Introduction to Mobile Robotics

Improved Multi-Robot Exploration

Wolfram Burgard
Cyrill Stachniss
Giorgio Grisetti
Maren Bennewitz
Christian Plagemann

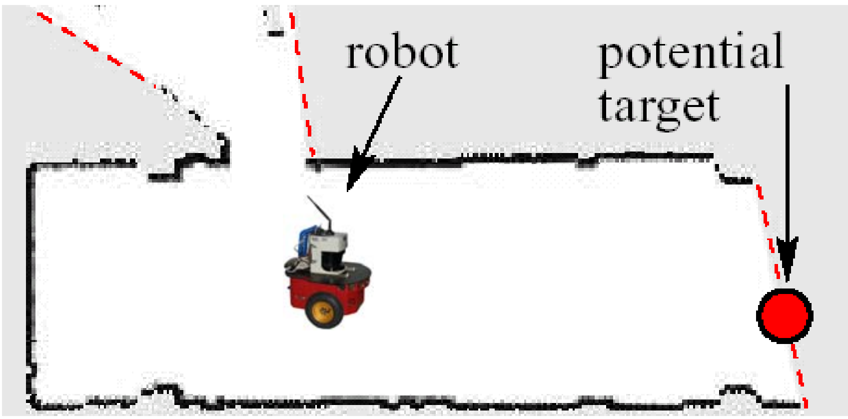
Problem

- A team of robots has to explore an initially unknown environment by sensor coverage
- Find an assignments of target locations to robots that minimizes overall exploration time

- Variants
  - Centralized coordination with communication
  - Background knowledge
  - Limited communication (range, bandwidth)

Centralized Coordination for Multi-Robot Exploration

- Robots share a common map
- The frontiers between free space and unknown areas are potential target locations



- Find a good assignment of frontier locations to robots to minimize overall exploration time

Centralized Coordination for Multi-Robot Exploration

1. Determine the frontiers (targets)

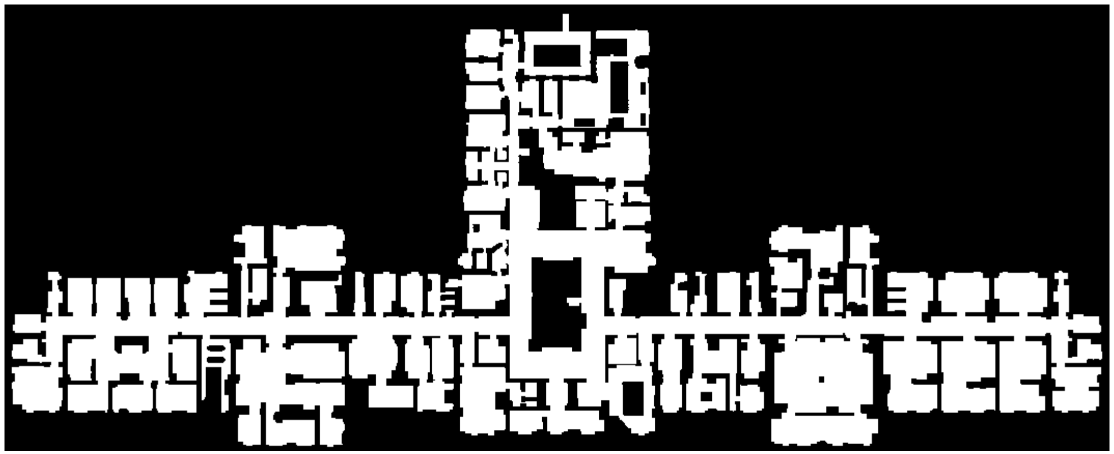
2. Compute the travel cost $V(i,t)$ of each robot i to each target location t

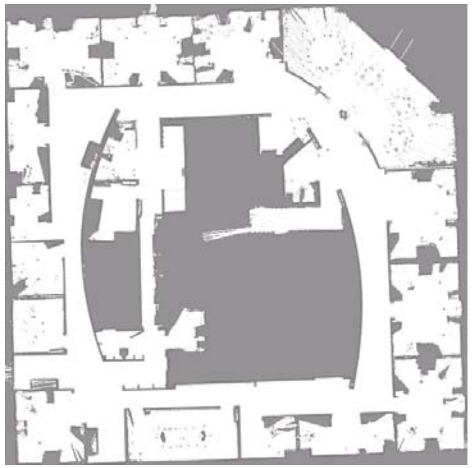
3. Assign an initial Utility $U(t) = 1$ to each frontier

4. While a robot exists that has no target assigned
   - Choose $(i^*,t^*) = \underset{(i,t)}{\operatorname{argmax}}\ U(t) - V(i,t)$
   - $U(t') = U(t') - P_{visible}(t^*, t')$

Using Background Knowledge

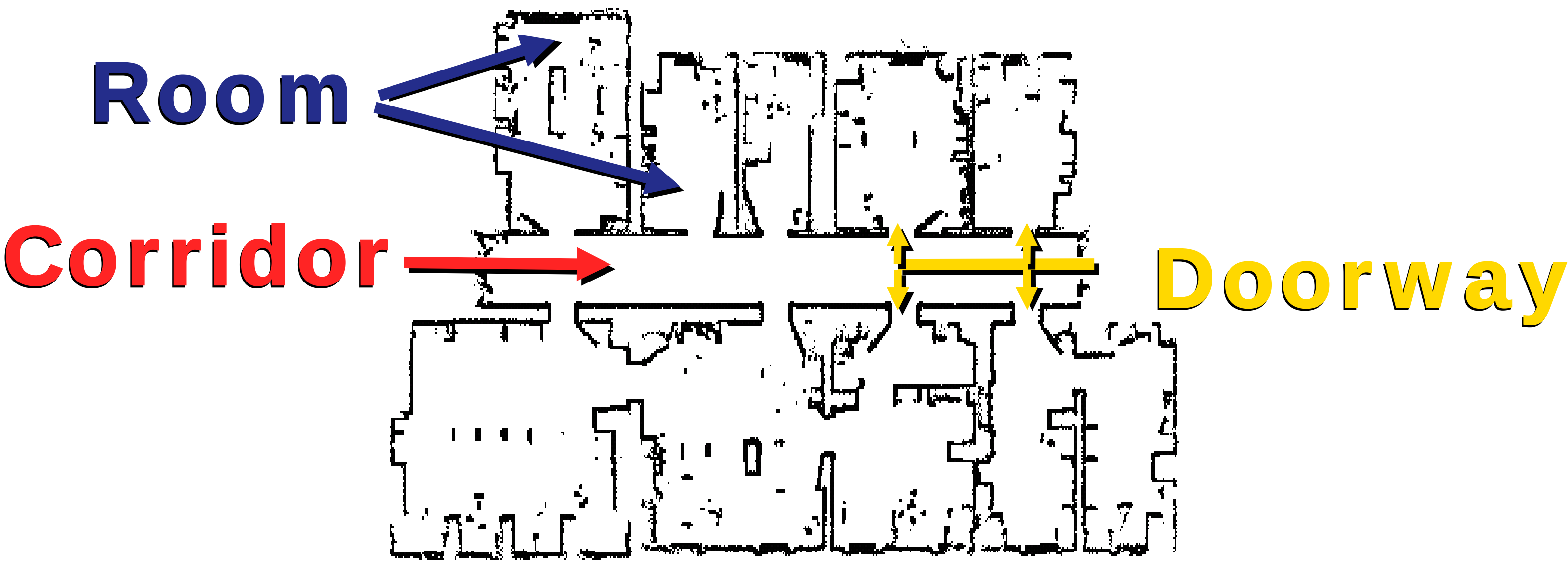
- Corridors typically provide multiple transitions to rooms (potentially unobserved places) compared to rooms
- The more transitions to unknown regions a place has, the better the robots can distribute themselves over the environment
- Can we extract such information and use it to improve the distribution of the robots?



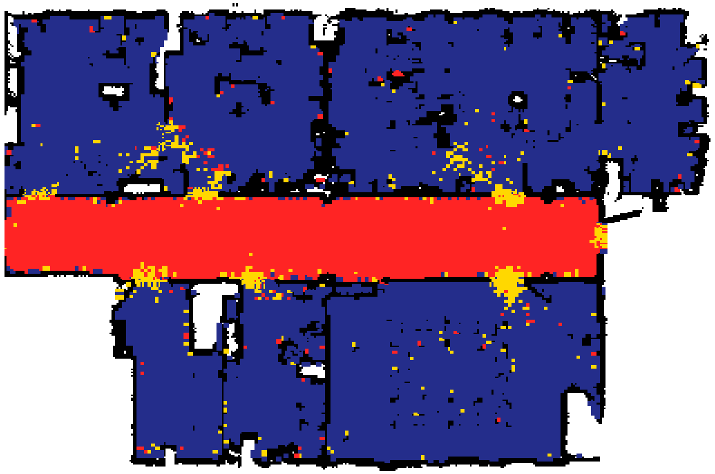


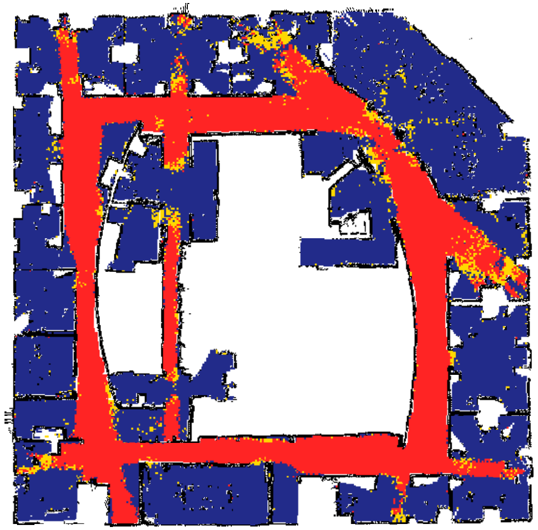
Using Information about Places

- Given we know in which the type of place the robot is in, we can do better assignments
- Techniques to classify places exist (see Chapter on AdaBoost)



Application to Exploration





- Assign a higher reward to regions that probably provide more frontiers to unknown areas.
- **Advantage:** Nearly no changes to the assignment algorithm are required.

