformation is based on matrix manipulation and algebraic graph theory

$$S^3 = \underset{\boldsymbol{\theta}\in\mathbb{R}^n, \boldsymbol{k}\in\mathbb{Z}^m}{\arg\min} \left\|\boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} - \check{\boldsymbol{\delta}} + 2\pi\boldsymbol{k}\right\|^2_{\boldsymbol{P}_\delta^{-1}}$$

$$\boxed{\boldsymbol{k}^\star \rightarrow (\boldsymbol{\theta}^{\star|\boldsymbol{k}^\star}, \boldsymbol{k}^\star)}$$

$$S^4 = \underset{\boldsymbol{k}\in\mathbb{Z}^m}{\arg\min} \left\|\boldsymbol{C}\boldsymbol{k} - \tfrac{1}{2\pi}\boldsymbol{C}\check{\boldsymbol{\delta}}\right\|^2_{(\boldsymbol{C}\boldsymbol{P}_\delta\boldsymbol{C}^\mathsf{T})^{-1}}$$

Trick: reduced incidence matrix and cycle basis matrix are orthogonal complements

$$\boldsymbol{C}\boldsymbol{A}^\mathsf{T} = \boldsymbol{0}_{\ell\times n}$$

Global orientation estimation

Constrained optimization

Unconstrained optimization

Mixed-integer optimization

Integer optimization

Integer optimization (v2.0)

- Then we can apply a last change of variables, to get the final formulation:

$$\boxed{\boldsymbol{\gamma} = \boldsymbol{C}\boldsymbol{k}}$$

$$S^4 = \arg\min_{\boldsymbol{k} \in \mathbb{Z}^m} \left\| \boldsymbol{C}\boldsymbol{k} - \tfrac{1}{2\pi}\boldsymbol{C}\check{\boldsymbol{\delta}} \right\|^2_{(\boldsymbol{C}\boldsymbol{P}_\delta\boldsymbol{C}^\mathsf{T})^{-1}}$$

$$\boxed{\boldsymbol{\gamma}^\star \to \boldsymbol{C}^\dagger \boldsymbol{\gamma}^\star}$$

$$S^5 = \arg\min_{\boldsymbol{\gamma} \in \mathbb{Z}^\ell} \|\boldsymbol{\gamma} - \hat{\boldsymbol{\gamma}}\|^2_{\boldsymbol{P}_\gamma^{-1}} \quad \text{with} \quad \hat{\boldsymbol{\gamma}} = \tfrac{1}{2\pi}\boldsymbol{C}\check{\boldsymbol{\delta}}, \;\; \boldsymbol{P}_\gamma = \boldsymbol{C}\boldsymbol{P}_\delta\boldsymbol{C}^\mathsf{T}$$

The final formulation is a quadratic optimization problem in integer variables

- Dimension of $\boldsymbol{\gamma}$ is ℓ (number of cycles in the graph)

Global orientation estimation

- Let us recap the transformations of the original problem

$$S^1 = \underset{\check{\boldsymbol{\theta}} \in (-\pi, +\pi]^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \check{\theta}_j - \check{\theta}_i - \check{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

Constrained optimization

$$\check{\boldsymbol{\theta}}^\star = \left\langle \boldsymbol{\theta}^{\star|\boldsymbol{k}^\star} \right\rangle_{2\pi}$$

$$S^2 = \underset{\boldsymbol{\theta} \in \mathbb{R}^n}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left\| \langle \theta_j - \theta_i - \check{\delta}_{ij} \rangle_{2\pi} \right\|^2_{\boldsymbol{P}^{-1}_{\delta_{ij}}}$$

Unconstrained optimization

$$\boldsymbol{\theta}^{\star|\boldsymbol{k}^\star}$$

$$S^3 = \underset{\boldsymbol{\theta}\in\mathbb{R}^n, \boldsymbol{k}\in\mathbb{Z}^m}{\arg\min} \left\| \boldsymbol{A}\boldsymbol{\theta} - \check{\boldsymbol{\delta}} + 2\pi\boldsymbol{k} \right\|^2_{\boldsymbol{P}^{-1}_{\delta}}$$

