A Comparison of SLAM Algorithms Based on a Graph of Relations

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

RAWSEEDS

- Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets
  - University of Freiburg
  - University of Zaragoza
  - Politecnico di Milano
  - Università degli Studi di Milano-Bicocca

- Funded by the European Commission

- http://www.rawseeds.org/

Motivation

- No gold standard to evaluate and compare the results of SLAM algorithms
- Most approaches depend on the used sensor setup, landmark detector/grid map, and estimation algorithm
- Often, ground truth is not available
- Even if there is ground truth information: how can we compare the results of SLAM algorithms?

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, …)

Measures

- Uncertainty
  (Entropy, KL-Divergence, …)

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

- Quality of estimates
  (observation likelihood, normalized estimation error squared, the measure described in this paper, …)

Sum of Squared Distances between Estimated and Ground Truth Poses

- Let $x_{1:T}$ be the estimated poses from time 1 to T.
- Let $x^*_{1:T}$ be the reference poses.
- Let $\oplus$ be the standard motion composition operator and $\ominus$ its inverse.
- Resulting error function

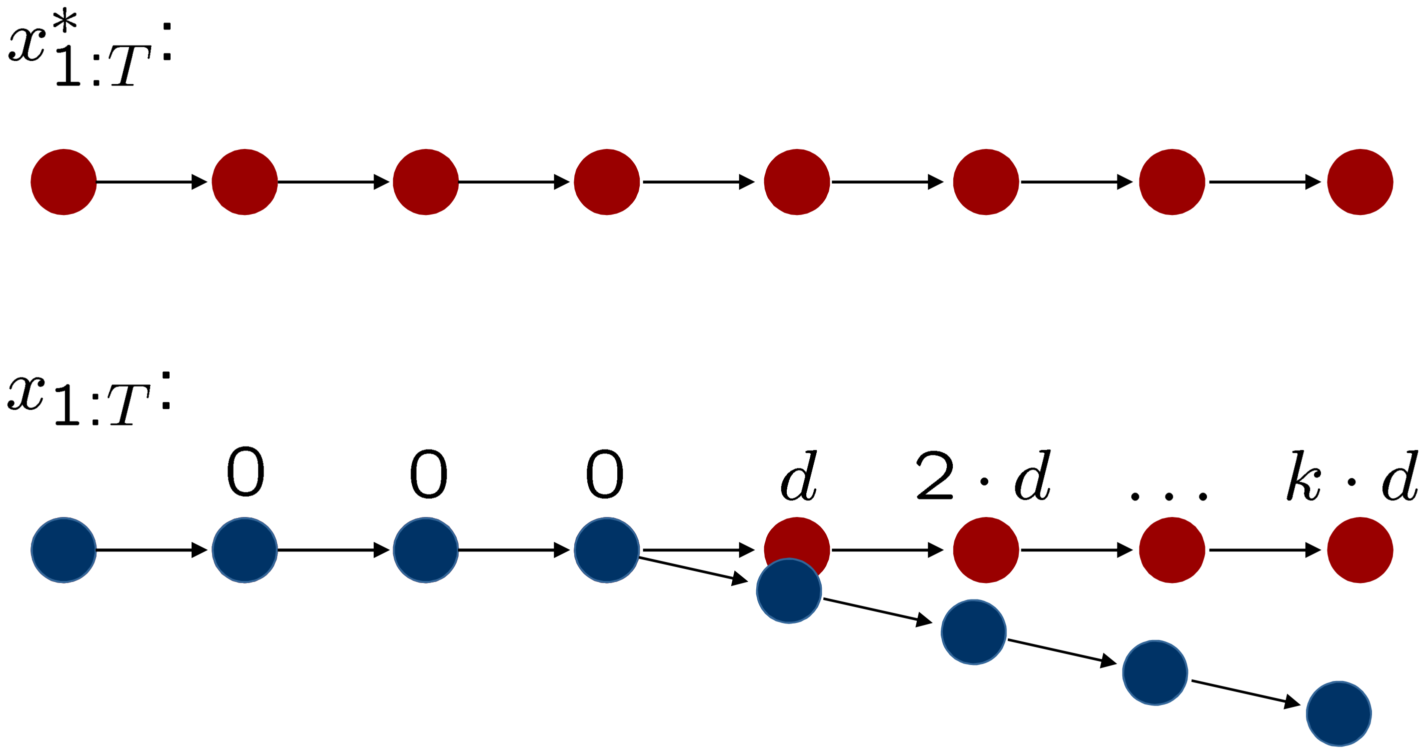
$$\varepsilon(x_{1:T}) = \sum_{t=1}^{T} ||x_t \ominus x_t^*||^2$$

Normalized Estimation Error Squared: Taking into Account Covariance

$$\varepsilon(k) = [x(k) - x^*(k)]^T P^{-1}(k \mid k)[x(k) - x^*(k)]$$

In SLAM, this measures the absolute error of the landmark estimates (weighted by inverse covariance).

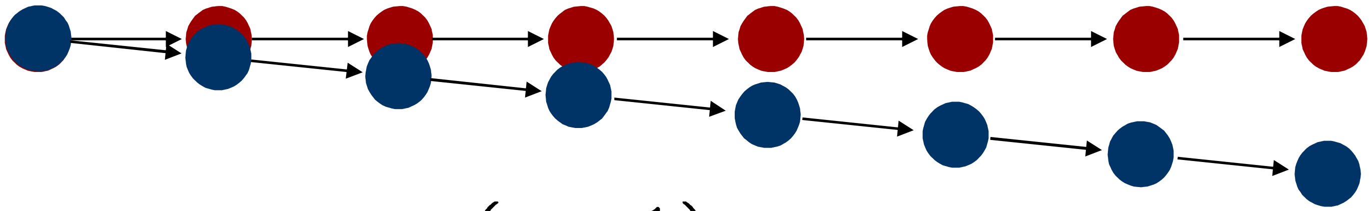
Problems with this Measure



$$\varepsilon(x_{1:T}) = \frac{k \cdot (k+1)}{2} d$$

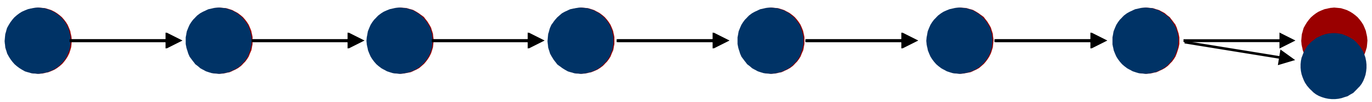
Problems with this Measure

- Error in the beginning:



$$\varepsilon(x_{1:T}) \;=\; \frac{(n-1)\cdot n}{2} d$$

- Error in the very end:



