Robot Mapping

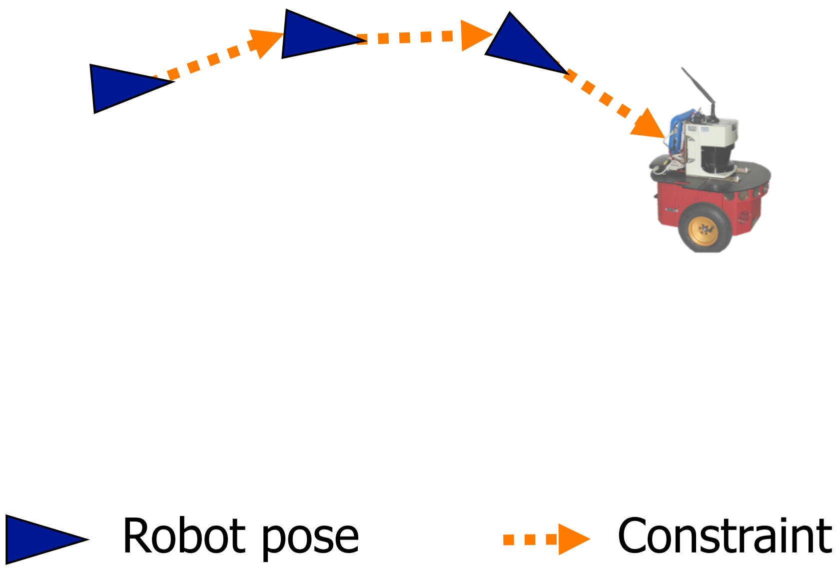
SLAM Front-Ends

Cyrill Stachniss

Partial image courtesy: Edwin Olson

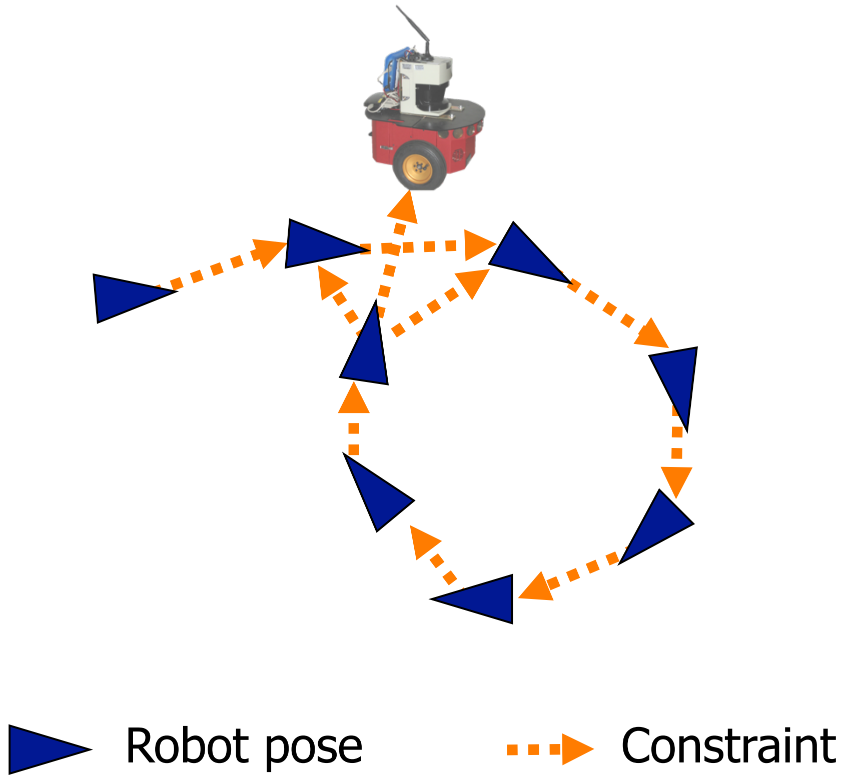
Graph-Based SLAM

- Constraints connect the nodes through odometry and observations

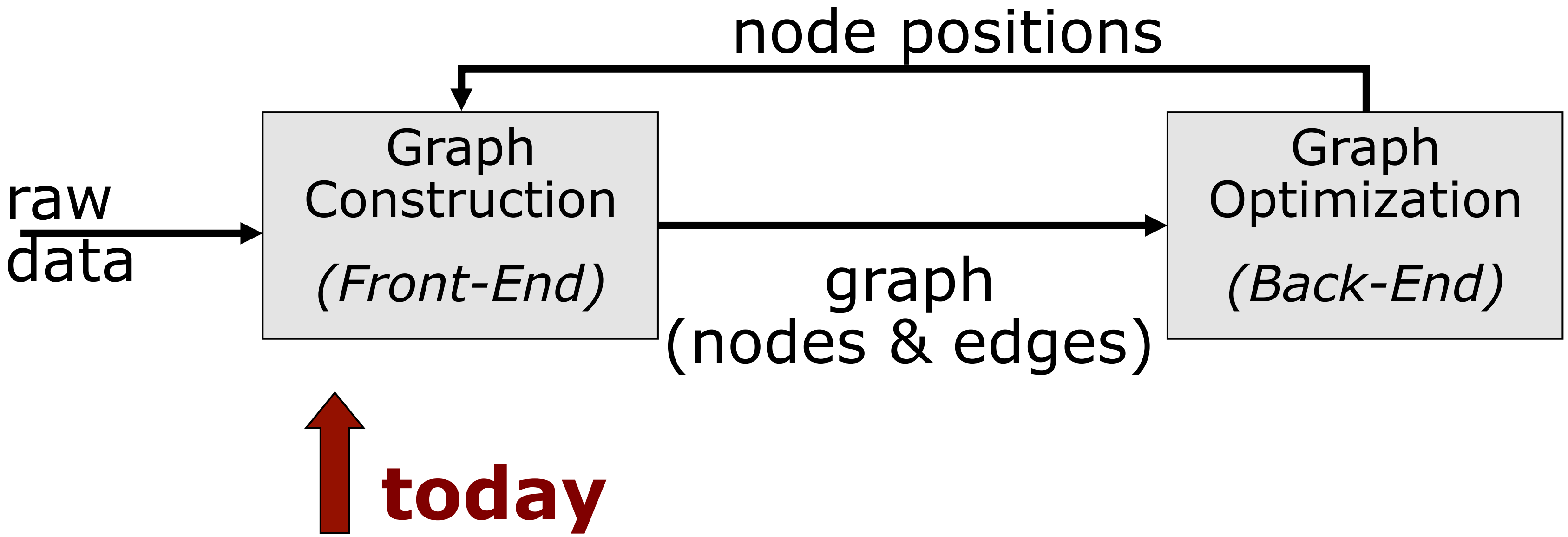


Graph-Based SLAM

- Constraints connect the nodes through odometry and observations
- How to obtain the constraints?



Interplay between Front-End and Back-End



Constraints From Matching

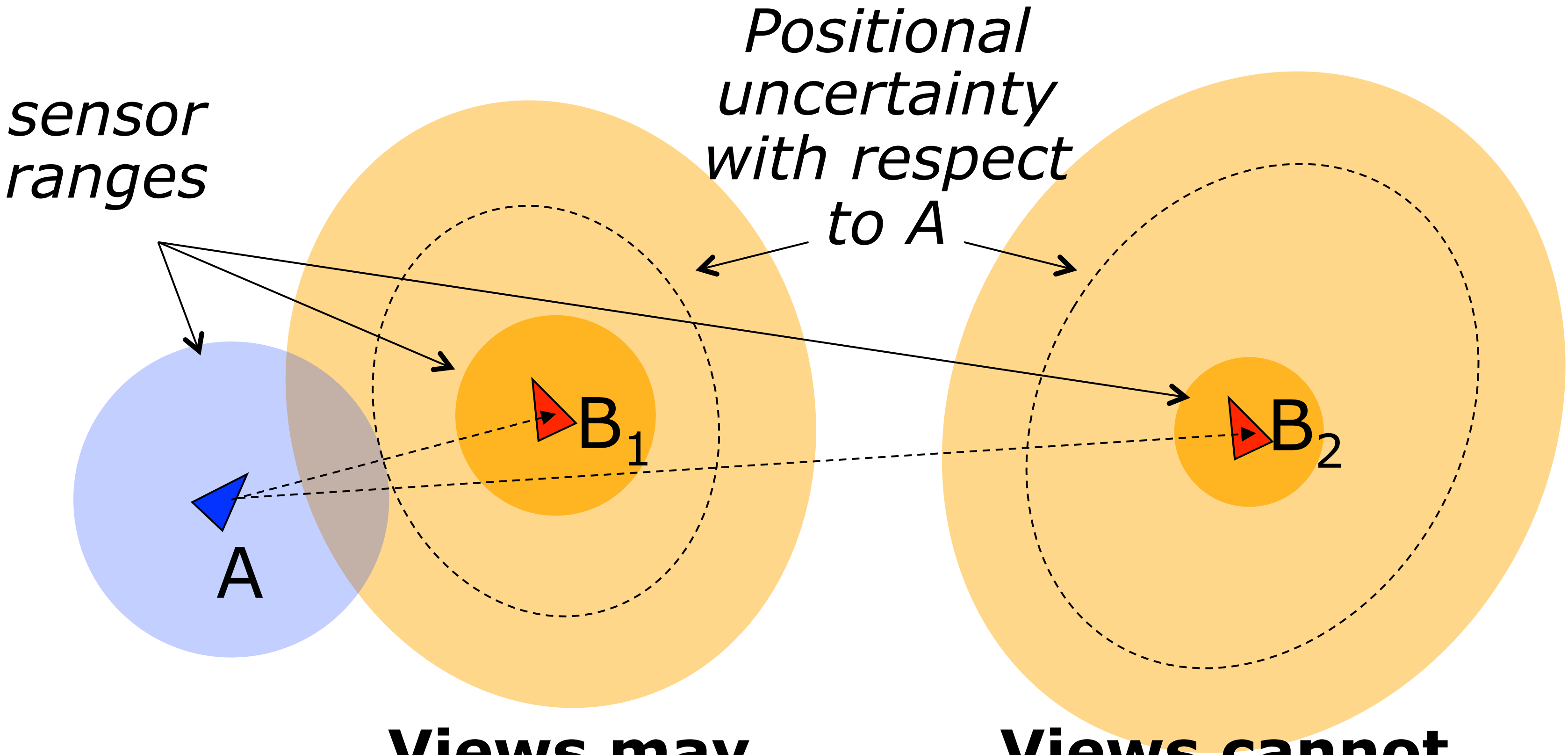
- Constraints can be obtained from matching observations

Popular approaches

- Dense scan-matching
- Feature-based matching
- Descriptor-based matching

Where to Search for Matches?

- Consider uncertainty of the nodes with respect to the current one



Views may overlap

Views cannot overlap

Note on the Uncertainty

- In graph-based SLAM, computing the uncertainty relative to A requires inverting the Hessian **H**
- Fast approximation by Dijkstra expansion (“propagate uncertainty along the shortest path in the graph”)
- Conservative estimate

Simple ICP-Based Approach

- Estimate uncertainty of nodes relative to the current pose
- Sample poses in relevant area
- Apply Iterative Closest Point algorithm
- Evaluate match
- Accept match based on a threshold

Problems?