Mixed-integer optimization

$$\left(\boldsymbol{\theta}^{\star|\boldsymbol{k}^\star}, \boldsymbol{k}^\star\right)$$

$$S^4 = \underset{\boldsymbol{k}\in\mathbb{Z}^m}{\arg\min} \left\| \boldsymbol{C}\boldsymbol{k} - \tfrac{1}{2\pi}\boldsymbol{C}\check{\boldsymbol{\delta}} \right\|^2_{(\boldsymbol{C}\boldsymbol{P}_\delta\boldsymbol{C}^\top)^{-1}}$$

Integer optimization

$$\boldsymbol{k}^\star = \boldsymbol{C}^\dagger \gamma^\star$$

$$S^5 = \underset{\gamma\in\mathbb{Z}^\ell}{\arg\min} \|\gamma - \hat{\gamma}\|^2_{\boldsymbol{P}^{-1}_{\gamma}}$$

Integer optimization (v2.0)

$$\boxed{\gamma^\star}$$

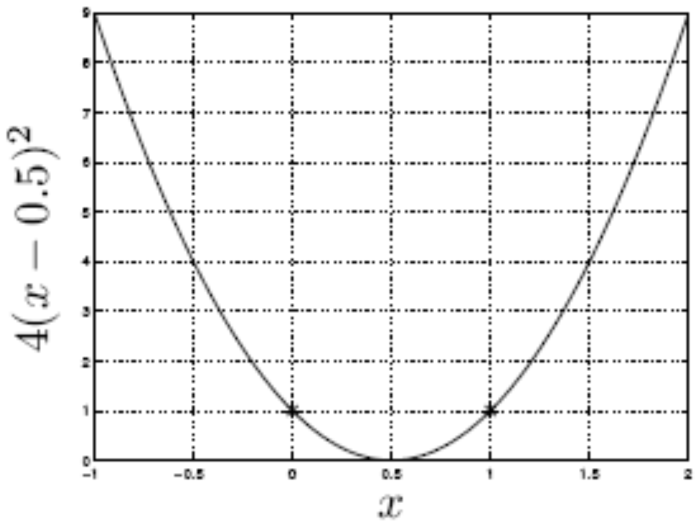
Therefore we can solve the original problem as follows:

Global orientation estimation

4. The global solution is <u>unique</u> with probability one

- <u>Proposition</u>: $S^5 = \arg\min_{\gamma \in \mathbb{Z}^\ell} \|\gamma - \hat{\gamma}\|^2_{P_\gamma^{-1}}$ admits a unique solution with probability one.

- <u>Proposition</u>: for any solution of the integer problem there exists a unique solution of the original maximum likelihood problem.



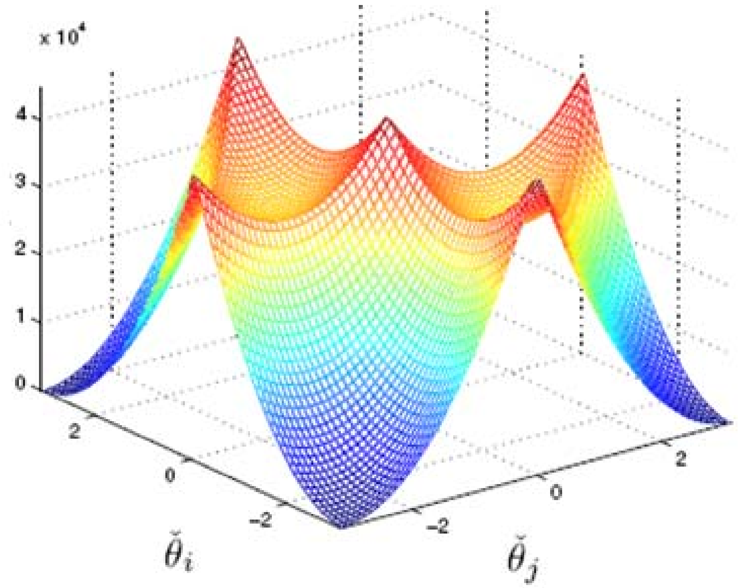
- <u>Corollary</u>: maximum likelihood orientation estimation admits a unique solution with probability one.