$$\varepsilon(x_{1:T}) \;=\; d$$

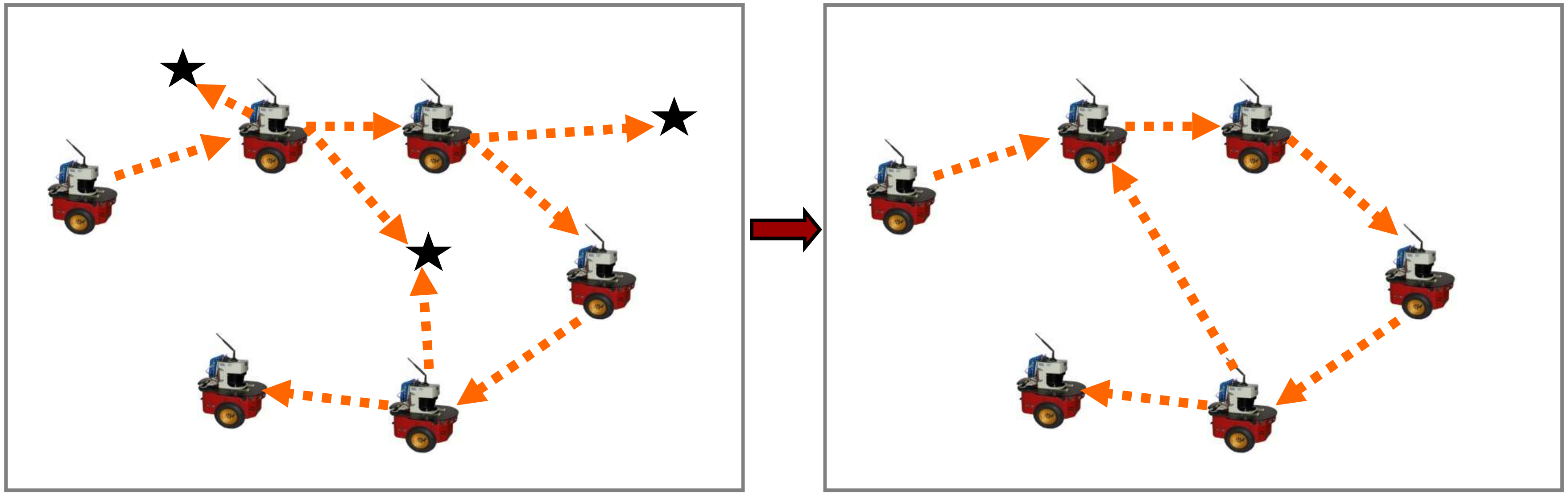
Desired Measure

- that does not automatically count errors multiple times,
- whose results does not depend on the order data is processed, and
- that is compatible with what is actually optimized in SLAM approaches.

Inspired by graph-based SLAM: consider the energy needed to transform the pose graph into ground truth

Pose Graph

- A pose-graph encodes the poses of the robot during mapping as well as constraints resulting from observations
- 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 (consecutive) poses

$$\delta_{i,j} \ominus \delta^*_{i,j}$$

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

Metric Based on Relative Poses

estimated relative movement

ground truth

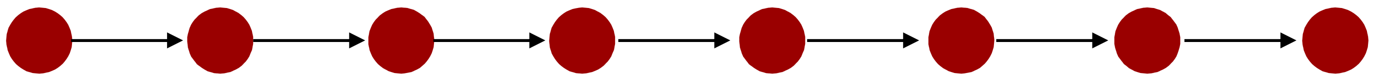
$$\varepsilon(\delta) = \frac{1}{N}\sum_{i,j} ||\delta_{i,j} \ominus \delta^*_{i,j}||^2$$

$$= \frac{1}{N}\sum_{i,j} trans(\delta_{i,j} \ominus \delta^*_{i,j})^2$$

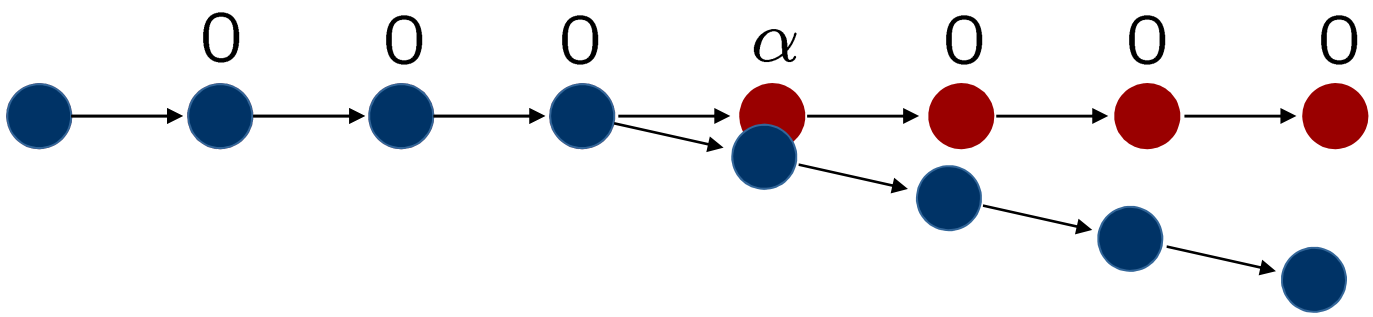
$$+ rot(\delta_{i,j} \ominus \delta^*_{i,j})^2$$

Application to the Example

$x^*_{1:T}$:



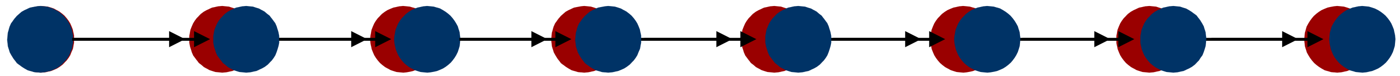
$x_{1:T}$:



$$\varepsilon(x_{1:T}) = \alpha$$

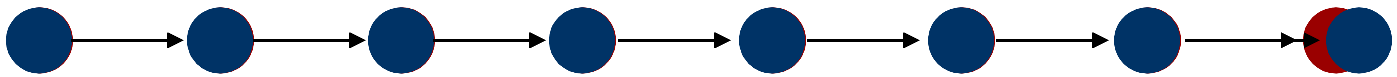
Translational Error

- Error in the beginning:



$$\varepsilon(x_{1:T}) = d \quad \text{vs.} \quad \varepsilon(x_{1:T}) = (n-1) \cdot d$$

- Error in the very end:

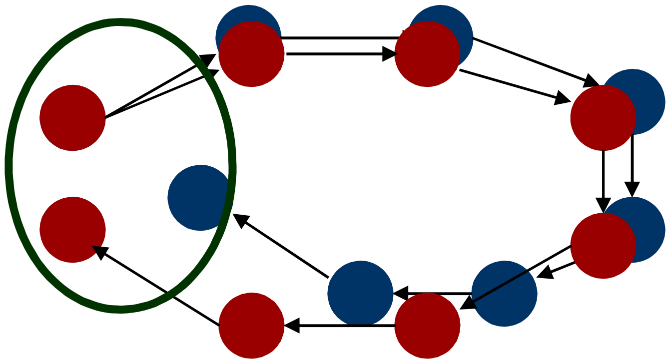


$$\varepsilon(x_{1:T}) = d$$

Properties

- Good for incremental errors
- Thus far, we only considered incremental links
- This can be sufficient for navigation,
- but does not help to close loops and achieve global consistency.

