VAN course
Lesson 11

Dr. Refael Vivanti
vivanti@gmail

Back to
some statistics

Canonical Parameterization

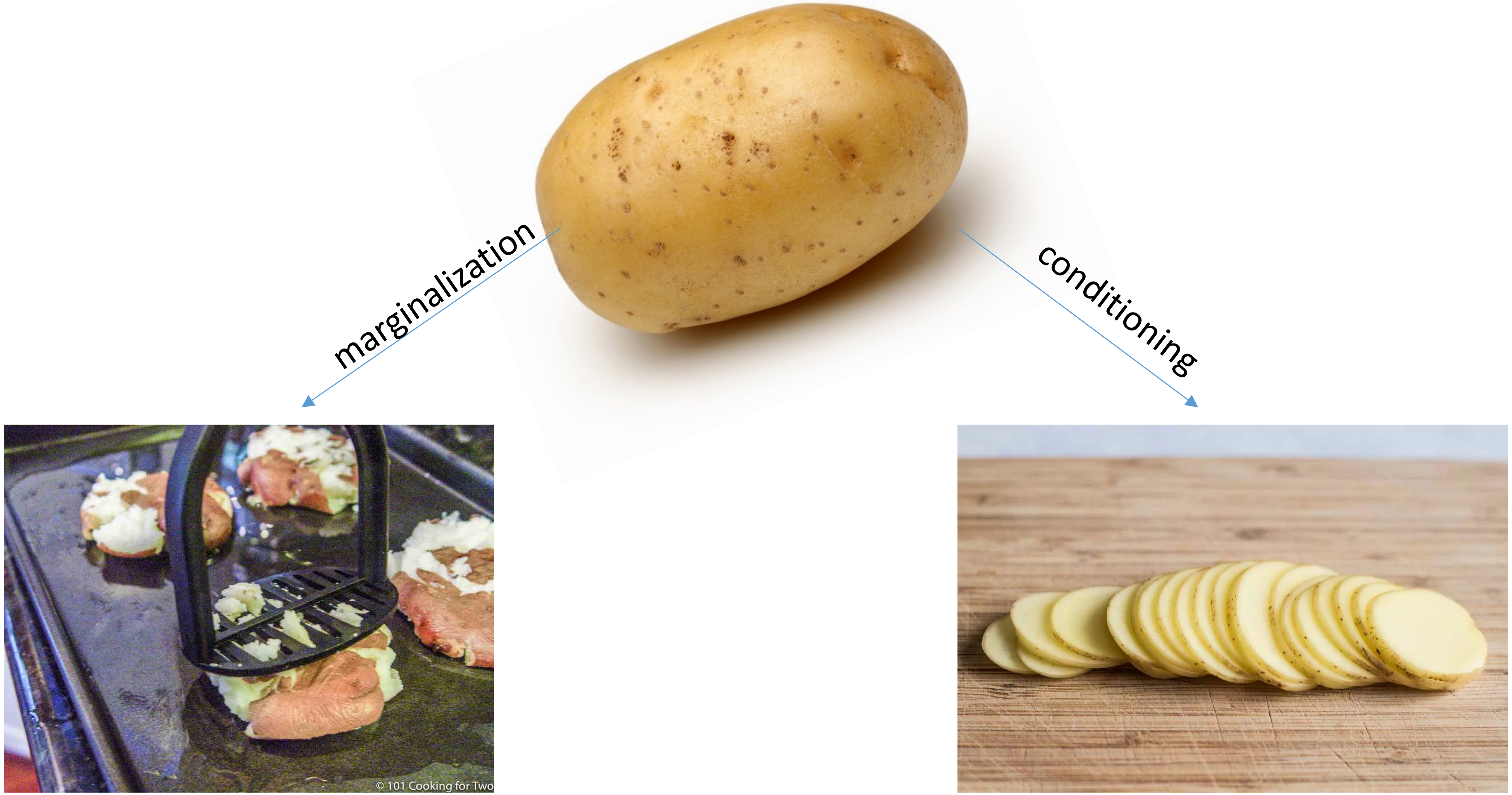
- Alternative representation for Gaussians
- Described by **information matrix** Ω

