# Autonomous Lifelong Navigation, Experiment Replication and Benchmarking

Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti,
Bastian Steder, Rainer Kümmerle, Christian Dornhege,
Michael Ruhnke, Alexander Kleiner, Juan D. Tardos

University of Freiburg, Germany and University of Zaragoza, Spain

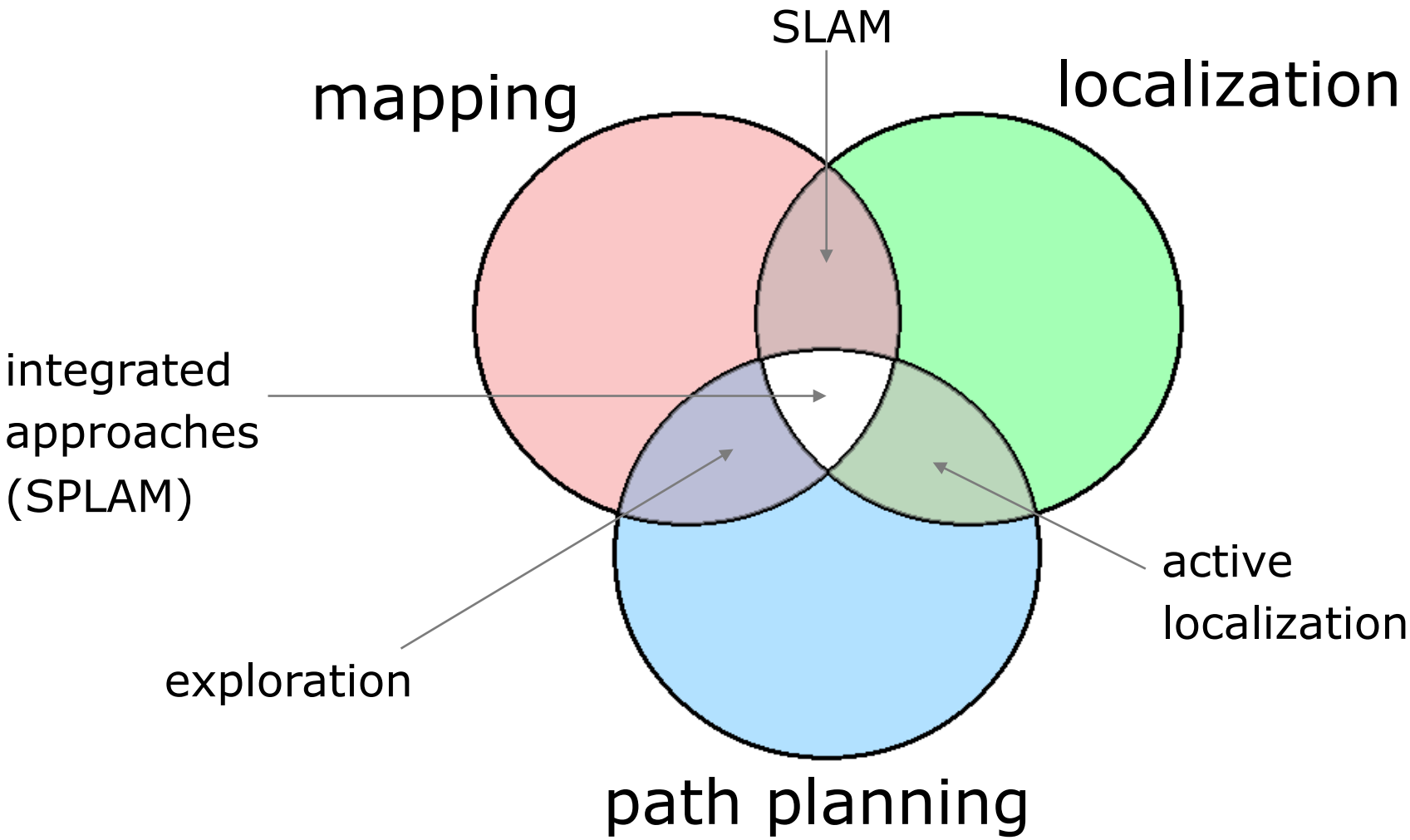
# Motivation

- No gold standard to evaluate and compare the results of navigation systems

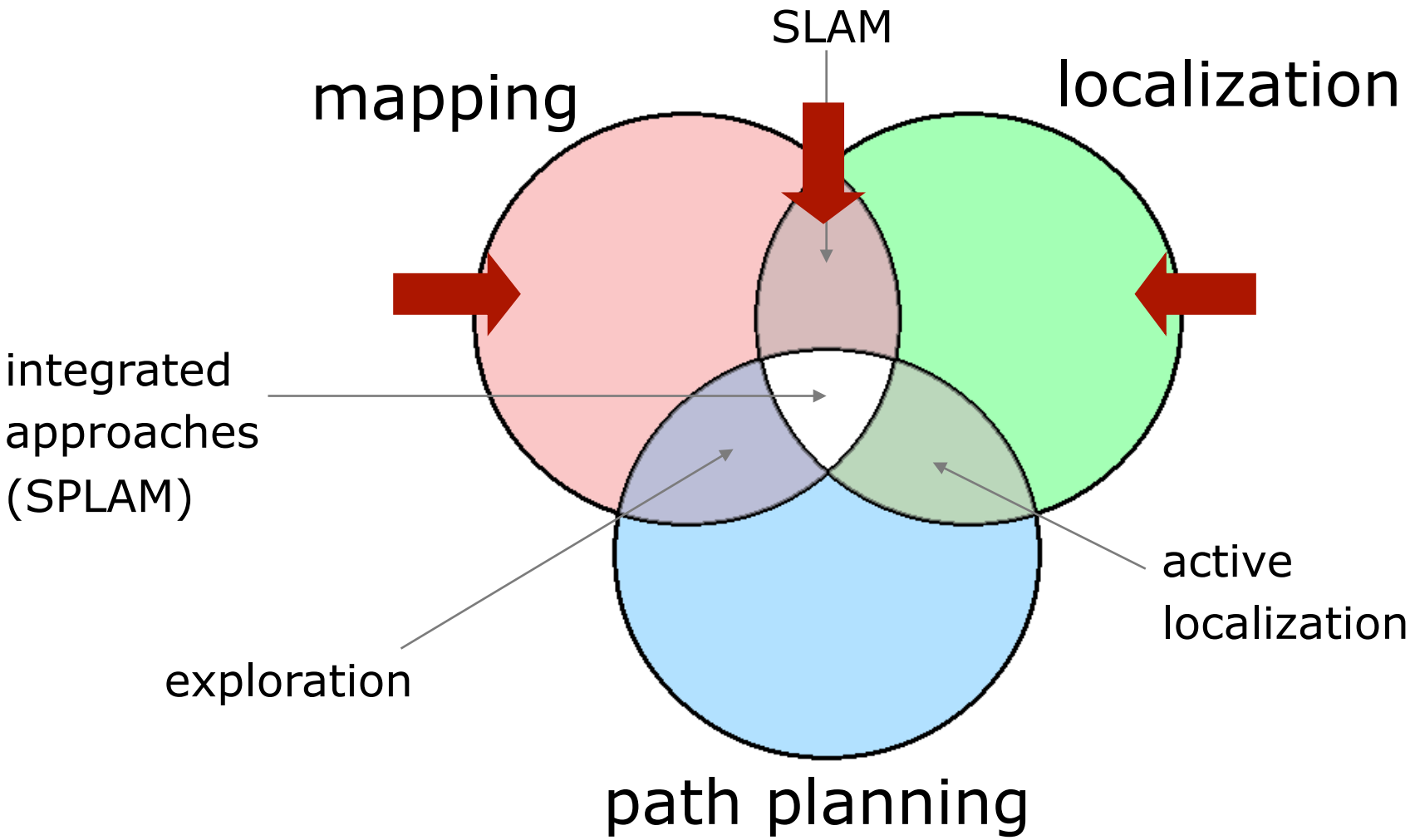
- Many measures can characterize a navigation system, for instance: position accuracy (local), execution time, energy efficiency.