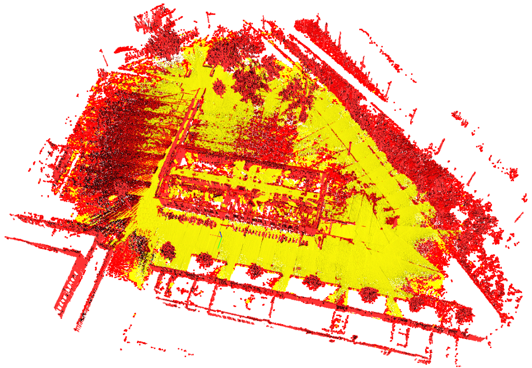
Problems

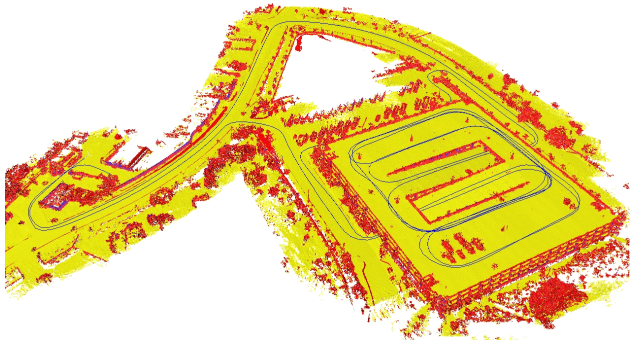
- ICP is sensitive to the initial guess
- Inefficient sampling
- Ambiguities in the environment

Problems

- **ICP is sensitive to the initial guess**
- **Inefficient sampling**
- Ambiguities in the environment

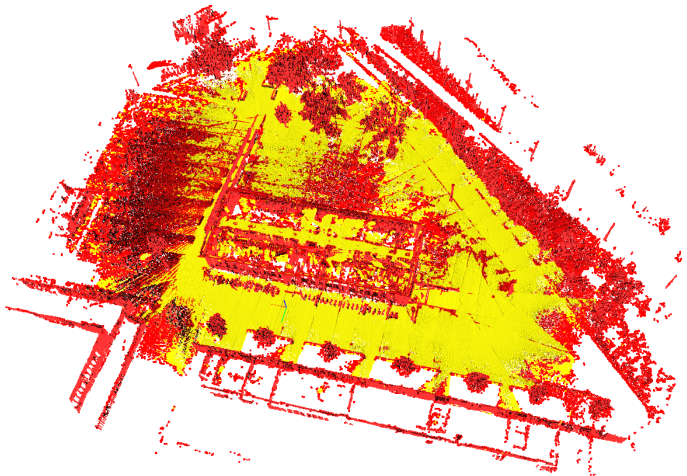
Examples



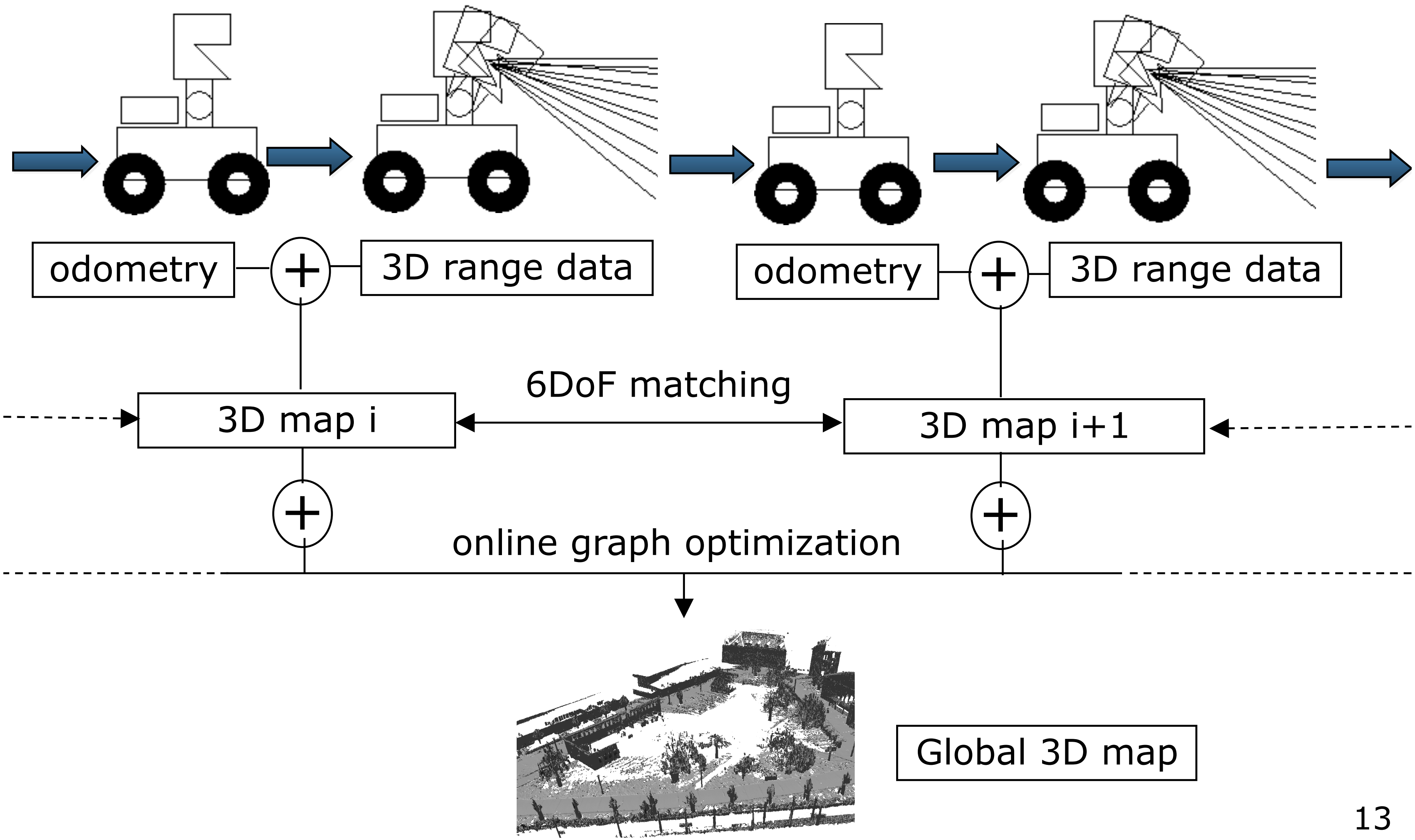


Learning 3D Maps with Laser Data

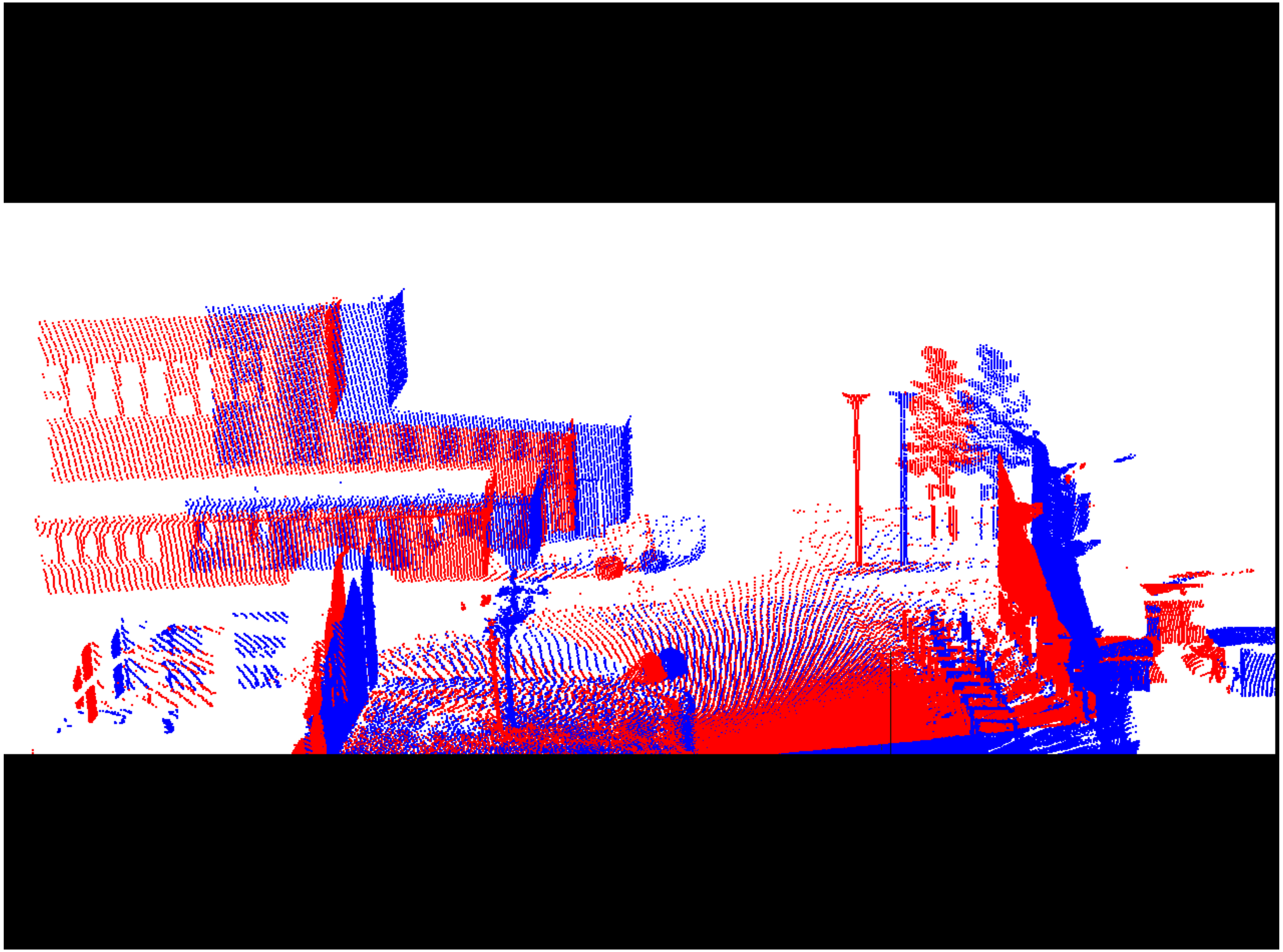
- Robot that provides odometry
- Laser range scanner on a pan-tilt-unit



Incremental 6D SLAM



Aligning Consecutive Maps



Aligning Consecutive Maps

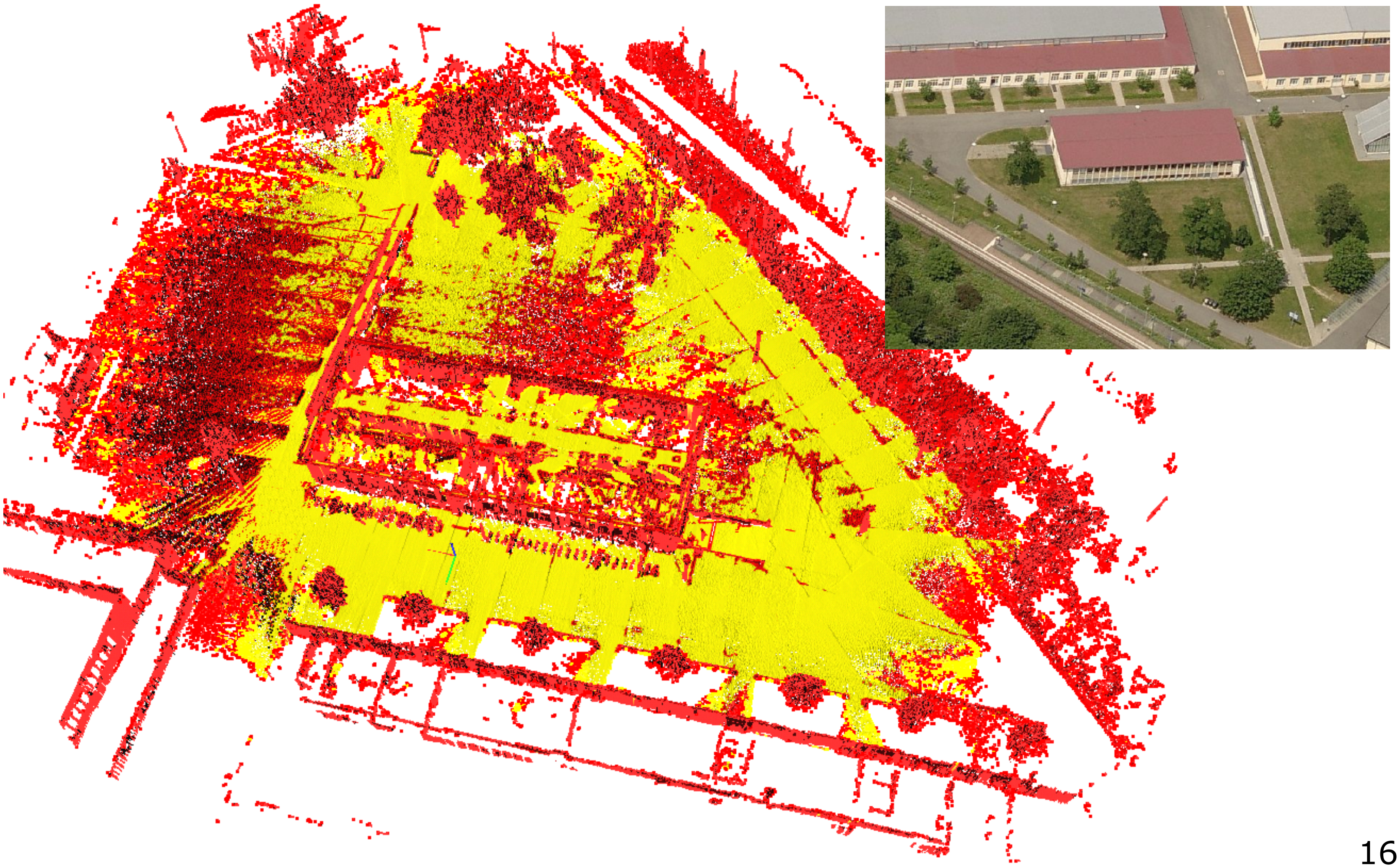
- Let $\mathbf{u}_{i_c}$ and $\mathbf{u}'_{j_c}$ be corresponding points
- Find the parameters R and t which minimize the sum of the squared error
- ICP

$$e(R,\mathbf{t}) = \sum_{c=1}^{C} d(\mathbf{u}_{i_c}, \mathbf{u}'_{j_c})$$

- ICP with additional knowledge

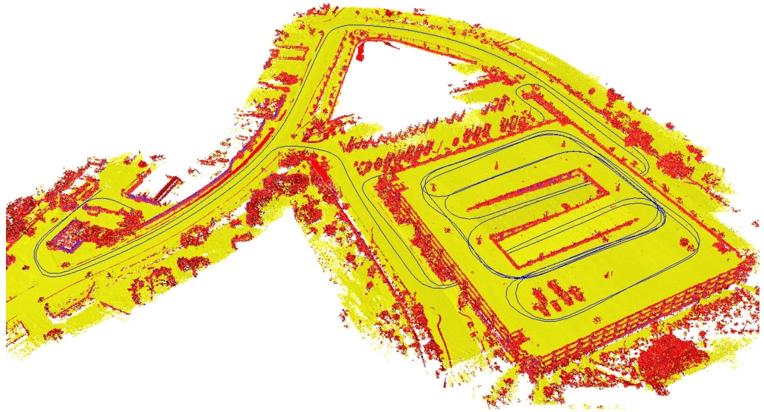
$$e(R,\mathbf{t}) = \underbrace{\sum_{c=1}^{C_1} d_v(\mathbf{u}_{i_c}, \mathbf{u}'_{j_c})}_{\text{vertical objects}} + \underbrace{\sum_{c=1}^{C_2} d(\mathbf{v}_{i_c}, \mathbf{v}'_{j_c})}_{\text{traversable}} + \underbrace{\sum_{c=1}^{C_3} d(\mathbf{w}_{i_c}, \mathbf{w}'_{j_c})}_{\text{non-traversable}}$$