Global orientation estimation

- We solved the problem of local minima, but what about computational complexity?

$$S^5 = \arg\min_{\gamma \in \mathbb{Z}^\ell} \|\gamma - \hat{\gamma}\|^2_{\boldsymbol{P}_\gamma^{-1}}$$

- Quadratic integer programming is hard to solve if the dimension of $\boldsymbol{\gamma}$ is large
- We propose a polynomial time algorithm that returns a set Γ of plausible solutions of the problem
  - ***MOLE2D*** (Multi-hypothesis Orientation-from-Lattice Estimation)

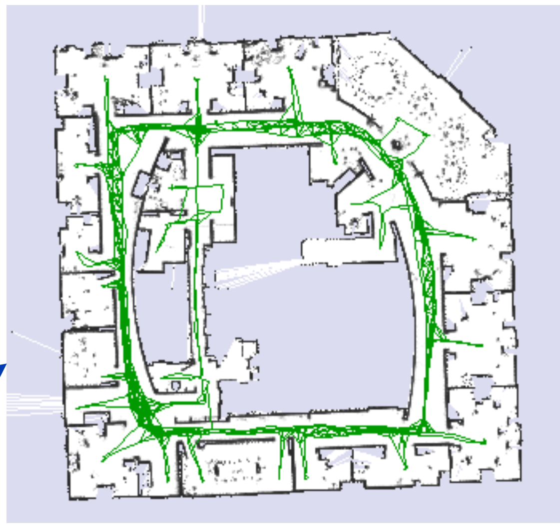


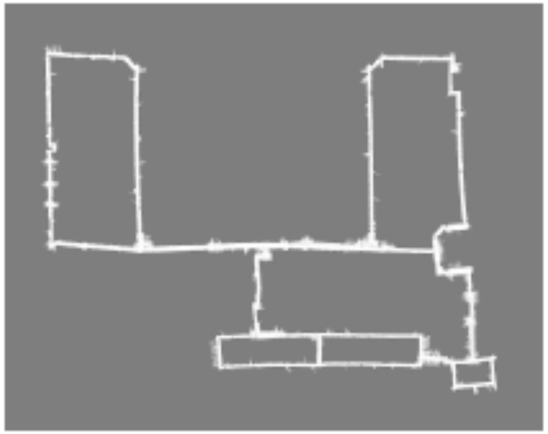
Trick: we have a probabilistic description of the measurements, hence we can perform inference on the admissible values of $\boldsymbol{\gamma}$

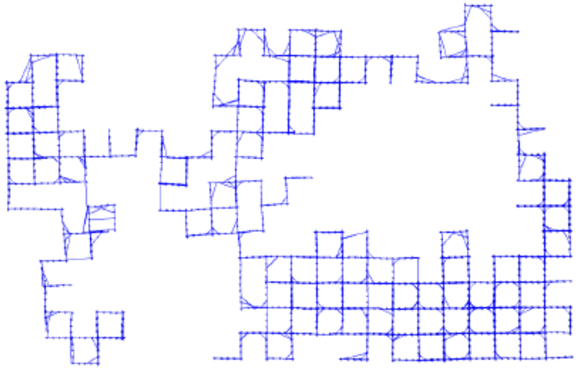
- Proposition: choosing the minimum cycle basis matrix minimizes the number of admissible $\boldsymbol{\gamma}$ in the algorithm MOLE2D.

