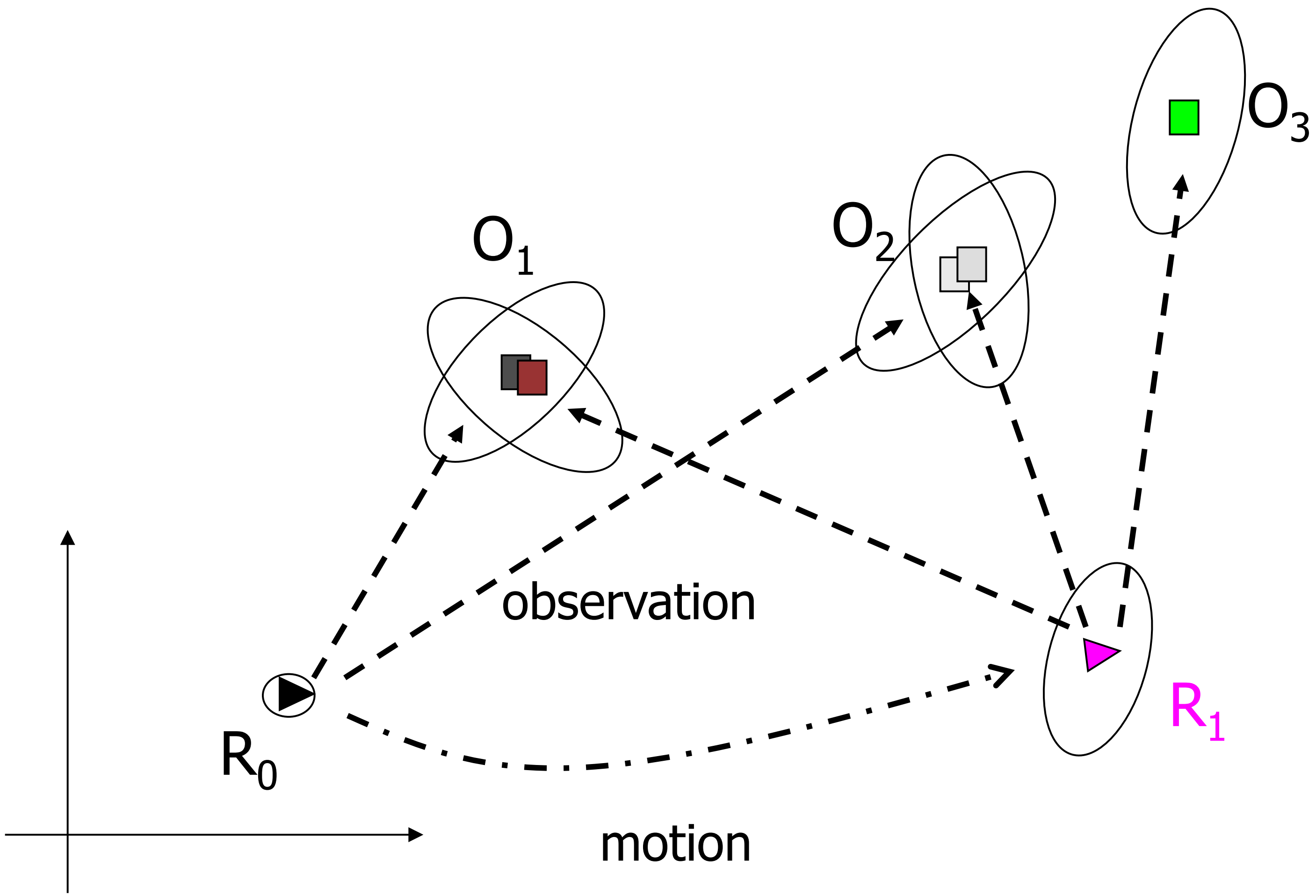
Data Association

Robot Localization and Mapping
16-833

Michael Kaess

November 20, 2024

Incremental Map Making



Courtesy J. Leonard

The Data Association Problem

- Geometry-based
  - Nearest Neighbor
  - Maximum Likelihood
- Consensus
  - Joint Compatibility
  - RANSAC
  - Consistency (over many time steps)
- Appearance
  - Descriptor
  - Bag-of-words
  - Ferns
  - Deep Learning
- Optimization
  - Non-Gaussian…

(Planar) EKF-SLAM Review

- State Vector

$$\mathbf{x}_t = [\mathbf{x}_r^\top, \ \mathbf{x}_{\ell_1}^\top, \ \ldots, \ \mathbf{x}_{\ell_n}^\top]^\top \longleftarrow \text{(3+2}n\text{) column vector}$$

- Belief

$$p(\mathbf{x}_t \,|\, \mathbf{c}_{1:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \sim \mathcal{N}(\mathbf{x}_t; \mu_t, \Sigma_t)$$

$$\mu_t = \begin{bmatrix} \mu_r \\ \mu_L \end{bmatrix} \qquad \Sigma_t = \begin{bmatrix} \Sigma_{rr} & \Sigma_{rL} \\ \Sigma_{Lr} & \Sigma_{LL} \end{bmatrix}$$

Courtesy R. Eustice

EKF-SLAM Process Model

- Robot Process Model

$$\mathbf{x}_{r_{t+1}} = \mathbf{f}(\mathbf{x}_{r_t}, \mathbf{u}_t, \varepsilon_t) \qquad \varepsilon_t \sim \mathcal{N}(0, R_t)$$

- (Stationary) Landmark Process Model

$$\mathbf{x}_{\ell_{i_{t+1}}} = \mathbf{x}_{\ell_{i_t}}$$

- Joint Process Model

$$\begin{aligned}
\mathbf{x}_{t+1} &= \begin{bmatrix} \mathbf{f}(\mathbf{x}_{r_t}, \mathbf{u}_t, \varepsilon_t) \\ \mathbf{x}_L \end{bmatrix} \\
&\approx \begin{bmatrix} \mathbf{f}(\mu_{r_t}, \mathbf{u}_t, 0) \\ \mu_L \end{bmatrix} + \begin{bmatrix} F_r & 0 \\ 0 & I \end{bmatrix} (\mathbf{x}_t - \mu_t) + \begin{bmatrix} F_\varepsilon \\ 0 \end{bmatrix} (\varepsilon_t - 0)
\end{aligned}$$

Courtesy R. Eustice

EKF-SLAM Prediction

- $bel(\mathbf{x}_t)$ $\qquad \mu_t = \begin{bmatrix} \mu_r \\ \mu_L \end{bmatrix}$

$$\Sigma_t = \begin{bmatrix} \Sigma_{rr} & \Sigma_{rL} \\ \Sigma_{Lr} & \Sigma_{LL} \end{bmatrix}$$

- $\overline{bel}(\mathbf{x}_{t+1})$

$$\bar{\mu}_{t+1} = \begin{bmatrix} \mathbf{f}(\mu_r, \mathbf{u}_t, 0) \\ \mu_L \end{bmatrix}$$

$$\bar{\Sigma}_{t+1} = \begin{bmatrix} F_r \Sigma_{rr} F_r^\top + F_\varepsilon R_t F_\varepsilon^\top & F_r \Sigma_{rL} \\ \Sigma_{Lr} F_r^\top & \Sigma_{LL} \end{bmatrix}$$

Courtesy R. Eustice

EKF-SLAM Observation Model

- Observation model for landmark i

$$\delta_t^i \sim \mathcal{N}(0, Q_t^i)$$

$$\begin{aligned} \mathbf{z}_t^i &= \mathbf{h}(\mathbf{x}_t, c_t^i) + \delta_t^i \\ &\approx \mathbf{h}(\bar{\mu}_t, c_t^i) + H_x^i(\mathbf{x}_t - \bar{\mu}_t) + \delta_t^i \end{aligned}$$

z is a k-dim column vector (e.g., k=2 for range and bearing)

- Sparse Jacobian (i.e., $\mathbf{h}(\mathbf{x}_t, c_t^i) \equiv \mathbf{h}(\mathbf{x}_r, \mathbf{x}_{\ell_i})$)

$$H_x^i = [H_r, 0, \ldots, H_{\ell_i}, \ldots, 0]$$

- Hence

$$\bar{\mu}_{z_t^i} = \mathbf{h}(\bar{\mu}_r, \bar{\mu}_{\ell_i})$$

$$\begin{aligned} S_t^i &= H_x^i \bar{\Sigma}_t H_x^{i\top} + Q_t^i \\ &= H_r \bar{\Sigma}_{rr} H_r^\top + H_r \bar{\Sigma}_{r\ell_i} H_{\ell_i}^\top + H_{\ell_i} \bar{\Sigma}_{\ell_i r} H_r^\top + H_{\ell_i} \bar{\Sigma}_{\ell_i \ell_i} H_{\ell_i}^\top + Q_t^i \end{aligned}$$

Stacked Observation Model

- Stacked observation model for m measurements

$$\mathbf{z}_t = \{\mathbf{z}_t^1,\ \mathbf{z}_t^2,\ \ldots,\ \mathbf{z}_t^m\}$$

$$\begin{aligned}
\mathbf{z}_t &= \begin{bmatrix} \mathbf{h}(\mathbf{x}_t, c_t^1) \\ \mathbf{h}(\mathbf{x}_t, c_t^2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_t, c_t^m) \end{bmatrix} + \begin{bmatrix} \delta_t^1 \\ \delta_t^2 \\ \vdots \\ \delta_t^m \end{bmatrix} = \mathbf{h}_s(\mathbf{x}_t, \mathbf{c}_t) + \Delta_t \\
&\approx \mathbf{h}_s(\mu_t, \mathbf{c}_t) + H_x(\mathbf{x}_t - \mu_t) + \Delta_t
\end{aligned}$$

where stacked Jacobian is given by
$$H_x = \begin{bmatrix} H_x^1 \\ H_x^2 \\ \vdots \\ H_x^m \end{bmatrix}$$

and $\Delta_t \sim \mathcal{N}(0, Q_t) \quad Q_t = \text{diag}(Q_t^1, \ldots, Q_t^m)$

Courtesy R. Eustice

EKF-SLAM Stacked (Batch) Update

- Update

$$\bar{\boldsymbol{\mu}}_{z_t} = h_s(\bar{\boldsymbol{\mu}}_t, \mathbf{c}_t)$$

$$S_t = H_t \bar{\Sigma}_t H_t^\top + Q_t$$

$$K_t = \bar{\Sigma}_t H_t^\top S_t^{-1}$$

K_t is a (3+2n) x (2k) gain matrix composed of a weighted linear combination of the columns of Σ_t associated with R and m_i indices

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - \bar{\boldsymbol{\mu}}_{z_t})$$

$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

Hence, in general, because K_t is full, it non-trivially changes <u>all</u> of the elements in Σ_t during the covariance update

EKF-SLAM Sequential Update

for i=1:k

$$
\begin{aligned}
\bar{\boldsymbol{\mu}}_{z_t^i} &= h(\bar{\boldsymbol{\mu}}_r, \mathbf{m}_i) \\
S_t^i &= H_t^i \bar{\Sigma}_t {H_t^i}^\top + Q_t^i \\
K_t^i &= \bar{\Sigma}_t {H_t^i}^\top {S_t^i}^{-1} \\
\bar{\boldsymbol{\mu}}_t &= \bar{\boldsymbol{\mu}}_t + K_t^i(\mathbf{z}_t^i - \bar{\boldsymbol{\mu}}_{z_t^i}) \\
\bar{\Sigma}_t &= (I - K_t^i H_t^i)\bar{\Sigma}_t
\end{aligned}
$$

Think of this loop as a zero motion prediction + update step

endfor

$$
\begin{aligned}
\boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t \\
\Sigma_t &= \bar{\Sigma}_t
\end{aligned}
$$

Return the updated state

Courtesy R. Eustice

Batch vs Sequential Updates

- When can we do this?
  - When sensor measurements are independent, i.e.

$$Q_t = \text{diag}(Q_t^1, \ldots, Q_t^k)$$

- Why?
  - Because of conditional independence

$$p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \prod_{i=1}^{k} p(\mathbf{z}_t^i \mid \mathbf{x}_t)$$

- Are Batch vs. Sequential Updates Equivalent?
  - Yes – when observation models are linear (KF)
  - Not quite – when observation models are nonlinear (EKF)
    - Why? because our **linearization point** for each H_t^i <u>changes</u> as we perform the updates sequentially

Courtesy R. Eustice

EKF-SLAM Landmark Initialization

- $\mathbf{z}_t$ needs to be bijective in order to be able to perform landmark initialization from a single observation
- In the case of range/bearing observations, $h()$ is bijective
- Hence, we can instantiate a new landmark into our state vector as follows…

EKF-SLAM State Augmentation

- Augmentation model

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_L \\ \mathbf{x}_{\ell_{n+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_L \\ \mathbf{g}(\mathbf{x}_r, \hat{\mathbf{z}}_t^i) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_L \\ \mathbf{g}(\mathbf{x}_r, \mathbf{z}_t^i - \delta_t^i) \end{bmatrix}$$

True measurement

Actual measurement corrupted by noise where $\mathbf{z}_t^i = \hat{\mathbf{z}}_t^i + \delta_t$

$$\approx \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_L \\ \mathbf{g}(\mu_r, \mathbf{z}_t^i) + G_r(\mathbf{x}_r - \mu_r) + G_\delta(\delta_t - 0) \end{bmatrix}$$

State Augmentation of Belief

- Before

$$\mu_t = \begin{bmatrix} \mu_r \\ \mu_L \end{bmatrix} \qquad \Sigma_t = \begin{bmatrix} \Sigma_{rr} & \Sigma_{rL} \\ \Sigma_{Lr} & \Sigma_{LL} \end{bmatrix}$$

- After

$$\mu_{t_{\text{aug}}} = \begin{bmatrix} \mu_r \\ \mu_L \\ \mu_{\ell_i} \end{bmatrix} = \begin{bmatrix} \mu_r \\ \mu_L \\ \mathbf{g}(\mu_r, \mathbf{z}_t^i) \end{bmatrix}$$

$$\Sigma_{t_{\text{aug}}} = \begin{bmatrix} \Sigma_{rr} & \Sigma_{rL} & \Sigma_{r\ell_i} \\ \Sigma_{Lr} & \Sigma_{LL} & \Sigma_{L\ell_i} \\ \Sigma_{\ell_i r} & \Sigma_{\ell_i L} & \Sigma_{\ell_i \ell_i} \end{bmatrix}$$

$$\Sigma_{\ell_i r} = \Sigma_{r\ell_i}^\top = G_r \Sigma_{rr}$$

$$\Sigma_{\ell_i L} = \Sigma_{L\ell_i}^\top = G_r \Sigma_{rL}$$

$$\Sigma_{\ell_i \ell_i} = G_r \Sigma_{rr} G_r^\top + G_\delta Q_t^i G_\delta^\top$$

Courtesy R. Eustice

EKF-SLAM Algorithm

- Pseudocode

```
while 1
  Do Prediction
  if no z_t are available
     return
  else
    Determine correspondence variables c_t
      if any c_t are new
         Do State Augmentation
      endif
      Do Update (either batch or sequential)
  endwhile
```

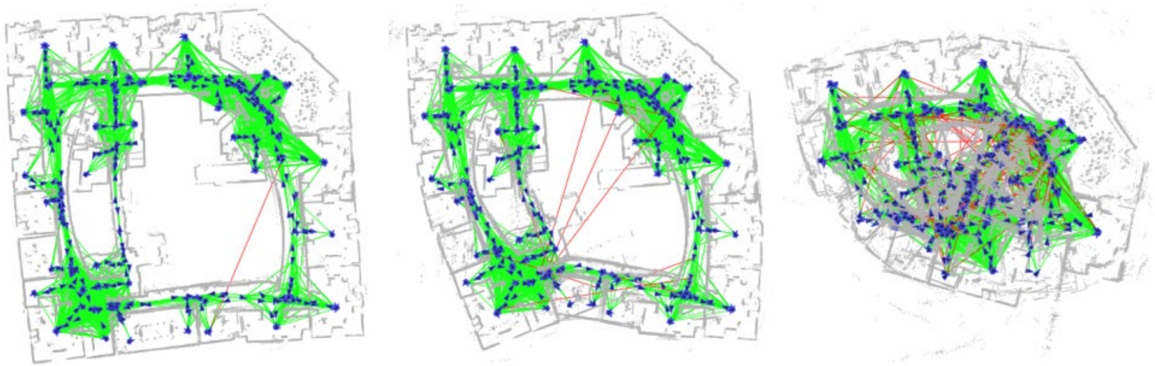

Courtesy R. Eustice

Data Association

- We’ve assumed known data association
  - “Observation *i* is of landmark *j*”

- What happens if we don’t have data association?