- However,
  - A navigation system is a closed-loop-system,
  - the actions taken influence the next observations, and
  - a **complete** system can be evaluated only by either real-world experiments, or by using highly accurate simulators.

# Taxonomy of Navigation



# Taxonomy of Navigation



# Related Work

- SLAM approaches
  (Smith&Cheeseman, Lu&Milios, Frese, Duckett et al., Dissanayake et al., Howard et al., Eustice et al., Grisetti et al., Stachniss et al., Olson, …)

- Competitions
  (DGC, RoboCup, NIST, …)

- Data repositories
  (Radish, RAWSEEDS, OpenSLAM, …)

# Benchmarking Lifelong Perception for Navigation

- The datasets for comparison can be arbitrary large
  - Use realistic simulators to produce these datasets
  - **Generate realistic long datasets by combining real data**
- The algorithms should cope with additional issues:
  - Changes in the environment
  - Impossibility to store all data of the robot

# Measures

- Uncertainty
  (Entropy, KL-Divergence, …)

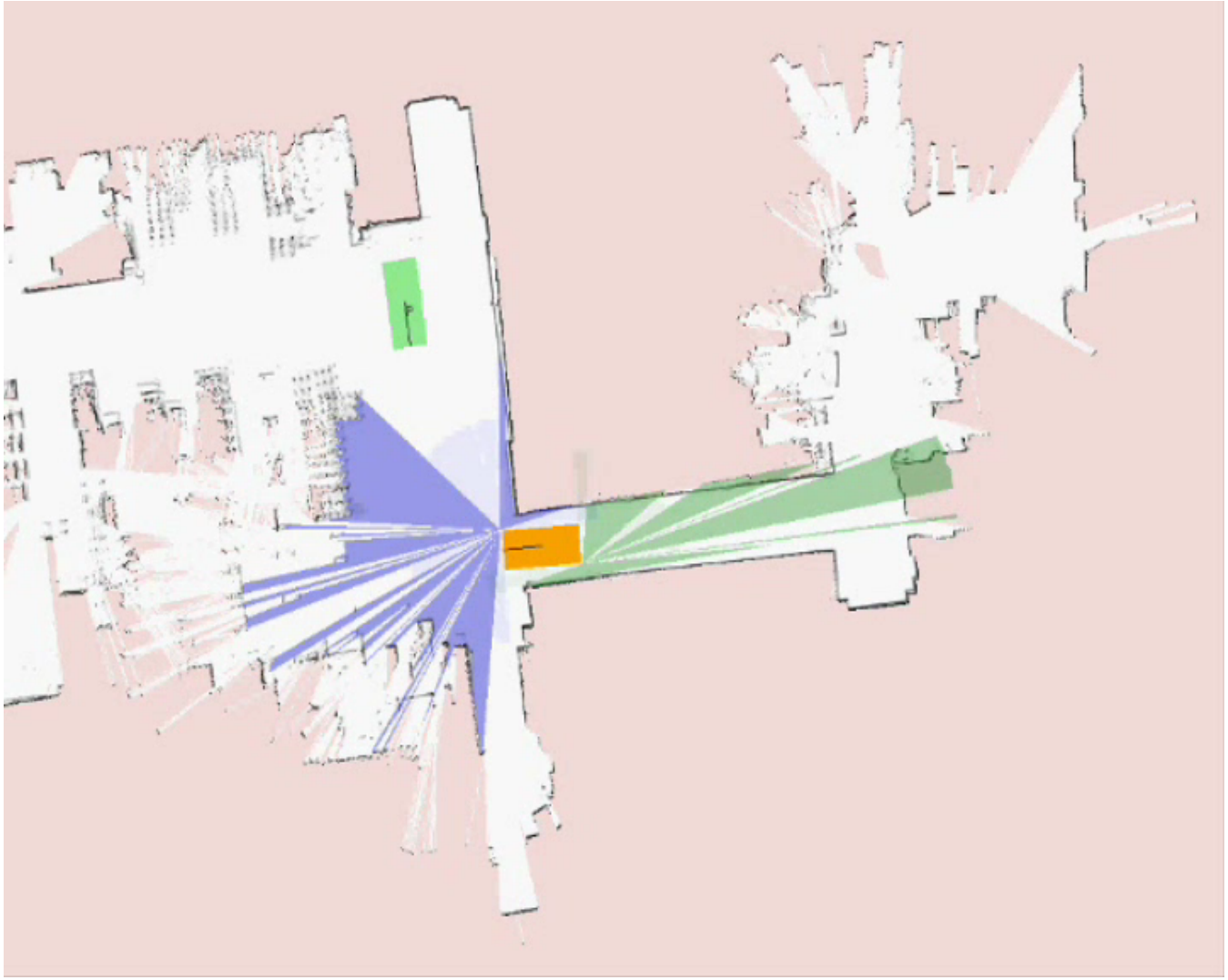
- Variance of estimators
  (Cramer-Rao-bound, Fisher information, …)

- Quality of estimates
  (observation likelihood, normalized estimation error squared, distance of relative observations, …)

# Practical Applications

# Practical Applications

# Practical Applications



# Benchmarking Perception for Navigation

- The robot should be accurately localized in its local surrounding
- The map should be topologically consistent
- The reaction times (latency) should be small enough to prevent collisions

- The robot should be maximally successful!

