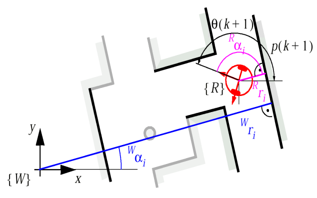
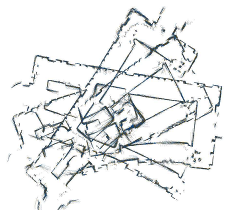
Robotic Mapping: A (High-Level) Survey

Sebastian Thrun, 2002
Original presentation: David Black-Schaffer, Kristof Richmond





Project

2

- Final milestone due at the very end of the semester
- You should have implemented (and turned in) your final strategy
  - Improvements, tweaks and bug-hunting are okay
  - You shouldn't be doing major software changes now!
- Milestone 5 code is what you will run at the end
- (Much of) this time is for **writeup**
- Project description gives high-level goals
  - But you should know how to write a research paper

HW 4

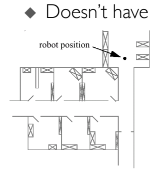
3

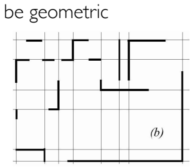
- Shorter than HW3
- Primarily about localization and mapping
- More conceptual, but still leave time for it

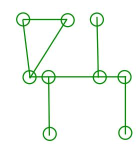
What is Mapping?

4

- Building a **model of the robot's environment**
  - Based on sensor data obtained in that environment
- Many different environment representations exist
- Mapping means constructing one
  - Based on sensor data
  - Doesn't have to be geometric







Mapping ≠ SLAM?

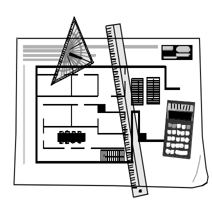
5

- Can you do mapping without localization?
  - Sort of.
- Gathering environment data doesn't depend on knowing where you are, *but…*
- If you don't track robot's location, the **relative position** of different sensor readings is harder to calculate
- Some problems (e.g., loop closure) may be impossible

How to Get a Map?

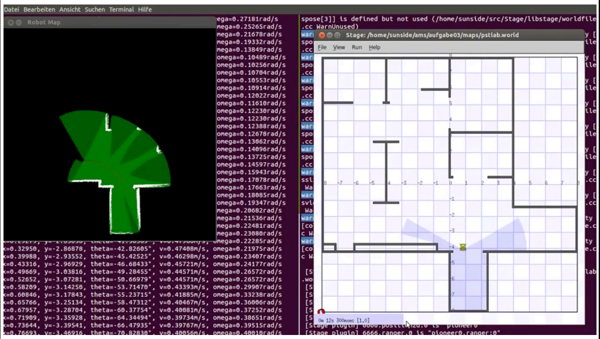
6

- By hand?
  - Slow, expensive and imprecise
- Automatically: Mapping
  - The robot learns its environment
- Can cope with dynamically changing environment
- Map is built from sensors that will be used to navigate



Basic Mapping Task

7



https://www.youtube.com/watch?v=eAbF3QBGwzA

Historical Overview

8

- Metric Mapping
  - Geometric representations
  - Occupancy Grids
  - Larger maps much more computationally intensive
- Topological Mapping
  - Milestones with connections
  - Require navigation information
  - Hard to scale

The Problems

9

- Measurement noise
  - Sensor and position noise are not independent
- Map size
  - High resolution maps can be very large
- Correspondence
  - Do multiple measurements at different times correspond to the same object?
  - Loop closure is a subset of this
- Dynamic environments
  - Many algorithms assume a static environment

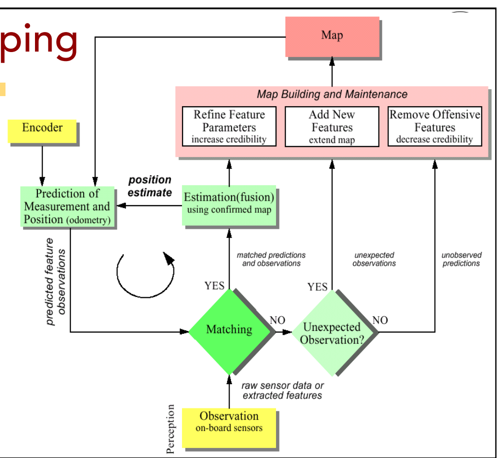
Mapping

10

- Basic requirements:
- A way to incorporate sensed information into world model
- Sufficient information for navigation tasks
  - Estimating position, path planning, obstacle avoidance, …
- Correctness
- Predictability
  - Most environments are a mixture of predictable and unpredictable features