- What are the consequences if we get data association wrong?
  - Wormholes



Olson and Agarwal, Inference on Networks of Mixtures for Robust Robot Mapping, IJRR 2013

Importance of Data Association

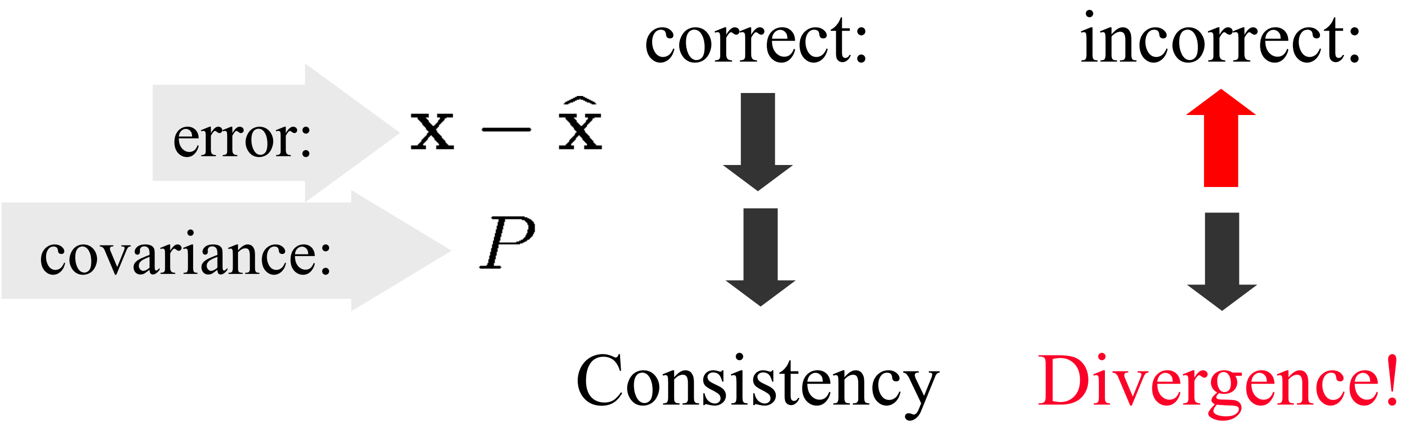
- Measurement z is used to improve estimate of x:

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

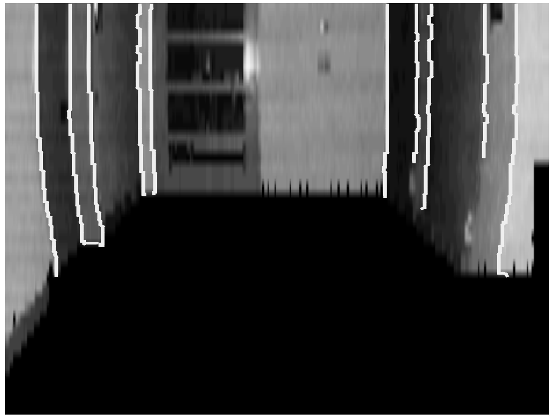
- If the association of measurement $\boldsymbol{E_i}$ with feature $\boldsymbol{F_j}$ is.....



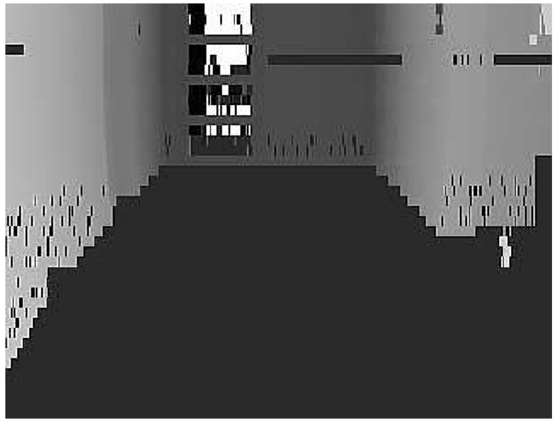
Courtesy R. Eustice

Data Association

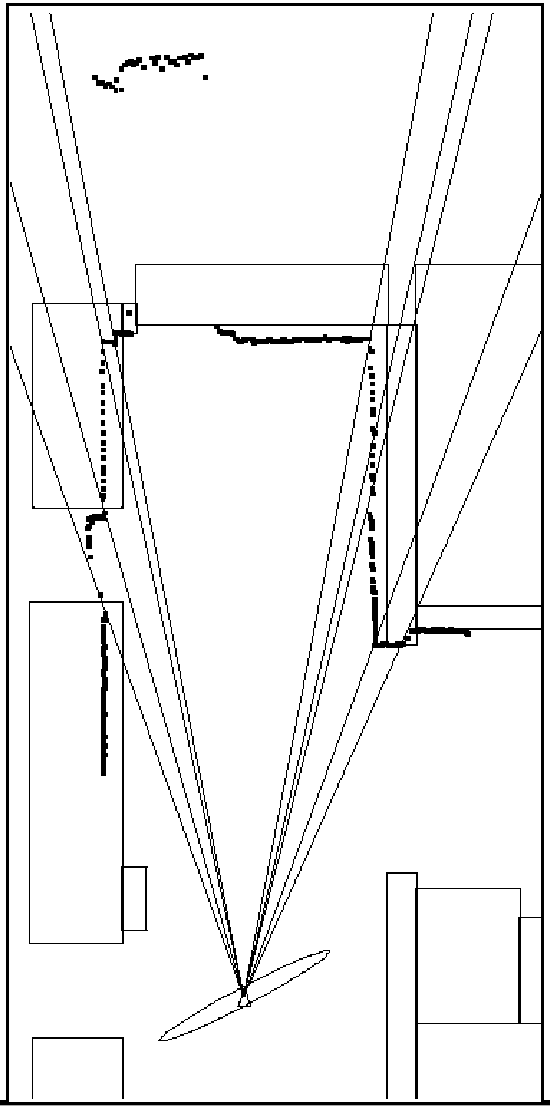
- Given an environment map
- And a set of sensor observations
- Associate observations with map elements



Vision



Laser



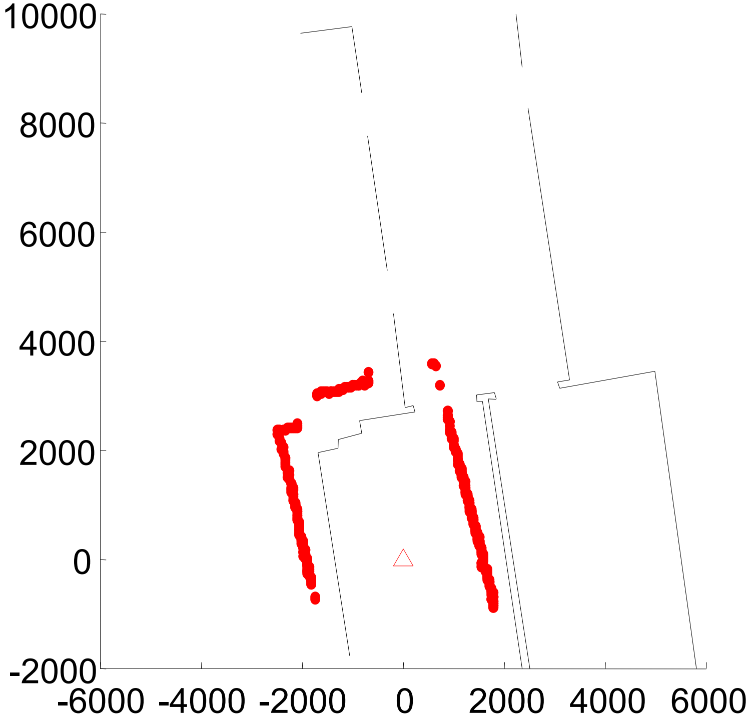
Courtesy R. Eustice

Difficulties: Clutter

- Influence of the type, density, precision and robustness of features considered:

Laser scanner:

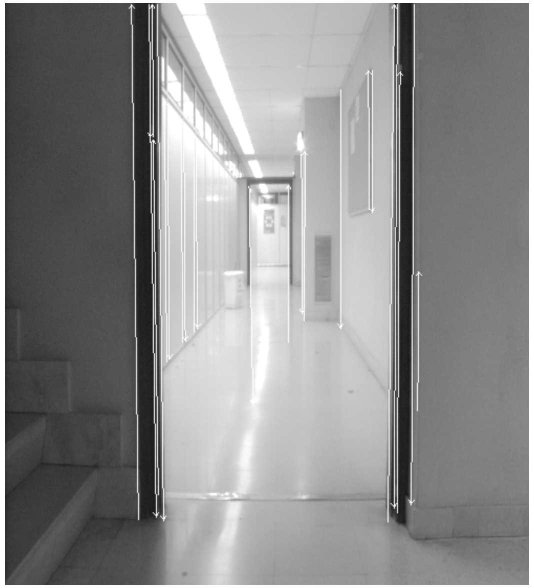
- *Small amount of features (n)*
- *Small amount of measurements (m)*
- Low spuriousness

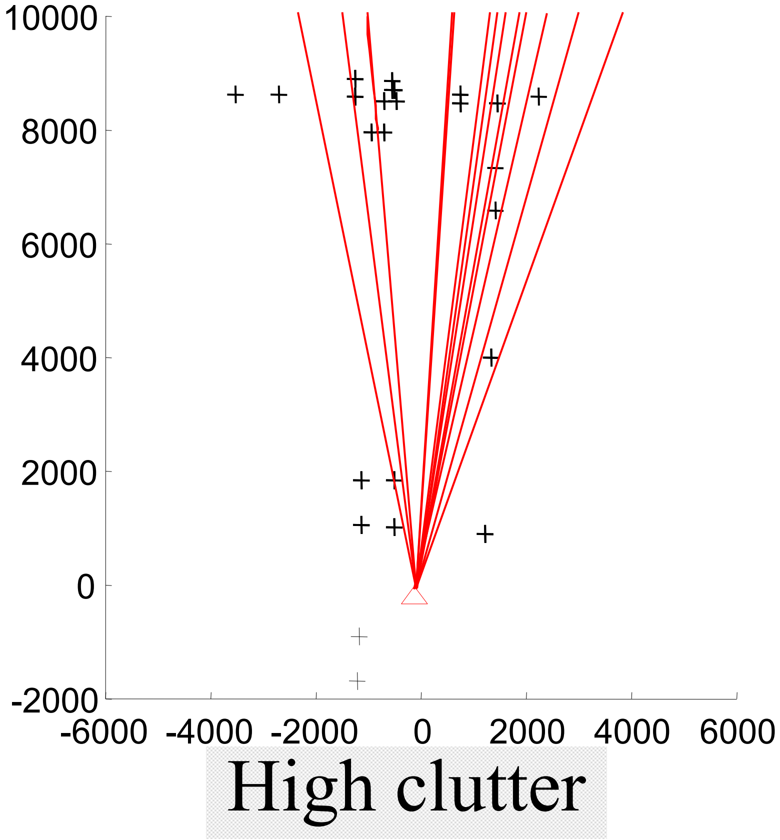


Low clutter

Difficulties: Clutter

- Vertical Edge Monocular vision:



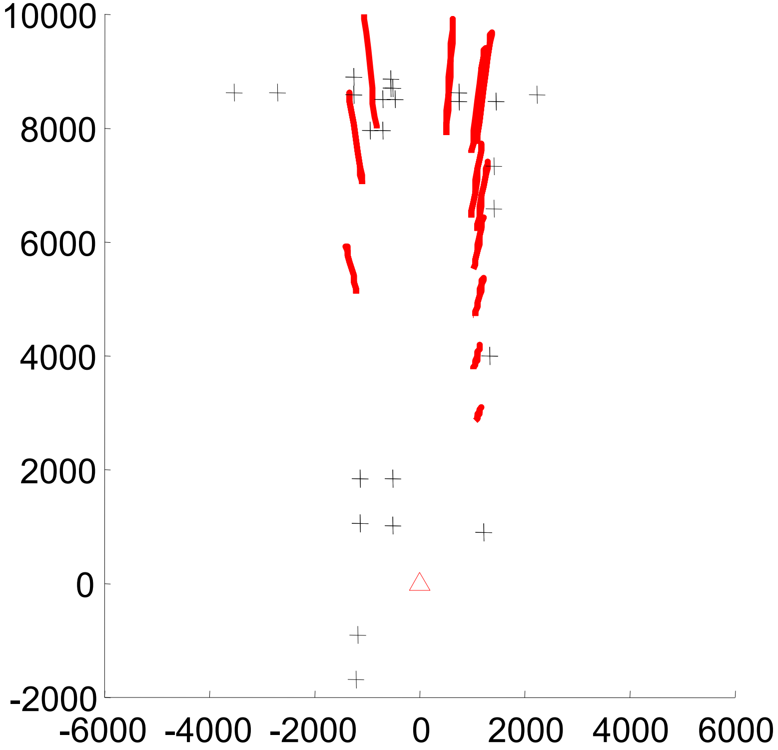


- Many features (n large)
- Many measurements (m large)
- no depth information
- higher spuriousness

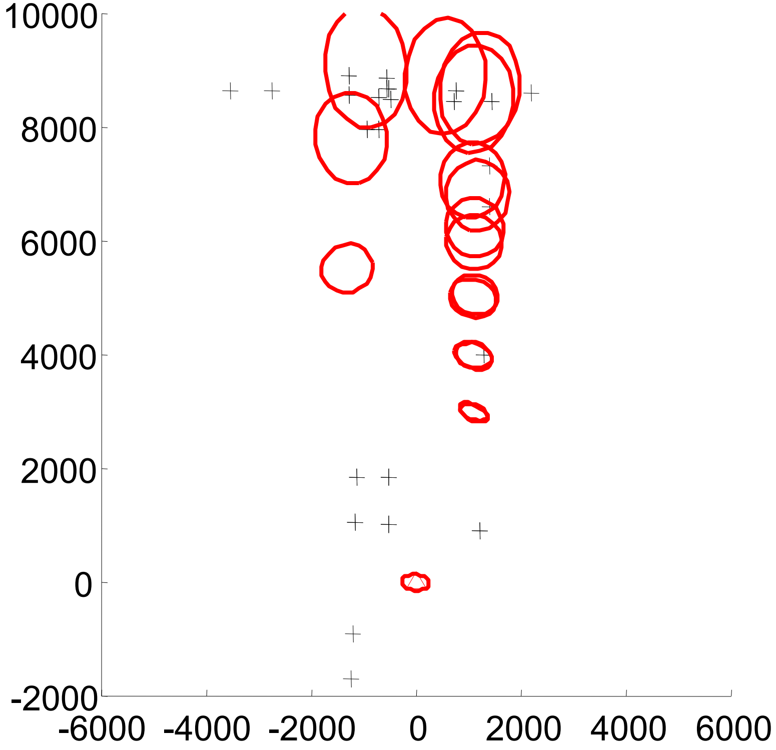
Courtesy R. Eustice

Difficulties: Imprecision

- Both the sensor and the vehicle introduce imprecision



Vertical Edge Trinocular vision:
variable depth precision
good angular precision



Robot imprecision:
introduces CORRELATED error

Courtesy R. Eustice

Approaches to Data Association

- Search in configuration space: find robot location with maximal data to map overlapping
  - Can be done with raw data
  - Or with features
    - Speed of convergence?