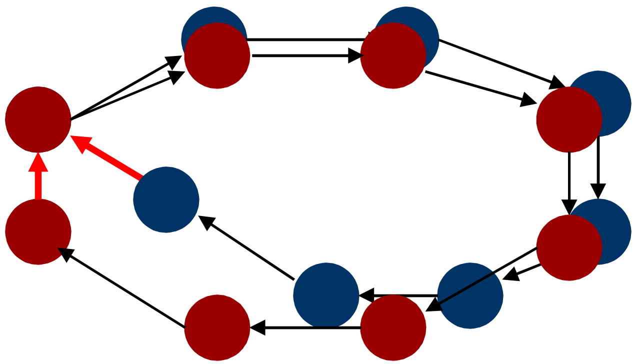
Addition of Loop Closure Links



In the loop closure area, the map will be locally inconsistent, which this is not reflected by

$$\varepsilon(x_{1:T}) = \sum_{t=1}^{T} \varepsilon_t$$

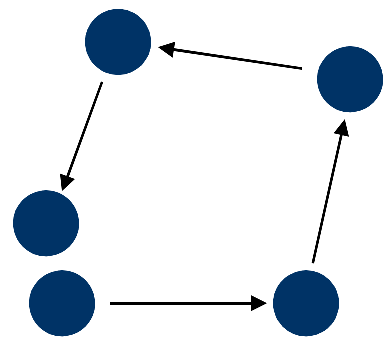
If we introduce an appropriate link, we increase the error, representing the faulty part of the map



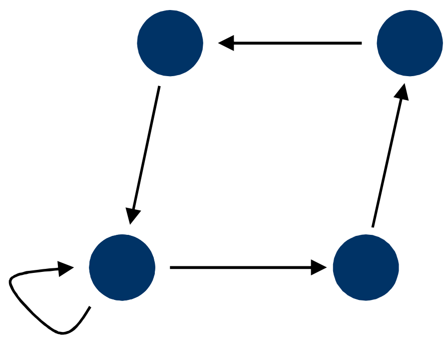
$$\varepsilon(x_{1:T}) = \varepsilon_{\text{loop}} + \sum_{t=1}^{T} \varepsilon_t$$

Selection of Relations

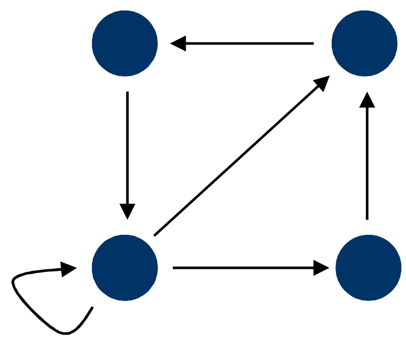
- No loop closure links:
  local consistency only
  (wrt. time)



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

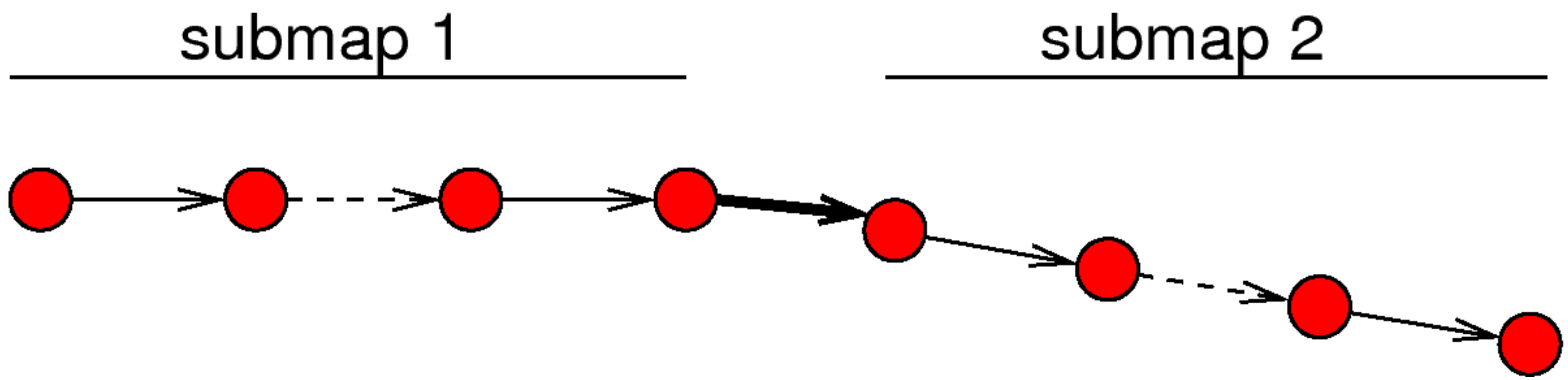


- Links between far
  away nodes:
  global consistency



Negative example for naive metric

- Minor angular error somewhere in the trajectory leads to a high error in the naive metric

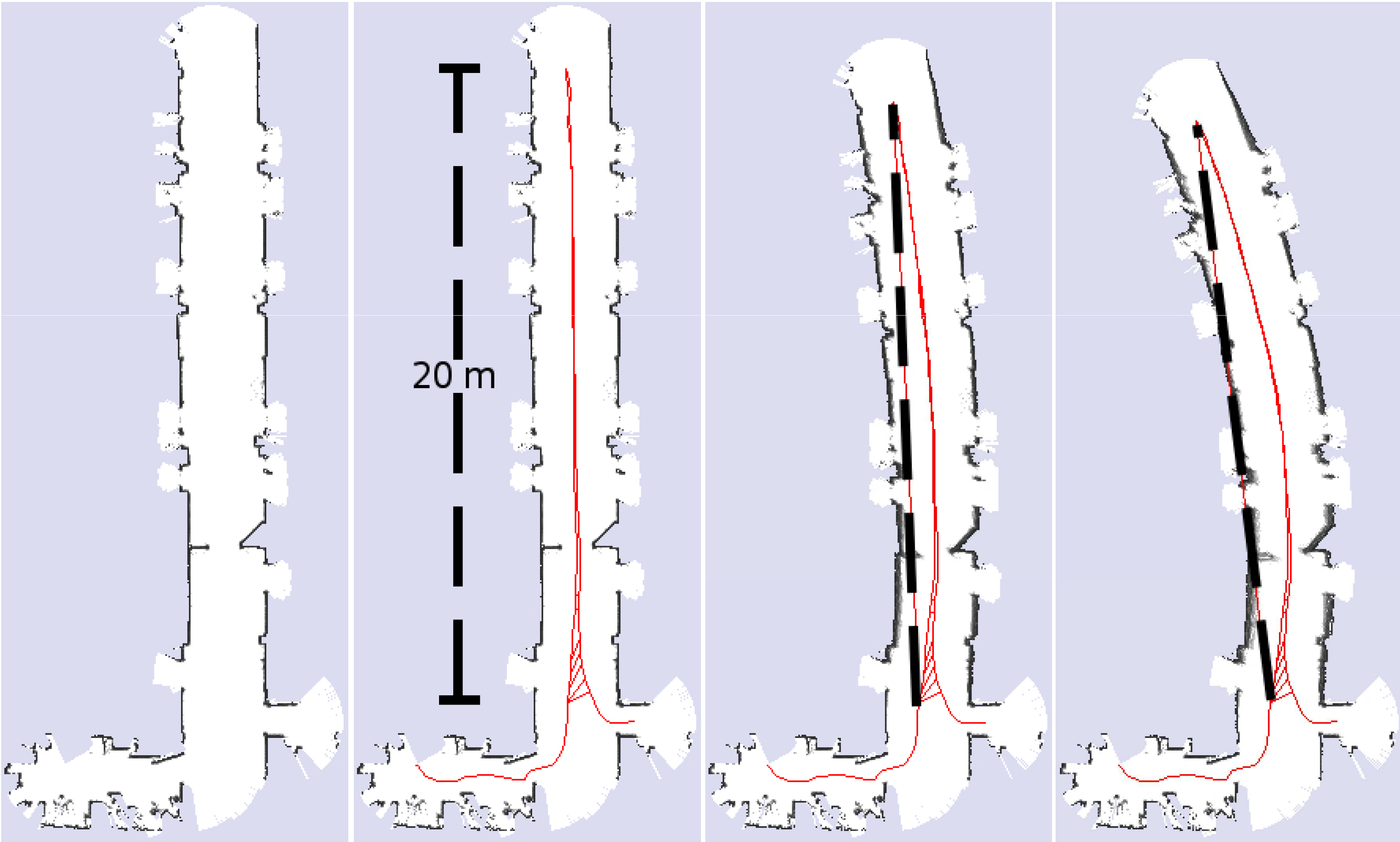


Our Approach

- Consider the energy needed to transform the result of a SLAM algorithm into the ground truth
- Inspiration from the ideas of graph-based SLAM