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

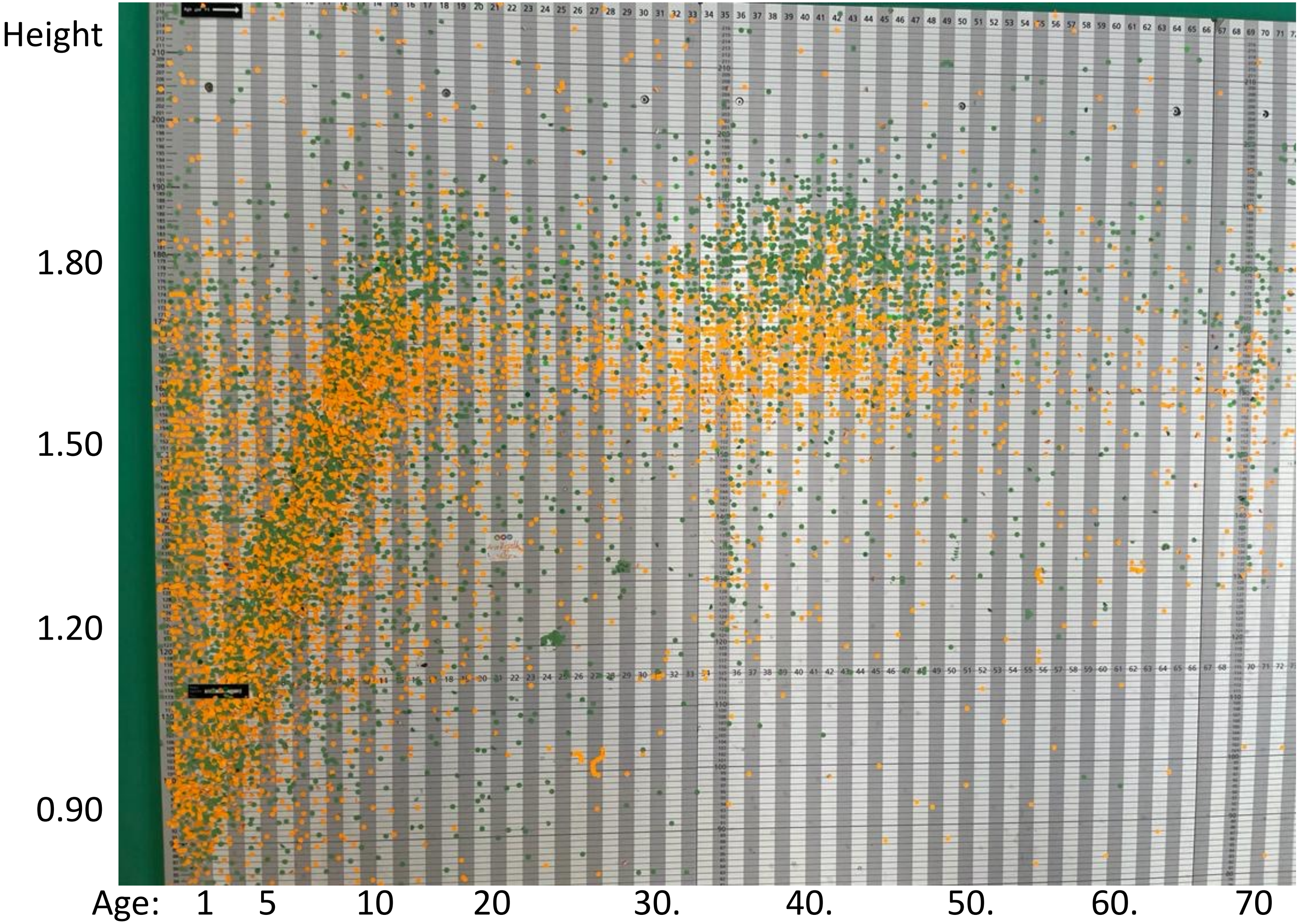
- and **information vector** ξ

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

Marginalization vs. conditioning



Real data from [Science Museum Jerusalem]():



Marginalization vs. conditioning

Example: the prob. of getting a 100:

student	Ex1	Ex2	Ex3	Ex4
Dudu	4/40	2/40	1/40	1/40
Maya	1/40	2/40	4/40	1/40
Yiftach	2/40	2/40	2/40	2/40
Arnon	5/40	1/40	1/40	1/40
Ehud	0/40	0/40	0/40	8/40

Marginalization:	$p(y) = \sum_x p(x,y)$	12/40	7/40	8/40	13/40
Conditioning:	$p(x) = \sum_x p(x \mid y = Dudu)$	4/8	2/8	1/8	1/8

Marginalization and Conditioning – how to

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

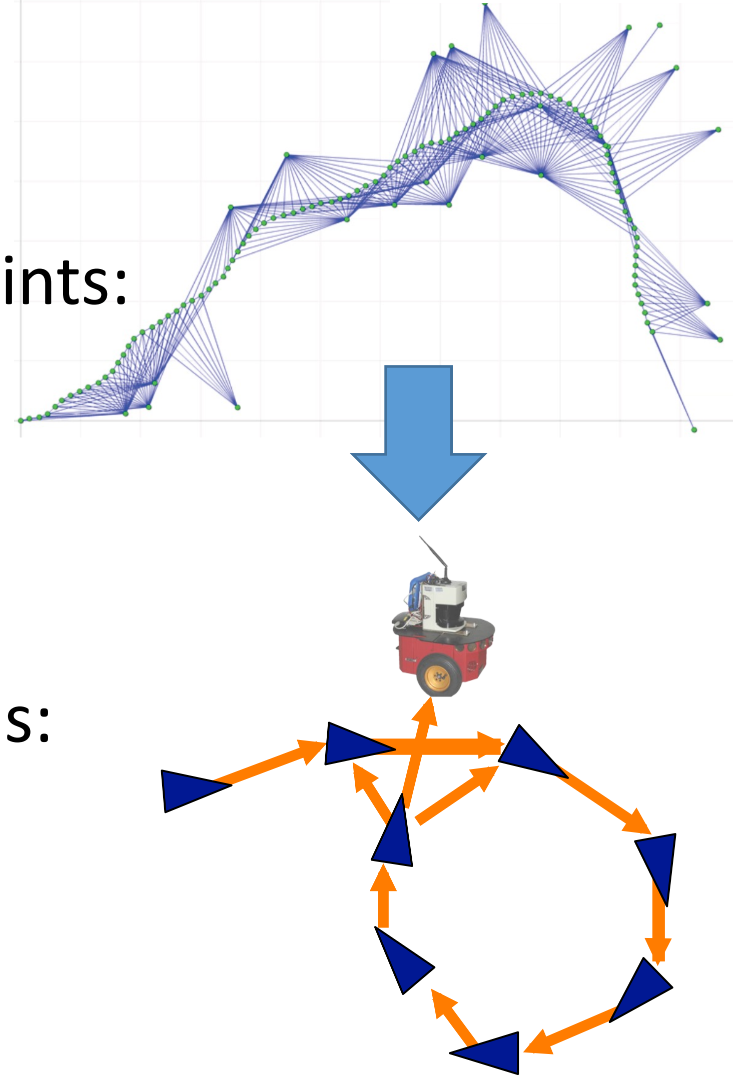
$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{bmatrix}, \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}\right) = \mathcal{N}^{-1}\left(\begin{bmatrix} \boldsymbol{\eta}_\alpha \\ \boldsymbol{\eta}_\beta \end{bmatrix}, \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}\right)$$

	MARGINALIZATION	CONDITIONING
	$p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha}, \boldsymbol{\beta})\, d\boldsymbol{\beta}$	$p(\boldsymbol{\alpha} \mid \boldsymbol{\beta}) = p(\boldsymbol{\alpha}, \boldsymbol{\beta}) / p(\boldsymbol{\beta})$
COV. FORM	$\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$	$\boldsymbol{\mu}' = \boldsymbol{\mu}_\alpha + \Sigma$ … $- \boldsymbol{\mu}_\beta)$ expensive $\Sigma' =$ … $_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}$
INFO. FORM	$\boldsymbol{\eta} = \boldsymbol{\eta}_\alpha - \Lambda$ … $_\beta$ expensive Λ … $\Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$	$\boldsymbol{\eta}' = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}$ $\Lambda' = \Lambda_{\alpha\alpha}$

Courtesy: R. Eustice

Pose Graph - compromises

Courtesy of Dellaert06ijrr: "Square Root SAM"