# Questions?

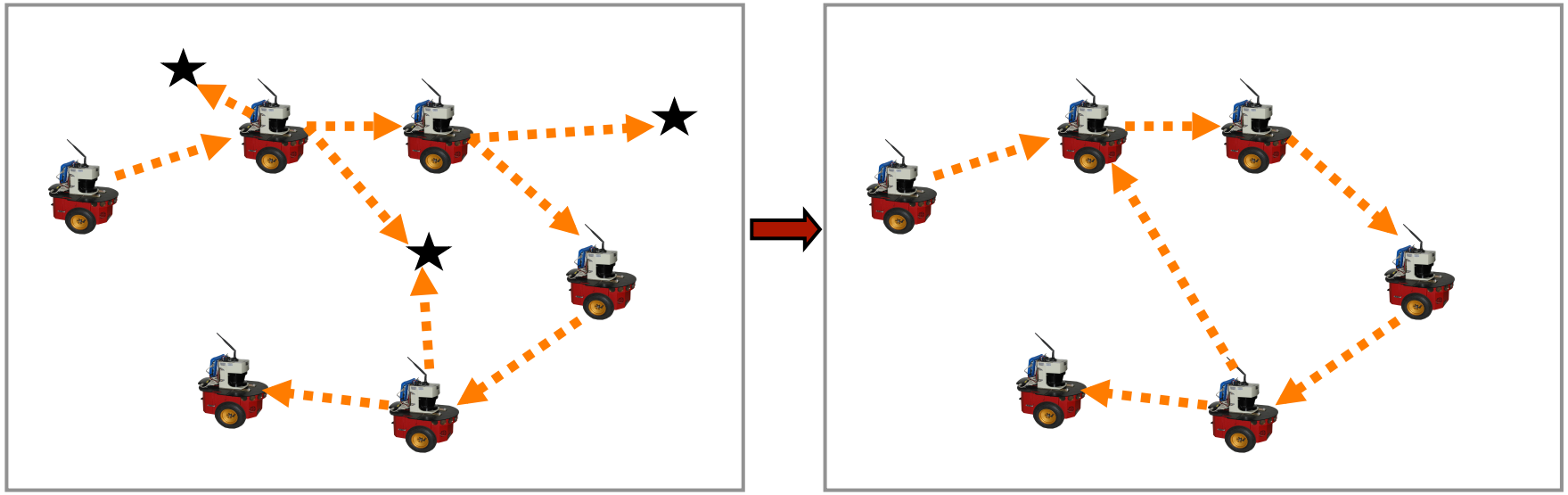
- How to measure the local consistency?
  - should be sensor independent
- How to generate arbitrary long sequences from real world data such that the local consistency of a SLAM method can be measured?
  - should minimize the human effort
  - should minimize the data to be transmitted
  - should encode the ground truth

# What do we get from SLAM?

- All SLAM approaches are at least able to estimate the current robot position $\boldsymbol{x}_t$
- Smoothing approaches estimate the trajectory $\boldsymbol{x}_{0:t}$
- Classical filtering approaches estimate only the map (landmark positions) $\boldsymbol{l}_{0:k}$
- The two representations are dual:
  - From the map one can recover the trajectory by localization
  - From the trajectory one can recover the map via mapping with known poses.
- We focus on the robot's trajectory because it is sensor independent and directly relates to the localization accuracy

# Pose Graph

- A pose graph encodes the poses of the robot during mapping as well as constraints resulting from observations.
- It is independent of the kind of observations
- and the type of the map (landmarks, grids, …).



Robot poses and observations

Pose graph

# Our Approach

- measures the relative errors between nearby poses

$$\mathbf{e}_{i,j} = (\mathbf{x}_j \ominus \mathbf{x}_i) \ominus \delta^*_{i,j}$$

estimated relative movement

ground truth

- which is exactly what approaches to graph-based SLAM seek to minimize.

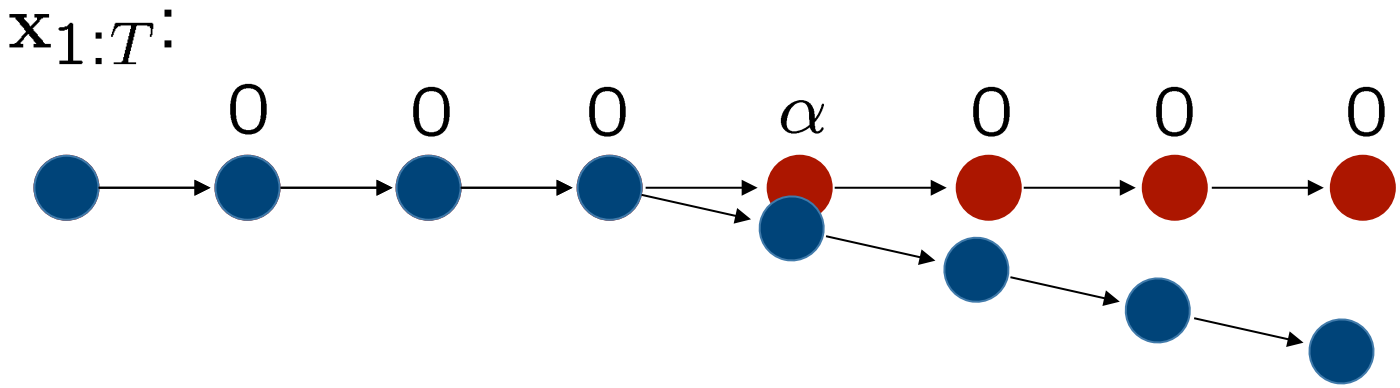
# Metric Based on Relative Poses

$$\varepsilon(\delta) = \frac{1}{N} \sum_{i,j} \mathbf{e}_{i,j}^T \boldsymbol{\Omega}_{i,j} \mathbf{e}_{i,j}$$

Uncertainty of the ground truth

- Which is the Chi-square error of the trajectory w.r.t. the ground truth measurements.

# Application to the Example



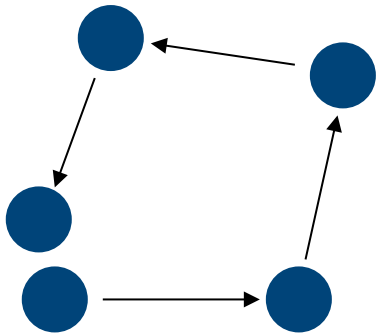

$$\varepsilon(\mathbf{x}_{1:T}) = \alpha$$

- The one localization error weighs exactly $\alpha$ in the metric. Errors between poses that are far apart appear less since they do not cause localization problems.