[ADD EXAMPLE FROM THE PAPER]

Benefit of (Manually Added) Relations for Global Consistency



Relative Relations from a Pose Graph

- Generate the true relative relations given background knowledge
- Compute the error of these relations given the robot’s poses estimated by the SLAM approach

[ADD FORMULAR FROM THE PAPER]

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

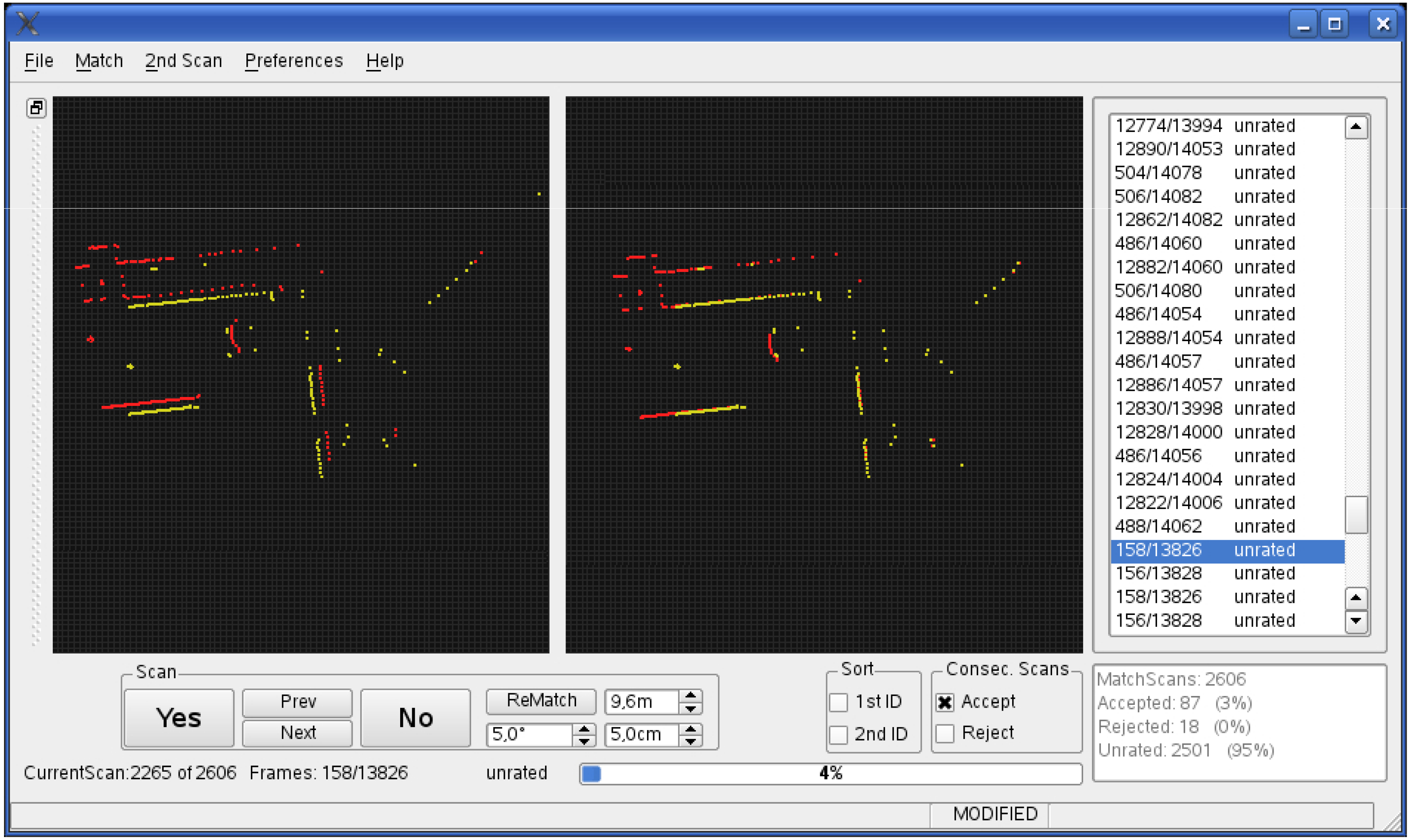
Determine Relations Manually

- Scan alignment paired with manual inspection can generate close to ground truth relations
- Significant manual efforts but need to be done only once per dataset

[ADD GUI IMAGE FROM THE PAPER]

Graphical Interface to Confirm & Correct Constraints

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



Experiments

- Application of measure to evaluate measure for different algorithms and environments
- Demonstrate that the scores correlate with the quality of the map
- Demonstrate that the plots allows us to localize errors in the alignments

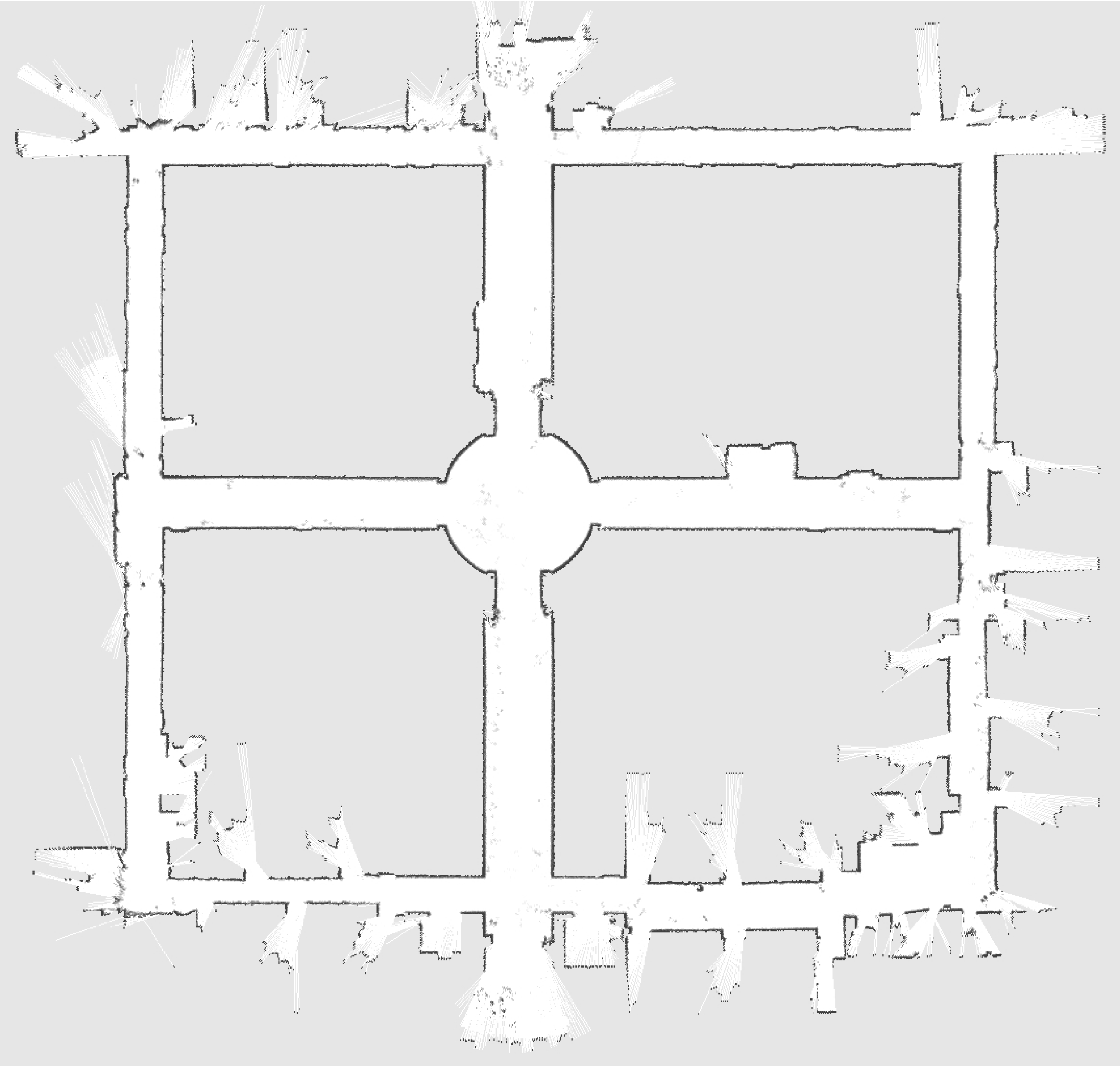
Relations Available Online

- We generated relations in this way for
  - MIT Killian Court
  - ACES Building at the University of Texas
  - Intel Research Lab Seattle
  - MIT CSAIL Building
  - Building 079 University of Freiburg

