MODULE 2: AUTONOMOUS BEHAVIOR

Nicholas HO, PhD

Institute of System Science, NUS

Reference (Optional)

- EE4308 Advances in Intelligent Systems and Robotics
- ME5402/EE5106R Advanced Robotics
- Siegwart, R., Nourbakhsh, I. R., Scaramuzza, D., & Arkin, R. C. (2011). *Introduction to autonomous mobile robots*. MIT press.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: modelling, planning and control*. Springer Science & Business Media.
- Jazar, R. N. (2010). *Theory of applied robotics: kinematics, dynamics, and control*. Springer Science & Business Media.

Topics

- Fundamentals of control techniques
- Localization schemes
- Robotic spatial mapping

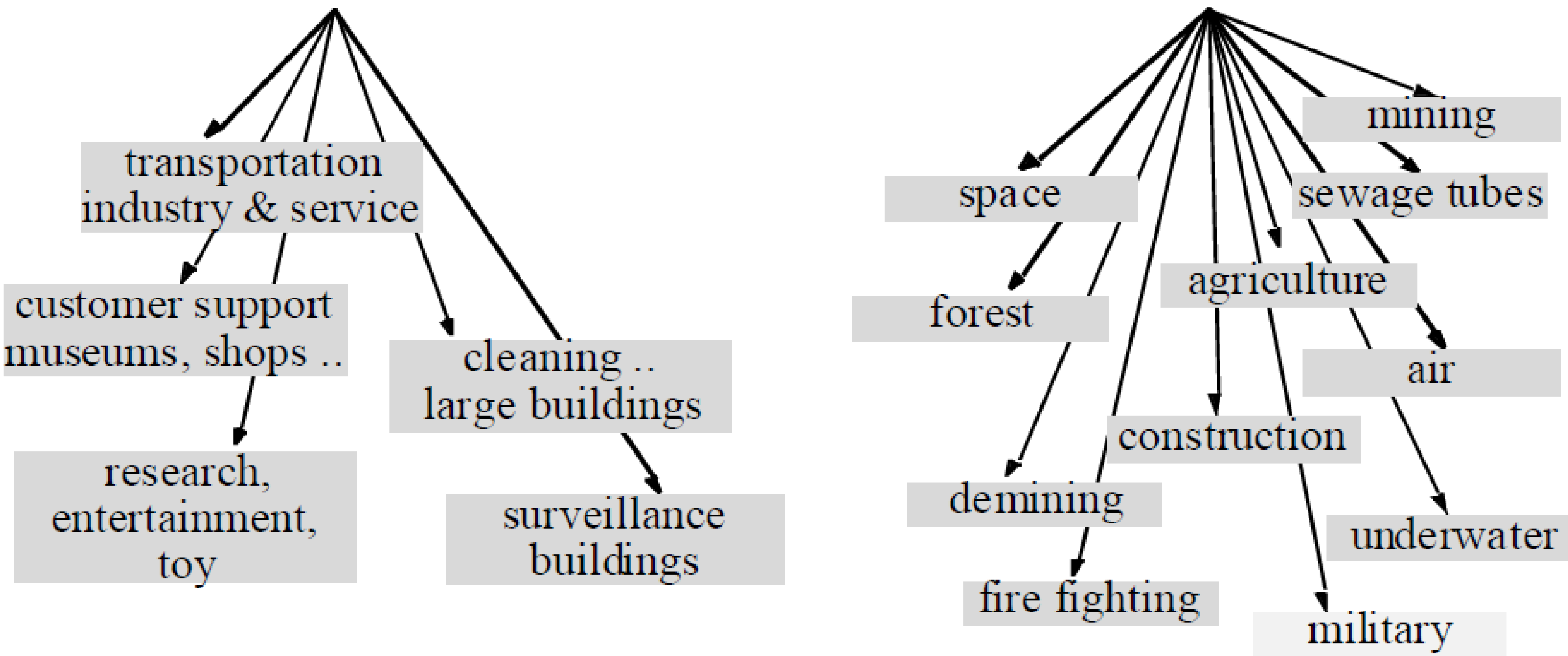
Applications of Mobile Robots

Indoor

Structured Environments

Outdoor

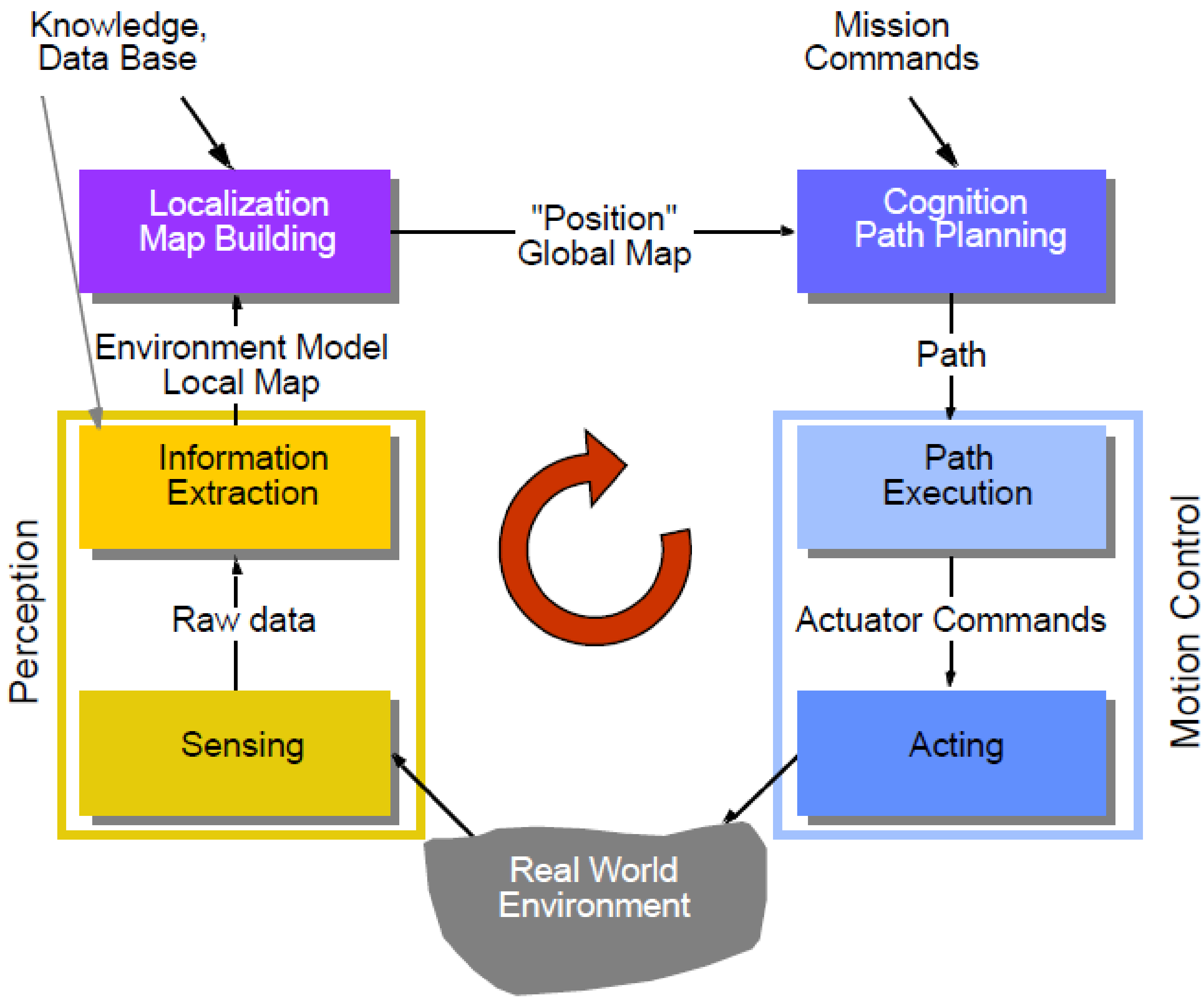
Unstructured Environments



Autonomous Mobile Robots

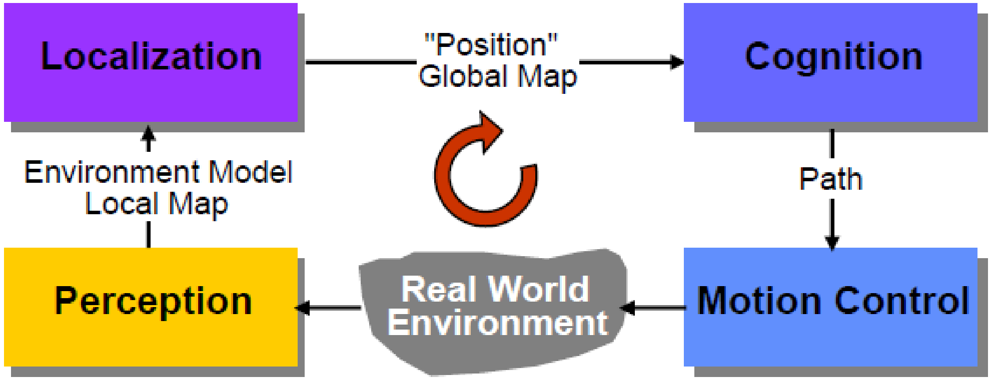
- **The 3 key questions in Mobile Robotics**
  - Where am I ?
  - Where am I going ?