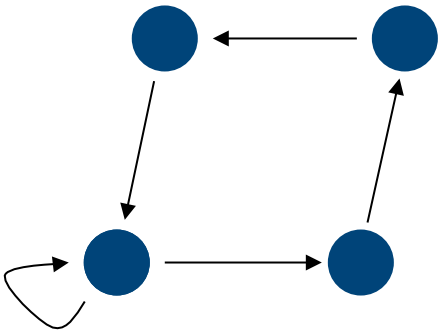
$$\Omega_* = \mathbf{I}$$

# Selection of Relations

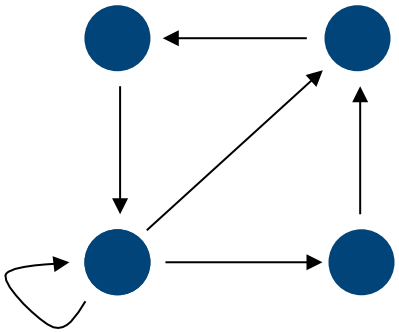
- No loop closure links:
  local consistency only
  (wrt. time)



- Loop closure links:
  local consistency
  (wrt. space)



- Links between far
  away nodes:
  global consistency

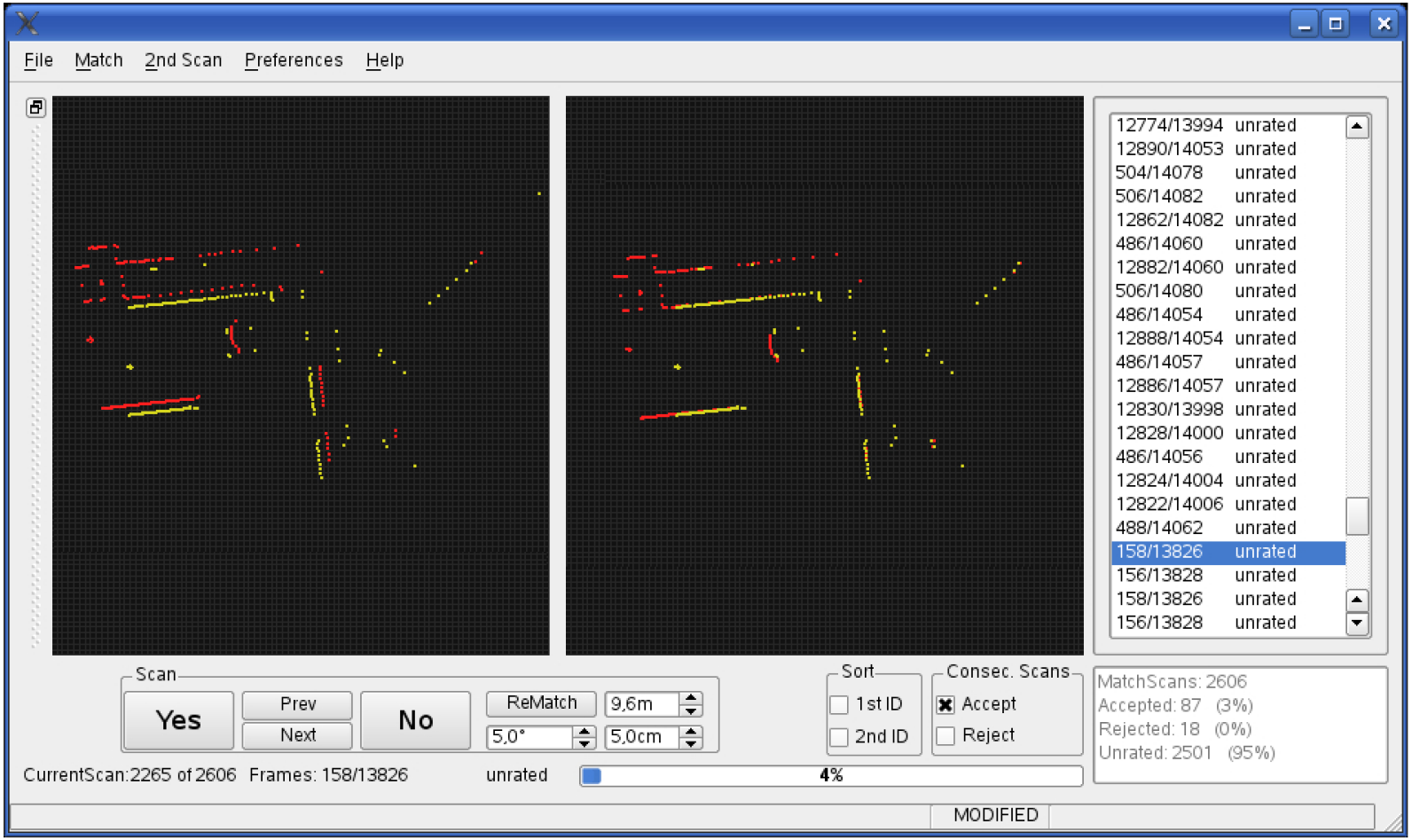


# Obtaining Ground Truth Relations

- Besides in simulation, true relations are hard to obtain
- Highly accurate measurement device (e.g., Simeo positioning system or any other sources)
- How can we use existing datasets for evaluations (Intel Research, MIT infinite corridor, …)?
- For laser-based SLAM, perform scan alignment paired with manual inspection

# Graphical Interface to Confirm & Correct Constraints

- Scan alignment paired with manual inspection to generate close to ground truth relations



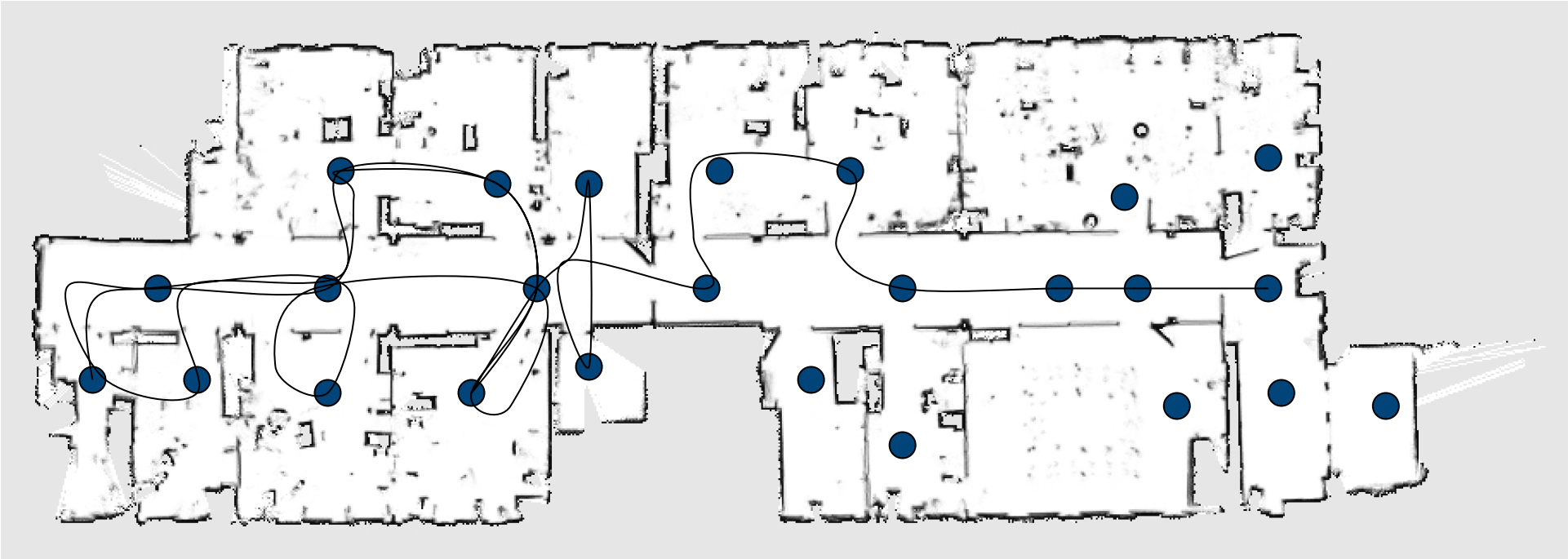
# Arbitrary Long Real World Datasets

- It is impossible to have a robot running forever
- The sequence of data needs to capture the relevant aspects of the phenomenon
  - Temporal changes:
    - day/night transition
    - moving objects
    - changes in the structure
- Obtaining ground truth should be as easy as possible
- Using **real world data**