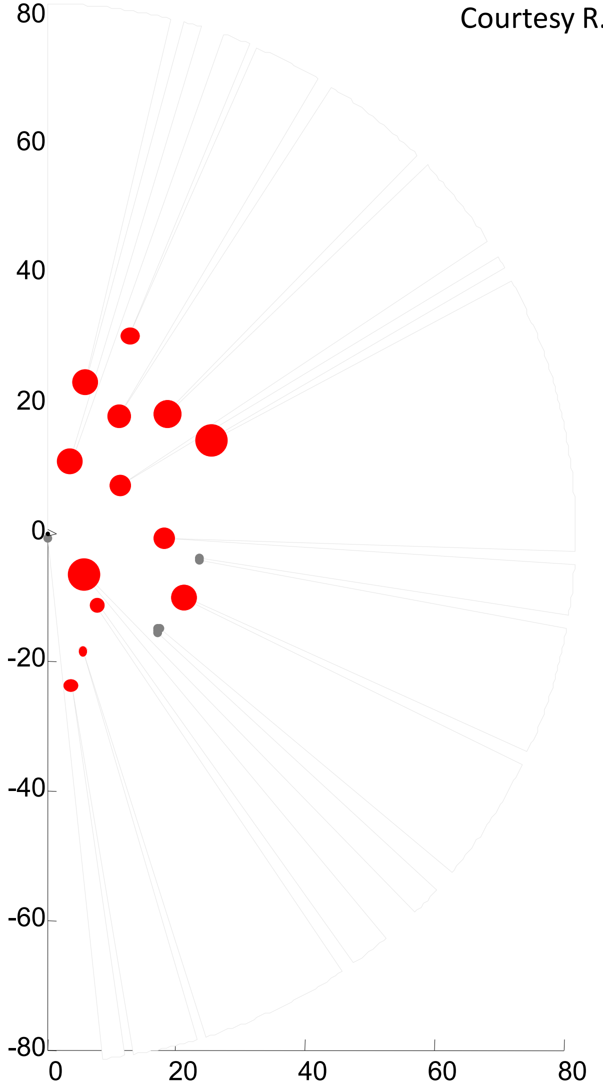
- Search in correspondence space: find a consistent correspondence hypothesis and compute robot location
  1. Extract features from data
  2. Feature-based map (points, lines, trees, …)
  3. Search for measurement to map feature correspondences
     - Exponential number of solutions?

Example

- Vehicle with SICK laser

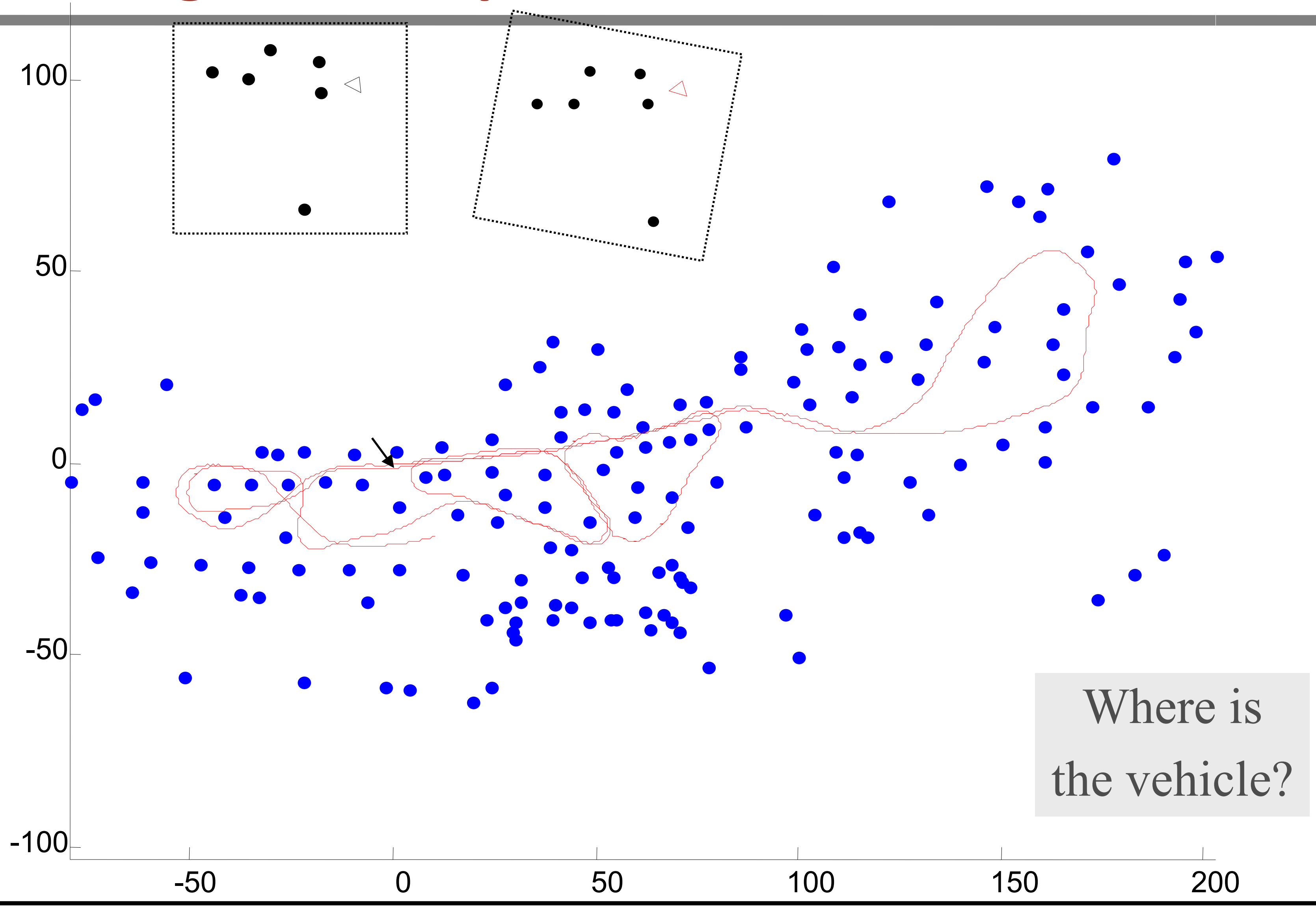
- Victoria Park, Sydney

- Detect trees:



Courtesy R. Eustice

Search in Configuration Space



Courtesy R. Eustice

Search in Configuration Space

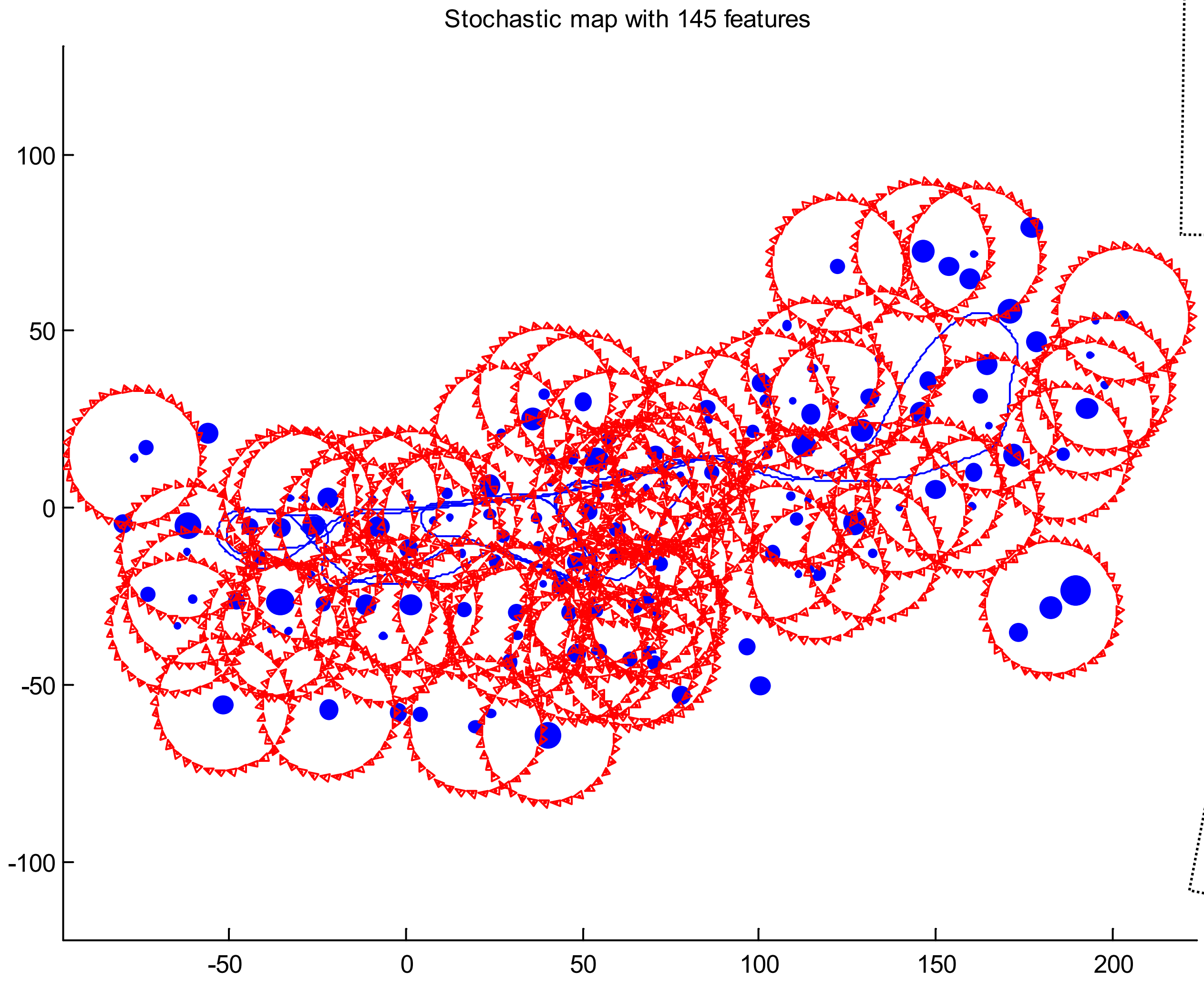
- Evaluate all vehicle locations looking for tree matchings

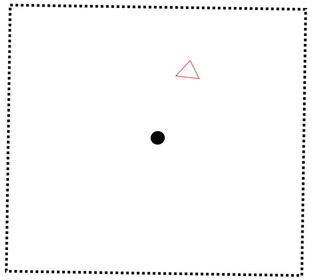
```
; (ui, vi): observation is relative to robot
; (xj, yj): absolute location of map feature j
;
mv = 0;
for x = xmin to xmax step xstep,
    for y = ymin to ymax step ystep,
        for t = tmin to tmax step tstep,
            v = 0;
            for i = 1 to m,
                (xi, yi) = compose(x,y,t,ui,vi);
                for j = 1 to n,
                    d = dist (xi, yi, xj, vj);
                    if d < dmax
                       v = v + 1;
                    fi
                rof
            rof
            if v > mv
               mv = v; best = (x,y,t);
            fi
        rof
    rof
rof
```

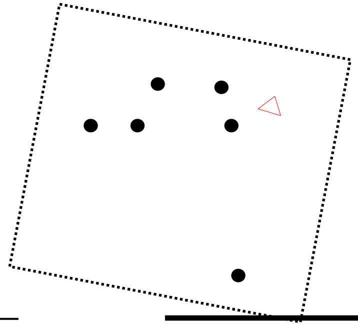
Courtesy R. Eustice

Data-driven search in configuration space (let the trees vote)

- Each pairing constrains the possible location of the vehicle (to a circle in this case).







Courtesy R. Eustice

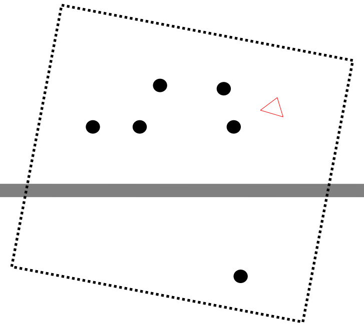
Data-driven search in configuration space (let the trees vote)

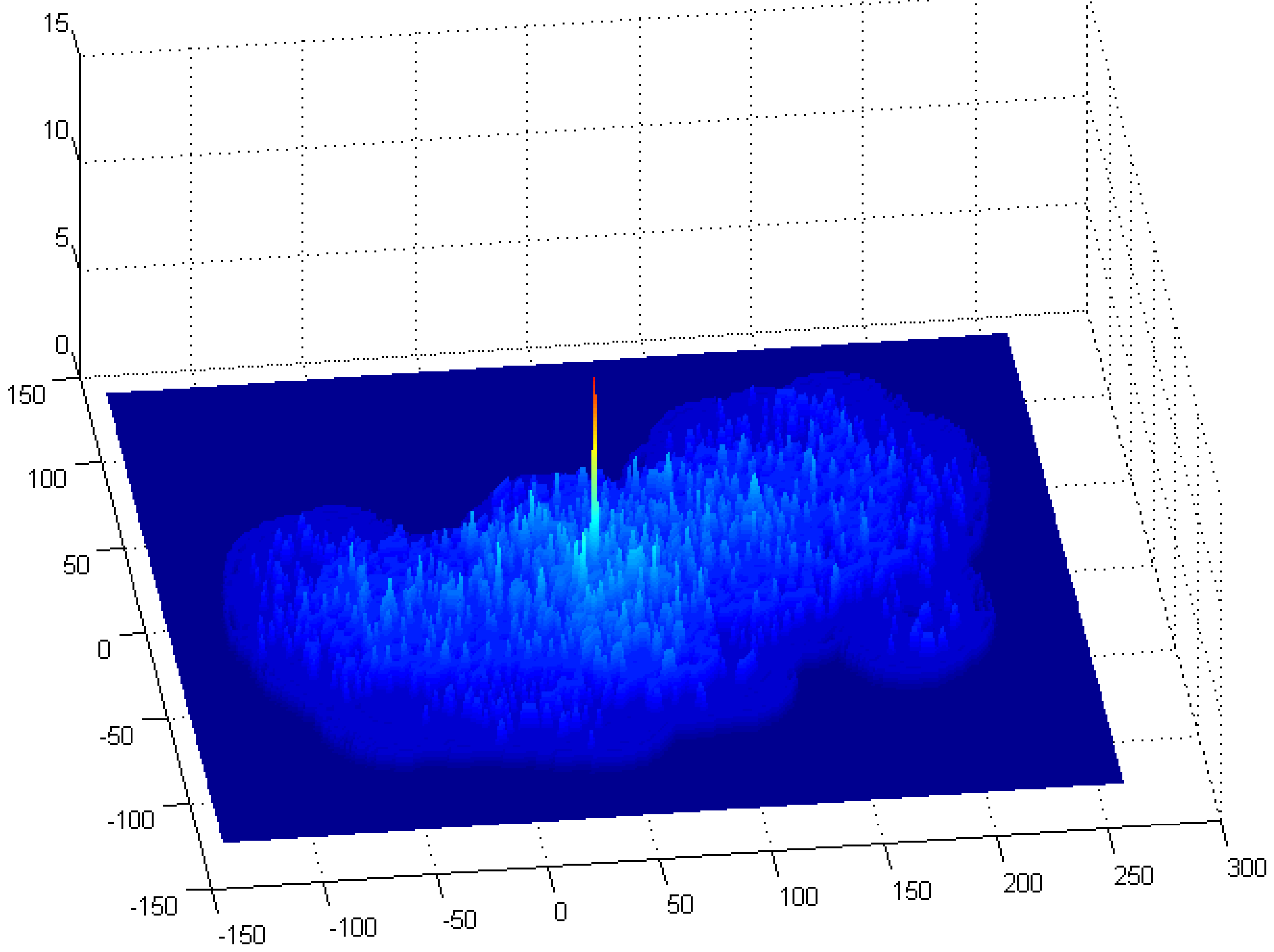
```
; (ui, vi): observation i relative to robot
; (xj, yj): absolute location of map feature j
;
for t = tmin to tmax step tstep,
   R = rotation(t);
   for i = 1 to m,
      (xi, yi) = R * (ui, vi);
      for j = 1 to n,
         (x, y) = (xj, yj) - (xi, vi);
         addvote(x, y, t);
      rof
   rof
rof
```

