Technical Overview of Robotic Mapping

Kaiserslautern

Robotics Research Lab

Department of Computer Science

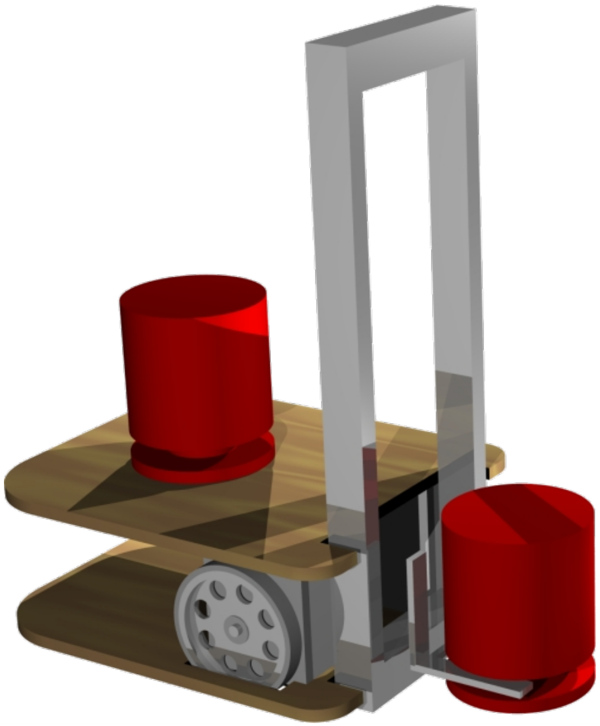
University of Kaiserslautern

Jonas Mitschang, 30.6.2007

RRLAB
THE ROBOTICS RESEARCH LAB

Contents

- Introduction
  - Motivation
  - Historical Overview
- Challenge in Robotic Mapping
- Types of Maps
  - Metrical Maps
  - Topological Maps
  - Metrical-Topological Hybrids
- Conclusion



Introduction: Motivation

- Solve problems / Navigation in environment
- Sensors
- Idea of storing knowledge
  - ➔ Build map autonomous, in real-time
- Highly active area in robotics and artificial intelligence
  - – Many achievements in last two decades
  - – Still many problems to solve

Introduction: Historical Overview

- Mapping since 1970s
- Classification:
  - Metrical and Topological
    - Today: smooth transition
    - Different advantages and disadvantages
    - Today: Hybrids
  - Robot-Centric and World-Centric
    - Robot-Centric: Simpler, no transformation
    - World-Centric: More abstracted, global map
    - Today: World-Centric
- Since 1990s: probabilistic approaches
- Simultaneous Localization and Mapping (SLAM)