# Acquiring the Initial Data

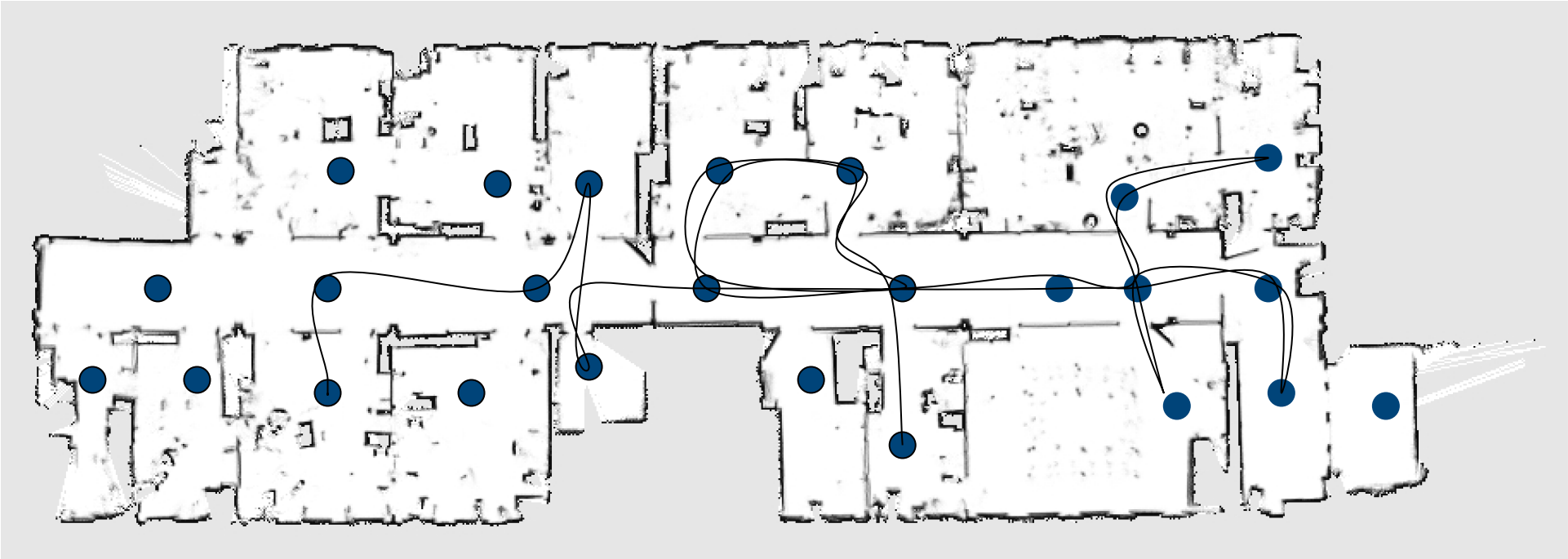
- Select a (large) set of positions in the environment, potentially marking them with a unique landmark (e.g., based on RFID).

# Acquiring the Initial Data



- Select a (large) set of positions in the environment, potentially marking them with a unique landmark (e.g., based on RFID).
- Acquire a set of datasets during several periods of the day.
- All datasets should **start** and **end** from one of the chosen positions, and traverse as many as possible.

# Acquiring the Initial Data



- Select a (large) set of positions in the environment, potentially marking them with a unique landmark (e.g., based on RFID).
- Acquire a set of datasets during several periods of the day.
- All datasets should **start** and **end** from one of the chosen positions, and traverse as many as possible.

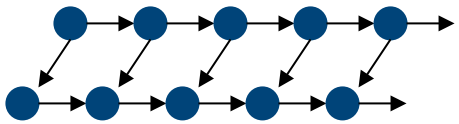
# Acquiring the Initial Data

- Select a (large) set of positions in the environment, potentially marking them with a unique landmark (e.g., based on RFID).
- Acquire a set of datasets during several periods of the day.
- All datasets should **start** and **end** from one of the chosen positions, and traverse as many as possible.

# Obtaining Ground Truth

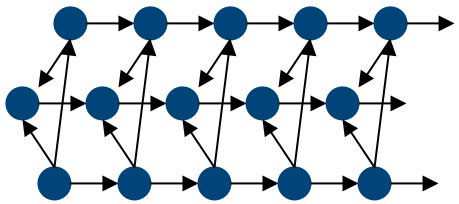
- Doing the manual alignment for each pair of nearby nodes has a cost up to O(N^2), where N is total length of all trajectories

# Obtaining Ground Truth



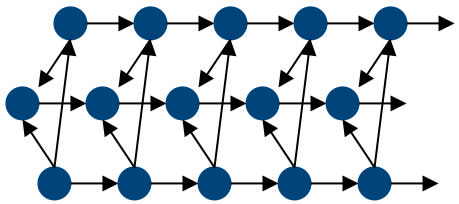
- Doing the manual alignment for each pair of nearby nodes has a cost up to O(N^2), where N is total length of all trajectories

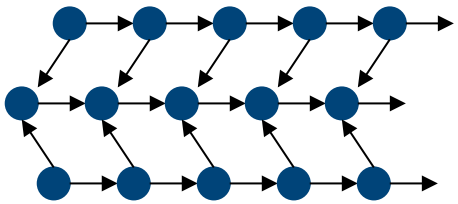
# Obtaining Ground Truth



- Doing the manual alignment for each pair of nearby nodes has a cost up to O(N^2), where N is total length of all trajectories

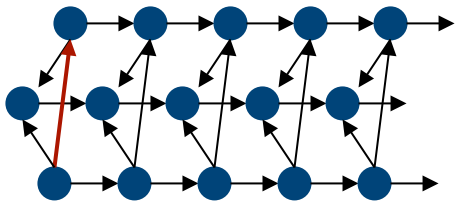
# Obtaining Ground Truth

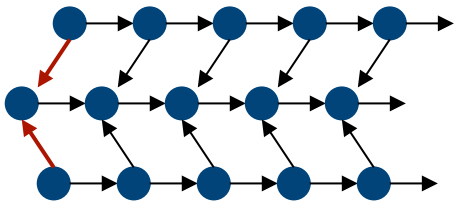




- Doing the manual alignment for each pair of nearby nodes has a cost up to O(N^2), where N is total length of all trajectories
- In fact this can be reduced to a cost more or less linear by
  - Inserting a constraint in a region only between the current node and the oldest one in that region.