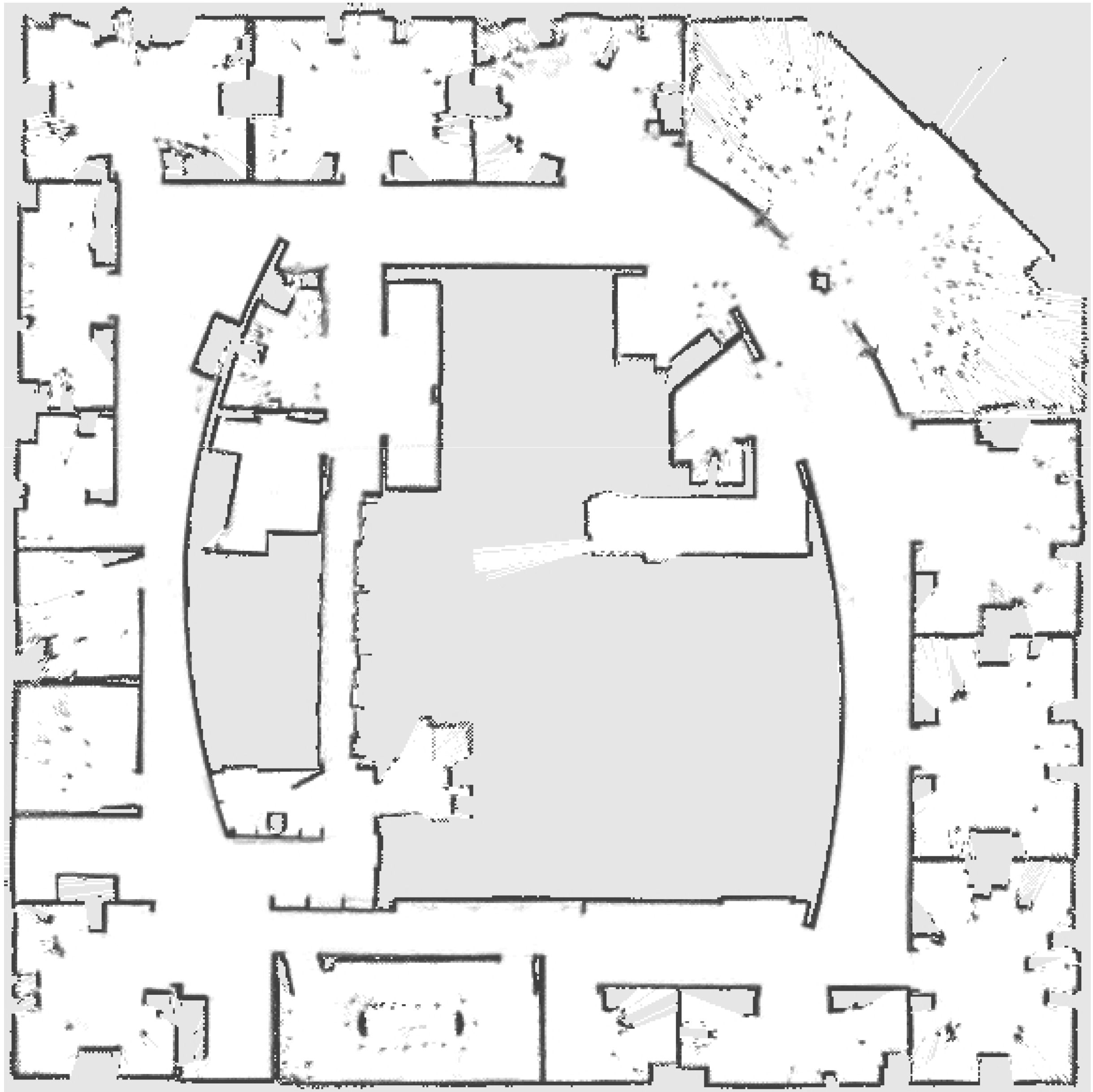
These are online available at

http://ais.informatik.uni-freiburg.de/slamevaluation/

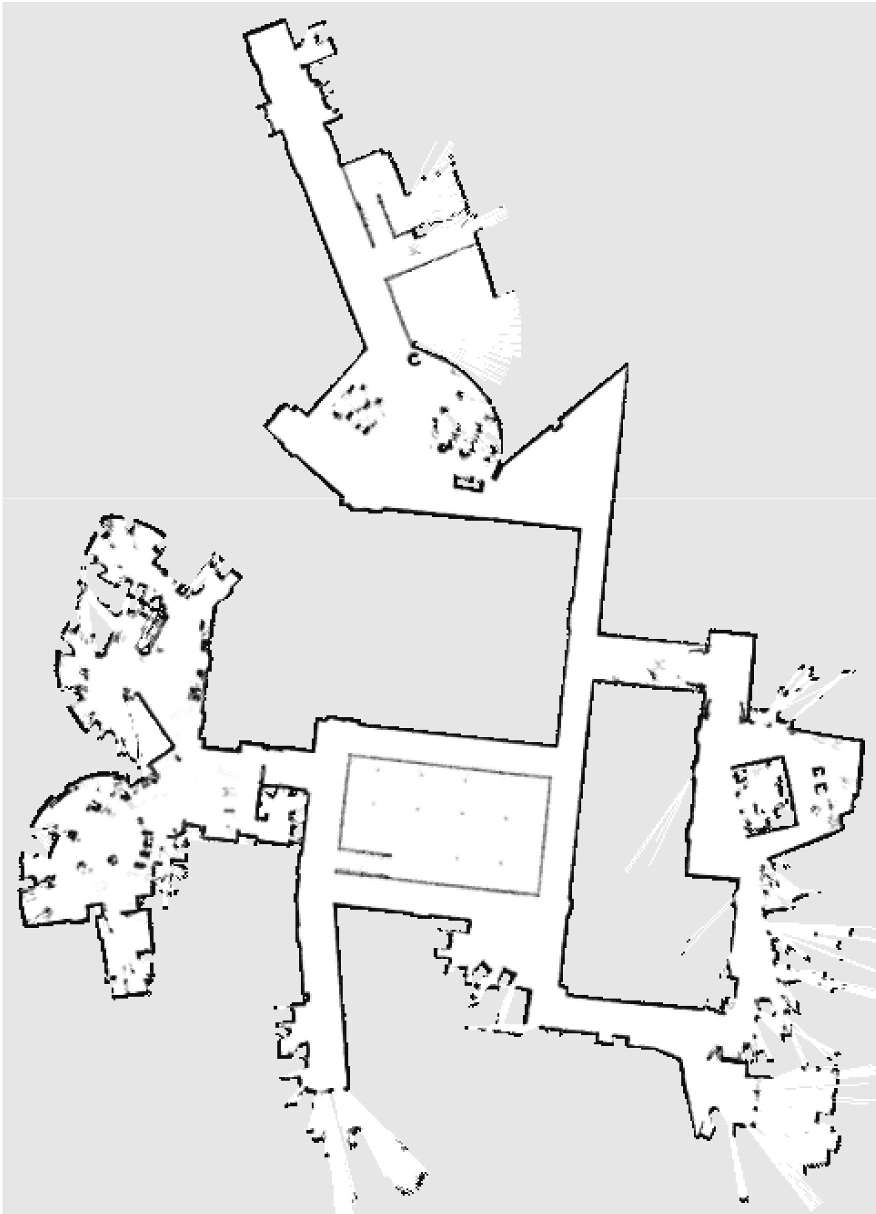
ACES Building



Intel Lab, Seattle



MIT CSAIL

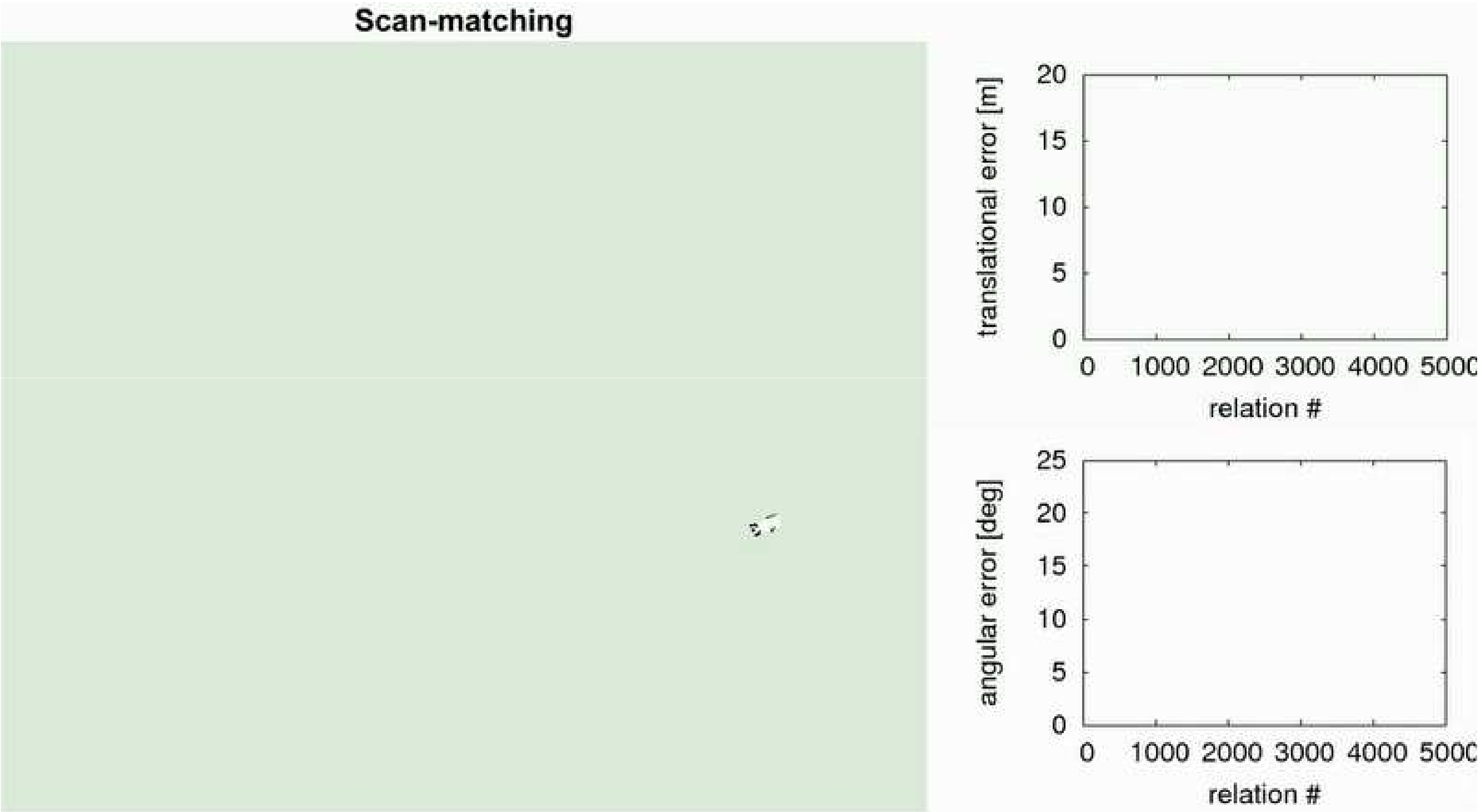


Building 79 Univ. of Freiburg

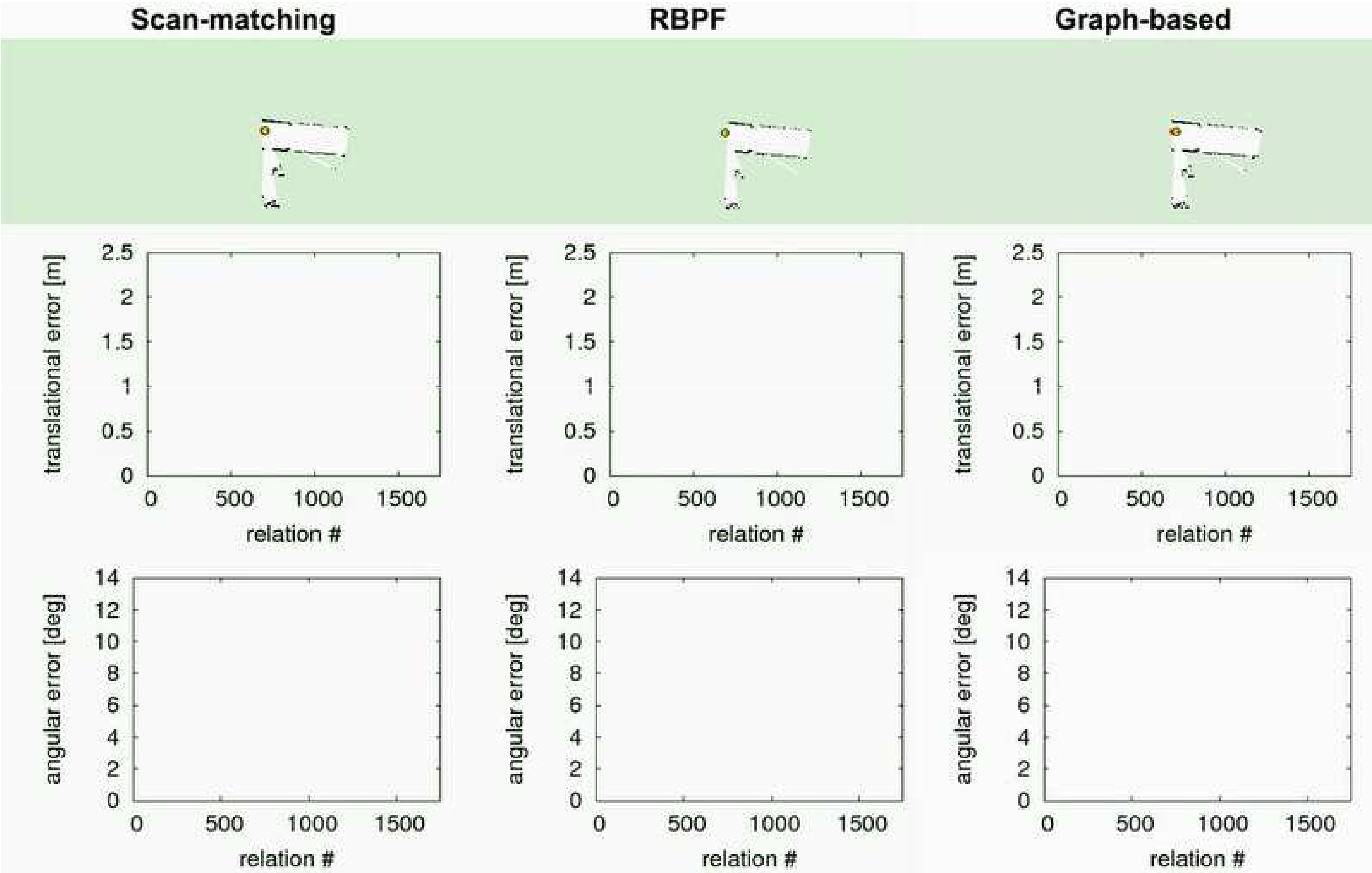
Experimental Evaluation

- Compared three approach using all datasets
  - Incremental scan-matching
  - GMapping (see http://www.openslam.org)