Results, impact, and implementation

- Tests on standard datasets and different choices of the cycle basis matrix
  - INTEL (1228 nodes, 1505 edges)
  - MITb (808 nodes, 828 edges)
  - M3500 (3500 nodes, 5598 edges)
  - M3500a = M3500 + noise (std = 0.1rad)
  - M3500b = M3500 + noise (std = 0.2rad)
  - M3500c = M3500 + noise (std = 0.3rad)







Results, impact, and implementation

- MOLE2D returns a small number of hypotheses for γ

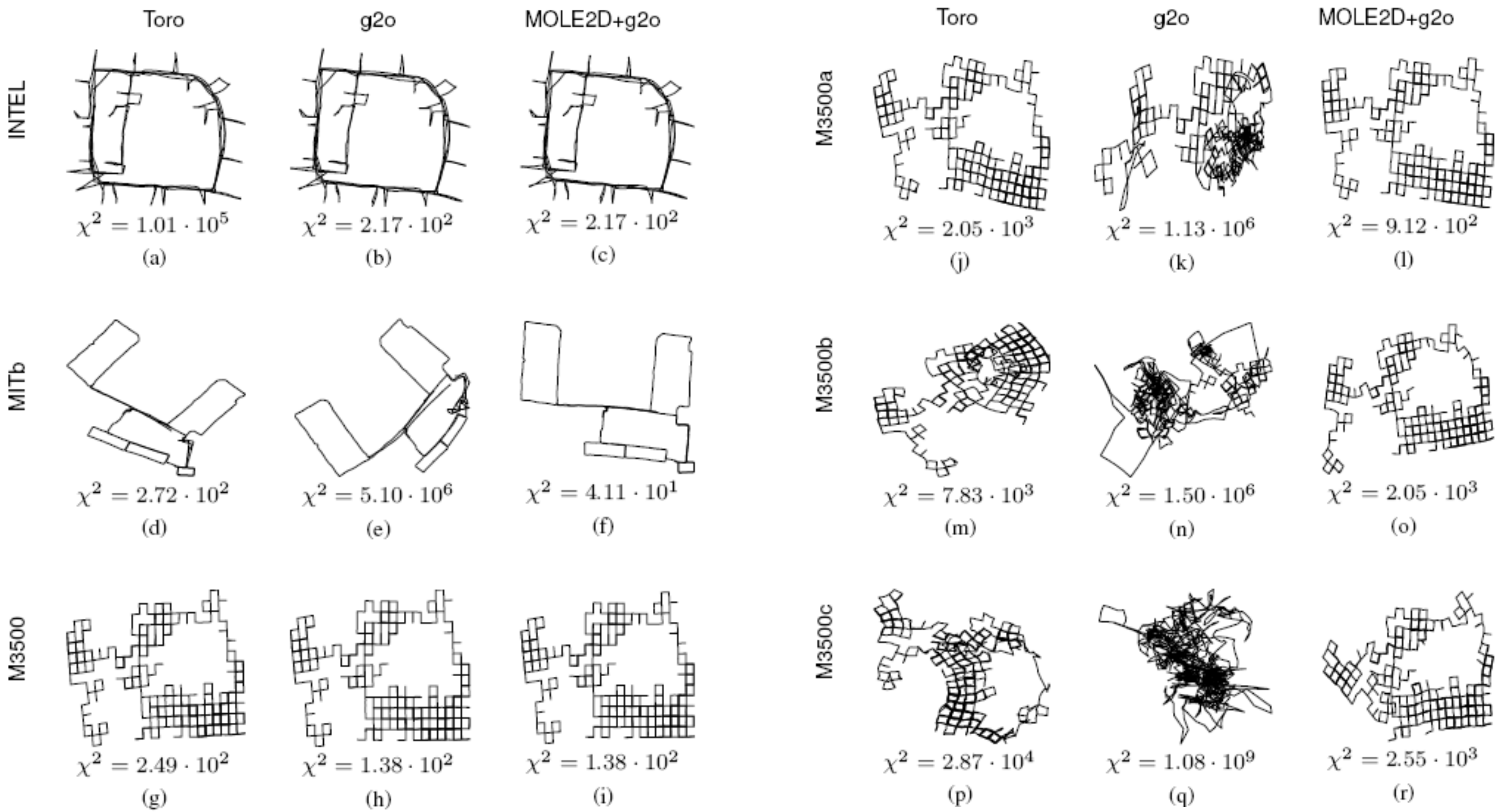
	cycle basis	K		u (%)	d (%)	$\lvert\Gamma\rvert$
INTEL	FCB_o	1		100.00	n/a	1
	FCB_m	1		100.00	n/a	1
	MCB_a	1		100.00	n/a	1
	MCB	1		100.00	n/a	1
MIT	FCB_o	2	iter. 1	80.00	25.78	16
			iter. 2	20.00	n/a	1
	FCB_m	1		100.00	n/a	1
	MCB_a	1		100.00	n/a	1
	MCB	1		100.00	n/a	1
M3500	FCB_o	5	iter. 1	52.92	1.69	$>10^{100}$
			iter. 2	21.54	15.61	$>10^{100}$
			iter. 3	20.06	100.00	$>10^{50}$
			iter. 4	5.12	100.00	972
			iter. 5	0.36	n/a	1
	FCB_m	2	iter. 1	98.62	2.41	$>10^{9}$
			iter. 2	1.38	n/a	1
	MCB_a	2	iter. 1	99.95	0.48	3
			iter. 2	0.05	n/a	1
	MCB	2	iter. 1	99.95	0.44	3
			iter. 2	0.05	n/a	1
M3500a	FCB_o	6		–	–	$>10^{30}$
	FCB_m	3		–	–	8
	MCB_a	2		–	–	1
	MCB	2		–	–	1
M3500b	FCB_o	29		–	–	$>10^{40}$
	FCB_m	4		–	–	27
	MCB_a	3		–	–	3
	MCB	3		–	–	3
M3500c	FCB_o	9		–	–	$>10^{100}$
	FCB_m	7		–	–	$>10^{4}$
	MCB_a	4		–	–	16
	MCB	4		–	–	16