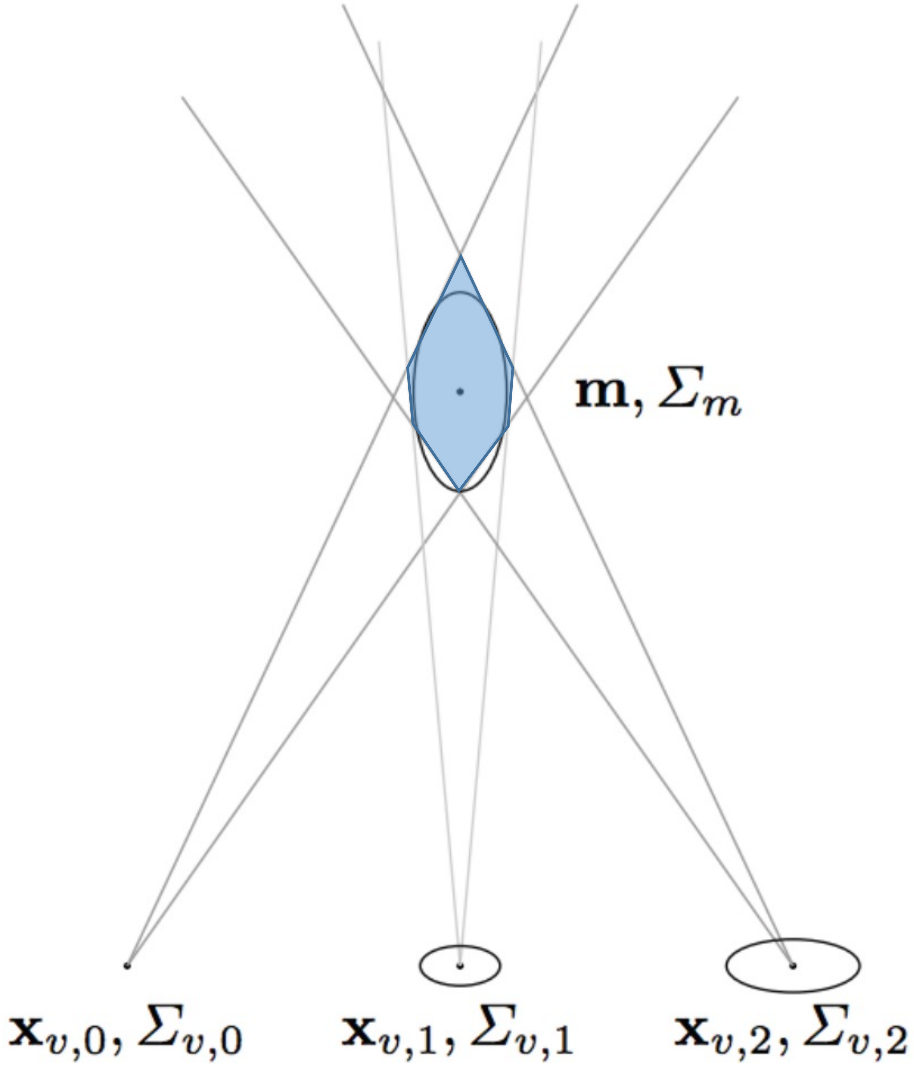


- We replaced our big factor graph with a pose graph
- How did we compromise?
- From each small factor graph of K cameras and P points:
  - We removed all points
  - We removed most cameras
  - $P(C_1,\ldots,C_K,p_1,..p_P) \underbrace{->}_{\text{marginalization}} P(C_1,C_K) \underbrace{->}_{\text{conditioning}} P(C_K|C_1)$
- In the full factor graph, point p might have F cameras:
  - We used this track by parts, in each small factor graph:
  - $P(C_1,\ldots,C_F,p_1) -> P(KF_2,p|KF_1) \cdot\cdot P(KF_N, p|KF_{N-1})$

Pose Graph - compromises

- Covariance approximation:
  - The small factor graph information is represented as a normal distribution
  - If all operations were linear, it’ll be ok.
  - But we do a non-linear projection
  - Using Gaussians is an **approximation**
    - Even if the original distributions were Gaussians.

- We ignored **a lot** of information
  - Surprisingly, the results are often accurate!



Pose Graph – how to

- To build and initialize the pose graph we need:
  - Each KF will be a vertex (symbol)
    - Initialization: global-coords pose of each KF camera.
  - Each successive KF pair, which participated in one small factor graph, will define a factor between two vertices:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{t2v}(\mathbf{Z}_{ij}^{-1}(\mathbf{X}_i^{-1}\mathbf{X}_j))$$

  - It also needs a covariance..

Pose Graph – how to

- How to calculate the covariance:

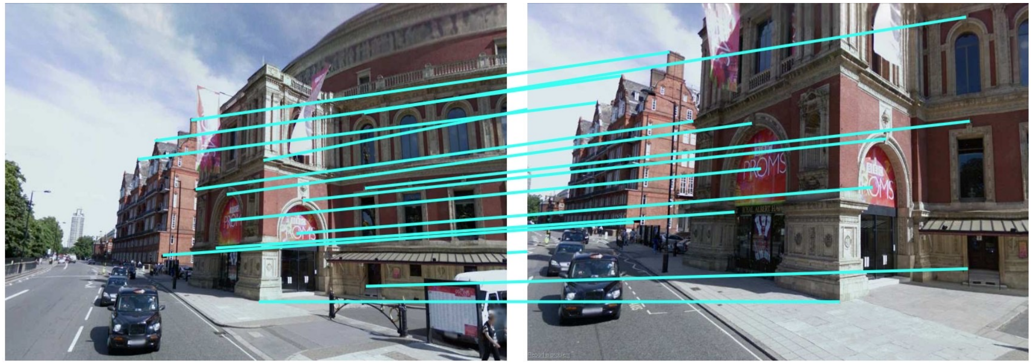
$$\Sigma_{all} \xrightarrow{marg.} \Sigma_{1N} \xrightarrow{inv.} \Omega_{1,N} \xrightarrow{cond.} \Omega_{N|1} \xrightarrow{inv.} \Sigma_{N|1}$$

$$\begin{array}{c} C_1 \\ \vdots \\ C_N \\ P_1 \\ \vdots \\ P_M \end{array} \begin{bmatrix} C_{11} & \cdots & C_{1N} & C_1P_1 & \cdots & C_1P_M \\ \vdots & \ddots & & & & \\ C_{N1} & & C_{NN} & C_NP_1 & & C_NP_M \\ C_1P_1 & & C_NP_1 & P_{11} & & P_{1M} \\ & & & & \ddots & \\ C_1P_M & & C_NP_M & P_{M1} & & P_{MM} \end{bmatrix} \rightarrow \begin{bmatrix} C_{11} & C_{1N} \\ C_{N1} & C_{NN} \end{bmatrix} \rightarrow \begin{bmatrix} I_{11} & I_{1N} \\ I_{N1} & I_{NN} \end{bmatrix} \rightarrow \begin{bmatrix} I_{NN} \end{bmatrix} \rightarrow \begin{bmatrix} C_{NN} \end{bmatrix}$$

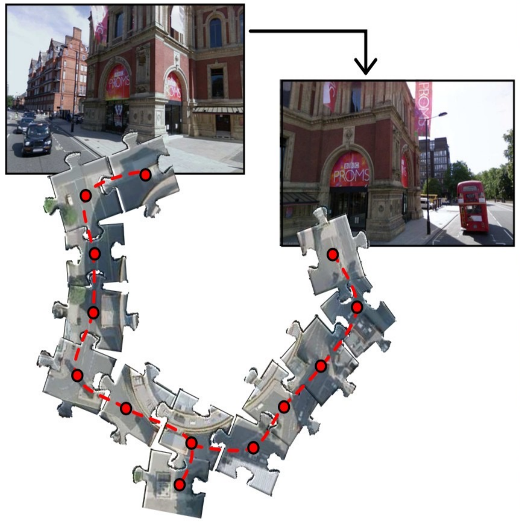
Loop Closure

Loop Closure

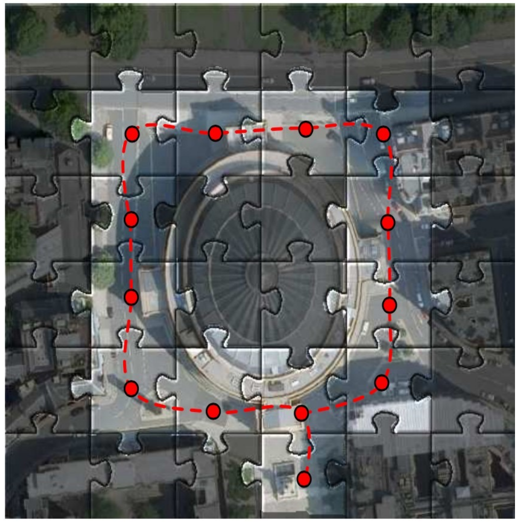
- Problem: Navigation drifts
  - We can reduce it, but not eliminate the problem
- If we revisit a location, it can help
  - Shorter path to origin in the pose graph
  - Constraints propagate to other vertices
  - Vertices get a ‘second opinion’
- This is called a “**Loop Closure**”