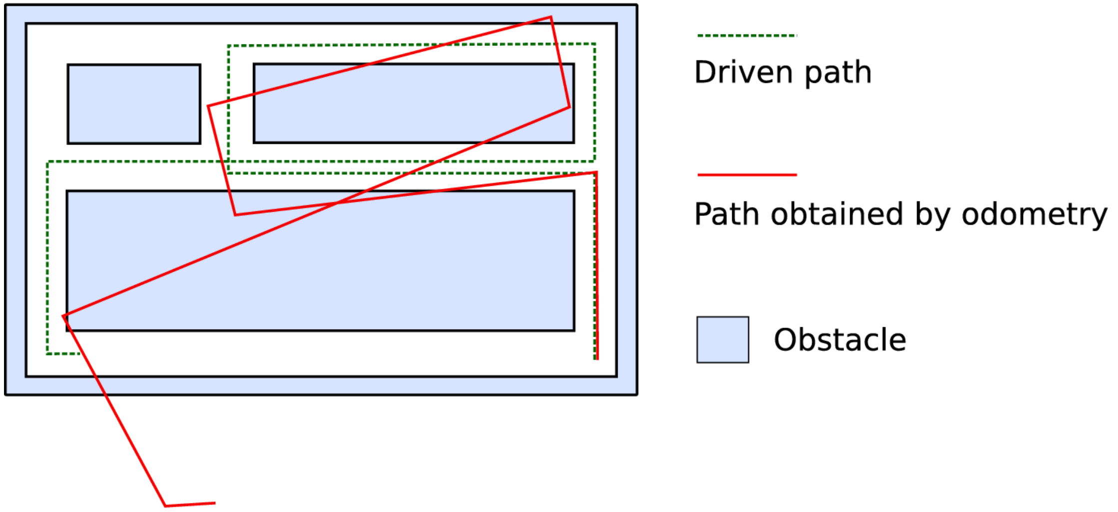
Challenge in Robotic Mapping

- Chicken and Egg Problem:
  - Construct reliable map with given pose
  - Determining robot's pose in a given map
- High Dimension
  - Limited computational power
  - Dimension depends on
    - Number of objects on map
    - Type of map (metrical / topological)
  - Unlimited complex (e.g. graphical 3D-Maps)

Challenge in Robotic Mapping

Sensors

- Ambiguous sensor data (correspondence problem)
  - Temporal growing number of hypotheses
  - Detection of loops
- Measurement noise
  - All electrical devices produce noise
  - Quantization
  - Solution: Integrate while moving slowly
- Limited field of view



Challenge in Robotic Mapping

- Dynamic Environment
  - Robot is not the only moving entity in the map
  - Hypotheses on what happened with the environment
    (e.g. moving people, doors opened and closed)
  - Common assumption: Robot is the only moving object (approximately true for short time windows => static world)

- Exploration and Path Finding
  - Good techniques for fully modeled maps (e.g. A*)
    Not for partial maps
  - Mapping should run in real-time
  - Loss of information for each movement of the robot

Types of Maps

- Classification in topological and metrical

	Topological	**Metrisch**
Scale	Large-scale space	Small-scale space
Sensor inputs	Abstracts sensor in-puts	Stores sensor inputs
Computational effort	Low	High
Memory consumption	Low	High
Sensitive to noise	Less	More
Real-time mapping	Yes	Depends on compu-tational power
Resolution	Very low	High

- Choice depends on field of application and available resources

Metrical Maps

- Objects (e.g. grids, shapes) with metrical coordinates
- Finer grained than topological maps
- More computational effort
- Today: mostly two-dimensional grids

SLAM: Simultaneous Localization and Mapping

- Unknown environment, simultaneous:
  - Constructing map
  - Tracking robot's pose
- Probabilistic approaches, e.g:
  - Kalman Filters (Bayes Filter)
  - Monte Carlo Methods