- The algorithm is very efficient in practice (MATLAB implementation)

	phase	Computation of $\hat{\gamma}$ and P_γ	INTEGER-SCREENING	Computation of Θ from Γ	Total
	lines	11–12	13	14–16	
INTEL	FCB_o	0.07	0.04	$\leq$0.01	0.10
	FCB_m	$\leq$0.01	0.03	$\leq$0.01	0.04
	MCB_a	$\leq$0.01	0.05	$\leq$0.01	0.05
	MCB	$\leq$0.01	0.04	$\leq$0.01	0.04
MIT	FCB_o	$\leq$0.01	0.04	$\leq$0.01	0.04
	FCB_m	$\leq$0.01	0.03	$\leq$0.01	0.03
	MCB_a	$\leq$0.01	0.03	$\leq$0.01	0.03
	MCB	$\leq$0.01	0.03	$\leq$0.01	0.03
M3500	FCB_o	0.72	0.79	$\leq$0.01	1.52
	FCB_m	$\leq$0.01	0.47	$\leq$0.01	0.47
	MCB_a	$\leq$0.01	0.21	$\leq$0.01	0.22
	MCB	$\leq$0.01	0.21	$\leq$0.01	0.22
M3500a	FCB_o	0.72	0.80	(Γ too large to	continue)
	FCB_m	$\leq$0.01	0.36	0.04	0.4
	MCB_a	$\leq$0.01	0.21	$\leq$0.01	0.22
	MCB	$\leq$0.01	0.21	$\leq$0.01	0.22
M3500b	FCB_o	0.71	1.19	(Γ too large to	continue)
	FCB_m	$\leq$0.01	0.51	0.12	0.64
	MCB_a	$\leq$0.01	0.23	0.03	0.26
	MCB	$\leq$0.01	0.23	0.03	0.26
M3500c	FCB_o	0.72	0.72	(Γ too large to	continue)
	FCB_m	$\leq$0.01	0.48	(Γ too large to	continue)
	MCB_a	$\leq$0.01	0.23	0.15	0.38
	MCB	$\leq$0.01	0.23	0.14	0.37