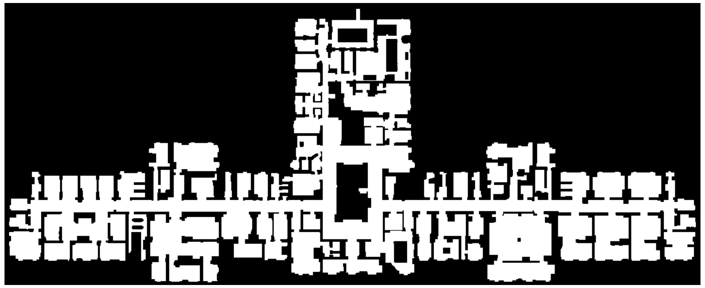
Improved Coordination for Multi-Robot Exploration

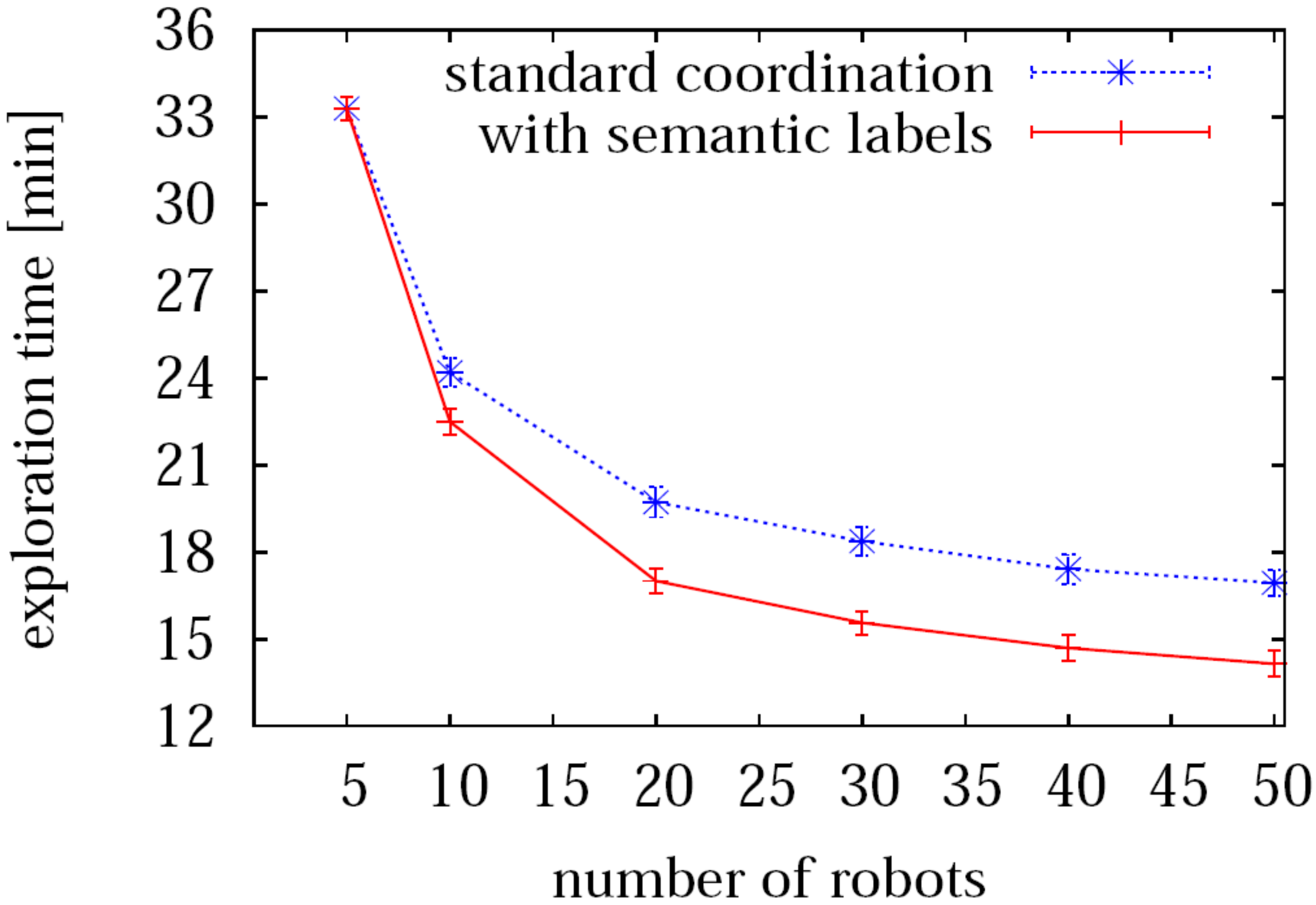
1. Determine the frontiers (targets)
2. Compute the travel cost $V(i,t)$ of each robot i to a target t
3. Assign a utility:

$$U(t) = c > 1 \quad \text{if } t \text{ is a corridor location}$$
$$U(t) = 1 \quad \text{if } t \text{ is not a corridor location}$$

4. While a robot exists that has no target assigned
   - Choose $(i^*,t^*) = \operatorname*{argmax}_{(i,t)} U(t) - V(i,t)$
   - $U(t') = U(t') - P_{visible}(t^*, t')$

Evaluation





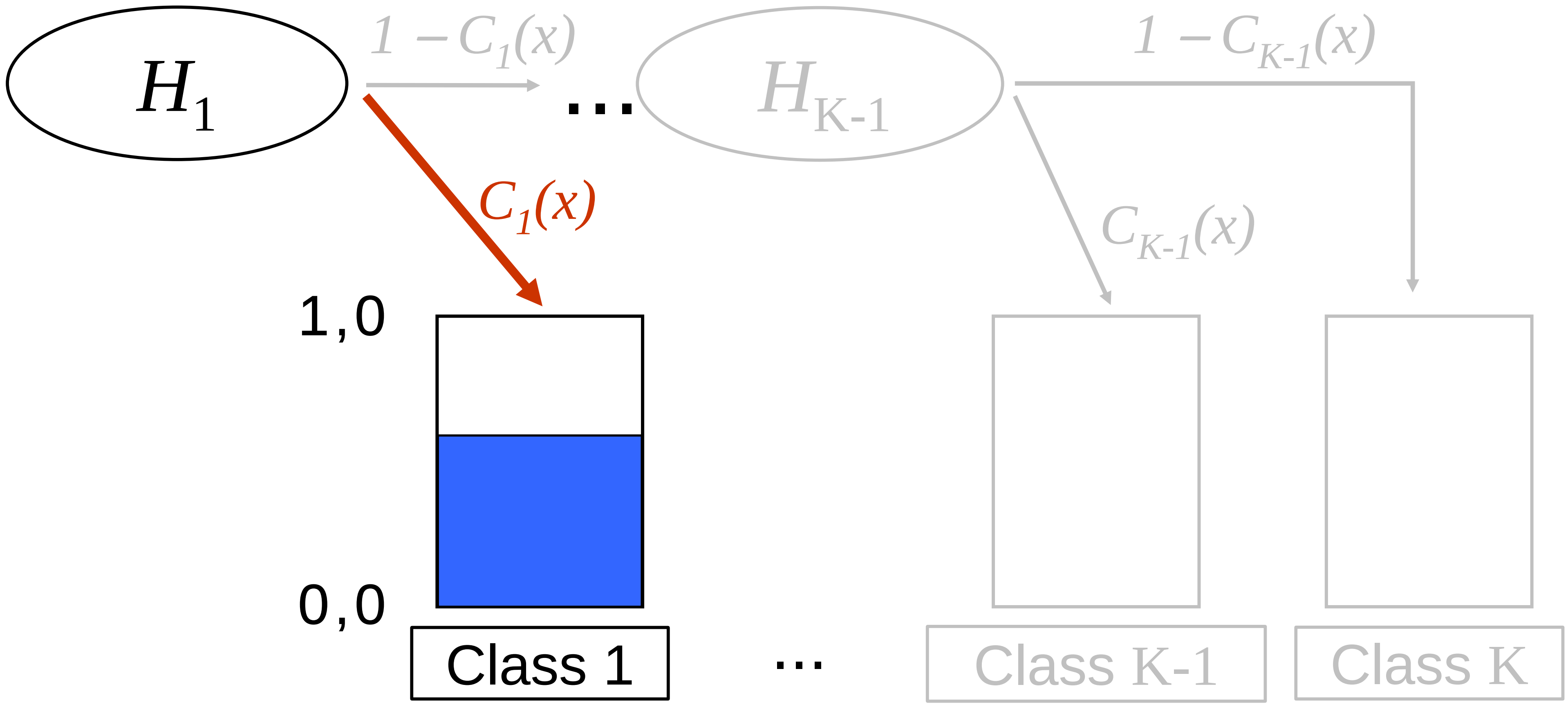
- Considering semantic place information leads to a significantly reduced exploration time

Noisy Place Classification

- In reality, the classification is affected by noise.
- If the robot classifies a location at which it has never been, is has to guess the corresponding observations.
- An extension to AdaBoost allows us to compute confidence values for the classifiers.

