

Robot Mapping

SLAM Front-Ends

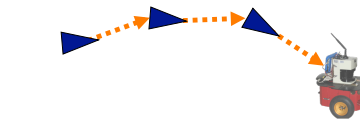
Cyrill Stachniss



Partial image courtesy: Edwin Olson

Graph-Based SLAM

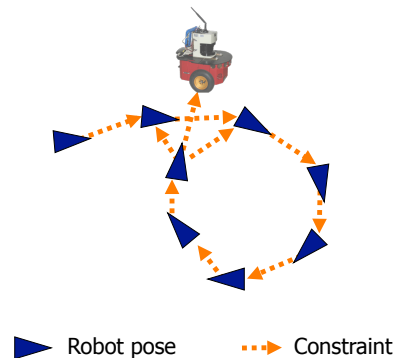
- Constraints connect the nodes through odometry and observations



▶ Robot pose - - - -> Constraint

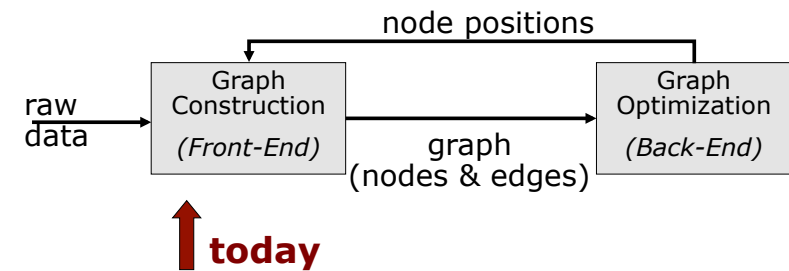
Graph-Based SLAM

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



▶ Robot pose - - - -> Constraint

Interplay between Front-End and Back-End



Constraints From Matching

- Constraints can be obtained from matching observations

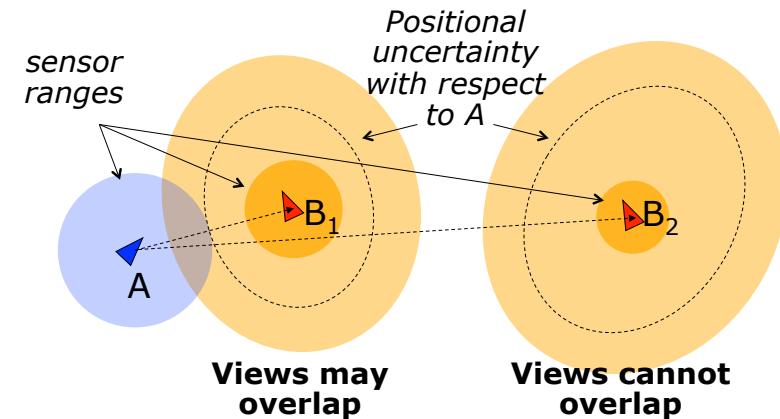
Popular approaches

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

5

Where to Search for Matches?

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



6

Note on the Uncertainty

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

7

Simple ICP-Based Approach

- Assuming a laser range sensor
- 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?

8

Problems

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

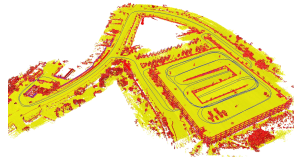
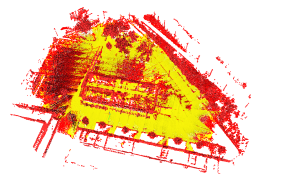
9