Courtesy R. Eustice

Results

- The most vc





Courtesy R. Eustice

Data Association (in Correspondence Space)

- $\boldsymbol{n}$ map features:

$$\mathcal{F} = \{F_1 \dots F_n\}$$

- $\boldsymbol{m}$ sensor measurements:

$$\mathcal{E} = \{E_1 \dots E_m\}$$

- **Goal:** obtain a hypothesis that associates each observation E_i with a feature F_{ji}

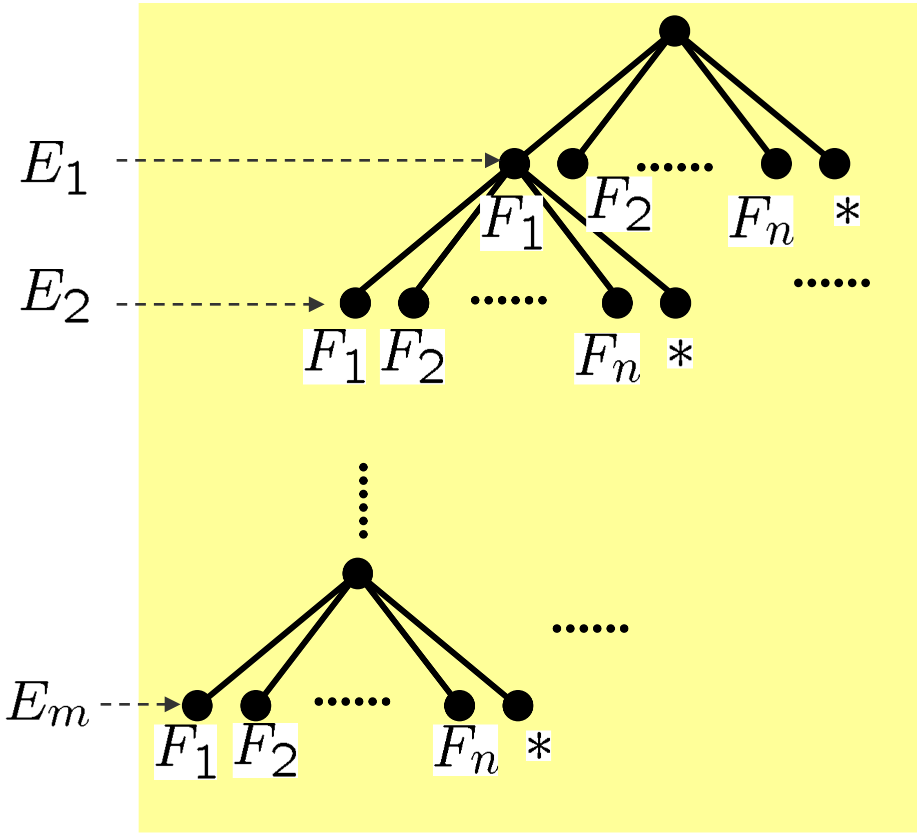
$$\mathcal{H}_m = [j_1 \dots j_i \dots j_m]$$

$$E_i \quad F_{j_i}$$

- Non matched observations:

$$j_i = 0$$

Interpretation tree (Grimson et al. 87):

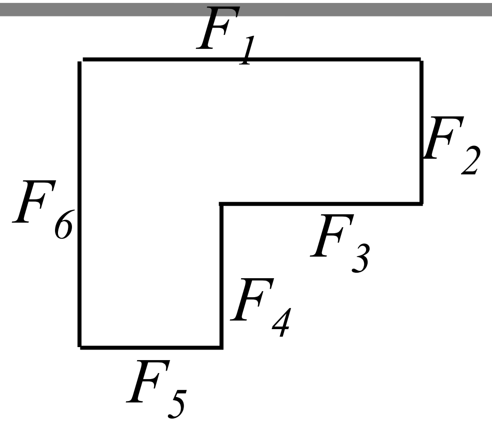


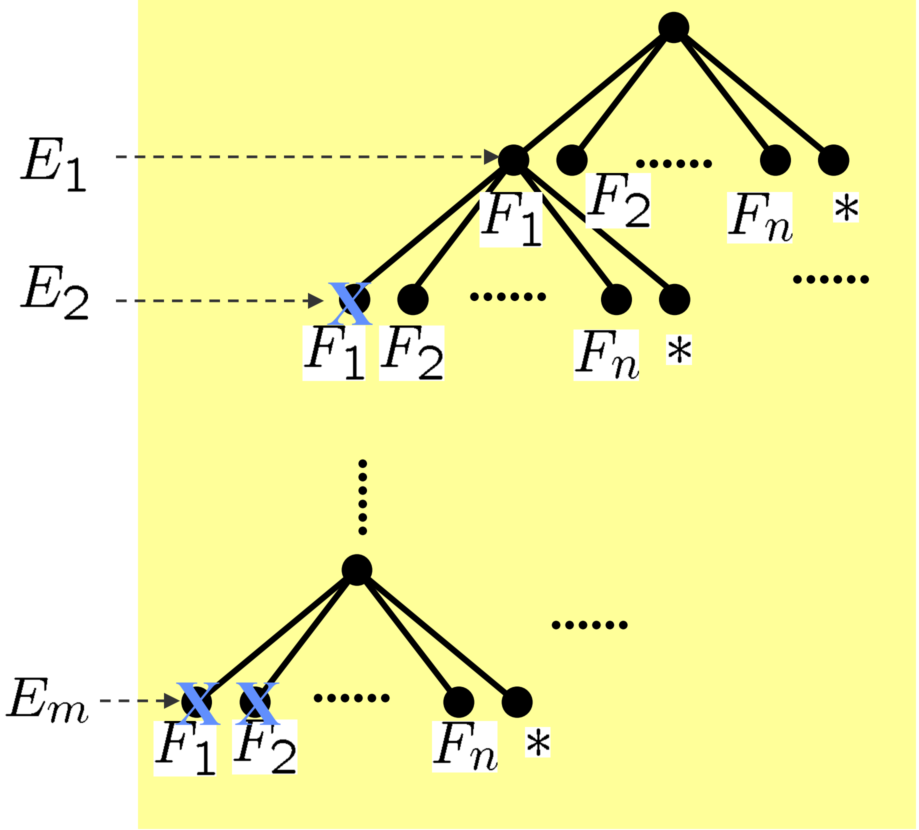
$(n+1)^m$ possible hypotheses

Courtesy R. Eustice

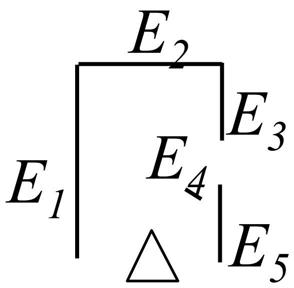
Use Constraints to Prune the Tree

- Map:





- Observations:

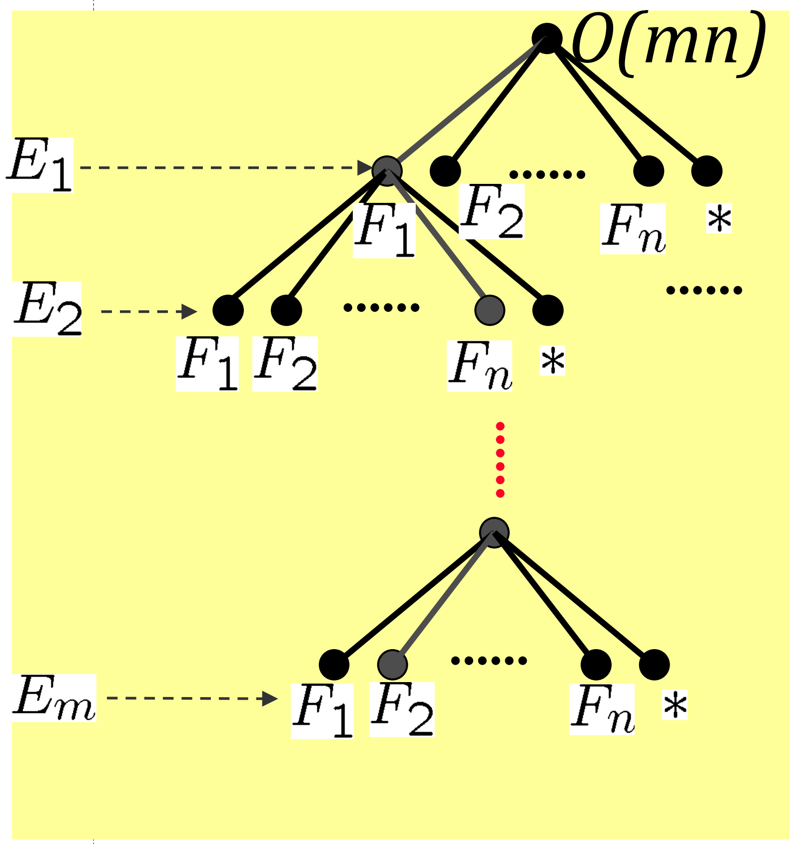


- Constraints:
  - Feature location (needs an estimation of robot location)
  - Geometric relations: angles, distances,… (location independent)

Nearest Neighbor

```
NN:
for i = 1 to m -- measurement E_i
    D2min = Mahalanobis2(E_i, F_1)
    nearest = 1
    for j = 2 to n -- feature F_j
        Dij2 = Mahalanobis2(E_i, F_j)
        if Dij2 < D2min then
            nearest = j
            D2min = Dij2
        fi
    rof
    if D2min <= Chi2(d_i,alpha) then
        H(i) = nearest
    else
        H(i) = 0
    fi
rof
```

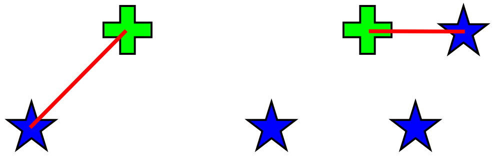
Greedy algorithm:



Courtesy R. Eustice

Data Association: Nearest Neighbor

- For each measurement, choose the closest landmark:



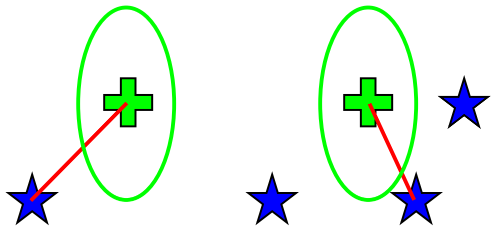
$$D_{kj}^{2,\mathrm{NN}} := \left\| h_{i_k j}(\hat{\mathbf{x}}) - \tilde{\mathbf{z}}_k \right\|^2$$

Accept if $$D_{kj}^{2,\mathrm{NN}} < D_{\max}^2$$

- For unmatched measurements, create new landmarks

Data Association: Maximum Likelihood

- Considers uncertainty of each landmark estimate



$$P(\tilde{\mathbf{z}}_k, j_k = j | Z^-) = \int_{\mathbf{x}} P(\tilde{\mathbf{z}}_k, j_k = j, \mathbf{x} | Z^-)$$
$$= \int_{\mathbf{x}} P(\tilde{\mathbf{z}}_k, j_k = j | \mathbf{x}, Z^-) P(\mathbf{x} | Z^-)$$
$$= \int_{\mathbf{x}} P(\tilde{\mathbf{z}}_k, j_k = j | \mathbf{x}) P(\mathbf{x} | Z^-)$$

$$D_{kj}^{2,\mathrm{ML}} := \| h_{i_k j}(\hat{\mathbf{x}}) - \tilde{\mathbf{z}}_k \|^2_{C_{i_k j}}$$

$$P(\tilde{\mathbf{z}}_k, j_k = j | Z^-)$$
$$= \int_{\mathbf{x}} \frac{1}{\sqrt{|2\pi\Gamma|}} e^{-\frac{1}{2}\| h_{i_k j}(\mathbf{x}) - \tilde{\mathbf{z}}_k \|^2_\Gamma} \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}\|^2_\Sigma}$$
$$\approx \frac{1}{\sqrt{|2\pi C_{i_k j}|}} e^{-\frac{1}{2}\| h_{i_k j}(\hat{\mathbf{x}}) - \tilde{\mathbf{z}}_k \|^2_{C_{i_k j}}}$$

Accept if $D_{kj}^{2,\mathrm{ML}} < \chi^2_{d,\alpha}$

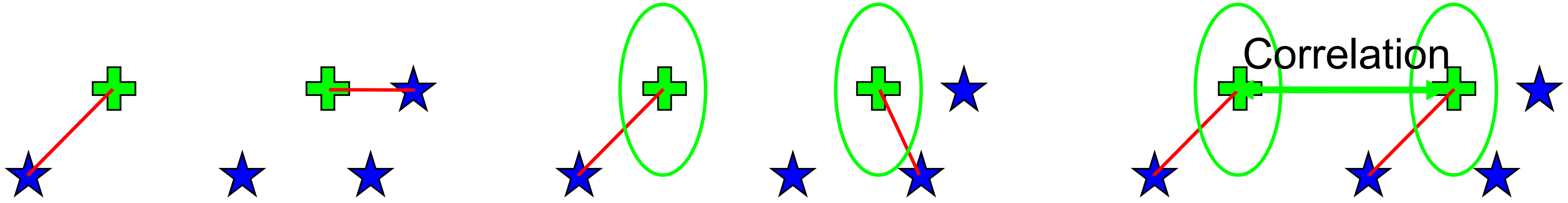
$$C_{i_k j} := \left.\frac{\partial h_{i_k j}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \Sigma \left.\frac{\partial h_{i_k j}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}}^T + \Gamma.$$