(a) Robust local motion estimation

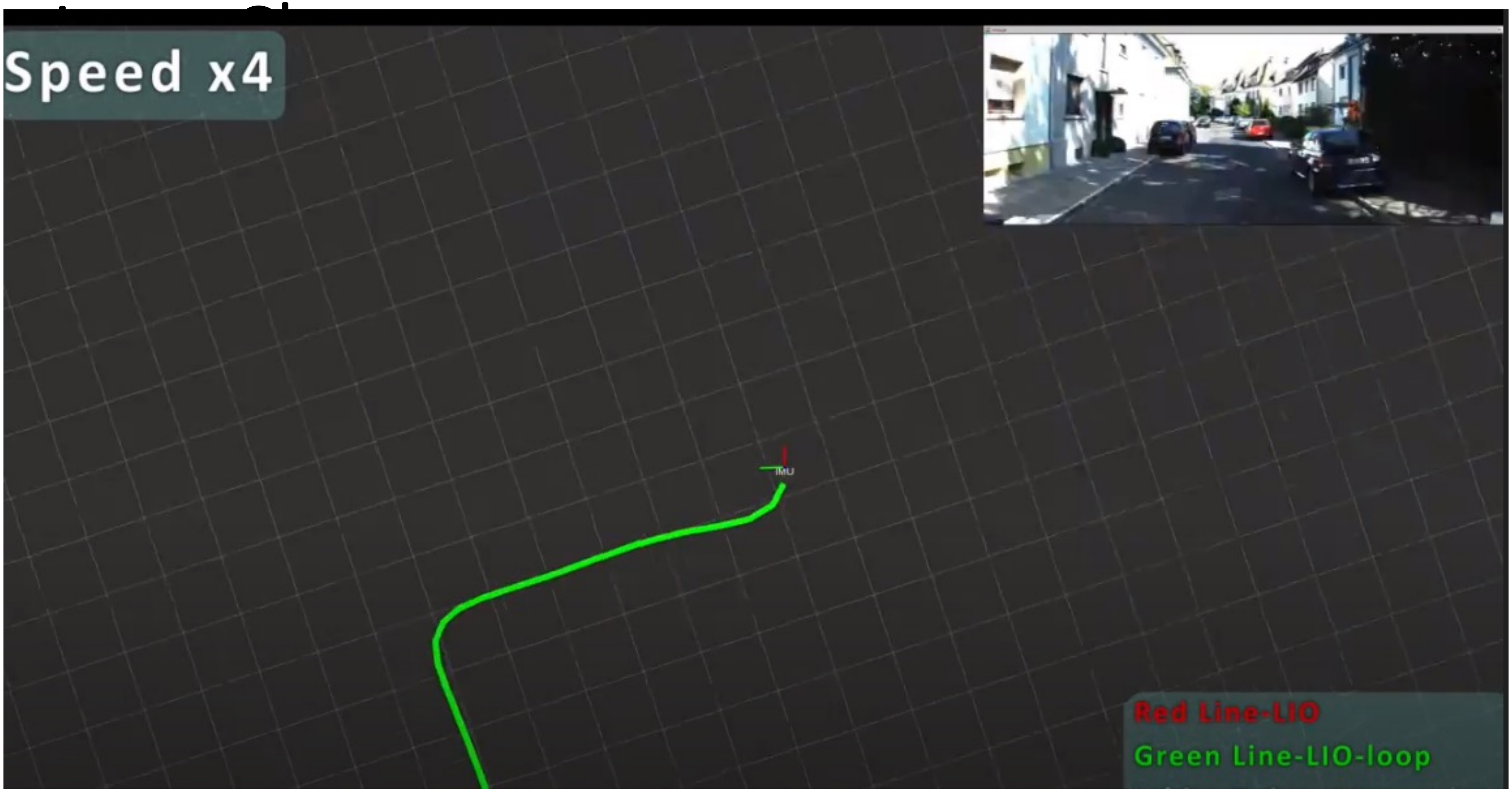


(b) Mapping and loop-closure detection



(c) Global optimisation

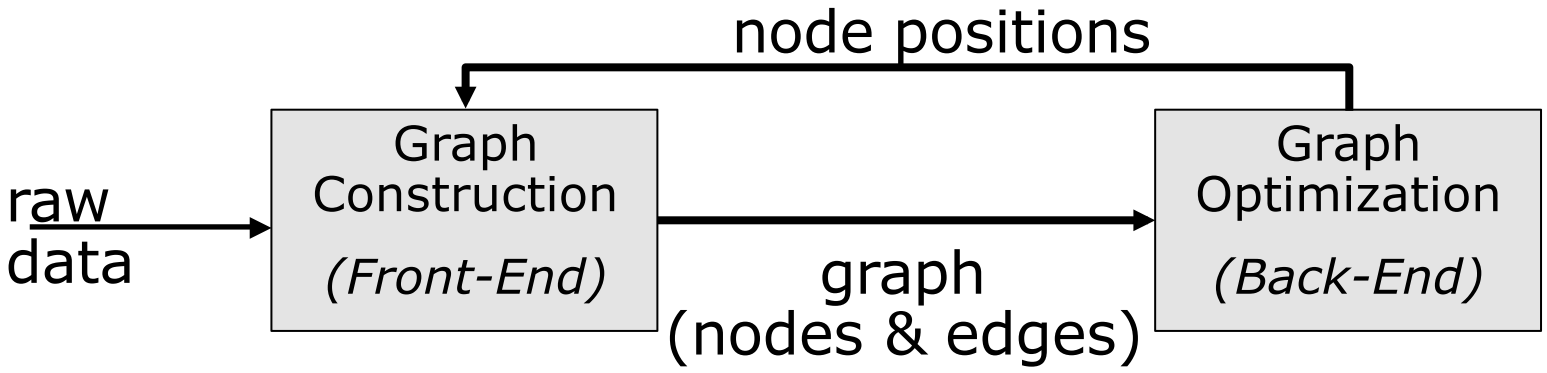
Images from: Chli09thesis – “Applying Information Theory to Efficient SLAM”, 2009