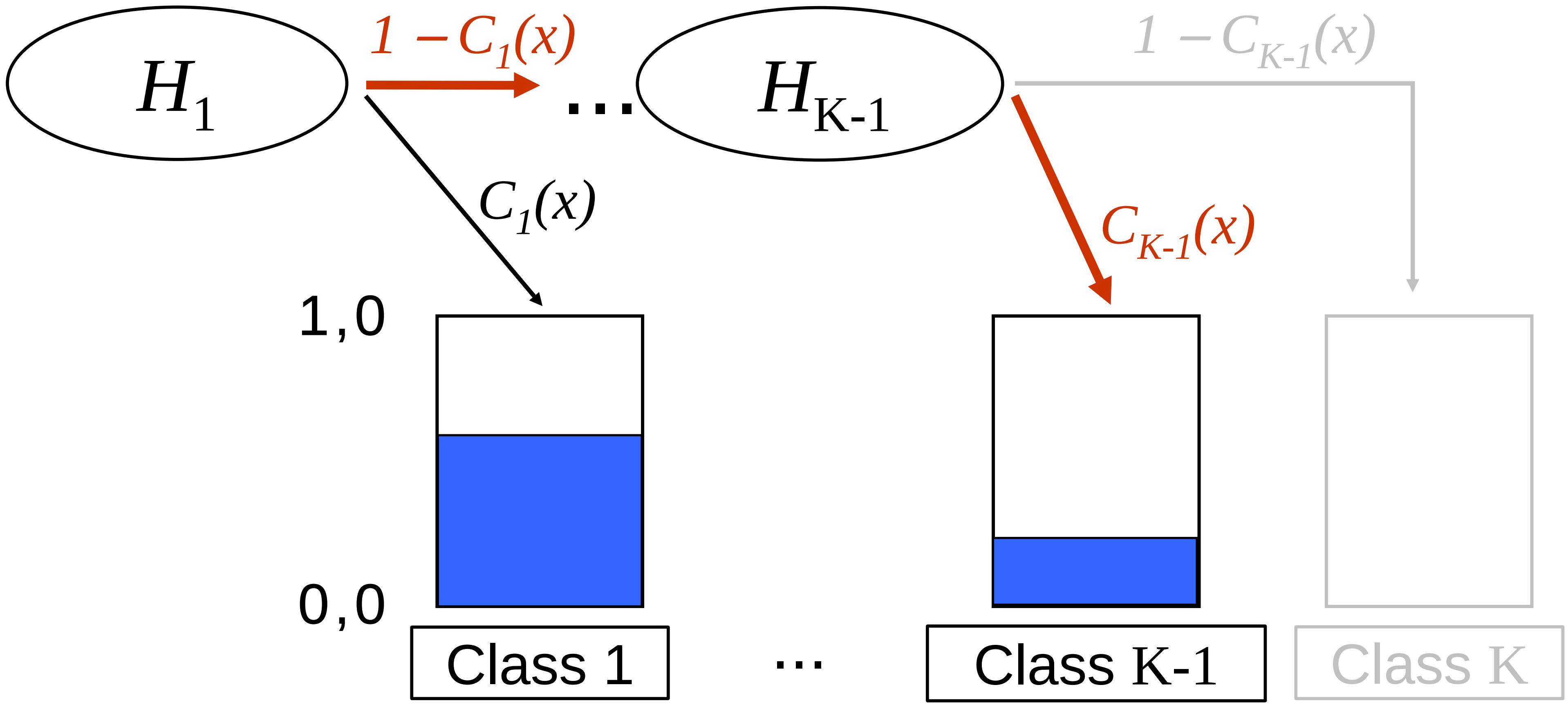
Classification Confidence

- Compute a confidence $C_k(x)$ value for each classification [Friedmann et al. 2000]



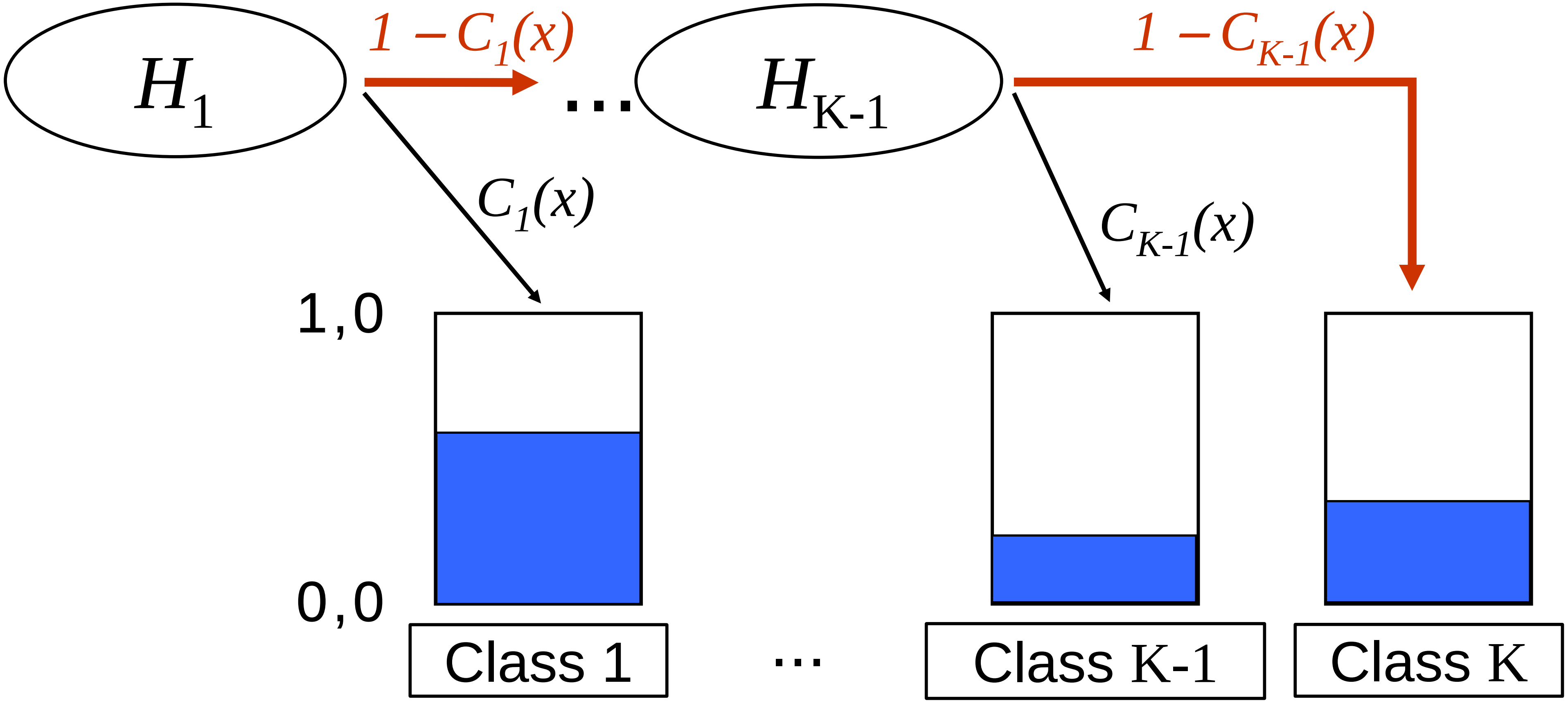
Classification Confidence

- Compute a confidence $C_k(x)$ value for each classification [Friedmann et al. 2000]

