Visual Navigation for Flying Robots

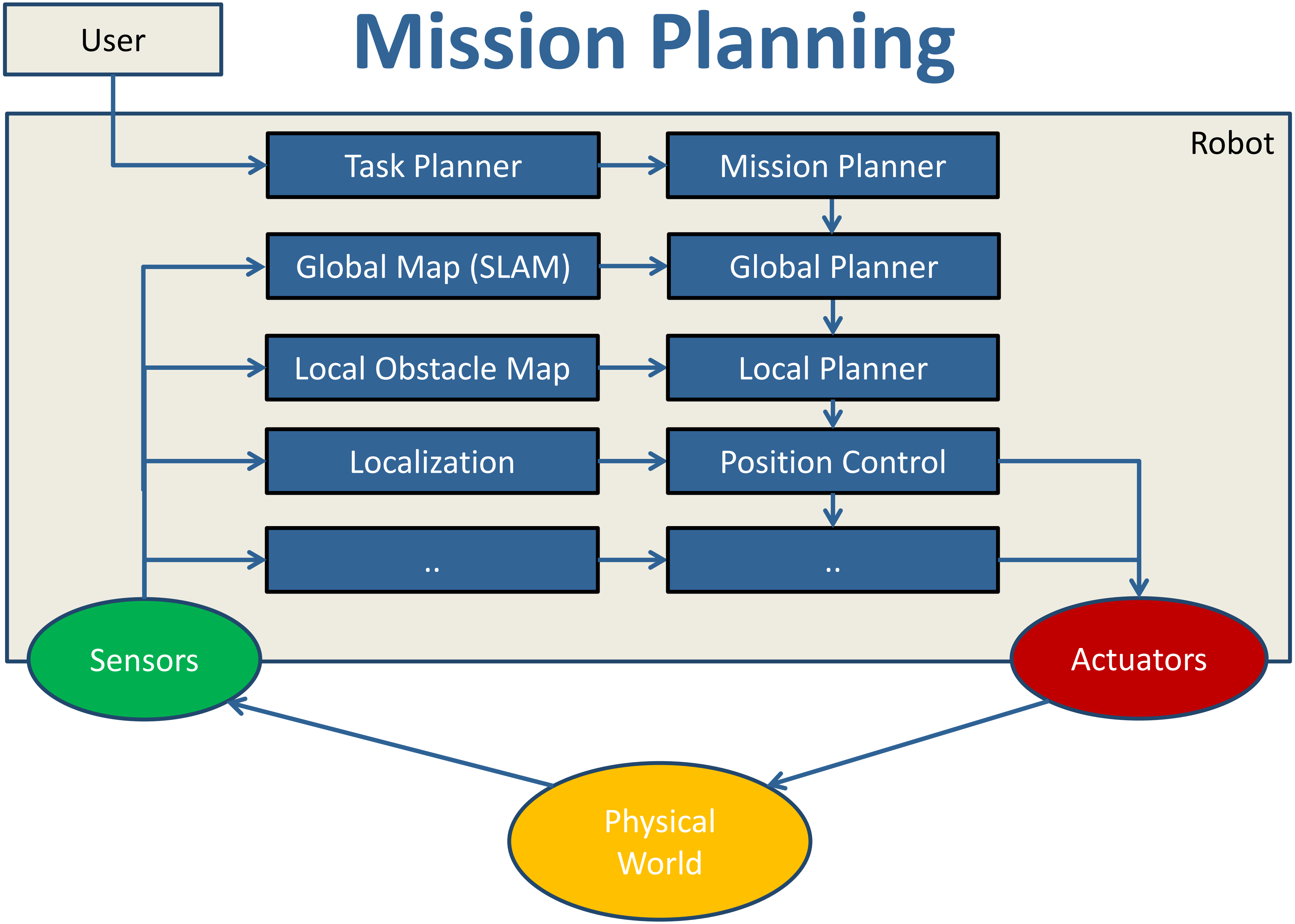
Exploration, Multi-Robot Coordination and Coverage

Dr. Jürgen Sturm

Agenda for Today

- Exploration with a single robot
- Coordinated exploration with a team of robots
- Coverage planning
- Benchmarking

Mission Planning



Mission Planning

- **Goal:** Generate and execute a plan to accomplish a certain (navigation) task
- Example tasks
  - Exploration
  - Coverage
  - Surveillance
  - Tracking
  - …

Task Planning

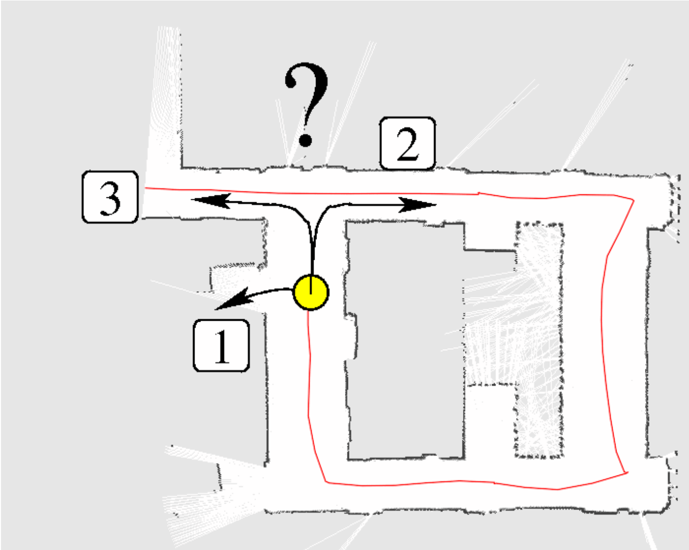
- **Goal:** Generate and execute a high level plan to accomplish a certain task
- Often symbolic reasoning (or hard-coded)
  - Propositional or first-order logic
  - Automated reasoning systems
  - Common programming languages: Prolog, LISP
- Multi-agent systems, communication
- Artificial Intelligence

Exploration and SLAM

- **SLAM is typically passive**, because it consumes incoming sensor data
- **Exploration actively guides the robot** to cover the environment with its sensors
- Exploration in combination with SLAM: Acting under pose and map uncertainty
- Uncertainty should/needs to be taken into account when selecting an action

Exploration

- By reasoning about control, the mapping process can be made much more effective
- Question: **Where to move next?**



- This is also called the **next-best-view problem**

Exploration

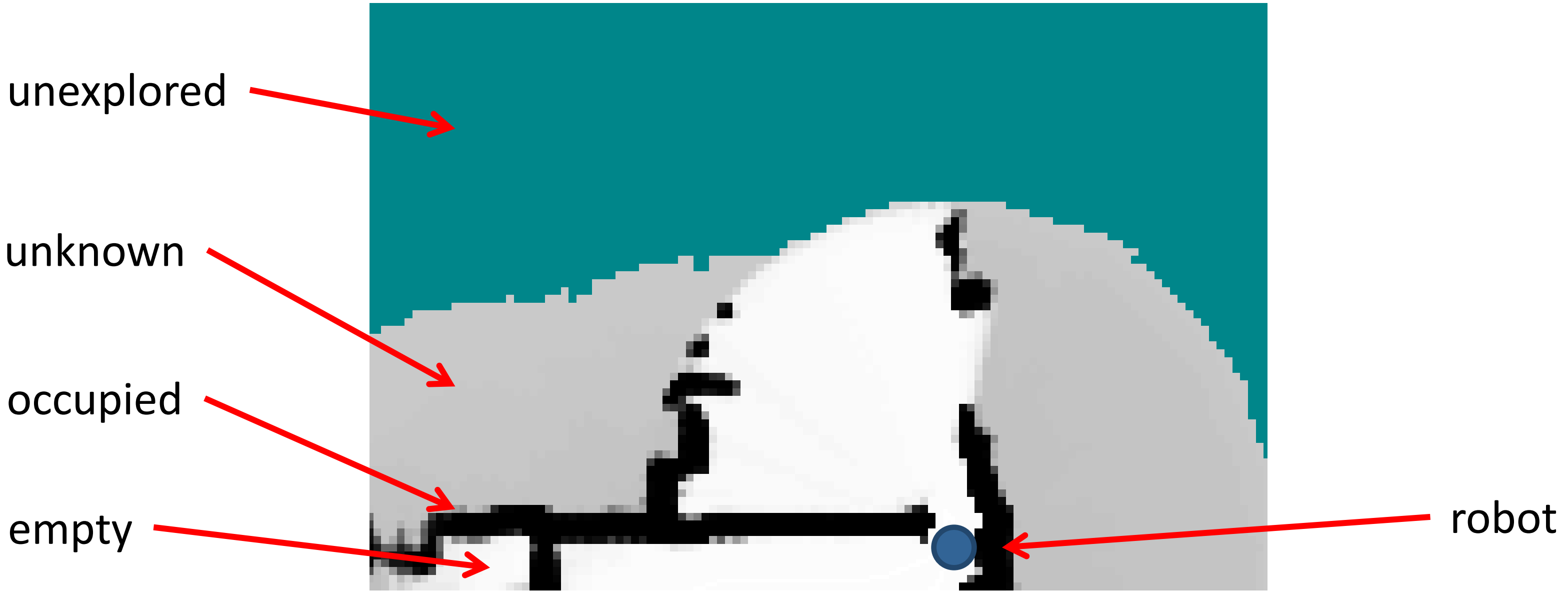
- Choose the action that maximizes utility

$$a^* = \arg\max_{a \in A} U(m, a)$$

- **Question:** How can we define utility?

Example

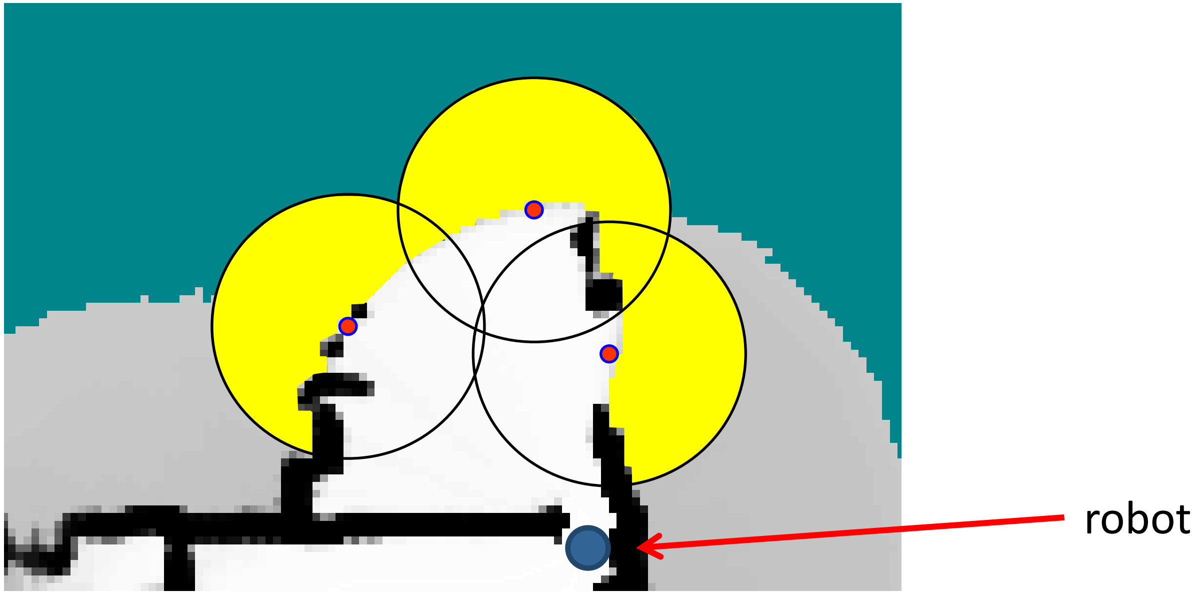
- Where should the robot go next?



Maximizing the Information Gain

- Pick the action a that maximizes the **information gain** given a map m

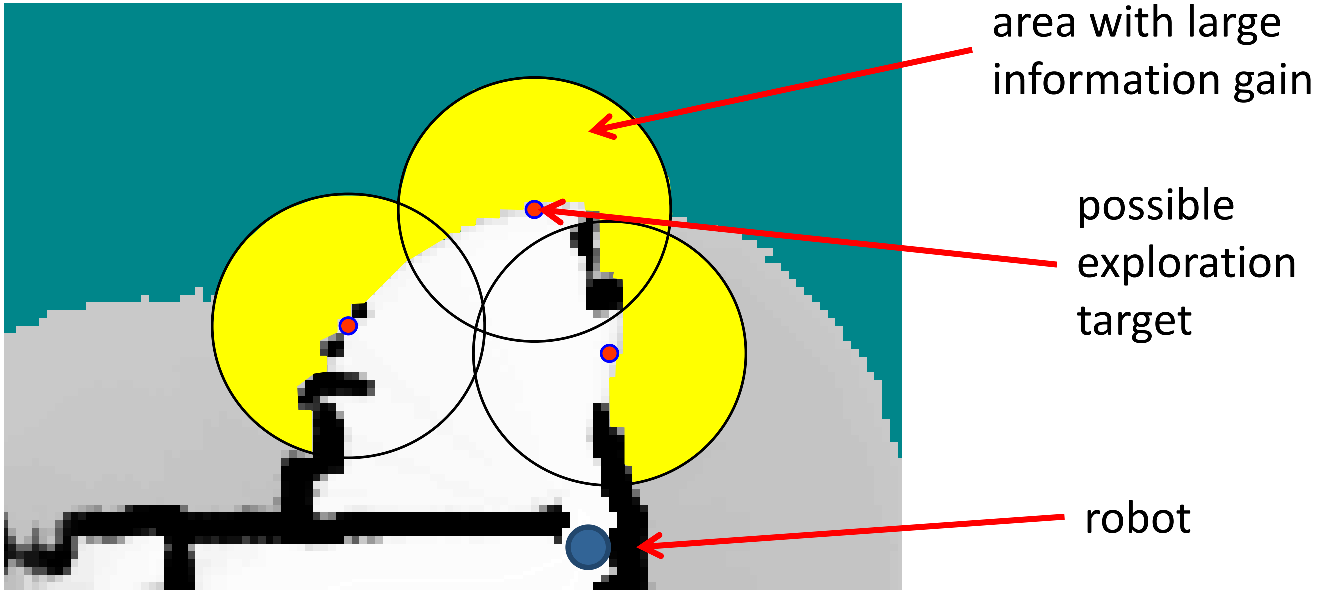
$$a^* = \arg\max_{a \in A} IG(m, a)$$



Maximizing the Information Gain

- Pick the action a that maximizes the **information gain** given a map m

$$a^* = \arg\max_{a \in A} IG(m, a)$$



Information Theory

- **Entropy** is a general measure for the uncertainty of a probability distribution
- Entropy = Expected amount of information needed to encode an outcome $X = x$

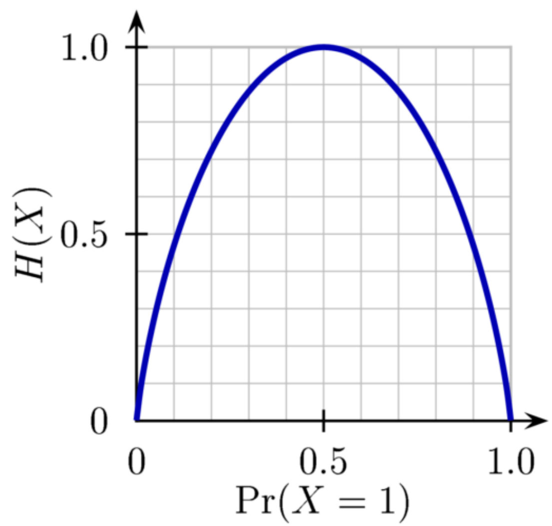
$$
\begin{aligned}
H(X) &= E(I(X)) \\
&= E(-\log p(X)) \\
&= -\sum_{i=1}^{n} p(x_i) \log p(x_i)
\end{aligned}
$$

Example: Binary Random Variable

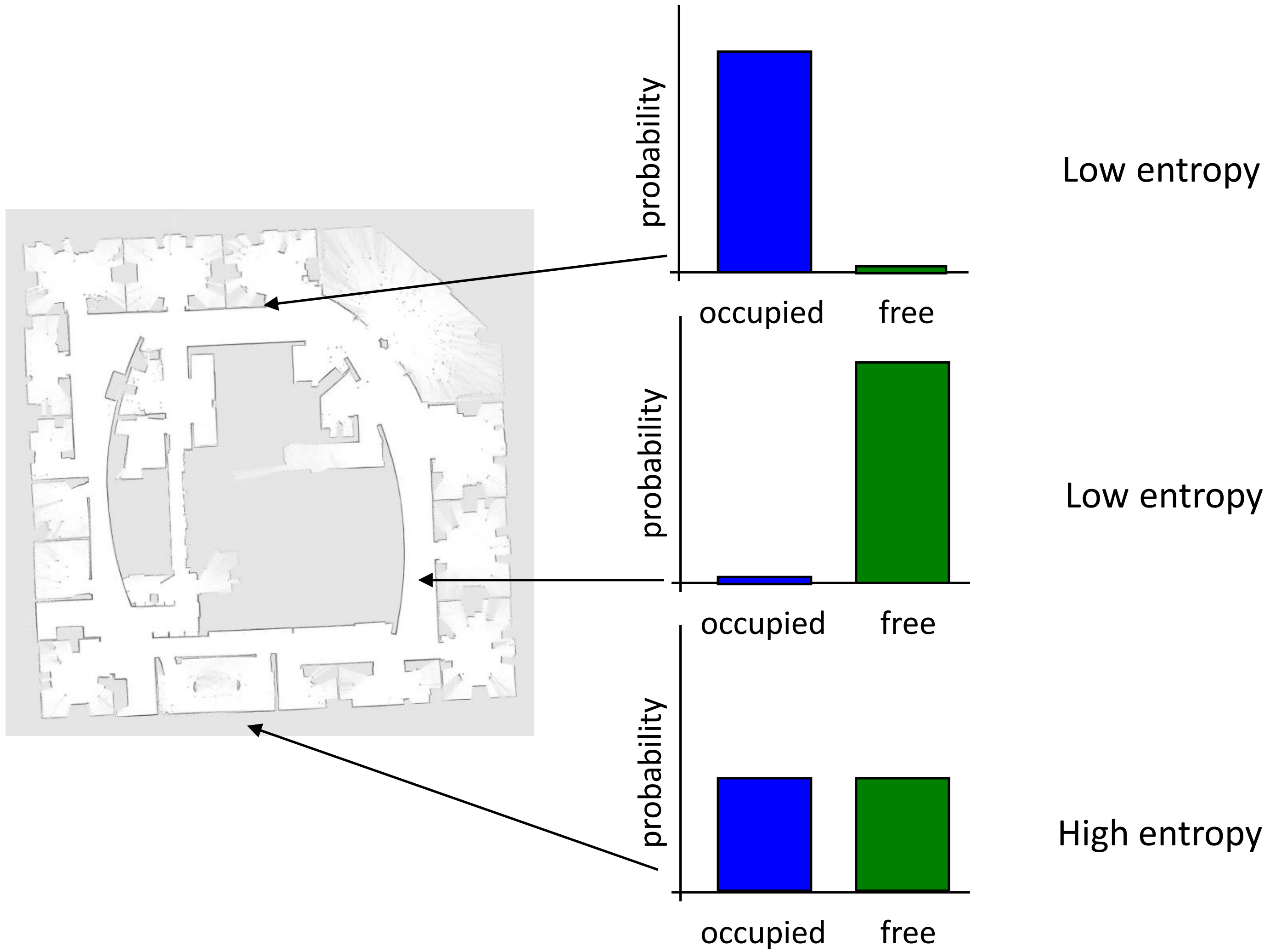
- Binary random variable $X \in \{0, 1\}$
- Probability distribution $P(X = 1) = p$
- How many bits do we need to transmit one sample of $p(X)$?
  - For p=0?
  - For p=0.5?
  - For p=1?