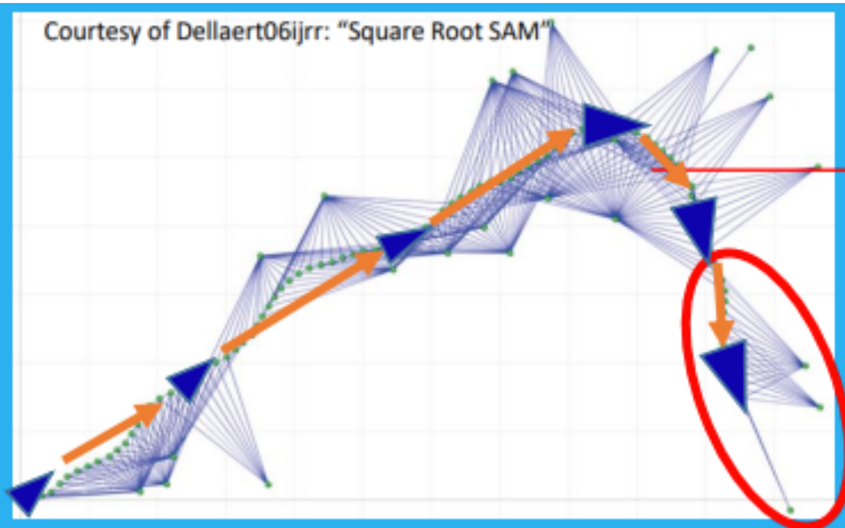
https://www.youtube.com/watch?v=LVbzuyOCCaM

Loop Closure

- When implementing an online system:
  - Front end finds new graph edges
    - To previous vertex – Stereo tracking, PnP, factor graph
    - To old known vertices – Loop closure
  - Backend
    - Global optimization – Pose graph



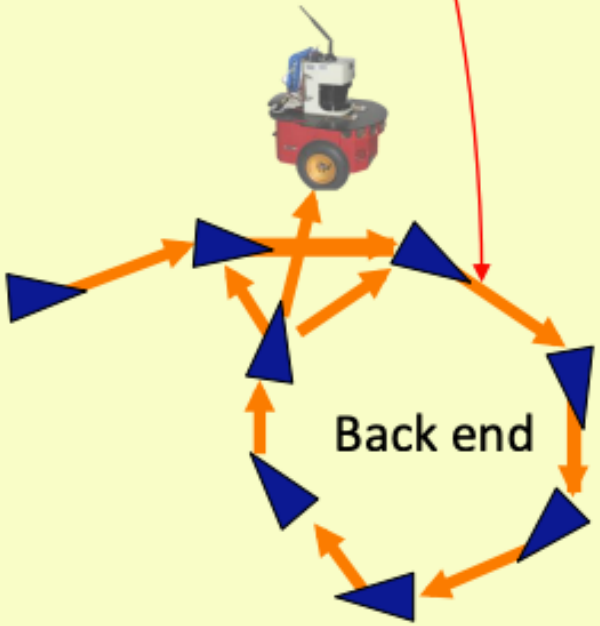
Courtesy of Cyrill Stachniss



We wish to solve the whole bundle

Front end

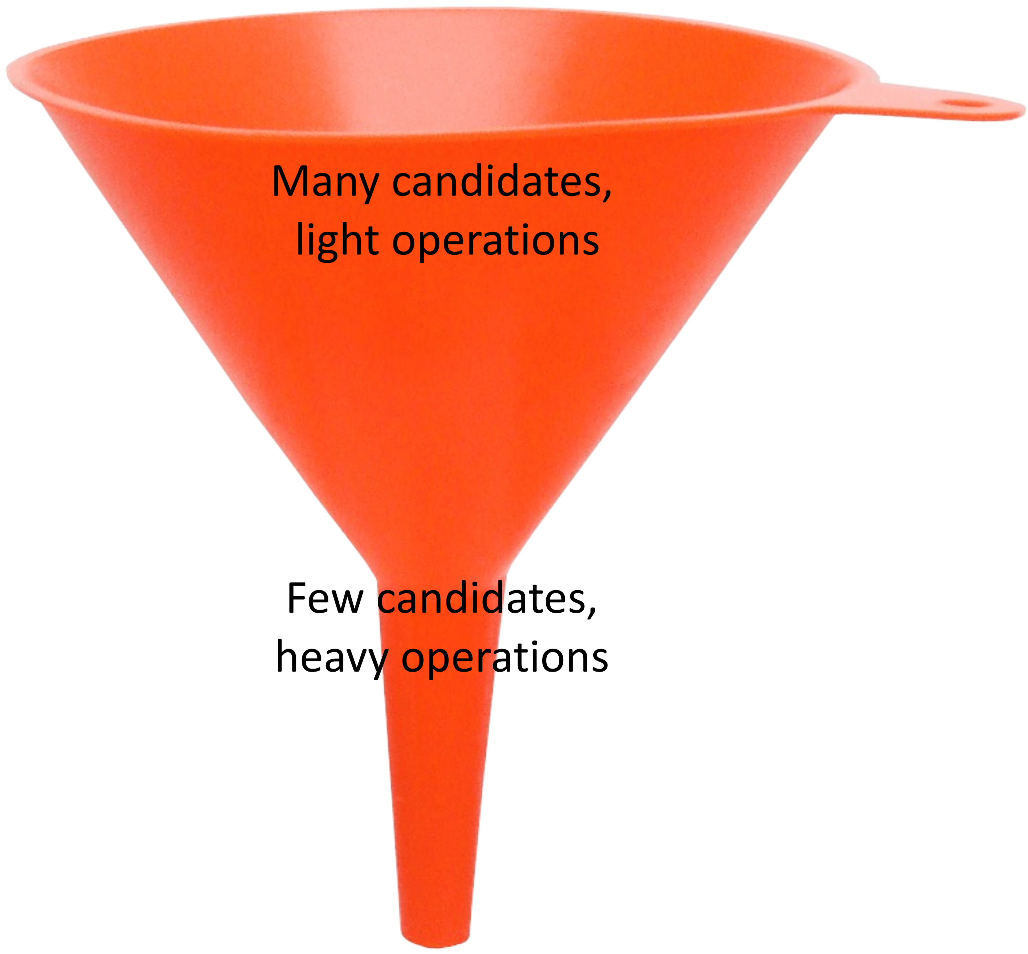
We settle for those
sub problems



And it works!