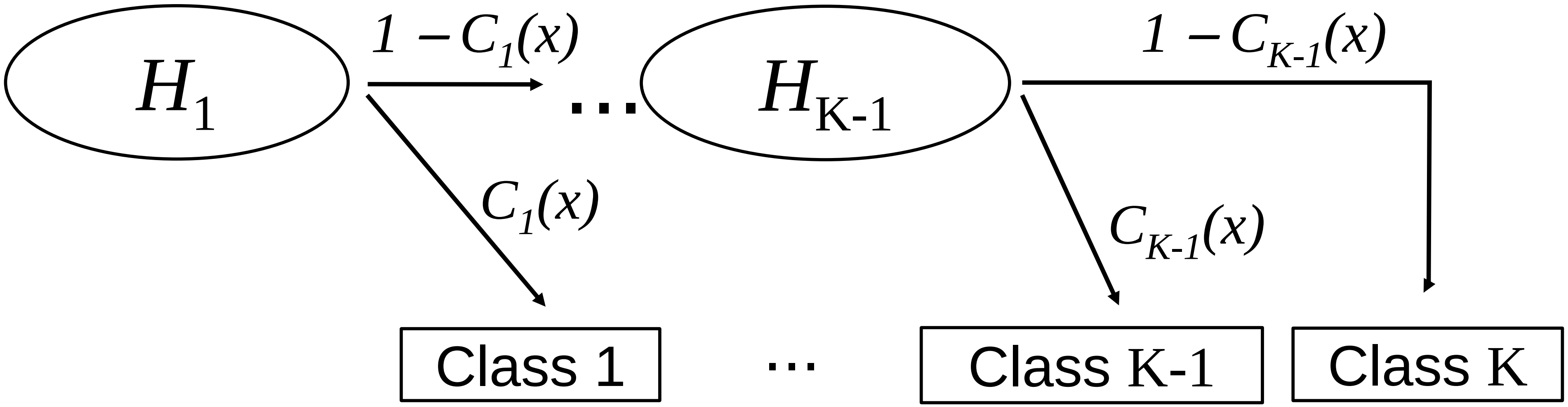


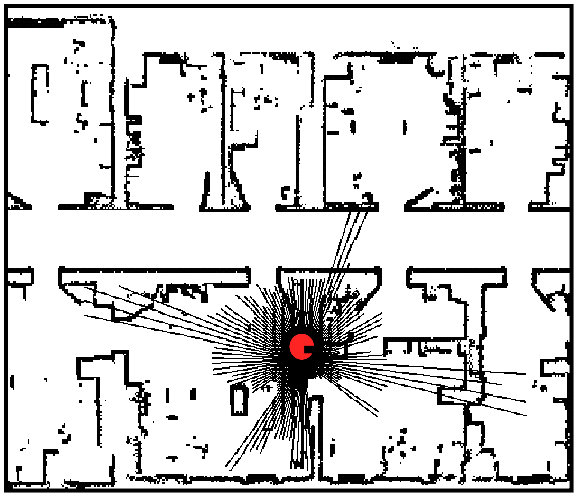
Classification Confidence

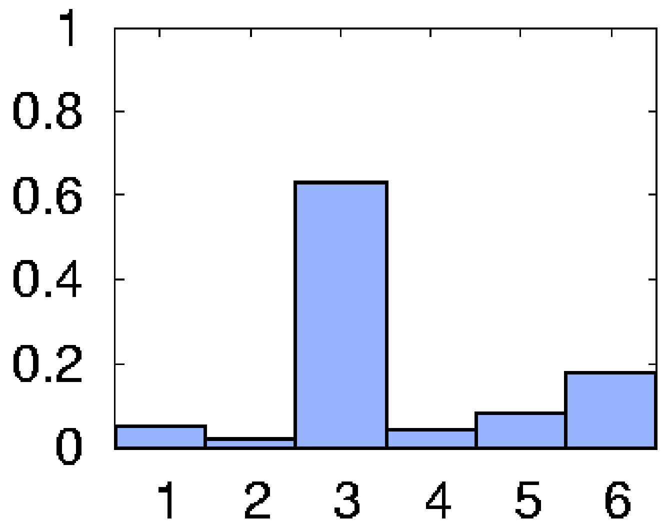
- Compute a confidence $C_k(x)$ value for each classification [Friedmann et al. 2000]



Posterior about Place Labels





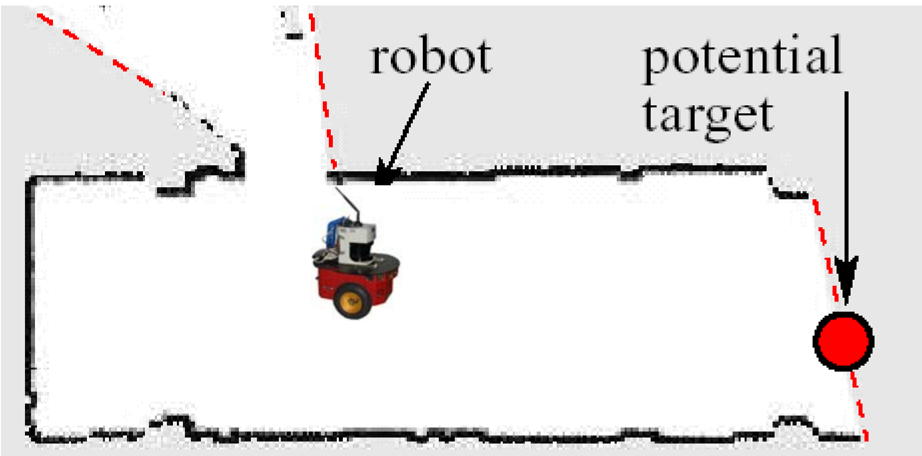


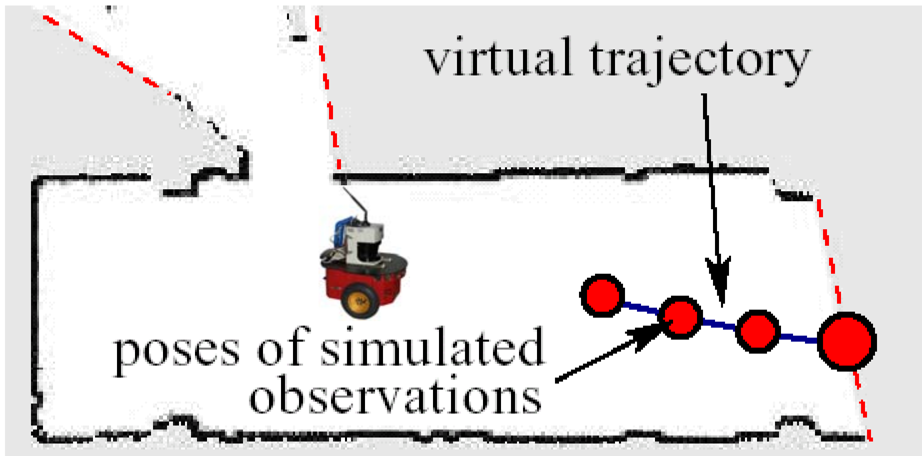
Improving Place Classification

- Measure the confidence of the decisions of the individual classifiers
- So far, the classification uses only one location
- We can improve the classification using information about past classifications
- We can use a Hidden Markov Models (HMM) to model potential transitions between places

Classifying a Target Location

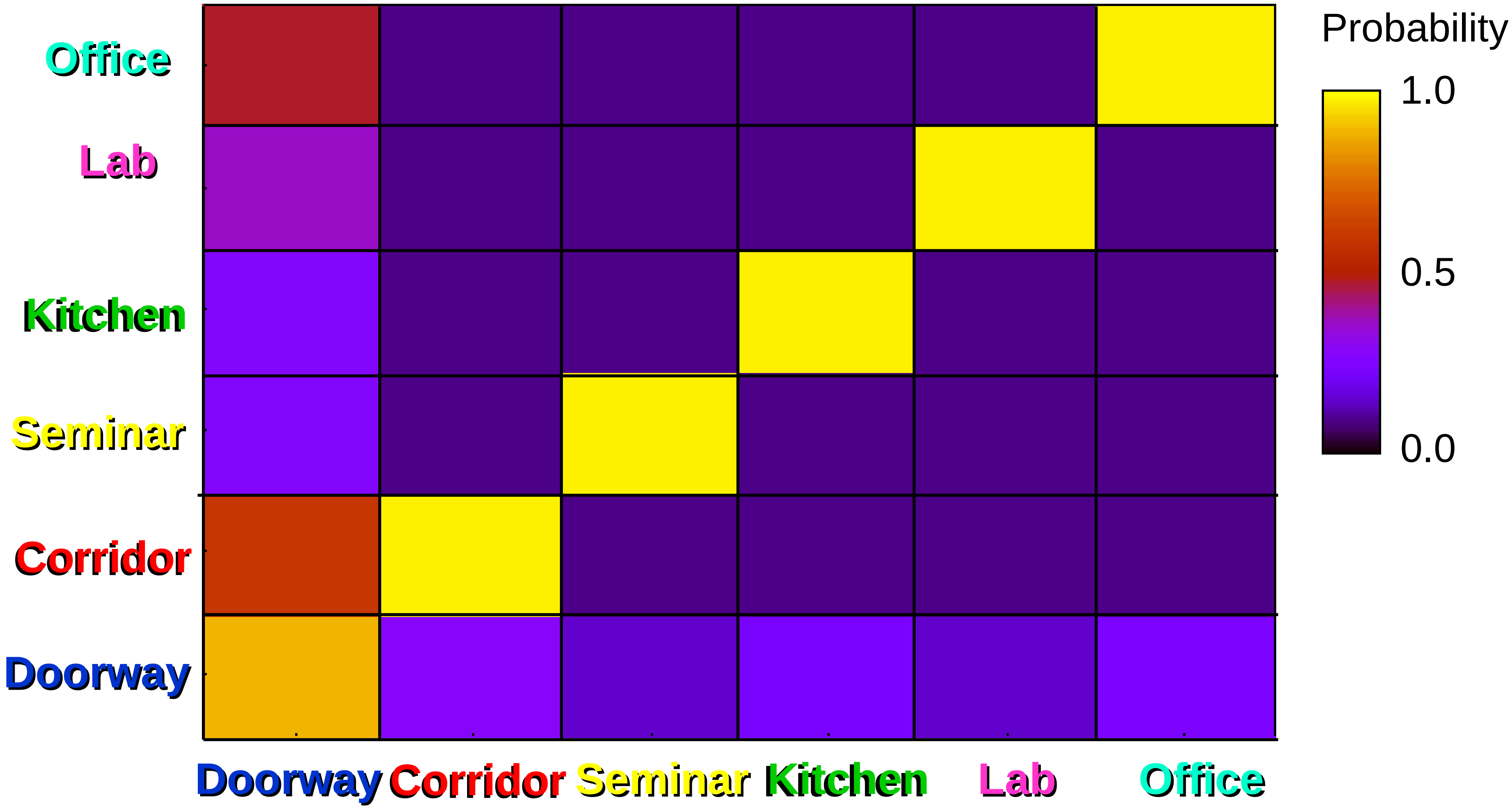
- Since the surroundings of a frontier location are partially unexplored, the classification of those places is quite noisy
- Apply HMM-filtering based on a virtual trajectory





Transition Matrix

- Dependency between subsequent positions



Hidden Markov Model

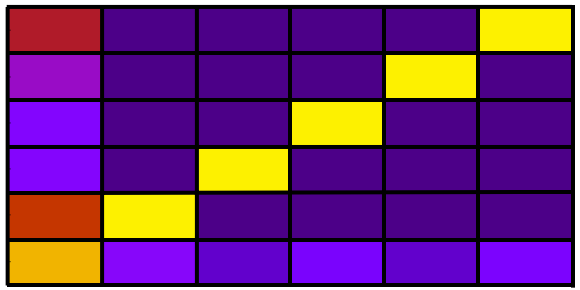
Place label

$$Bel(\xi_t) = \alpha P(z_t|\xi_t) \sum_{\xi_{t-1}} P(\xi_t|\xi_{t-1}, u_{t-1}) Bel(\xi_{t-1})$$