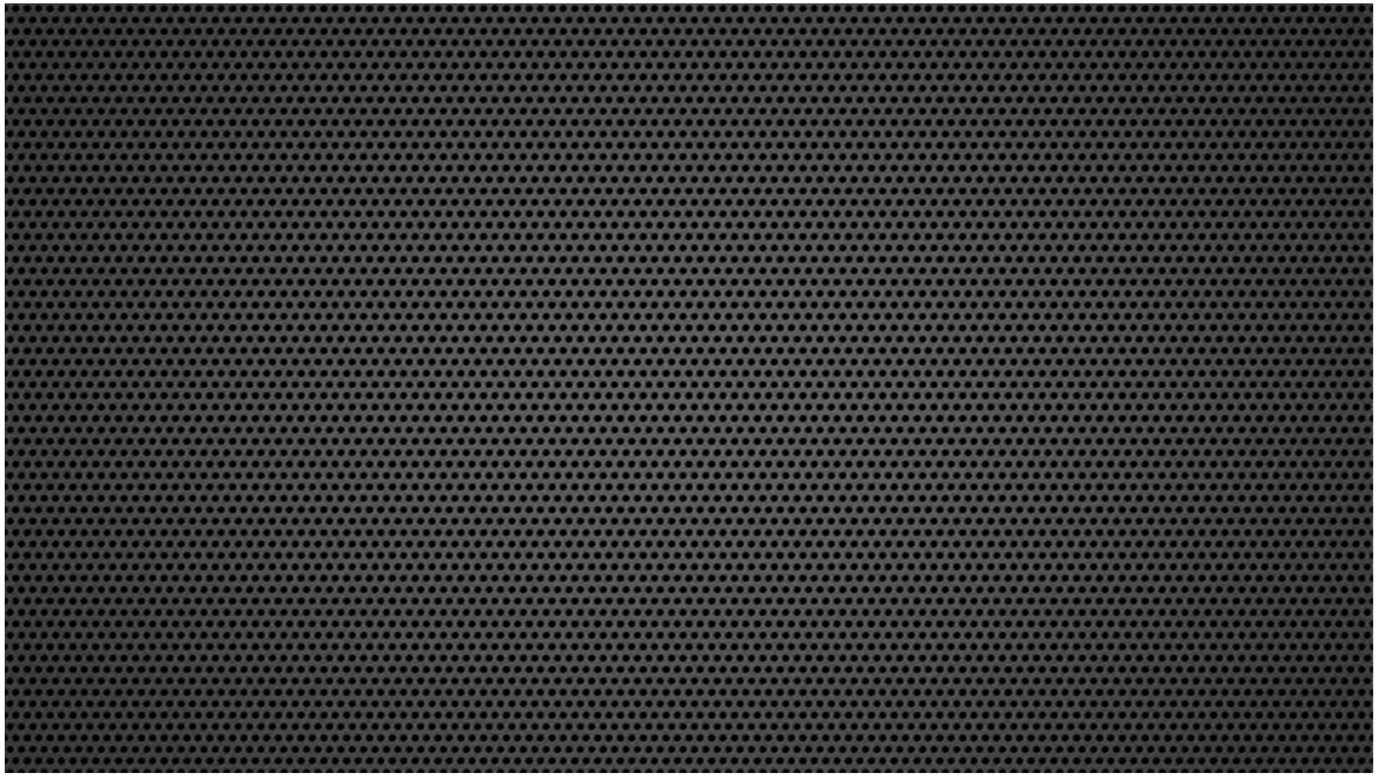
Results, impact, and implementation

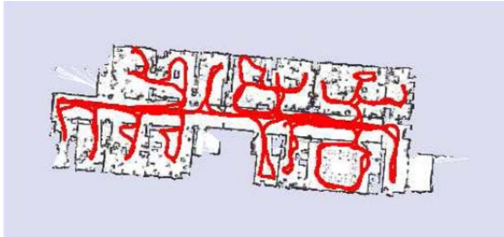
- Solving the orientation estimation problem we can obtain an accurate initial guess to be used in pose graph optimization

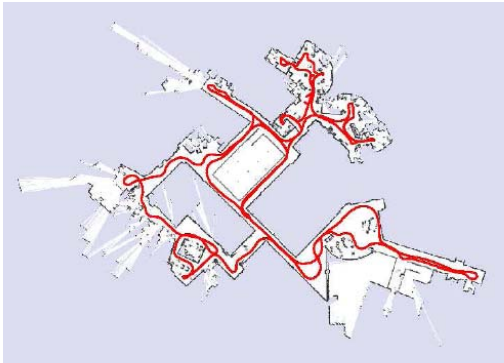


Results, impact, and implementation

- Real tests using a ROS implementation of the algorithm



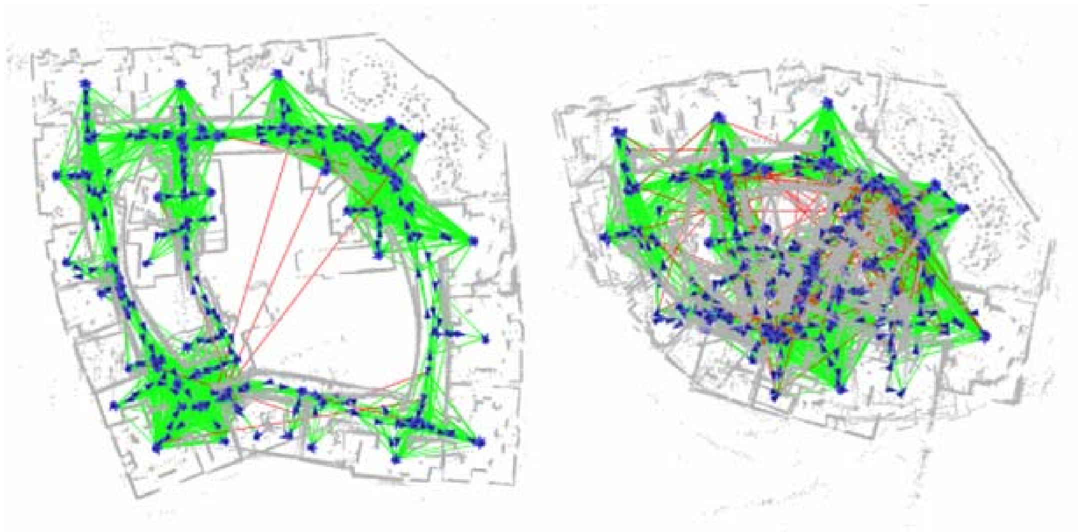




- MOLE2D computes a global estimate of robot orientations (no local minima):
  - we can quantify how far is the estimate from the actual robot orientations (uncertainty quantification)
  - we can to solve pose graph optimization with extreme levels of noise (robustness)
  - we have an accurate initial guess for pose graph optimization, hence iterative techniques require less iterations (efficiency)

Tricks:
- introduce integer variables to get rid of non-convexities
- exploit graph theoretical notions in the optimization
- exploit probabilistic description of measurements to speed-up computation

- Open issues:
  - In practical problems, several relative measurements are outliers
  - State-of-the-art techniques catastrophically fail in presence of few outliers



courtesy of E. Olson and P. Agarwal