# Obtaining Ground Truth





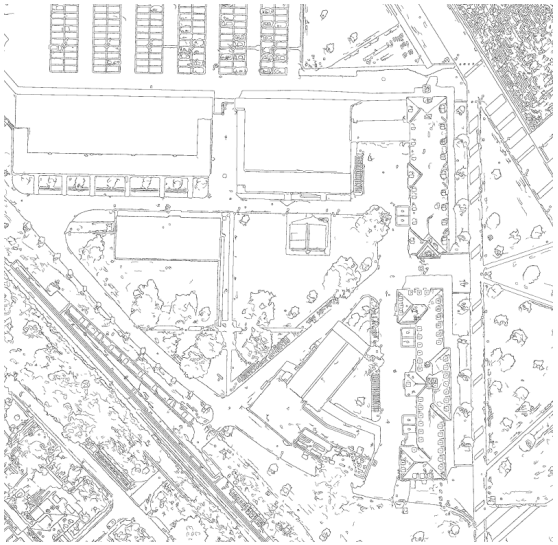
- Doing the manual alignment for each pair of nearby nodes has a cost up to O(N^2), where N is total length of all trajectories
- In fact this can be reduced to a cost more or less linear by
  - Inserting a constraint in a region only between the current node and the oldest one in that region.
  - All other constraints can be obtained by chaining pairs of measurements.
- **Still a lot of work but more feasible**

# What Else?

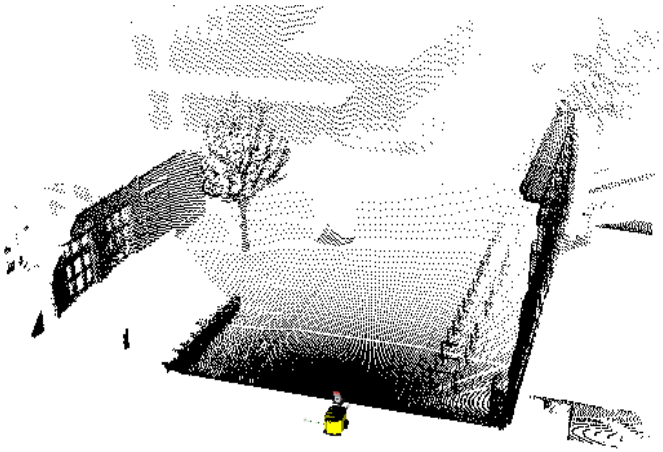
- Statistical tests: Does the algorithm yield a significant improvement over the state of the art?
- Such tests are typically based on a large number of data.
- How can we do this in a life-long setting?

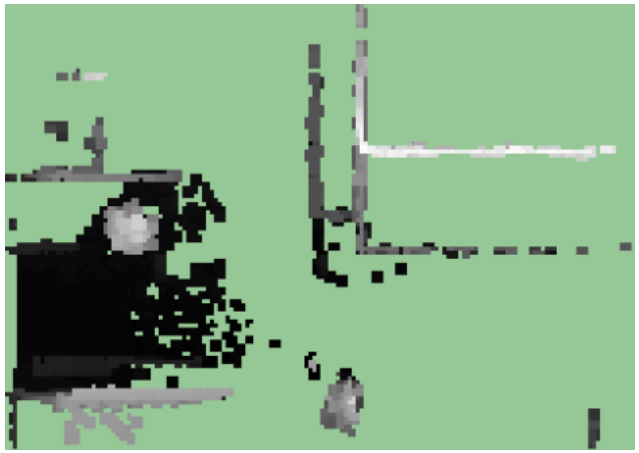
# SLAM using Aerial Images

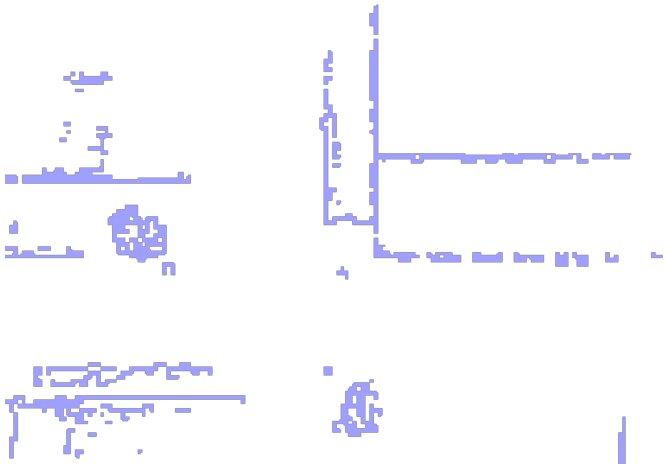
- Changes in color often occur at height variations.
- Canny edge extraction to get a likelihood field.
- Consider different view-points of the wheel-based robot and the aerial image.



# Processing the Sensor Data





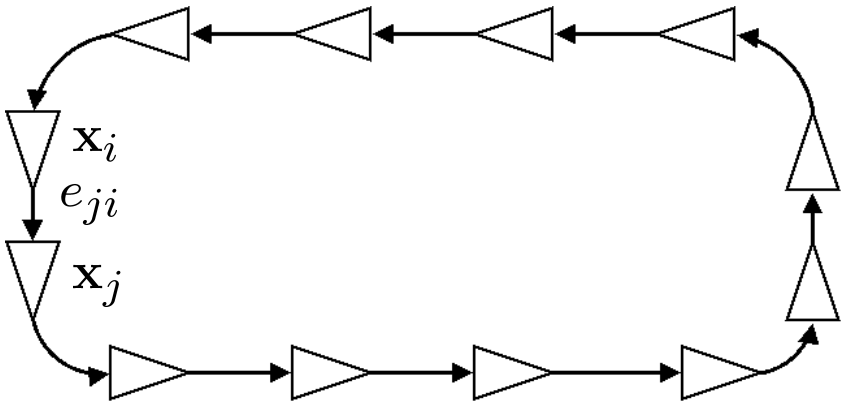


3D range data
Compute the z-buffer
from top
Detect large variations
in height

# Graph-SLAM

- Use a graph structure to represent the SLAM problem.
- Poses as nodes, spatial constraints between the poses as edges.

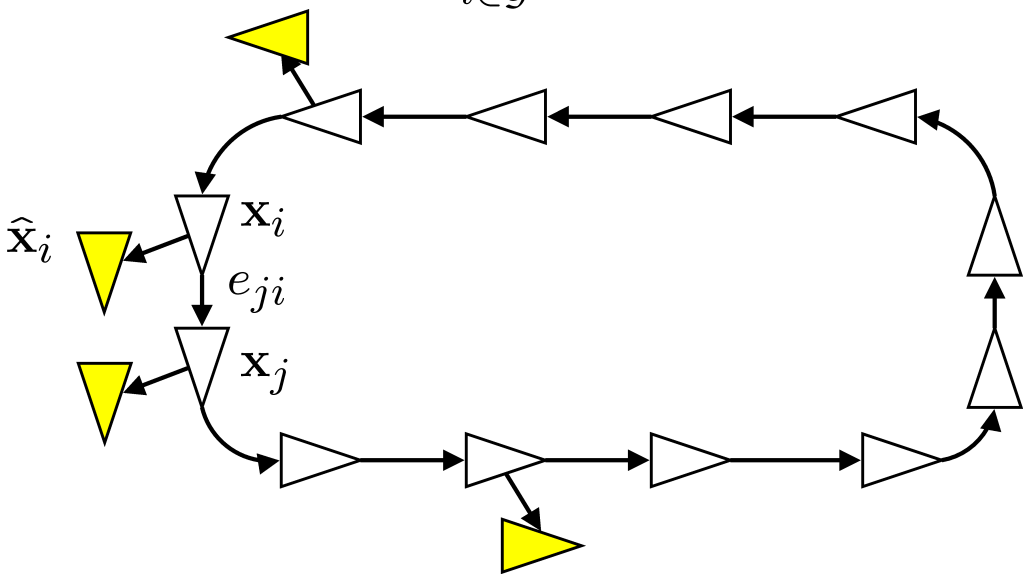
$$(\mathbf{x}_1, \ldots, \mathbf{x}_n)^* = \underset{(\mathbf{x}_1,\ldots,\mathbf{x}_n)}{\text{argmin}} \sum_{\langle j,i \rangle} e_{ji}(\mathbf{x}_i, \mathbf{x}_j)^T \Omega_{ji} e_{ji}(\mathbf{x}_i, \mathbf{x}_j)$$



# Graph-SLAM with Prior

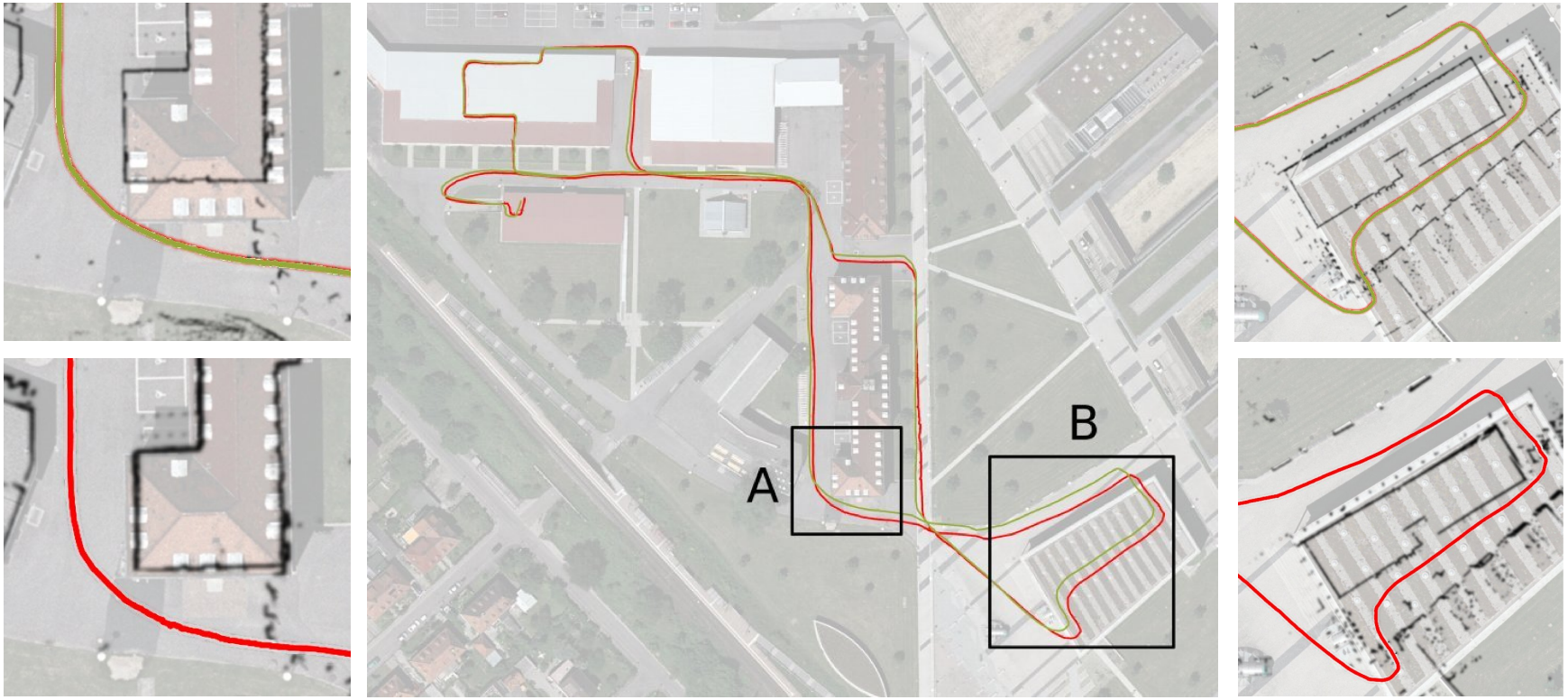
- Integrate the prior into graph-SLAM

$$(\mathbf{x}_1, \ldots, \mathbf{x}_n)^* = \underset{(\mathbf{x}_1,\ldots,\mathbf{x}_n)}{\text{argmin}} \sum_{\langle j,i \rangle} e_{ji}(\mathbf{x}_i, \mathbf{x}_j)^T \Omega_{ji} e_{ji}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i \in \mathcal{G}} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^T R_i (\mathbf{x}_i - \hat{\mathbf{x}}_i)$$