Mapping

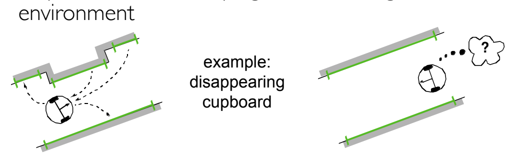
11



Challenges: Maintenance

15

- Map Maintenance: keeping track of changes in the environment

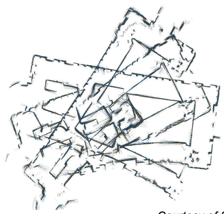


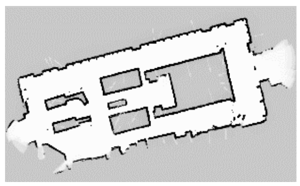
- Often uses a *measure of belief* of each feature
- If localizing: did the world change, or are you lost?

Challenges: Cyclic Environments

16

- Small local error accumulates
- This is usually irrelevant for navigation
- However, when closing loops, global error does matter



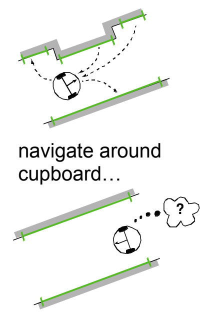


Courtesy of Sebastian Thrun

Dynamic Environments

17

- Dynamic changes require continuous (re-)mapping
- High-level features help if you can get them
  - Humans are more dynamic map features than walls
  - Can also learn that certain things are dynamic over time



Current State of Mapping

18

- Algorithms
  - Robust for static, structured, and limited-size environments
  - Probabilistic
  - Correspondence problem
  - Incremental vs. Multi-pass
    - Incremental algorithms only need one "pass" through data, and can possibly be run in real time
- Continuing areas
  - Dynamic environments
  - Semantic labeling of environments
  - Planning exploration paths of unknown environments

Why Probabilistic Mapping?

19

- Noise in commands and sensors
  - Commands are not executed exactly (e.g., slippage leads to odometry errors)
  - Sensors noise due to real world (e.g., angle of incidence and scattering)

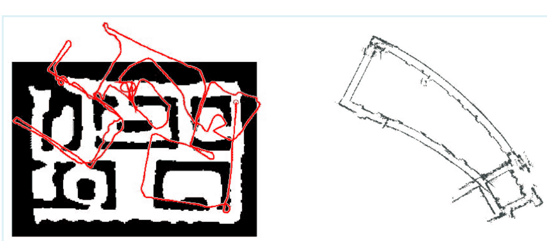
basically everything we've ever covered about error

- Noise is not statistically independent
  - Control errors accumulate with time/distance
  - Can't "average out" noise

Command Noise

20

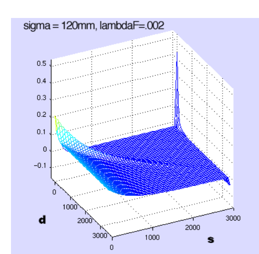
- Odometry Errors: heading and distance measurements accumulate errors with time



Sensor Noise

21

- Sensor probability model depends on the characteristics of the sensor and the object being sensed



Basic sonar probability model for angle of incidence = 0

Z-axis = P(s|d at angle = 0)

sigma: std. deviation for gaussian return from ideal surface

lambdaF: false positive rate

lambdaS: missed rate

d: distance to target on map

s: measured distance from sonar

Bayes Rule

22

- $p(x \mid d) = \eta\, p(d \mid x)\, p(x)$*
  - $p(x \mid d)$ is the probability of (the map) x being true given the (sensor) measurement d
  - $p(d \mid x)$ is the probability of the (sensor) measurement being being d given (an object at) x
  - $p(x)$ is the prior probability (of the map)

* η = *normalization constant*

Bayes Rule over Time

23

- Notation
  - s = pose of robot (x, y, Θ)
  - u = command given to robot
  - z = sensor measurement
  - m = map
- All are functions of time
  - z_t = sensor measurements at time t
  - z^t = all sensor measurements up to time t
  - (same for s, u, and m)

Bayes Filter

s: robot pose
u: command
z: sensor reading
m: map
$\square_t$/$\square^t$: *value at/up to time t*

24

$$p(s_t, m \mid z^t, u^t) = \eta \cdot p(z_t \mid s_t, m) \int p(s_t \mid u_t, s_{t-1}) p(s_{t-1}, m \mid z^{t-1}, u^{t-1}) ds_{t-1}$$

