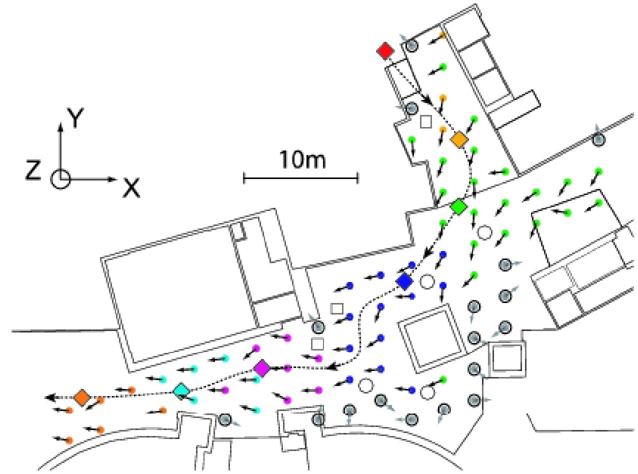


Body-relative Navigation Using Uncalibrated Cameras

Olivier Koch
koch@csail.mit.edu

Seth Teller
teller@csail.mit.edu

Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

Motivation

- Navigation guidance to humans
  - Finding our way in complex/new environments
- Why is it hard?
  - No external source of localization (GPS)
  - Unknown environment (no map)
- Why should you care?
  - Soldiers in the field
  - Visually impaired
  - Guidance in public places (hospitals, museums)

Vision-based navigation

Four Pointgrey Firefly MV Cameras (640x480 8-bit grayscale images)
FOV: 360° (h) x 90° (v)

Why vision?

- ✓ Light, inexpensive, compact
- ✓ Rich information (vs laser rangerfinders)
- ✓ No temporal drift (vs inertial sensors)

Uncalibrated cameras

Four Pointgrey Firefly MV Cameras (640x480 8-bit grayscale images)
FOV: 360° (h) x 90° (v)

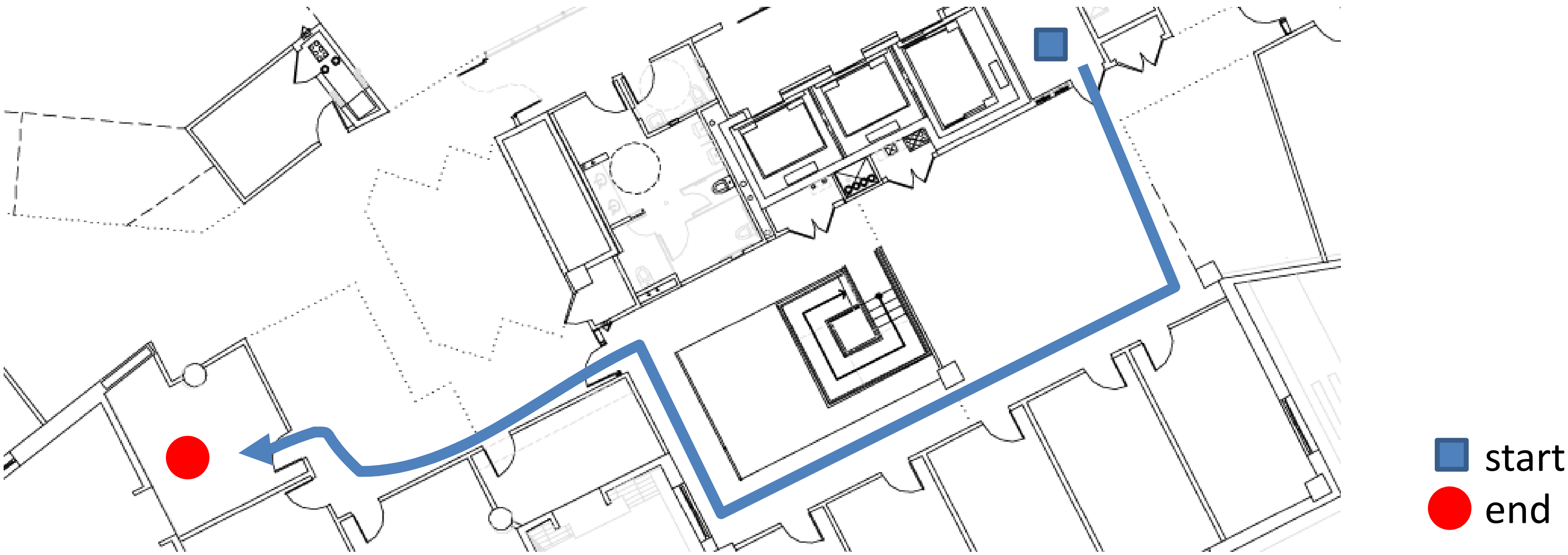
Why use uncalibrated cameras?

- Intrinsic calibration is tedious
- Extrinsic calibration is hard for body-worn applications

Problem statement

Input

Live video stream from wearable set of uncalibrated cameras



Problem statement

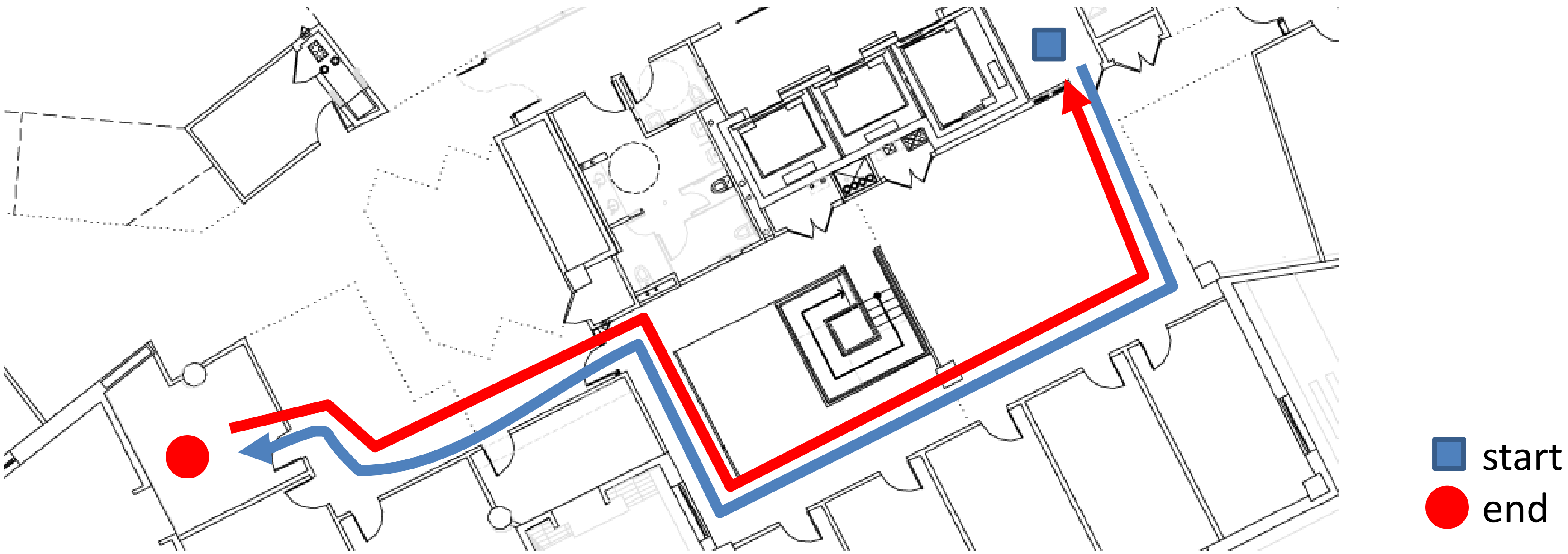
Input

Live video stream from wearable set of uncalibrated cameras

Output

Body-relative guidance for:

- Homing (going back to start point)
- Replay (from start point to end point)
- Point-to-point navigation

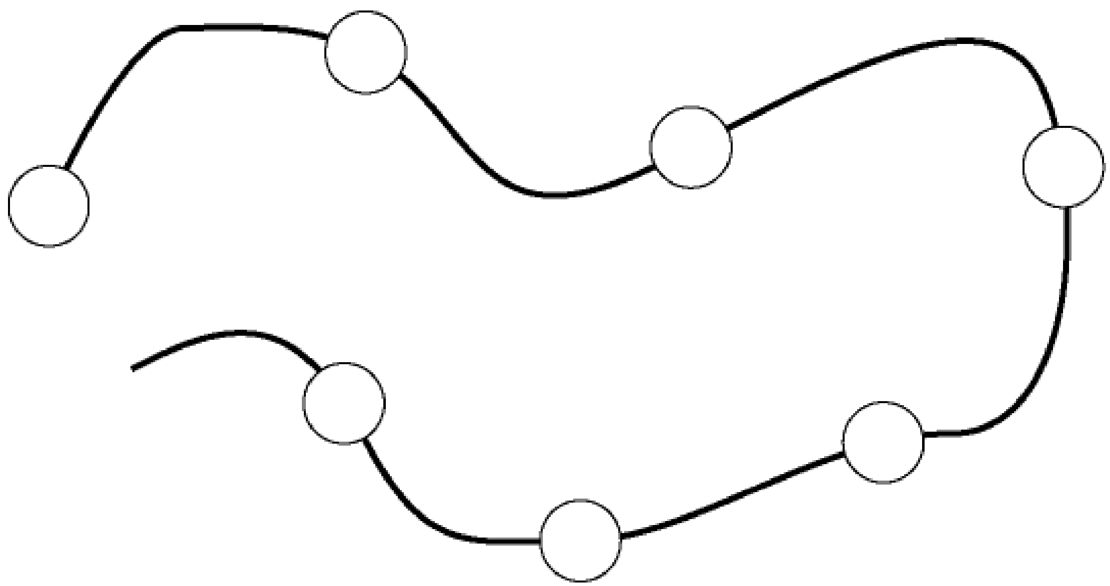


Sample dataset

Method overview

- Exploration path: undirected graph (*place graph*)
- Node: physical location in the world
- Edge: physical path between two nodes traversed by the user

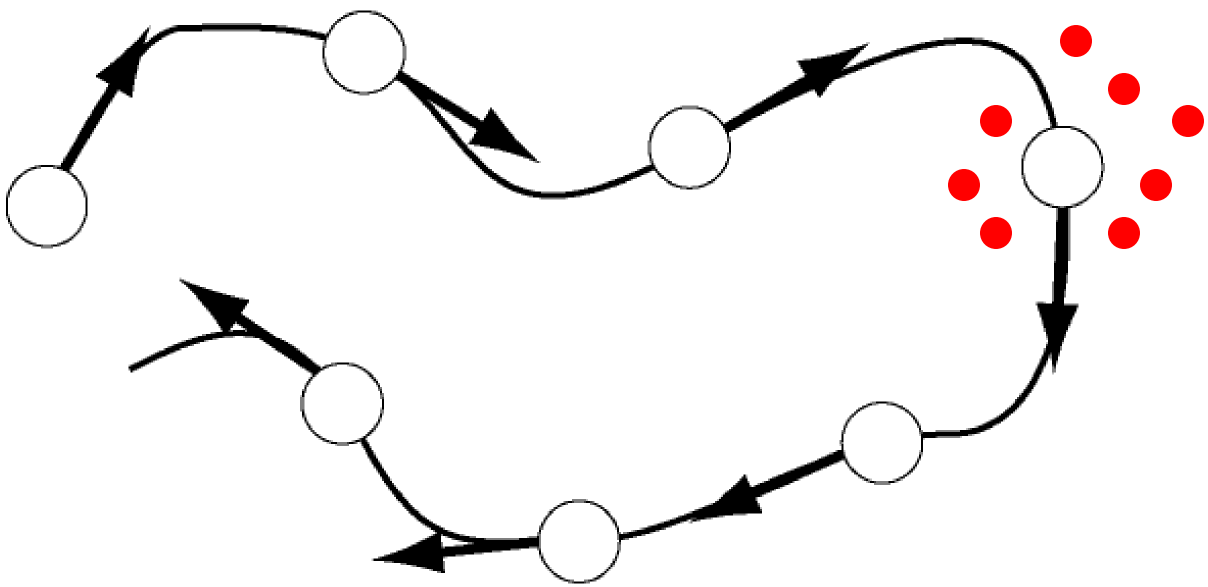
✓ Makes no assumption on user motion between nodes



Method overview

- Local node orientation: direction of the user leaving the node
  - ➢ Assume smooth user motion
- Local node observations (visual features)
  - ➢ Assume distinctive feature visibility

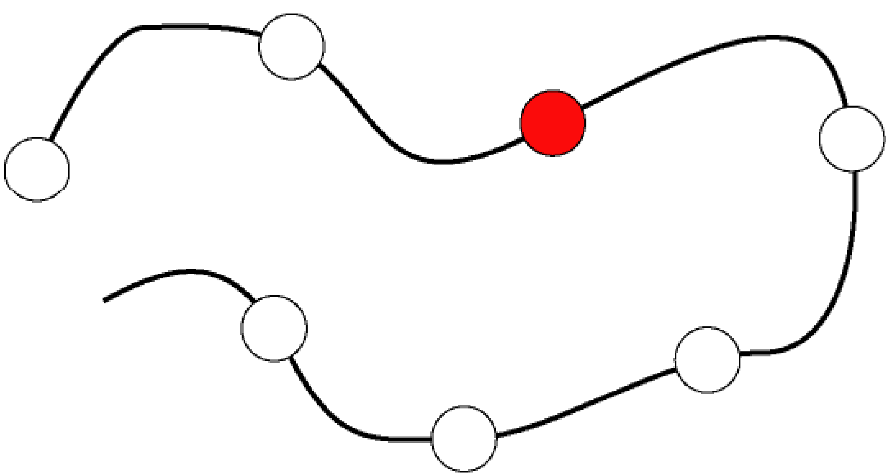
✓ No global coordinate frame



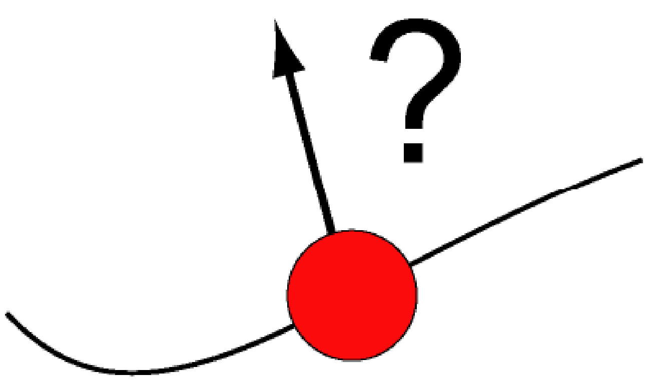
Method overview

- Node-to-node "hopping" problem
- ✓ Does not require metric mapping of the environment
- ➢ Assumes that user stays in the graph during guidance

Determine location of user in the graph
(*local node estimation)*

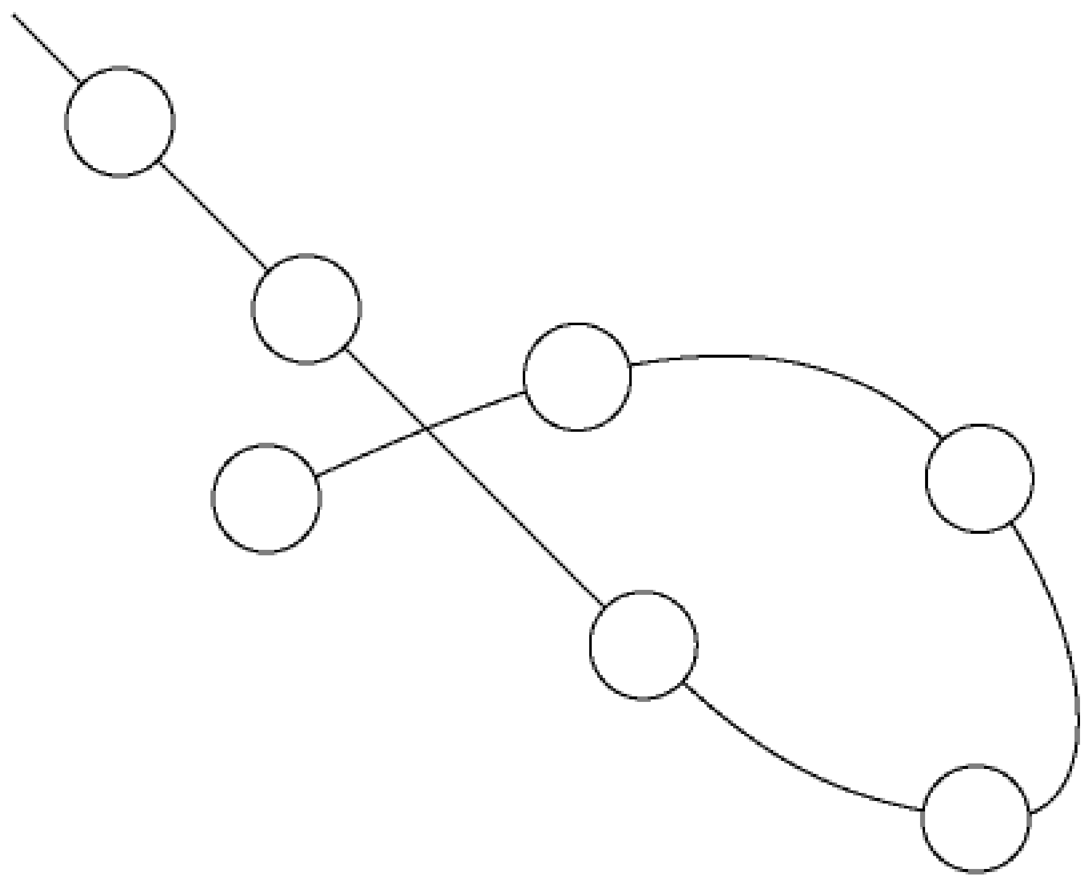


Guide the user at that location
(*rotation guidance*)



Method overview

- Loop closure detection



Limitations & advantages

- **Limitations**
  - User leaving exploration path
  - Smooth user motion
  - Distinctive features visibility

- **Advantages**
  - Provides intuitive, body-relative guidance
  - Requires no extrinsic or intrinsic camera calibration
  - Scales to arbitrary large environments

Related work

Visual Simultaneous Localization and Mapping (SLAM)

- Davison et al., MonoSLAM: Real-Time Single Camera SLAM, PAMI ’07
- J. Neira et al., Data association in O(n) for Divide and Conquer SLAM, RSS ’07
- Wolf et al., Robust Vision-Based Localization by Combining an Image Retrieval System with Monte Carlo Localization, IEEE Transactions Robotics ’05
- Konolige, Agrawal et al., . Mapping, Navigation and Learning for Off-road Traversal, Journal of Field Robotics ‘08

Metric and topological localization

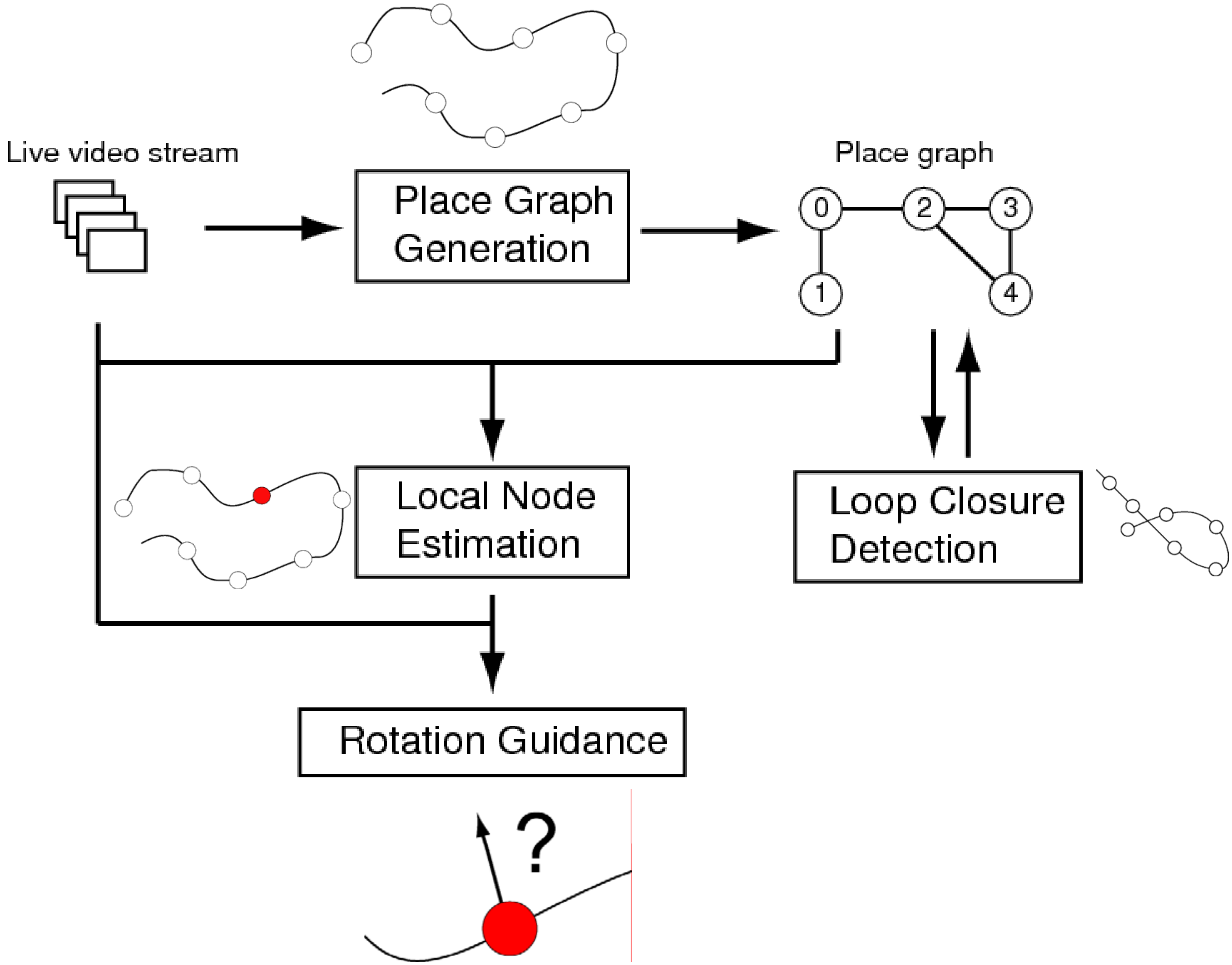
- Zhang & Kosecka, Hierarchical Building Recognition, Image and Vision Computing ‘07
- B. Kuipers, Using the topological skeleton for scalable global metrical map-building, IROS ’04

Related work

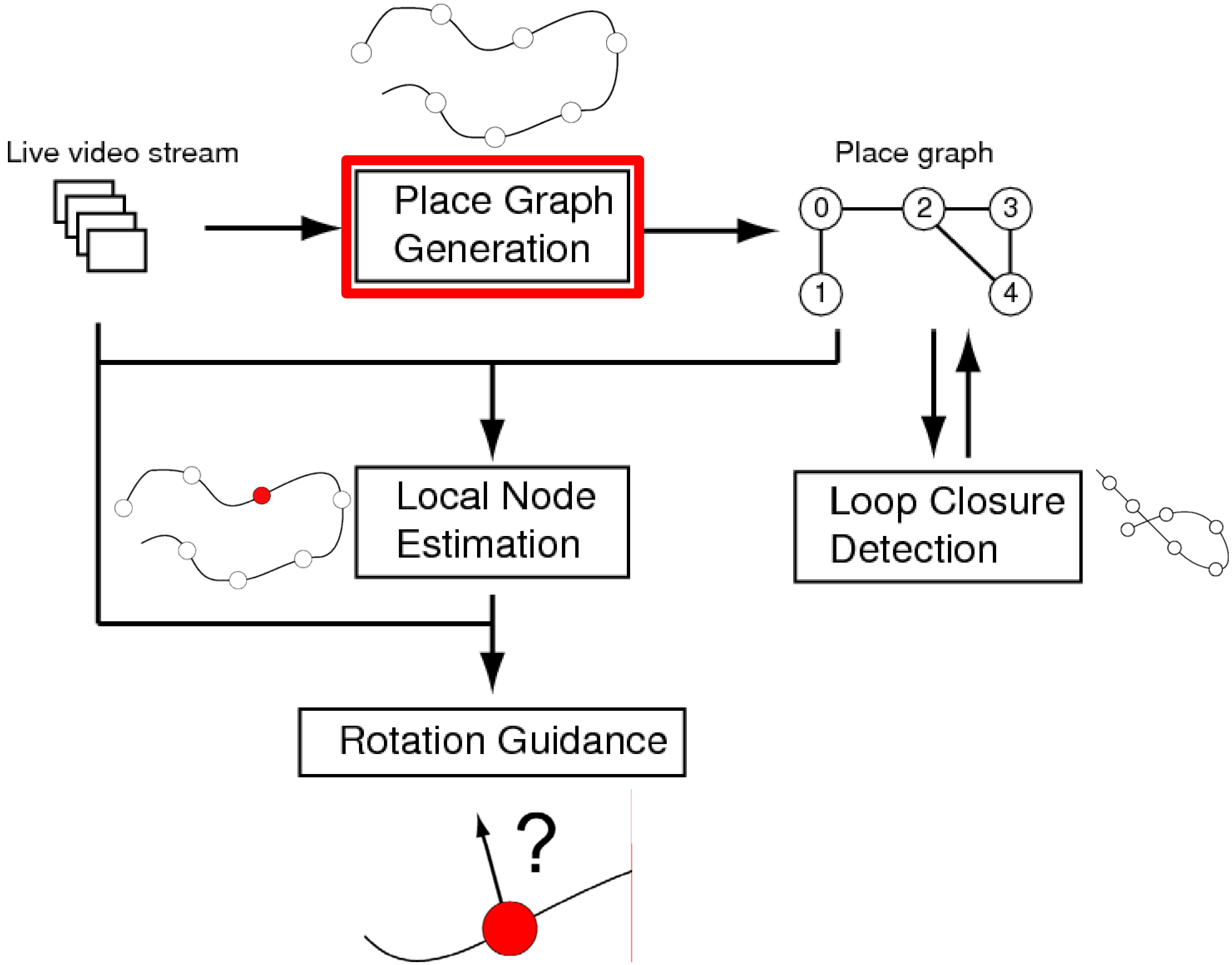
Appearance-based navigation

- Cummins & Newman, Probabilistic Appearance Based Navigation and Loop Closing, ICRA ’07
- Collet, Landmark learning and guidance in insects, Ph. Trans. Roy. Soc. London, 1992
- Chen & Birchfield, Qualitative vision-based mobile robot navigation, ICRA’06
- Zhang & Kleeman, Robust appearance-based visual route following for navigation in large-scale outdoor environments, IJRR’09

Method overview



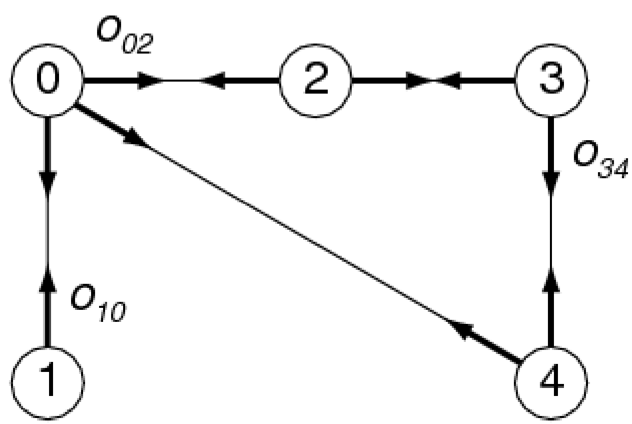
Method overview



The place graph

- World as an undirected graph G = (V, E)

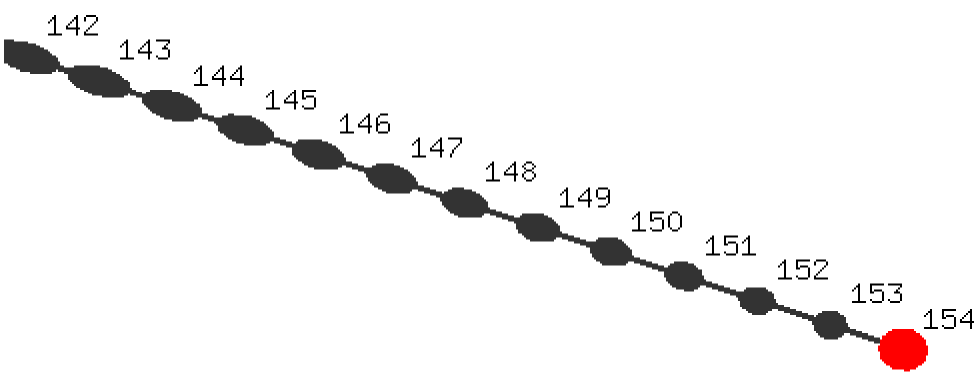
Object	Represents…	Data Structure
Node	Location in the world	Visual features (e.g. SIFT)
Edge	Physical path between two nodes	N/A



The place graph

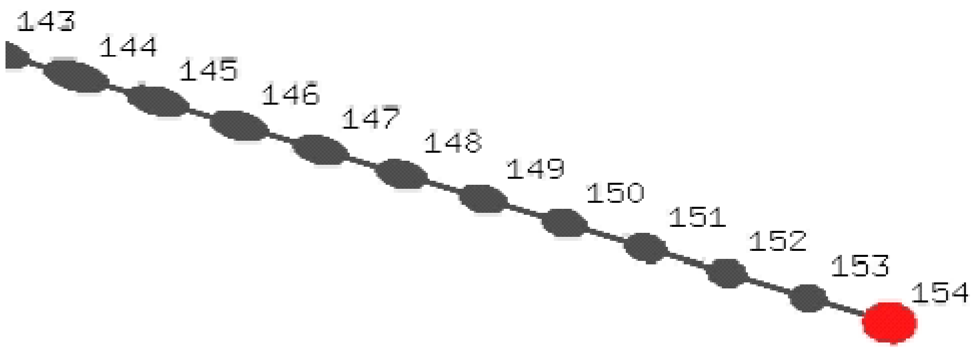
- Similarity function $\Psi()$
  - Input: two sets of features F_1, F_2
  - Output: average L2-distance for all feature matches between F_1 and F_2

- Creating a node whenever $\Psi > \delta$

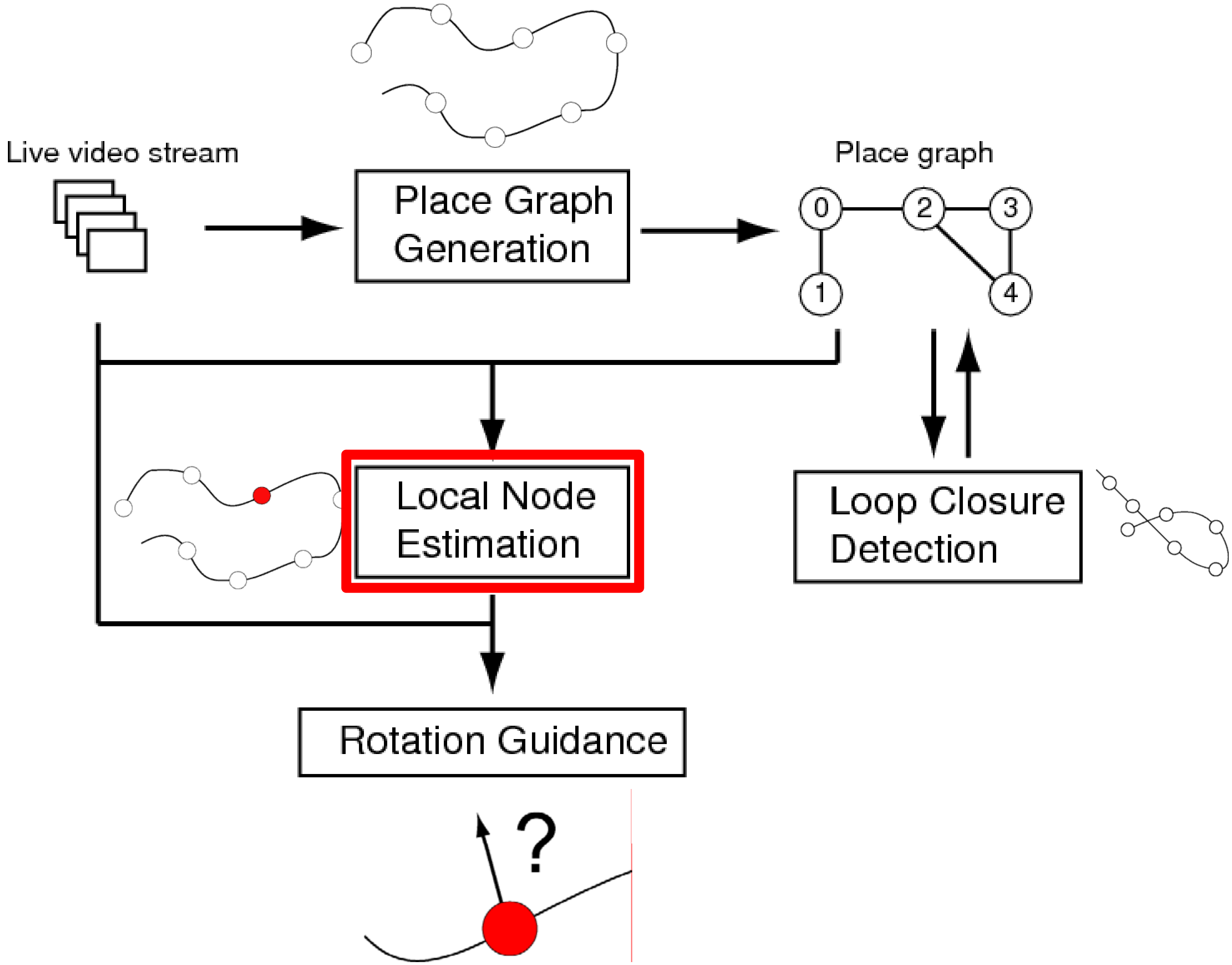


In practice, new node every three seconds (5 meters) at human-walking speed.

The place graph



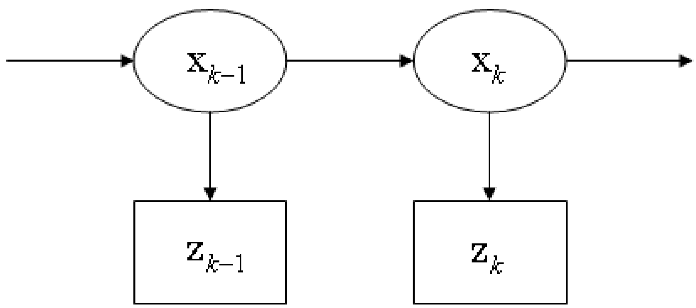
Method overview



Local node estimation

- **Input** position of the user in the map at time *t-1*
  observations at time *t*
- **Output** position in the map at time *t*

- User motion = Markov process
- Recursive Bayesian estimation
  - State $\mathbf{x}_k$: position in the map at time k *(node label)*
  - Measurement $\mathbf{z}_k$: observations at time k *(SIFT)*

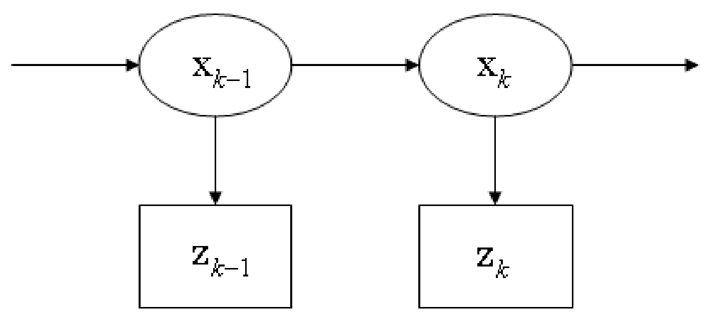


Local node estimation

$$p(x_k \mid z_{k-1}) = \sum p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{k-1}) \quad \text{(prediction)}$$

$$p(x_k \mid z_k) = \lambda\ p(z_k \mid x_k)\, p(x_k \mid z_{k-1}) \quad \text{(update)}$$

$$p(z_k \mid x_k) \sim 1/\varepsilon + \psi(x_k, z_k) \qquad p(x_k \mid x_{k-1}) = N(0, \sigma)$$



Local node estimation

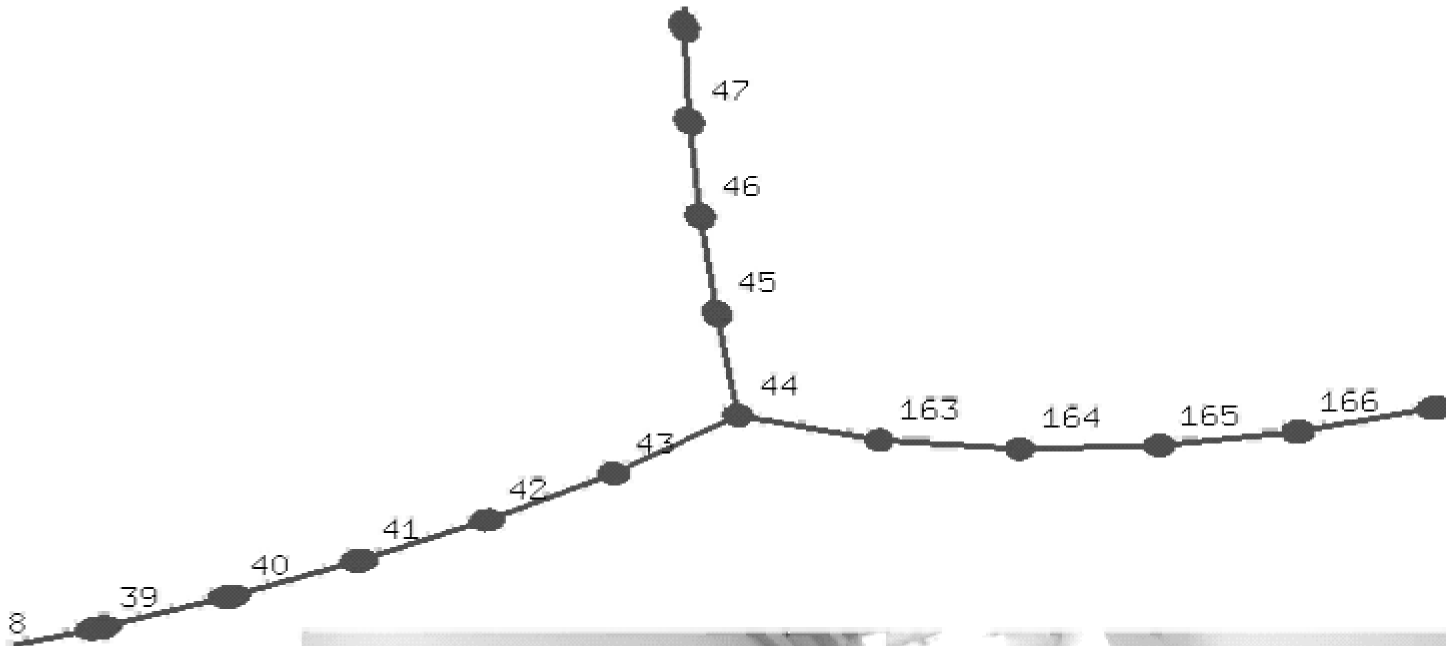
$$p(x_k \mid z_{k-1}) = \sum p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{k-1}) \quad \text{(prediction)}$$

$$p(x_k \mid z_k) = \lambda \; p(z_k \mid x_k)\, p(x_k \mid z_{k-1}) \quad \text{(update)}$$

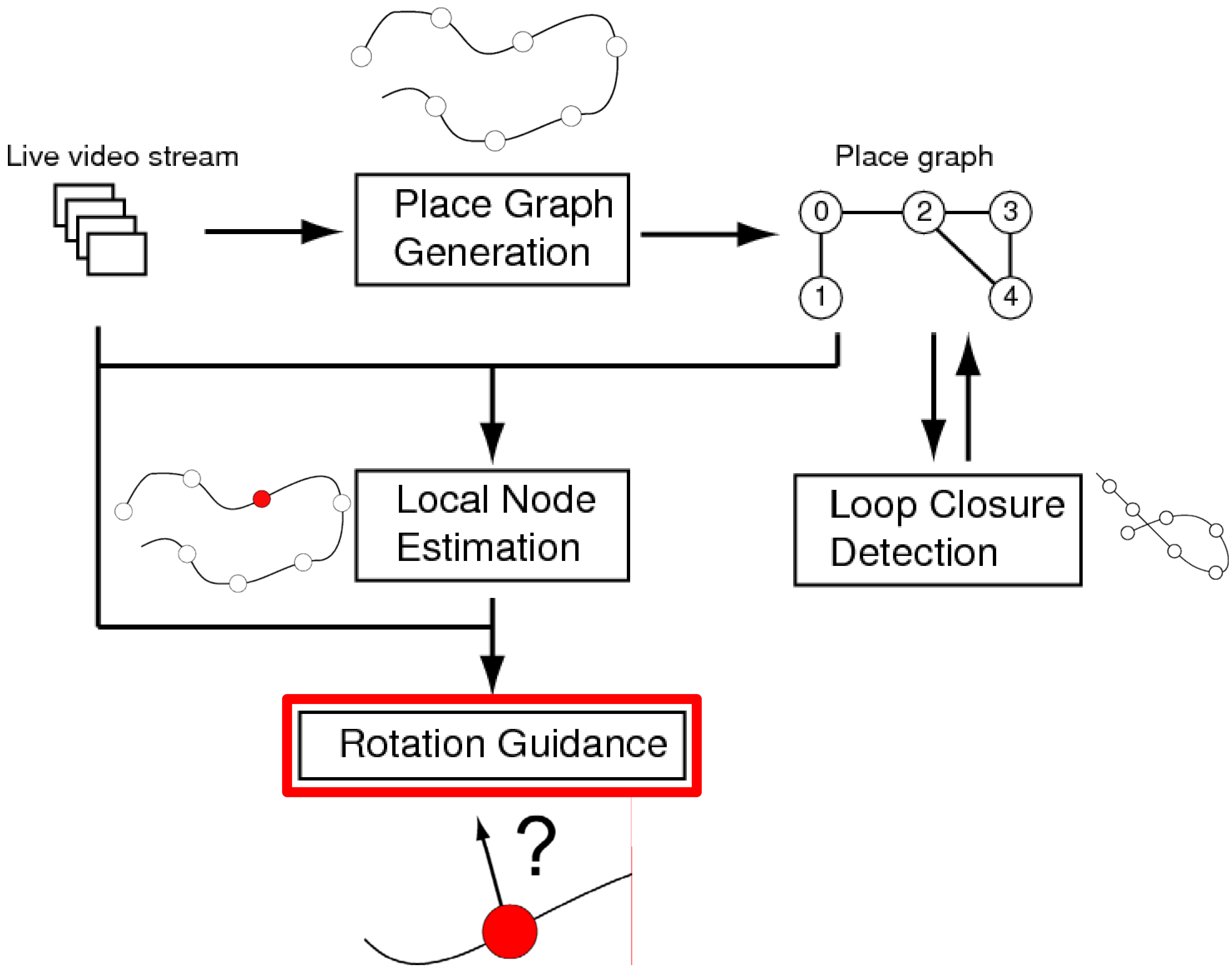
$$p(z_k \mid x_k) \sim 1 \big/ \varepsilon + \psi(x_k, z_k) \qquad p(x_k \mid x_{k-1}) = N(0, \sigma)$$

- Compute pdf over a local neighborhood of current position only
- No new node creation

Local node estimation



Method Overview



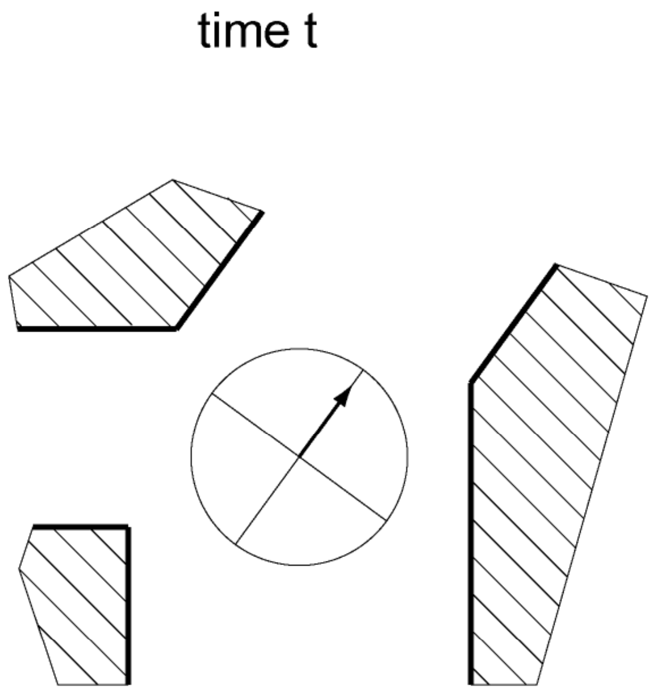
Rotation Guidance

- **Input** current position of user in the graph
  current observations
- **Output** guidance to next node in user's body frame

- **Approach** visual learning

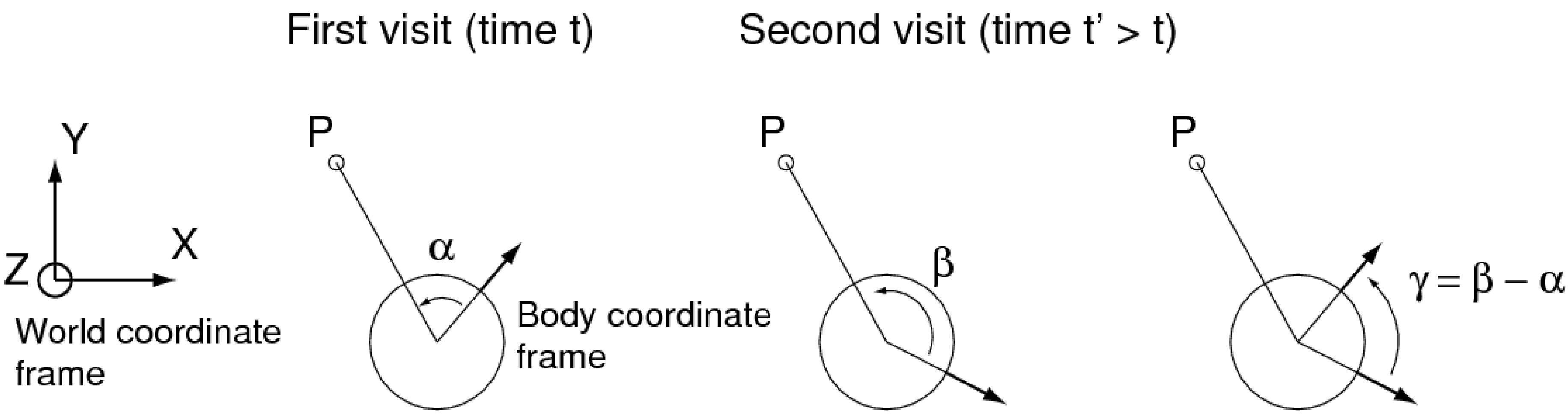
The relative orientation problem

- **Problem** Estimate the relative user orientation between two visits of the same location (in 2D)



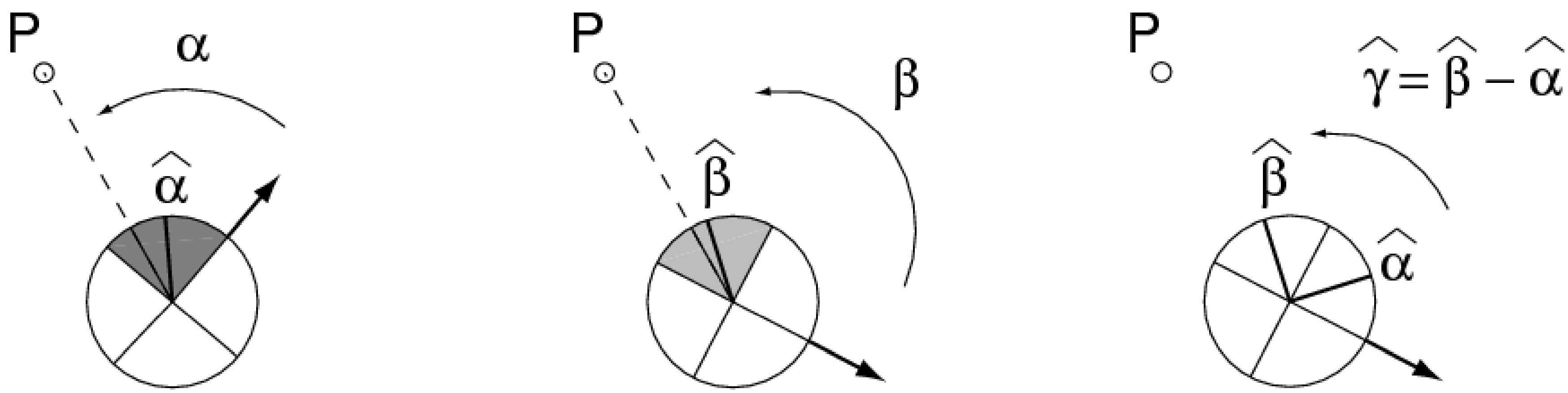
The relative orientation problem

- Assuming intrinsic & extrinsic camera calibration
- World features = bearing measurements ($\alpha,\beta,\ldots$)



The relative orientation problem

- Assuming **no** intrinsic & extrinsic camera calibration
  - Bearing α $\rightarrow$ coarse bearing $\widehat{\alpha}$
  - $\widehat{\alpha}$: average of all possible measurements on the camera



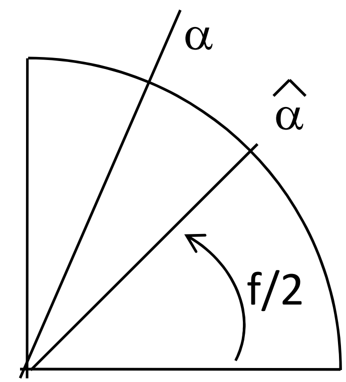
Four cameras, covering each 90° of FOV

The relative orientation problem

- **Principle** For a large number of measurements, and given the assumptions below, using $\hat{\alpha}$ instead of α yields a statistically valid estimate of the relative orientation γ.

- **Assumptions**
  - Observations are uniformly distributed in image space
  - Observations are made from the same vantage point during revisit

The relative orientation problem



$\alpha \sim U(0, 2\pi) \Rightarrow \delta_\alpha = \hat{\alpha} - \alpha \sim U(-f/2, f/2)$
where f is the camera horizontal field of view.

Variance $\sigma_\delta^2 = f^2/12$ ($\sigma_\delta = 26°$ for $f = 90°$)

Central limit theorem: for a large number of observations $\{\alpha_i\}_{0 \leq i < n}$, the average of $\{\delta_i\}$ is normally distributed with a standard deviation $\sigma = \sigma_\delta/\sqrt{n}$ ($\sigma = 2.6°$ for $n = 100$).

$\delta_\alpha \sim N(0, \sigma), \delta_\beta \sim N(0, \sigma) \Rightarrow \delta_\gamma \sim N(0, 2\sigma)$.

The match matrix

$\alpha \in [0, 2\pi)$ is continuous.
$\hat{\alpha}$ is discrete: $\hat{\alpha} \in \{\alpha_1, \cdots, \alpha_n\}$.
(e.g. $\hat{\alpha} \in \{-3\pi/4, -\pi/4, \pi/4, 3\pi/4\}$).

$\hat{\gamma}$ is discrete: $\hat{\gamma} \in \{\gamma_{ij} \mid \hat{\alpha} = \alpha_i, \hat{\beta} = \beta_j\}_{0 \leq i,j < n}$.

We represent $\hat{\gamma}$ as a matrix H (match matrix): $H = (\gamma_{ij})$.

The match matrix

- *H(i,j)* = user rotation associated to a match btw camera i and camera j
- *H* = coarse approximation of the full camera calibration

- *H* is anti-symmetric

$$H(i,j) = -H(j,i)$$

- *H* satisfies the “circular equality”

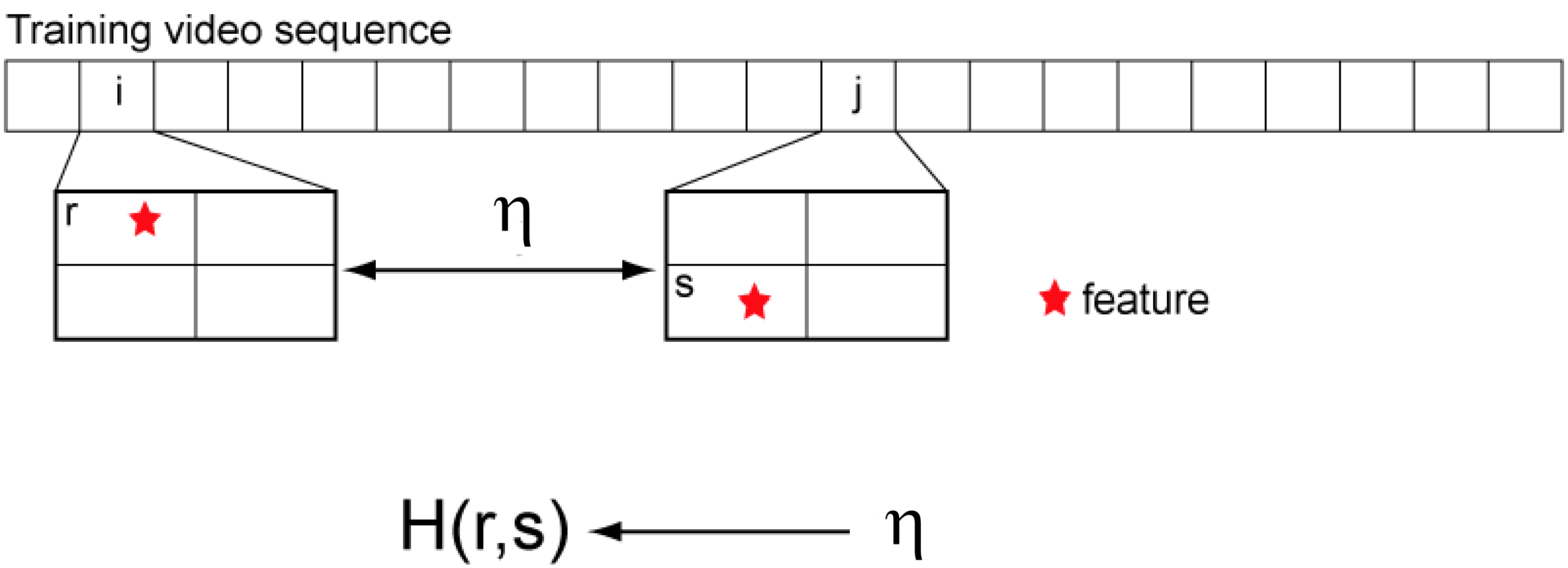
$$\sum_{0 \leq i < n} H(i, (i+1) \mod n) = 0 \mod 2\pi$$

Learning the match matrix

- **Training Phase**
  - Learn match matrix from training data
  - Once for a given camera configuration
  - Does not depend on training environment

- **Training algorithm**
  - User rotates in place in arbitrary environment
  - Algorithm “learns” the match matrix

Learning the match matrix

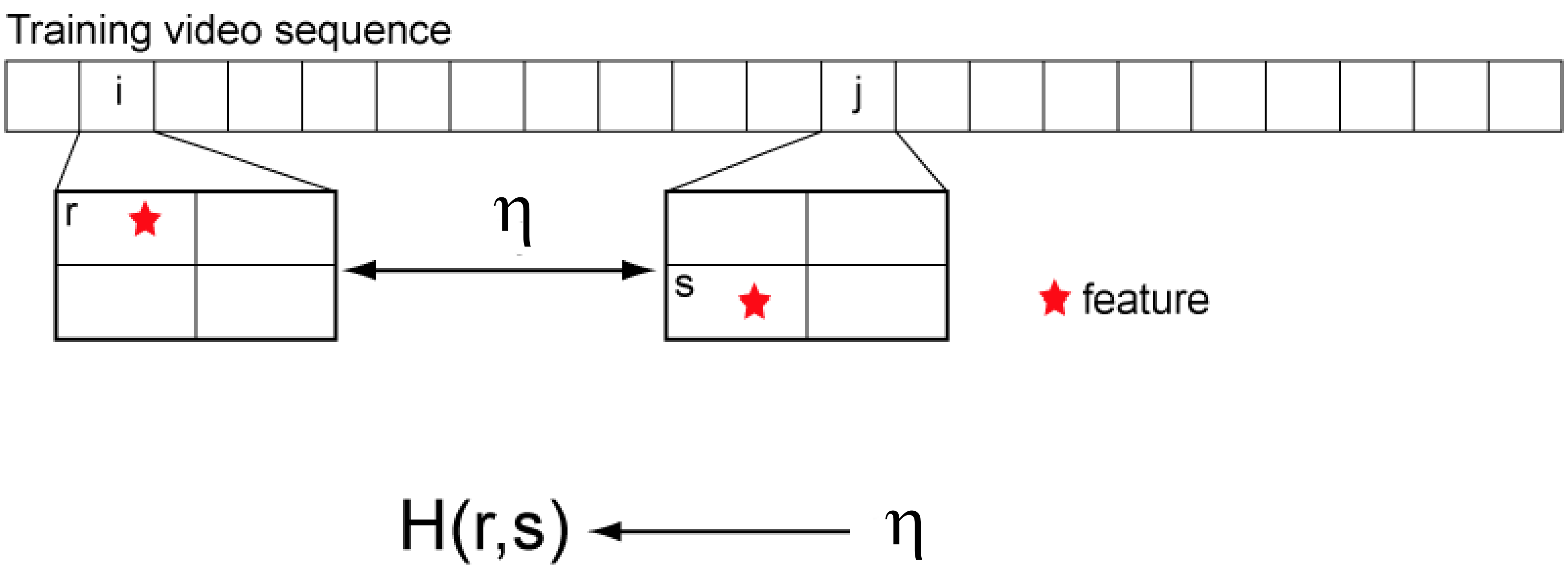


User rotates in place n_r times in an arbitrary environment (n_r =2)

For each pair of frames (f_i,f_j) in the training sequence:

- Estimate corresponding user rotation η (e.g. assuming constant rot. speed)
- Compute feature matches between f_i and f_j
- For each match *m* between a feature on camera ***r*** and a feature on camera ***s***, update $H(\boldsymbol{r},\boldsymbol{s})$ with η

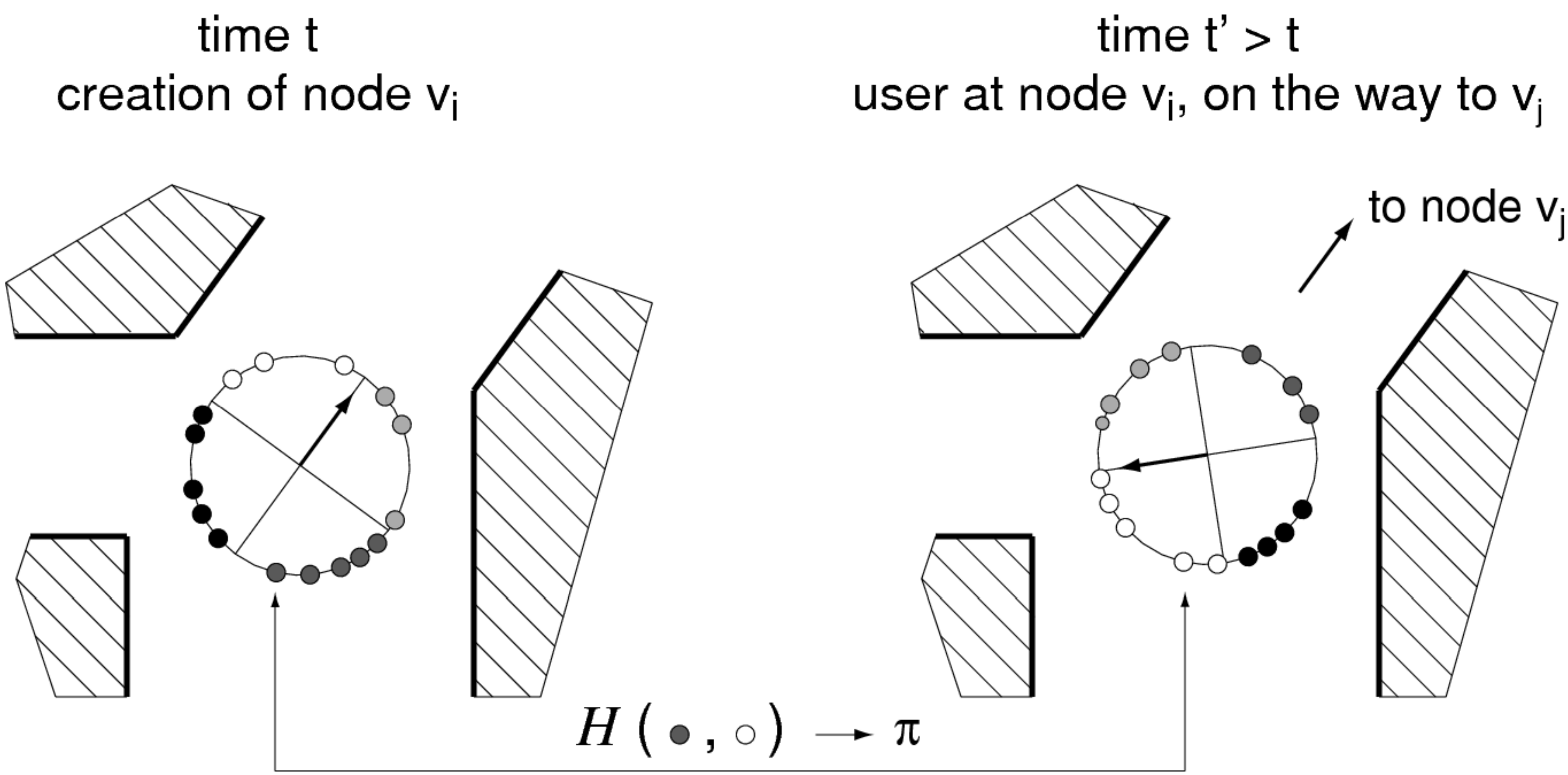
Learning the match matrix



- Training algorithm
  - Runs in arbitrary environment
  - Done once for a given camera configuration
  - Fast (a few minutes) and simple
  - Quadratic complexity in # frames and # features/frame

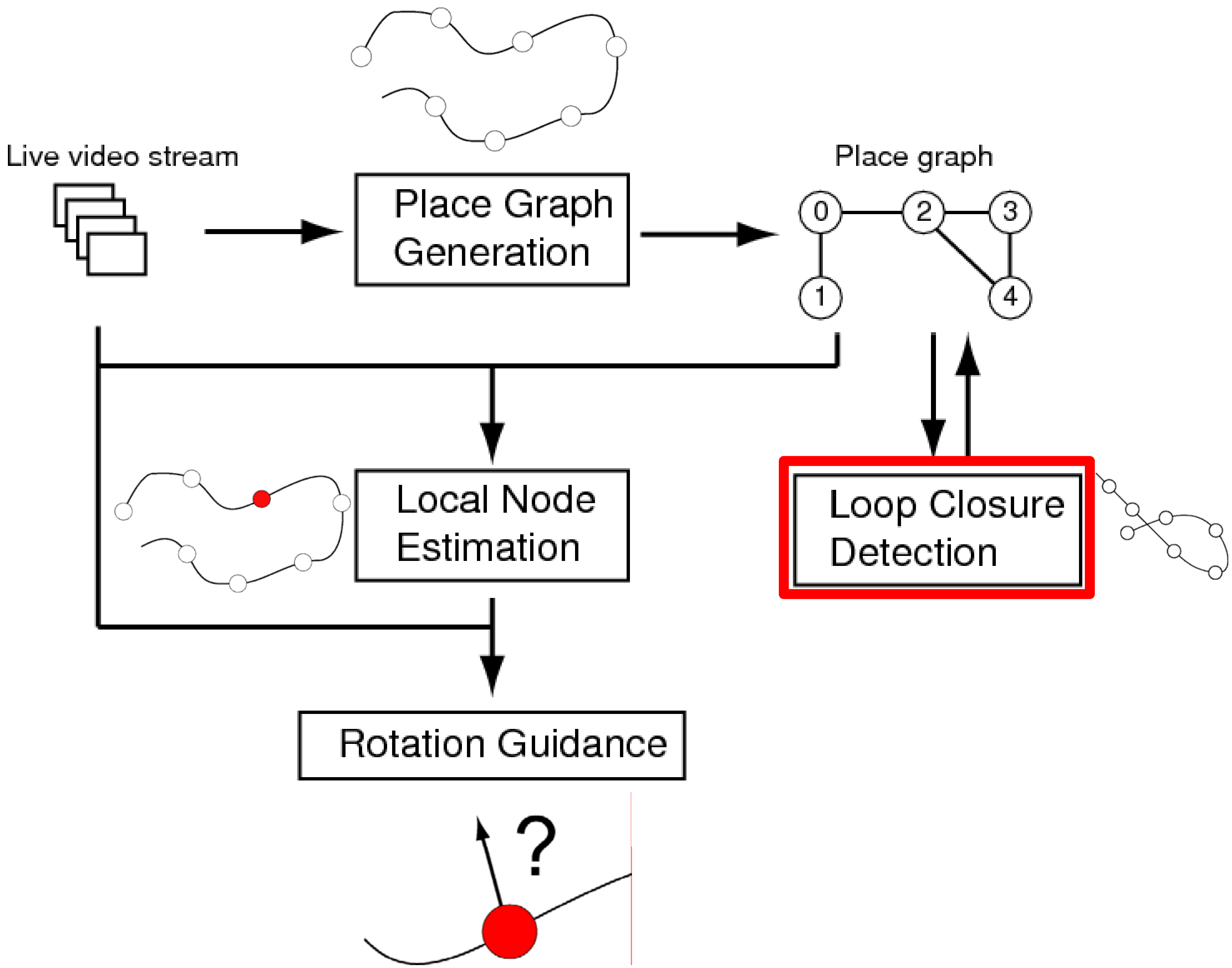
Learning the match matrix

Rotation guidance using the match matrix



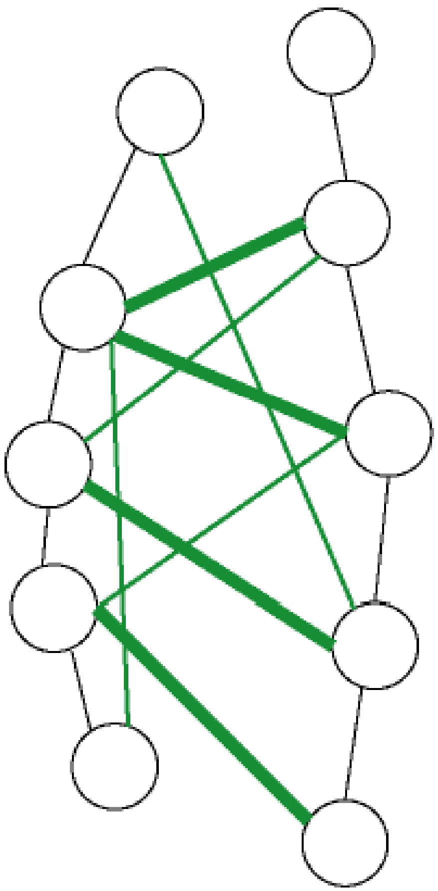
Rotation guidance using the match matrix

Method Overview

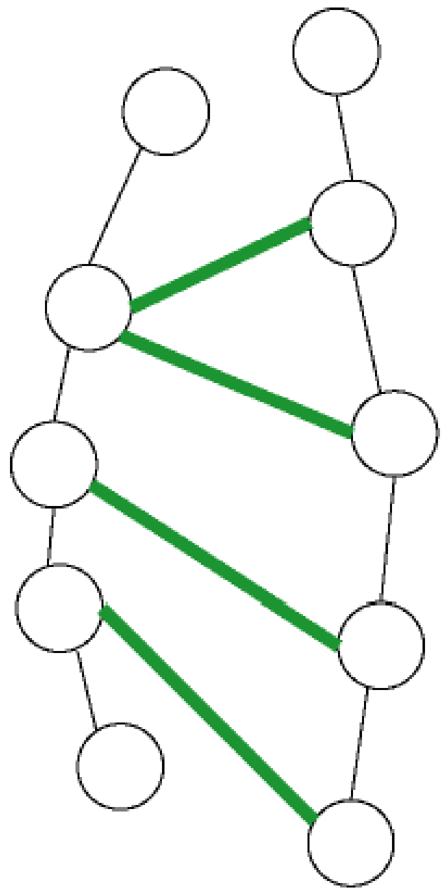


Loop closure detection

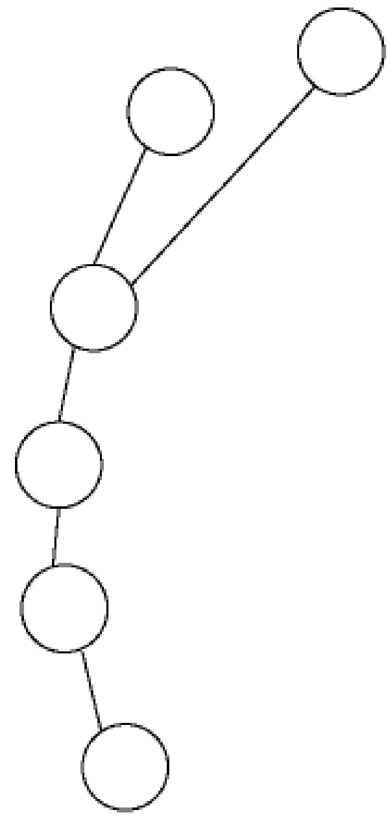
1. Compute node similarity



2. Extract similar sequences

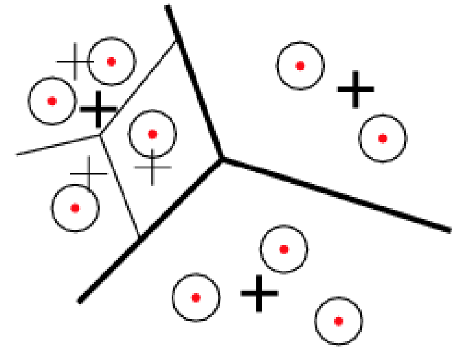


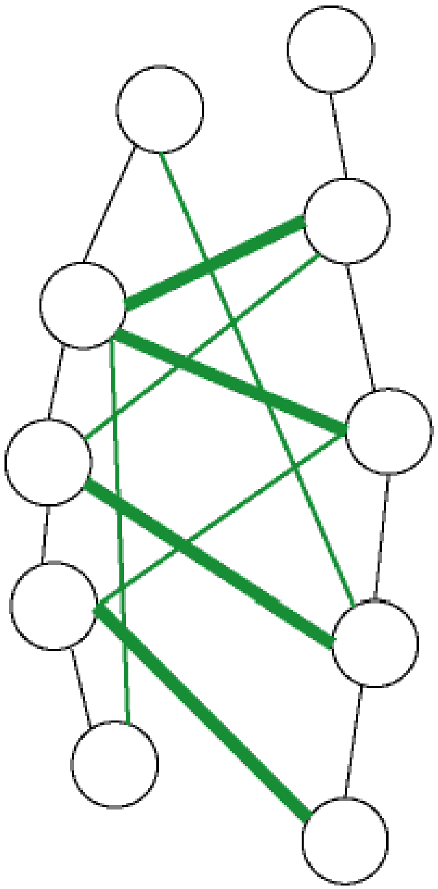
3. Update place graph



Loop closure detection

1. Compute node similarity





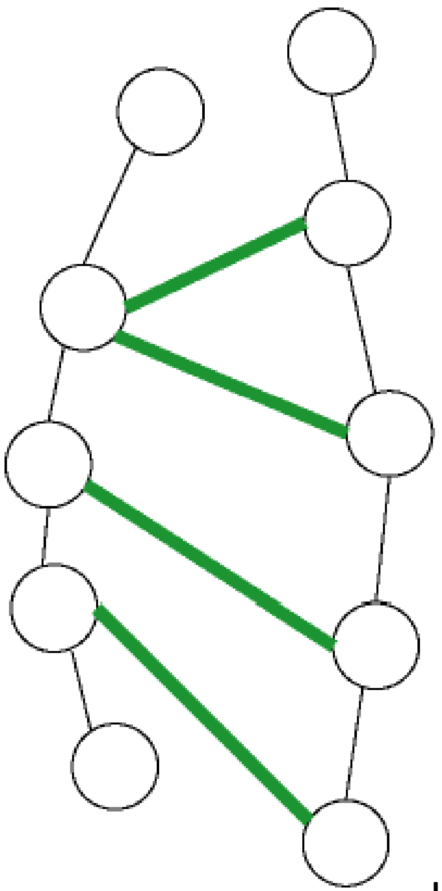
- "Bags of words"
  word w = (c_w, r_w)
  words store list of node labels
- Incremental vocabulary
- Optimized search using search tree

✓ Fully incremental
✓ No a priori vocabulary

Filliat, Interactive Learning of Visual Topological Navigation, IROS'08

Loop closure detection

2. Extract similar sequences



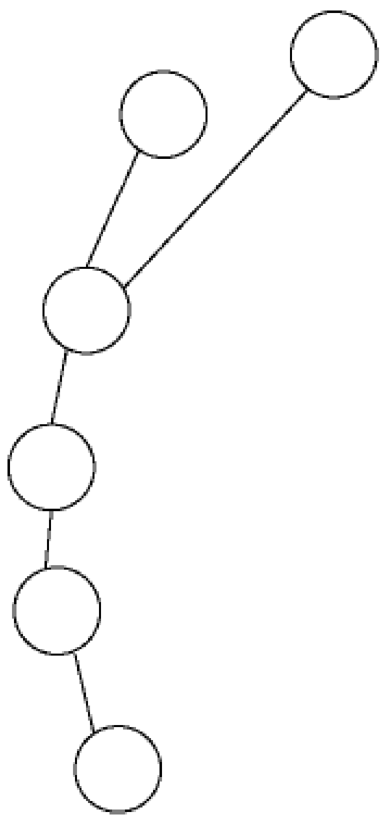
- Smith & Waterman algorithm
- Inspired from molecular biology
- Output: similar node subsequences

✓ Robust loop closure detection
➢ Does not detect “instantaneous” loop closure

Ho & Newman, Detecting Loop Closure with Scene Sequences, IJCV’07

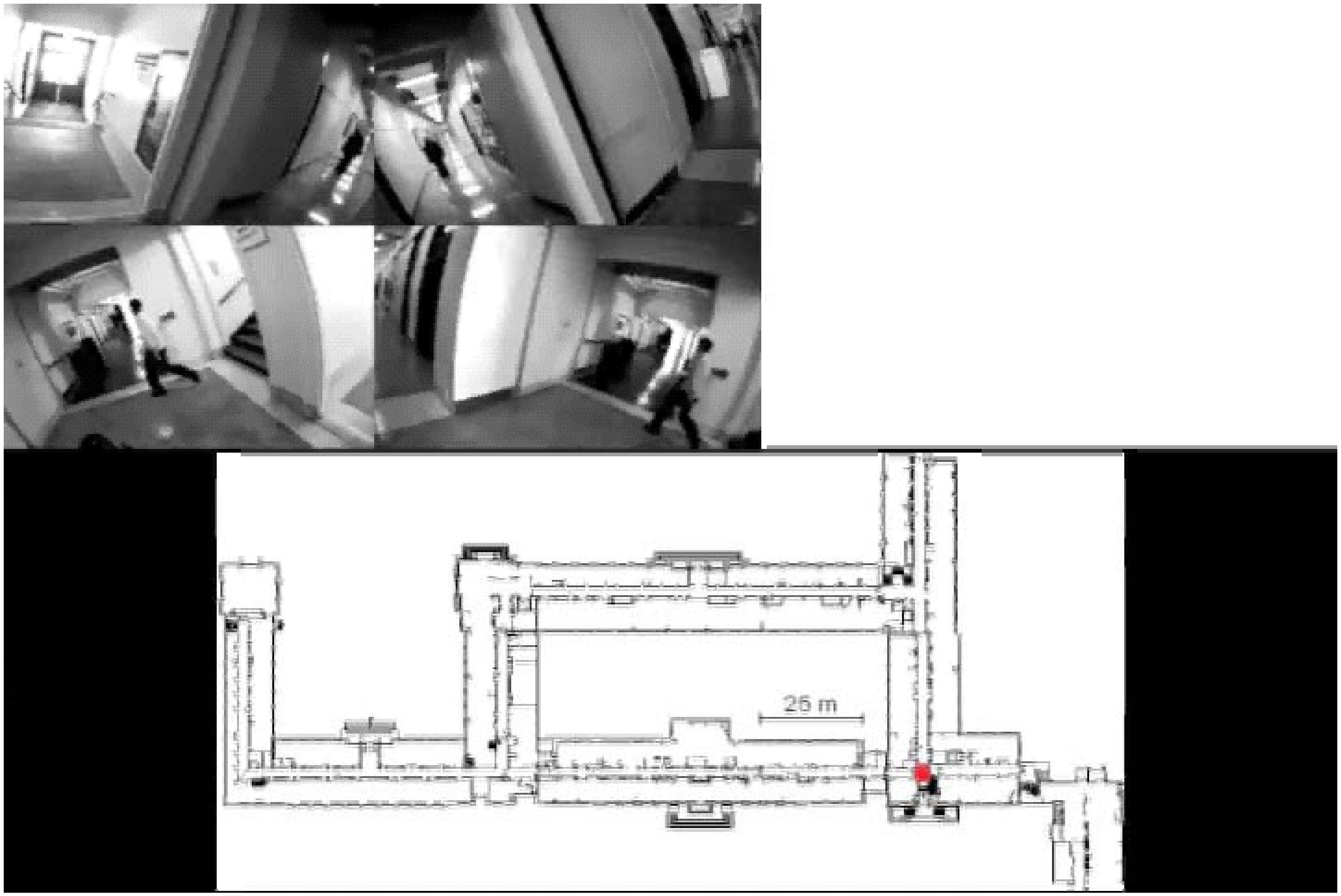
Loop closure detection

3. Update place graph



- Merge node sequences

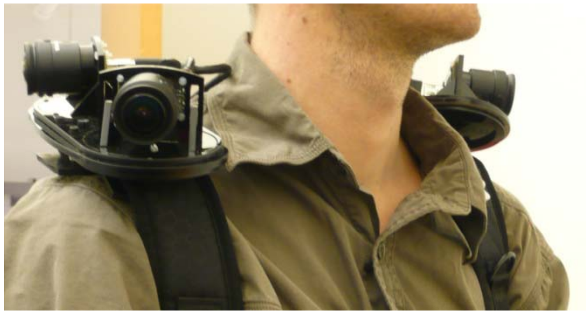
Loop closure detection



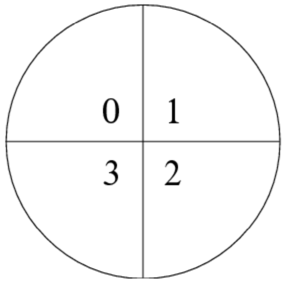
Match matrix

- Anti-symmetry: error ~ 14.5°
- Circular equality: error ~ 1.5°

$$H = \begin{bmatrix} -19.9 & 91.3 & -164.7 & -66.9 \\ -101.8 & -11.9 & 101.4 & -151.5 \\ 155.1 & -95.9 & -16.2 & 105.9 \\ 59.9 & 164.1 & -93.4 & -6.7 \end{bmatrix}$$



$$H_0 = \begin{pmatrix} 0 & 90 & 180 & -90 \\ -90 & 0 & 90 & 180 \\ 180 & -90 & 0 & 90 \\ 90 & 180 & -90 & 0 \end{pmatrix}$$

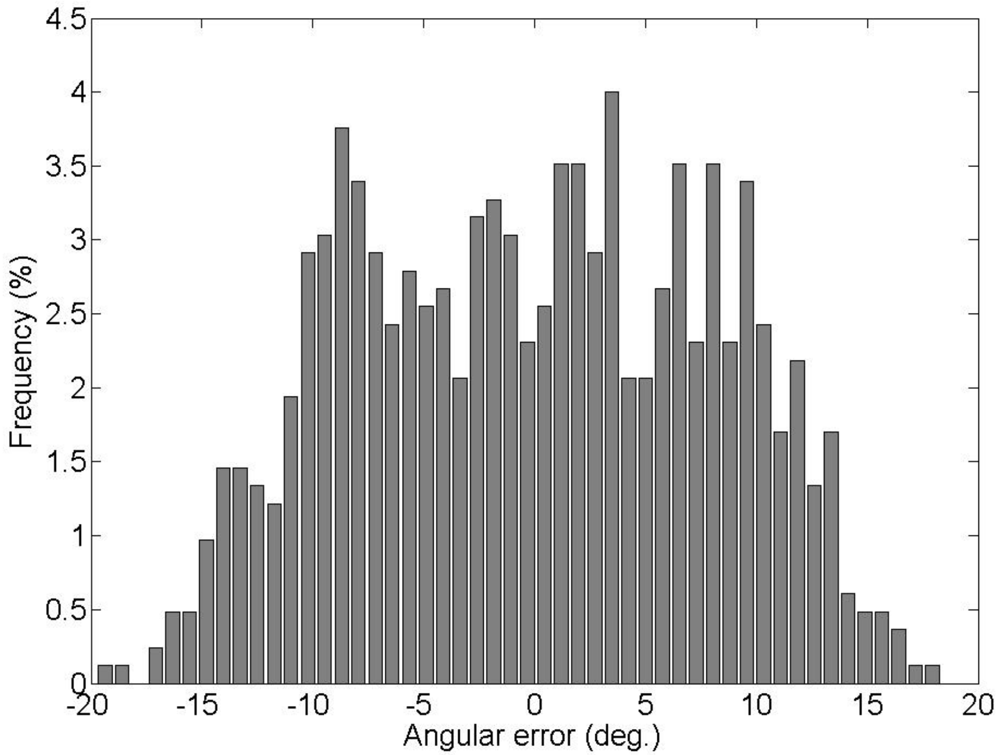


Potential sources of error

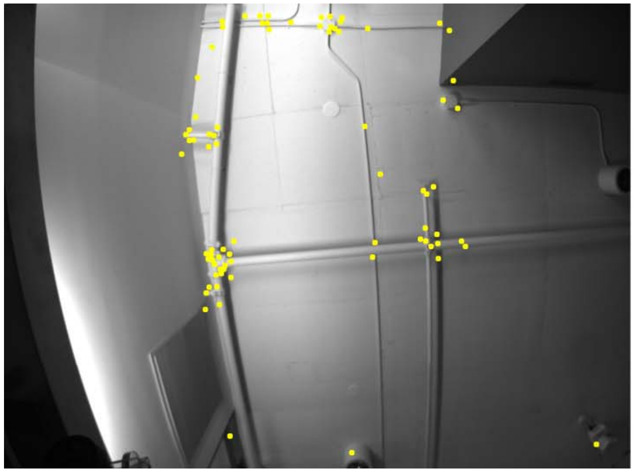
- Constant rotation speed during training
- Non-homogeneous feature distribution in image space
- Baseline due to translation during revisit
- Feature mismatches

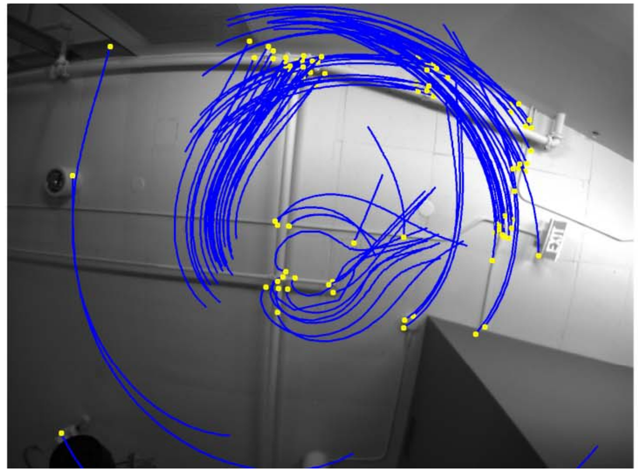
Rotation guidance vs IMU

- User rotates in place in arbitrary environment
- Compare rotation guidance against IMU
- Standard deviation: 8.5° (max error: 20°)



Large-scale rotation baseline

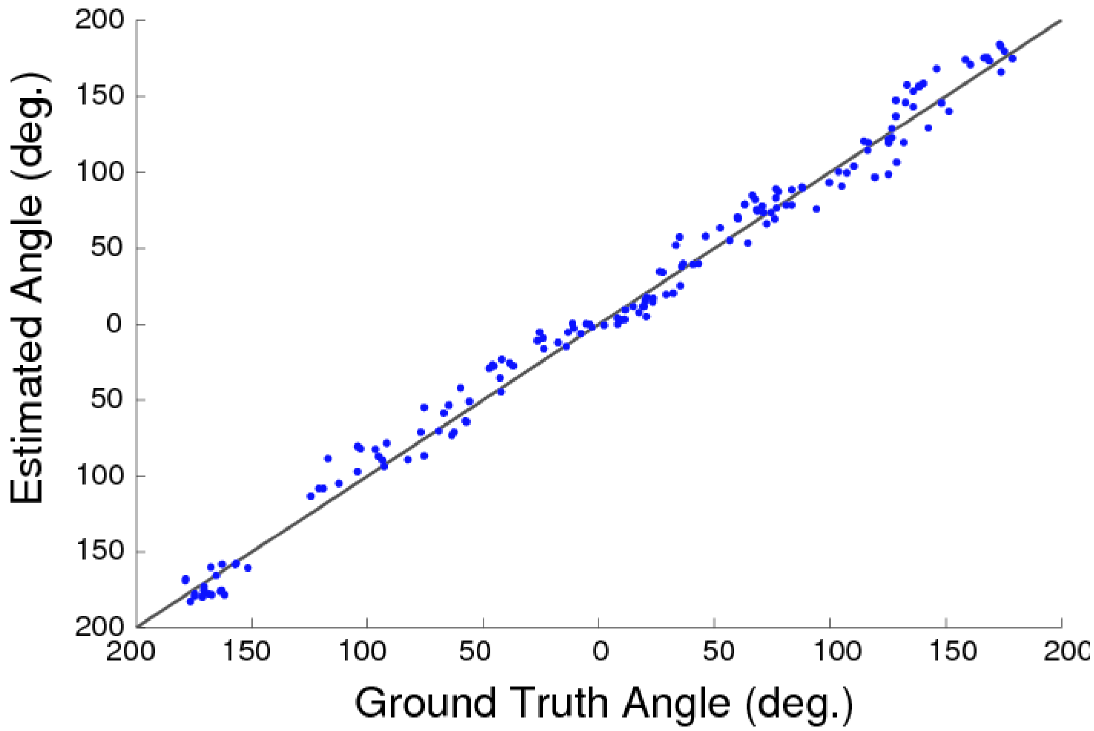

First image


Second image (matches in blue)


Aligned images

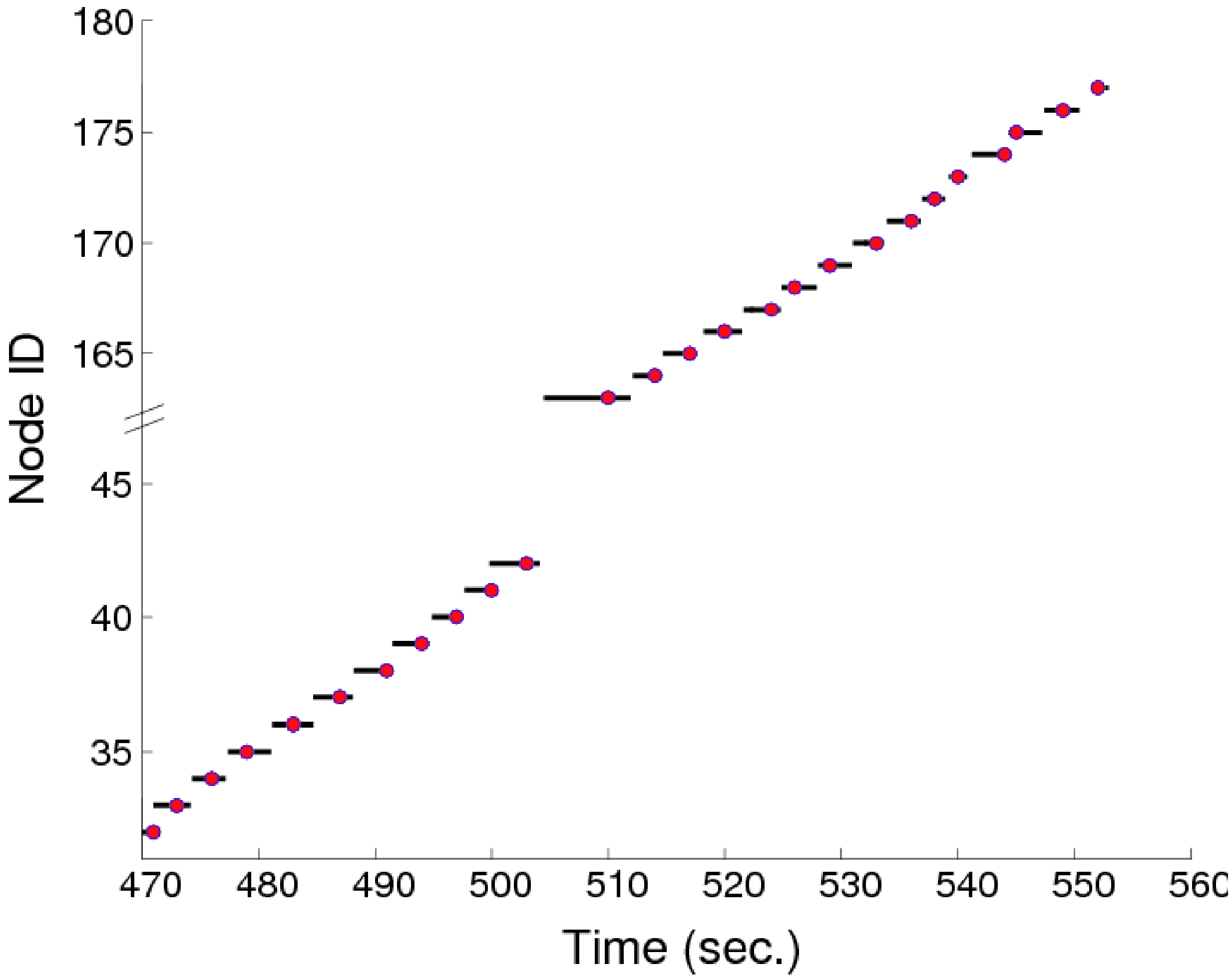
- High-resolution camera pointing upward
- Average difference between SIFT feature orientations
- Standard error (vs IMU) < 2°
- Ground-truth <u>throughout exploration path</u>
- Requires no intrinsic/extrinsic camera calibration

Rotation guidance vs ground-truth



- 200 checkpoints
- Standard error: 10.5° (max error: 15°)

Local node estimation vs ground-truth



Real-world explorations

Name	Scenario	Duration	Length	# frames	# nodes	# checkpoints
MEZZANINE	replay	10 min.	400m	6,000	91	36
GALLERIA	homing	15 min.	700m	9,000	154	150
CORRIDORS	point-to-point	30 min.	1,500m	18,000	197	0

GALLERIA dataset

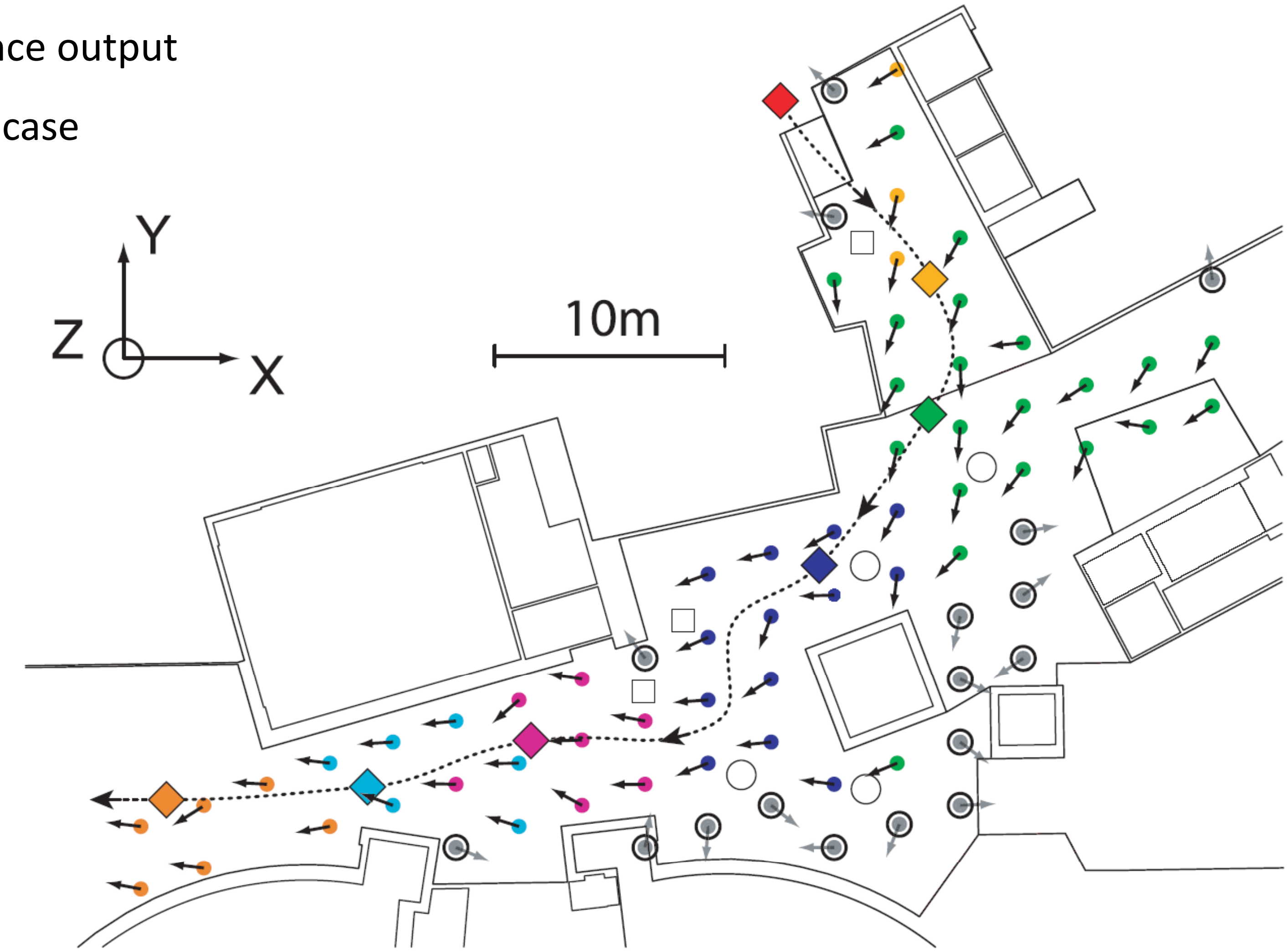
CORRIDORS dataset

Off-path trajectories (GALLERIA dataset)

Place graph node

Guidance output

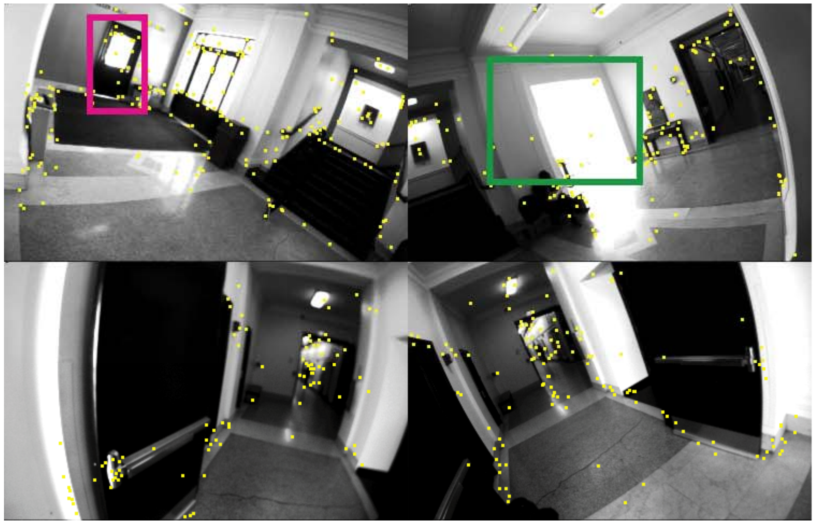
Failure case

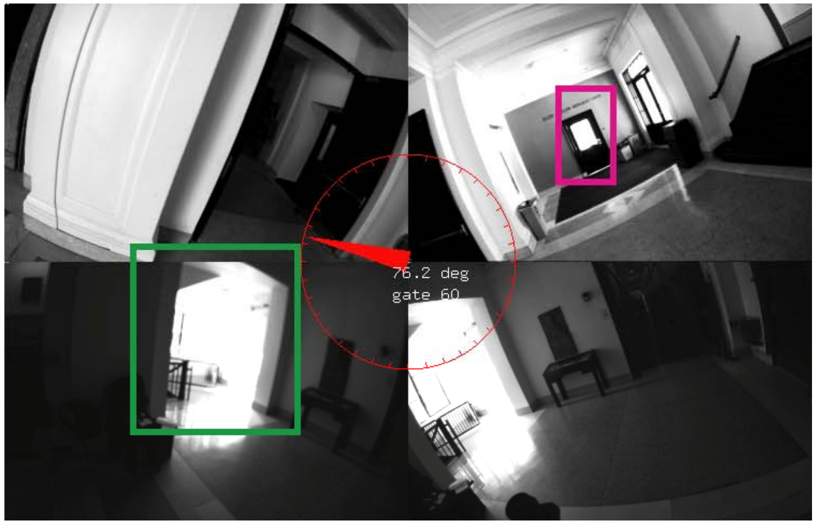


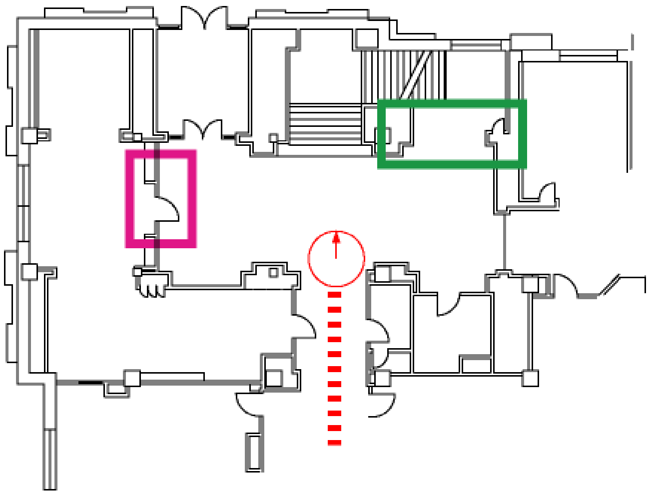
Rotation guidance overlaid on 2D map. Values are exact, not notional.

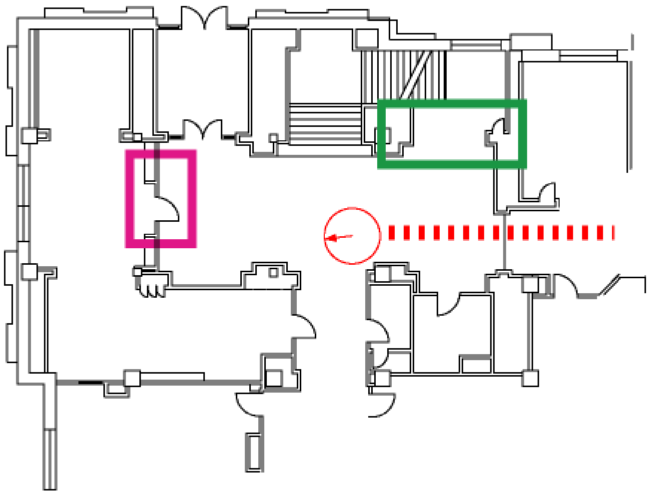
Off-path trajectories (CORRIDORS dataset)

Rotation guidance (CORRIDORS dataset)

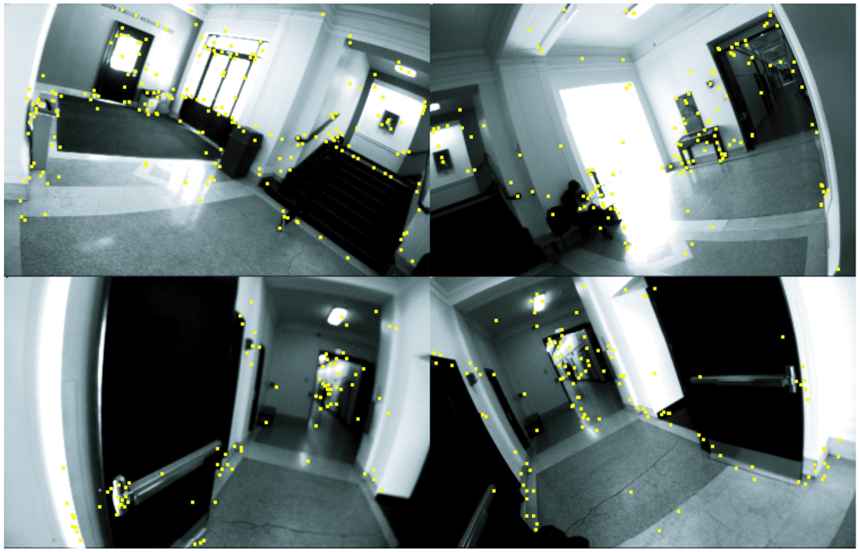




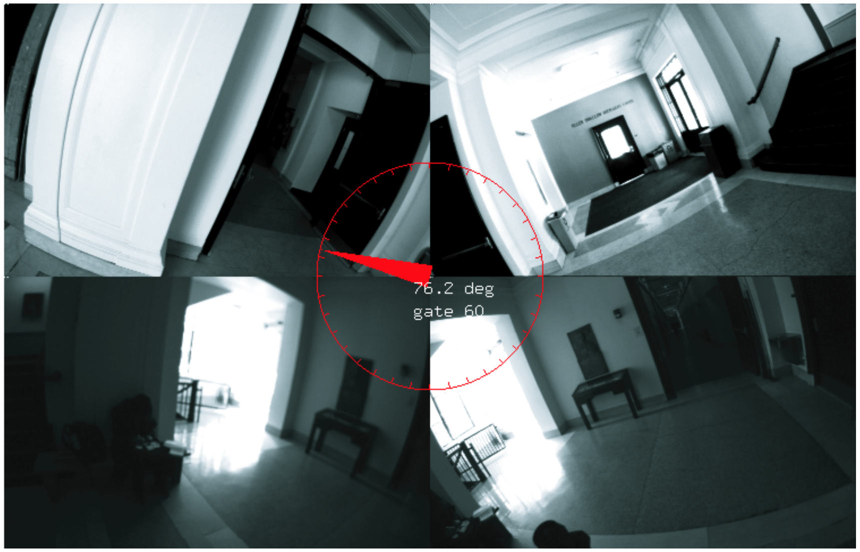




Rotation guidance (CORRIDORS dataset)

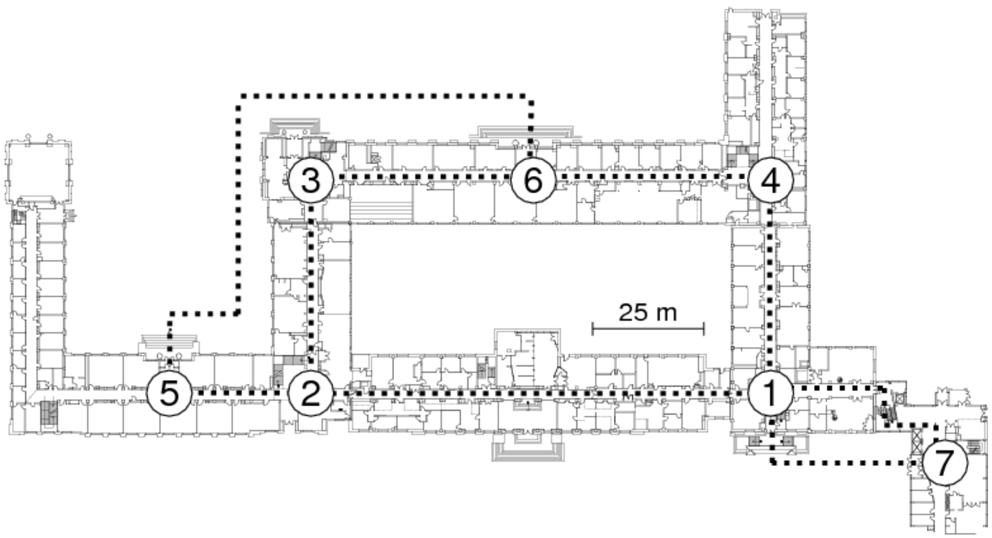


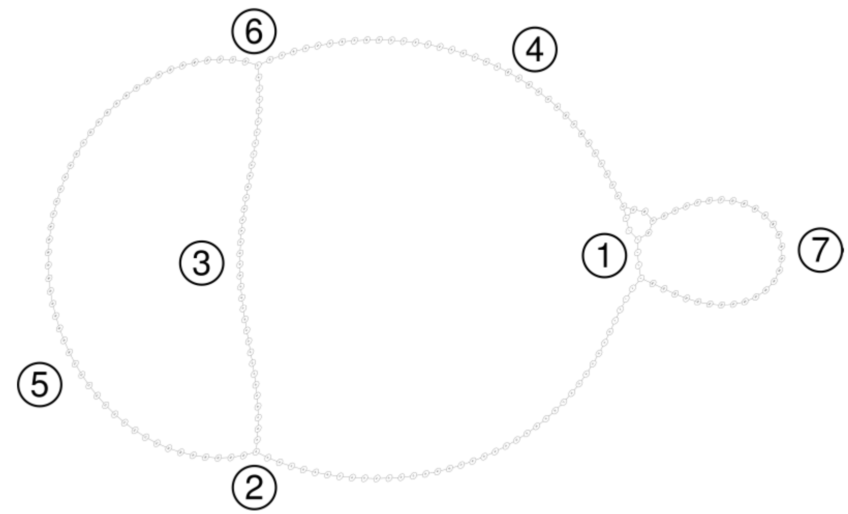
First visit
(SIFT features in yellow)



Revisit
(body-relative rotation guidance)

Loop-closure (CORRIDORS dataset)





Exploration path manually overlaid on 2D map
1,500 meters (30 min.)

Place graph (spring-mass model)
500 nodes (before loop closure)
197 nodes (after)

Conclusion

Assumptions

- ➢ Large number of visual features visible at all time
- ➢ Uniform distribution of observations in image space
- ➢ Rigid-body transformation between cameras is fixed but can change slightly
- ➢ Training phase (short, once for a camera configuration)

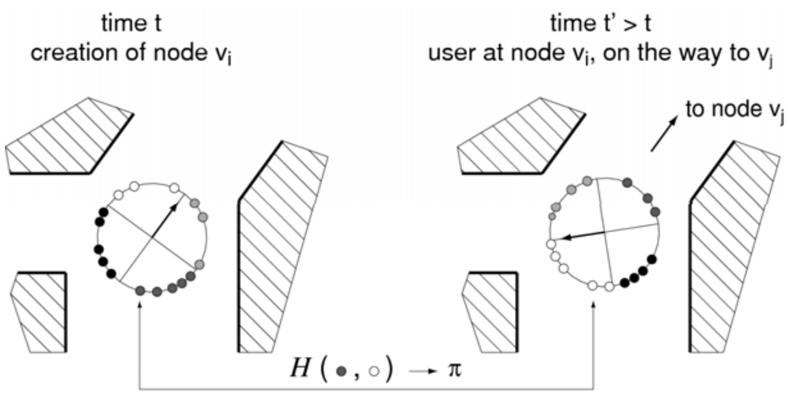
Advantages

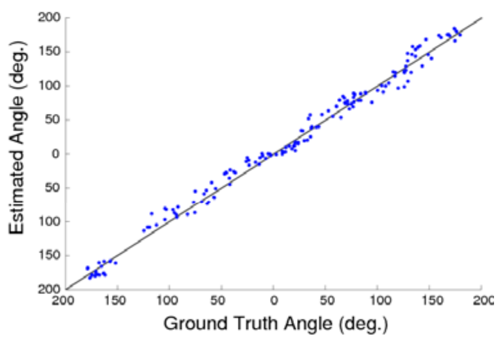
- ✓ Requires no extrinsic or intrinsic camera calibration
- ✓ Scales to large environments (several km)
- ✓ Provides guidance in the user's body frame
- ✓ Robust to off-path trajectories and high-frequency user motion

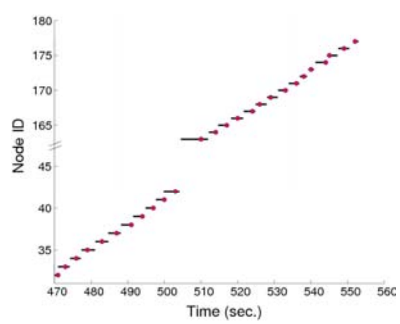
Future Work

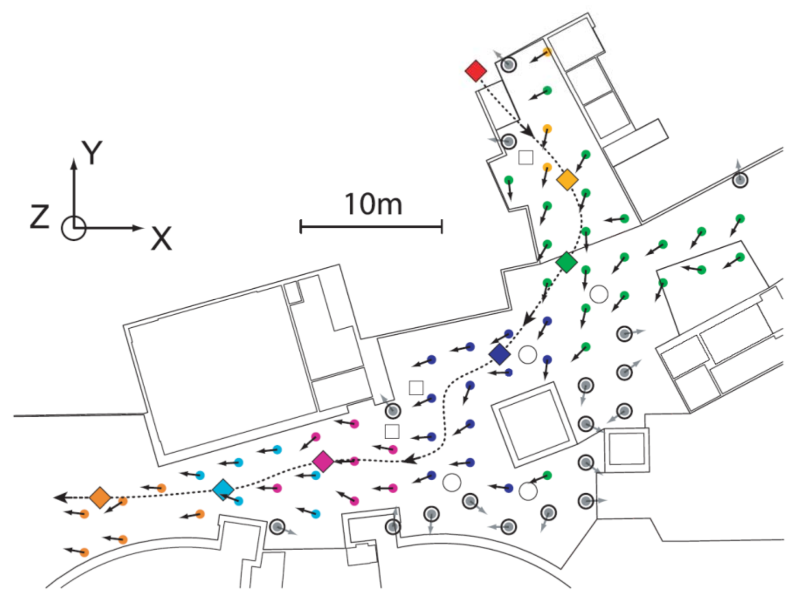
- Extend to 3D motion (stair ascent/descent)
- User study on multiple real human users
- Application to robotics

Questions


time t
creation of node v_i
time t' > t
user at node v_i, on the way to v_j
to node v_j
$H(\bullet, \circ) \rightarrow \pi$


Estimated Angle (deg.)
Ground Truth Angle (deg.)


Node ID
Time (sec.)


Y
Z
X
10m