Normalizer

Observation likelihood

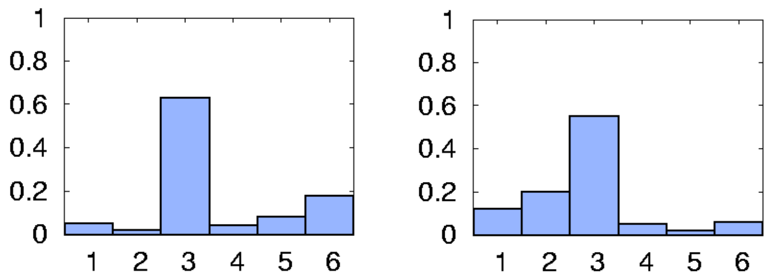
Transition model



Likelihood of the observation, given to robot is at a certain type of place

Observation Likelihood

- How to determine $P(z_t|\xi_t)$?
- Our observation is a distribution (histogram)



- One way to compare distributions is the KL-divergence

KL-Divergence

- A measure for the difference between two distributions
- For the discrete case it is defined as

$$KLD(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

- The smaller the value the closer the distributions
- The KLD is not symmetric. A symmetric variant is

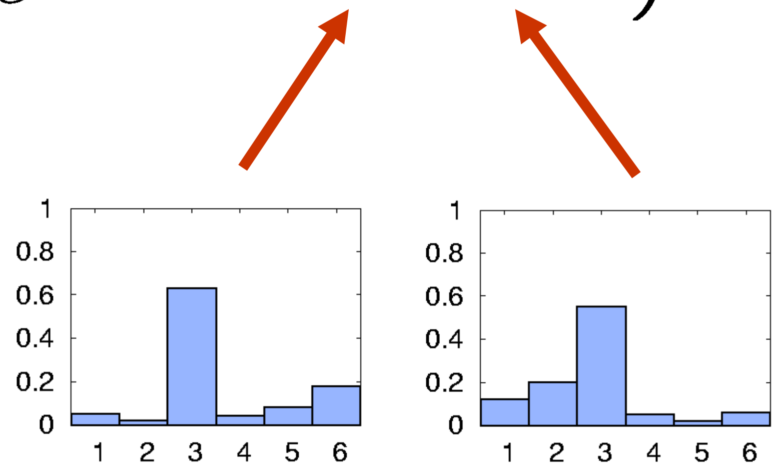
$$KLD_{sym}(P||Q) = KLD(P||Q) + KLD(Q||P)$$

Observation Likelihood

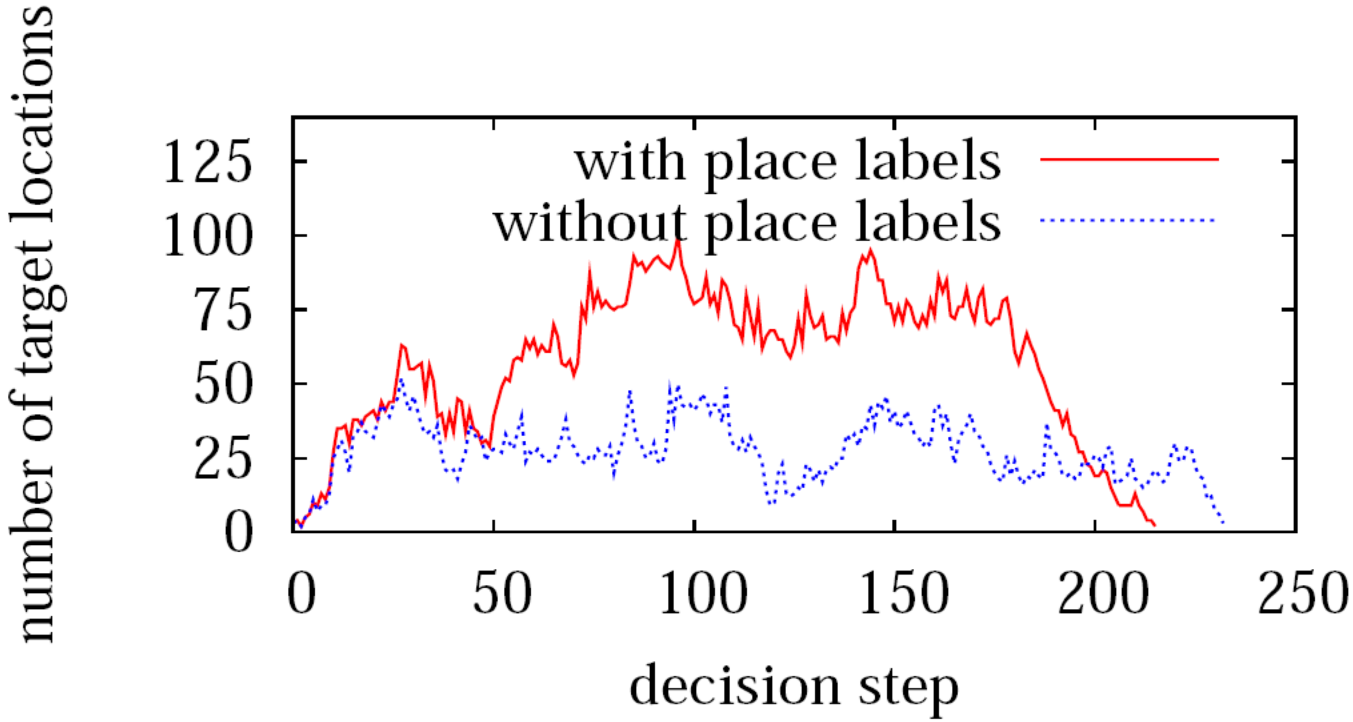
- Record a set of observations for each class
- Compute the average histogram for each class

- Compare the learned histogram with the current observation using the KL-divergence:

$$P(z_t|\xi_t) = e^{-KLD(z_t, z_{learned})}$$

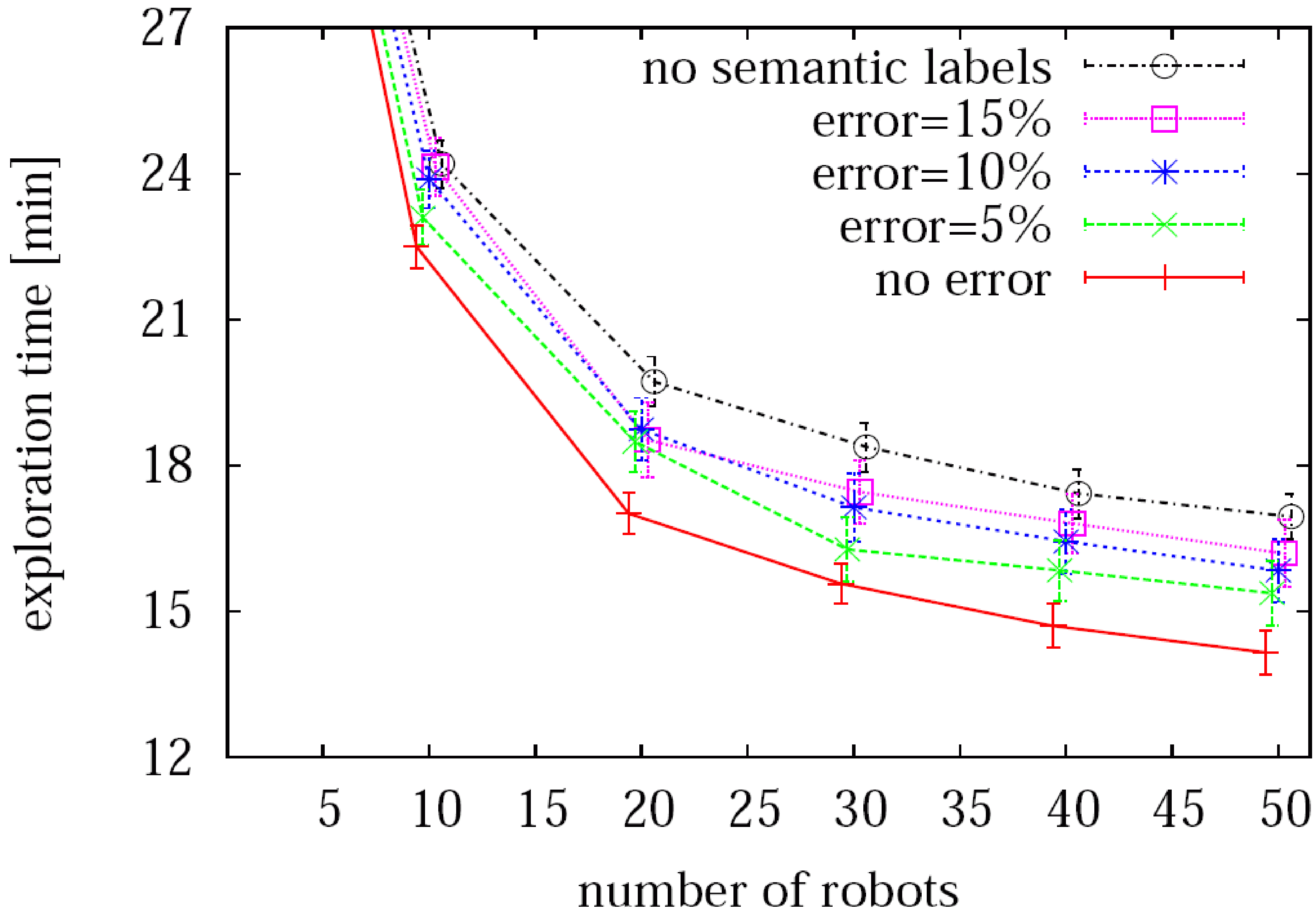


Number of Target Locations

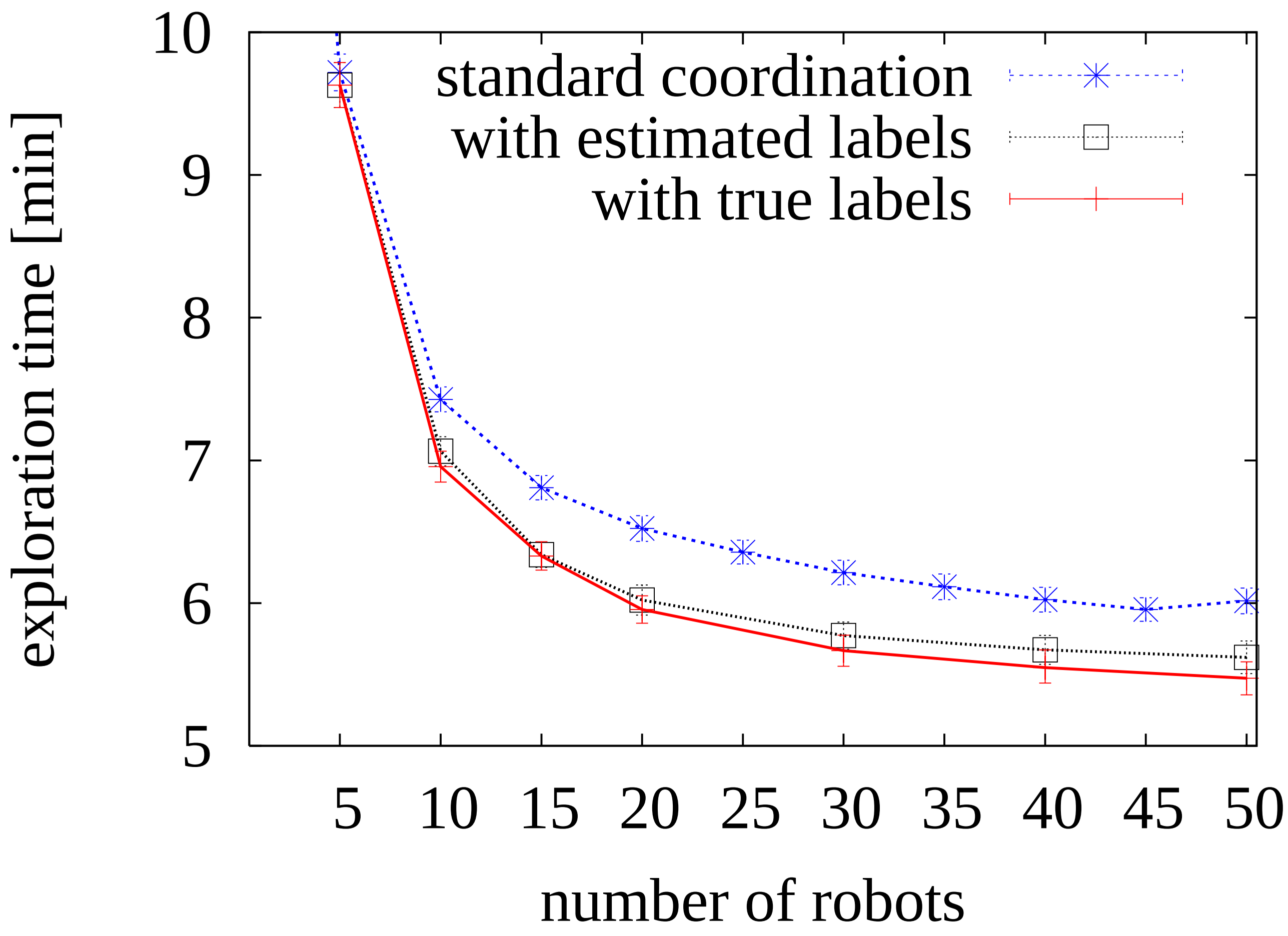


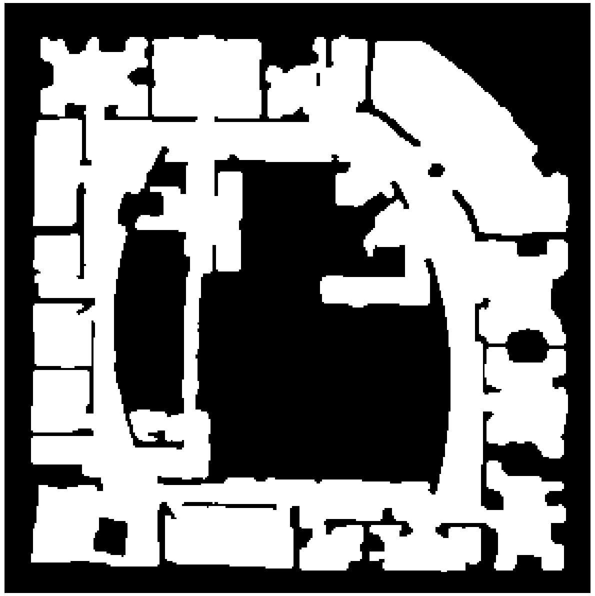
- By focusing the exploration on corridors, more potential target locations are available
- This often results in a better distribution of robots over the environment
- Up to ~20% less interference between robots

Influence of the Estimation Error



Results with Estimated Labels





Conclusions

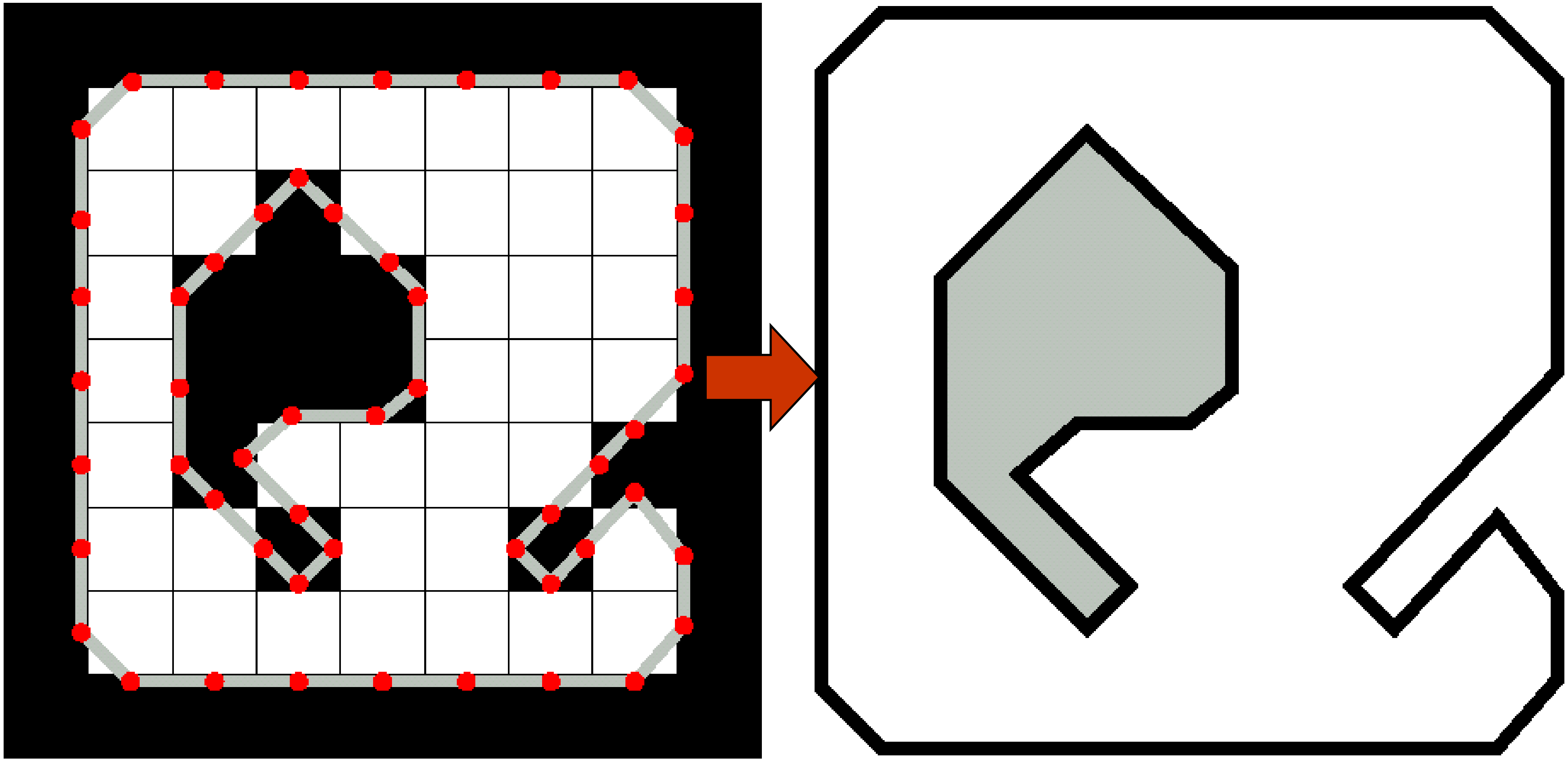
- **Centralized approach** allows the robots to **efficiently coordinate** their actions
- By utilizing the information about the type of the place, one obtains a significant **increase in performance** for multiple exploring robots

Limited Communication

- Communication is only possible when the robots stay close together:
  - Distributed information storage
  - Centralized ➡ decentralized coordination
- Communication link has a serious bandwidth limitations:
  - Which information to transmit?
  - How to compactly represent the model of the world?