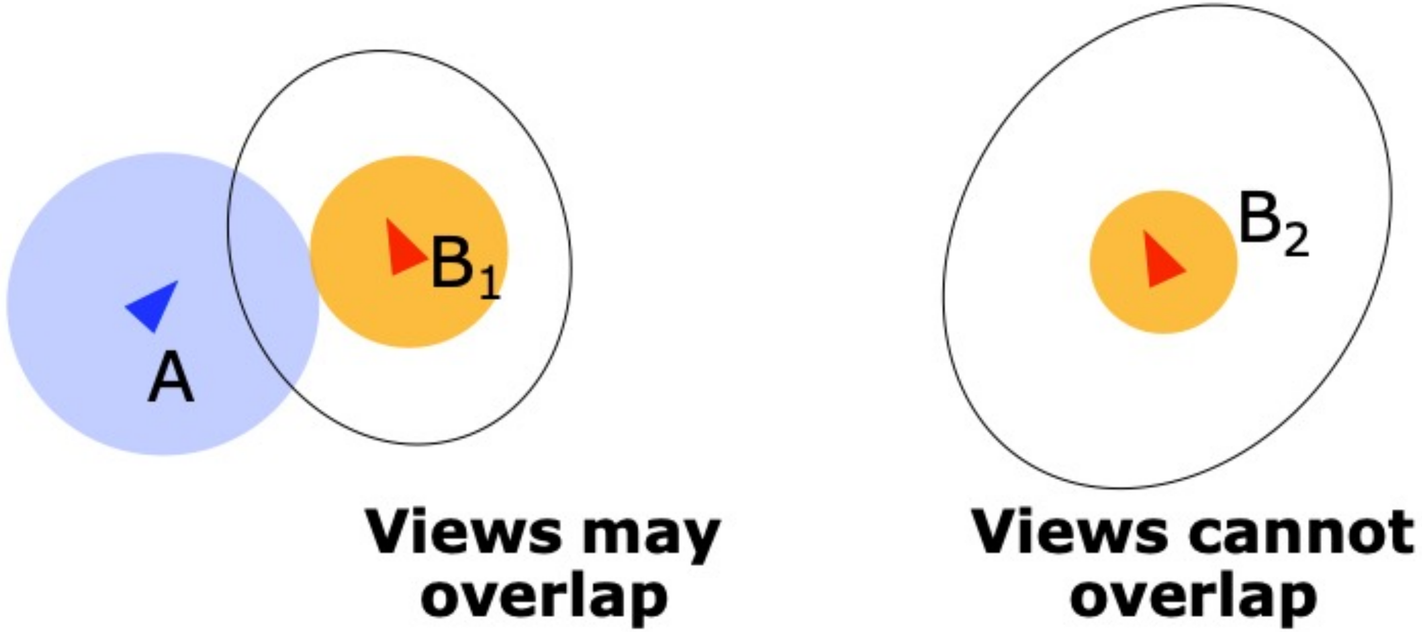
Loop Closure

- How can we spot a loop closure?
  - Find candidates (light):
    - Geometric intersection
  - Validate candidates (heavy):
    - 3D points clouds matching using ICP
    - Visual descriptor-based matching
  - Calculate edges and factors:
    - Find transformation using matches/ICP
  - Outlier removal:
    - Olson's method



Loop Closure - Geometric intersection

Geometric intersection: Where to Search for Matches?



- “Intersection” means that B pose **is** pose A with high probability
- Note: even if location overlaps, pose may not.

Loop Closure - Geometric intersection

- We wish to find: $\Delta x^T \Omega_{n|i} \Delta x < d$

  Where: $\Delta x = t2v(X_i^{-1} X_n)$

  and $\Omega_{n|i}$ is the conditional information matrix of $x_n | x_i$
- This requires marginalization to remove all other x_j
- Inverting the full information matrix is too expensive for front-end.
- Fast approximation:
  - Find shortest path using Dijkstra
  - Conservative Cov estimation:
  - Compose the incremental covariances along the path.

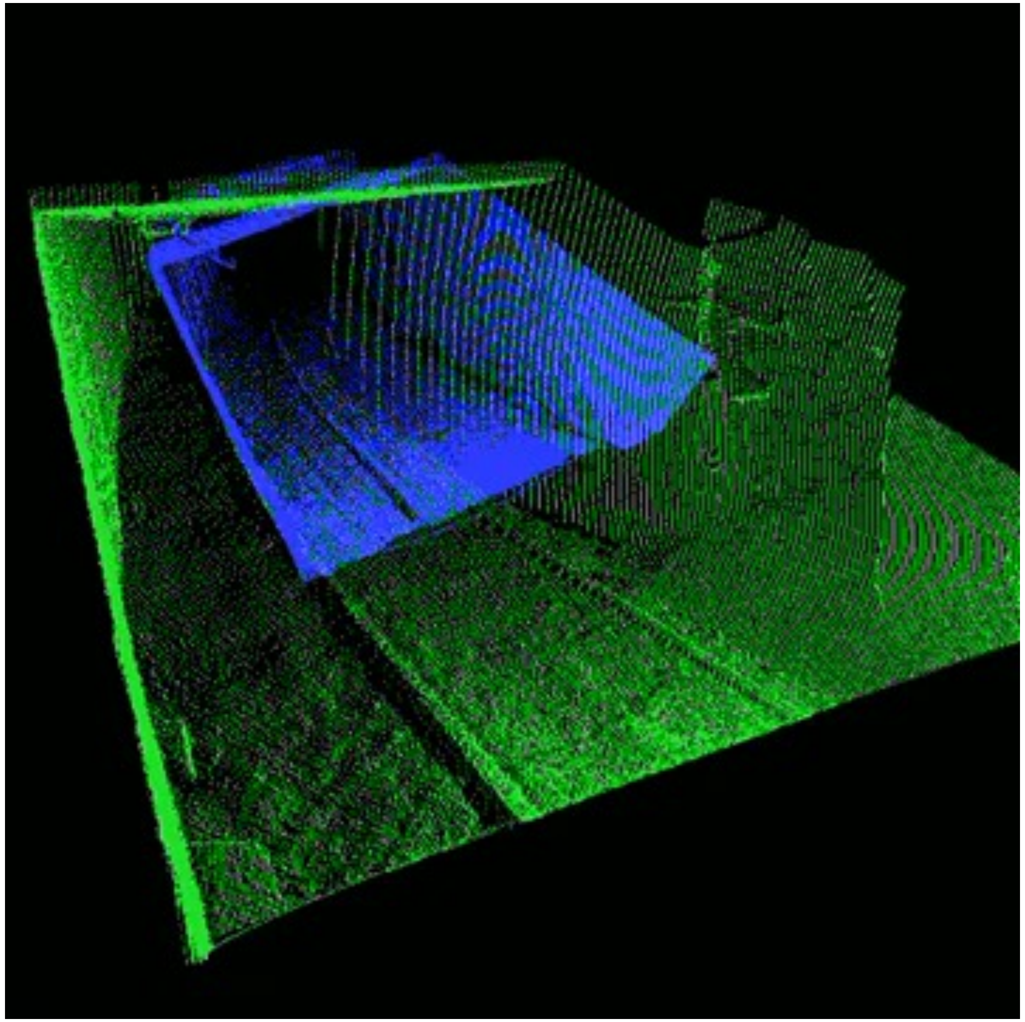
Assume $\mathbf{x} \sim \mathcal{N}(\mathbf{m}_x, \mathbf{\Sigma}_x)$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{m}_y, \mathbf{\Sigma}_y)$ then

$$\mathbf{Ax} + \mathbf{By} + \mathbf{c} \sim \mathcal{N}(\mathbf{Am}_x + \mathbf{Bm}_y + \mathbf{c}, \mathbf{A\Sigma}_x\mathbf{A}^T + \mathbf{B\Sigma}_y\mathbf{B}^T)$$

$$x + y \sim N(m_x + m_y, \Sigma_x + \Sigma_y)$$

Loop Closure - Validate candidates with ICP

- For any candidate-pair for edge:
  - Find the corresponding 3D point cloud
  - Optional: extract unique structures
    - Like trees of cars
    - Walls are large but with low information
  - Find transformation
    - Using ICP, RANSAC and least squares minimization
  - Evaluate edge
    - Matches percent
    - Mean distance
  - If it's good, set an edge
    - The factor is the calculated relative transformation



Courtesy of [Andreas Nuechter]()

Loop Closure - Validate candidates with ICP

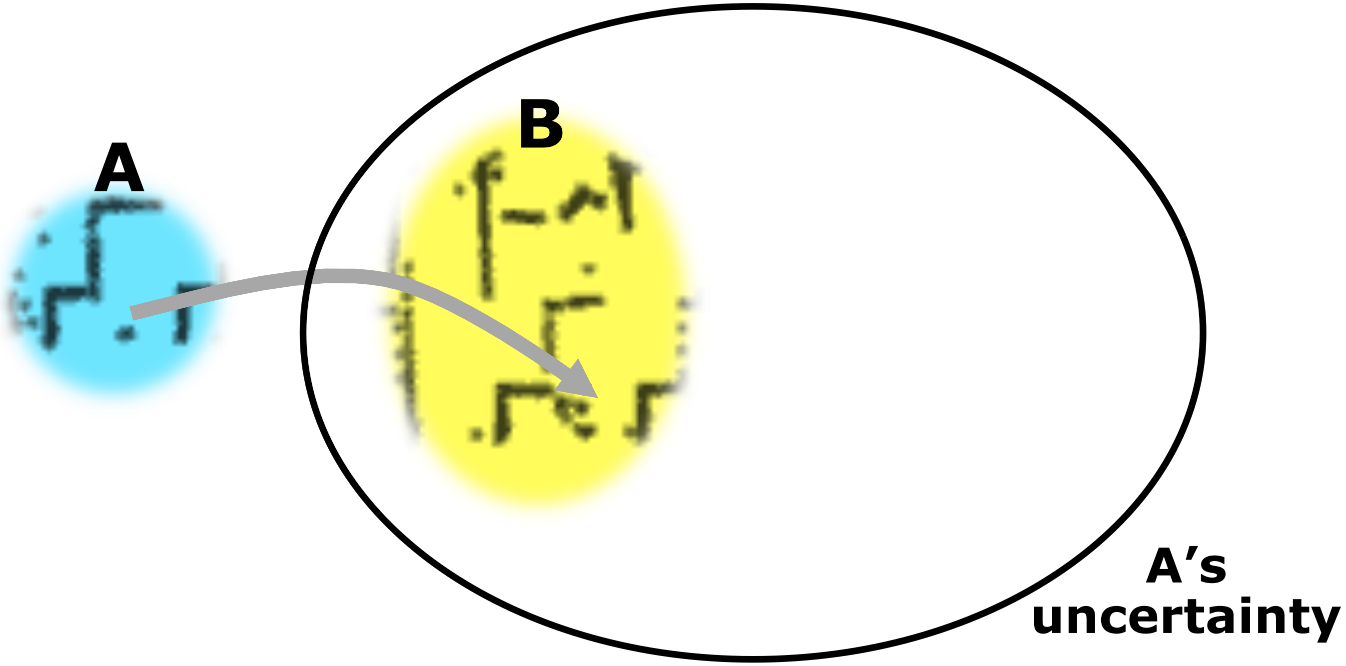
Problems

- ICP is sensitive to the initial guess
- Make many initial guesses? Inefficient sampling
- Ambiguities in the environment

Courtesy of Cyrill Stachniss

Loop Closure - Validate candidates with ICP
Ambiguities - Global Ambiguity

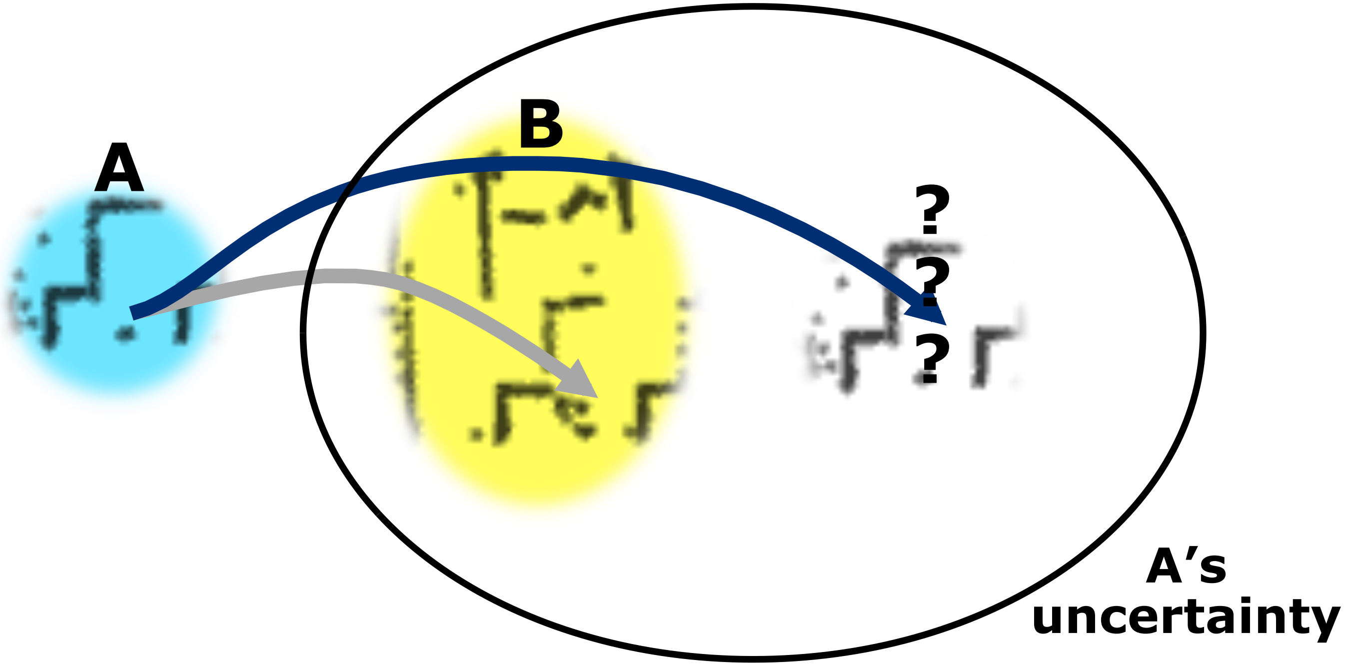
- B is inside the uncertainty ellipse of A
- Are A and B the same place?



Courtesy of Cyrill Stachniss

Loop Closure - Validate candidates with ICP
Ambiguities - Global Ambiguity

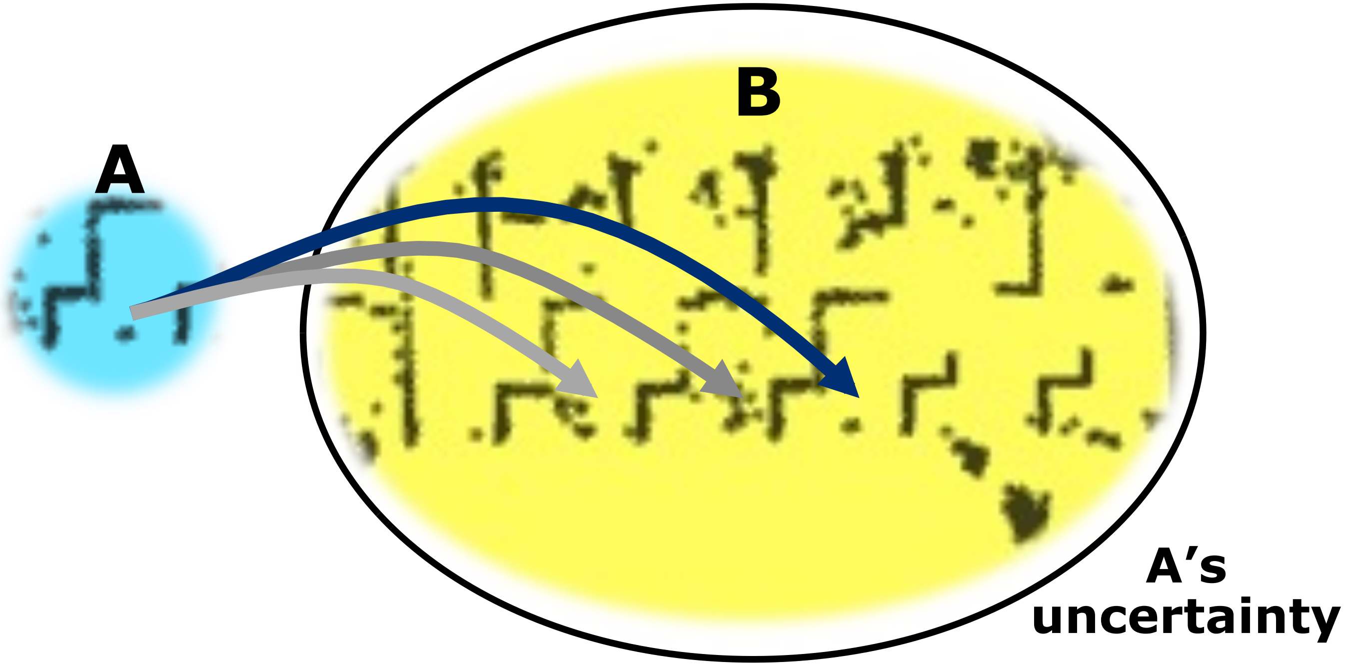
- B is inside the uncertainty ellipse of A
- A and B might not be the same place



Courtesy of Cyrill Stachniss

Loop Closure - Validate candidates with ICP
Ambiguities - Global Ambiguity

- B is inside the uncertainty ellipse of A
- A and B are not the same place

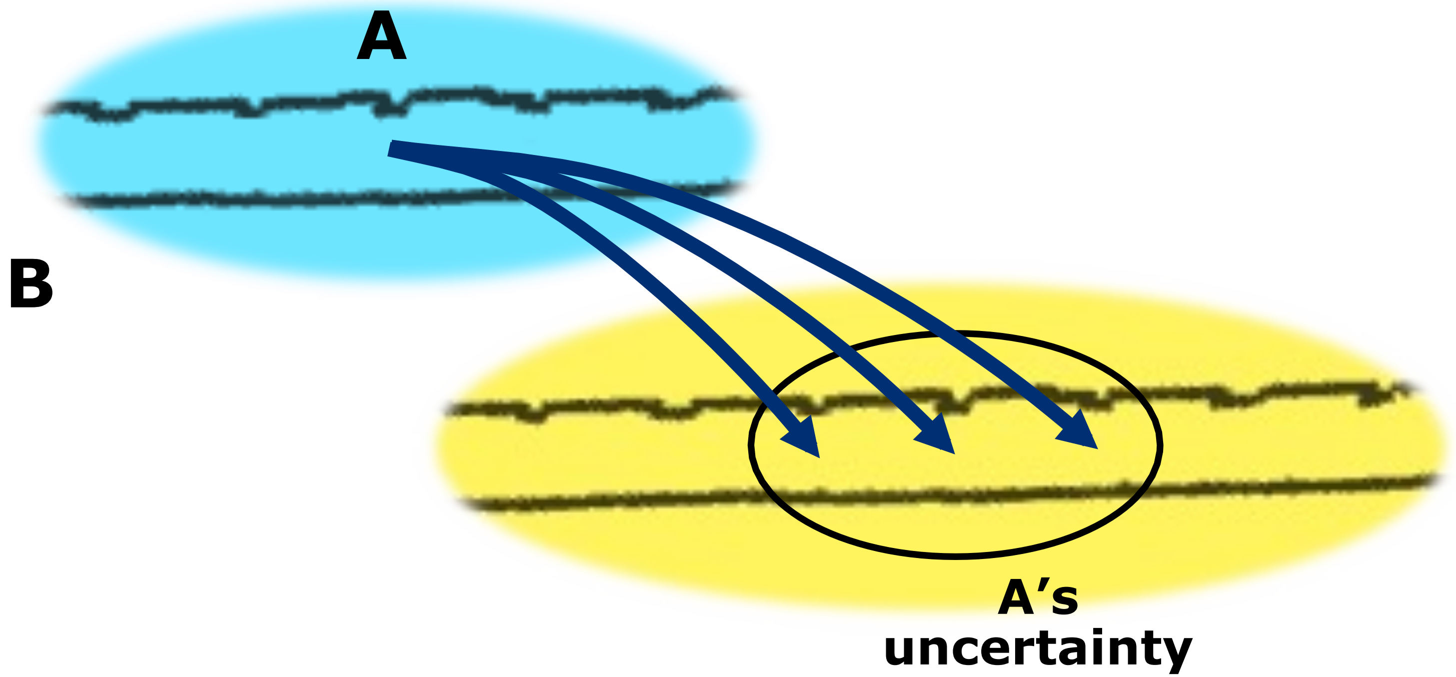


Courtesy of Cyrill Stachniss

Loop Closure - Validate candidates with ICP

Ambiguities - Local Ambiguity

- "Picket Fence Problem": largely overlapping local matches



Courtesy of Cyrill Stachniss

Loop Closure - Validate with descriptors

- For any candidate-pair for edge:
  - Extract features descriptors from both images
  - Find correspondences
    - Using ANN
  - Remove outliers, evaluate edge
    - With RANSAC and Fundamental Matrix
  - If it's good, set an edge
    - Calculated relative transformation
    - First with PnP
    - Then with small factor graph for the Cov matrix
  - Much more robust than point-cloud methods
    - Low ambiguity rate
    - Relative transformation may still be wrong