- Probability of a location (pose) and a map, given sensor readings and command.
- $p(z_t \mid s_t, m)$ is the sensor model
- $p(s_t \mid u_t, s_{t-1})$ is the motion model
- $p(s_{t-1}, m \mid z^{t-1}, u^{t-1})$ is probability we were **actually** where we thought we were last timestep

Sensor and motion models are usually static

SLAM

25

- SLAM
  - Simultaneous Localization And Mapping
  - Figure out where we are and what our world looks like at the same time
- Localization
  - Where are we?
  - Position error accumulates with movement
- Mapping
  - What does the environment look like?
  - Sensor error (not independent of position error)

Mapping ≠ SLAM?

26

- Can you do mapping without localization?
  - Sort of.
- Gathering environment data doesn't depend on knowing where you are, *but…*
- If you don't track robot's location, the **relative position** of different sensor readings is harder to calculate
- Some problems (e.g., loop closure) may be impossible

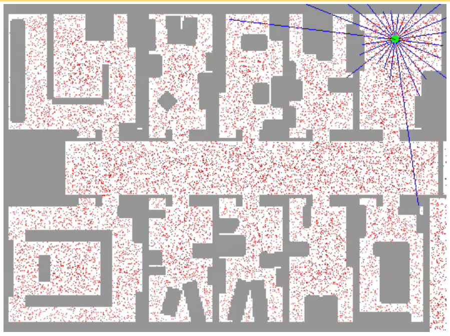
Monte Carlo Localization

27

- Probabilistic
  1. Start with a uniform distribution of possible poses (x, y, Θ)
  2. Compute the probability of each pose given current sensor data and a map
  3. Normalize probabilities
     - Throw out low probability points
     - Blur current points (we never know exactly where we are)
- Performance
  - Excellent in mapped environments
  - Need non-symmetric geometries

Monte Carlo Localization

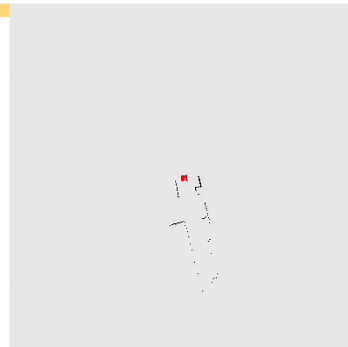
28



Thrun, Sebastian. "Animation of Monte Carlo Localization using laser range finders"

Monte Carlo SLAM

29



Thrun, Sebastian. "Animation: Online mapping with Monte Carlo Localization."

Mapping Methods

30

	Kalman	EM	Occupancy Grids	Dogma
Representation	Landmark Locations	Point Obstacles	Occupancy Grids	Occupancy Grids
Incremental	YES	NO	YES	NO
Requires Poses	NO	NO	YES	YES
Handles Correspondence	NO	YES	YES	YES
Dynamic Environments	limited	NO	limited	YES

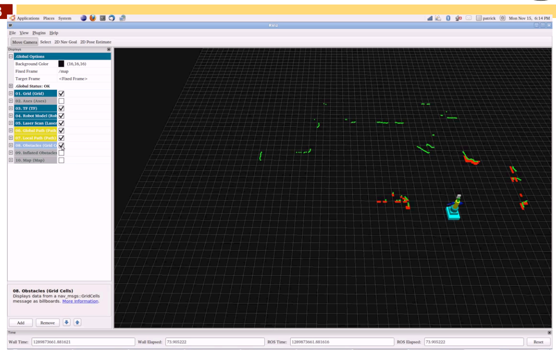
Kalman Filter Performance

32

- Pros:
  - Full (Gaussian) posterior probabilities
  - Incremental
  - Good convergence
- Cons:
  - Limited model
  - Correspondence problem
  - Limited map size
- Many improvements exist!