Example: Binary Random Variable

- Binary random variable $X \in \{0, 1\}$
- Probability distribution $P(X = 1) = p$
- How many bits do we need to transmit one sample of $p(X)$?
- Answer:



Example: Map Entropy



The overall entropy is the sum of the individual entropy values

Information Theory

- **Entropy of a grid map**

$$H(p(x_t)) = -\sum_{c \in m} p(c) \log p(c) + (1 - p(c)) \log(1 - p(c))$$

grid cells

probability that the cell is occupied

- **Information gain** = reduction in entropy

$$IG(t+1 \mid t) = H(p(x_t)) - H(p(x_{t+1}))$$

Maximizing the Information Gain

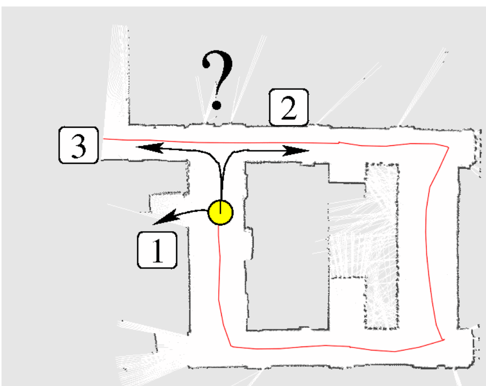
- To compute the information gain one needs to know the observations obtained when carrying out an action

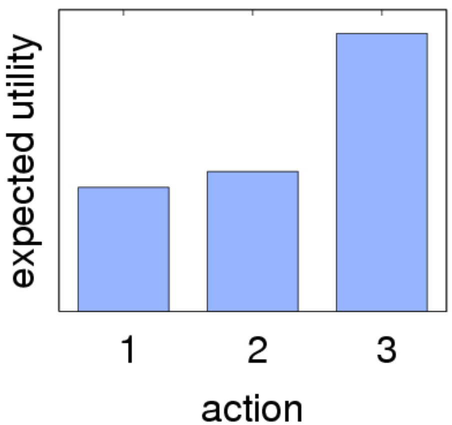
$$a^* = \arg\max_{a \in A} IG(m, a)$$

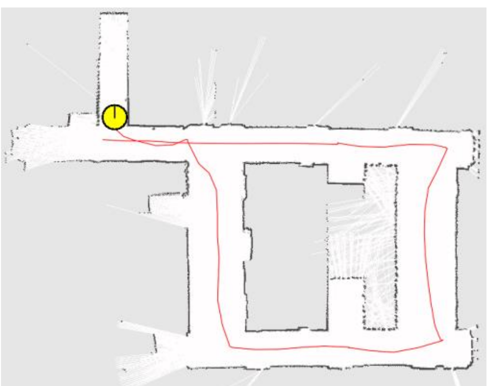
- This quantity is not known! Reason about potential measurements

$$a^* = \arg\max_{a \in A} \int IG(m, z) p(z \mid a) \mathrm{d}z$$

Example







Exploration Costs

- So far, we did not consider the cost of executing an action (e.g., time, energy, …)

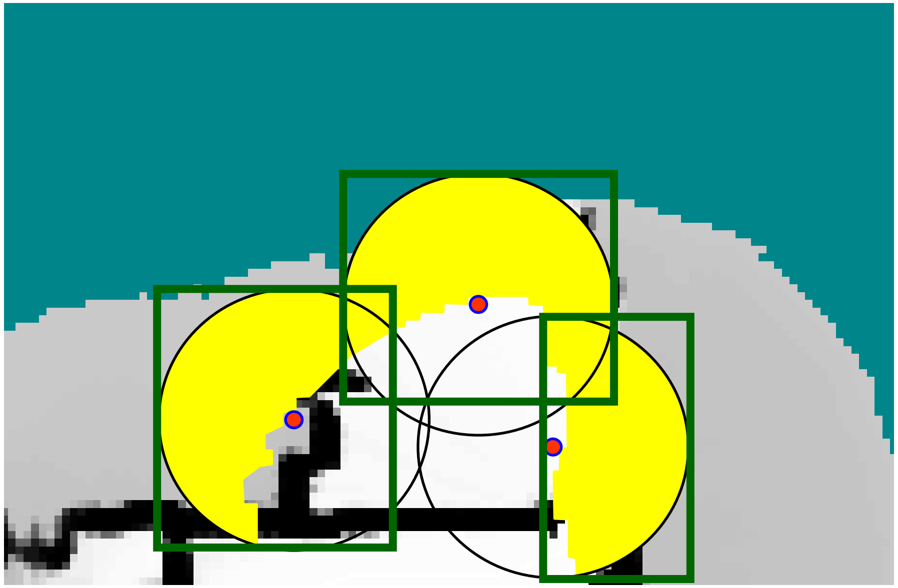
- **Utility = uncertainty reduction – cost**

- Select the action with the highest expected utility

$$a^* = \arg\max_{a \in A} IG(m, a) - \alpha \cdot E(cost(m, a))$$

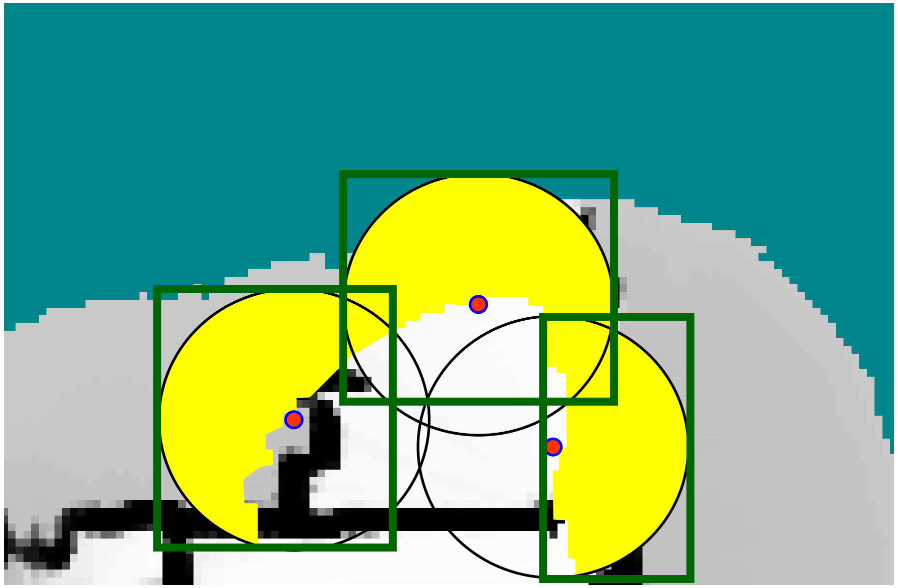
Exploration

- For each location <x,y>
  - Estimate the number of cells robot can sense (e.g., simulate laser beams using current map)
  - Estimate the cost of getting there



Exploration

- **Greedy strategy:** Select the candidate location with the highest utility, then repeat…

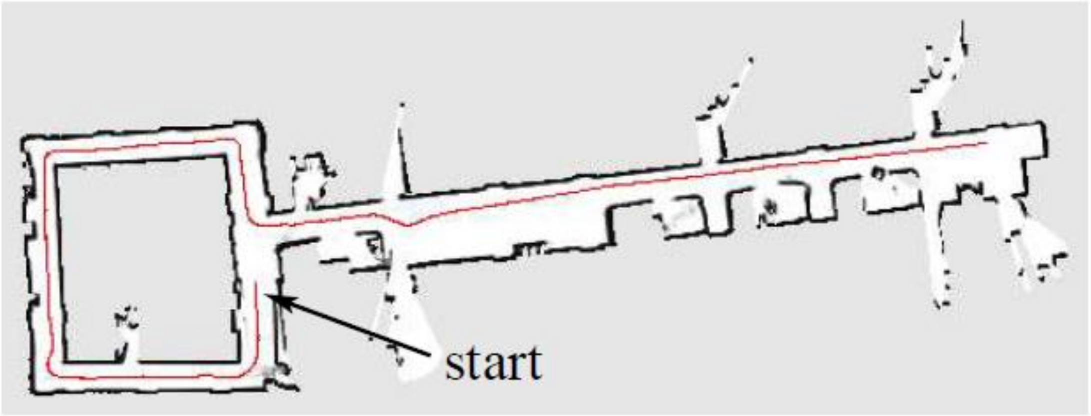


Exploration Actions

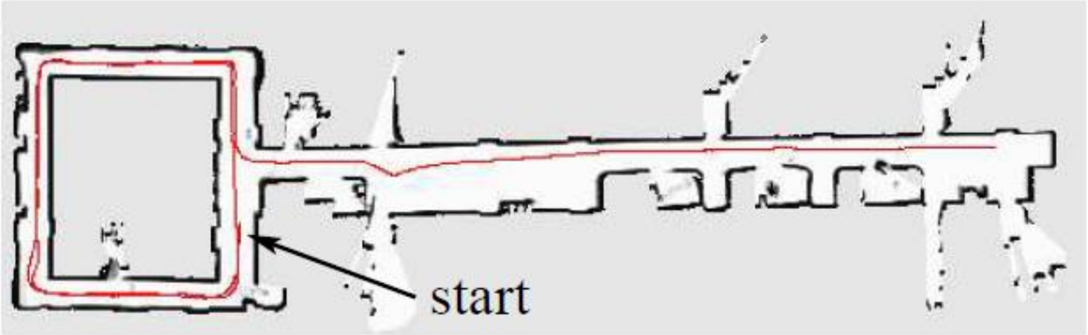
- So far, we only considered reduction in map uncertainty
- In general, there are many sources of uncertainty that can be reduced by exploration
  - Map uncertainty (visit unexplored areas)
  - Trajectory uncertainty (loop closing)
  - Localization uncertainty (active re-localization by re-visiting known locations)

Example: Active Loop Closing
[Stachniss et al., 2005]

- Reduce map uncertainty

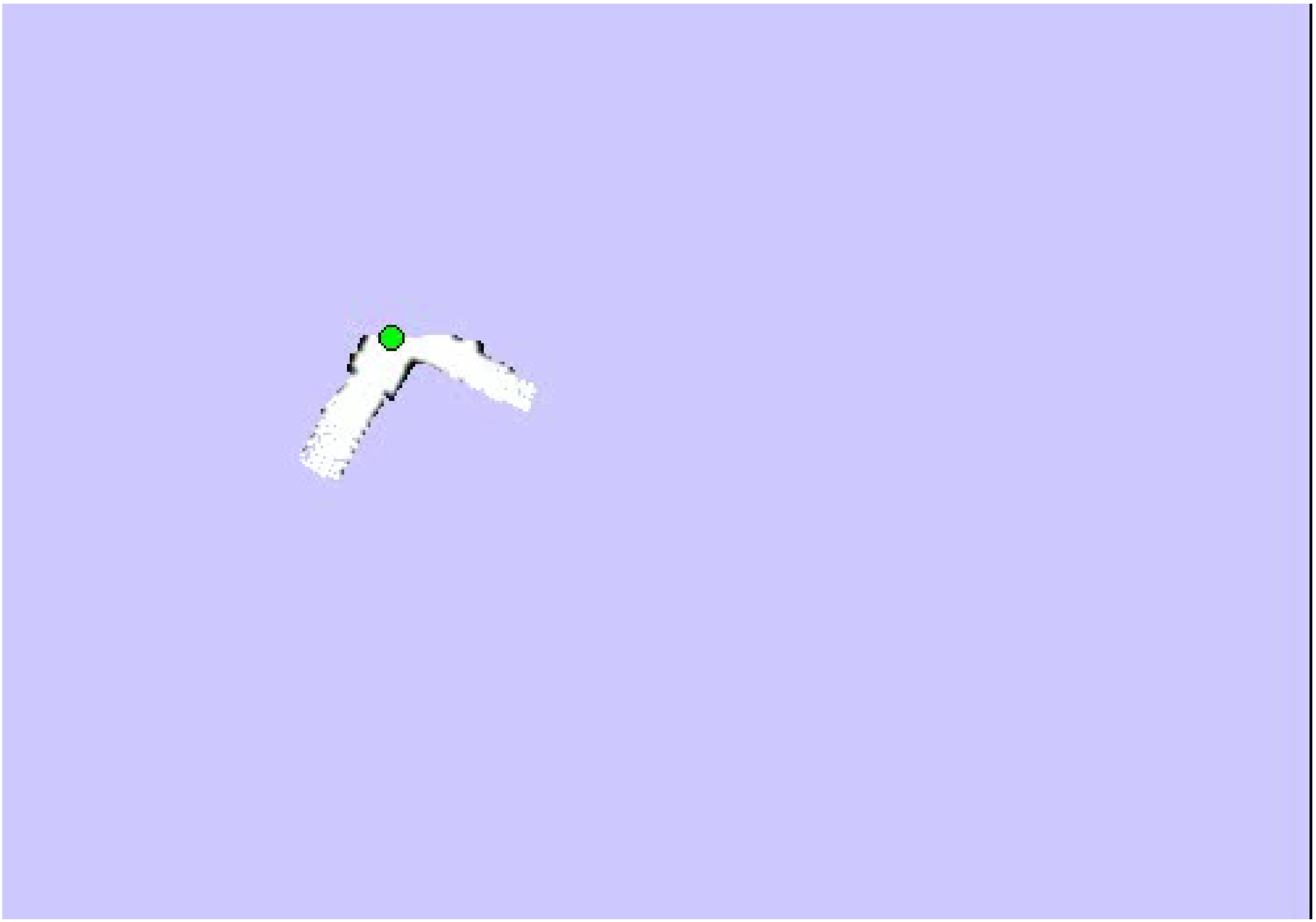


- Reduce map + path uncertainty



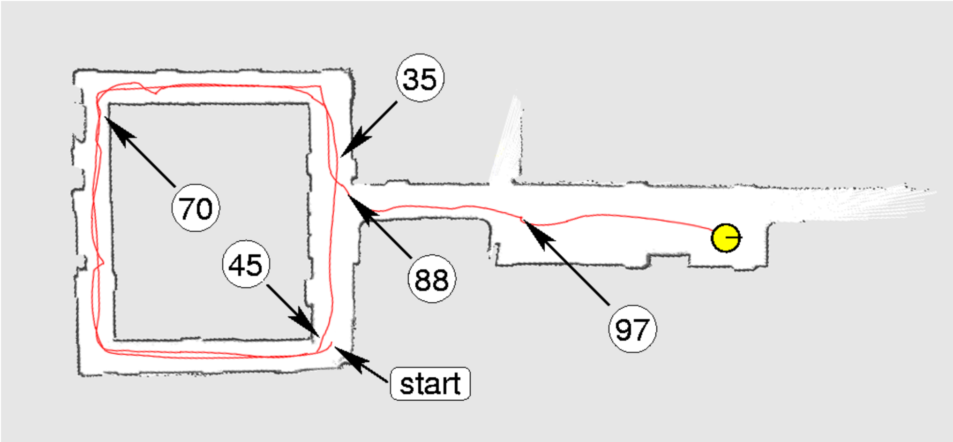
Example: Active Loop Closing

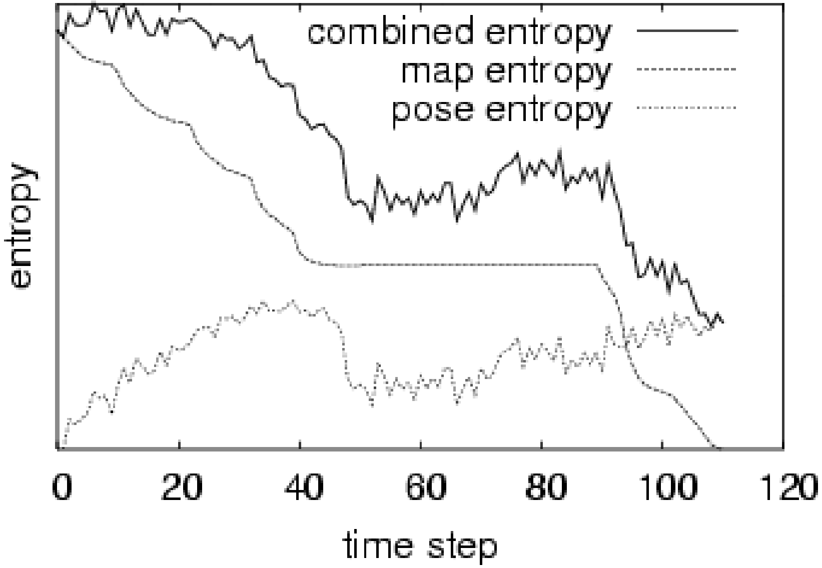
[Stachniss et al., 2005]



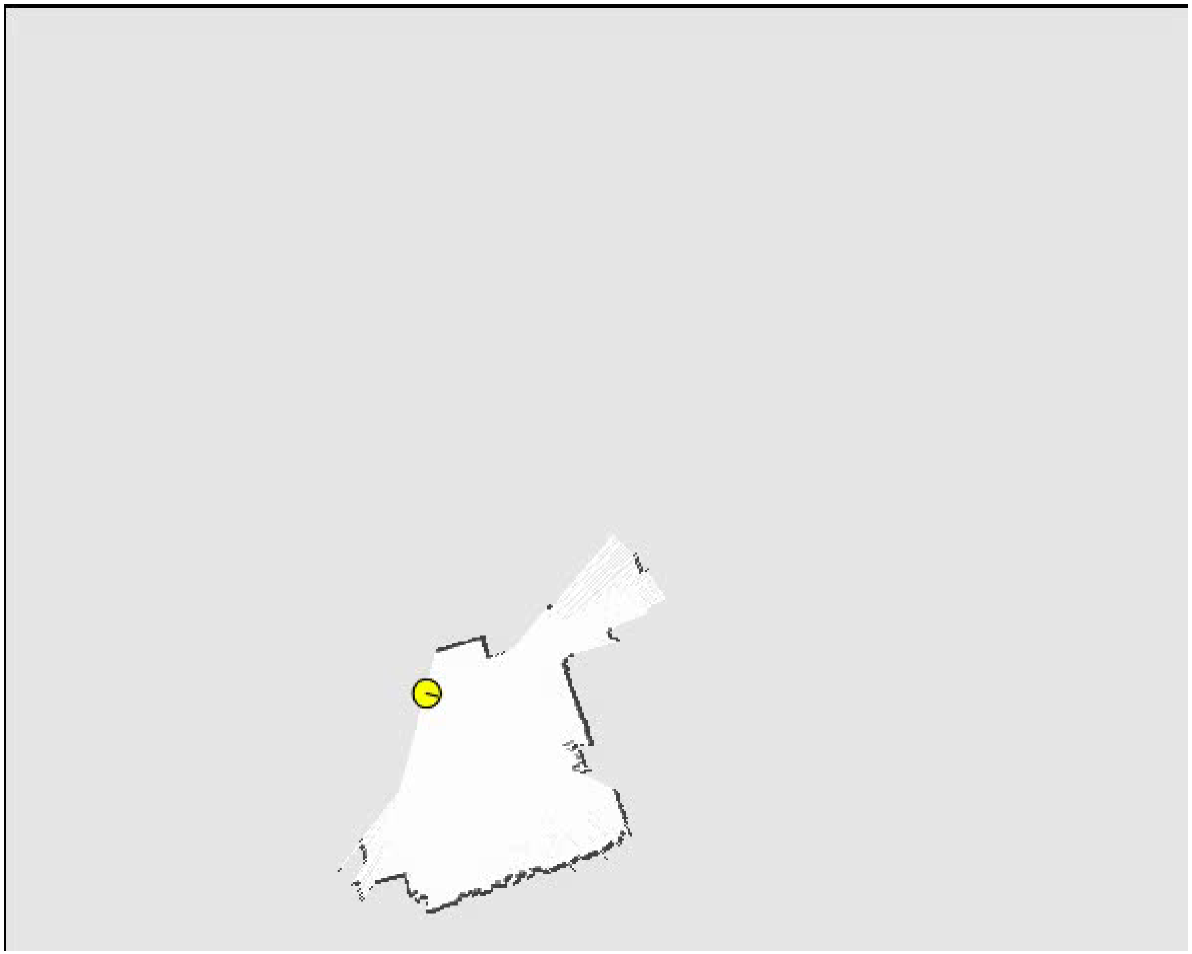
Example: Active Loop Closing
[Stachniss et al., 2005]

- Entropy evolution

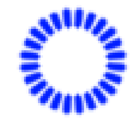




Example: Reduce uncertainty in map, path, and pose [Stachniss et al., 2005]

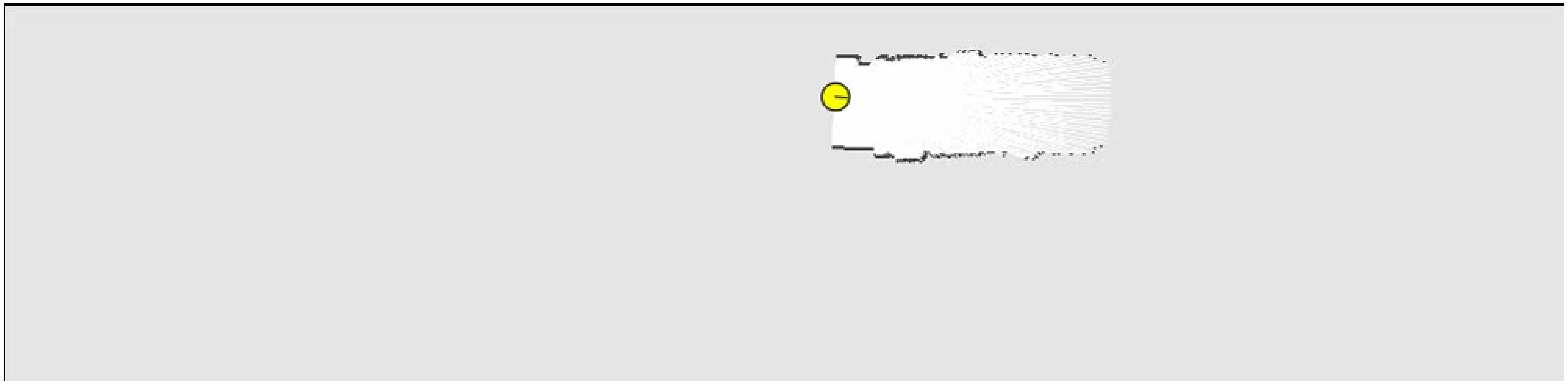


Selected target location



Corridor Exploration

[Stachniss et al., 2005]

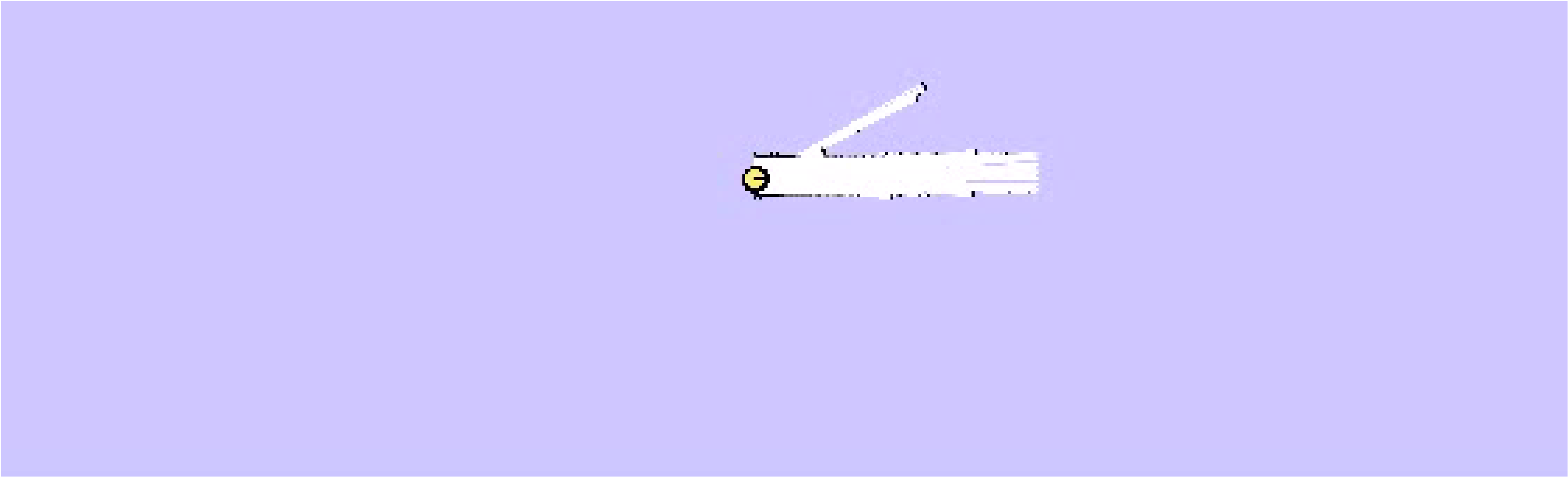


- The decision-theoretic approach leads to **intuitive behaviors:** "re-localize before getting lost"
- Some animals show a similar behavior

Multi-Robot Exploration

Given: Team of robots with communication

Goal: Explore the environment as fast as possible



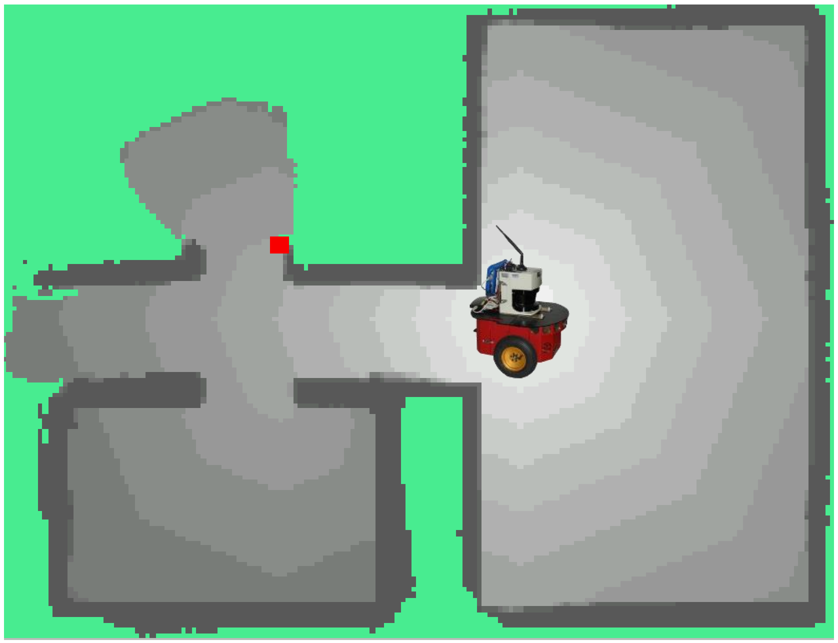
[Wurm et al., IROS 2011]

Complexity

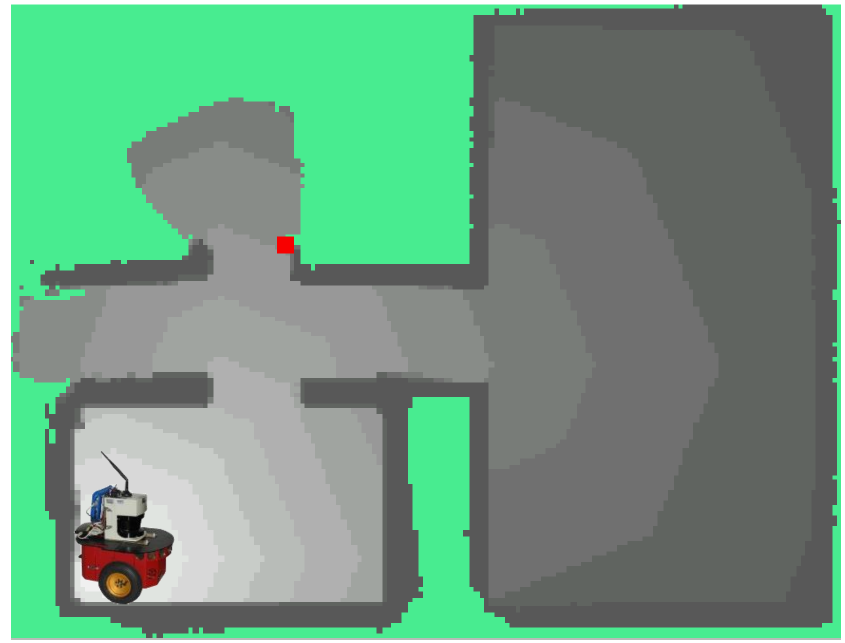
- Single-robot exploration in known, graph-like environments is in general **NP-hard**
- Proof: Reduce traveling salesman problem to exploration
- Complexity of multi-robot exploration is **exponential** in the number of robots

Motivation: Why Coordinate?

Robot 1



Robot 2



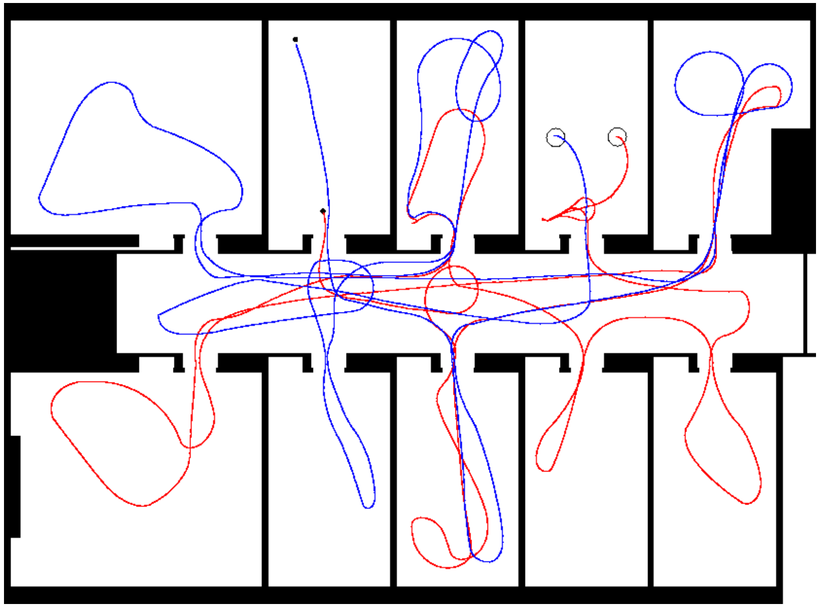
- Without coordination, two robots might choose the same exploration frontier

Levels of Coordination

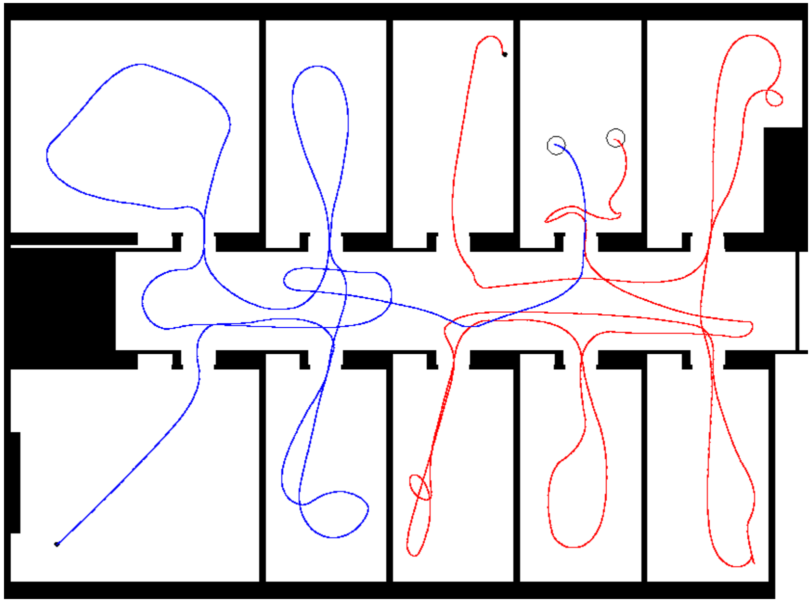
1. **No exchange of information**
2. **Implicit coordination**: Sharing a joint map
   - Communication of the individual maps and poses
   - Central mapping system
3. **Explicit coordination:** Determine better target locations to distribute the robots
   - Central planner for target point assignment
   - Minimize expected path cost / information gain / …

Typical Trajectories

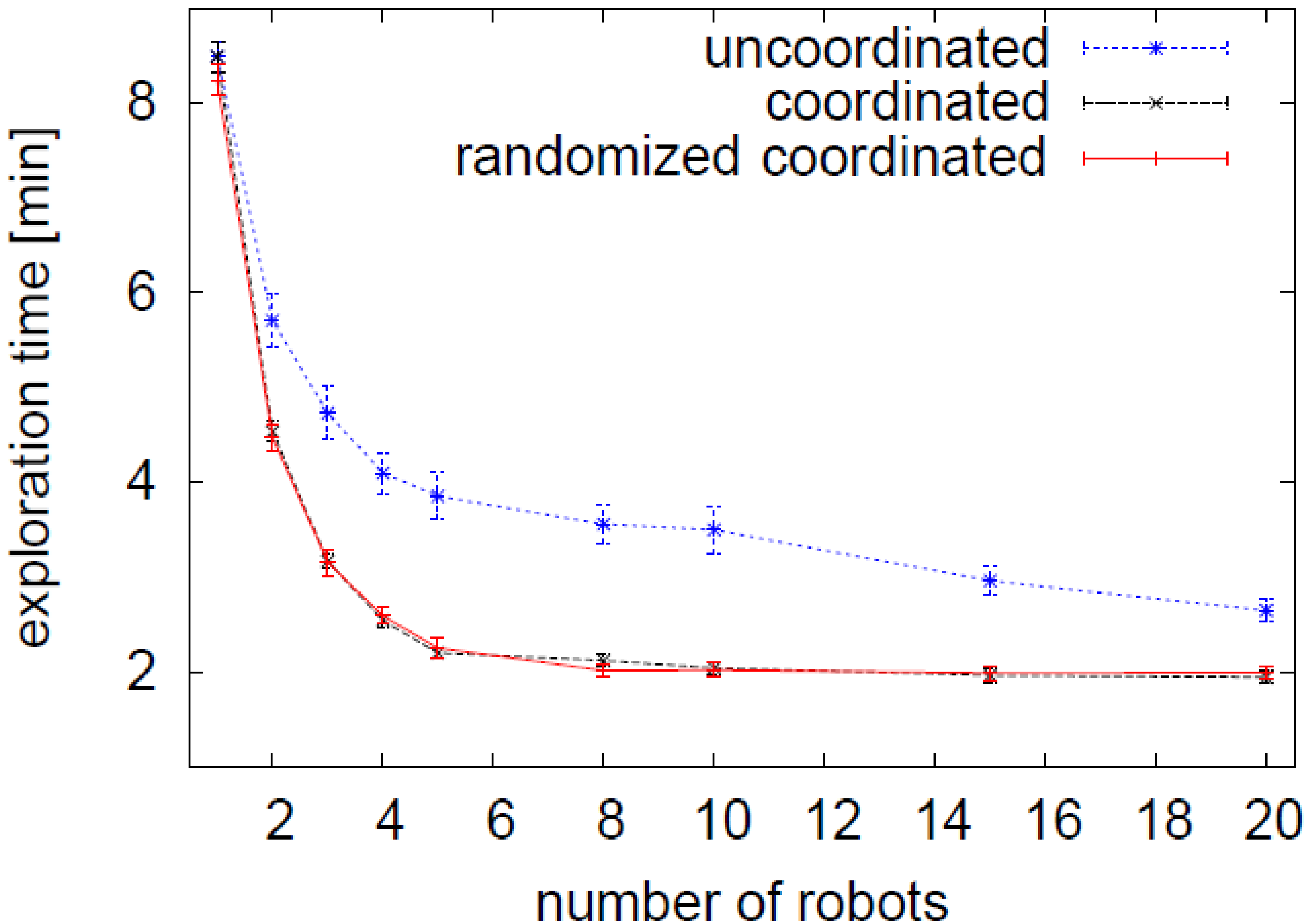
Implicit coordination:



Explicit coordination:



Exploration Time
[Stachniss et al., 2006]



Coordination Algorithm

In each time step:

- Determine set of exploration targets

$$S = \{s_1, \dots, s_n\}$$

- Compute for each robot i and each target j the expected cost/utility C_{ij}
- Assign robots to targets using the **Hungarian algorithm**

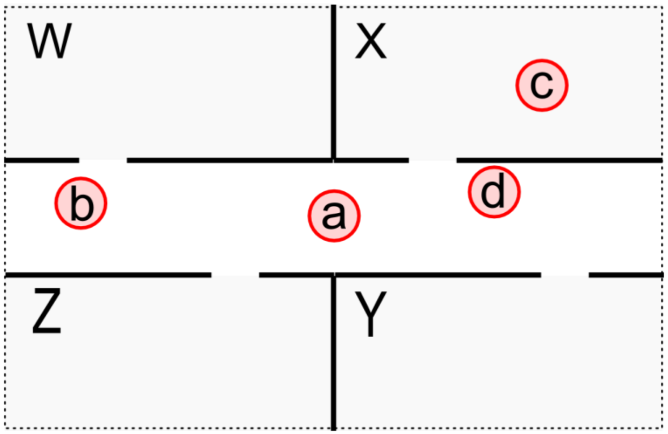
Hungarian Algorithm
[Kuhn, 1955]