Data Association – Joint Compatibility

Nearest Neighbor

Maximum Likelihood
Individual Compatibility

Joint Compatibility



Landmark

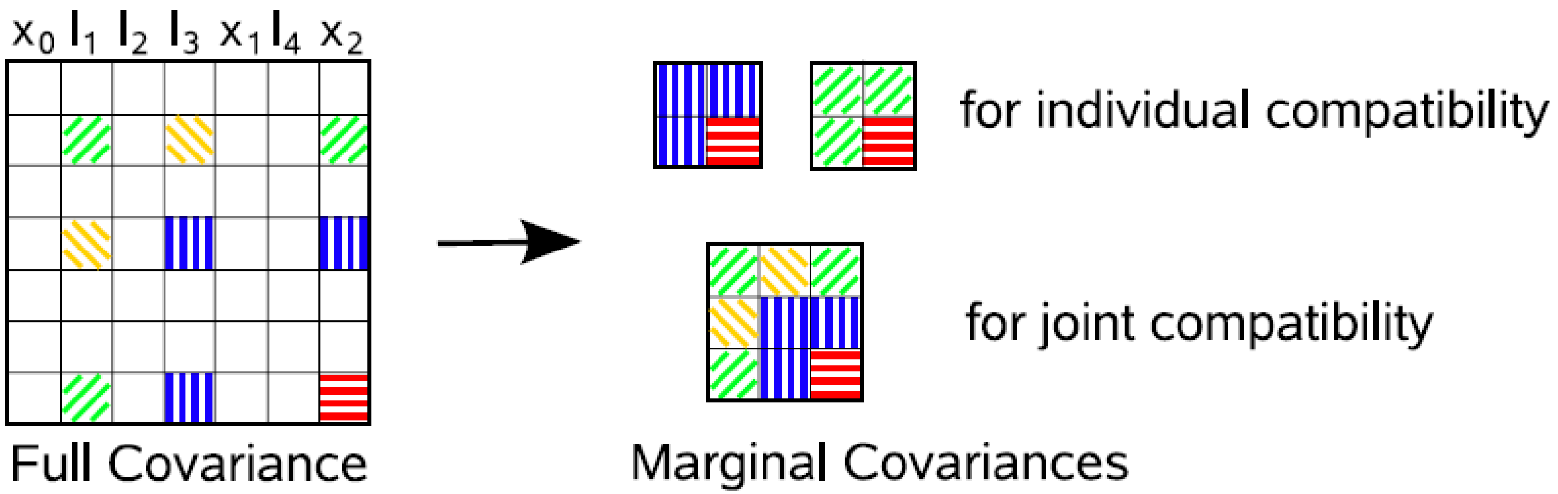
Measurement / Feature

Assignment

Estimation uncertainty

Kaess et al., RAS 09

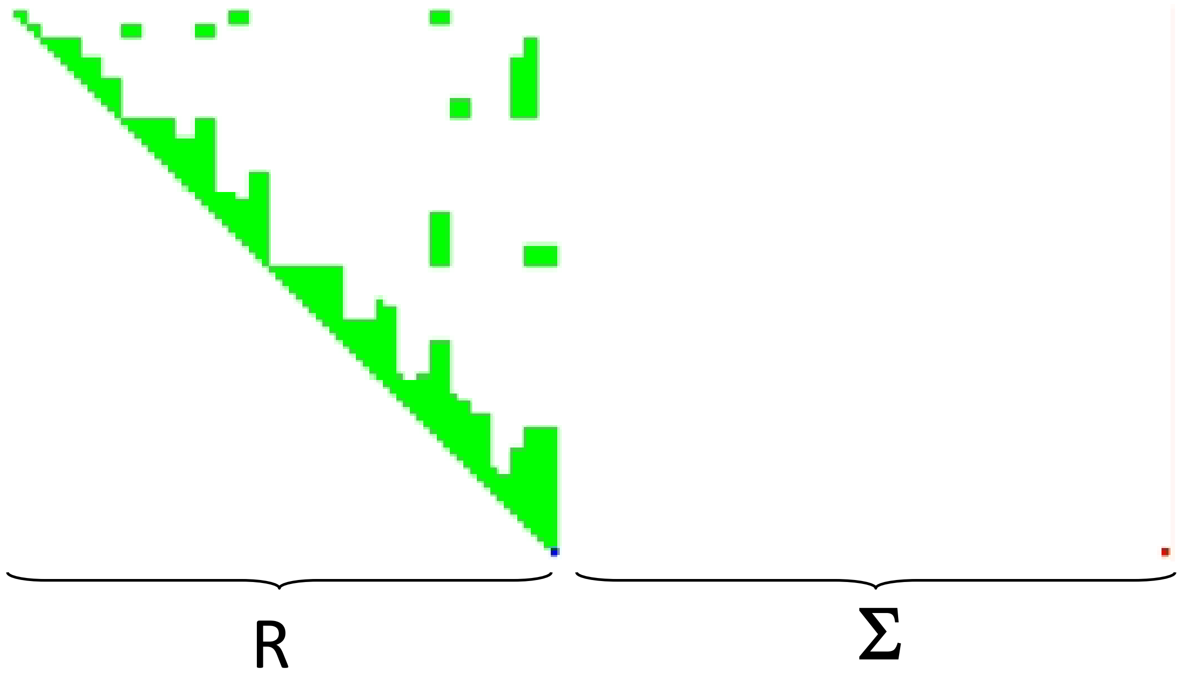
Marginal Covariances



Too expensive

Key insight: Only a few entries are needed !

Recovering Marginal Covariances



Example for n=81, nnz=717

Non-zero entry
Calculated entry
Accessed entry

Golub 1980:

Only calculate entries corresponding to non-zeros in R (**sparse!**)

Can recover additional entries as needed

Joint Compatibility Branch and Bound (JCBB)

Purpose: Find the largest hypothesis with jointly consistent pairings

```
JCBB(H, i): -- find pairings for feature E_i, i <= m
  if i > m -- leaf node?
        if pairings(H) > pairings(Best) -- did better?
              Best = H
  else
        for j = 1 to n
              if individual_compatibility(E_i, F_j) and then
                 joint_compatibility(H, E_i, F_j)
                    JCBB([H j], i + 1) –- pairing (E_i, F_j) accepted

  if pairings(H) + m – i >= pairings(Best) –- can do better?
        JCBB([H 0], i + 1) -- star node: E_i not paired
```

Courtesy R. Eustice

JCBB Discussion

- Upsides:
  - Nice probabilistic foundation
  - Quite effective in practice
- Downsides:
  - Still exponential time in the worst case
  - Like all joint assignment schemes, adds latency to data association.

Courtesy R. Eustice

Vision: Exploit Appearance!

- Feature descriptor (SIFT, FAST, BRIEF, ORB, SuperPoint…)

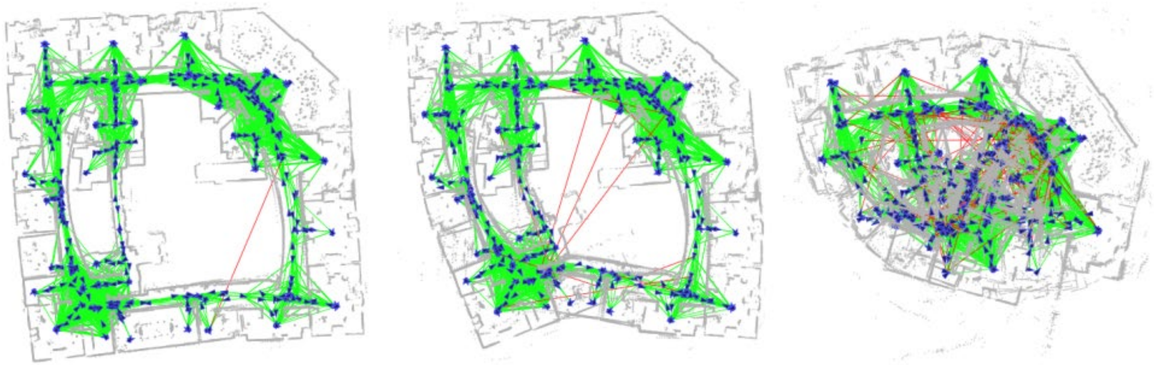
- For loop closing: High similarity between features that are extracted from two images taken in the same environment
- Only tells us that similar environment, still need to check geometry
- Look at all features in an image:
  - Bag-of-words, e.g. FAB-MAP (Cummins and Newman, IJRR 2008)
  - Randomized Ferns (Ozuysal et al., CLF 2010)
  - Deep Learning (SuperGlue…)

Data Association during Optimization

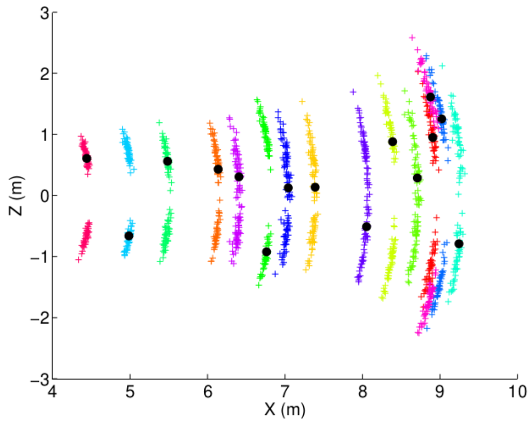
- Mixed discrete/continuous optimization problem, computationally very challenging
- Approximations:
  - EM / MCMC (e.g. “SFM without Correspondences”, Dellaert 2000)
  - Nonparametric

Robust Perception

- Failure causes:
  - Perceptual aliasing
  - Wheel slip
  - Sensor failure
  - Nonlinear function -> non-Gaussian posterior

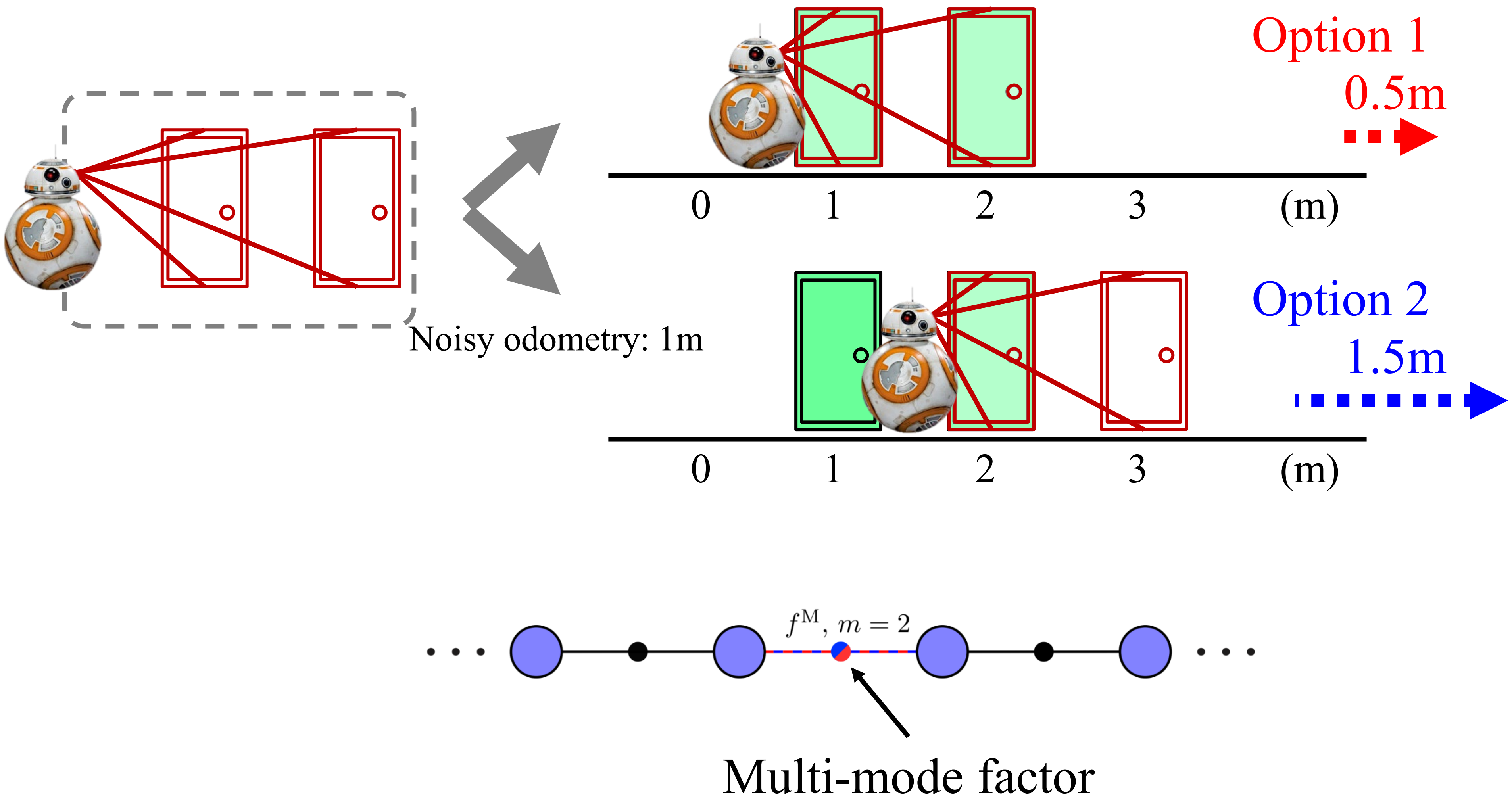


Olson and Agarwal, Inference on Networks of Mixtures for Robust Robot Mapping, IJRR 2013



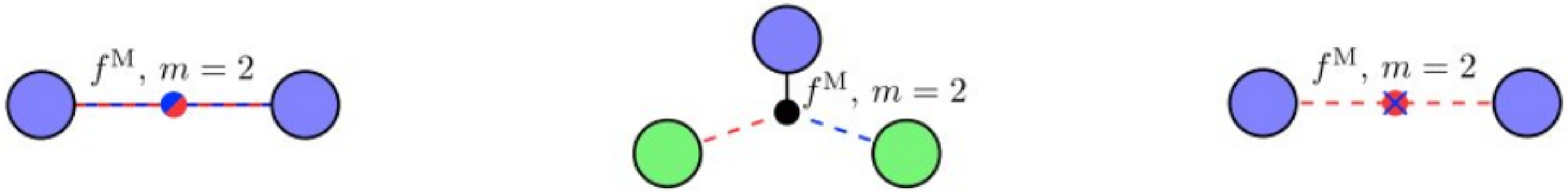
- How to solve in the backend?