Metrical Maps

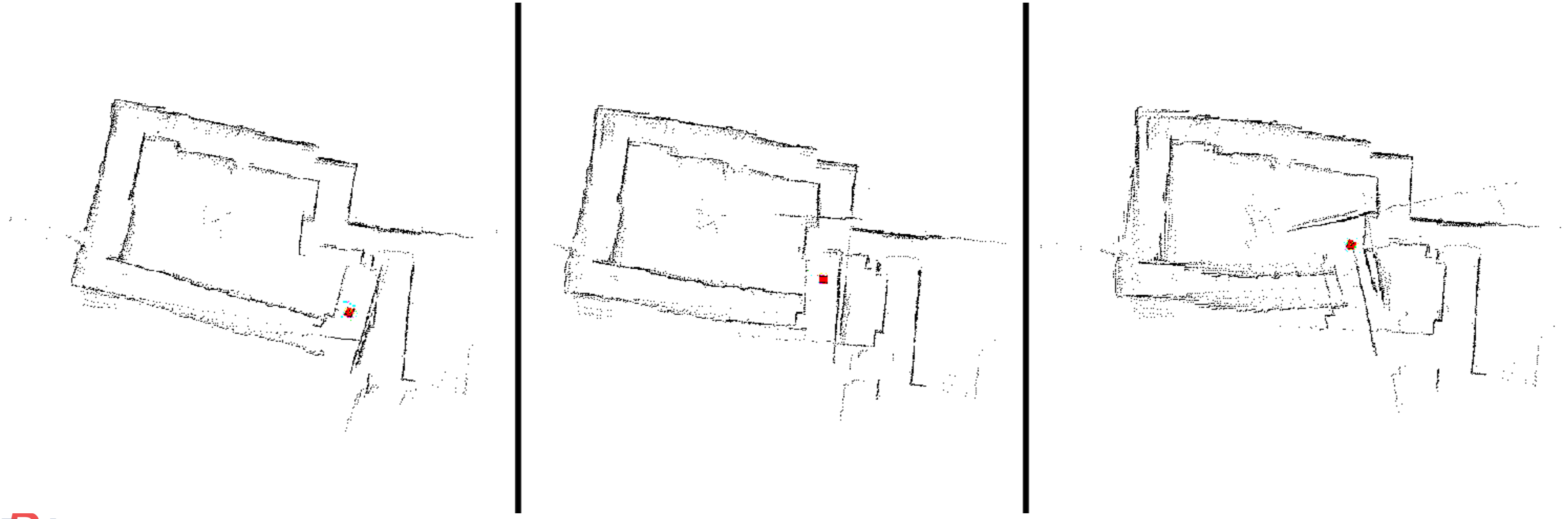
- Kalman Filters
  - Invented 1960 by Rudolf Kálmán
  - Estimate state of dynamic system with incomplete (noisy) data
  - Linear quadratic estimation (LQE): Minimize error
  - Efficient recursive (Bayes) filter
  - Incremental (SLAM)
  - Disadvantage: Does not solve the correspondence problem

Metrical Maps

- Expectation Maximization (EM) algorithm
  - SLAM alternative to Kalman Filters
  - Tries to maximize the expectation for map and pose
  - Stores all sensory inputs
  - Processes data multiple times
  - Thus: not incremental
  - Solves SLAM problem by iterating between two steps:
    - Expectation Step: Find all possible robot poses in map
    - Maximization Step: Calculate most likely map for poses

Metrical Maps

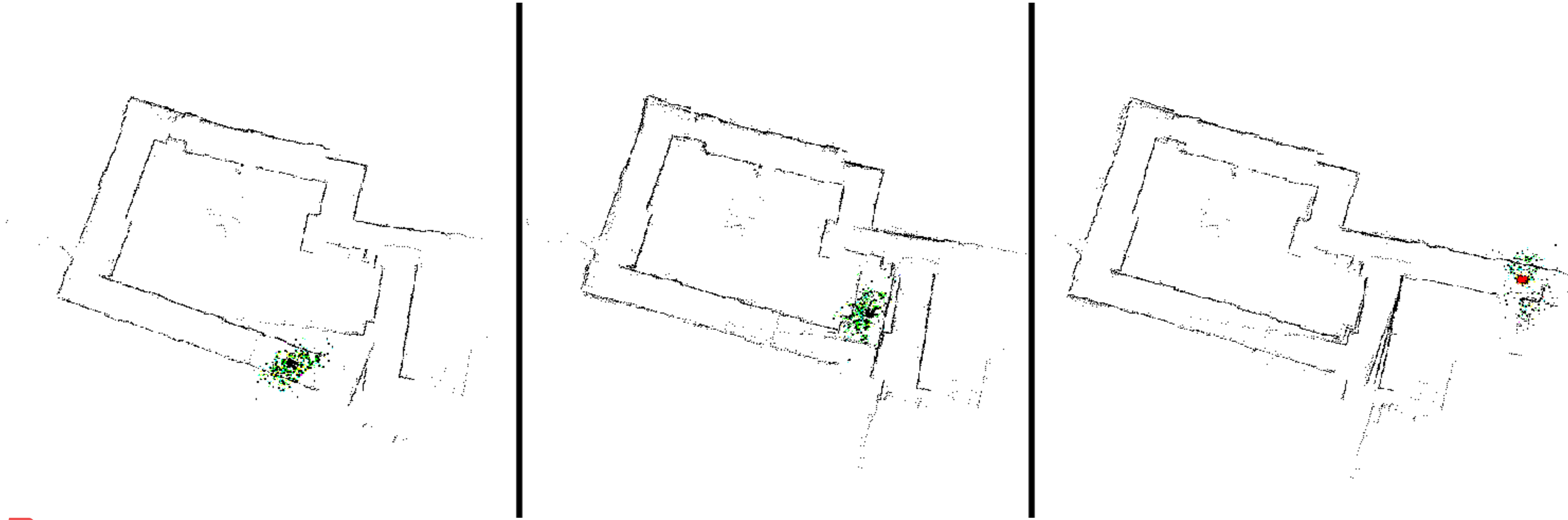
- Incremental Maximum Likelihood Method
  - Combines strength of EM and Kalman Filters
  - Simple and popular
  - Incremental => real-time mapping
  - Disadvantage: No cyclic loops



RLAB THE ROBOTICS RESEARCH LAB

Metrical Maps

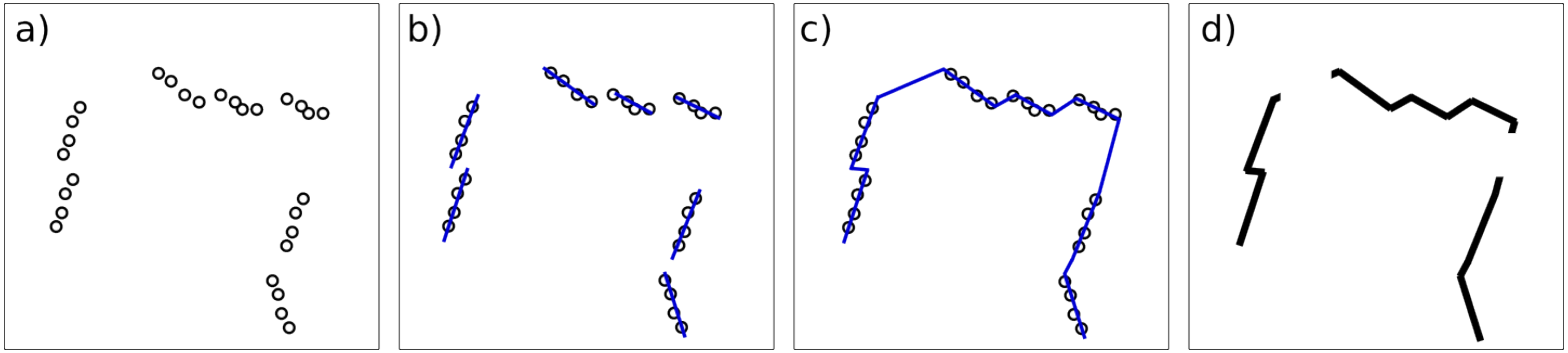
- Hybrid approaches
  - Allows cyclic maps
  - Inconsistence: Reset map to backwards in time
  - Disadvantages:
    - complex ambiguities (nested loops) not supported
    - not real-time



Metrical Maps

Discrete Segment Evolution

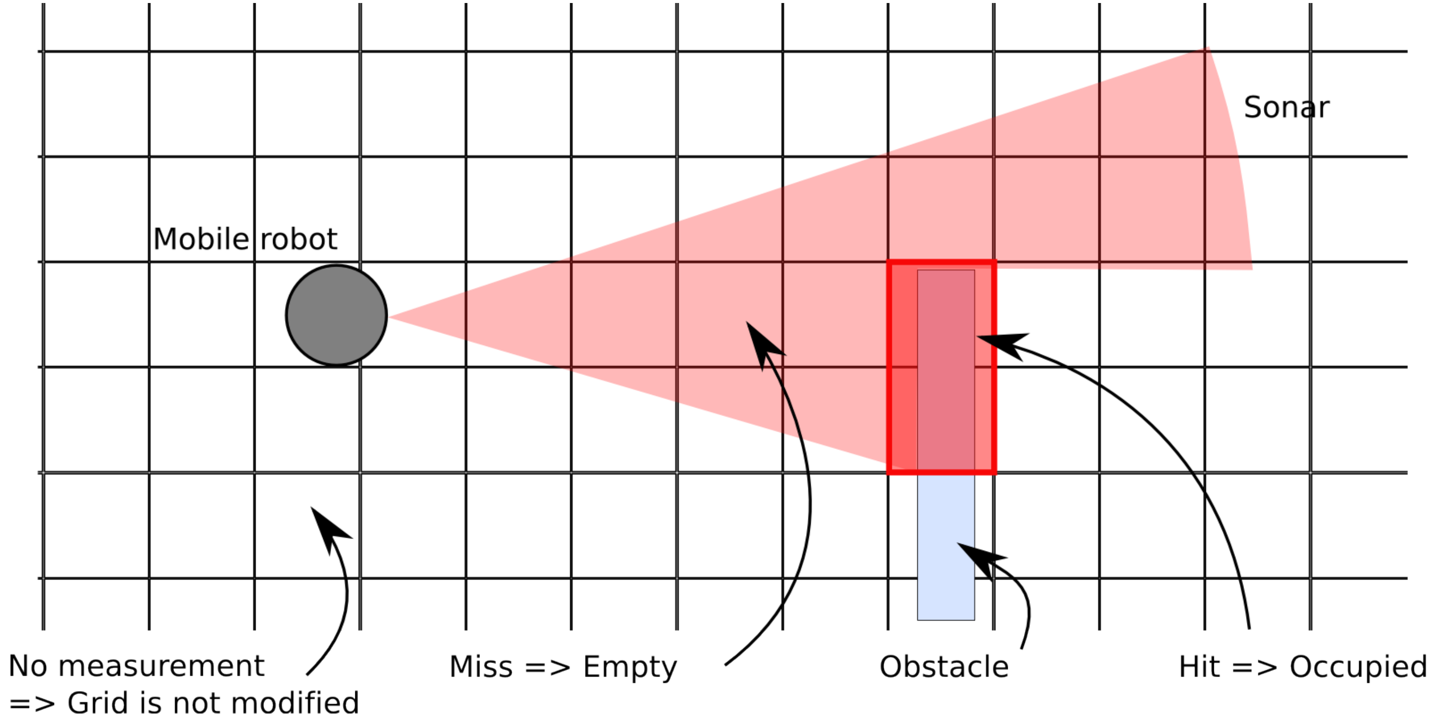
- Advantage: No odometry information
- Specialized for 2D range sensors:
  1) Approximate scan points with line segments
  2) Segment sorting step
  3) Splitting into multiple lists
- Creating the map by overlay and matching



Metrical Maps

Occupancy Grid Maps

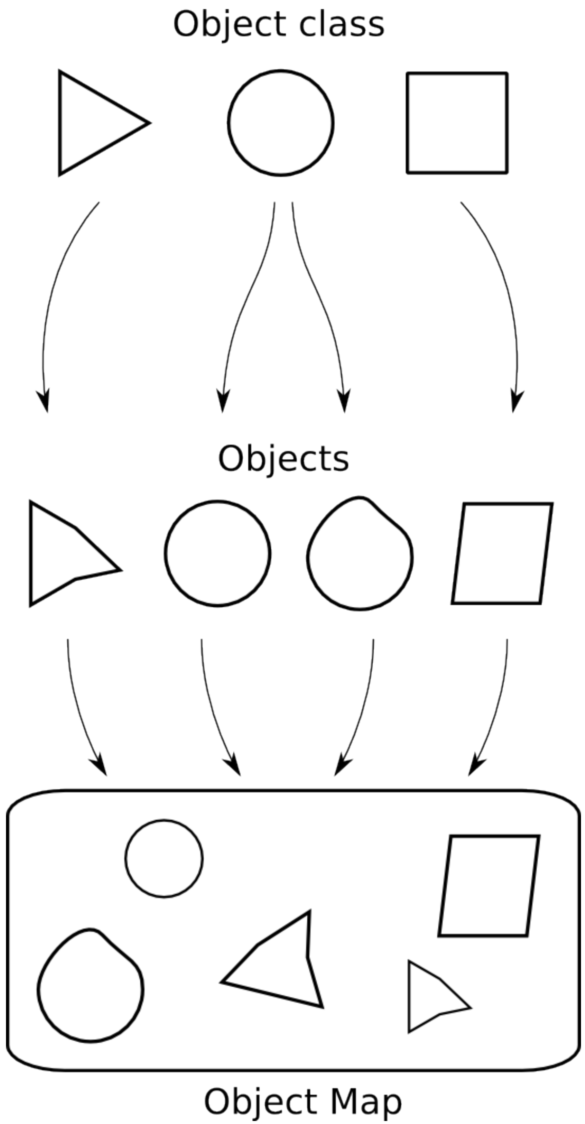
- Known robot pose (mostly no SLAM)
- Two- or three-dimensional grid
- Robust and easy to implement



Metrical Maps

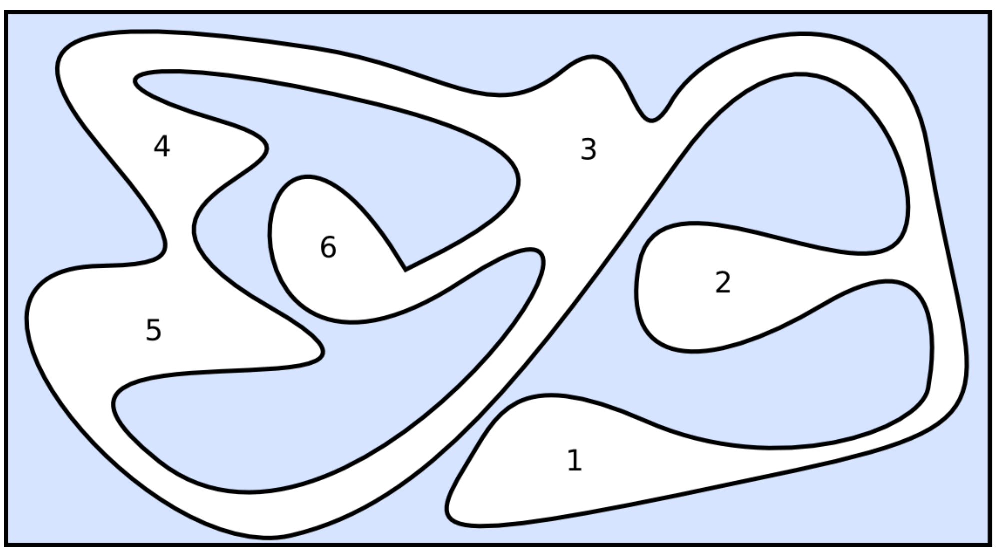
Object Maps

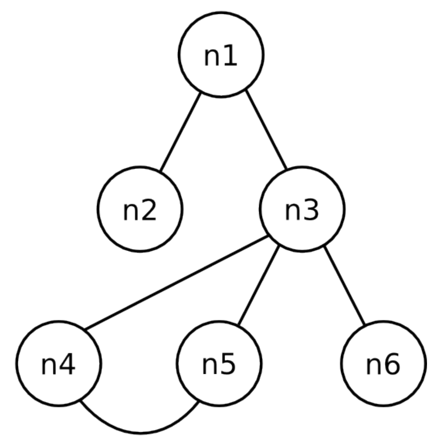
- Store geometric shapes
- More accurate
  - Predefined objects (classes)
  - Modification of objects
- Dynamic environment: Object properties
- More compact
- Better Human-Computer Interaction



Topological Maps

- Environment as (cyclic) graph
  - Navigation information on edges
- Large-scale space
- Formal guarantees that the correct map is generated
- Less computational and memory effort

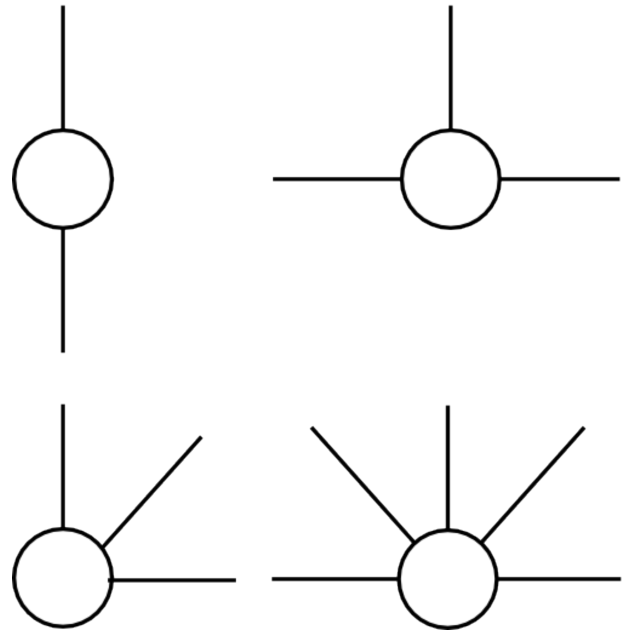




Topological Maps

TOUR Model

- One of the first topological approaches (Kuipers, 1977)
- Space is described using five entities:
  - Street networks (signatures)
  - Routes
  - Relative position of places
  - Dividing boundaries
  - Regions / Grouping

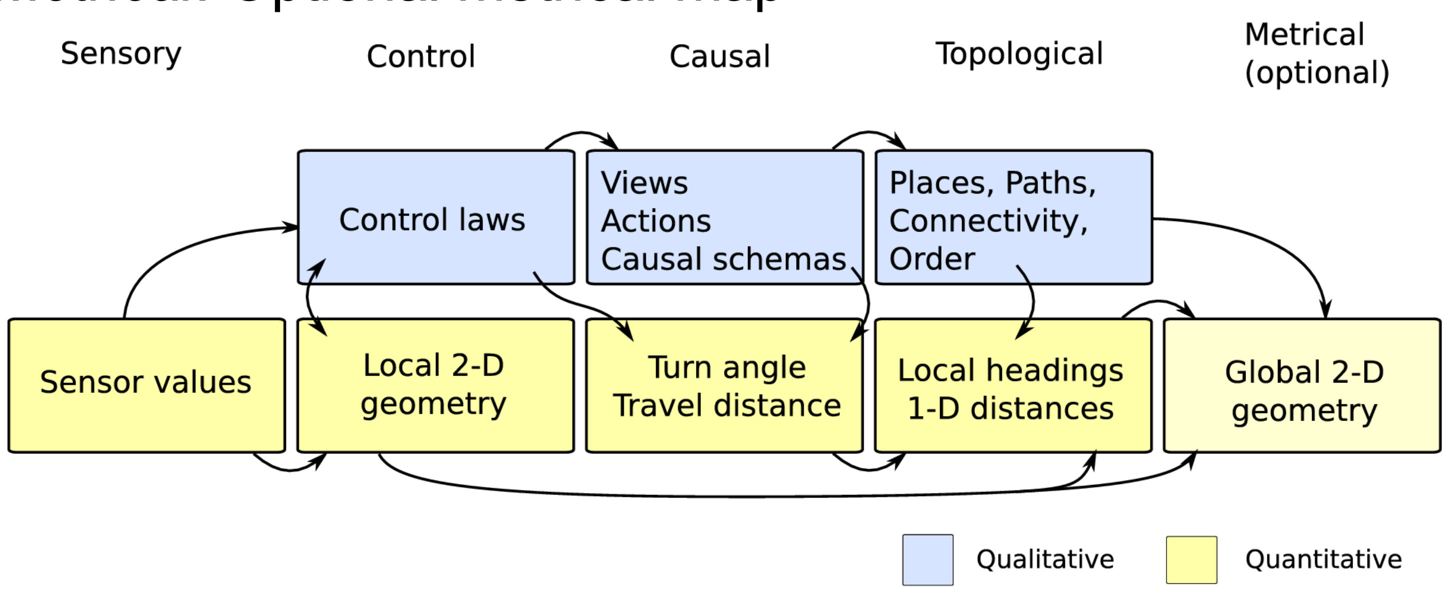


Street network signatures

Topological Maps

Spatial Semantic Hierarchy (SSH)

- Multiple levels of partial knowledge:
  - *Sensory*: Continuous world
  - *Control*: Control laws
  - *Causal*: Discrete states in environment
  - *Topological*: Topological map (places, paths, regions)
  - *Metrical*: Optional metrical map



Metrical-Topological Hybrids

Cognitive Mapping

- 2D local maps: „Map in the Head“
- Topological links: „Atlas in the Head“
  - Little by little strengthened
  - Modify erroneous connections over time
  - Strong enough: connect metrical maps
  - Store more data when more resources available

Hybrid SSH

- Extension: Local Perceptual Map (LPM)
  - SLAM
  - local path planing
  - obstacle avoidance
- Problems: simply discard LPM

Summary

- All algorithms: Advantages and disadvantages
  - e.g. most algorithms assume static world

- Situation encouraged over last two decades
- Still much to do
  - Unstructured environment
    - outdoor: vegetation, underwater etc.
    - indoor: (moving) people
  - Other domains like multi robot mapping

- “Do the right thing” function

References

- Gregory Dudek, Paul Freedman, Souad Hadjres. Using Local Information in a Non-Local Way for Mapping Graph-Like Worlds. 1993.
- B. J. Kuipers. The Spatial Semantic Hierarchy. Artificial Intelligence, 119, 2000.
- Benjamin Kuipers. The Skeleton in the Cognitive Map: A Computational Hypothesis. 2001.
- B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, F. Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy, 2004.
- Benjamin Kuipers. Modeling Spatial Knowledge. In IJCAI, 1977.
- B. J. Kuipers. The cognitive map: Could it have been any other way? 1983b
- S. Thrun. Robotic Mapping: A Survey. In G. Lakemeyer, B. Nebel (Hrsg.), Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002. to appear.

References

- Rolf Lakaemper, Longin Jan Latecki, Xinyu Sun, Diedrich Wolter. Geometric Robot Mapping, 2005.
- Sidomar T. Monteiro, Hideichi Nakamoto, Hideki Ogawa, Nobuto Matsuhira. Robust mobile robot map building using sonar and vision. 2005.
- E. Remolina, B. Kuipers. Towards a general theory of topological maps, 2004.
- A. Scott. Quantitative and qualitative comparison of three laser-range mapping algorithms using two types of laser scanner data, 2000.
- Benjamin Kuipers, Yung-Tai Byun. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. Technischer Bericht AI90-120, 1, 1990.

Thank you for your attention!

Appendix: Discrete Segment Evolution

Algorithm:

Initial map is equal to the first scan: $G_0 = S_0$

- Three steps:
  - Correspondence: New scan S_i is positioned over previous map G_{i-1} (assumes small changes, position: old pose)
  - Alignment: Rotate and translate S_i => new pose
  - Merging: Combine S_i and G_{i-1} => Result is new map G_i