- Combinatorial optimization algorithm
- Solves the assignment problem in polynomial time $O(n^3)$
- General idea: Algorithm modifies the cost matrix until there is zero cost assignment

Hungarian Algorithm: Example

targets

robots	W	X	Y	Z
a	3	2	3	2
b	2	5	6	3
c	7	1	3	5
d	6	2	3	5



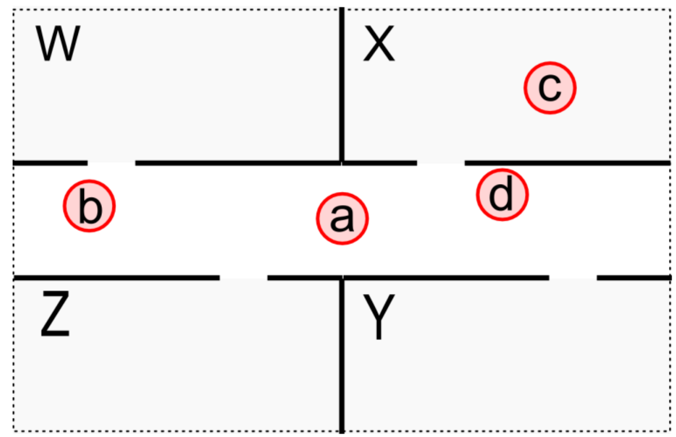
1. Compute the cost matrix (non-negative)

Hungarian Algorithm: Example

targets

robots

	W	X	Y	Z
a	3	2	3	2
b	2	5	6	3
c	7	1	3	5
d	6	2	3	5



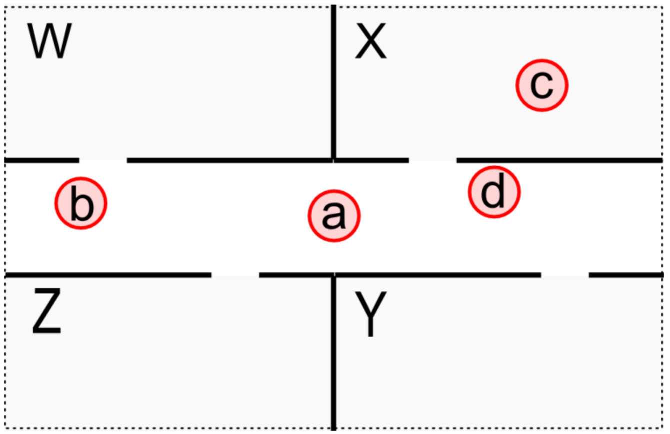
2. Find minimum element in each row

Hungarian Algorithm: Example

targets

robots

	W	X	Y	Z	
a	3	2	3	2	2
b	2	5	6	3	2
c	7	1	3	5	1
d	6	2	3	5	2

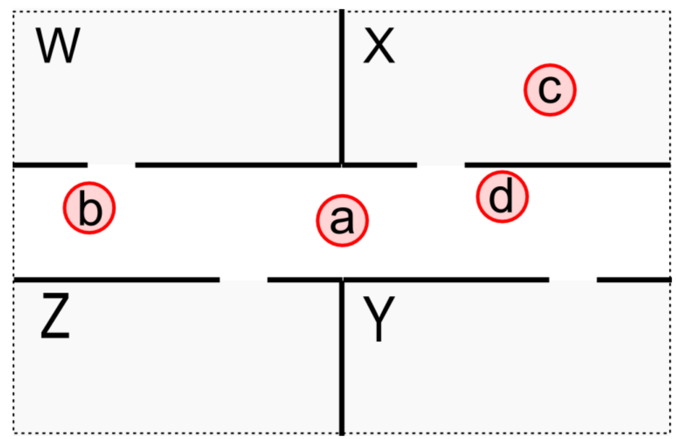


3. Subtract minimum from each row element

Hungarian Algorithm: Example

targets

robots	W	X	Y	Z
a	1	0	1	0
b	0	3	4	1
c	6	0	2	4
d	4	0	1	3
	0	0	1	0

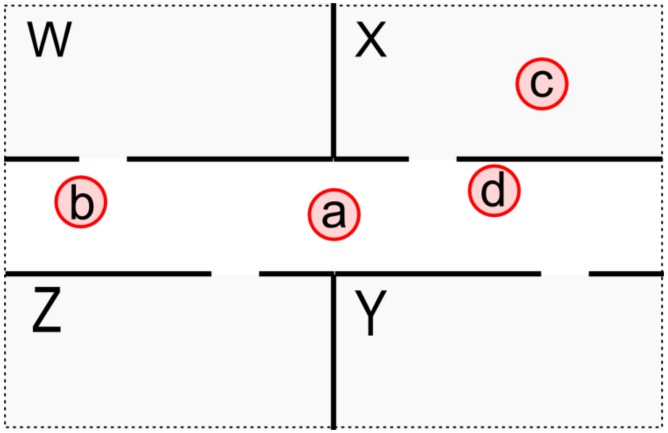


4. Find minimum element in each column

Hungarian Algorithm: Example

targets

robots	W	X	Y	Z
a	1	0	0	0
b	0	3	3	1
c	6	0	1	4
d	4	0	0	3
	0	0	1	0

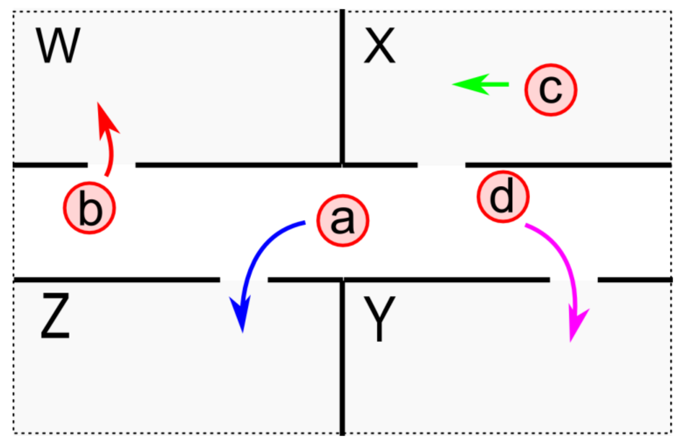


5. Subtract minimum from each column element

Hungarian Algorithm: Example

targets

robots	W	X	Y	Z
a	1	0	0	$\boxed{0}$
b	$\boxed{0}$	3	3	1
c	6	$\boxed{0}$	1	4
d	4	0	$\boxed{0}$	3



6a. Assign (if possible)

Hungarian Algorithm: Example

targets

robots	W	X	Y	Z
a	1	0	0	$\boxed{0}$
b	$\boxed{0}$	3	3	1
c	6	$\boxed{0}$	1	4
d	4	0	$\boxed{0}$	3

6b. If no assignment is possible:

- Connect all 0’s by vertical/horizontal lines
- Find the minimum in all remaining elements and subtract
- Add to all double crossed out coefficients
- Repeat step 2 – 6

Hungarian Algorithm: Example

targets

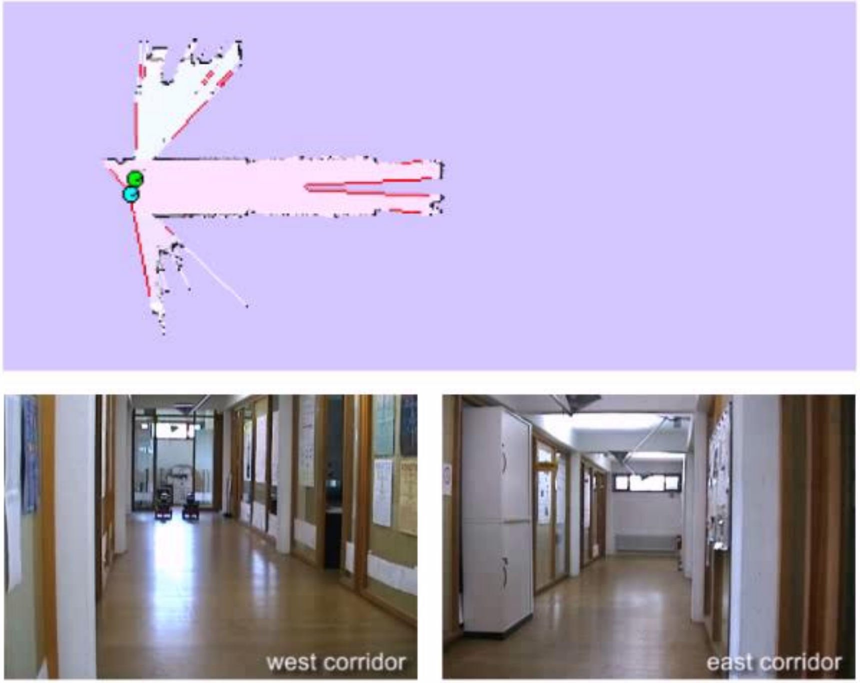
robots	X	Y	X'	Y'
a	2	3	2	3
b	5	6	5	6
c	1	3	1	3
d	2	3	2	3

If there are not enough targets:
Copy targets to allow multiple assignments

Example: Segmentation-based Exploration

[Wurm et al., IROS 2008]

- Two-layer hierarchical role assignments using Hungarian algorithm (1: rooms, 2: targets in room)
- Reduces exploration time and risk of interferences

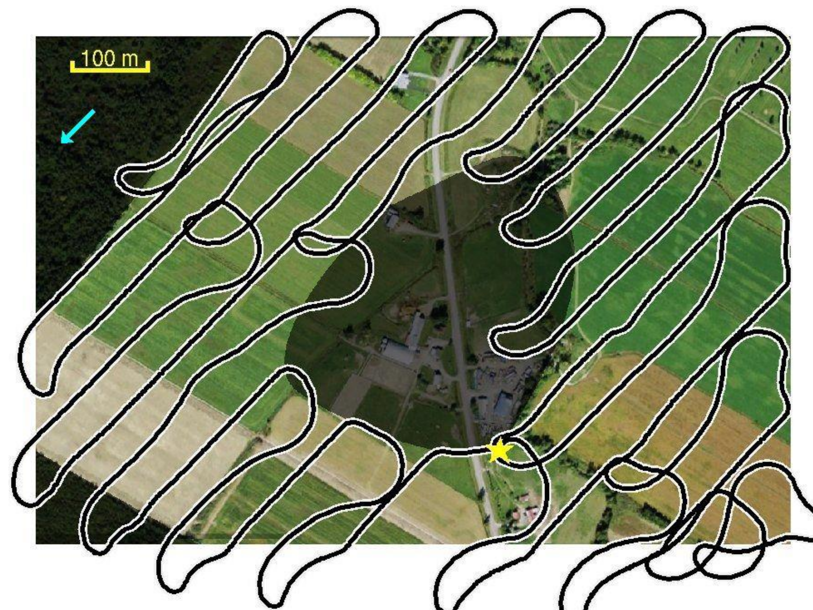


Summary: Exploration

- Exploration aims at generating robot motions so that an **optimal map** is obtained
- **Coordination** reduces exploration time
- **Hungarian algorithm** efficiently solves the assignment problem (centralized, 1-step lookahead)
- Challenges (active research):
  - Limited bandwidth and **unreliable communication**
  - **Decentralized planning** and task assignment

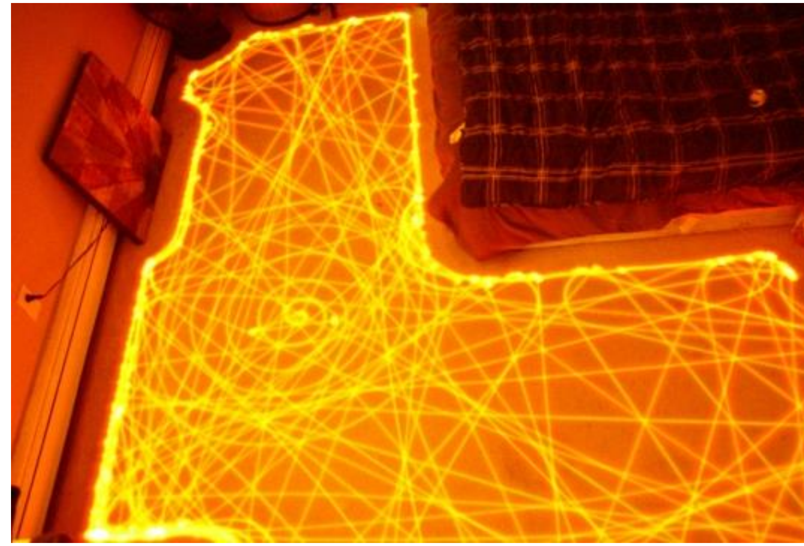
Coverage Path Planning

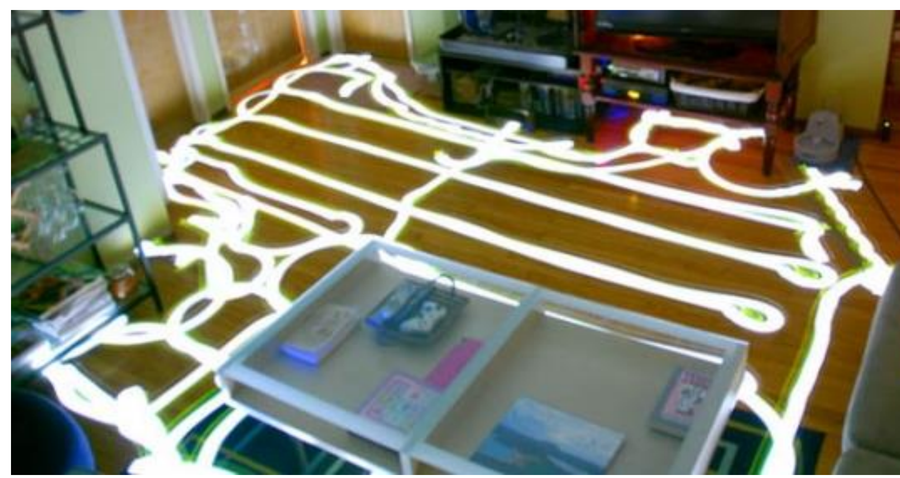
- **Given:** Known environment with obstacles
- **Wanted:** The shortest trajectory that ensures complete (sensor) coverage



[images from Xu et al., ICRA 2011]

Coverage Path Planning





Coverage Path Planning: Applications

- For flying robots
  - Search and rescue
  - Area surveillance
  - Environmental inspection
  - Inspection of buildings (bridges)
- For service robots
  - Lawn mowing
  - Vacuum cleaning
- For manipulation robots
  - Painting
  - Automated farming

Coverage Path Planning

- What is a good coverage strategy?
- What would be a good cost function?

Coverage Path Planning

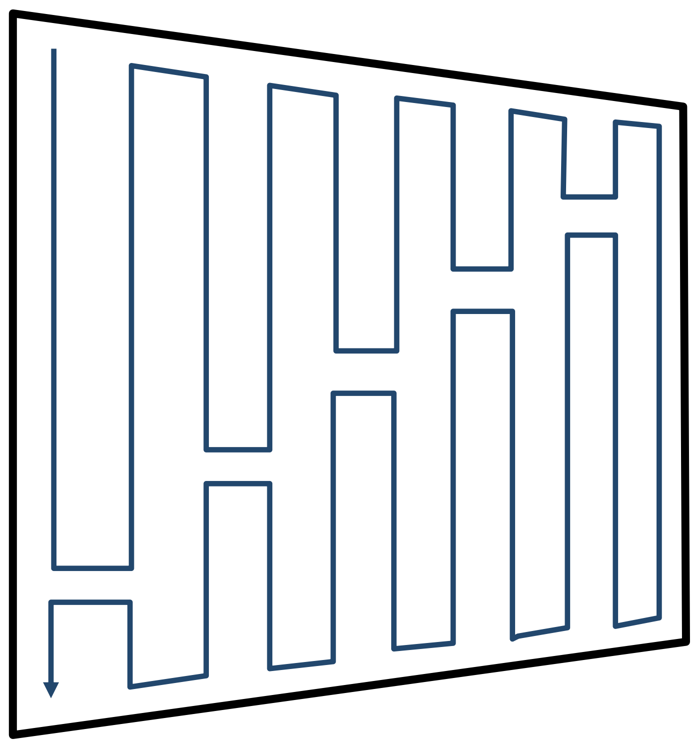
- What is a good coverage strategy?
- What would be a good cost function?
  - Amount of redundant traversals
  - Number of stops and rotations
  - Execution time
  - Energy consumption
  - Robustness
  - Probability of success
  - …

Coverage Path Planning

- Related to the traveling salesman problem (TSP):
  "Given a weighted graph, compute a path that visits every vertex once"
- In general **NP-complete**
- Many approximations exist
- Many approximate (and exact) solvers exist

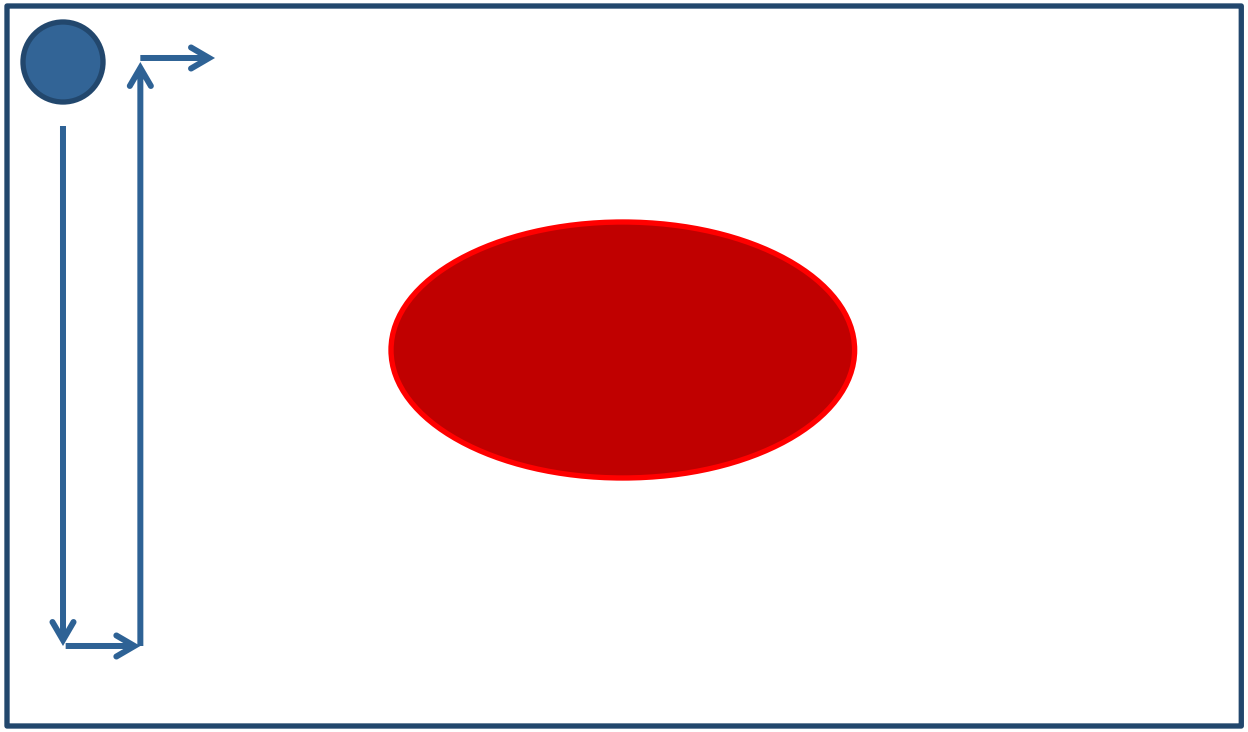
Coverage of Simple Shapes

- Approximately optimal solution often easy to compute for simple shapes (e.g., trapezoids)