Factor Graph with Ambiguity



Factor Graph with Ambiguity

- Common multi-modal factors:



(a) Multi-measurement **(b)** Multi-association **(c)** Boolean

- How to solve? Clearly the result (posterior) will not be Gaussian

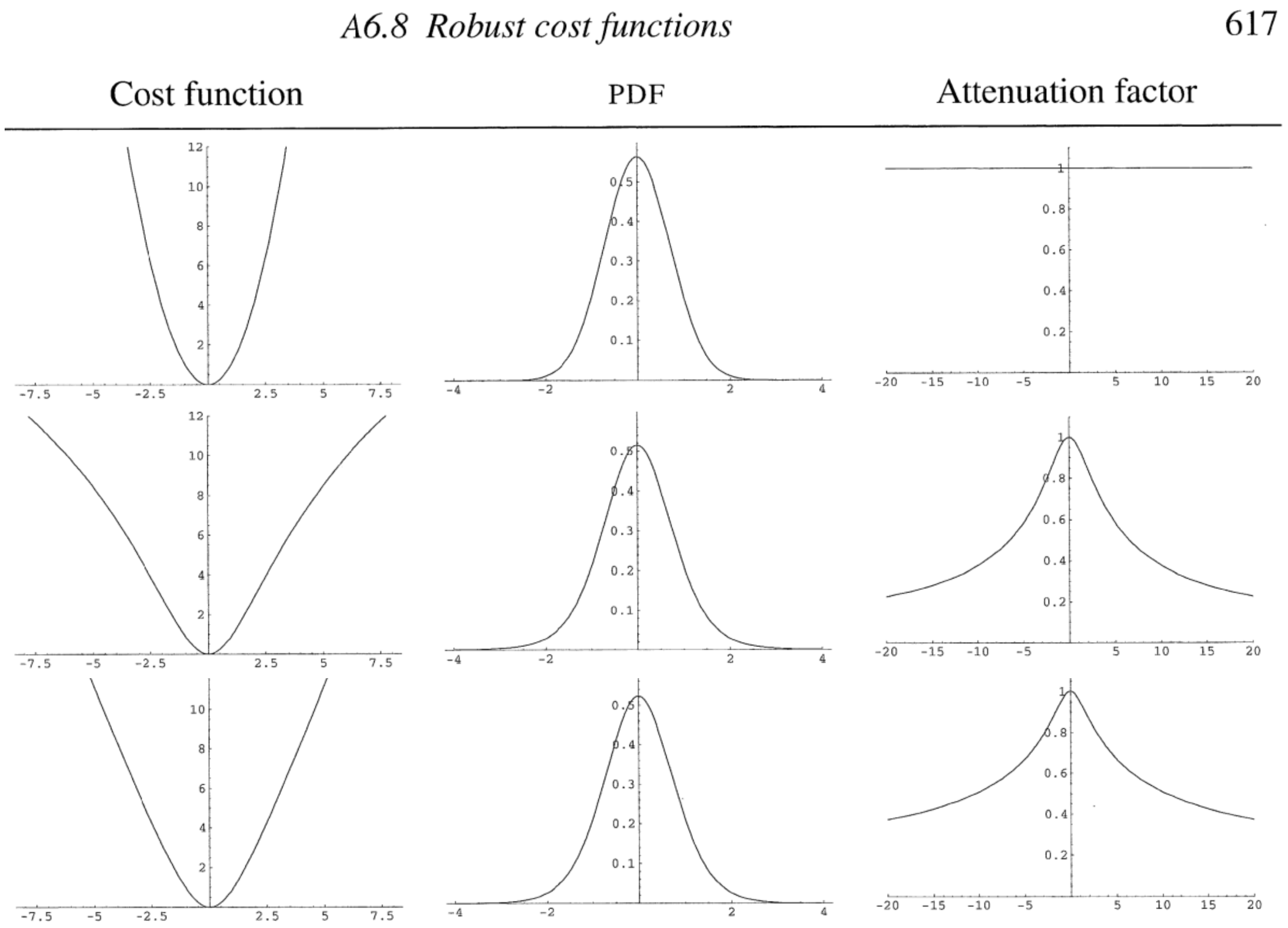
Robust Estimators

Relevant paper: Chebrolu et al., "Adaptive Robust Kernels for Non-Linear Least Squares Problems", RA-L 2021

Quadratic/
Gaussian

Cauchy

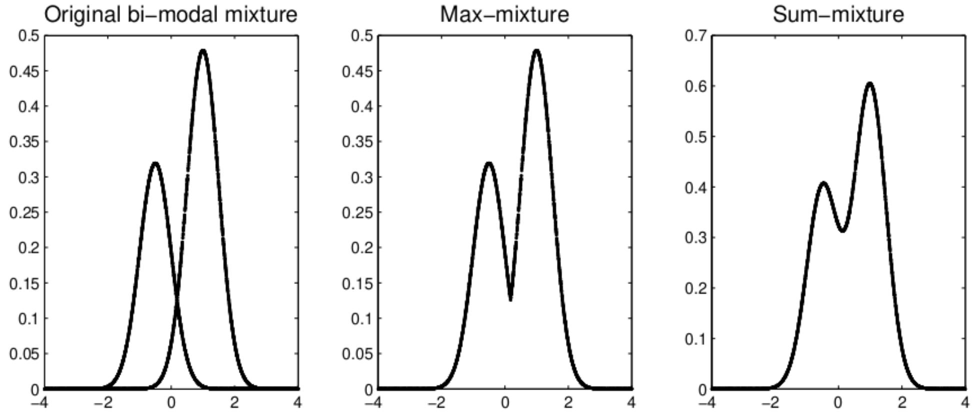
Pseudo Huber



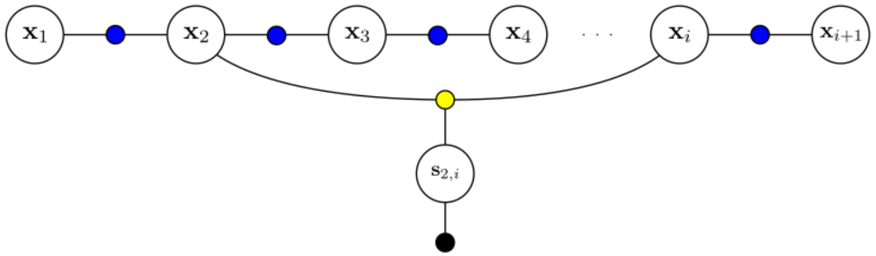
Hartley and Zisserman, Multiple View Geometry, second edition, 2003

Other Solutions

- Max-Mixtures



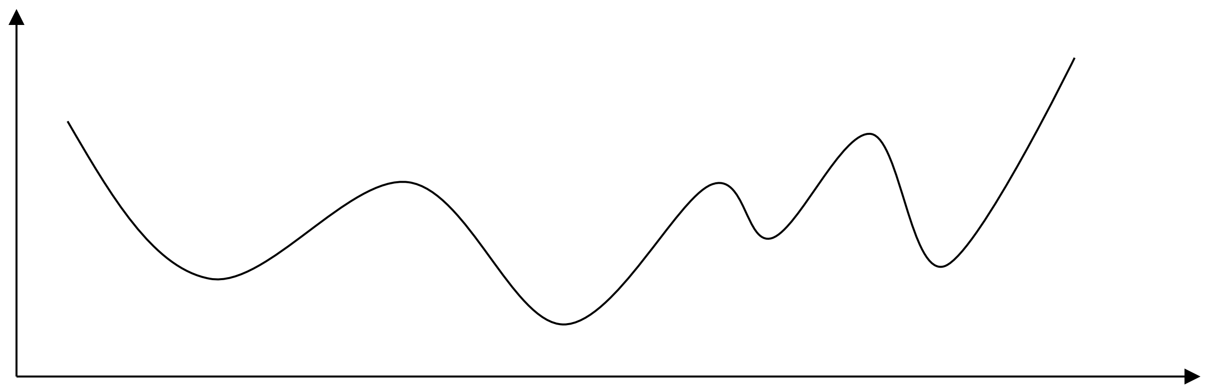
- Switchable Constraints
  - Switch variable
  - Continuous “switch”



Olson and Agarwal, “Inference on Networks of Mixtures for Robust Robot Mapping”, IJRR 2013
Sünderhauf and Protzel, “Switchable Constraints for Robust Pose Graph SLAM”, IROS 2021

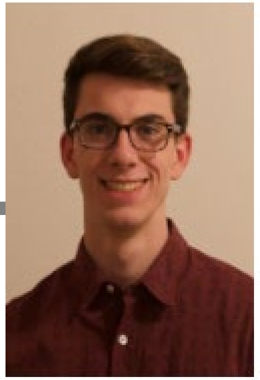
Global Solver for Pose Graphs

- Solves semidefinite relaxation of pose graph problem
- “capable of efficiently solving a generally intractable problem within a restricted (but still practically relevant) operational regime”
- Can even certify correctness of results in some situations



Rosen, Carlone, Bandeira, Leonard, “SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group”, IJRR 2019

Robust Solver: Graduated Non-Convexity



- GNC:

  Yang, Antonante, Tzoumas, Carlone, “Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection”, RA-L 2020

- riSAM:
  - Incremental GNC
  - Scale invariant graduated kernel
  - Novel trust region method

  McGann, Rogers, Kaess, “Robust Incremental Smoothing and Mapping (riSAM), ICRA 2023

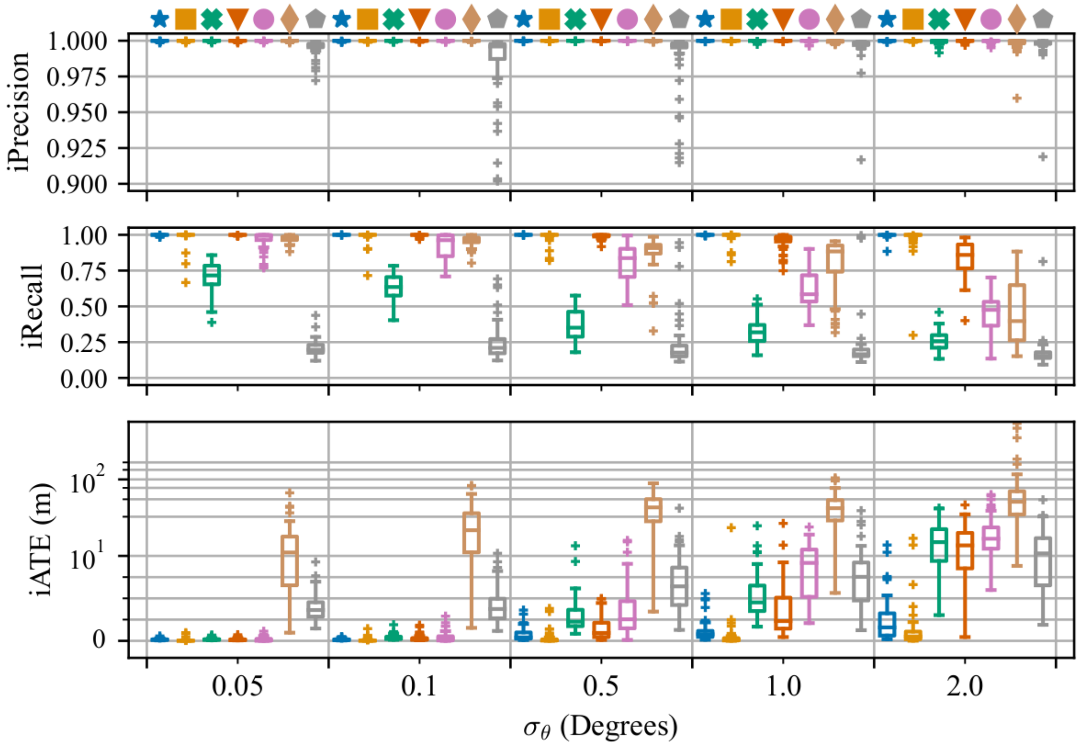


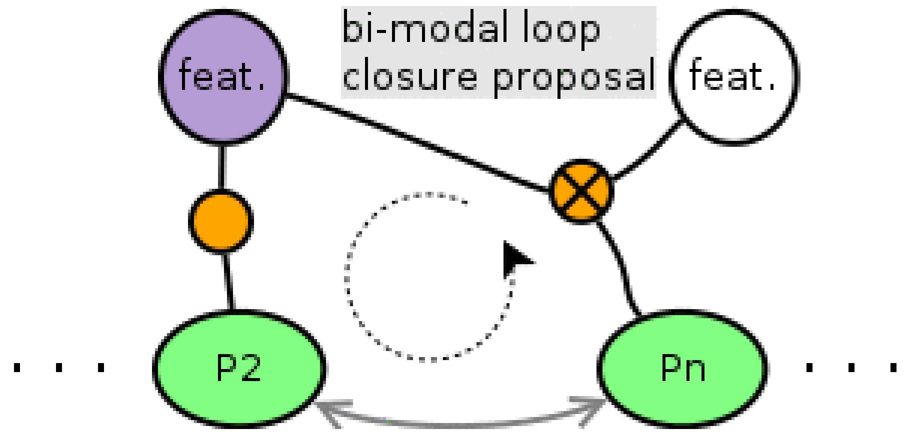
Fig. 6: Performance of our incremental algorithm riSAM (★), prior batch methods GNC (■) and PCM (✖), and prior incremental methods Geman-McClure (▼), Max-Mixture (●), Huber (◆), and Switchable Constraints (⬟) evaluated on random grid world trajectories generated with increasing levels of odometric noise. Our method, riSAM, outperforms all others methods as the initialization quality decreases with larger odometric noise. *Note: iATE plot has linear scale up to* 10^2 *and log scale above.*

Nonparametric Belief Solution to the Bayes Tree

New multi-modal loop closure model directly in factor graph

Simultaneously consider:
- New; or
- Previous feature;

NOT null-hypothesis:
- Deferred data association



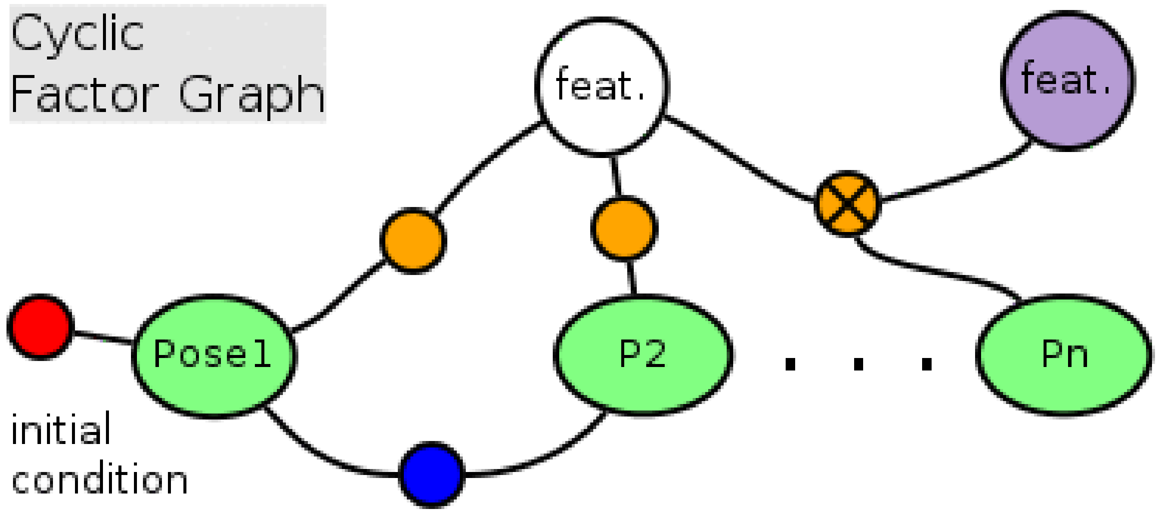
Discrete belief is marginalized out, belief is kept in continuous domain

$$[Z \mid \Theta_j] \propto \sum_{\forall \gamma \in \Gamma} [Z \mid \Theta_j, \gamma] [\gamma], \qquad \gamma \sim Cat(\rho)$$

Nonparametric Belief Solution to the Bayes Tree

Bayes tree (from iSAM2) removes cycles in original factor graph

- Refactor with elimination game;
- Near optimal representation & loss-less;
- Nonparametric Belief Propagation on tree.