Problems

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

10

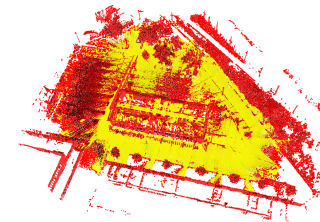
Examples



11

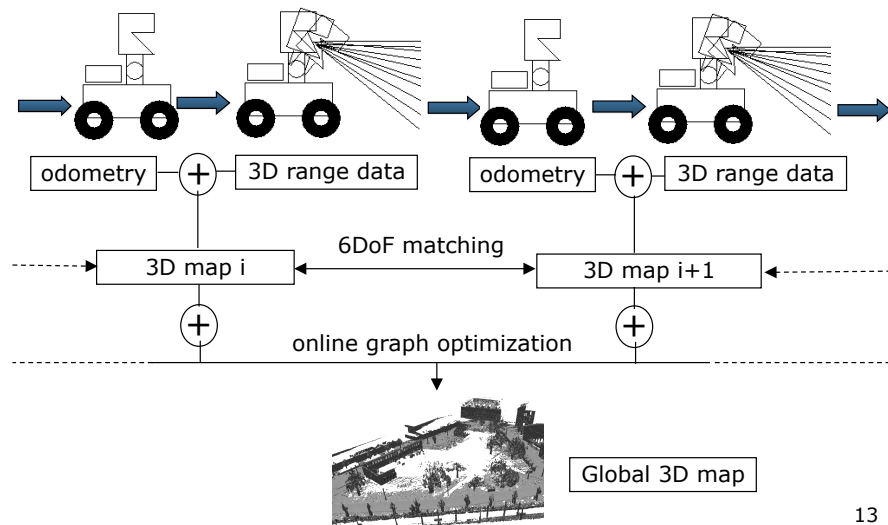
Learning 3D Maps with Laser Data

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

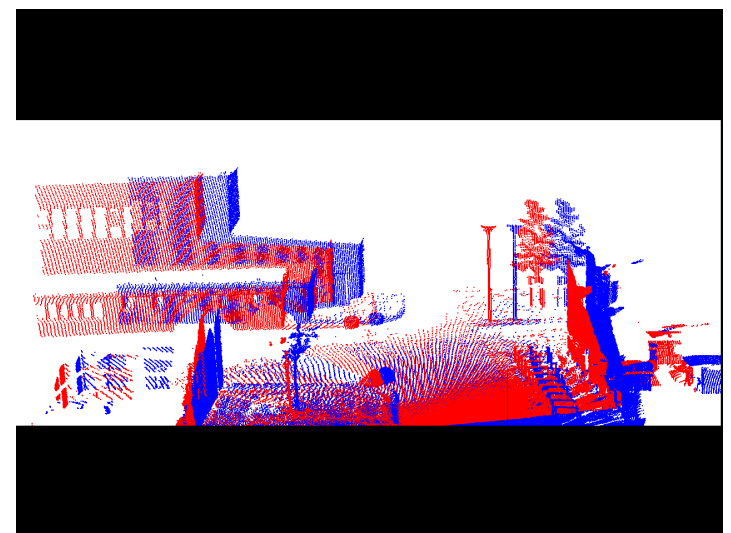


12

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

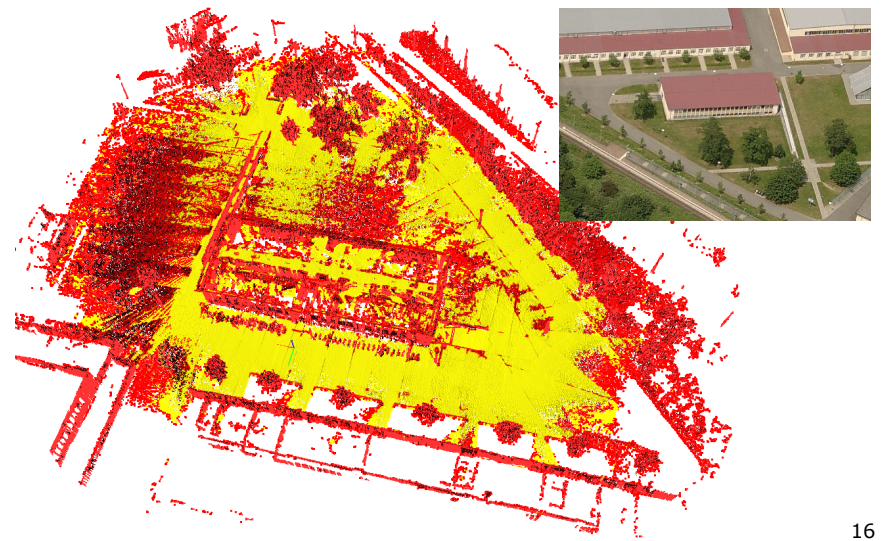
ICP

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

- ICP with additional knowledge

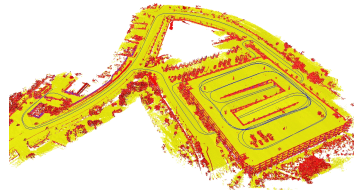
$$e(R, t) = \underbrace{\sum_{c=1}^{C_1} d_v(\mathbf{u}_{i_c}, \mathbf{u}'_{j_c})}_{\text{vertical objects}} + \underbrace{\sum_{c=1}^{C_2} d(\mathbf{v}_{i_c}, \mathbf{v}'_{j_c})}_{\text{traversable}} + \underbrace{\sum_{c=1}^{C_3} d(\mathbf{w}_{i_c}, \mathbf{w}'_{j_c})}_{\text{non-traversable}}$$

Online Estimated 3D Map



Mapping with a Robotic Car

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



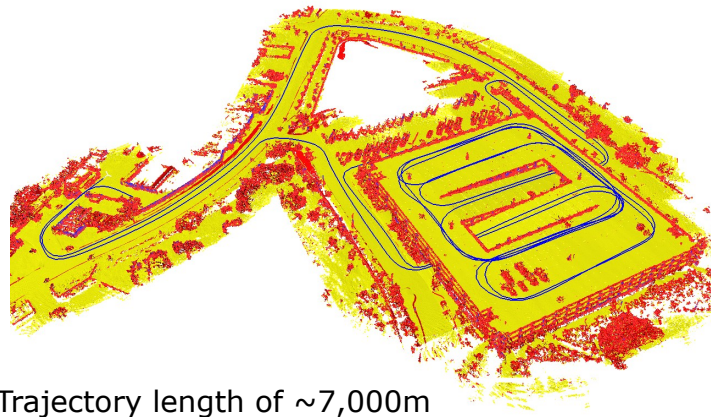
17

Parking Garage



18

Resulting Map



- Trajectory length of $\sim 7,000\text{m}$
- 1661 local 3D maps, cell size of $20\text{cm} \times 20\text{cm}$

19

Mapping with Aerial Vehicles

- Flying vehicles equipped with cameras and an IMU



21

Examples of Camera Images



22

SURF Features

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



23

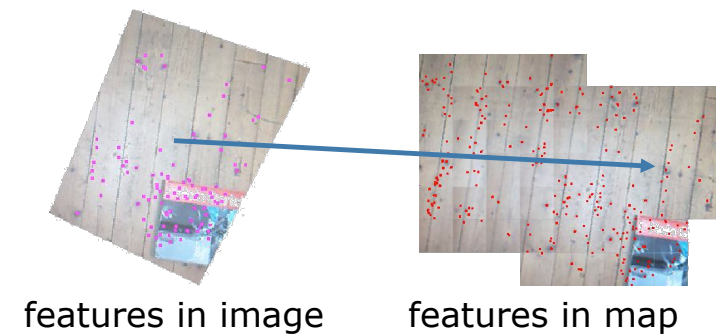
Determining the Camera Pose

Wanted: $x, y, z, \phi, \theta, \Psi$ (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

24

Feature Matching for Pose Estimation



25

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

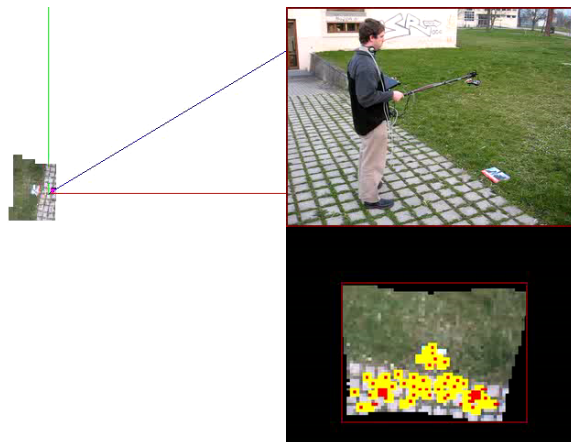
26

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

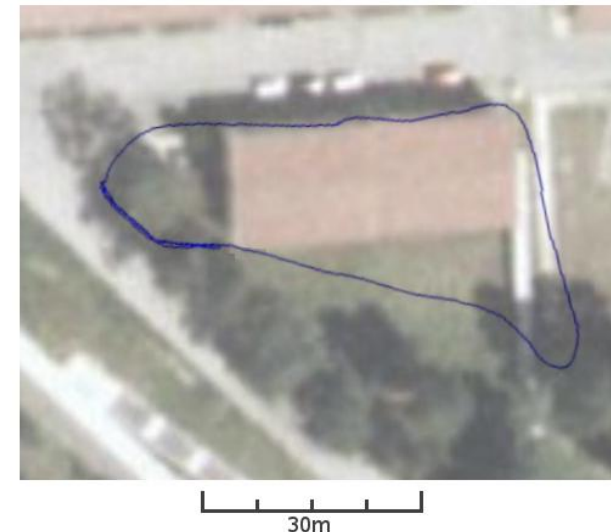
27

Outdoor Example



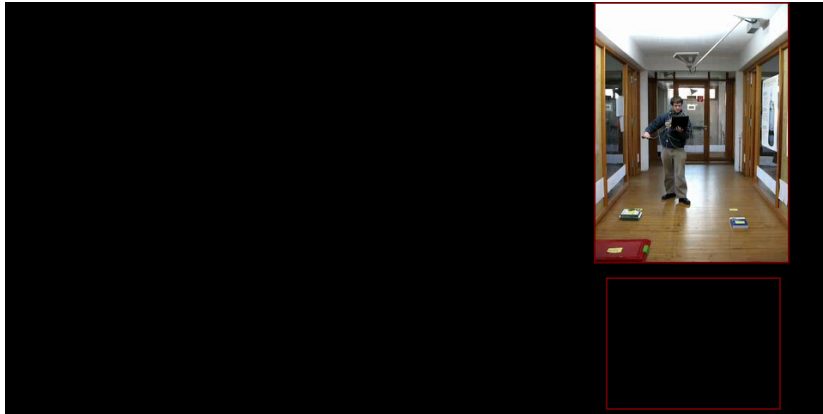
28

Resulting Trajectory



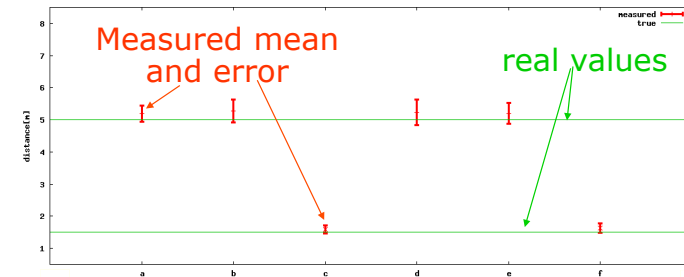
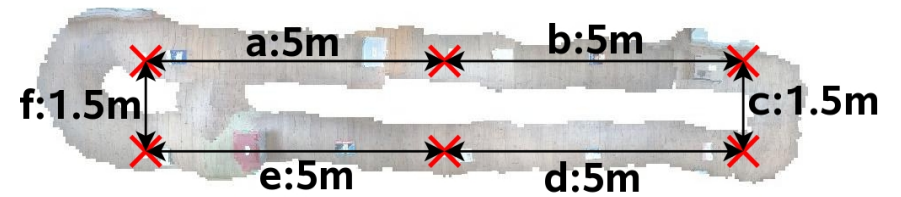
29

Indoor Example



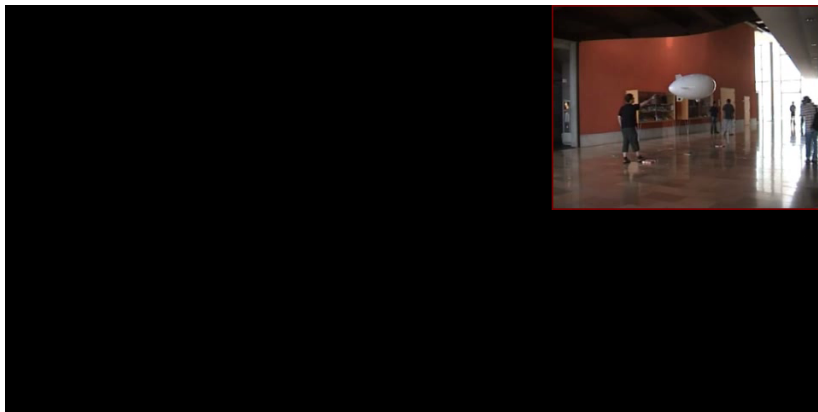
30

Ground Truth



31

System on a Blimp



32

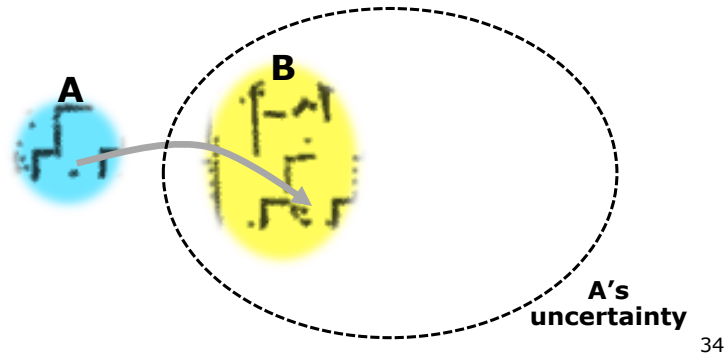
Problems

- 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

33

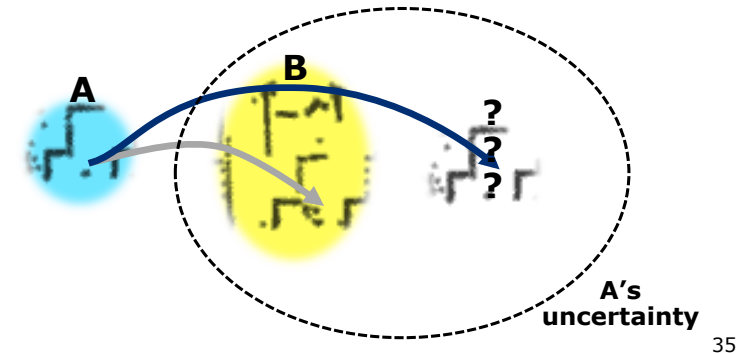
Ambiguities - Global Ambiguity

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



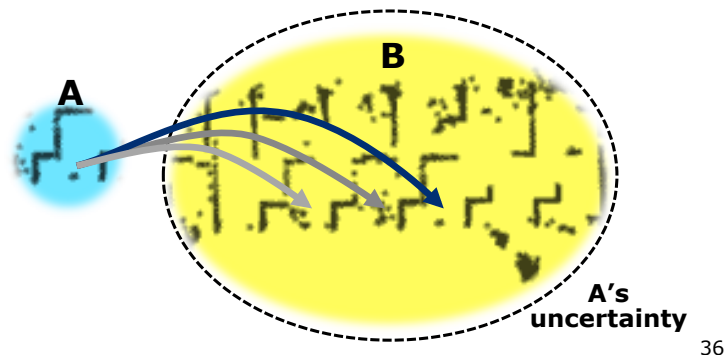
Ambiguities - Global Ambiguity

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



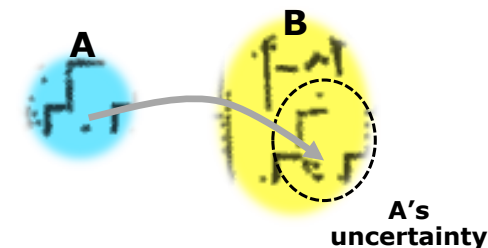
Ambiguities - Global Ambiguity

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



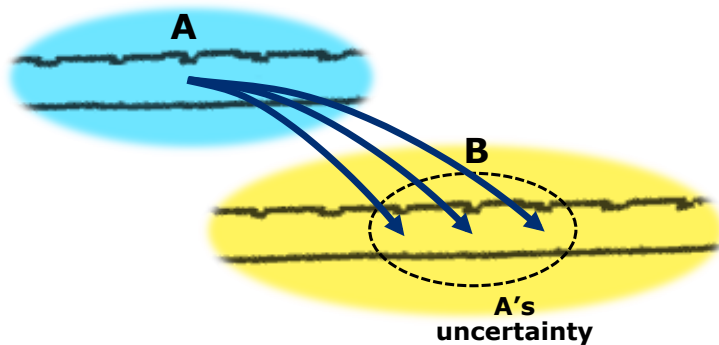
Ambiguities - Global Sufficiency

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



Ambiguities - Local Ambiguity

- "Picket Fence Problem": largely overlapping local matches

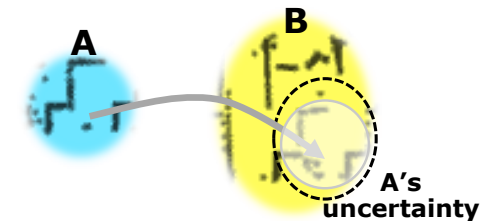


38

Global Match Criteria

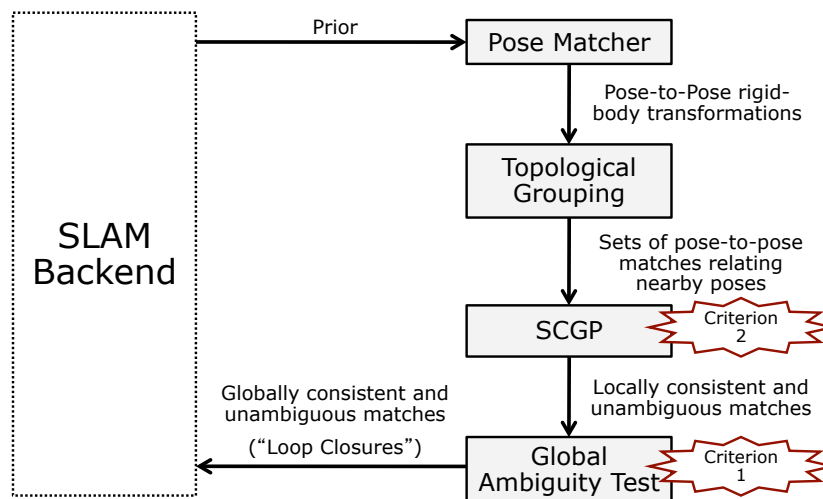
1. Global Sufficiency: There is no possible disjoint match ("A is not somewhere else entirely")
2. Local unambiguity: There are no overlapping matches ("A is either here or somewhere else entirely")

Both need to be satisfied for a match



39

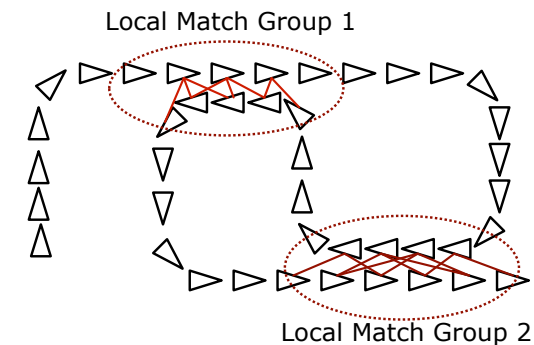
Olson's Proposal



40

Topological Grouping

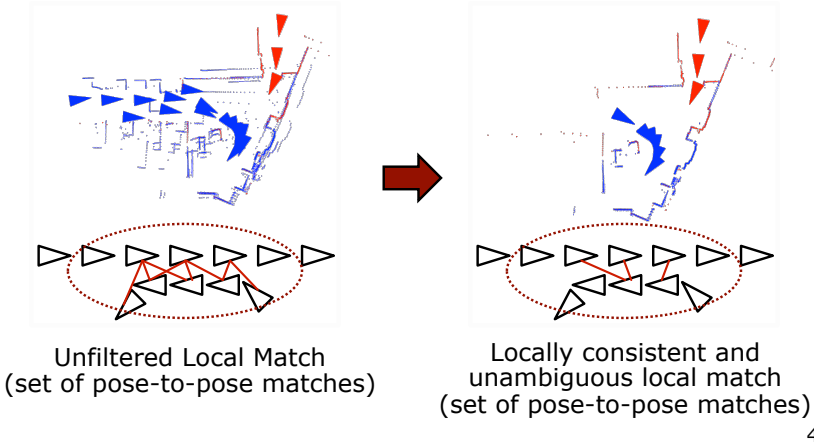
- Group together topologically-related pose-to-pose matches to form local matches
- Each group asks a "topological" question: Do two local maps match?



41

Locally Unambiguous Matches

Goal:



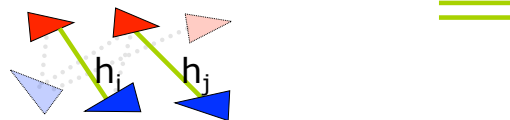
Locally Consistent Matches

- 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"!

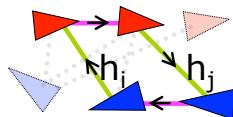
43

Do Two Hypotheses Agree?

- Consider two hypotheses i and j in the set:



- Form a loop using edges from the prior graph

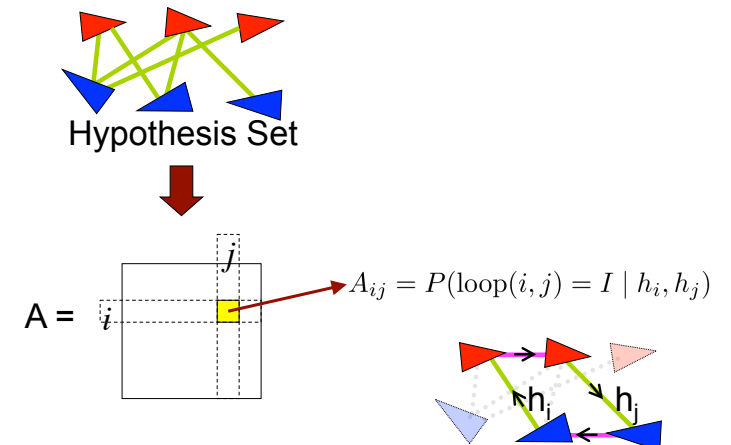


Rigid-body transformation around the loop should be the identity matrix

44

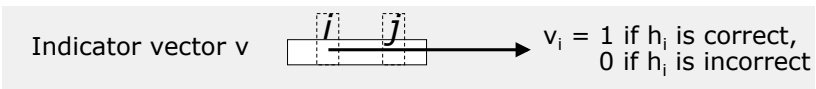
Idea of Olson's Method

- 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)



46

Single Cluster Graph Partitioning

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

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

Sum of all pair-wise consistencies between hypotheses in \mathbf{v}

Number of hypotheses in \mathbf{v}

Gallo et al 1989

- Densest subgraph problem

47

Consistent Local Matches

- We want find \mathbf{v} that maximizes $\lambda(\mathbf{v})$

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

- Treat as continuous problem
- Derive and set to zero

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

- Which leads to (for symmetric \mathbf{A})

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

48

Consistent Local Matches

- $\mathbf{A} \mathbf{v} = \lambda \mathbf{v}$: Eigenvalue/vector problem
- The dominant eigenvector \mathbf{v}_1 maximizes

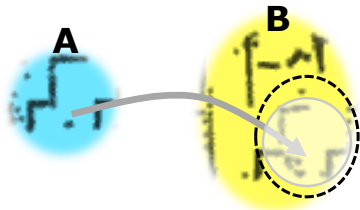
$$\lambda(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{A} \mathbf{v}}{\mathbf{v}^T \mathbf{v}}$$

- The hypothesis represented by \mathbf{v}_1 is maximally self-consistent subset
- If λ_1/λ_2 is large (e.g., $\lambda_1/\lambda_2 > 2$) then \mathbf{v}_1 is regarded as locally unambiguous
- Discretize \mathbf{v}_1 after maximization

49

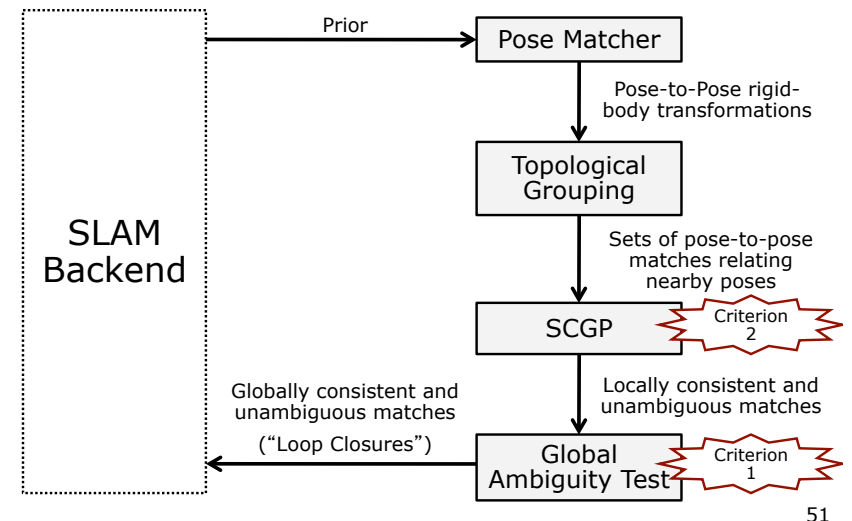
Global Consistency

- **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



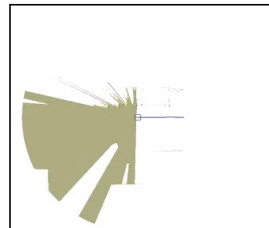
50

Olson's Proposal



51

Example



52

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

53

Literature

Spectral Clustering

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