  - Graph-based SLAM system based on TORO (see http://www.openslam.org)
- We provide resulting scores for all algorithms and datasets

Example

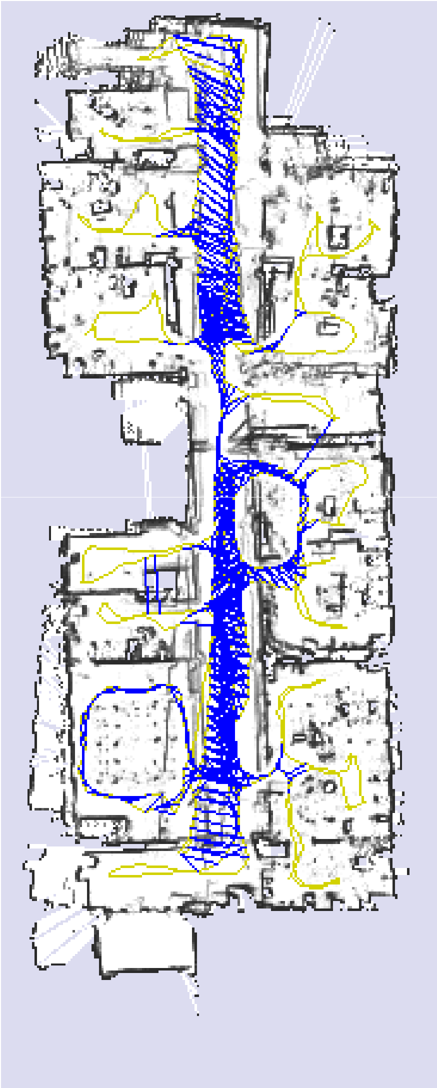

Scan-matching
translational error [m]
20
15
10
5
0
0
1000
2000
3000
4000
relation #
angular error [deg]
25
20
15
10
5
0
0
1000
2000
3000
4000
relation #

Example



Scan-matching
RBPF
Graph-based
translational error [m]
2.5
2
1.5
1
0.5
0
0
500
1000
1500
relation #
angular error [deg]
14
12
10
8
6
4
2
0
0
500
1000
1500
relation #