  - How do I get there ?
- **To answer these questions the robot has to**
  - have a **model of the environment** (given or autonomously built)
  - **perceive and analyze the environment** (aka perception)
  - **find its position within the environment** (aka localization)
  - **plan and execute the movement** (aka navigation)
- Locomotion and Navigation
  - Perception, Localization, Planning and motion generation

General Control Scheme for Mobile Robot Systems



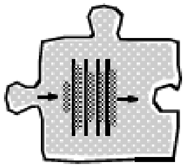
Control Architectures / Strategies

- Control Loop
  - dynamically changing
  - no compact model available
  - many sources of uncertainty

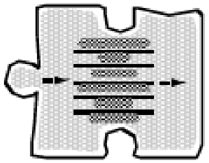


Two Approaches

- ➢ *Classical AI*
  - o *complete modeling*
  - o *function based*
  - o *horizontal decomposition*

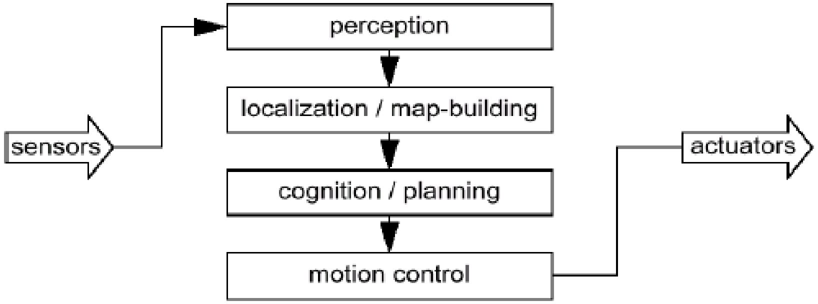


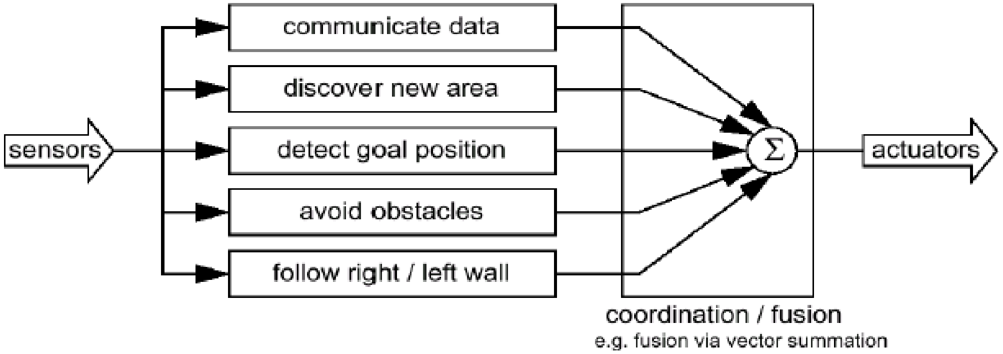
- ➢ *New AI, AL*
  - o *sparse or no modeling*
  - o *behavior based*
  - o *vertical decomposition*
  - o *bottom up*