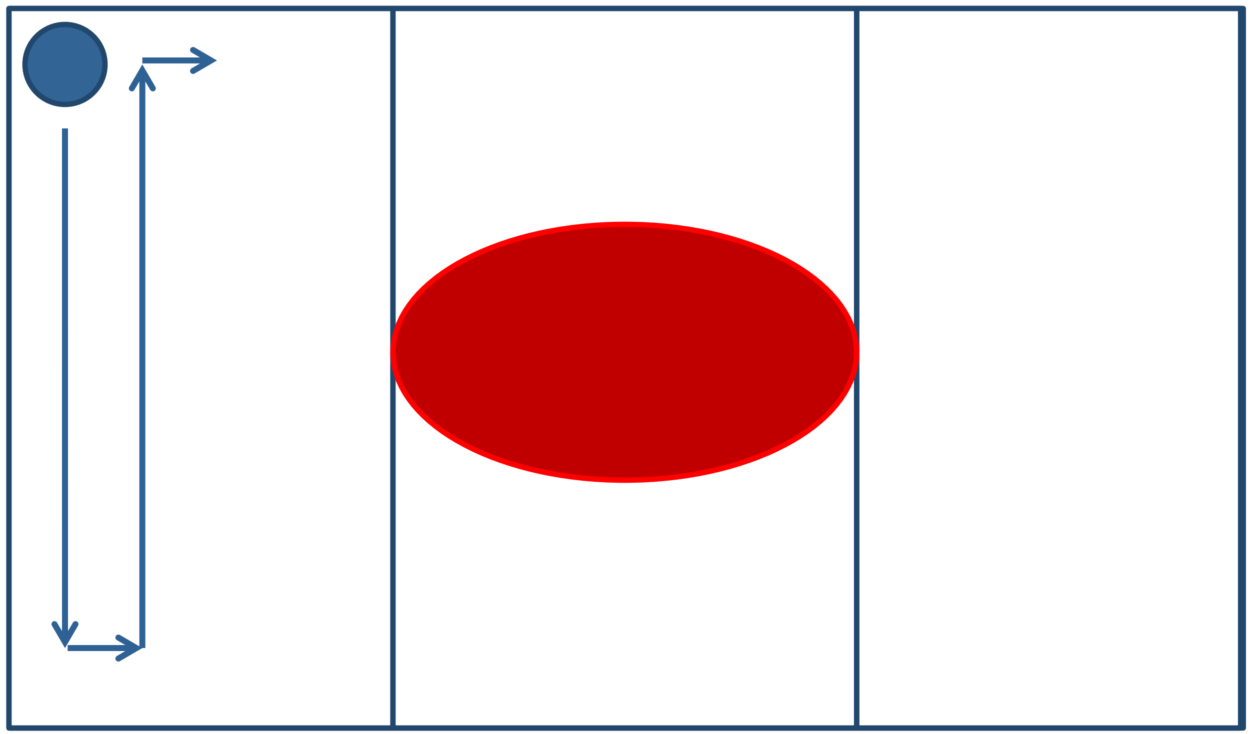
Idea

[Mannadiar and Rekleitis, ICRA 2011]

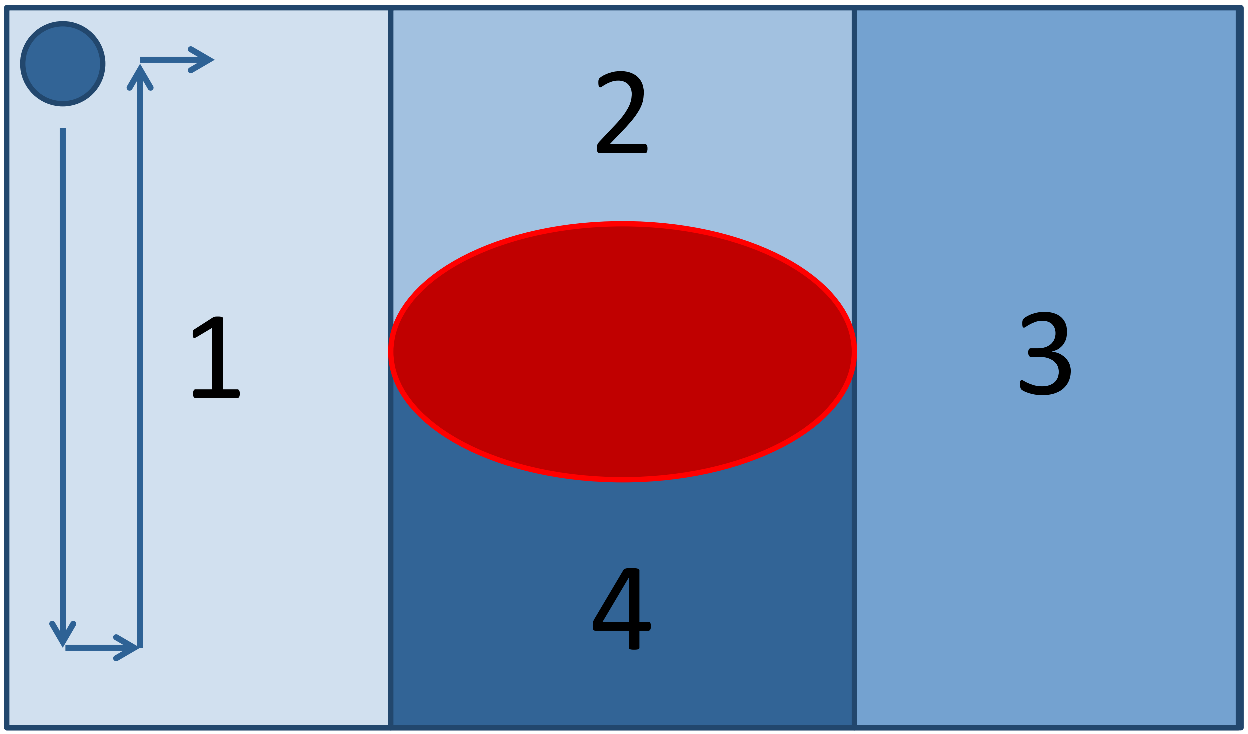


Idea

[Mannadiar and Rekleitis, ICRA 2011]



Idea
[Mannadiar and Rekleitis, ICRA 2011]



Coverage Based On Cell Decomposition
[Mannadiar and Rekleitis, ICRA 2011]

Approach:

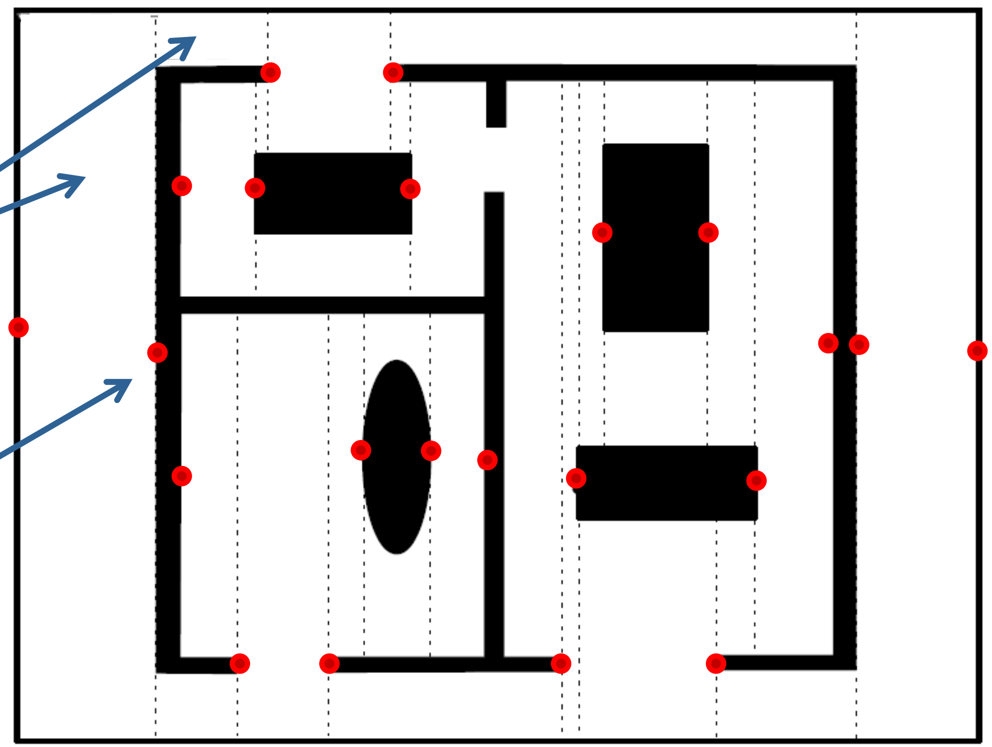
1. Decompose map into “simple” cells
2. Compute connectivity between cells and build graph
3. Solve coverage problem on reduced graph

Step 1: Boustrophedon Cellular Decomposition [Mannadiar and Rekleitis, ICRA 2011]

- Similar to trapezoidal decomposition
- Can be computed efficiently in 2D

cells

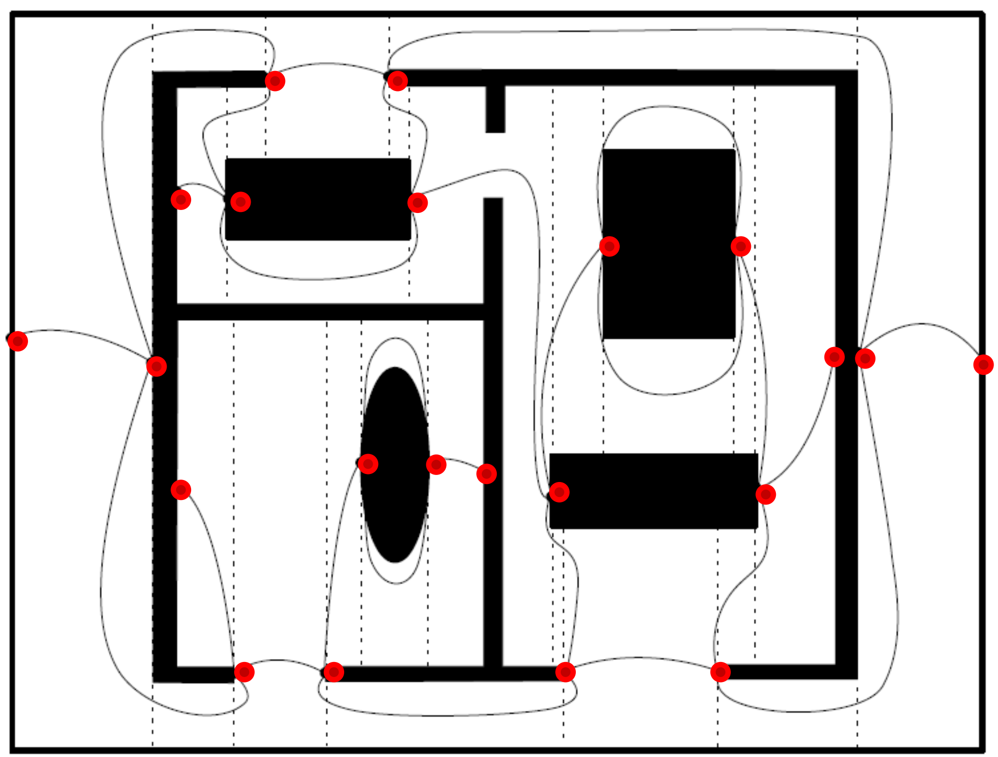
critical points (=produce splits or joins)



Step 2: Build Reeb Graph

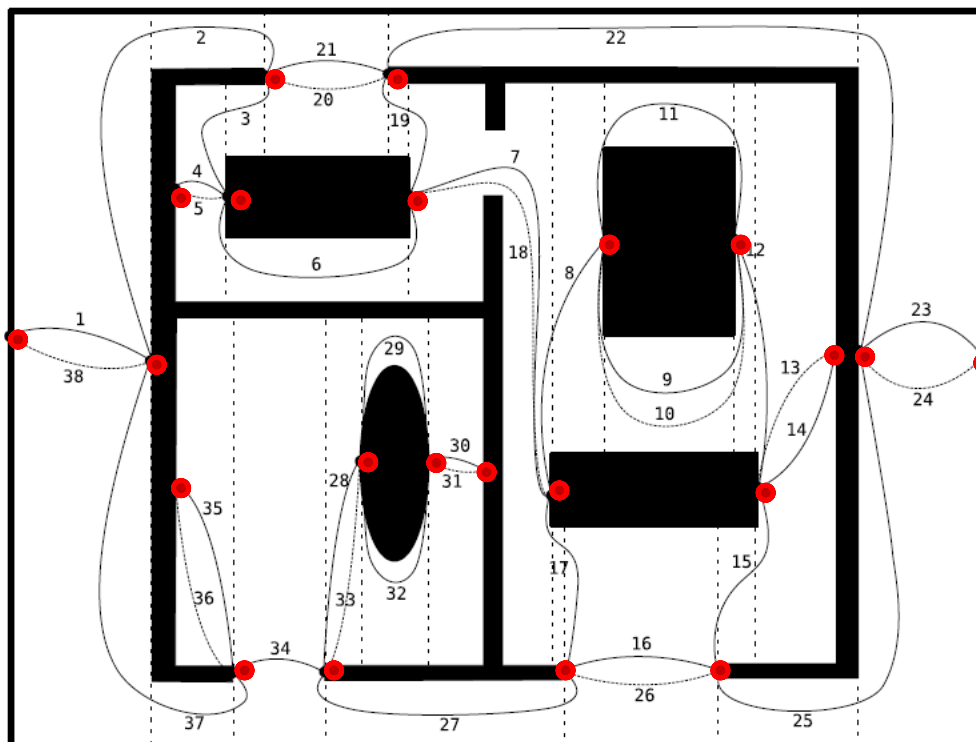
[Mannadiar and Rekleitis, ICRA 2011]

- Vertices = Critical points (that triggered the split)
- Edges = Connectivity between critical points



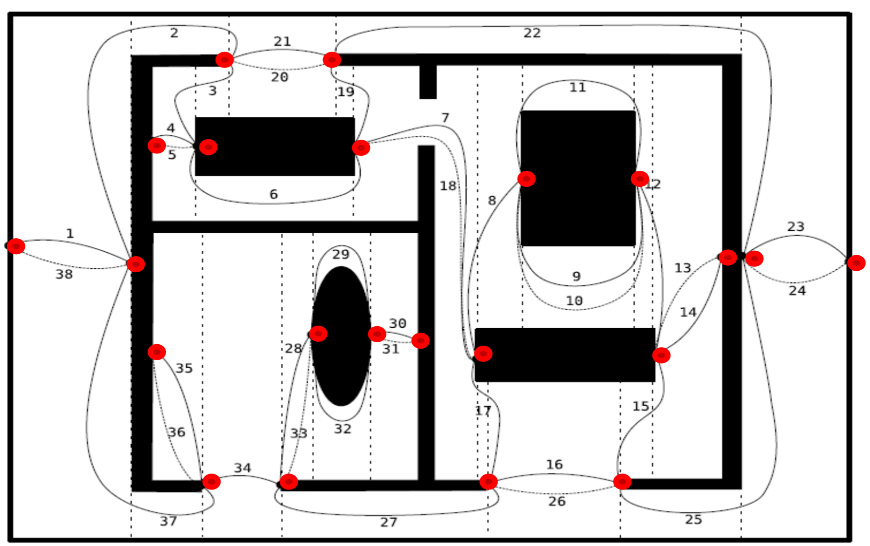
Step 3: Compute Euler Tour
[Mannadiar and Rekleitis, ICRA 2011]

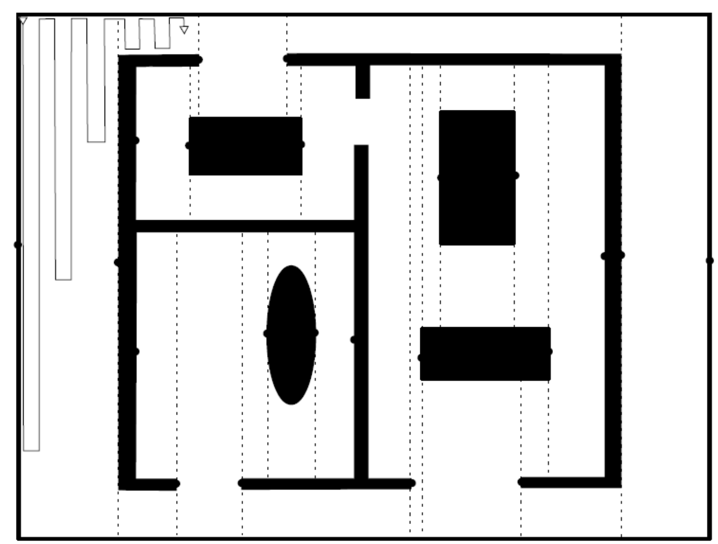
- Extend graph so that vertices have even order
- Compute Euler tour (linear time)



Resulting Coverage Plan
[Mannadiar and Rekleitis, ICRA 2011]

- Follow the Euler tour
- Use simple coverage strategy for cells
- Note: Cells are visited once or twice





What are the desired properties of a good scientific experiment?