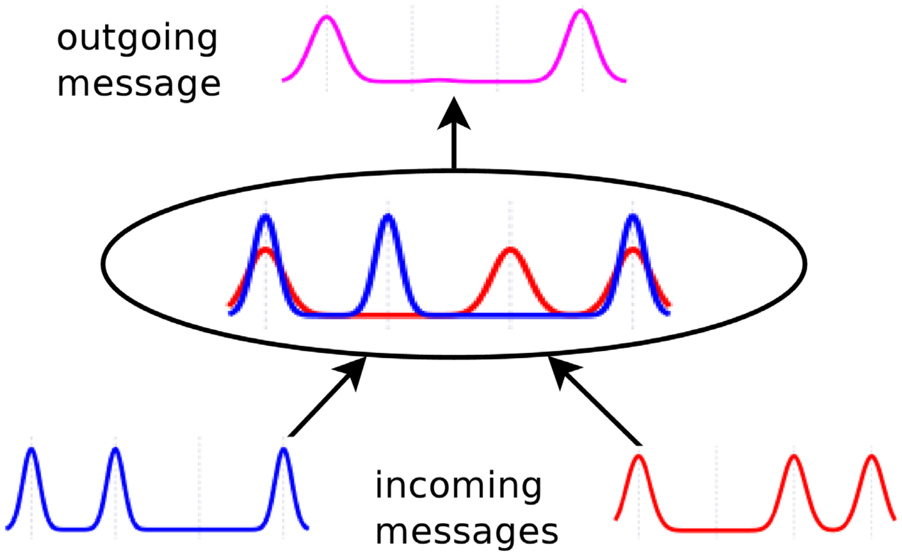
Nonparametric Belief Solution to the Bayes Tree

Recovers a non-Gaussian, multi-modal posterior for SLAM

- a.k.a **Multi-modal iSAM**
- Nonparametric belief propagation
- Implementation on github.com:
- dehann/IncrementalInference.jl

Possible via:

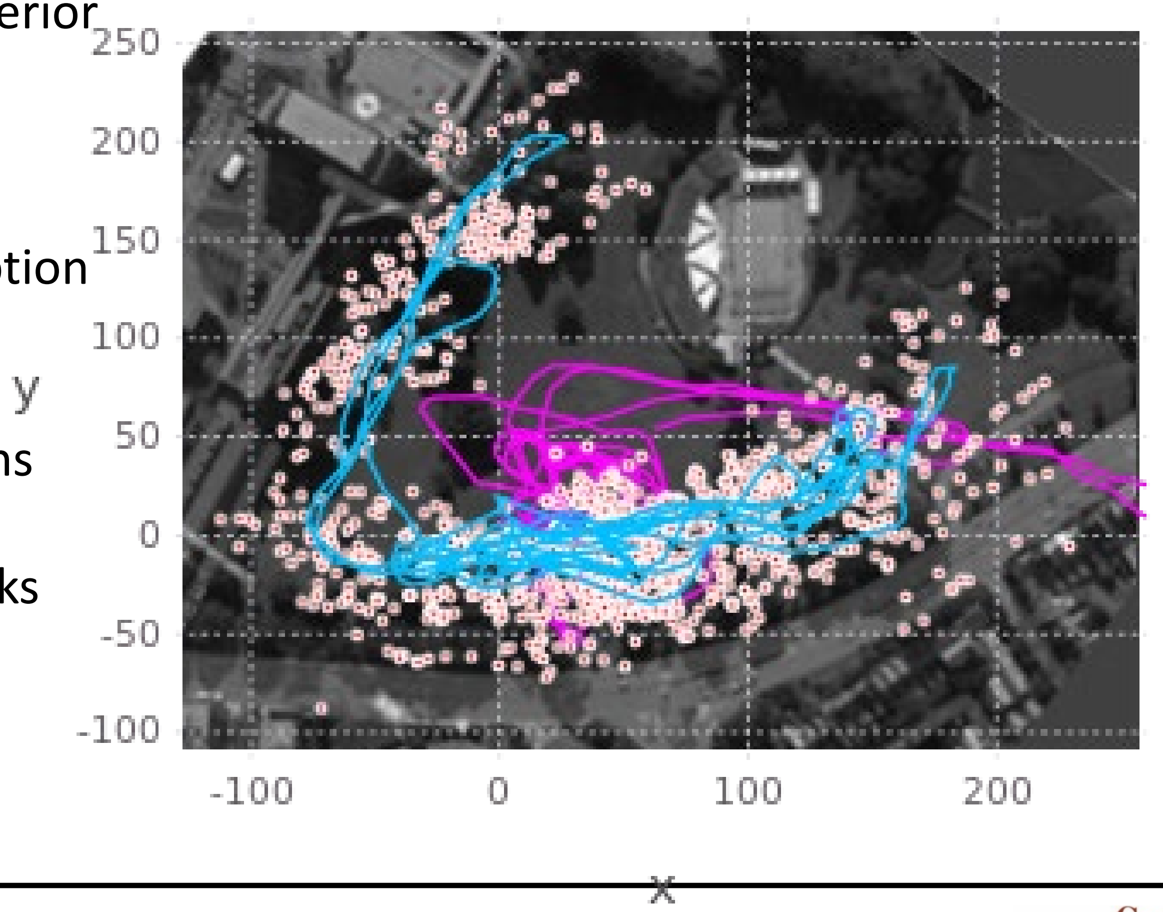
- Approximating belief
- Efficient posterior estimation
- Low dimensions per clique
- Weak modes decay (consensus)
- Multi-processor enabled



Nonparametric Belief Solution to the Bayes Tree

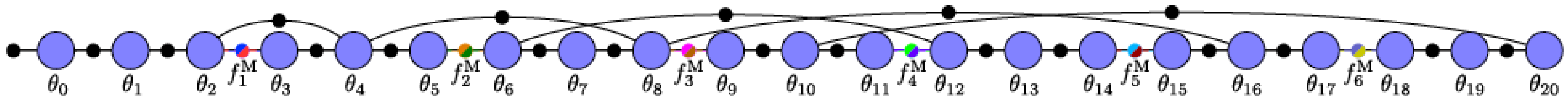
Multi-modal iSAM with SLAM example
- Non-Gaussian
- Estimates multi-modal posterior

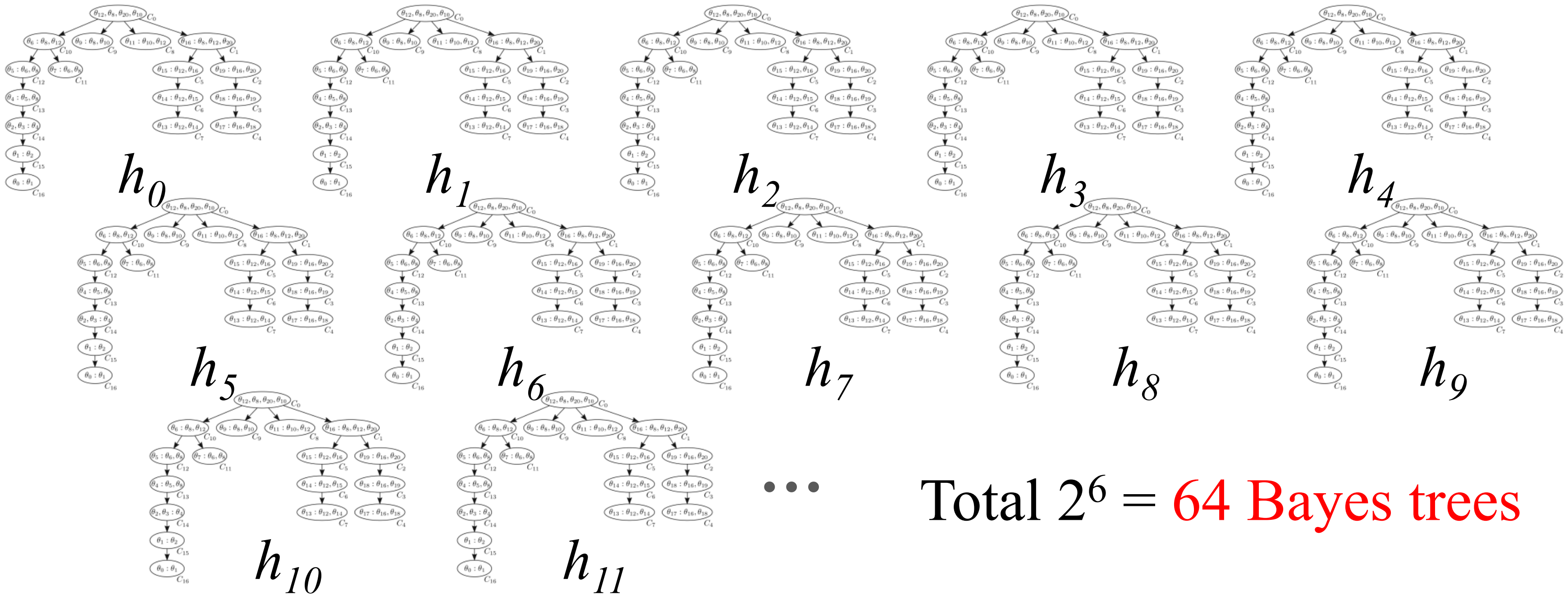
Works at scale
- Victoria park dataset:
- Eg. 10% loop closure corruption
- MAP blue trace (& trees)
- MLE magenta trace
- Several thousand dimensions
- 1700 bimodal constraints
- 482 uncertain tree landmarks
- > 2^482 modes
- ~3 hours compute
  for 25mins trajectory



An Old Idea: Multi-hypothesis Tracking

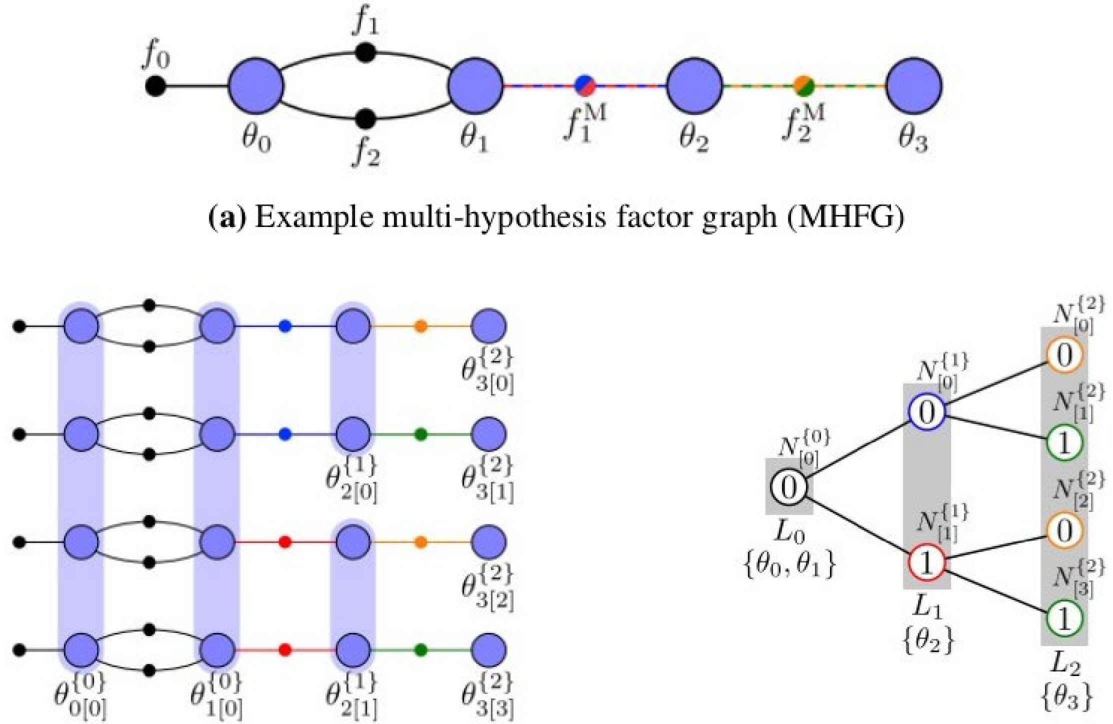
- Run one solver per hypothesis: expensive for SLAM!
- Exponential growth: Requires pruning





Total 2^6 = 64 Bayes trees

Overlapping Computation



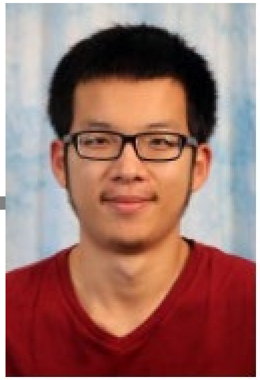
(a) Example multi-hypothesis factor graph (MHFG)

(b) Expanded MHFG

(c) Example hypo-tree

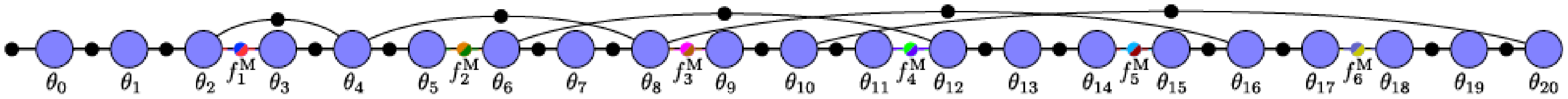
Save computation: Calculate shared components only once

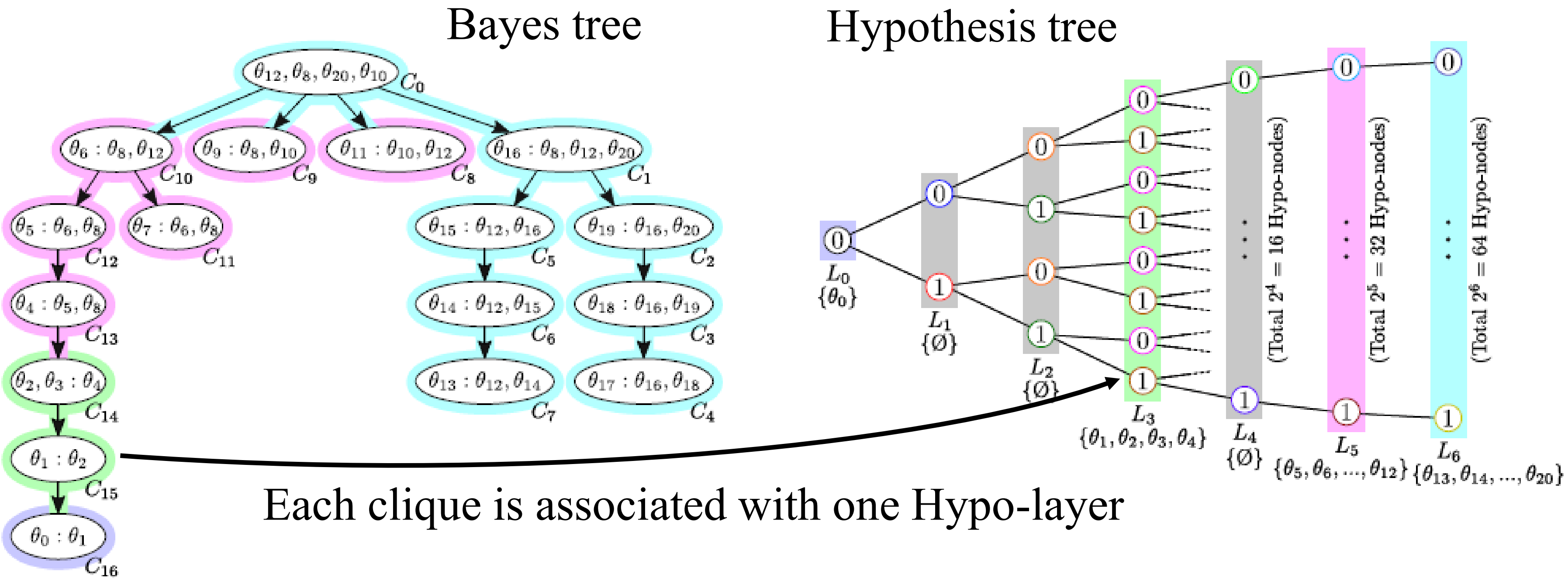
Hsiao and Kaess, ICRA 2019



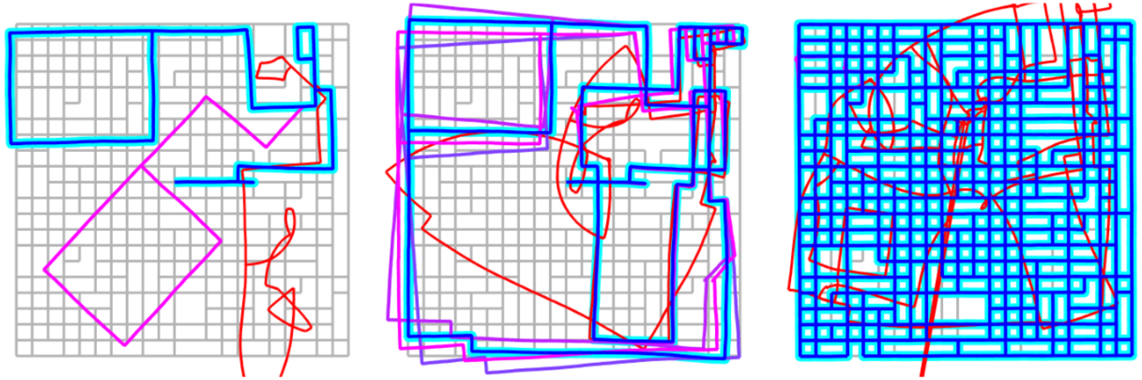
Multi-hypothesis Bayes Tree

- One Bayes tree contains all hypotheses:

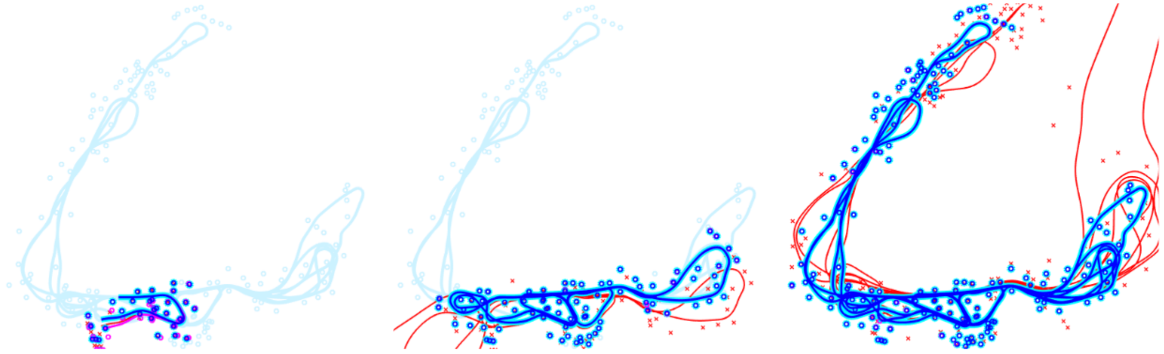




Mapping with Multiple Hypotheses



(a) Type #1: ambiguity in odometry estimates with total 2^{173} complexity



(b) Type #2: ambiguity in data association with total 2^{412} complexity

Mapping with Planar Surfaces

Hsiao, Westman, Kaess, “Dense Planar-Inertial SLAM with Structural Constraints”, ICRA 2018

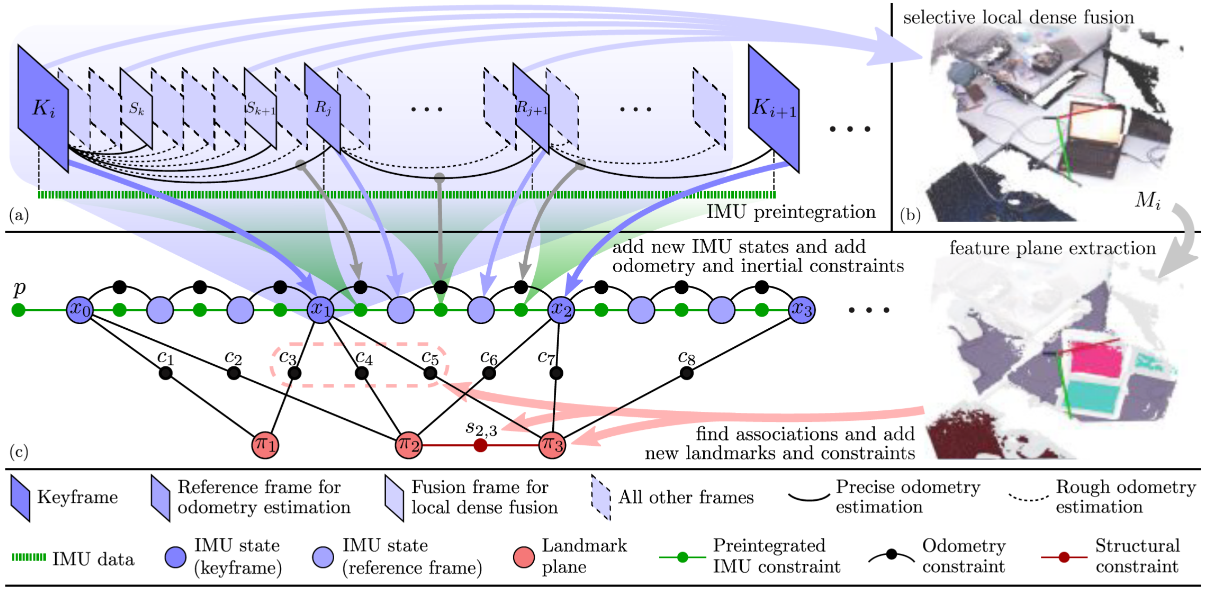
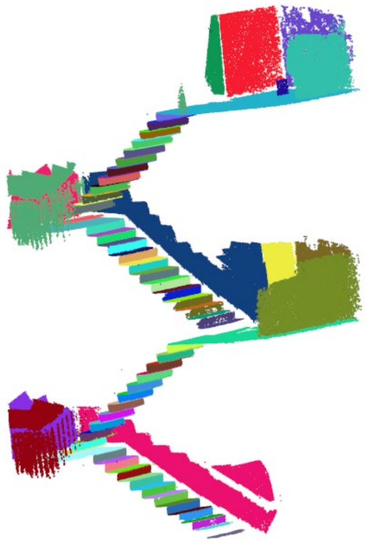
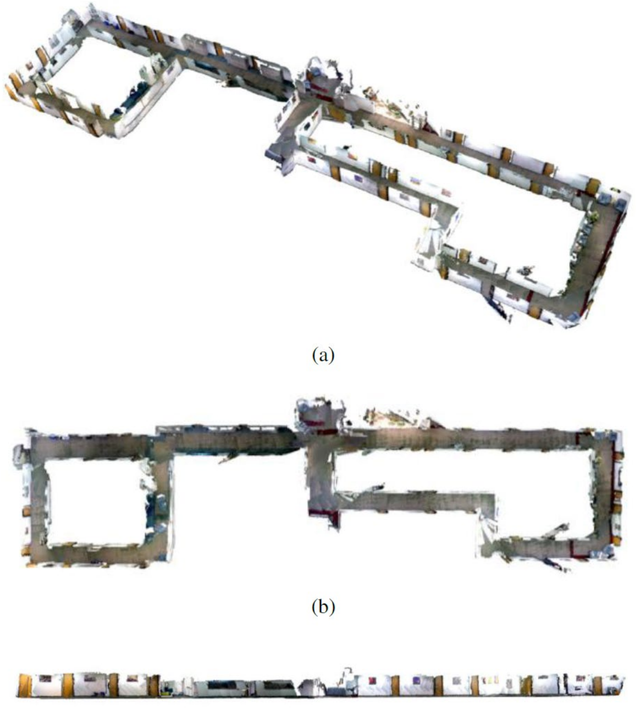


Fig. 3: The system structure of DPI-SLAM, which is similar to KDP-SLAM [12] but modified to allow fusing IMU measurements: (a) IMU preintegration, fast dense RGB-D odometry algorithm, and frame labeling process. (b) Selective local depth fusion algorithm. (c) Optimization of IMU states and planar landmarks in the global factor graph $\mathcal{G}$ with structural constraints and loop closing. Note that for the set of all keyframes $\mathcal{K}$, all reference frames $\mathcal{R}$, and all fusion frames $\mathcal{U}$, $\mathcal{K} \subset \mathcal{R} \subset \mathcal{U}$ holds. Also, the loop closing constraints are not shown here but in Fig. 5 for readability.





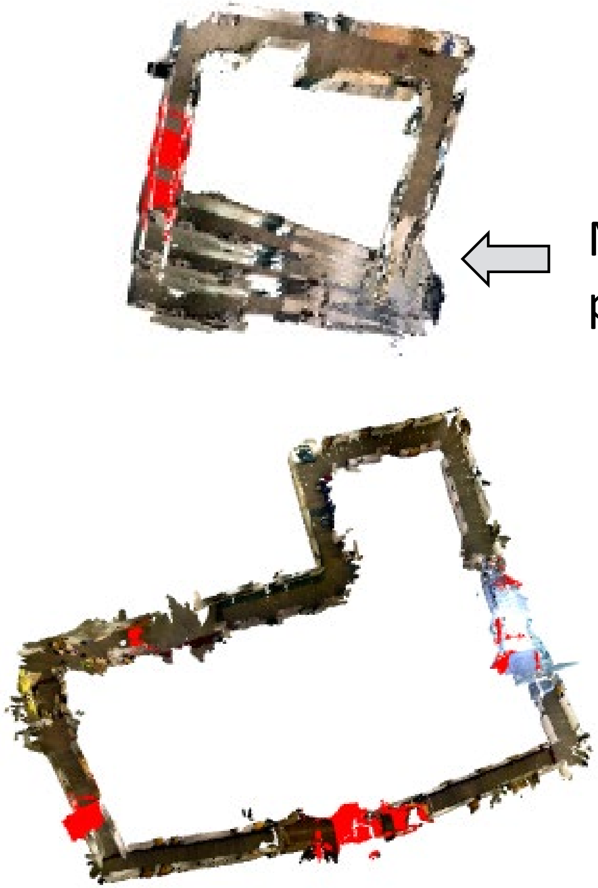
(a)

(b)

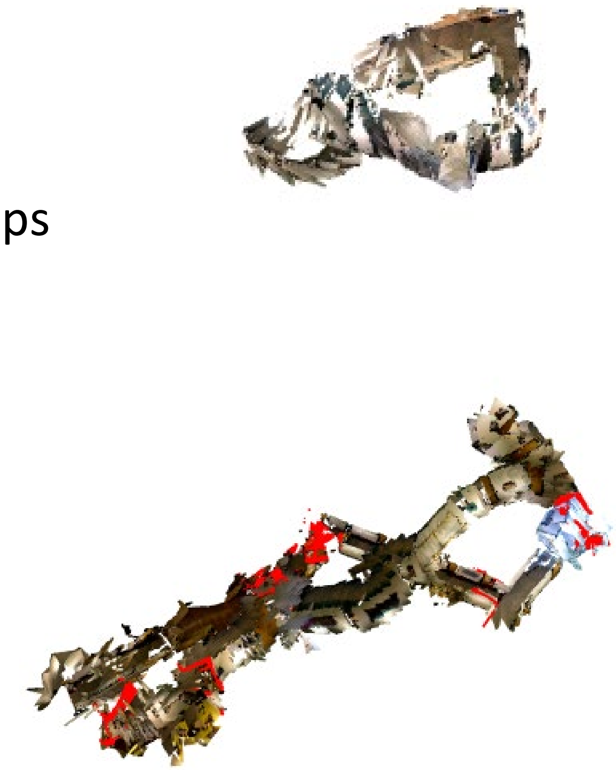
(c)

Hsiao and Kaess, ICRA 2019

Multi-hypothesis RGB-D Maps



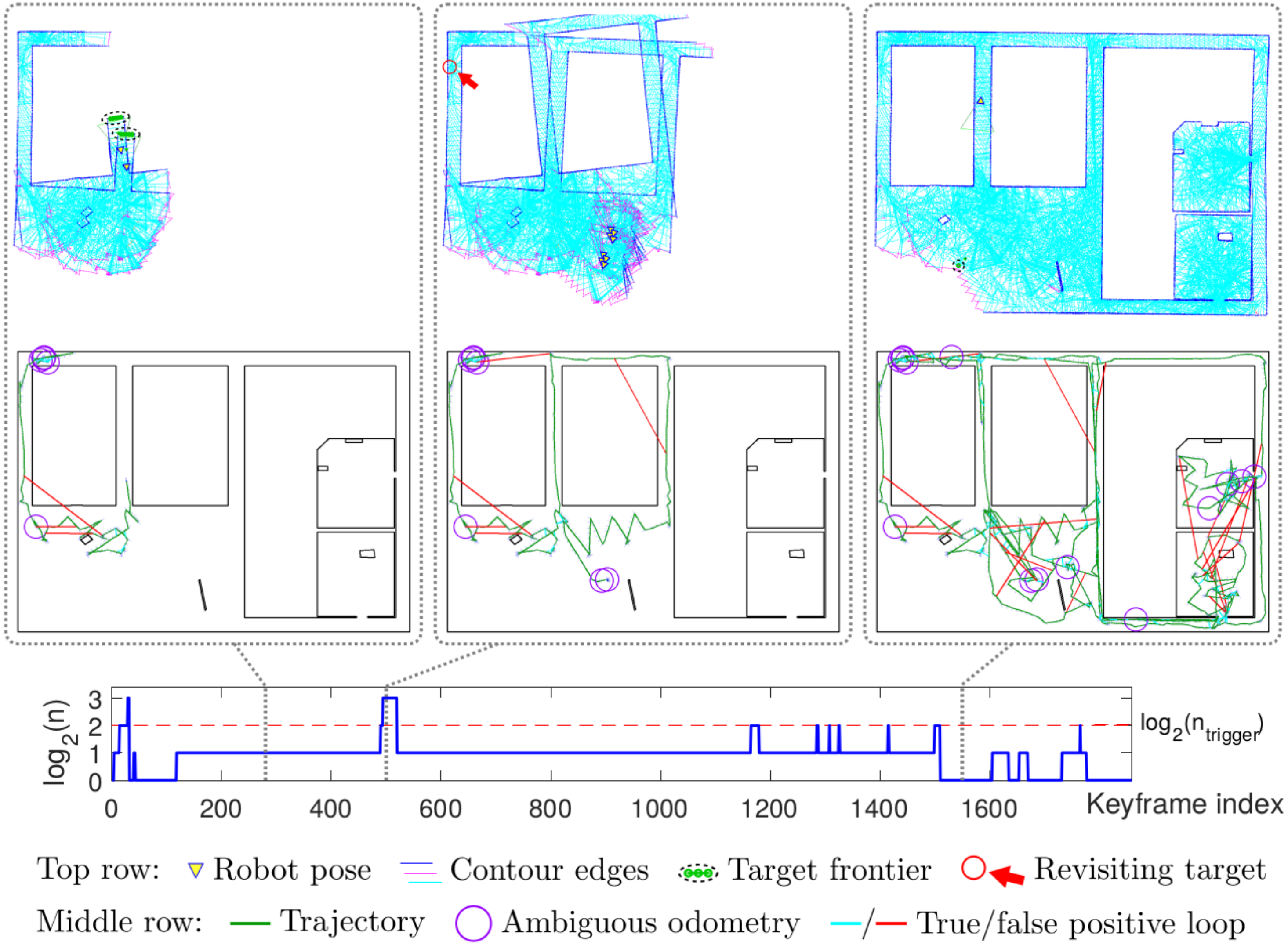
Multiple possible maps



Multi-hypothesis

Single hypothesis

Active Disambiguation



Hsiao, Mangelson, Suresh, Debrunner, Kaess, IROS 2020

Active Disambiguation