Control Architectures: Two Approaches

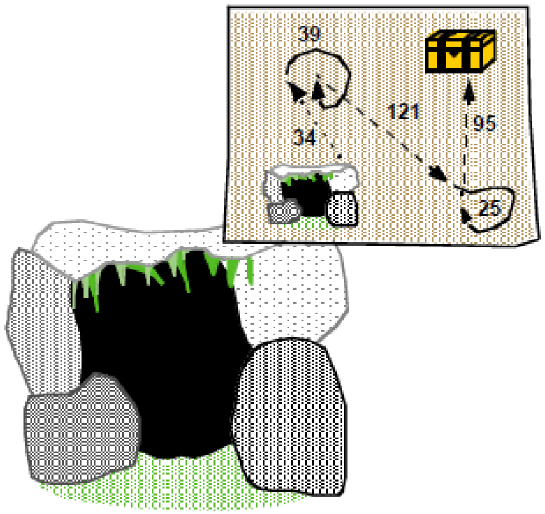
- **Classical AI (model based navigation)**
  - *complete modelling*
  - *function based*
  - *horizontal decomposition*
- **New AI, AL (behavior based navigation)**
  - *sparse or no modelling*
  - *behaviour based*
  - *vertical decomposition*
  - *bottom up*
- Possible Solution
  - *Combine Approaches*





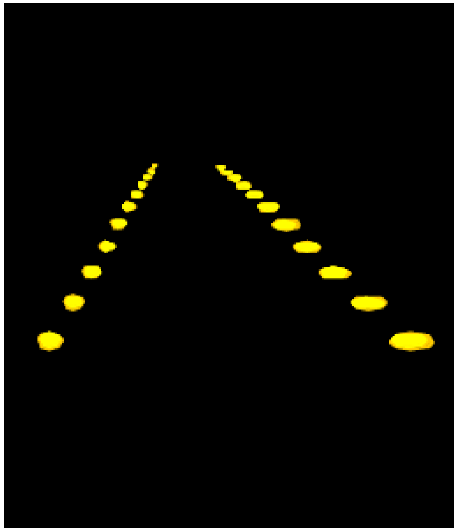
Environment Representation and Modelling

- Odometry



How to find a treasure

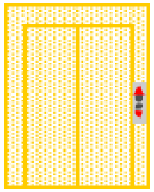
- Modified Environments



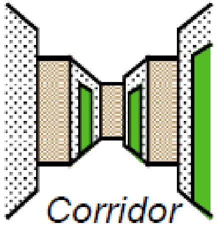
Landing at night

  ➢ *expensive, inflexible*

- Feature-based Navigation

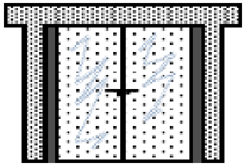


Elevator door



Corridor crossing

Eiffel Tower

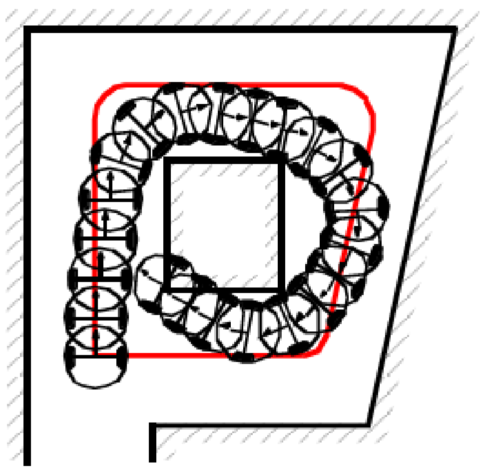


Entrance

  ➢ *still a challenge for artificial systems*

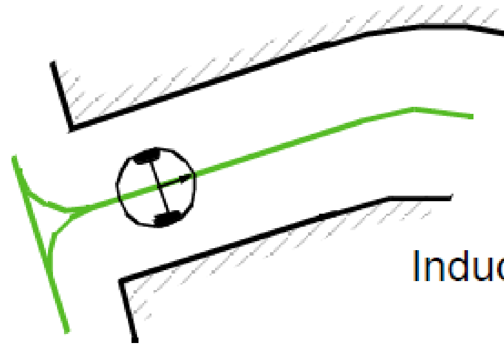
Methods for Navigation: Approaches with Limitations

- Incrementally
  (dead reckoning)

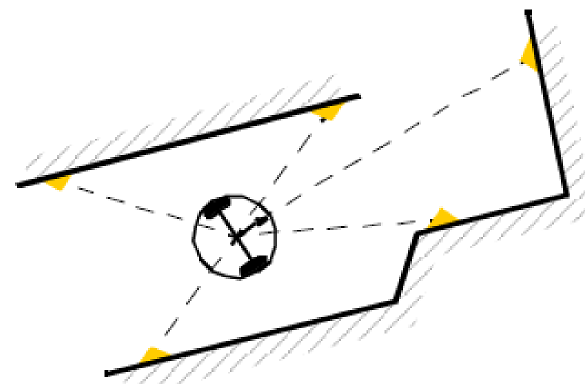


Odometric or initial sensors
(gyro)

- Modifying the environments
  (artificial landmarks / beacons)



Inductive or optical tracks (AGV)