- Reproducibility / repeatability
  - Document the experimental setup
  - Choose (and motivate) an your evaluation criterion
- Experiments should allow you to validate/falsify competing hypotheses

Current trends:

- Make data available for review and criticism
- Same for software (open source)

Benchmarks

- Effective and affordable way of conducting experiments
- Sample of a task domain
- Well-defined performance measurements
- Widely used in computer vision and robotics
- **Which benchmark problems do you know?**

Example Benchmark Problems

Computer Vision

- Middlebury datasets (optical flow, stereo, …)
- Caltech-101, PASCAL (object recognition)
- Stanford bunny (3d reconstruction)

Robotics

- RoboCup competitions (robotic soccer)
- DARPA challenges (autonomous car)
- SLAM datasets

Image Denoising: Lenna Image

- 512x512 pixel standard image for image compression and denoising
- Lena Söderberg, Playboy magazine Nov. 1972
- Scanned by Alex Sawchuck at USC in a hurry for a conference paper



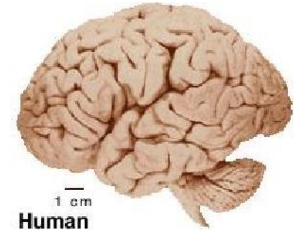

http://www.cs.cmu.edu/~chuck/lennapg/

Object Recognition: Caltech-101

- Pictures of objects belonging to 101 categories
- About 40-800 images per category
- Recognition, classification, categorization





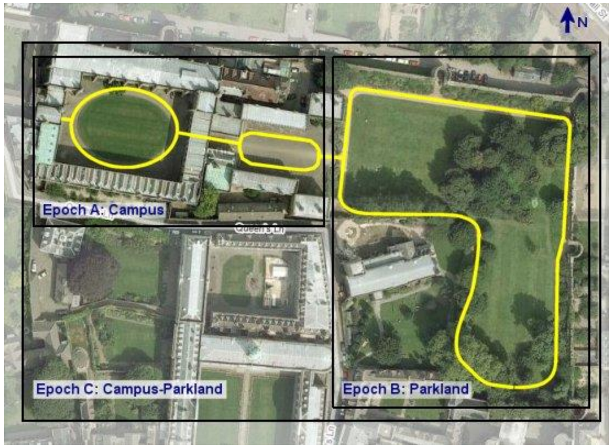
RoboCup Initiative

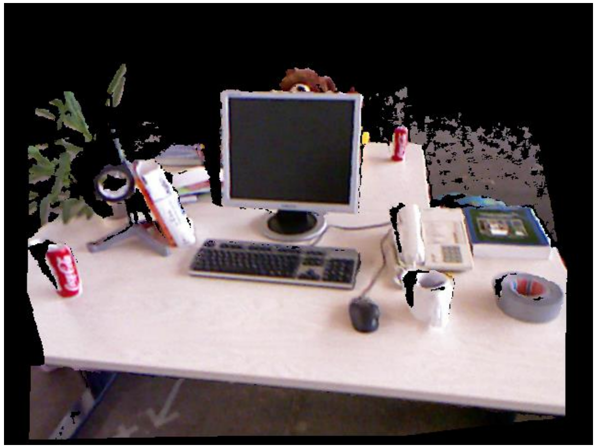
- Evaluation of full system performance
- Includes perception, planning, control, …
- Easy to understand, high publicity
- “By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, complying with the official rule of the FIFA, against the winner of the most recent World Cup.”

RoboCup Initiative

SLAM Evaluation

- Intel dataset: laser + odometry [Haehnel, 2004]
- New College dataset: stereo + omni-directional vision + laser + IMU [Smith et al., 2009]
- TUM RGB-D dataset [Sturm et al., 2011/12]
- …





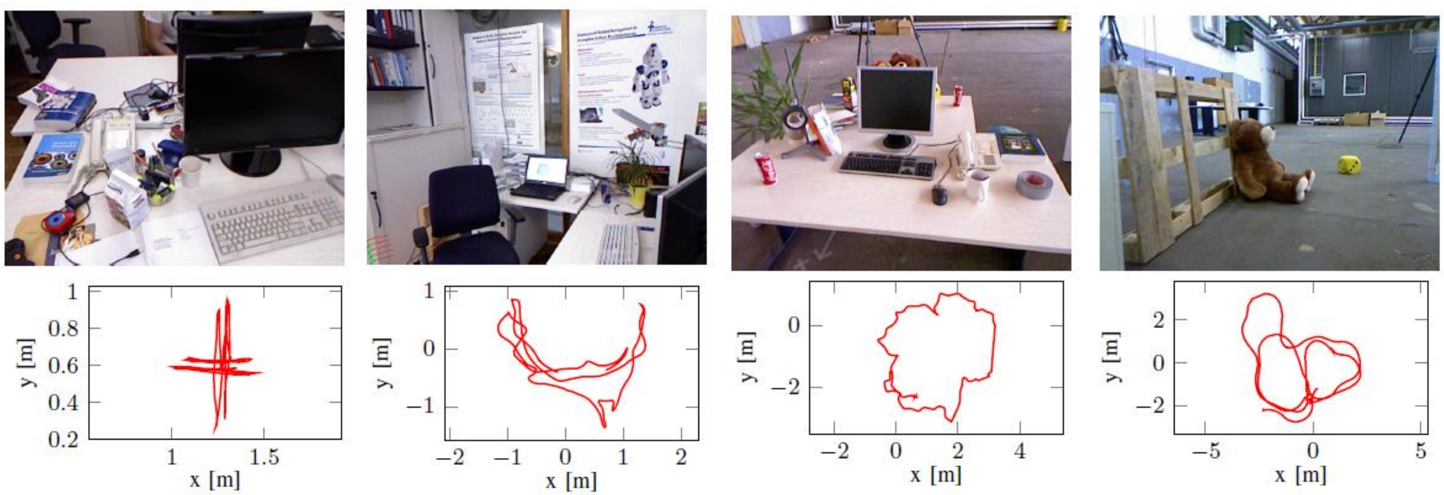
TUM RGB-D Dataset

[Sturm et al., RSS RGB-D 2011; Sturm et al., IROS 2012]

- RGB-D dataset with ground truth for SLAM evaluation
- Two error metrics proposed (relative and absolute error)
- Online + offline evaluation tools
- Training datasets (fully available)
- Validation datasets (ground truth not publicly available to avoid overfitting)

Recorded Scenes

- Various scenes (handheld/robot-mounted, office, industrial hall, dynamic objects, …)
- Large variations in camera speed, camera motion, illumination, environment size, …



(a) fr1/xyz (b) fr1/room (c) fr2/desk (d) fr2/slam

Dataset Acquisition

- Motion capture system
  - Camera pose (100 Hz)
- Microsoft Kinect
  - Color images (30 Hz)
  - Depth maps (30 Hz)
  - IMU (500 Hz)
- External video camera (for documentation)

Motion Capture System

- 9 high-speed cameras mounted in room
- Cameras have active illumination and pre-process image (thresholding)
- Cameras track positions of retro-reflective markers



Calibration

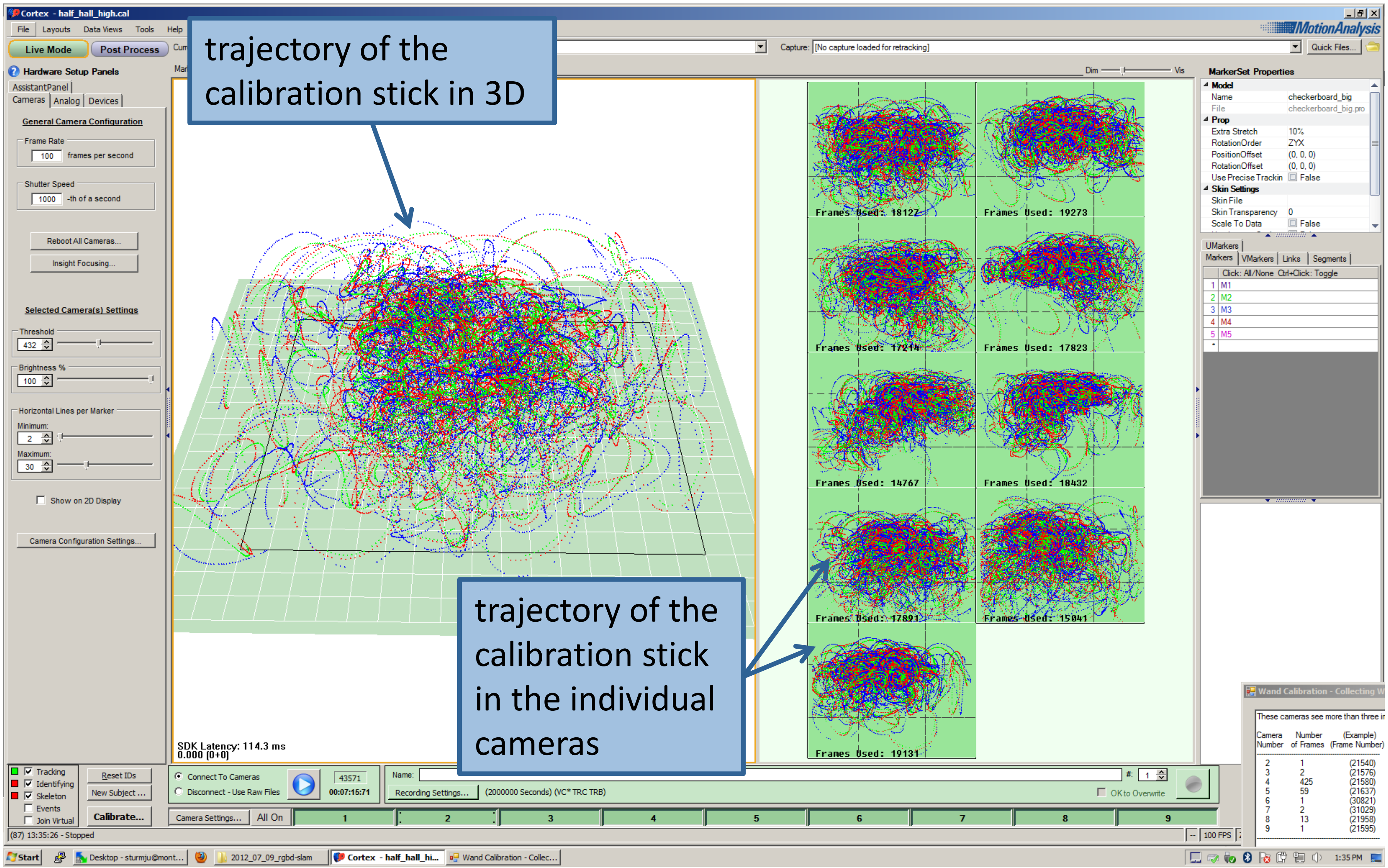
Calibration of the overall system is not trivial:

1. Mocap calibration
2. Kinect-mocap calibration
3. Time synchronization

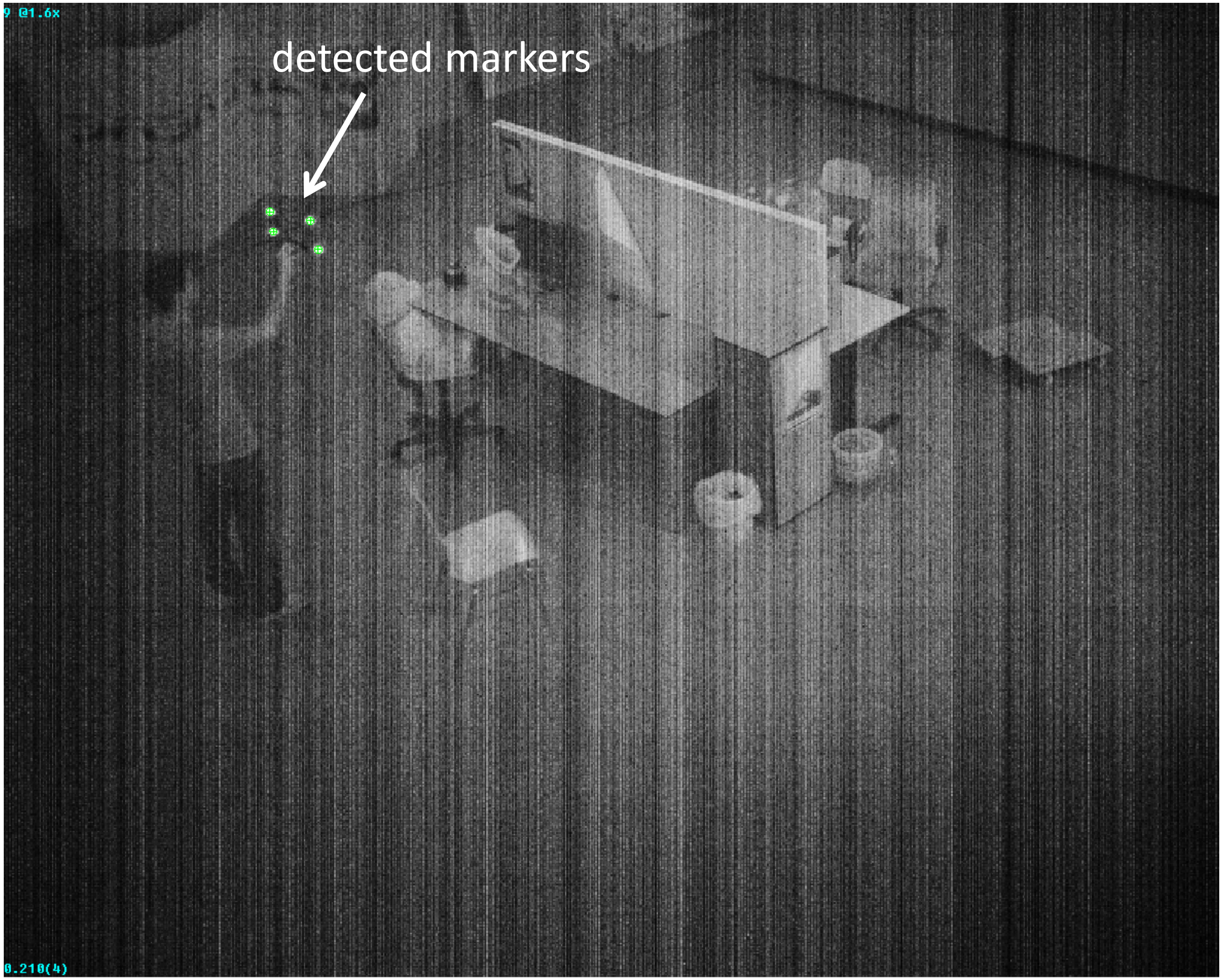
Calibration Step 1: Mocap

- Need at least 2 cameras for position fix
- Need at least 3 markers on object for full pose
- Calibration stick for extrinsic calibration

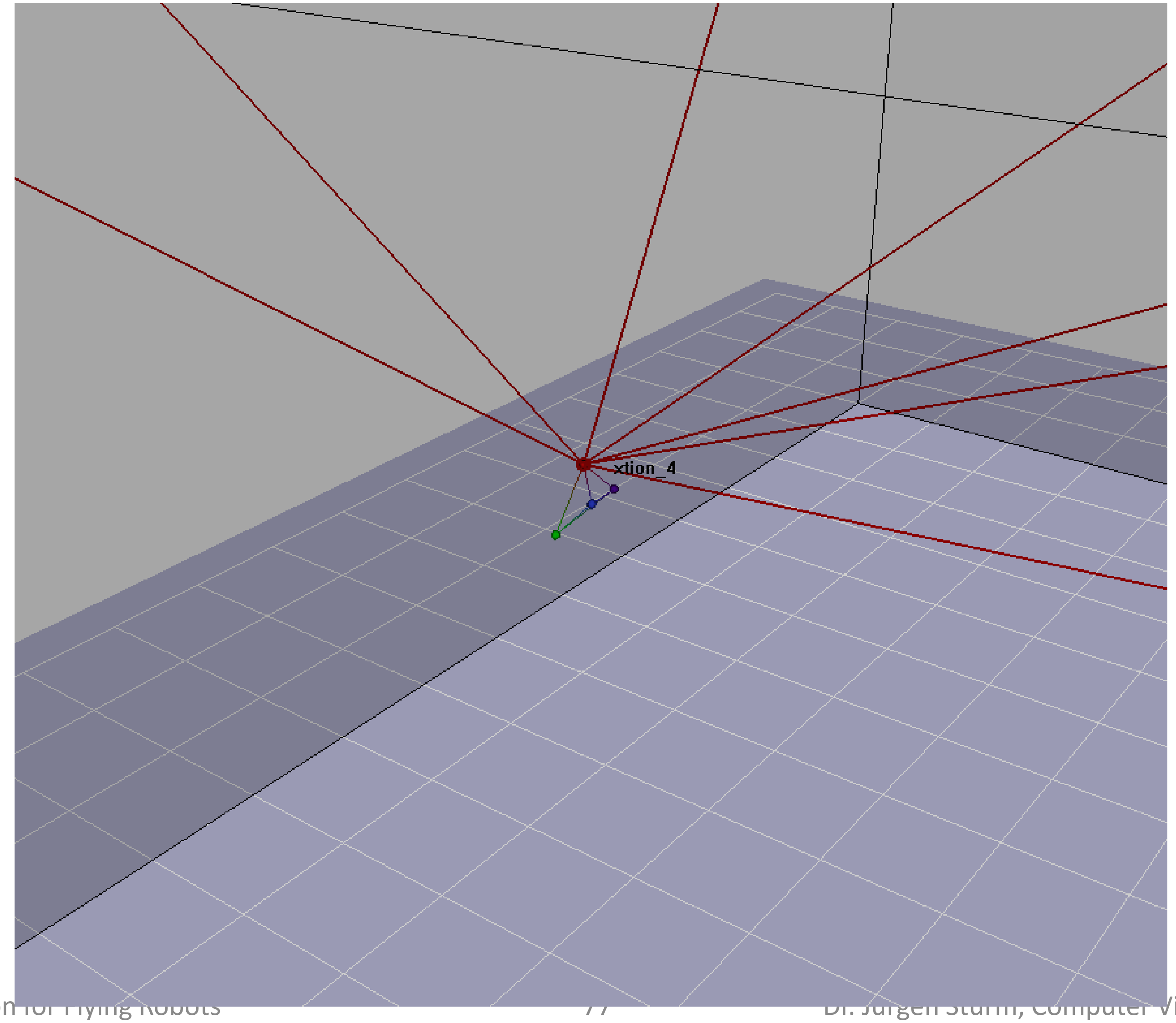
Calibration Step 1: Mocap



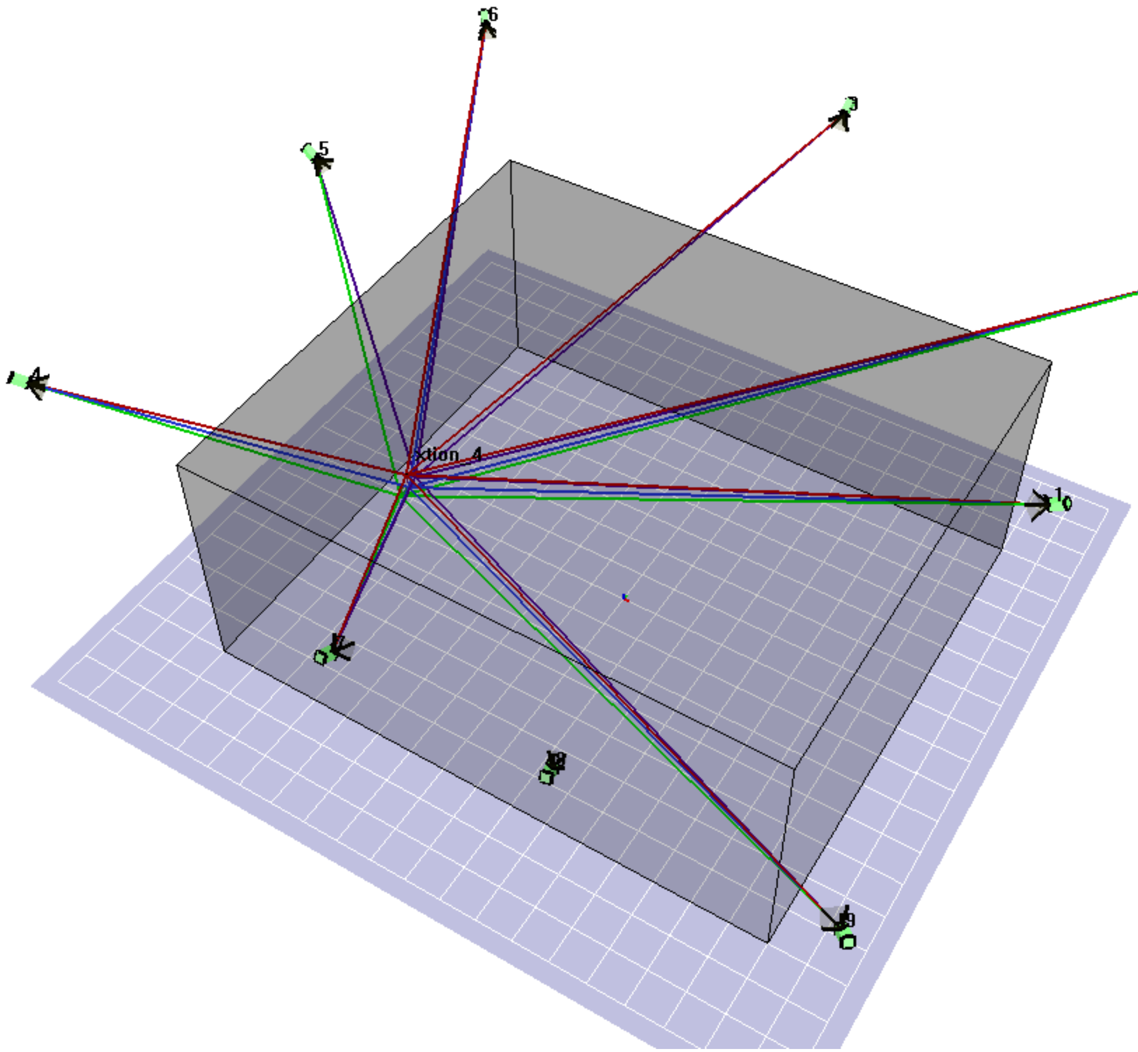
Example: Raw Image from Mocap



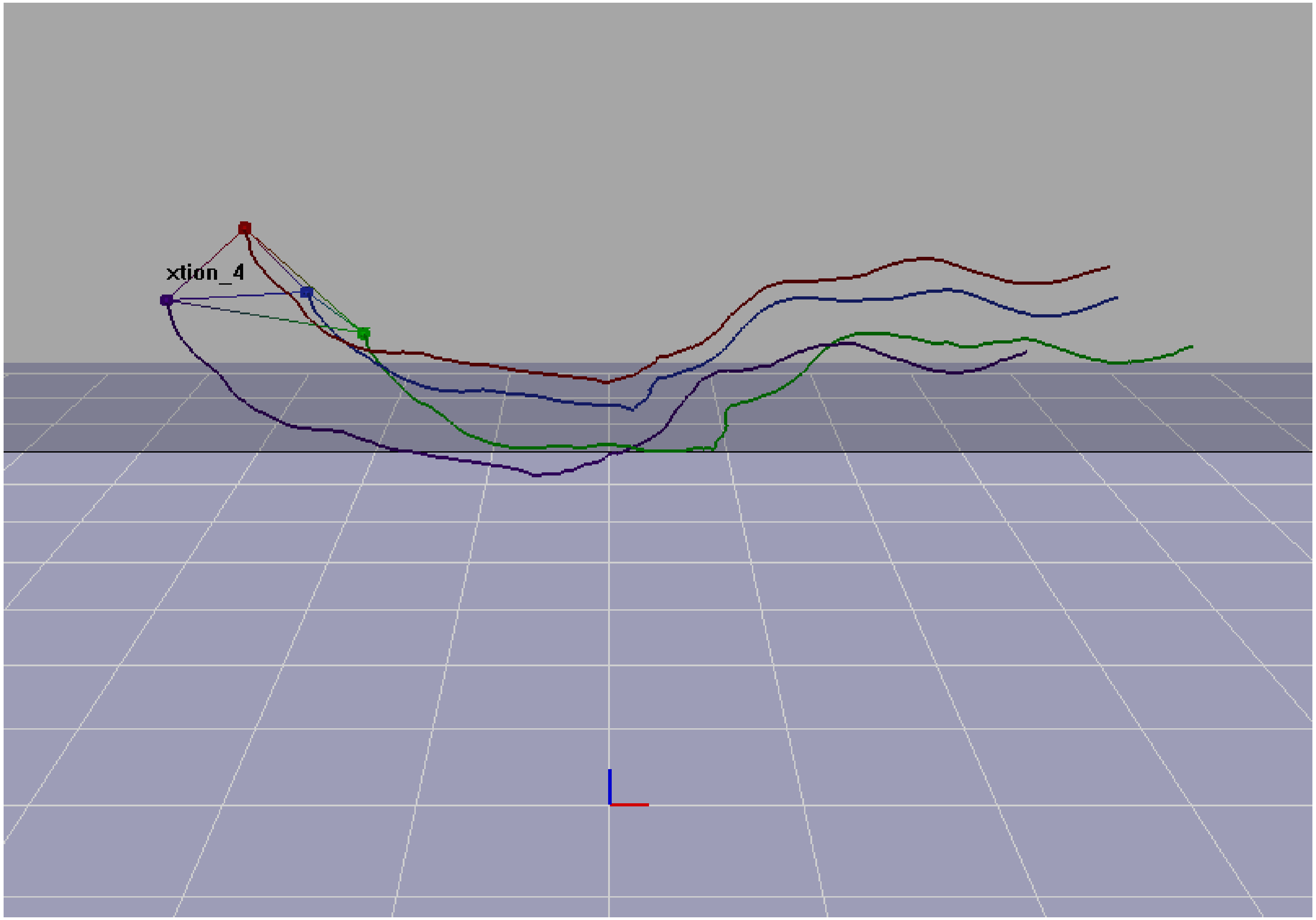
Example: Position Triangulation of a Single Marker



Example: Tracked Object (4 Markers)

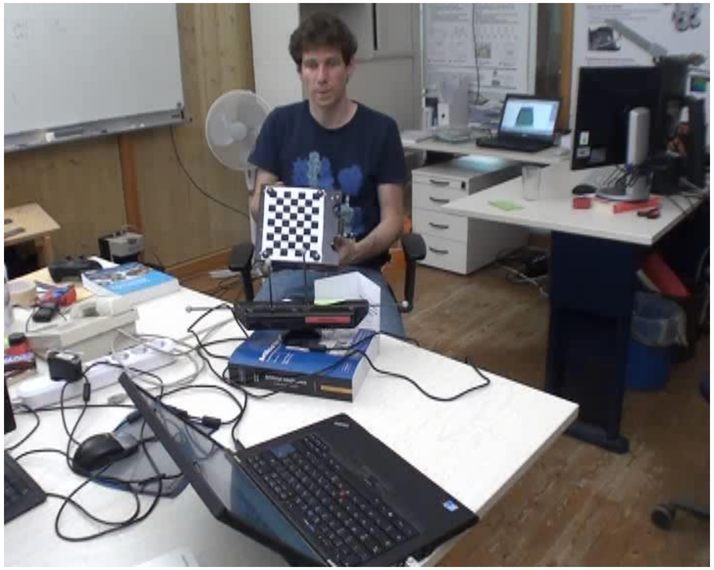


Example: Recorded Trajectory



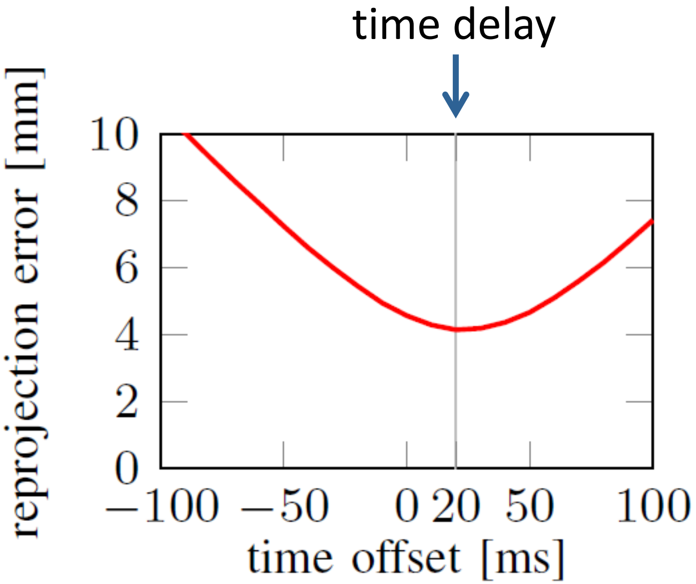
Calibration Step 2: Mocap-Kinect

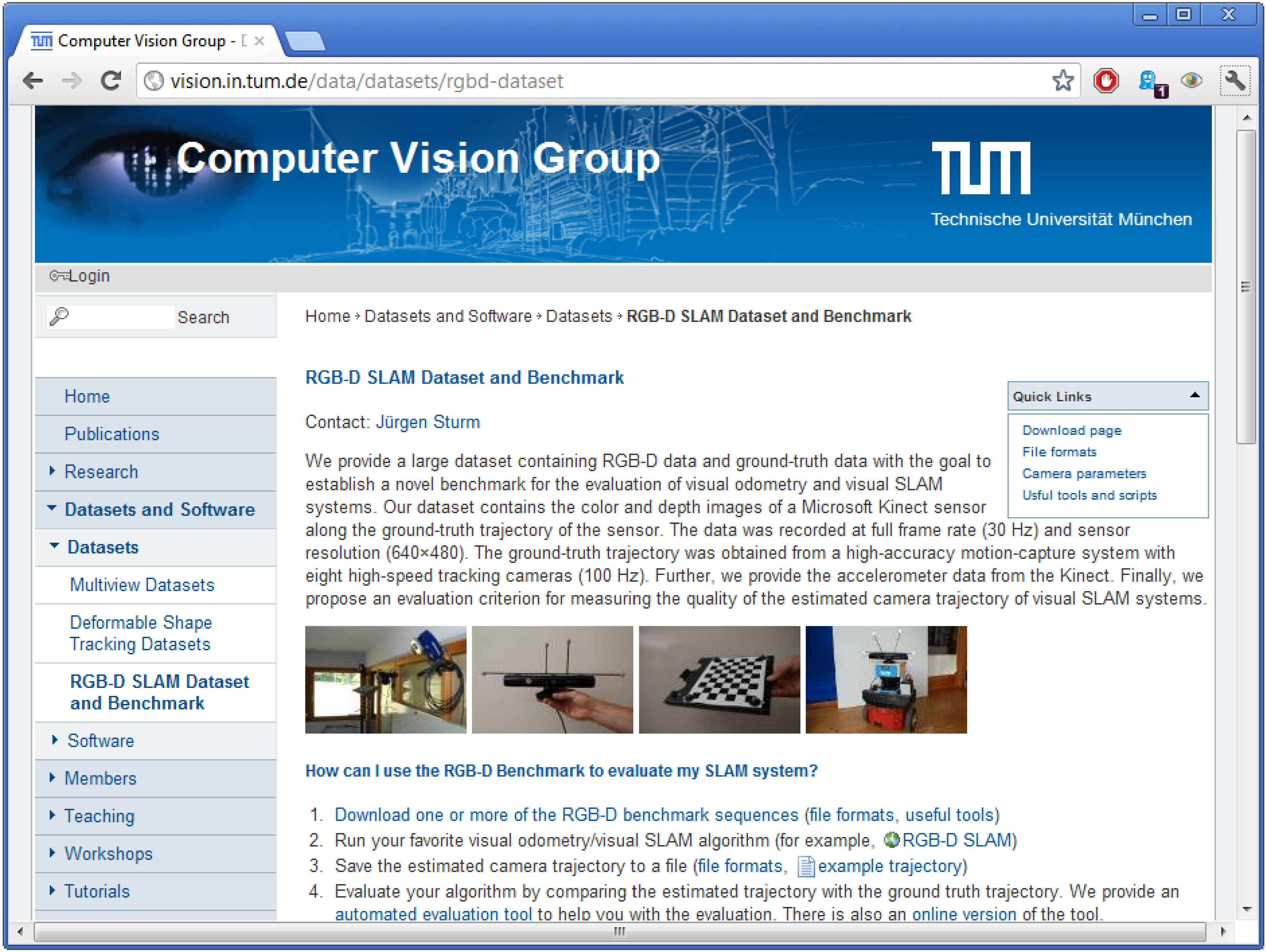
- Need to find transformation between the markers on the Kinect and the optical center
- Special calibration board visible both by Kinect and mocap system (manually gauged)

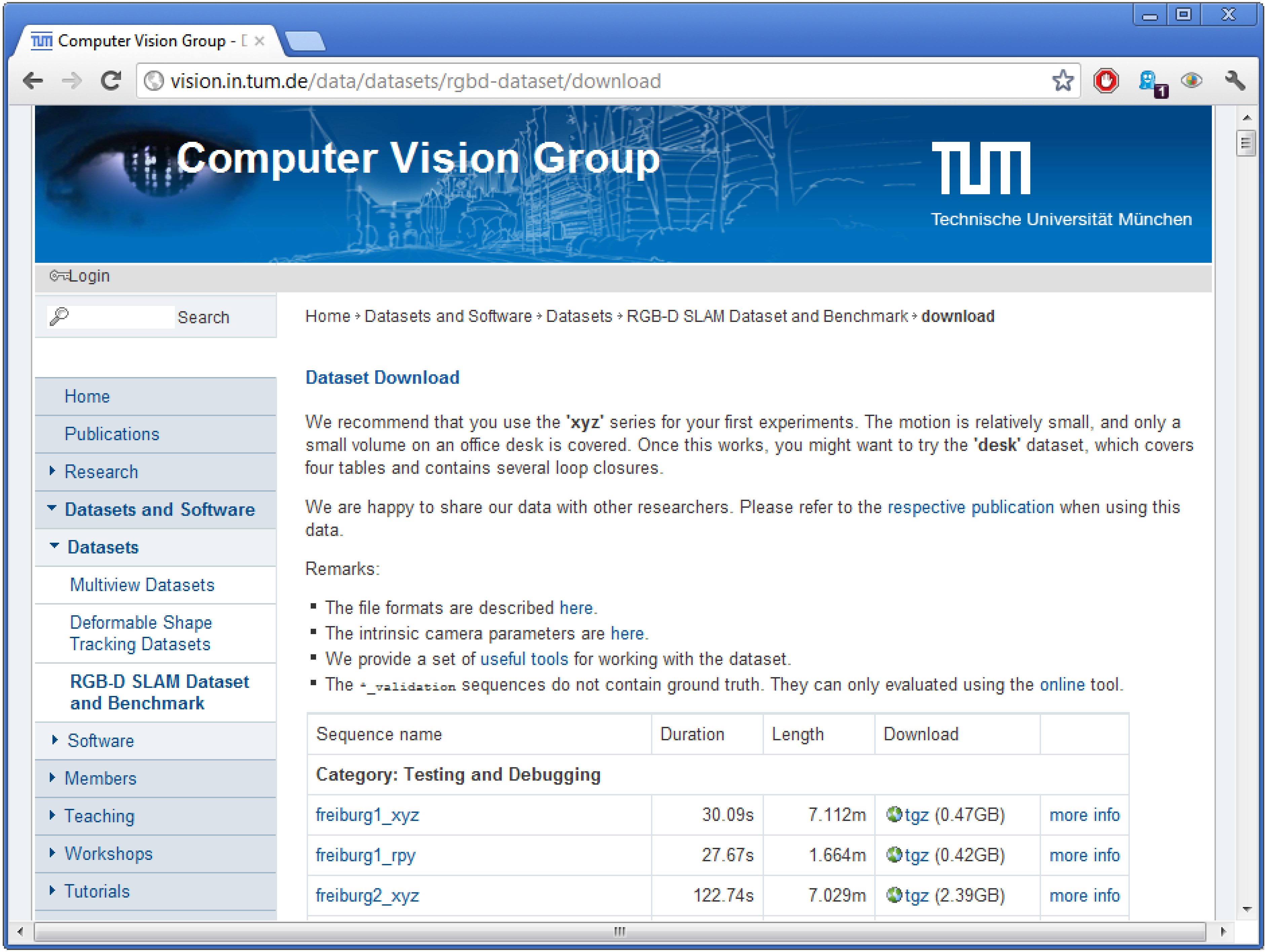


Calibration Step 3: Time Synchronization

- Assume a constant time delay between mocap and Kinect messages
- Choose time delay that minimizes reprojection error during checkerboard calibration




Computer Vision Group - [
vision.in.tum.de/data/datasets/rgbd-dataset
Computer Vision Group
TUM
Technische Universität München
Login
Search
Home › Datasets and Software › Datasets › RGB-D SLAM Dataset and Benchmark
Home
Publications
Research
Datasets and Software
Datasets
Multiview Datasets
Deformable Shape Tracking Datasets
RGB-D SLAM Dataset and Benchmark
Software
Members
Teaching
Workshops
Tutorials
RGB-D SLAM Dataset and Benchmark
Quick Links
Download page
File formats
Camera parameters
Usful tools and scripts
Contact: Jürgen Sturm
We provide a large dataset containing RGB-D data and ground-truth data with the goal to establish a novel benchmark for the evaluation of visual odometry and visual SLAM systems. Our dataset contains the color and depth images of a Microsoft Kinect sensor along the ground-truth trajectory of the sensor. The data was recorded at full frame rate (30 Hz) and sensor resolution (640×480). The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz). Further, we provide the accelerometer data from the Kinect. Finally, we propose an evaluation criterion for measuring the quality of the estimated camera trajectory of visual SLAM systems.
How can I use the RGB-D Benchmark to evaluate my SLAM system?
1. Download one or more of the RGB-D benchmark sequences (file formats, useful tools)
2. Run your favorite visual odometry/visual SLAM algorithm (for example, RGB-D SLAM)
3. Save the estimated camera trajectory to a file (file formats, example trajectory)
4. Evaluate your algorithm by comparing the estimated trajectory with the ground truth trajectory. We provide an automated evaluation tool to help you with the evaluation. There is also an online version of the tool.


Computer Vision Group - [
vision.in.tum.de/data/datasets/rgbd-dataset/download
Computer Vision Group
Technische Universität München
Login
Search
Home › Datasets and Software › Datasets › RGB-D SLAM Dataset and Benchmark › download
Home
Publications
Research
Datasets and Software
Datasets
Multiview Datasets
Deformable Shape Tracking Datasets
RGB-D SLAM Dataset and Benchmark
Software
Members
Teaching
Workshops
Tutorials
Dataset Download
We recommend that you use the 'xyz' series for your first experiments. The motion is relatively small, and only a small volume on an office desk is covered. Once this works, you might want to try the 'desk' dataset, which covers four tables and contains several loop closures.
We are happy to share our data with other researchers. Please refer to the respective publication when using this data.
Remarks:
The file formats are described here.
The intrinsic camera parameters are here.
We provide a set of useful tools for working with the dataset.
The *_validation sequences do not contain ground truth. They can only evaluated using the online tool.
Sequence name
Duration
Length
Download
Category: Testing and Debugging
freiburg1_xyz 30.09s 7.112m tgz (0.47GB) more info
freiburg1_rpy 27.67s 1.664m tgz (0.42GB) more info
freiburg2_xyz 122.74s 7.029m tgz (2.39GB) more info


Computer Vision Group - [
vision.in.tum.de/data/datasets/rgbd-dataset/download
Computer Vision Group
Technische Universität München
Login
Search
Home
Publications
Research
Datasets and Software
Datasets
Multiview Datasets
Deformable Shape Tracking Datasets
RGB-D SLAM Dataset and Benchmark
Software
Members
Teaching
Workshops
Tutorials
wnload
n is relatively small, and only a
the 'desk' dataset, which covers
tive publication when using this
ed using the online tool.
freiburg1_xyz: RGB movie
Click here for more information
Download this movie as avi
Related movies: RGB movie and, depth movie, external camera view
freiburg1_rpy 27.67s 1.664m tgz (0.42GB) more info
freiburg2_xyz 122.74s 7.029m tgz (2.39GB) more info
tgz (0.47GB) more info

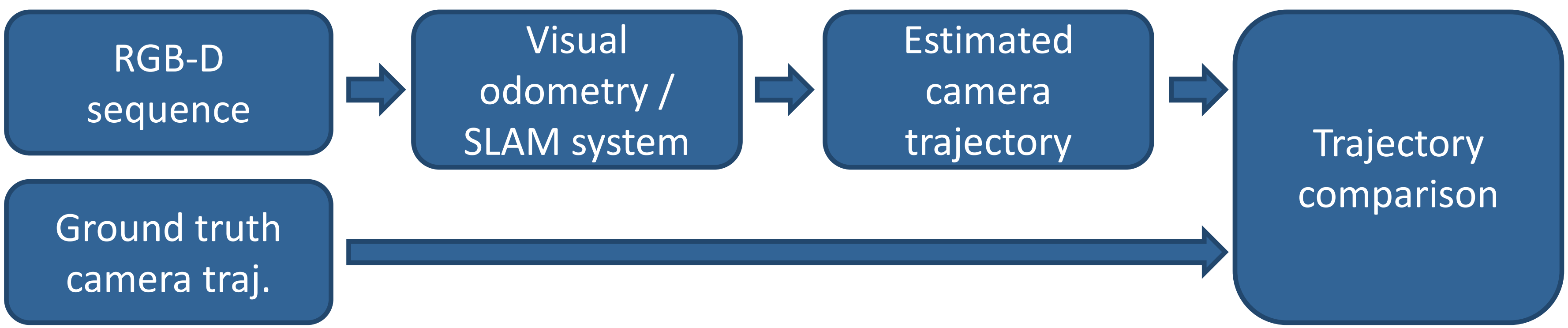
Dataset Website

- In total: 39 sequences (19 with ground truth)
- One ZIP archive per sequence, containing
  - Color and depth images (PNG)
  - Accelerometer data (timestamp ax ay az)
  - Trajectory file (timestamp tx ty ty qx qy qz qw)
- Sequences also available as ROS bag and MRPT rawlog

http://vision.in.tum.de/data/datasets/rgbd-dataset

What Is a Good Evaluation Metric?

- Compare camera trajectories
  - Ground truth trajectory $Q_1, \ldots, Q_n \in \mathrm{SE}(3)$
  - Estimate camera trajectory $P_1, \ldots, P_n \in \mathrm{SE}(3)$
- Two common evaluation metrics
  - Relative pose error (drift per second)
  - Absolute trajectory error (global consistency)

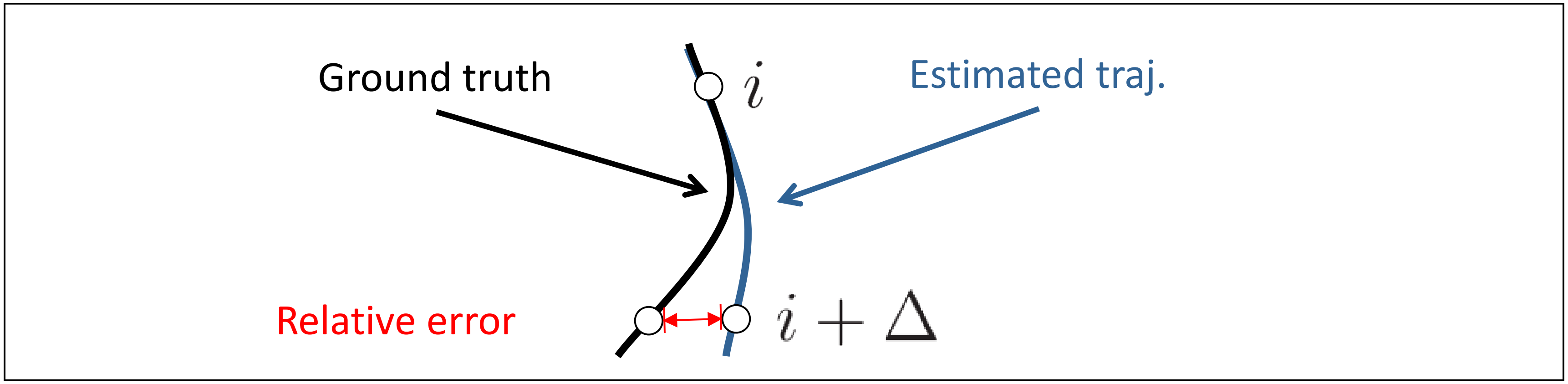


Relative Pose Error (RPE)

- Measures the (relative) **drift**
- Recommended for the evaluation of visual odometry approaches

$$E_i := \left(Q_i^{-1} Q_{i+\Delta}\right)^{-1} \left(P_i^{-1} P_{i+\Delta}\right)$$

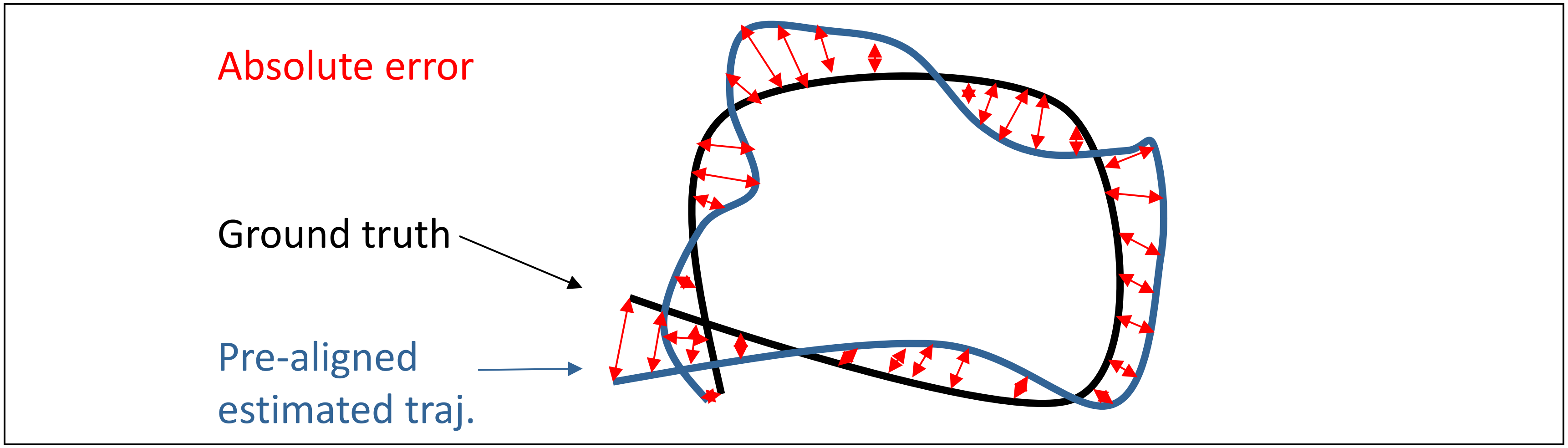
Relative error   True motion   Estimated motion



Absolute Trajectory Error (ATE)

- Measures the **global error**
- Requires pre-aligned trajectories
- Recommended for SLAM evaluation

$$E_i := Q_i^{-1} S P_i$$

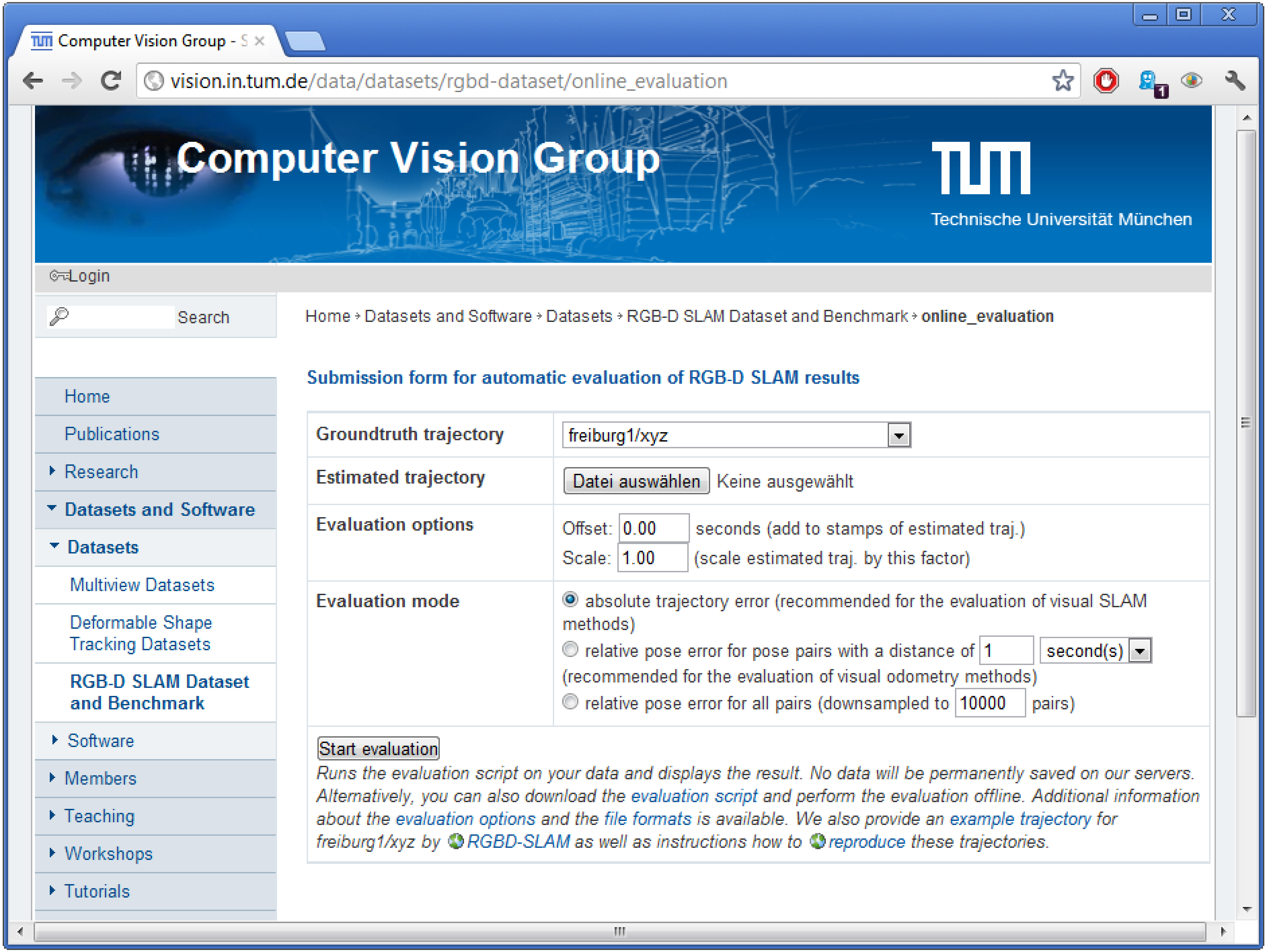


Evaluation metrics

- Average over all time steps

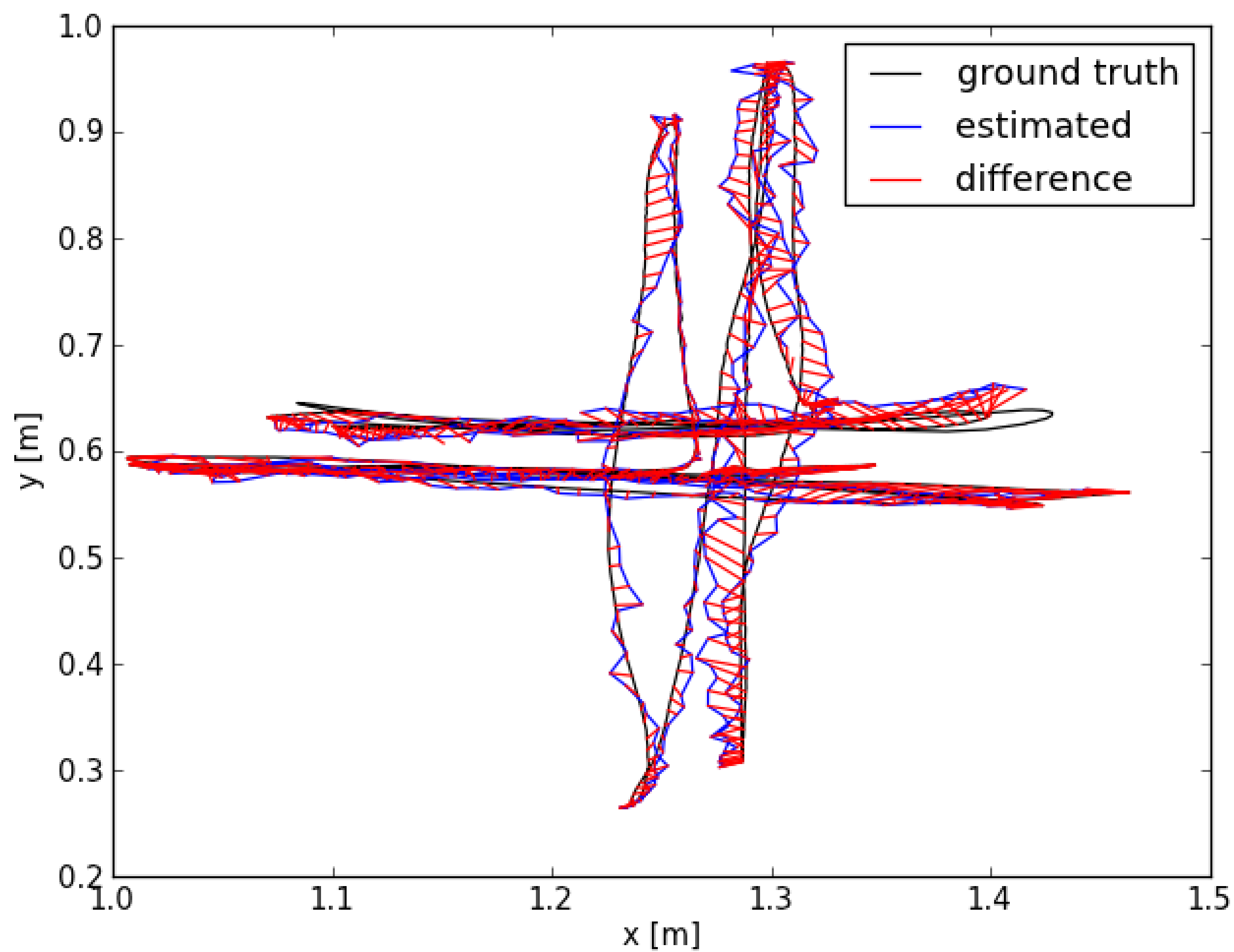
$$\mathrm{RMSE}(E_{1:n}) := \left(\frac{1}{m}\sum_{i=1}^{m} \| trans(E_i) \|^2\right)^{1/2}$$

- Reference implementations for both evaluation metrics available
- Output: RMSE, Mean, Median (as text)
- Plot (png/pdf, optional)


Computer Vision Group
vision.in.tum.de/data/datasets/rgbd-dataset/online_evaluation
Computer Vision Group
TUM
Technische Universität München
Login
Search
Home › Datasets and Software › Datasets › RGB-D SLAM Dataset and Benchmark › online_evaluation
Home
Publications
Research
Datasets and Software
Datasets
Multiview Datasets
Deformable Shape Tracking Datasets
RGB-D SLAM Dataset and Benchmark
Software
Members
Teaching
Workshops
Tutorials
Submission form for automatic evaluation of RGB-D SLAM results
Groundtruth trajectory
freiburg1/xyz
Estimated trajectory
Datei auswählen
Keine ausgewählt
Evaluation options
Offset: 0.00 seconds (add to stamps of estimated traj.)
Scale: 1.00 (scale estimated traj. by this factor)
Evaluation mode
absolute trajectory error (recommended for the evaluation of visual SLAM methods)
relative pose error for pose pairs with a distance of 1 second(s) (recommended for the evaluation of visual odometry methods)
relative pose error for all pairs (downsampled to 10000 pairs)
Start evaluation
Runs the evaluation script on your data and displays the result. No data will be permanently saved on our servers. Alternatively, you can also download the evaluation script and perform the evaluation offline. Additional information about the evaluation options and the file formats is available. We also provide an example trajectory for freiburg1/xyz by RGBD-SLAM as well as instructions how to reproduce these trajectories.

vision.in.tum.de/data/datasets/rgbd-dataset/online_evaluation

```
compared_pose_pairs 786 pairs
absolute_translational_error.rmse 0.013473 m
absolute_translational_error.mean 0.012029 m
absolute_translational_error.median 0.011176 m
absolute_translational_error.std 0.006068 m
absolute_translational_error.min 0.000939 m
absolute_translational_error.max 0.034727 m
```



Summary – TUM RGB-D Benchmark

- Dataset for the evaluation of RGB-D SLAM systems
- Ground-truth camera poses
- Evaluation metrics + tools available

Discussion on Benchmarks

Pro:

- Provide objective measure
- Simplify empirical evaluation
- Stimulate comparison

Con:

- Introduce bias towards approaches that perform well on the benchmark (overfitting)
- Evaluation metrics are not unique (many alternative metrics exist, choice is subjective)

Lessons Learned Today

- How to generate plans that are robust to uncertainty in sensing and locomotion
- How to explore an unknown environment
  - With a single robot
  - With a team of robots
- How to generate plans that fully cover known environments
- How to benchmark SLAM algorithms