Particle Filter SLAM

33



https://www.youtube.com/watch?v=khSrWtB0Xik

Expectation Maximization (EM)

35

- Find most likely map (and poses)

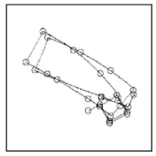
1. Expectation step:
   - Calculate probabilities of robot poses for current guess of map
2. Maximization step:
   - Calculate single most likely map for distribution of robot poses

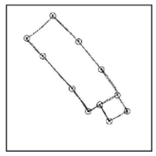
- Iterate

EM Performance

36

- Pros:
  - Resolves correspondences
- Cons:
  - Non-incremental
  - No posterior probabilities for map
  - Slow
  - Greedy
- Improvements: Hybrid approaches
  - Incremental computation
  - Maintain a few possible robot poses





Occupancy Grids

37

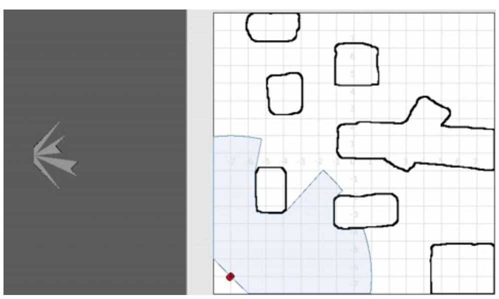
- Impose grid on space to be mapped
- Find *inverse sensor model*

$$p(m_{x,y}|z_t,s_t)$$

- Update odds that grid cells are occupied

Occupancy Grid (simple)

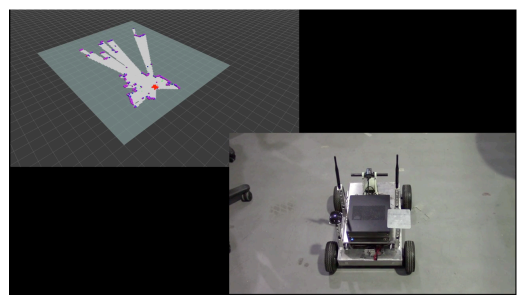
38



https://www.youtube.com/watch?v=zOj4s9TEmg8

Occupancy Grid (real)

39



https://www.youtube.com/watch?v=iD47JWVqTCk

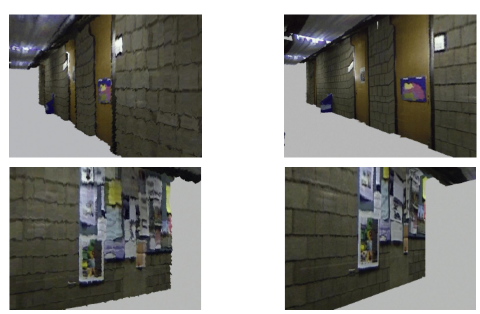
Occupancy Grids

42

- Pros
  - Simple
  - Accurate
  - Incremental
- Cons
  - Require known poses
  - Independence assumptions
- Extensions: Object Maps
  - Reduced memory requirements
  - Better for dynamic environments
  - Limited by available object models

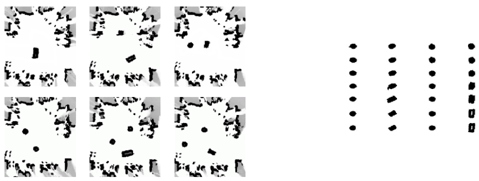
Object Maps

43



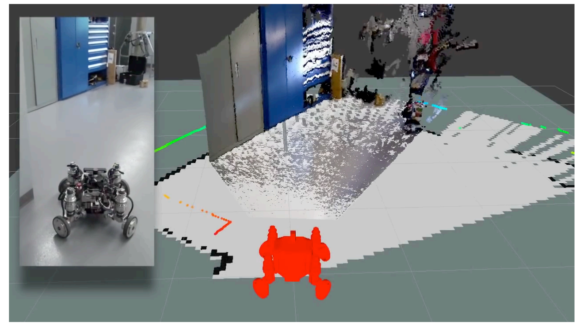
Dynamic Environments

- Kalman filters
- Decaying occupancy grids
- Dogma
  - Dynamic occupancy grid mapping algorithm



A Real Example

Using a Laser and an RGB-D Sensor



https://www.youtube.com/watch?v=o1GSQanY-Do

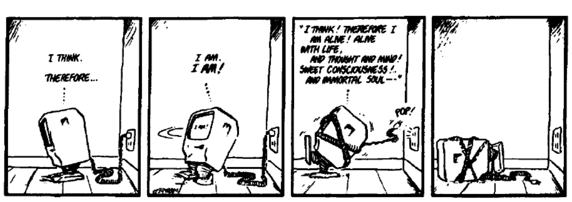
Mapping

- Turning percepts (sensor readings) into a model of an environment (a map)
  - Maps come in many forms
  - Must have sufficient information for navigation tasks
  - Estimating position, path planning, obstacle avoidance, …
- Many challenges
  - Difficult environments: cycles, dynamism, …
  - Sensor noise and precision
  - Actuator noise
  - Labeling environment