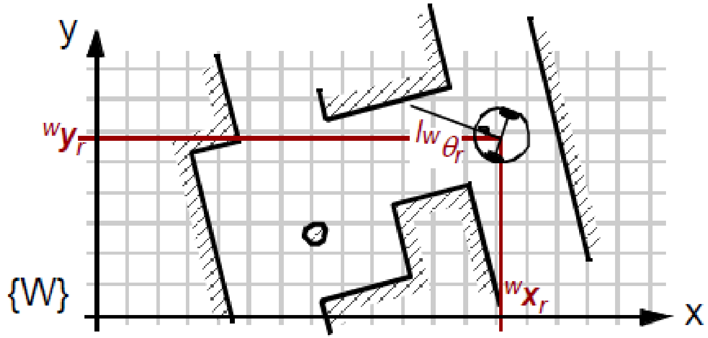


Reflectors or bar codes

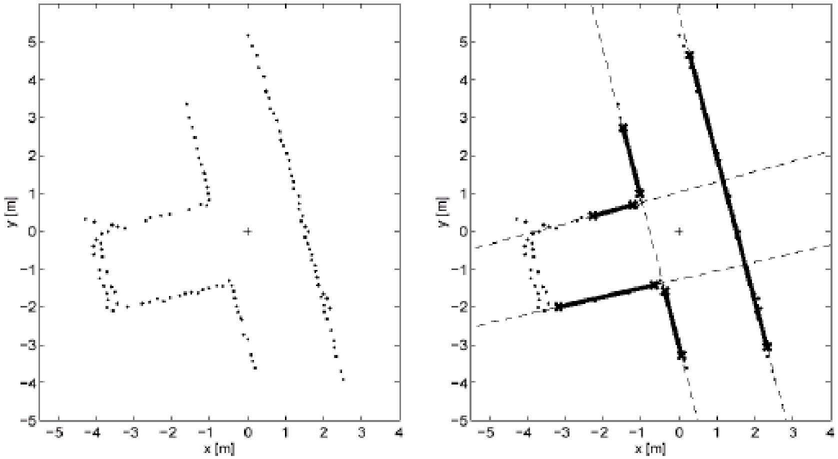
➢ *expensive, inflexible*

Method for Localization: The Quantitative Metric Approach

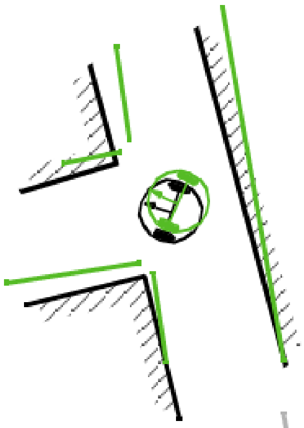
1. A priori Map: Graph, metric



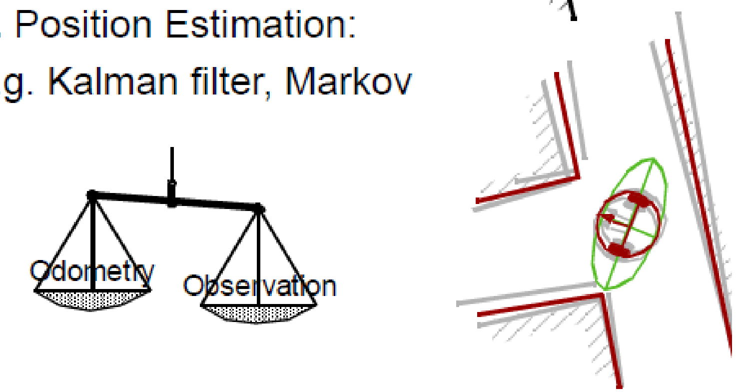
2. Feature Extraction (e.g. line segments)



3. Matching:
Find correspondence of features



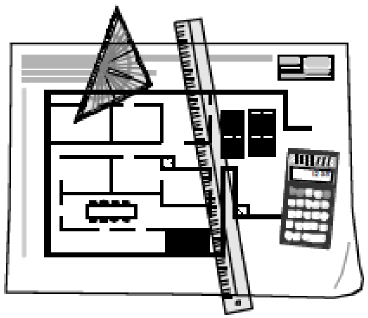
4. Position Estimation:
e.g. Kalman filter, Markov



- representation of uncertainties
- optimal weighting according to a priori statistics

Map Building: How to Establish a Map

1. By Hand



2. Automatically: **Map Building**

The robot **learns** its environment

Motivation:

- by hand: hard and costly
- dynamically changing environment
- different look due to different perception

3. Basic Requirements of a Map:

- *a way to incorporate newly sensed information into the existing world model*
- *information and procedures for estimating the robot's position*
- *information to do path planning and other navigation task (e.g. obstacle avoidance)*

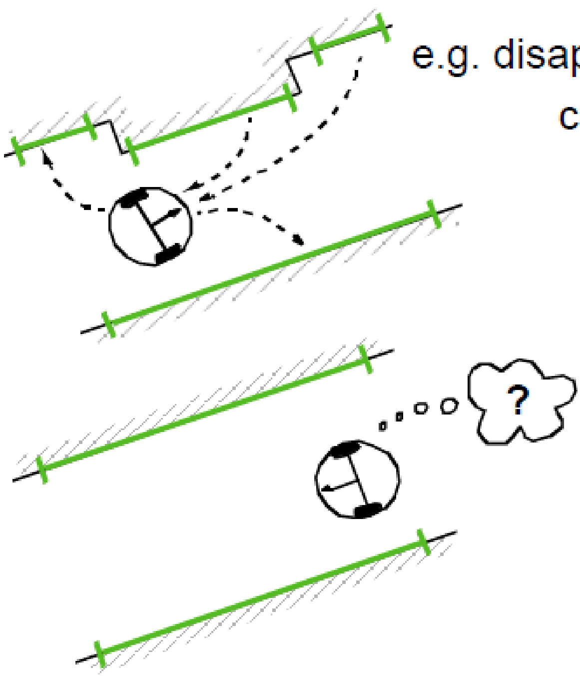
predictability

- Measure of Quality of a map
  - *topological correctness*
  - *metrical correctness*
- But: Most environments are a mixture of predictable and unpredictable features
  $\rightarrow$ hybrid approach