Maps Obtained

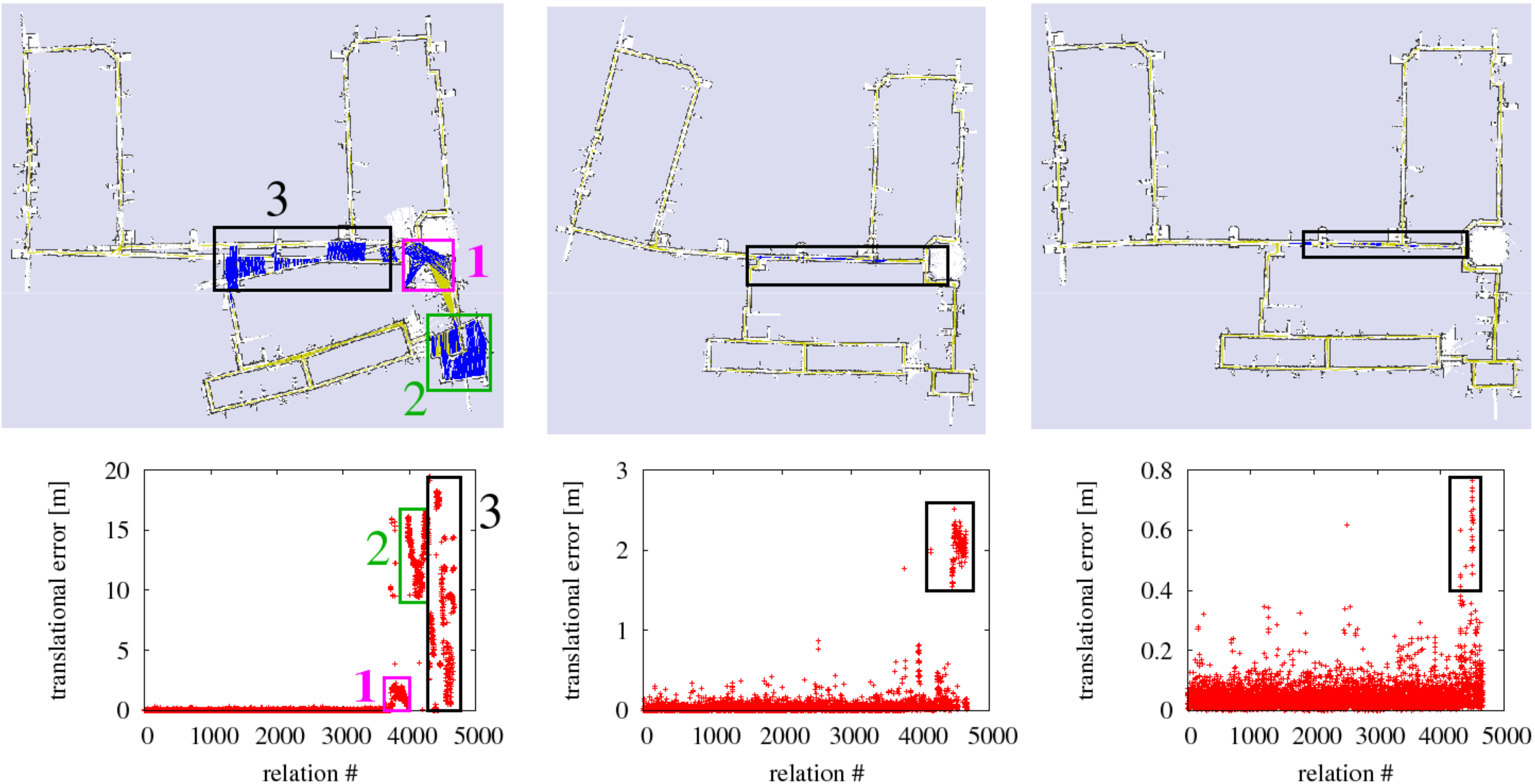


Scan Matching

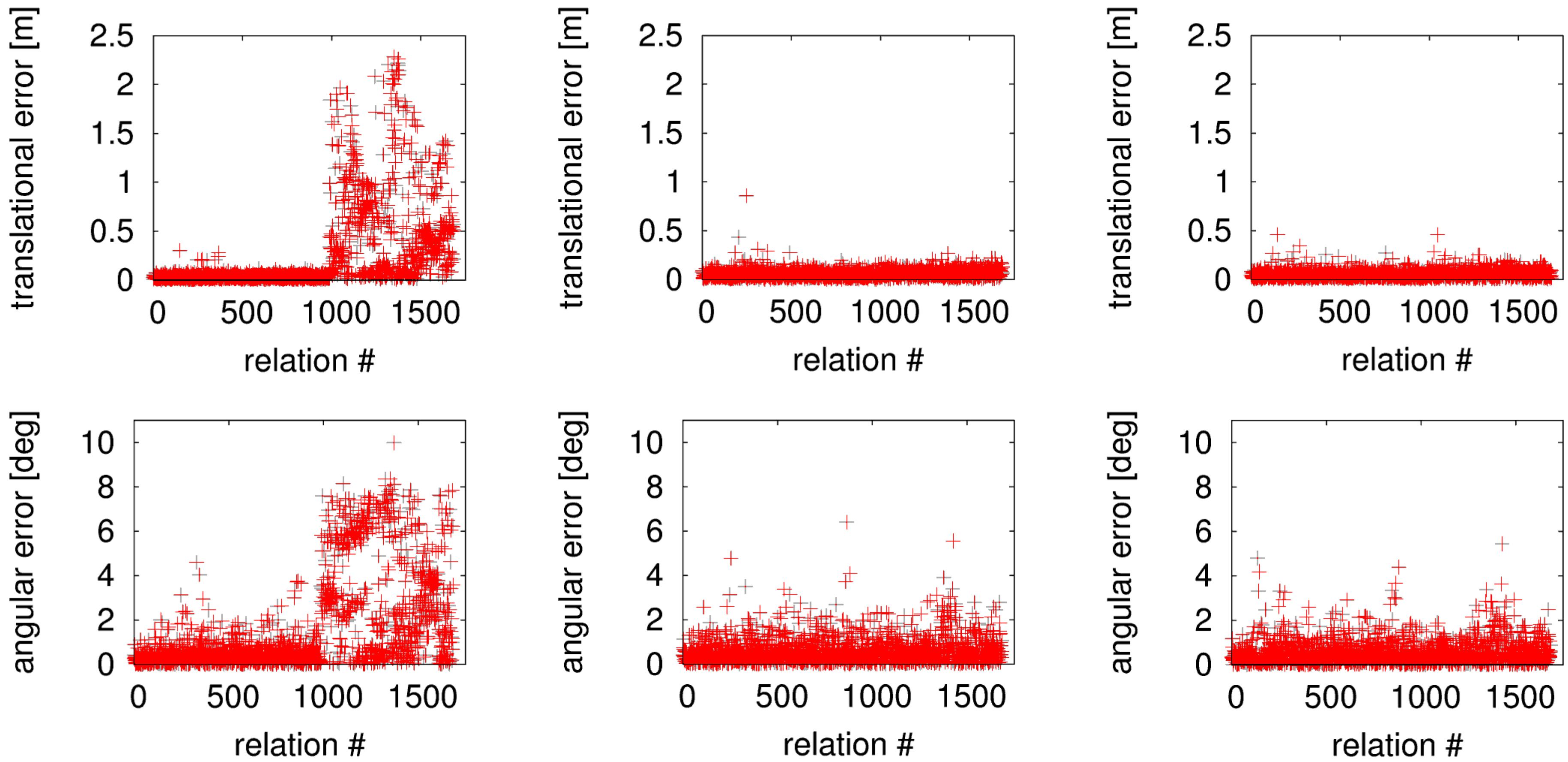
RBPF

Graph mapping

Analyzing Error Plots



Error Plots



Scan Matching | RBPF | Graph mapping

Numerical Evaluation

Trans. error m / m^2	Scan Matching	RBPF (50 part.)	Graph Mapping
Aces (abs)	0.173 ± 0.614	0.060 ± 0.049	0.044 ± 0.044
Aces (sqr)	0.407 ± 2.726	0.006 ± 0.011	0.004 ± 0.009
Intel (abs)	0.220 ± 0.296	0.070 ± 0.083	0.031 ± 0.026
Intel (sqr)	0.136 ± 0.277	0.011 ± 0.034	0.002 ± 0.004
MIT (abs)	1.651 ± 4.138	0.122 ± 0.386[1]	0.050 ± 0.056
MIT (sqr)	19.85 ± 59.84	0.164 ± 0.814[1]	0.006 ± 0.029
CSAIL (abs)	0.106 ± 0.325	0.049 ± 0.049[1]	0.004 ± 0.009
CSAIL (sqr)	0.117 ± 0.728	0.005 ± 0.013[1]	0.0001 ± 0.0005
FR 79 (abs)	0.258 ± 0.427	0.061 ± 0.044[1]	0.056 ± 0.042
FR 79 (sqr)	0.249 ± 0.687	0.006 ± 0.020[1]	0.005 ± 0.011

Numerical Evaluation

Rot. error deg / deg^2	Scan Matching	RBPF (50 part.)	Graph Mapping
Aces (abs)	1.2 ± 1.5	1.2 ± 1.3	0.4 ± 0.4
Aces (swr)	3.7 ± 10.7	3.1 ± 7.0	0.3 ± 0.8
Intel (abs)	1.7 ± 4.8	3.0 ± 5.3	1.3 ± 4.7
Intel (sqr)	25.8 ± 170.9	36.7 ± 187.7	24.0 ± 166.1
MIT (abs)	2.3 ± 4.5	0.8 ± 0.8[1]	0.5 ± 0.5
MIT (sqr)	25.4 ± 65.0	0.9 ± 1.7[1]	0.9 ± 0.9
CSAIL (abs)	1.4 ± 4.5	0.6 ± 1.2[1]	0.05 ± 0.08
CSAIL (sqr)	22.3 ± 111.3	1.9 ± 17.3[1]	0.01 ± 0.04
FR 79 (abs)	1.7 ± 2.1	0.6 ± 0.6[1]	0.6 ± 0.6
FR 79 (sqr)	7.3 ± 14.5	0.7 ± 2.0[1]	0.7 ± 1.7

What if no Trajectory is Available?

- Algorithms such as EKF-based SLAM often do not store the whole trajectory of the robot
- No relations can be evaluated