- Optimization carried out based on non-linear conjugate gradients

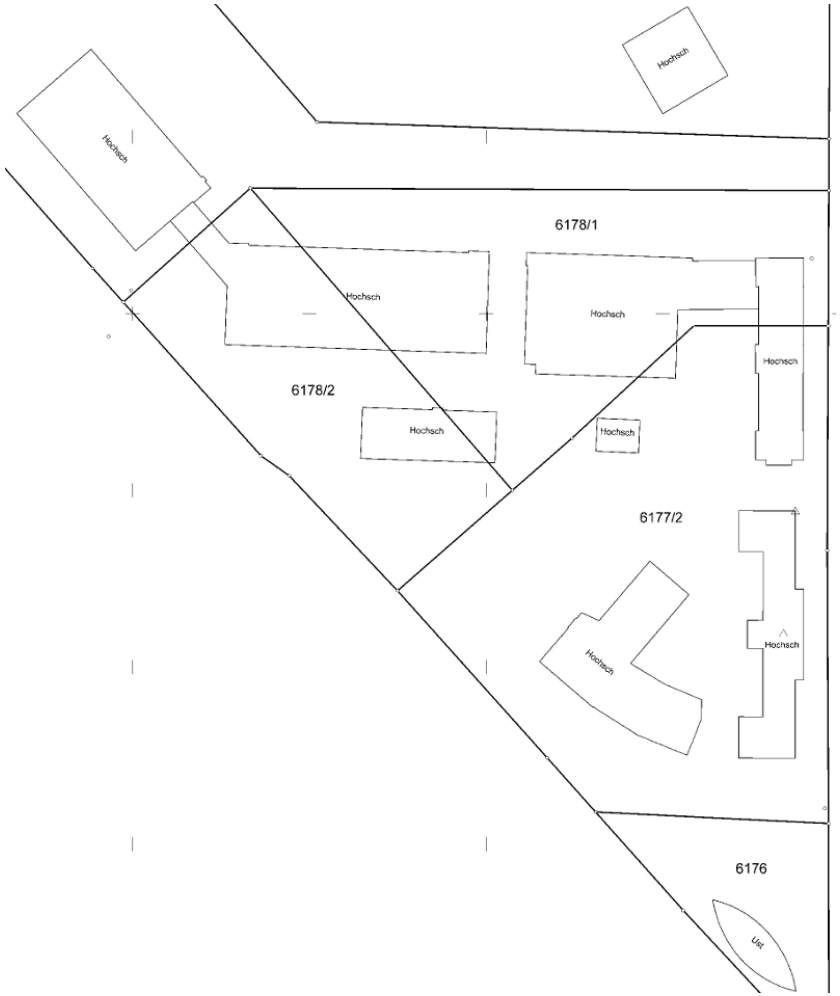
# Global Map Accuracy




**By exploiting the prior information, we achieve a better global consistency.**

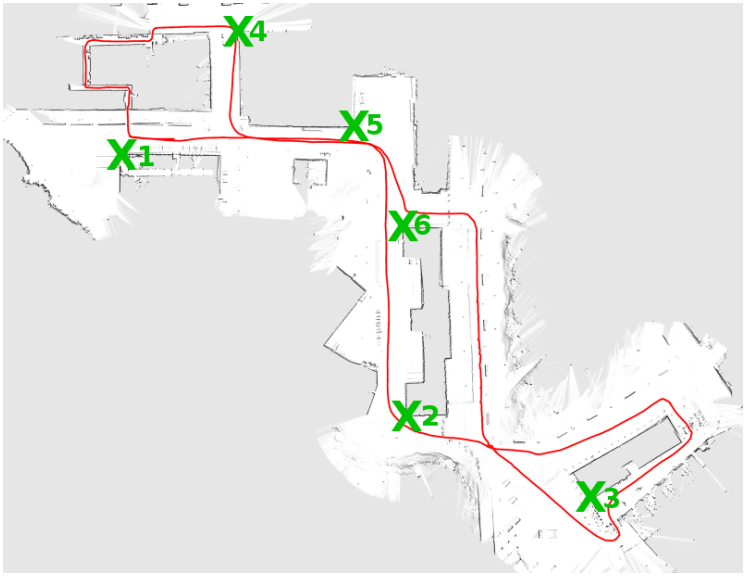
# Ground Truth Data

- Survey data provided by the German Land Registry office
- Outer walls of the buildings in a Gauss-Krüger reference frame

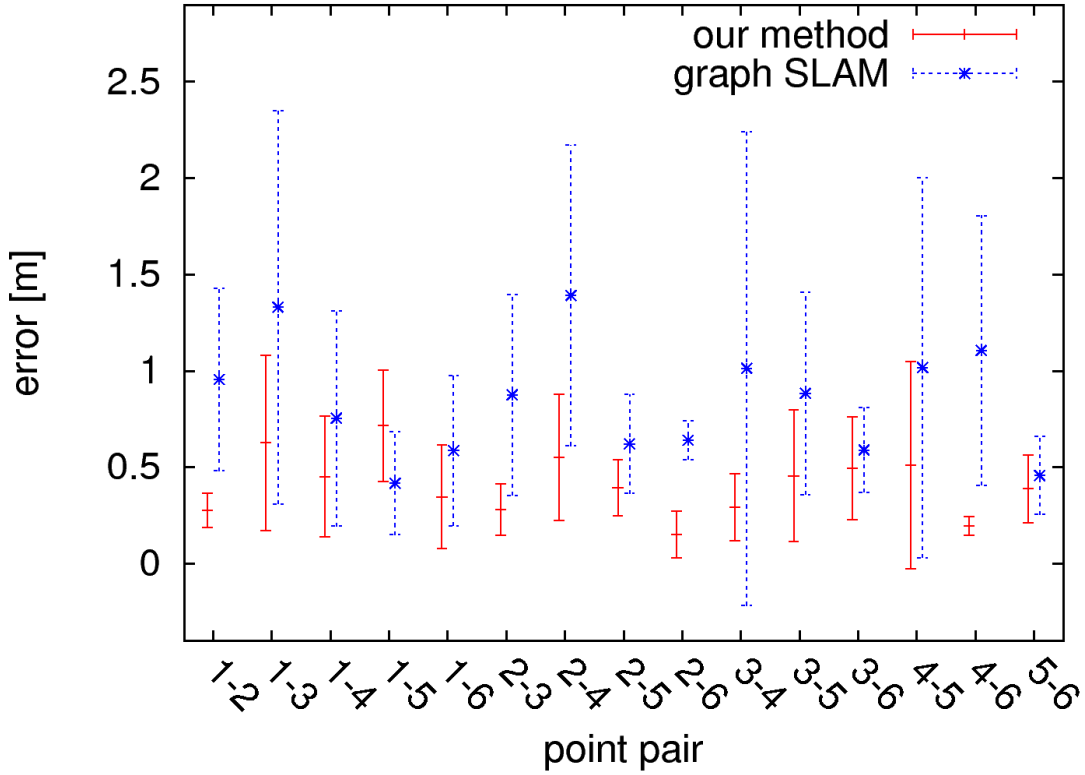


# Statistical Results (1)

- Five data sets.
- We evaluate the pair-wise distance between known points on the map (corners of buildings).



# Statistical Results (2)




- Our method achieves a higher accuracy than graph-SLAM without prior information
- and is more stable.

# Problems

- How to measure the usefulness of a map
- An accurate map might lead to a less efficient behavior than a less accurate map.
- How to benchmark a complete system?

# My Suggestion

- Treat the application as a reinforcement learning problem
- The robot should have the ability to choose between
  - Task execution
  - Localization
  - SLAM actions
- Choosing the optimal trade-off between exploration and exploitation is the key challenge?

# Where will we Arrive at?

- Robotic systems that reliably perform their applications
- We measure them by their performance:
  - What do they cost (also during runtime)?
  - How many tasks did they solve and what was the benefit of this?

# Thanks!