  model-based vs. behaviour-based

Map Building: The Problems

1. Map Maintaining: Keeping track of changes in the environment

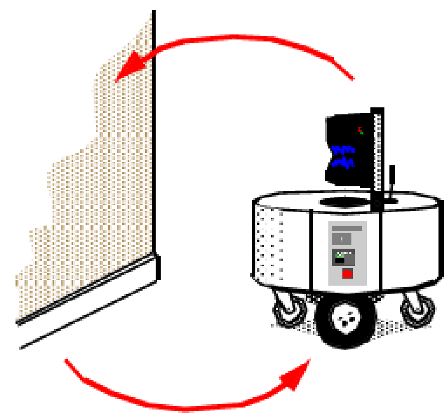


e.g. disappearing cupboard

- e.g. measure of belief of each environment feature

2. Representation and Reduction of Uncertainty

position of robot -> position of wall

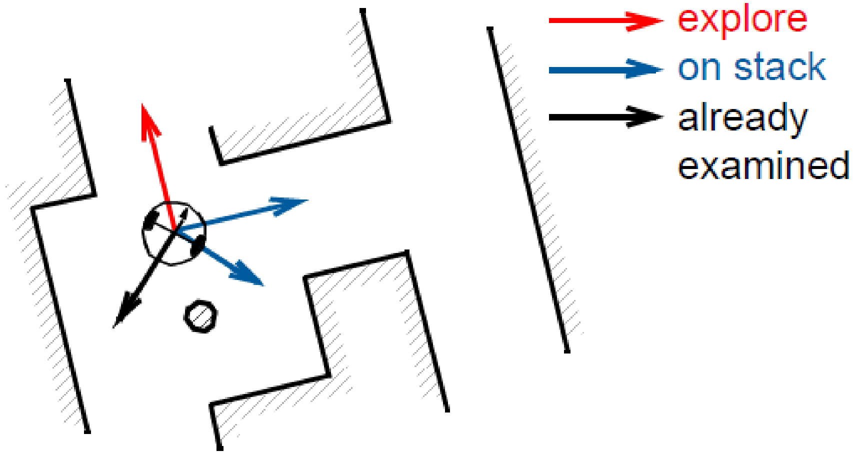


position of wall -> position of robot

- probability densities for feature positions
- additional exploration strategies

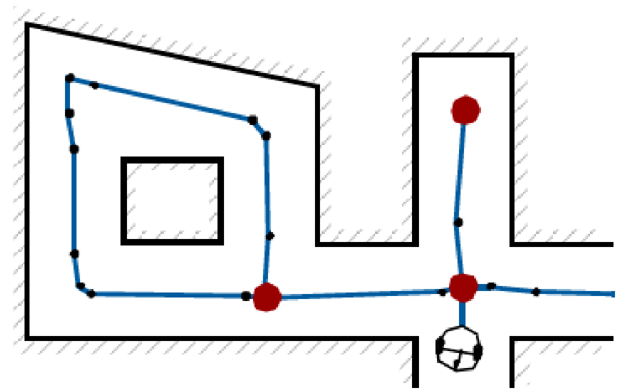
Map Building: Exploration and Graph Construction

1. Exploration



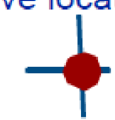
- provides correct topology
- must recognize already visited location
- backtracking for unexplored openings

2. Graph Construction



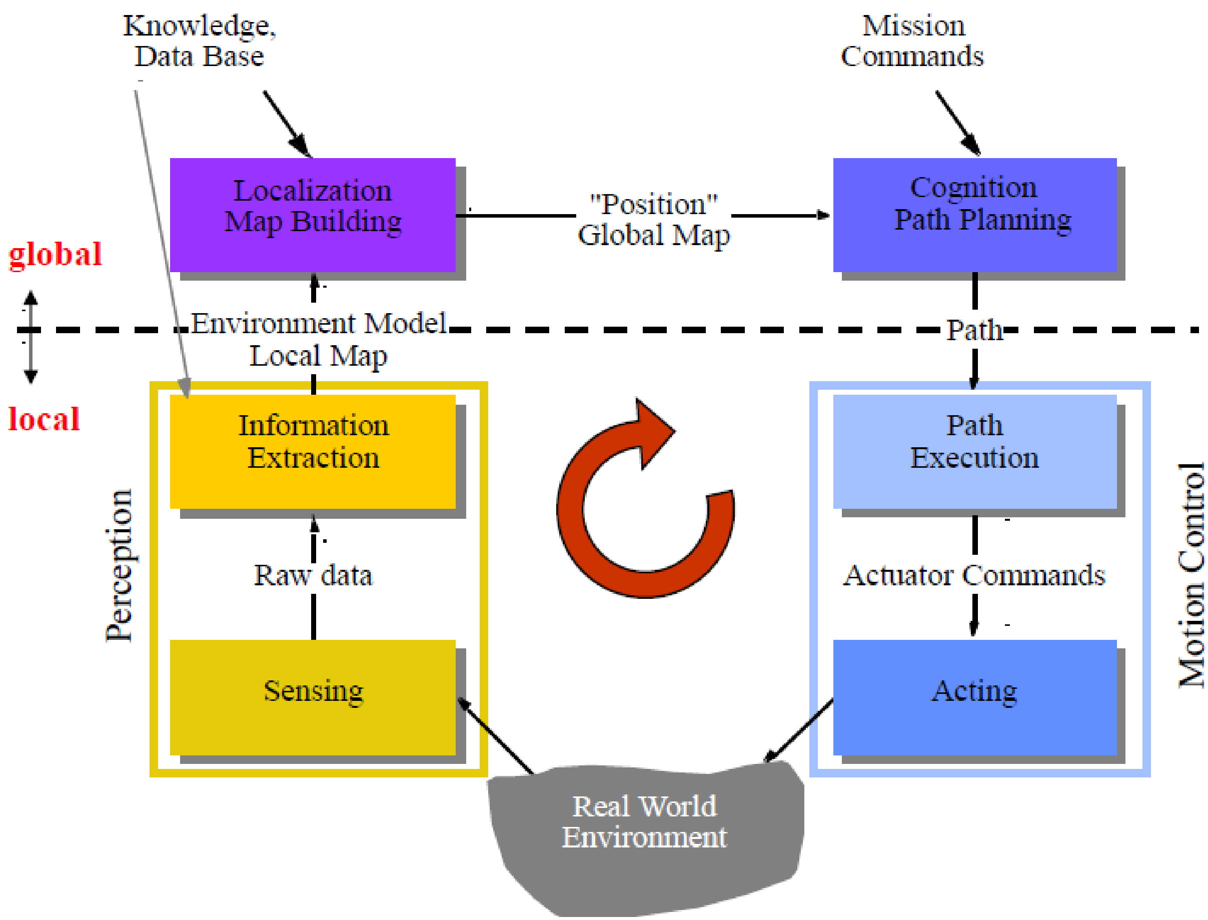
Where to put the nodes?

- Topology-based: at distinctive locations



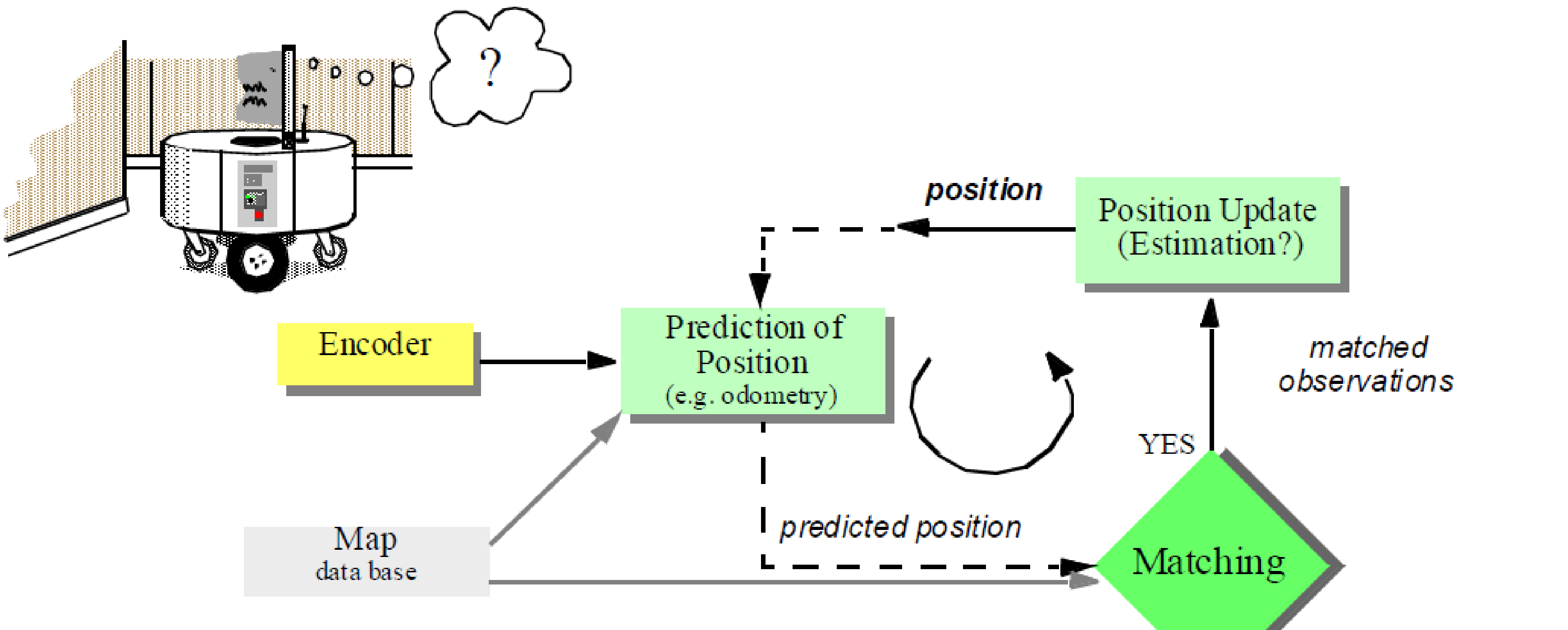
- Metric-based: where features disappear or get visible

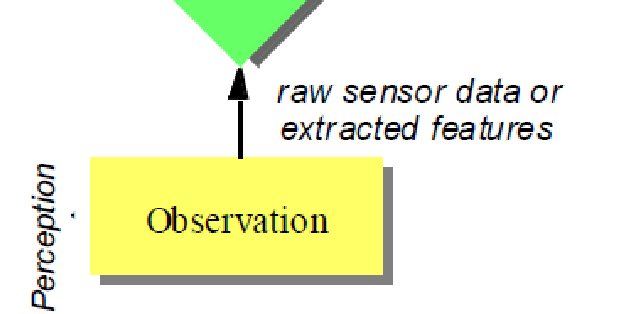
Control of Mobile Robots



- Most functions for safe navigation are 'local' not involving localization nor cognition
- Localization and global path planning ➔ slower update rate, only when needed
- This approach is pretty similar to what human beings do

Localization, Where am I?





- *Odometry, Dead Reckoning*
- *Localization based on external sensors, beacons or landmarks*
- *Probabilistic Map Based Localization*

Challenges of Localization

- Knowing the absolute position (e.g. GPS) is not sufficient
- Localization in human-scale in relation with environment
- Planning in the *Cognition* step requires more than position as input
- Perception and motion plays an important role
  - Sensor noise
  - Sensor aliasing
  - Effector noise
  - Odometric position estimation

Sensor Noise

- Sensor noise in **mainly influenced by environment** e.g. surface, illumination …

- **or by the measurement principle itself**. E.g. interference between ultrasonic sensors

- Sensor noise drastically reduces the useful information of sensor readings. **The solution is:**
  - to take **multiple reading** into account
  - employ **temporal and/or multi-sensor fusion**

Sensor Aliasing

- In robots, **non-uniqueness of sensors' readings is the norm**
- Even with multiple sensors, there is a **many-to-one mapping** from environmental states to robot's perceptual inputs
- Therefore **the amount of information perceived by the sensors is generally insufficient to identify the robot's position from a single reading**
  - Robot's localization is usually based on **a series of readings**
  - Sufficient information is recovered by the robot **over time**

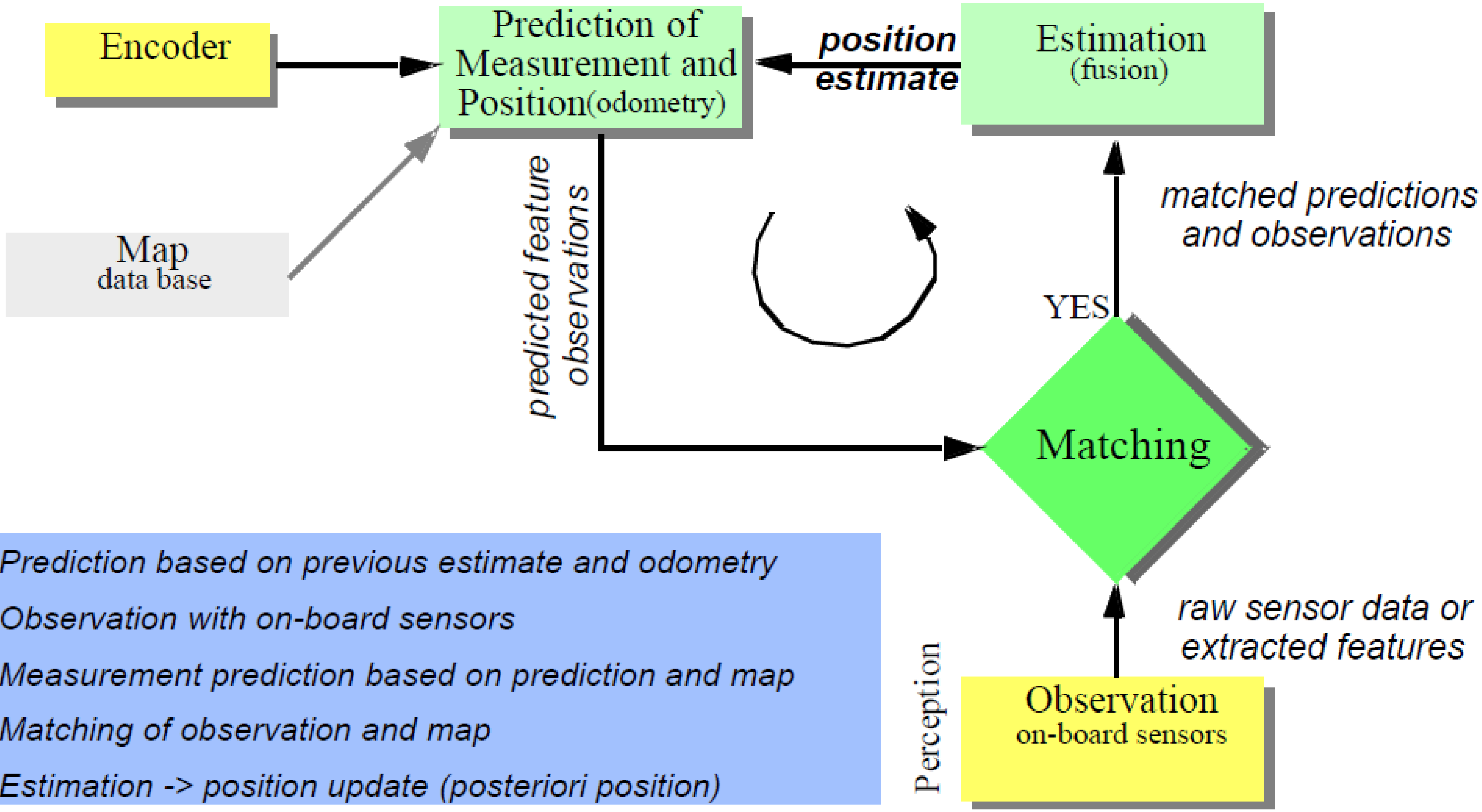
State-of-the-Art:
Current Challenges in Map Representation

- **Real world is dynamic**
- **Perception is still a major challenge**
  - Error prone
  - Extraction of useful information difficult
- How to **build up topology** (boundaries of nodes)
- Travelling across **open space (i.e. without any surrounding landmarks)**
- **Sensor fusion**

Probabilistic, Map-Based Localization

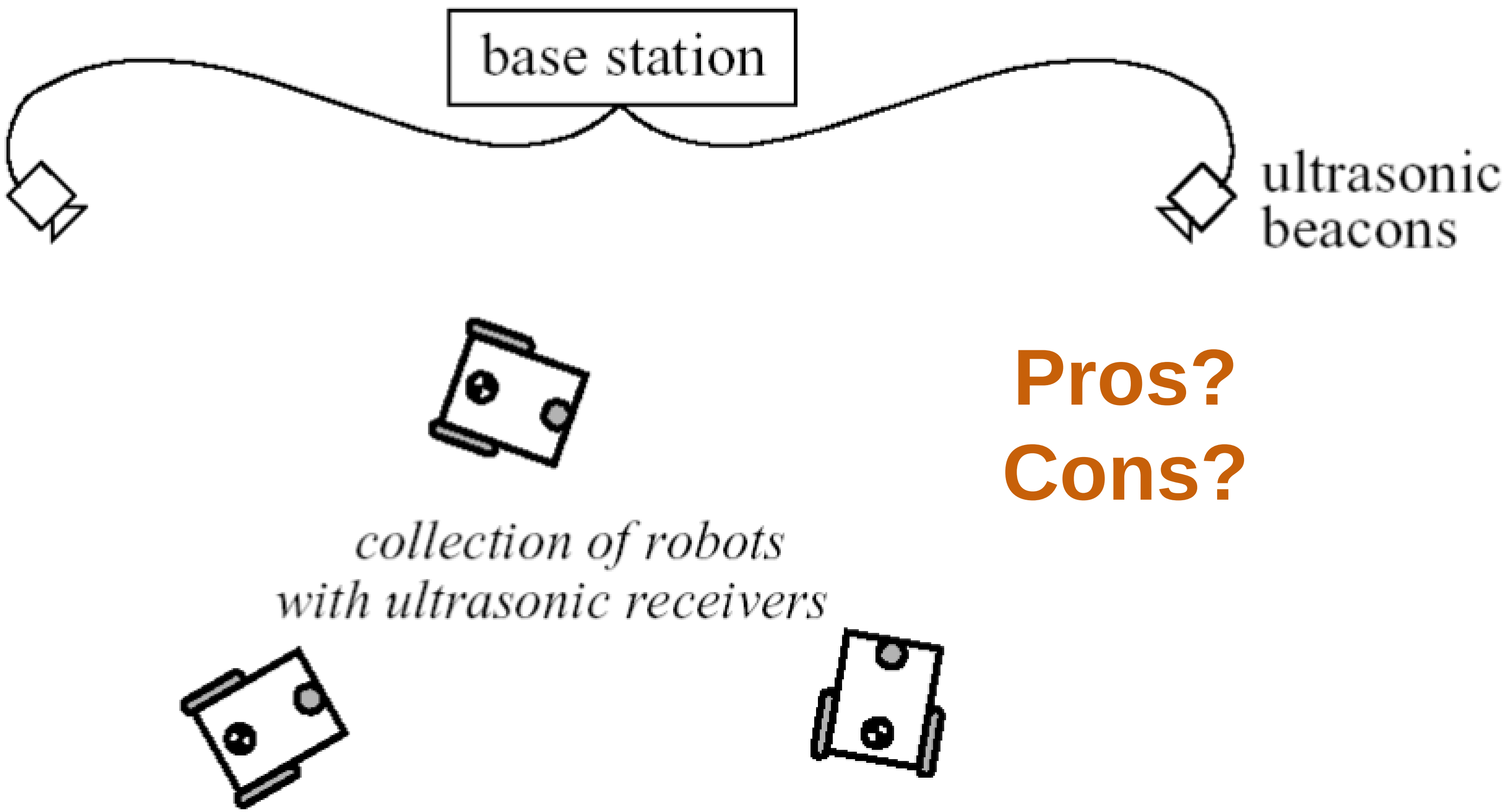
- Consider a mobile robot moving in a known environment
- As it start to move, say from a precisely known location, it might **keep track of its location using odometry**
- However, **after a certain movement the robot will get very uncertain about its position**
  - update using an observation of its environment.
- **observation also lead to an estimate of the robots position which data can than be fused with the odometric estimation** to get the best possible update of the robots actual position

The Five Steps for Map-Based Localization



1. Prediction based on previous estimate and odometry
2. Observation with on-board sensors
3. Measurement prediction based on prediction and map
4. Matching of observation and map
5. Estimation -> position update (posteriori position)

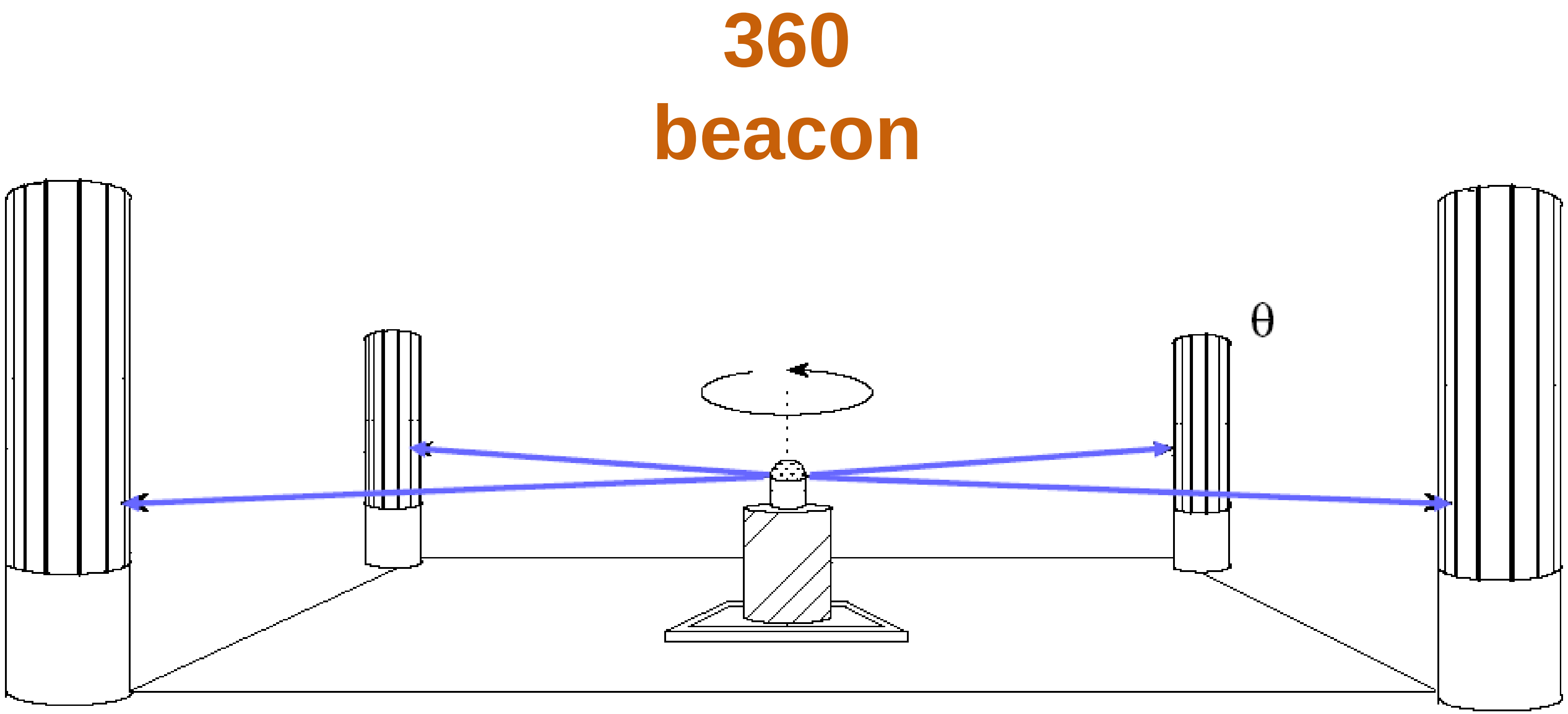
Positioning Beacon Systems: Triangulation



Pros?

Cons?

Positioning Beacon Systems: Triangulation



Autonomous Map Building

- Starting from an **arbitrary initial point**,
- a mobile robot should be able to **autonomously explore** the environment with its on board sensors,
- **gain knowledge** about it,
- **interpret the scene**,
- **build an appropriate map**
- and **localize itself** relative to this map

SLAM

The Simultaneous Localization and Mapping Problem

End of Module 2