- Two solutions:
  - Recover the trajectory by localization
  - Transfer the idea of relations to landmarks (which yields a landmark-pose-graph)

Offline Trajectory Recovery by Localization

- EKF-based approaches provide landmark estimates
- Data associations are made during SLAM
- To recover the poses of the robot during mapping, run EKF-based localization given the landmark locations, data associations, and motion/sensor model used during SLAM
- Approach using relation can be applied directly

Relations based on Landmark Locations

- Alternatively, one can retrieve from known landmark locations relative relations via a triangle mesh
- Take care: data associations between landmarks might need to be done manually

Towards the Rescue of the Sum of Squared Distances

- Let $x_{1:T}$ be the estimated poses from time 1 to T.
- Let $x^*_{1:T}$ be the reference poses.
- Error function

$$\varepsilon(x_{1:T}) \;=\; \min_{A} \sum_{t=1}^{T} ||x_t \ominus A x_t^*||^2$$

where A is the set of affine transformations

Conclusions

- Approach to comparing SLAM algorithms
- Based on relative relations between poses
- Independent of the used algorithm
- Independent of the sensor setup since it operates on the trajectory estimate
- Manually corrected datasets available online
- Results for comparisons available online

Limitations

- The measure mostly captures localization
- Assumption is that good localization implies accurate map
- It does not directly capture, whether the map contains all obstacles in the real world.