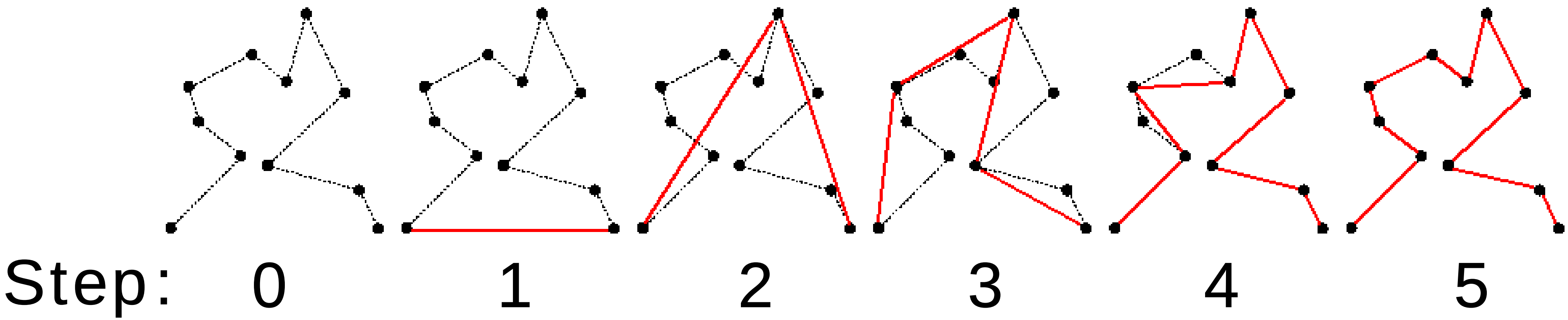
Deriving Polygonal Models

- Compact model based on a set of polygons
- Polygons are modeled by occupied, free, and unknown line segments

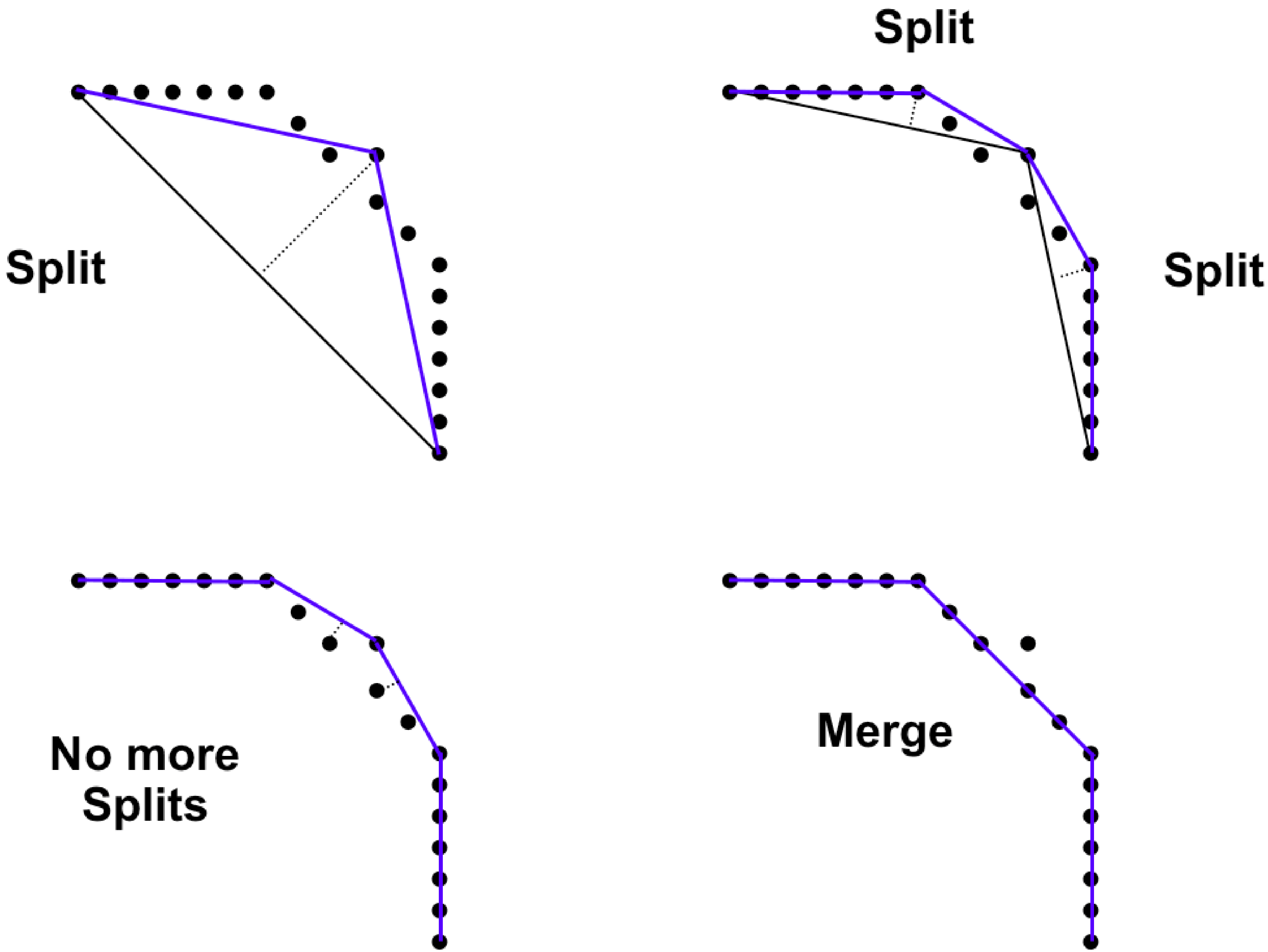


Incremental Update

- Each robot maintains a polygonal approximation of its field of view
- How to obtain a polygonal representation about the environment from laser data?
- One possible solution is the Douglas-Peucker Algorithm



Douglas-Peucker Algorithm: “Split and Merge”



Picture by J. Tardos

Douglas-Peucker Algorithm: "Split and Merge"

Algorithm

Split

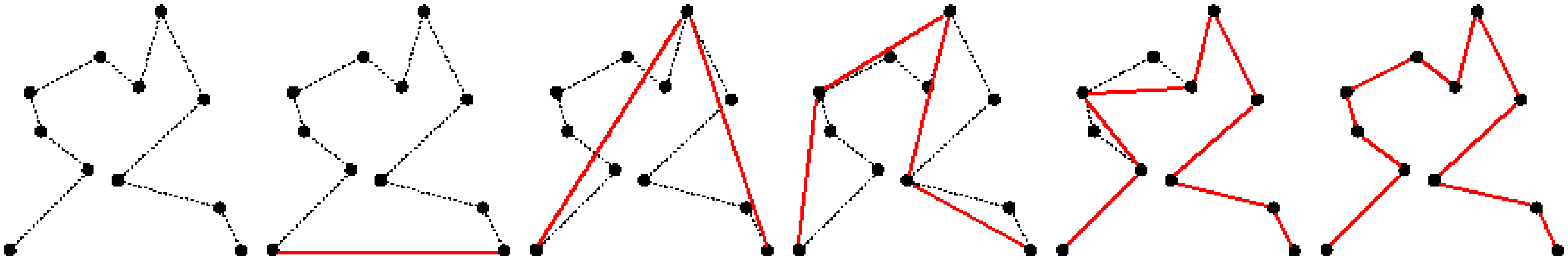
- Obtain the line passing by the two extreme points
- Find the most distant point to the line
- If distance > threshold, split and repeat with the left and right point sets

Merge

- If two consecutive segments are close/collinear enough, obtain the common line and find the most distant point
- If distance <= threshold, merge both segments

Incremental Update

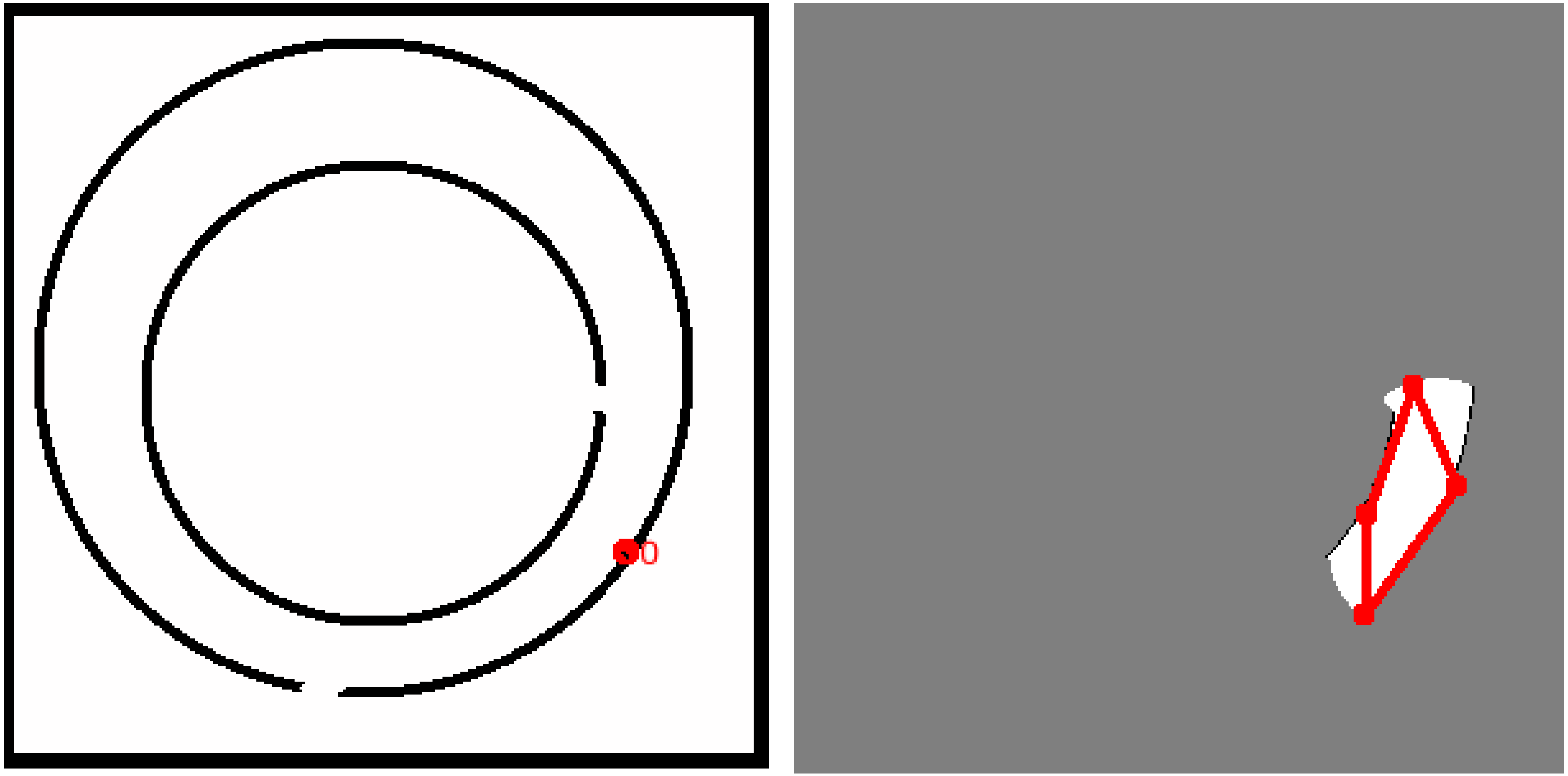
- Douglas-Peucker Algorithm is used to recursively refine the approximation



Step: 0 1 2 3 4 5

- Incremental updates are computed using the minimum edit distance

Video: Polygonal Approximation

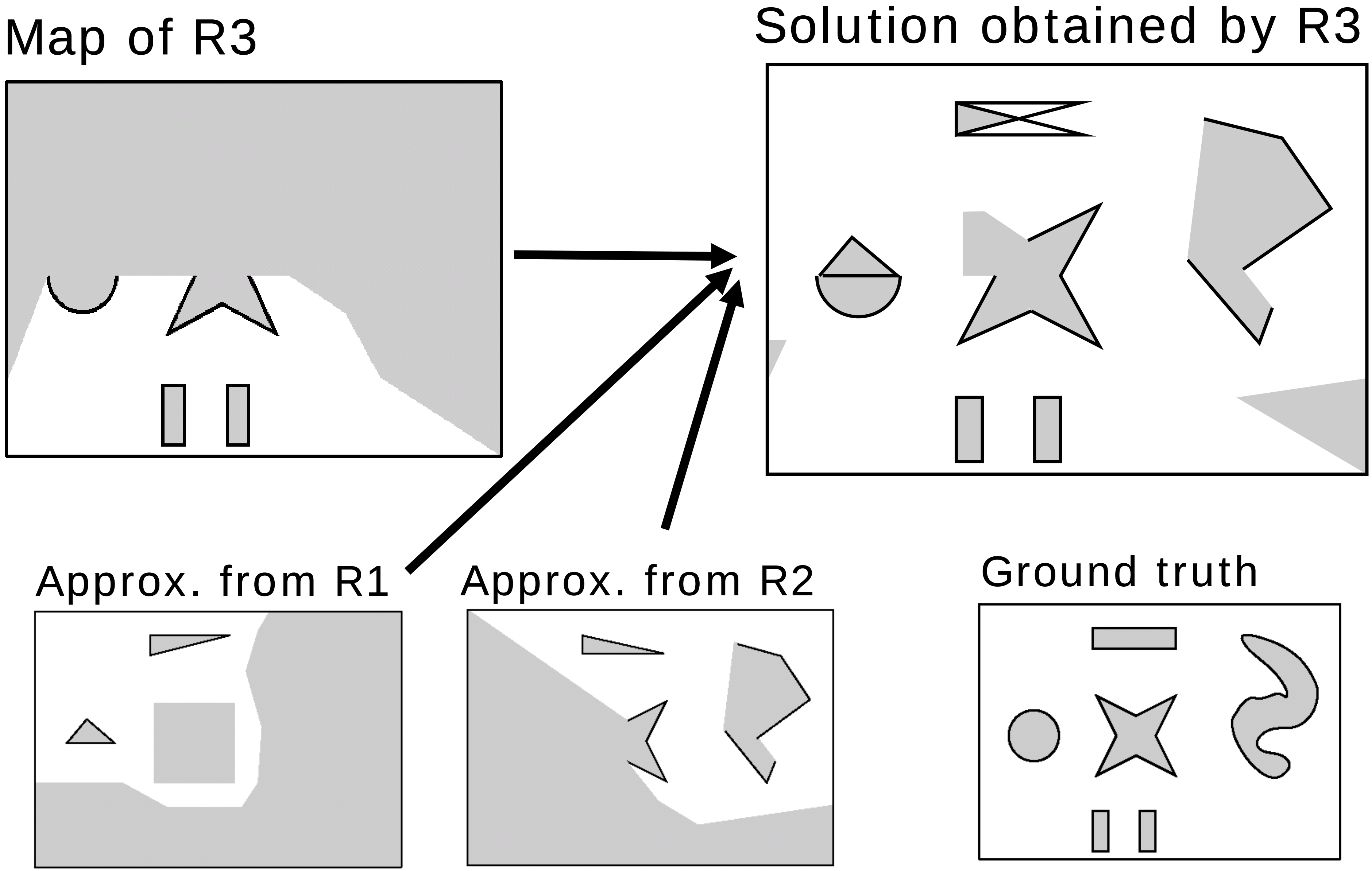


A robot building and maintaining a polygonal model of his environment

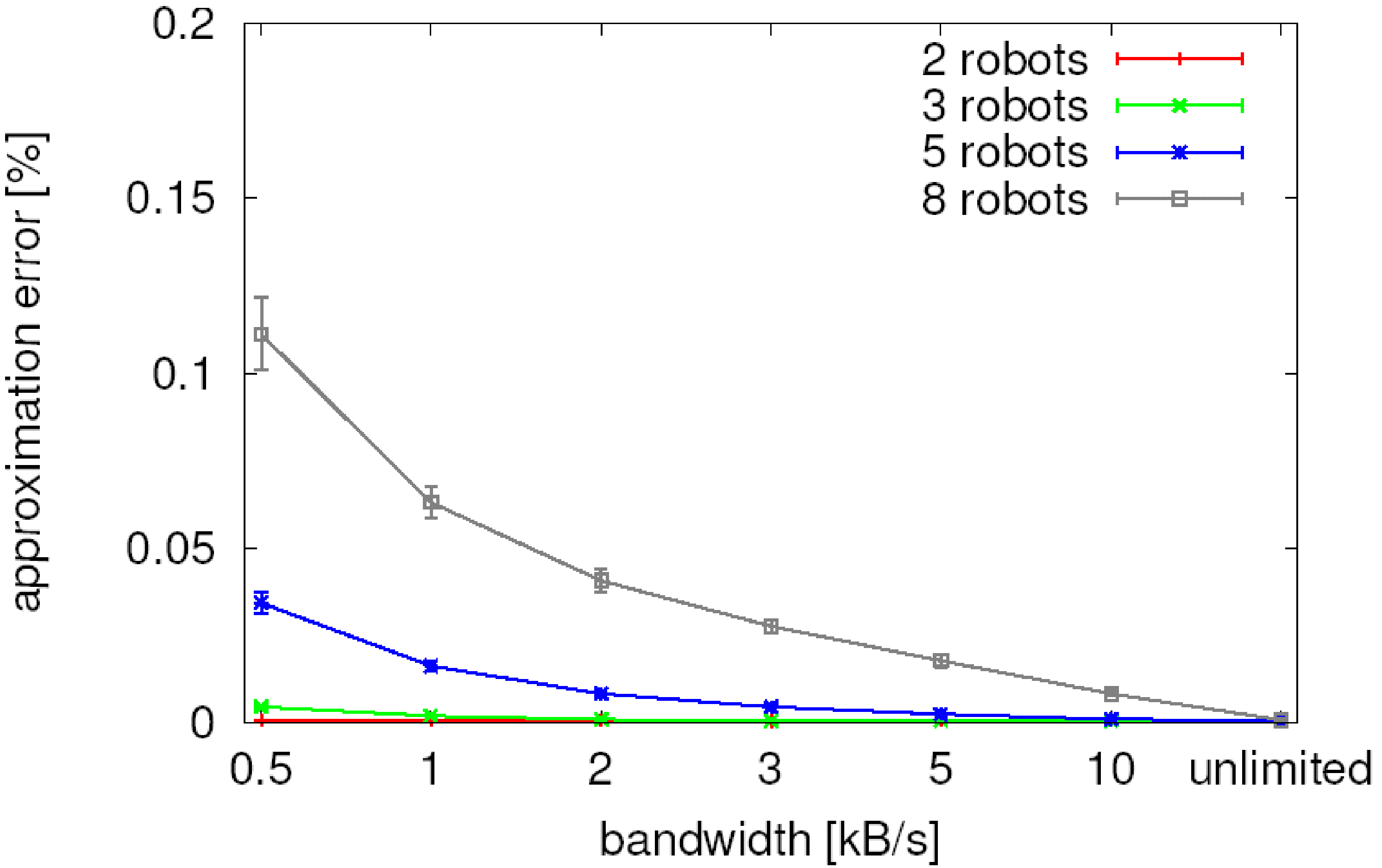
Merging Environmental Models

- Each robot integrates the knowledge obtained by its team-mates into a map
- Polygonal approximations are used to update the unobserved areas of the robot itself
- In case of ambiguities, the priority: occupied, free, unknown is used

Merging Maps



Approx. Error vs. Bandwidth



Even for low bandwidth networks, the error of the polygonal model is small

Bandwidth Usage

- Transmitting raw laser data leads to a significantly higher network traffic compared to the usage of polygonal approximations:
  - About 1 order of magnitude for maximum detail level of the polygonal model
  - About 2 orders of magnitude for medium and poor detail levels of the polygonal model

Polygonal representations clearly outperform classic representations regarding bandwidth requirements

Conclusions

- In case of **limited communication**, **compressed models** of the environment are needed
- Polygonal representations clearly outperform classic representations regarding bandwidth requirements