Online Estimated 3D Map



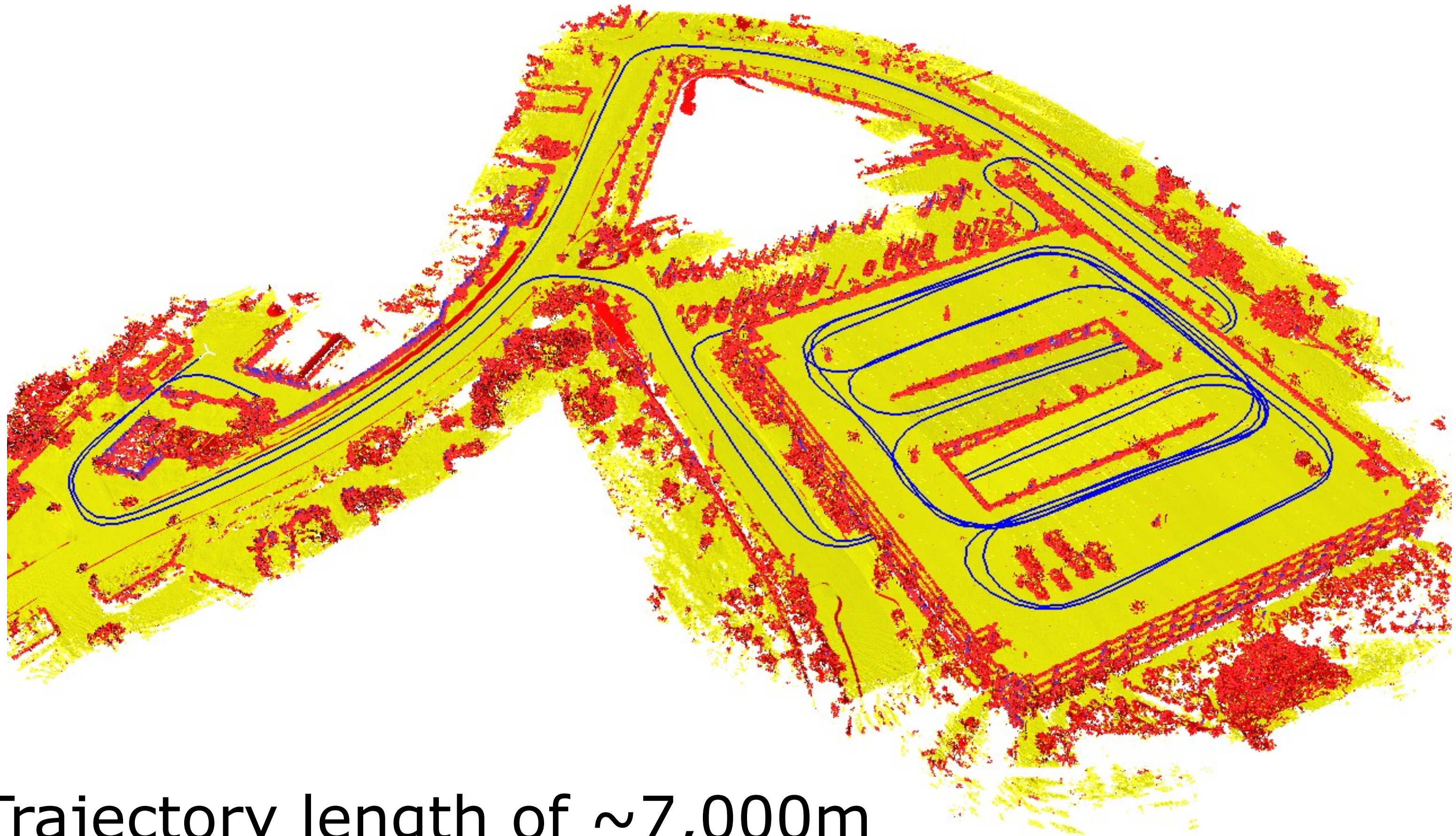
Mapping with a Robotic Car

- 3D laser range scanner (Velodyne)
- Use map for autonomous driving



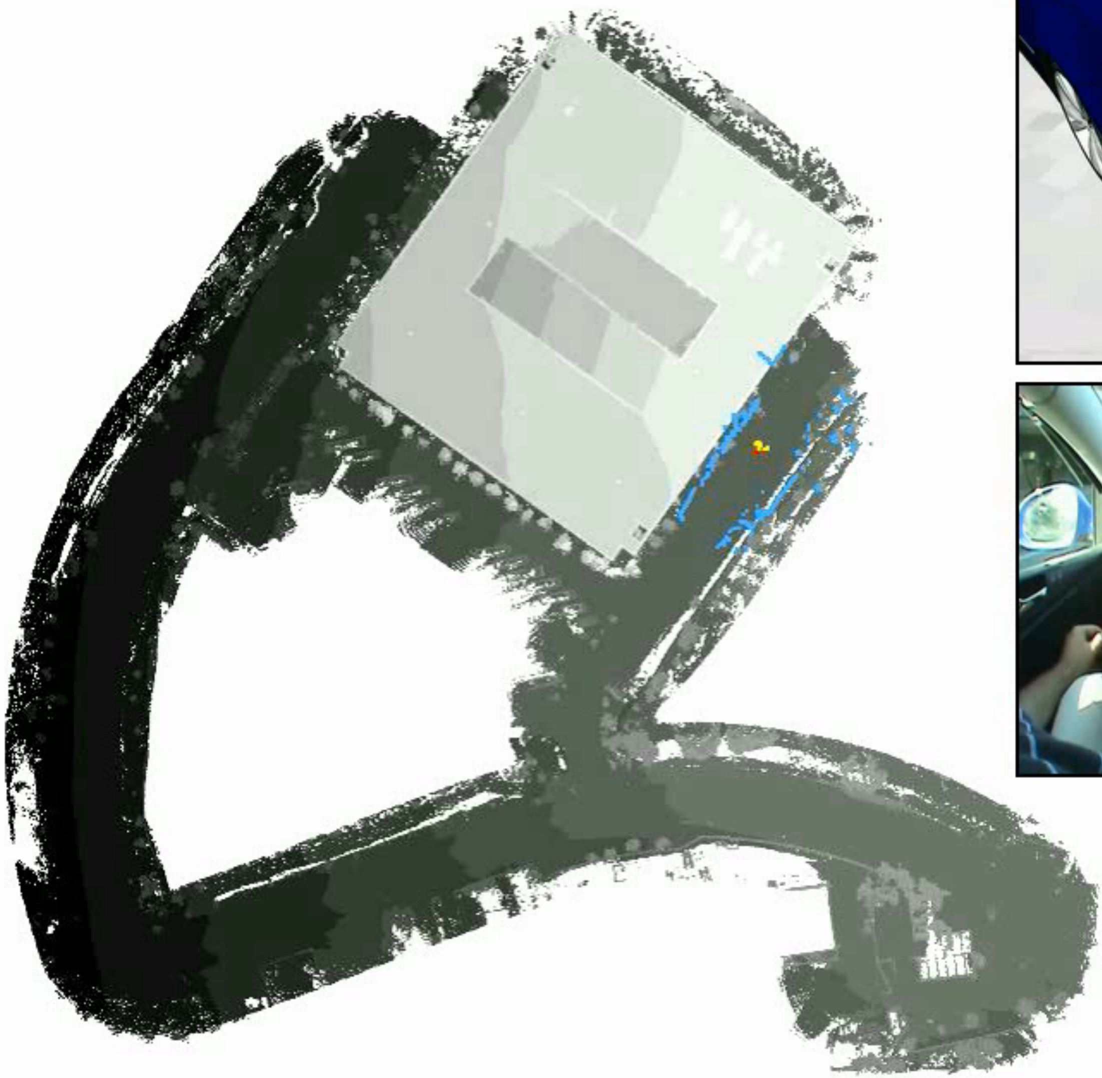
Parking Garage

Resulting Map



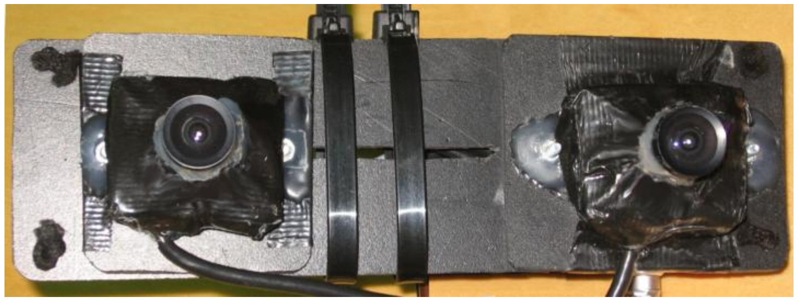
- Trajectory length of ~7,000m
- 1661 local 3D maps, cell size of 20cm x 20cm

Map-based Autonomous Parking



Mapping with Arial Vehicles

- Flying vehicles equipped with cameras and an IMU

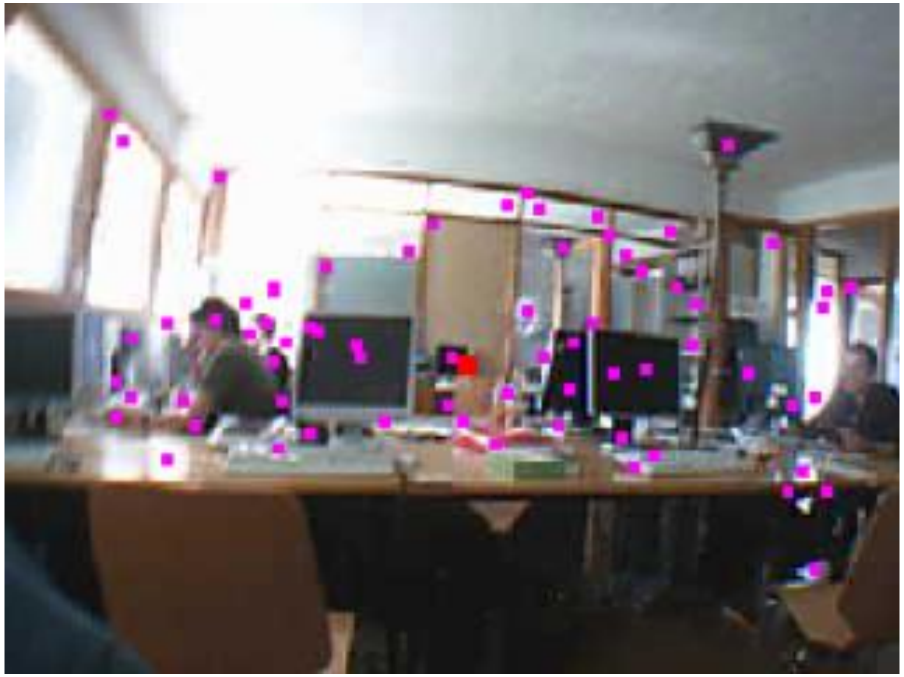


Examples of Camera Images




SURF Features

- Provide a description vector and an orientation
- Descriptor is invariant to rotation and scale



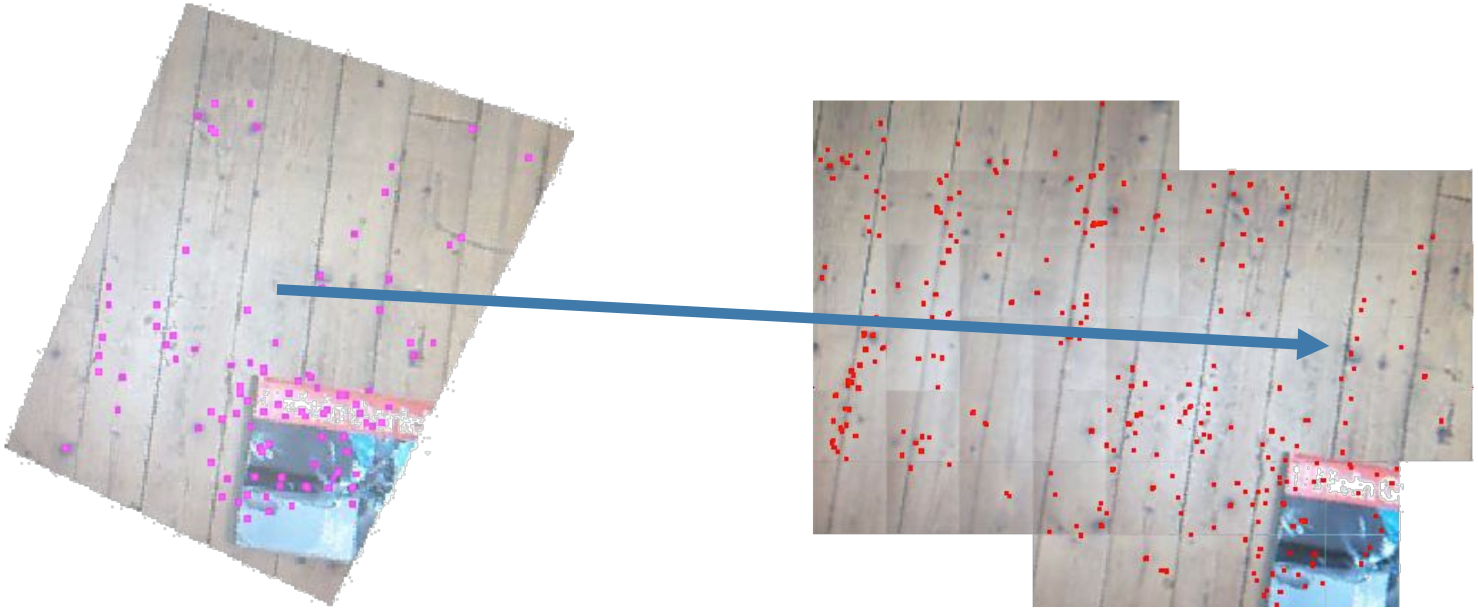
Determining the Camera Pose

Wanted: x, y, z, φ, θ, Ψ (roll, pitch, yaw)

- IMU determines roll and pitch accurately
- x, y, z and the heading (yaw) have to be calculated based on the camera images

➡ 3D positions of **two** image features is sufficient to determine the camera pose

Feature Matching for Pose Estimation



features in image   features in map

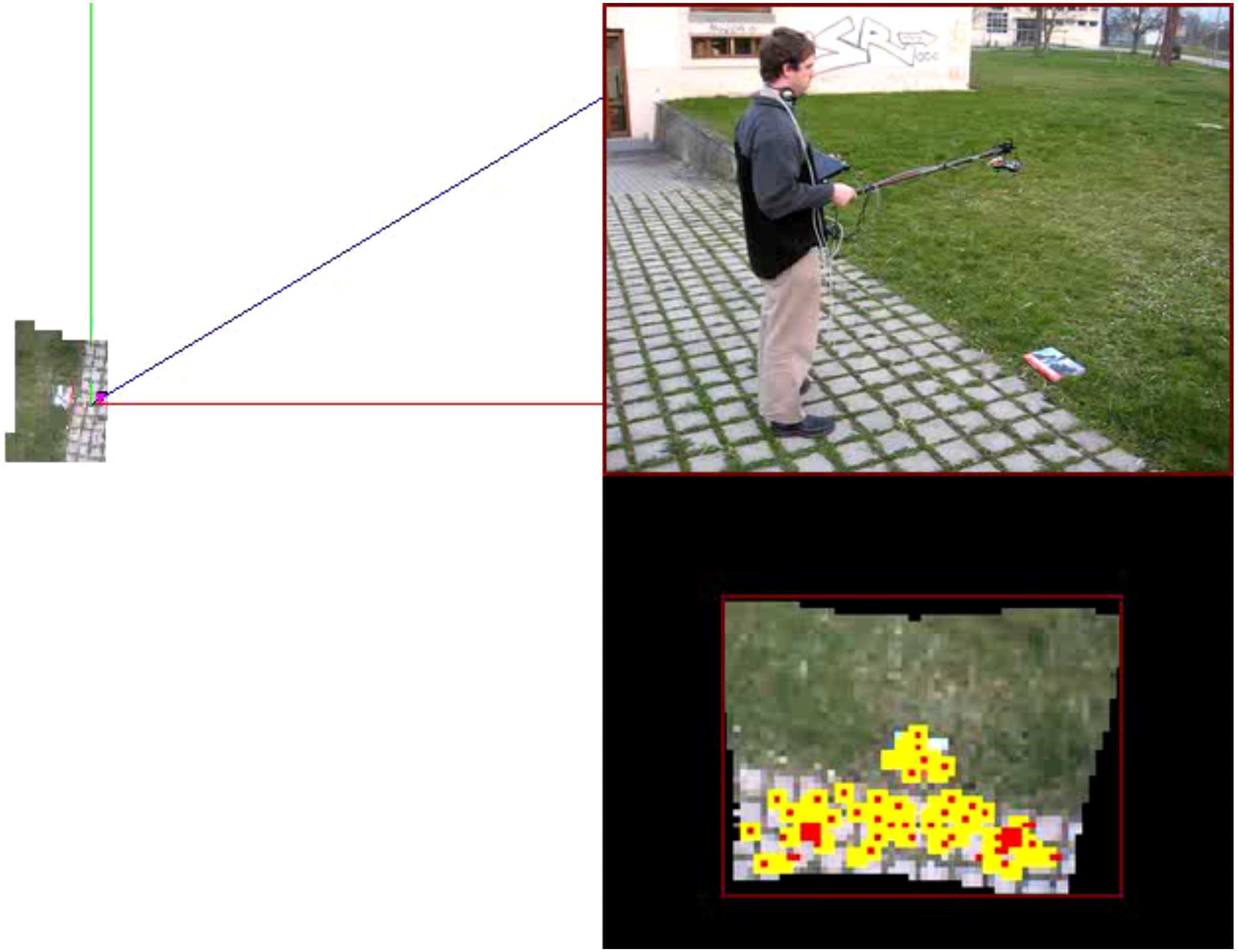
Camera Pose Estimation

1. Find possible matches (kd-tree)
2. Order matches by descriptor distance
   - Use two matches to calculate the camera position, start with the best one
   - Re-project all features accordingly to get a quality value about this pose
   - Repeat until satisfactory pose is found
3. Update map

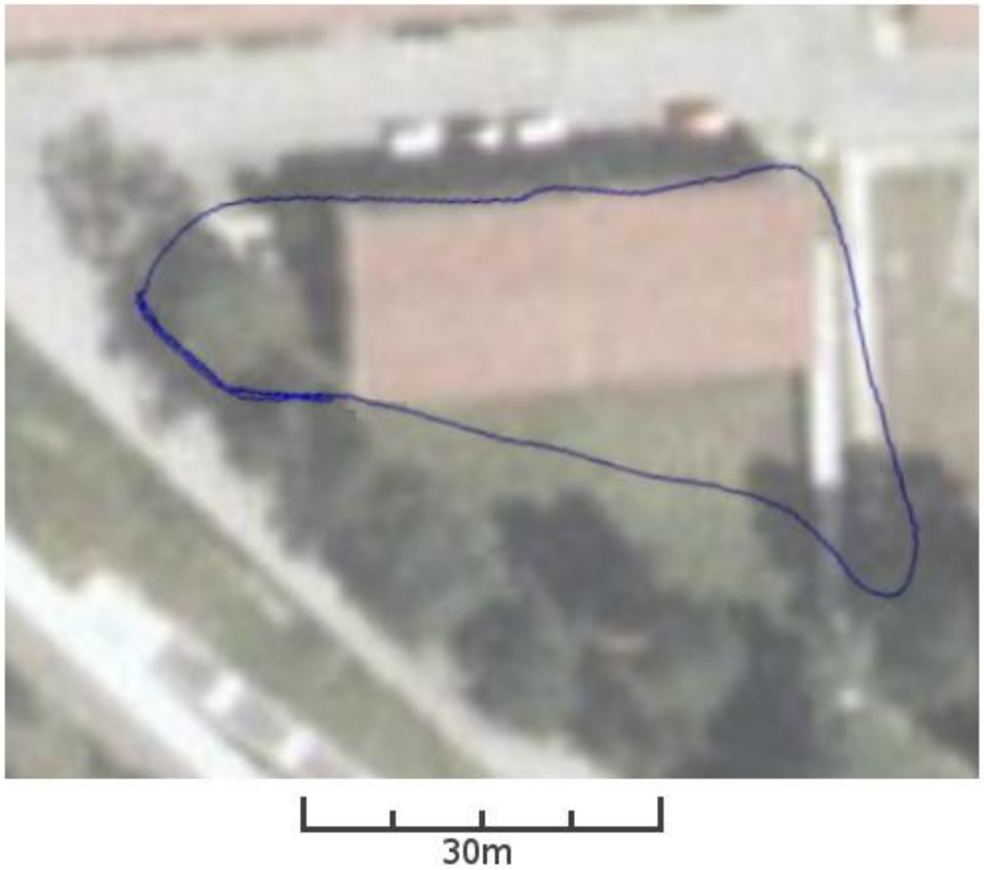
Finding Edges

- **Visual odometry:** Match features against the N previously observed ones
- **Localization:** Match against features in the map in a given region around the odometry estimate (local search)
- **Loop closing:** Match a subset of the features against all map features. Match leads to a localization step

Outdoor Example

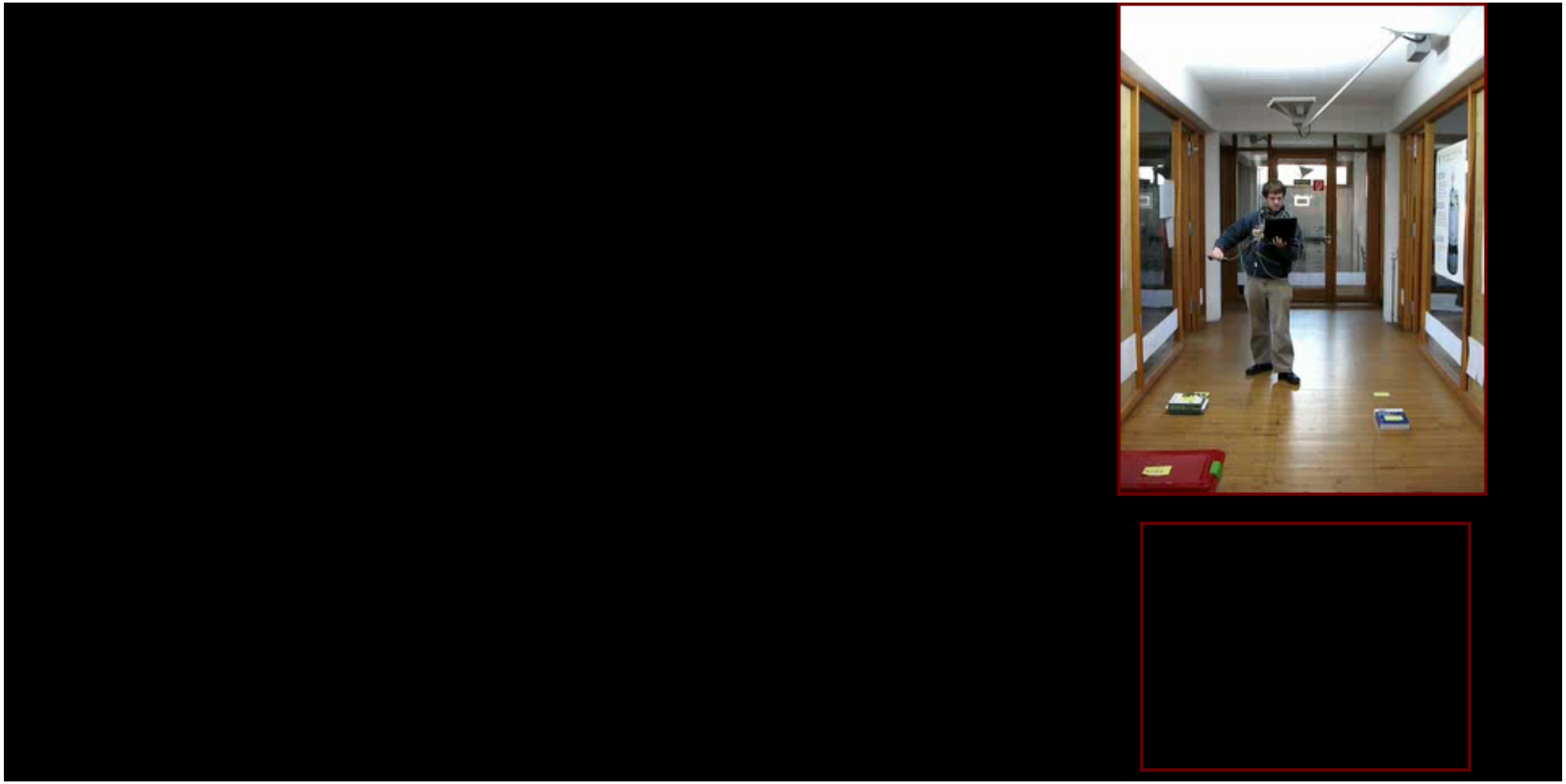


Resulting Trajectory

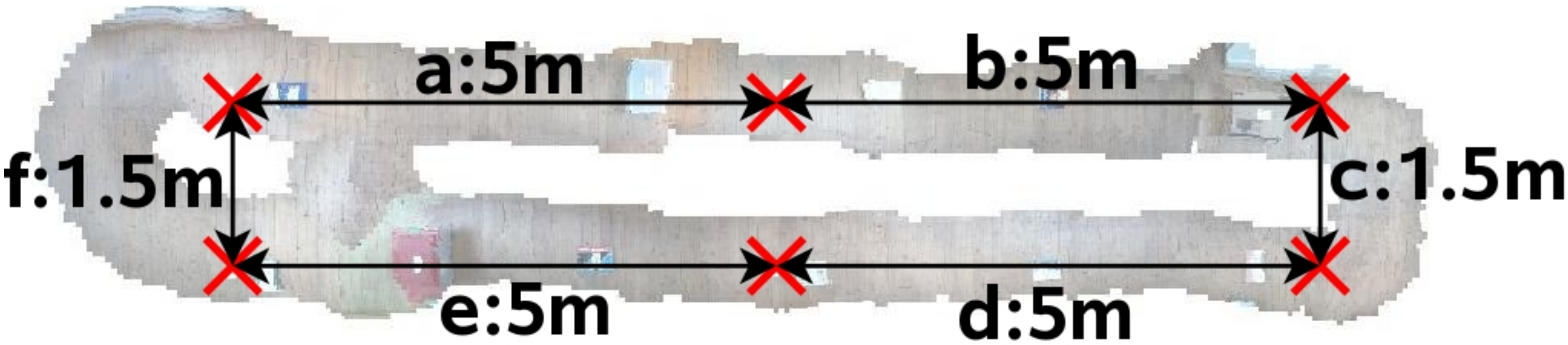


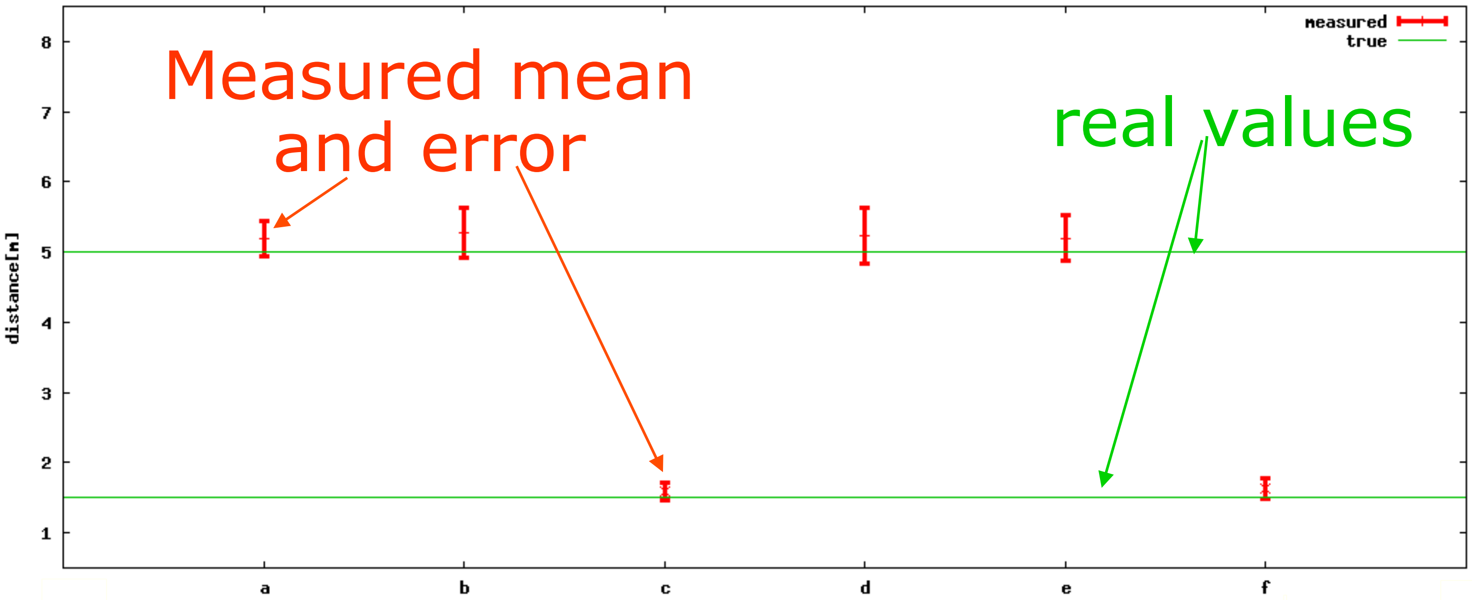
- Length (Google Earth): 188m
- Estimated length: 208m

Indoor Example

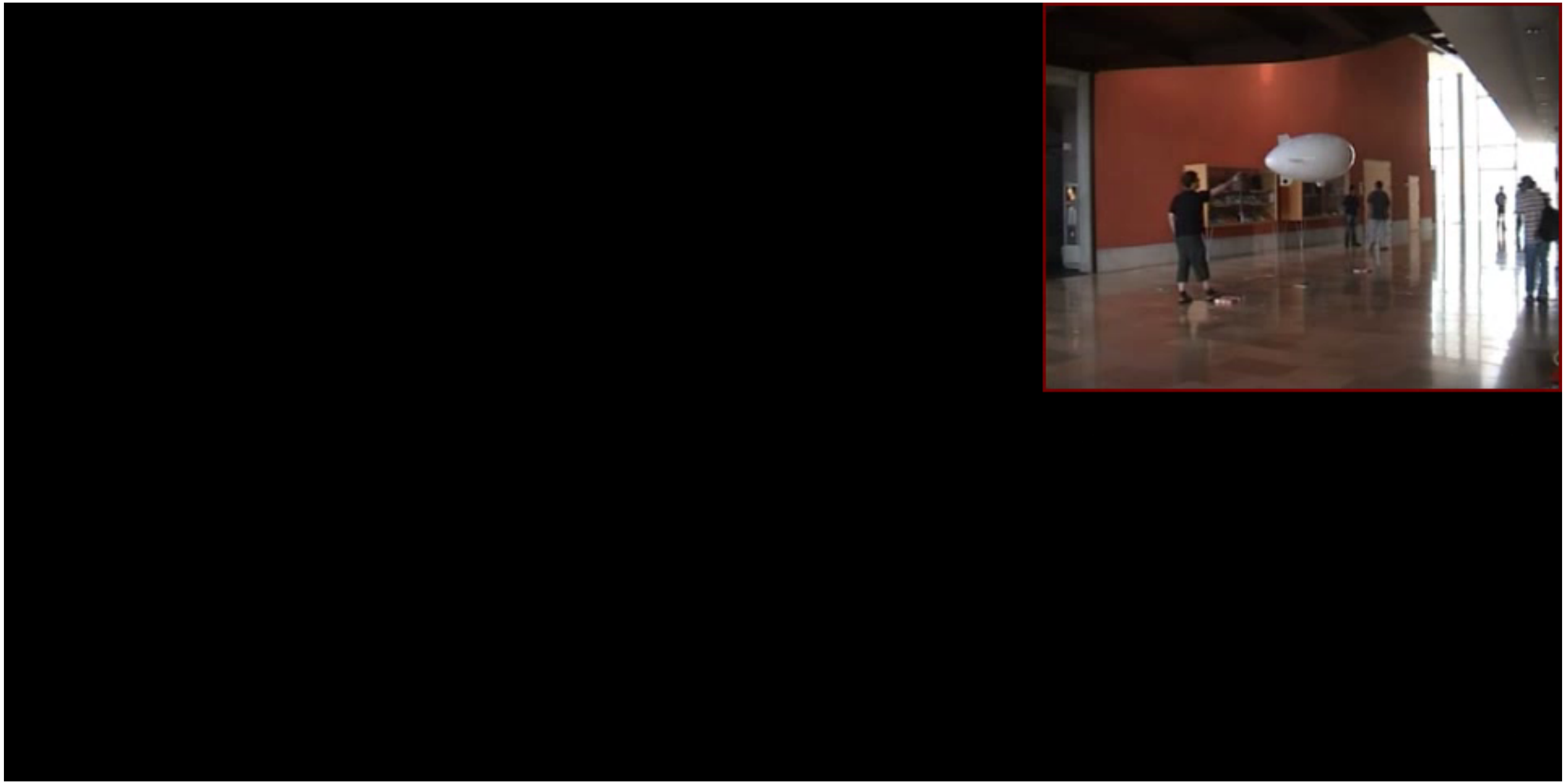


Ground Truth





System on a Blimp



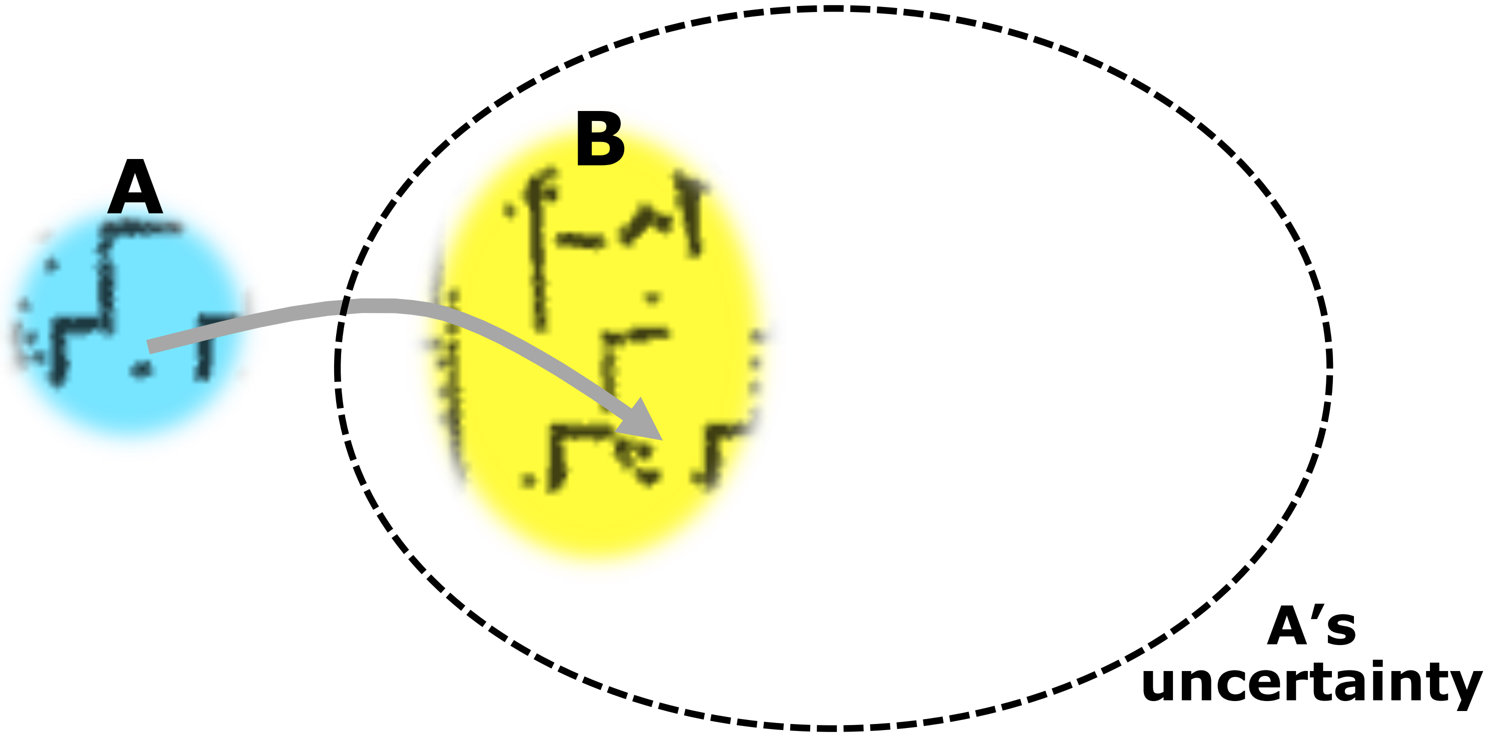
Problems

- ICP is sensitive to the initial guess
- Inefficient sampling
- **Ambiguities in the environment**

- Dealing with ambiguous areas in an environment is essential for robustly operating robots

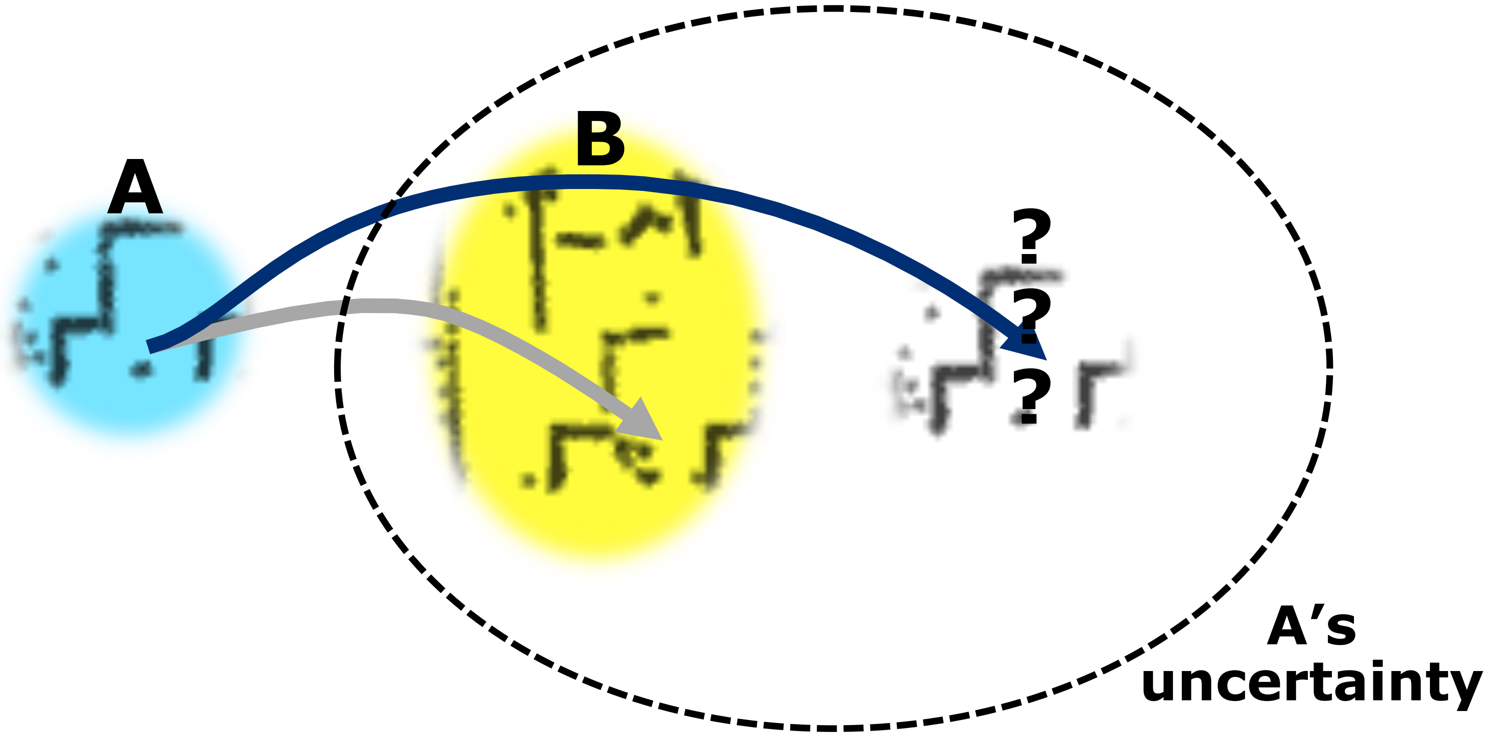
Ambiguities - Global Ambiguity

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



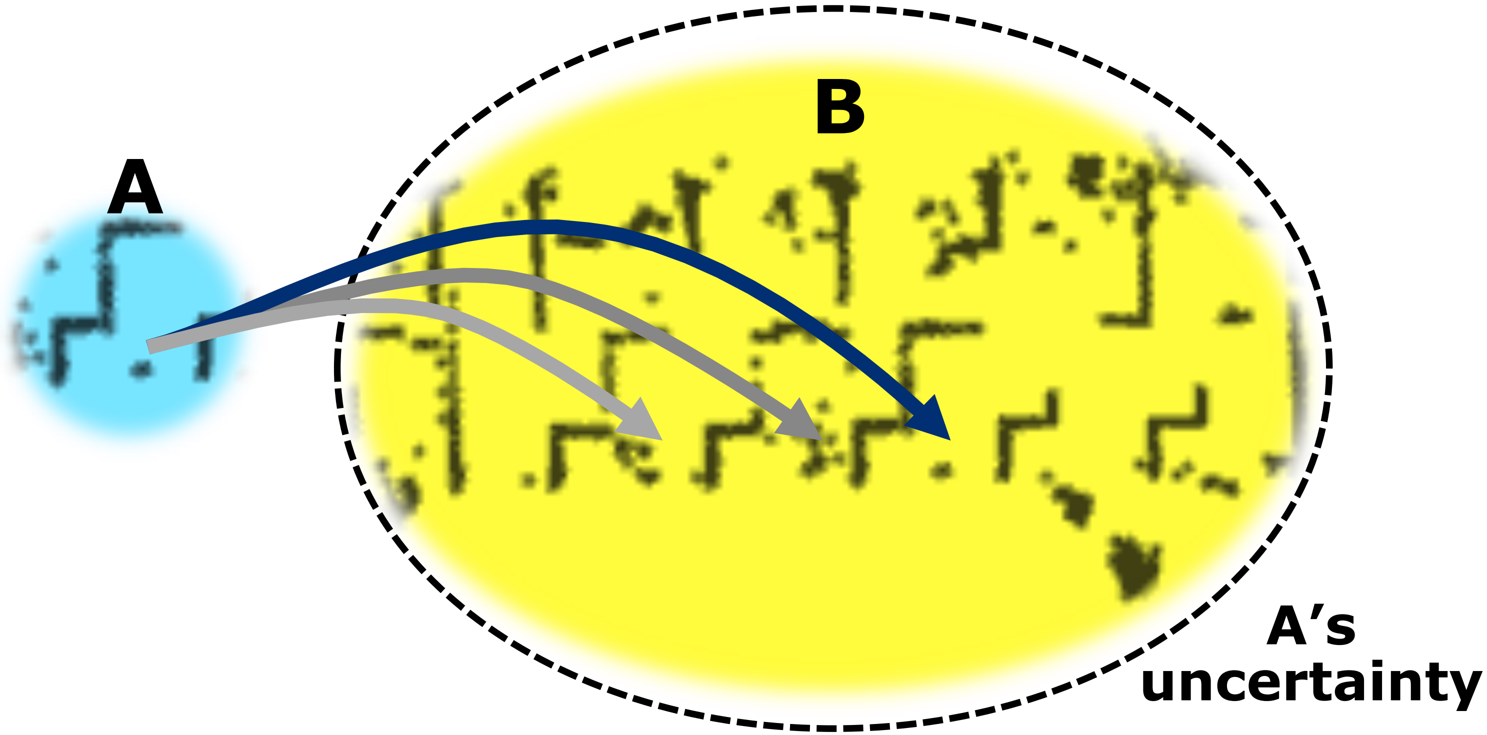
Ambiguities - Global Ambiguity

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



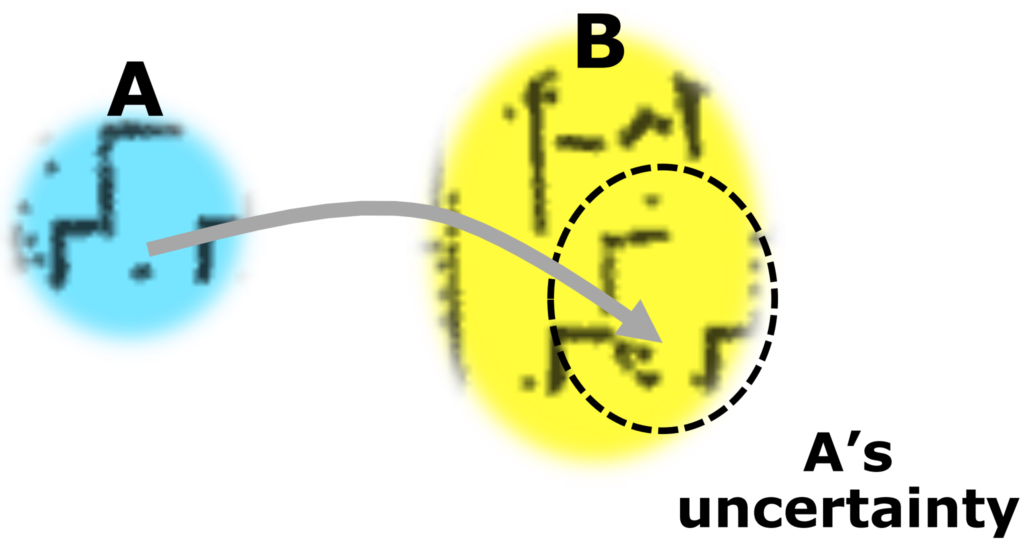
Ambiguities - Global Ambiguity

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



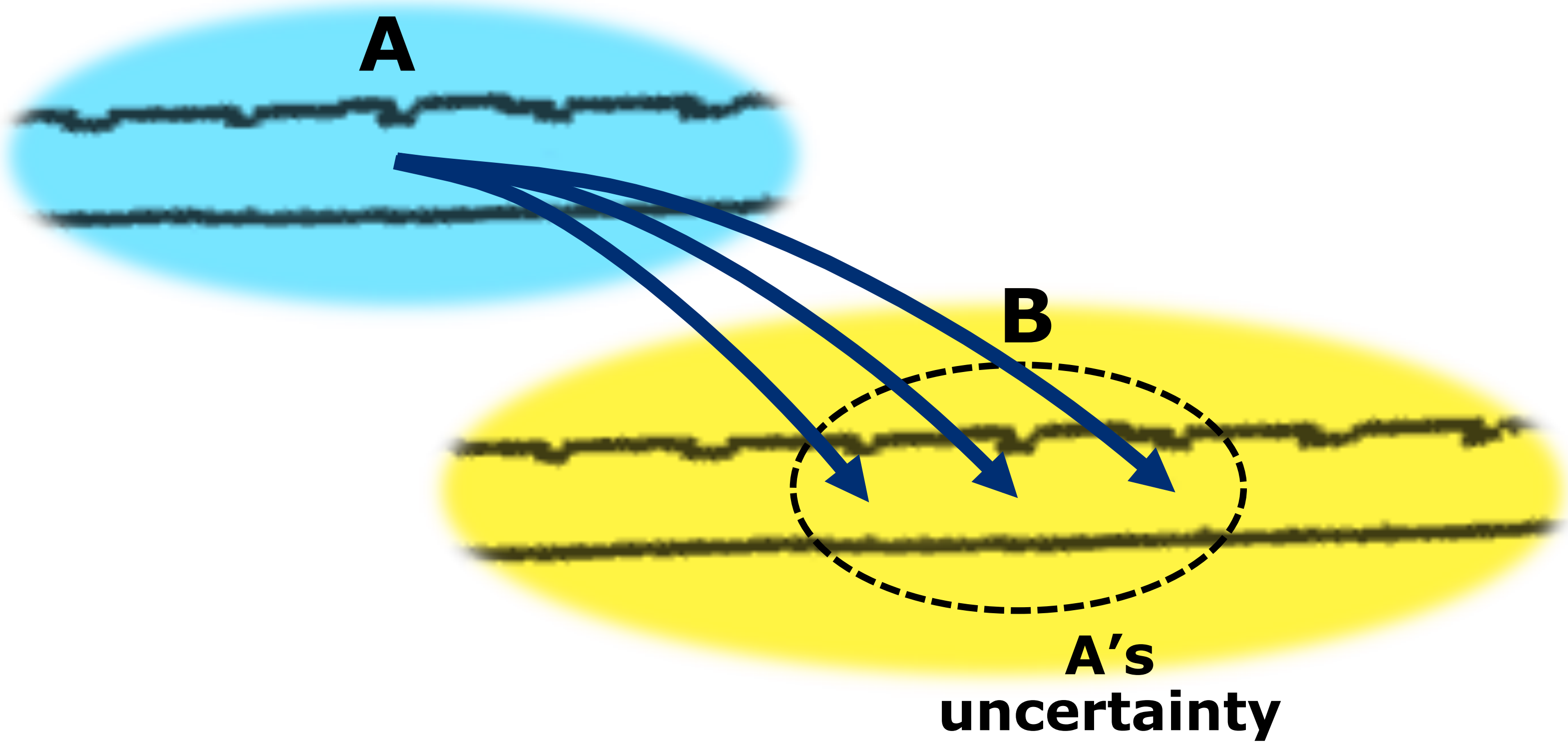
Ambiguities - Global Sufficiency

- B is inside the uncertainty ellipse of A
- The is no other possibility for a match



Ambiguities - Local Ambiguity

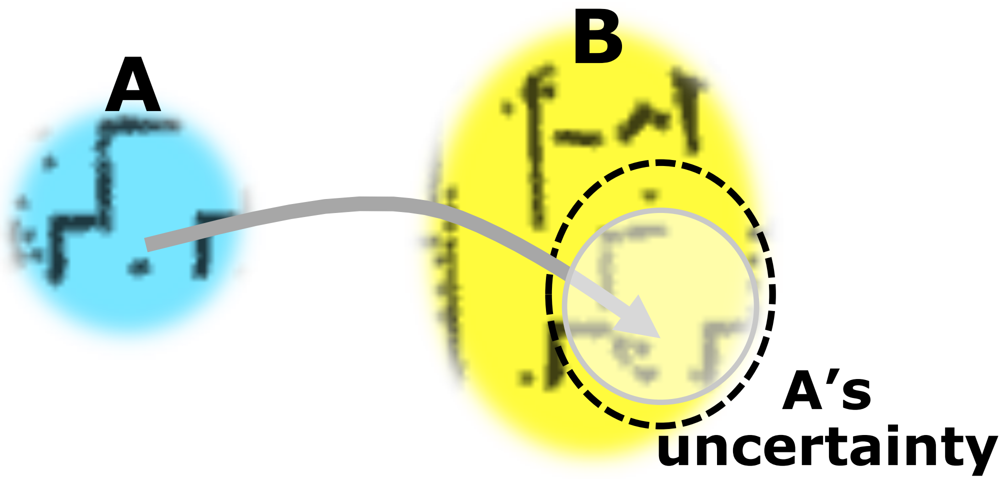
- “Picket Fence Problem”: largely overlapping local matches



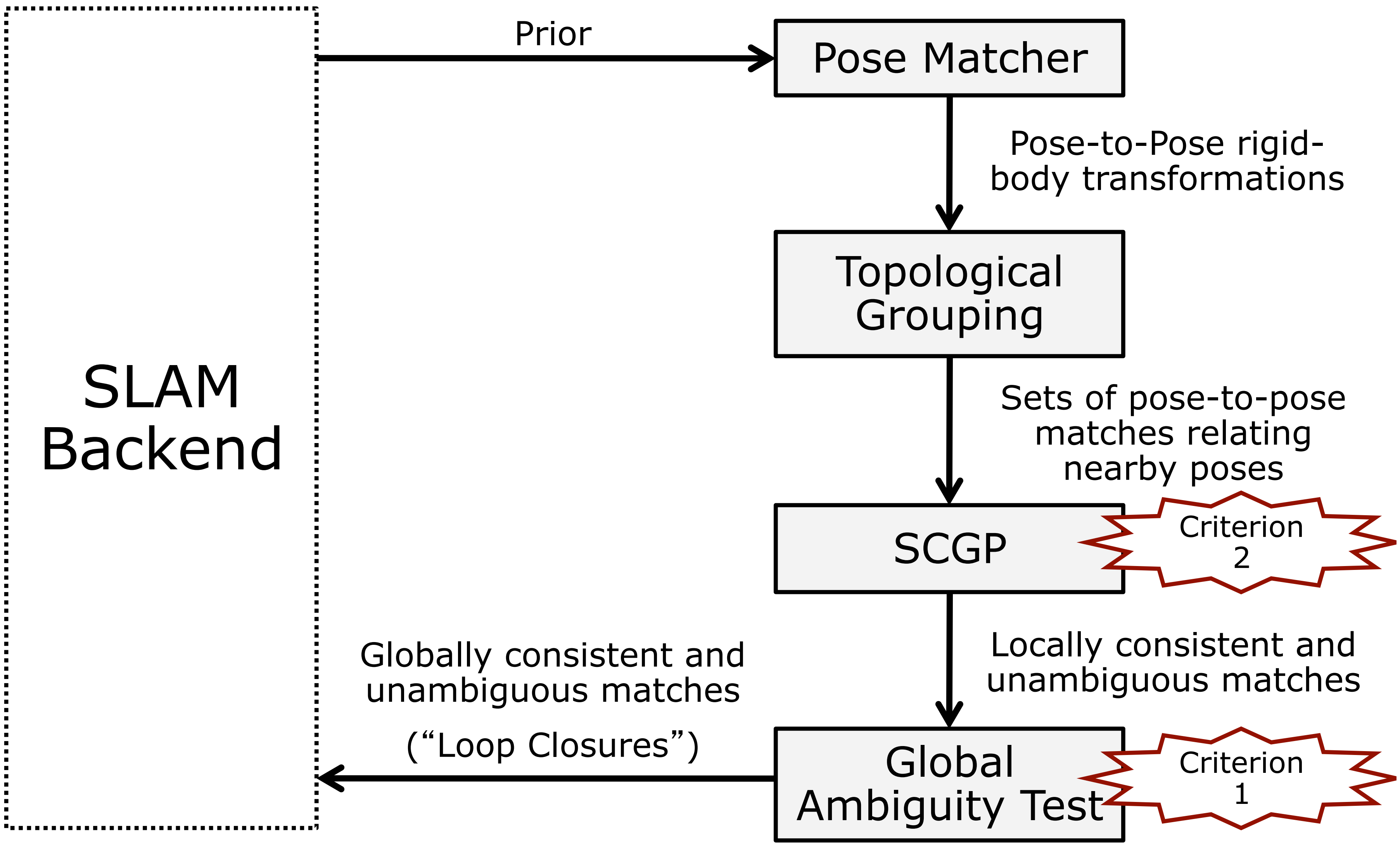
Global Match Criteria

1. Global Sufficiency: There is no disjoint match ("A is not somewhere else entirely")
2. Local unambiguity: There are no overlapping matches ("A is either here or somewhere else entirely")

Both need to be satisfied for a match

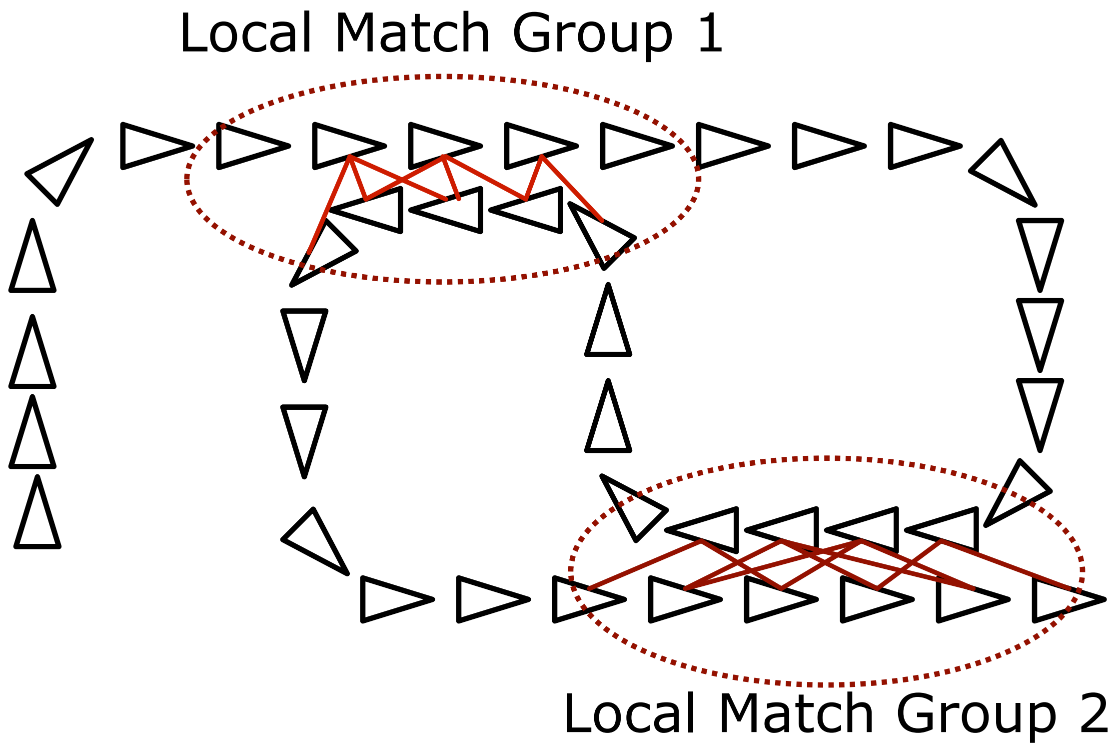


Olson's Proposal



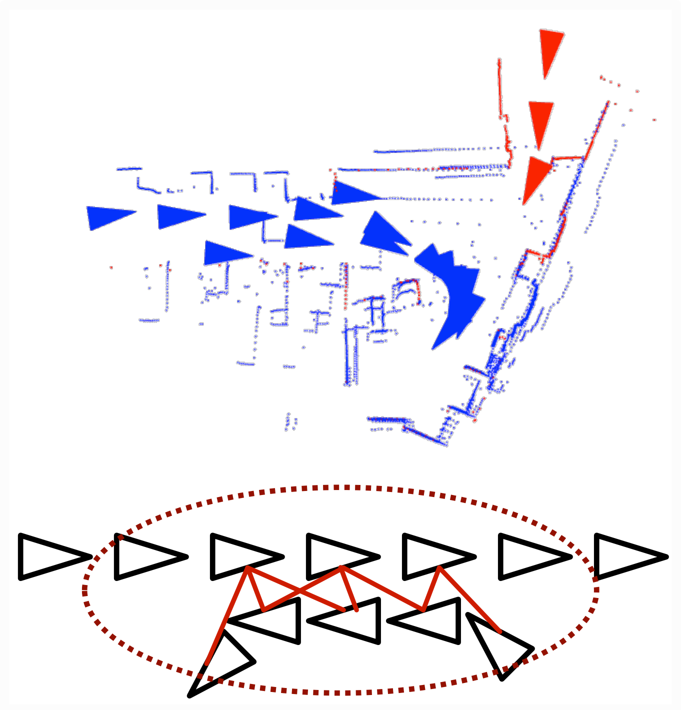
Topological Grouping

- Group together topologically-related pose-to-pose matches to form local matches
- Each group asks a "topological" question: Do two local maps match?

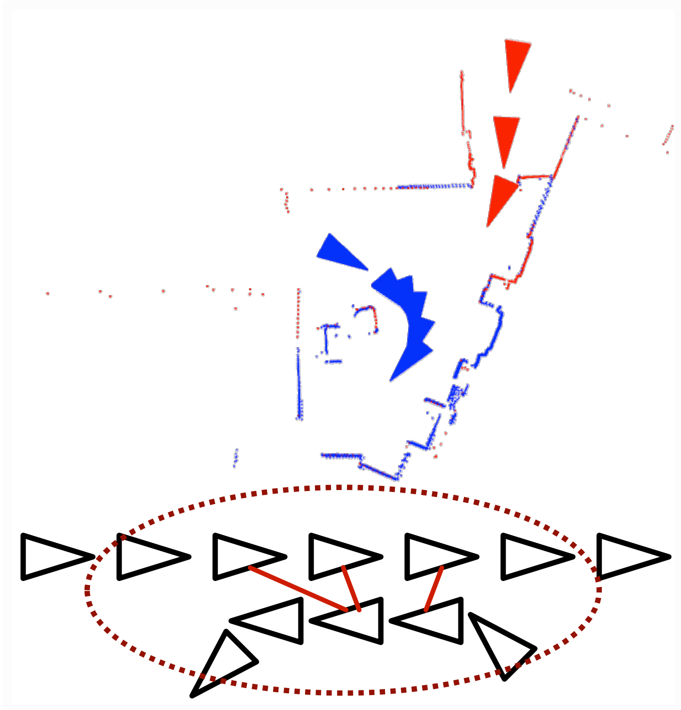


Locally Unambiguous Matches

Goal:



Unfiltered Local Match
(set of pose-to-pose matches)



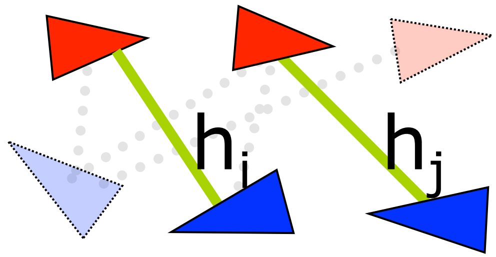
Locally consistent and
unambiguous local match
(set of pose-to-pose matches)

Locally Consistent Matches

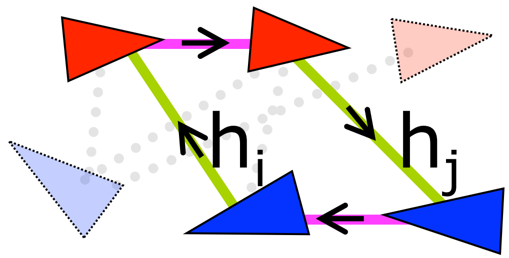
- Correct pose-to-pose hypotheses must agree with each other
- Incorrect pose-to-pose hypotheses tend to disagree with each other
- Find subset of self-consistent of hypotheses
- Multiple self-consistent subsets, are an indicator for a “picket fence”!

Do Two Hypotheses Agree?

- Consider two hypotheses **i** and **j** in the set:



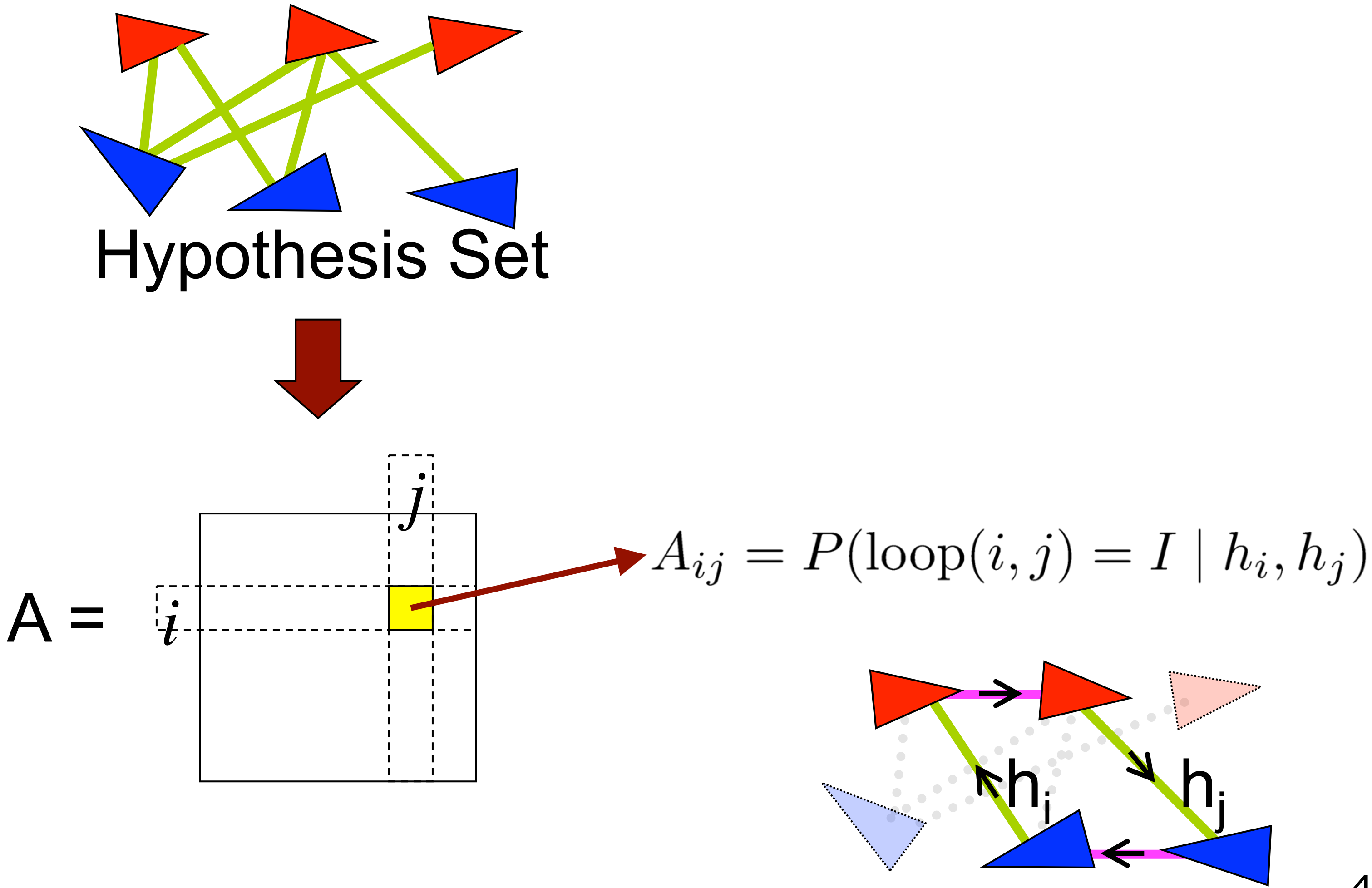
- Form a loop using edges from the prior graph



Rigid-body transformation around the loop should be the identity matrix

Idea of Olson's Method

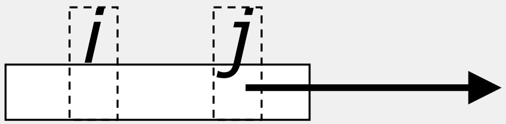
- Form pair-wise consistency matrix **A**



Single Cluster Graph Partitioning

- Idea: Identify the subset of consistent hypotheses
- Find the best **indicator vector** (represents a subset of the hypotheses)

Indicator vector v



v_i = 1 if h_i is correct,
0 if h_i is incorrect

Single Cluster Graph Partitioning

- Identify the subset of hypotheses that is maximally self-consistent
- Which subset **v** has the **greatest average pair-wise consistency** λ?

$$\lambda = \frac{\mathbf{v}^{\mathbf{T}}\mathbf{A}\mathbf{v}}{\mathbf{v}^{\mathbf{T}}\mathbf{v}}$$

Sum of all pair-wise consistencies between hypotheses in v

Number of hypotheses in v

Gallo et al 1989

- Densest Subgraph Problem

Consistent Local Matches

- We want find **v** that maximizes λ(**v**)

$$\lambda(\mathbf{v}) = \frac{\mathbf{v^T A v}}{\mathbf{v^T v}}$$

- Treat as continuous problem
- Derive and set to zero

$$\frac{\partial \lambda(\mathbf{v})}{\partial \mathbf{v}} = 0$$

- Which leads to (for symmetric A)

$$\frac{\partial \lambda(\mathbf{v})}{\partial \mathbf{v}} = 0 \quad \Longleftrightarrow \quad A\mathbf{v} = \lambda \mathbf{v}$$

Consistent Local Matches

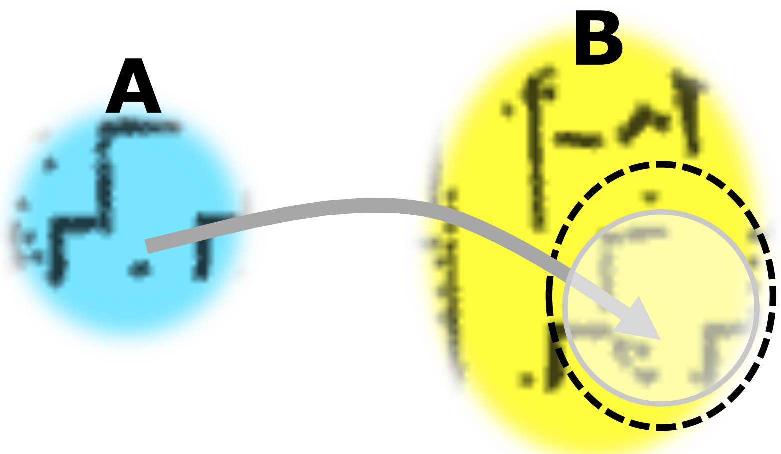
- $A\mathbf{v} = \lambda\mathbf{v}$: Eigenvalue/vector problem
- The dominant eigenvector $\mathbf{v_1}$ maximize

$$\lambda(\mathbf{v}) = \frac{\mathbf{v^T A v}}{\mathbf{v^T v}}$$

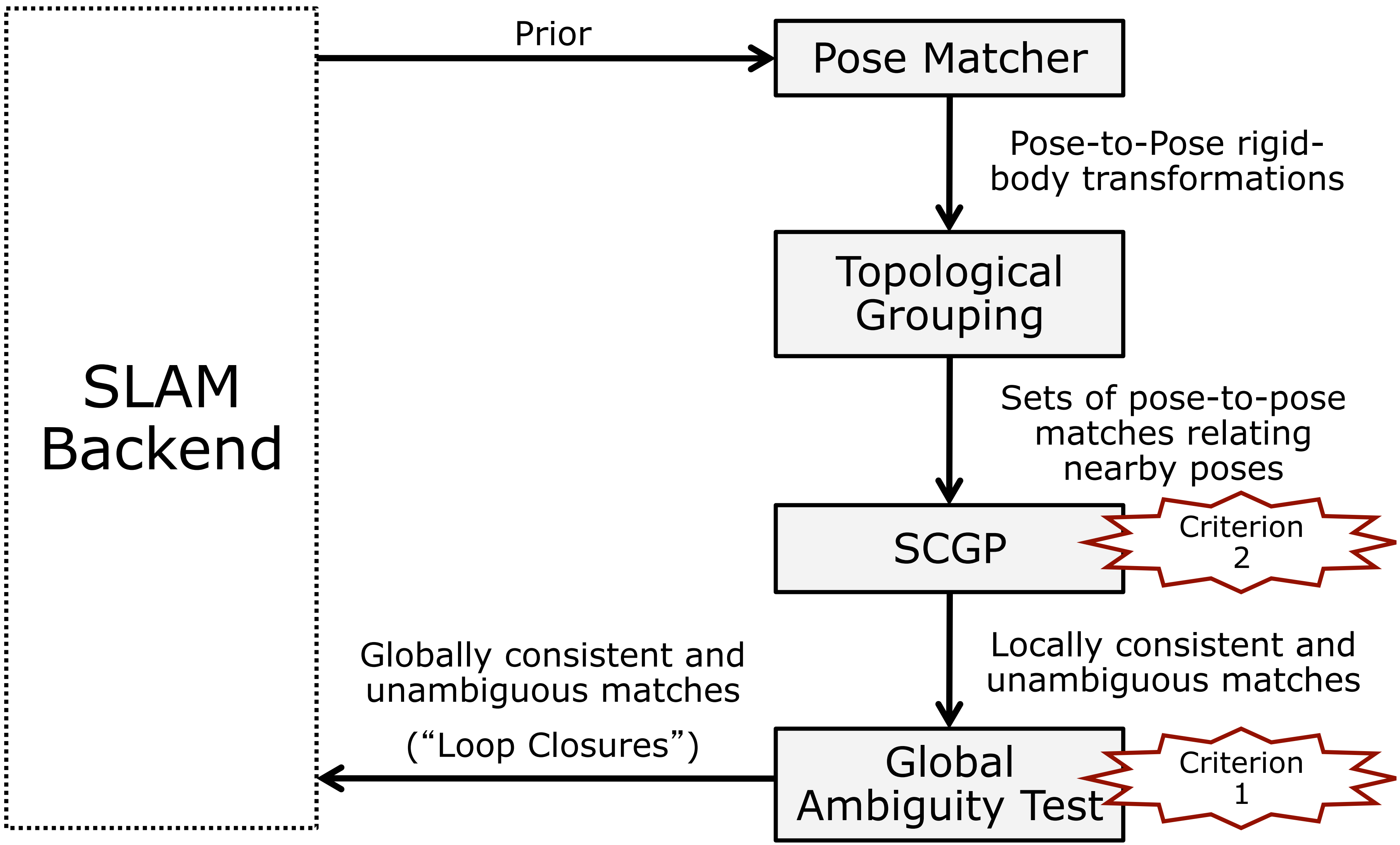
- The hypothesis represented by $\mathbf{v_1}$ is maximally self-consistent subset
- If λ_1/λ_2 is large (>2) then $\mathbf{v_1}$ is locally unambiguous
- Discretize $\mathbf{v_1}$ after maximization

Global Consistency

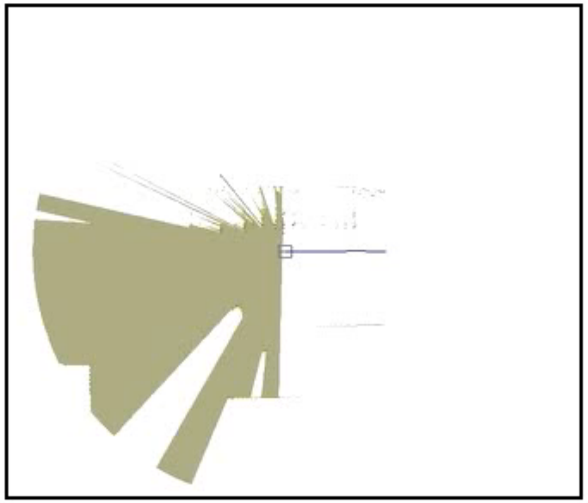
- **Correct method:** Can two copies of A be arranged so that they both fit inside the covariance ellipse?
- **Approximation:** Is the dimension of A at least half the length of the dominant axis of the covariance ellipse?
- Potential failures for narrow local matches



Olson's Proposal



Example



Conclusions

- Matching local observations is used to generate pose-to-pose hypotheses
- Local matches assembled from pose-to-pose hypotheses
- Local ambiguity ("picket fence") can be resolved via SCGP's confidence metric
- Positional uncertainty: more uncertainty requires more evidence

Literature

Spectral Clustering

- Olson: "Recognizing Places using Spectrally Clustered Local Matches"