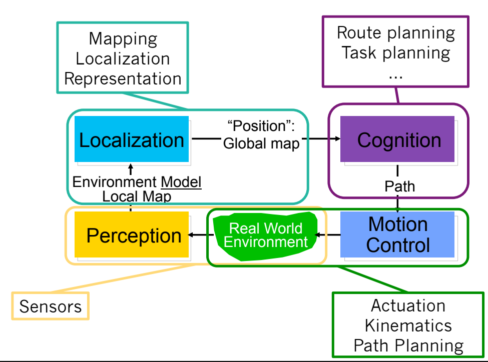
Approaches

- Many algorithms exist
- All modern approaches are probabilistic
- Many are particle filter based
- Usually SLAM is involved
  - Doesn't strictly have to be
- Different approaches address different challenges

Cognition and Control



Cognition and Mobility



What Is Autonomy?

50

- Autonomous robots…
  - Perform their tasks in the world by themselves
  - Do not require human control / intervention
  - Learn about their environment and tasks
  - Avoid damage (to themselves, people, property)
  - Adapt to changing situations
  - Make and execute decisions
  - Possess some degree of self-sufficiency
- Intelligently and safely perform tasks
- Without direct human control

All of these?

Artificial Intelligence

51

- Key types
  - Strong AI: mental/thought capabilities equal to (or better than) human
  - Weak (bounded) AI: intelligent actions or reasoning in some limited situations

- These are problematic
  - How do we measure it?
  - What's an 'intelligent action'?
    - In practice, 'previously human only'
  - Is there something ineffable missing?

- How does it change when it's a robot?

Autonomous Task Performance

52

- Many subtasks
  - Understanding and modeling of the **mechanism**
    - Kinematics, dynamics, odometry
    - Reliable control of actuators
  - Understanding and modeling the **environment**
    - Integration of sensors
  - Understanding and modeling the **task**
    - Generation of task-specific motions
    - Creation of flexible control policies, new situations
- Coping with noise and uncertainty
- Probably physical tasks!

Intelligent Action Needs…

53

- Knowledge Representation
- Search
- Planning
- Learning
- Inference

DARPA Grand Challenge 1

54

DGC 1 Challenges

55

- Localization
  - But not mapping
- Sensor management
  - What sensors?
  - Where's the road?
- Narrow pass
  - Switchbacks, turns
  - Tunnels
- Actuator management

✓ Knowledge Representation
✓ Search
✓ Planning
✗ Learning
✗ Inference

DARPA Grand Challenge 2

56

DGC 2 Challenges

57

- All of the above, plus…
- Visual parsing of traffic elements
  - Sensing + Knowledge
- Awareness of other cars
  - Sensing + Planning
- Non-3D-guided tasks
- Faster speeds
- Safety

✓ Knowledge Representation
✓ Search
✓ Planning
✗ Learning
✓ Inference

Google Self-Driving Car

58

Google Challenges

59

- All of the above, plus…
- Sublimely oblivious drivers
- Full-speed actuator management
- Legal management
- Ethical management

✓ Knowledge Representation
✓ Search
✓ Planning
? Learning
✓ Inference

DARPA Robot Challenge 1

60

DRC 1 Challenges

61

- All of the above, plus…
- Non-designed environment
  - Balancing
  - (Much) harder actuation
  - Bipedal motion
- Yet more sensor hassles
- Weight, power
- Manipulation
  - !!!

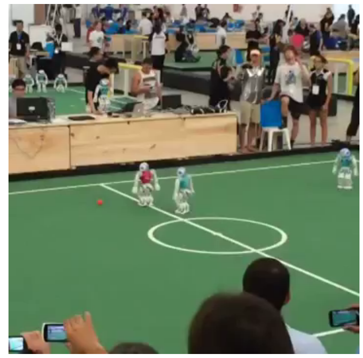
✓ Knowledge Representation
✓ Search
✓ Planning
✓ Learning
✓ Inference
✗ Autonomy

DRC 1

62

Robocup

63



Robocup

64

- All of the above, plus…
- Object interaction
  - Balls, walls, ...
- Deliberately difficult goal task
  - Streets designed for easy driving
  - S&R not designed
- Enormous robot design space
- Antagonistic agents

✓ Knowledge Representation
✓ Search
✓ Planning
✓ Learning
✓ Inference
✓ Autonomy

Other Robot Tasks

Intelligent Action Needs…

66

- Knowledge Representation
- Search
- Planning
- Learning
- Inference

Other Robot Tasks

67

UC
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND