Autonomous Robotic Systems

CS 685

Jana Kosecka, 4444 Research II
kosecka@gmu.edu , 3-1876
Office hours Tue 2-3pm

1

Logistics

- Grading: Homeworks + Project 65% Exam: 35%
- Prerequisites: basic statistical concepts, geometry, linear algebra, calculus, CS 480, CS 580
- Course web page

https://cs.gmu.edu/~kosecka/cs685/

- Homeworks about every 2 weeks, Midterm, Final Project
- Choose from the list of projects, suggest your own
- Implement one of the covered methods on robot/robot simulator, come up with new ideas of robotics tasks
- Write a report and prepare the final presentation

2

Recommended Text

- R. Siegwart and I. Nourbakhsh: Introduction to Autonomous Mobile Robots, MIT Press, 2004
- [1] S. LaValle: Planning Algorithms, Cambridge Press, http://planning.cs.uiuc.edu/
- [2] S. Thrun, W. Burghart, D. Fox: Probabilistic Robotics http://robots.stanford.edu/probabilistic-robotics/
- [4] S. Russell and P. Norvig: Artificial Intelligence: A Modern Approach
- [5] R. Sutton and A. G. Barto: Introduction to Reinforcement Learning (on-line materials see course www)

3

Overview of the topics

- Kinematics, Kinematic Chains, Mobile Robot kinematics
- Notion of state, sensing state, elementary control
- Motion planning, Graph Based Methods, Potential Field Based methods, Sampling Based Methods, Configurations Space

- Robot Perception – Image Features, Stereo, Motion Estimation and 3D reconstruction, Object Detection, Semantic Segmentation

- Foundations of Probabilistic Robotics
- State estimation and Tracking
- Localization using Particle Filters
- Simultaneous Localization and Mapping using vision and RGB-D data

- Dynamic Programming and Markov Decision Processes
- Learning how to act – Reinforcement Learning

4

Course Logistics

- Required language Python
- Robot simulators, real robots

- Availability of robotics platforms
- Pioneers with range sensors, cameras

- Turtlebot Pyrobot open source robotics platform http://pyrobot.com
- Humanoid – Small soccer league
- Simulators – AI Habitat   AI-Thor https://ai2thor.allenai.org/
- CARLA Autonomous Driving Simulator https://carla.org/
- List of resources for mobile robotics http://www.mobilerobots.org//

- Possibilities of programming real robots equipped with range sensors, RGB-D cameras
- Current trends and areas of robotic technologies

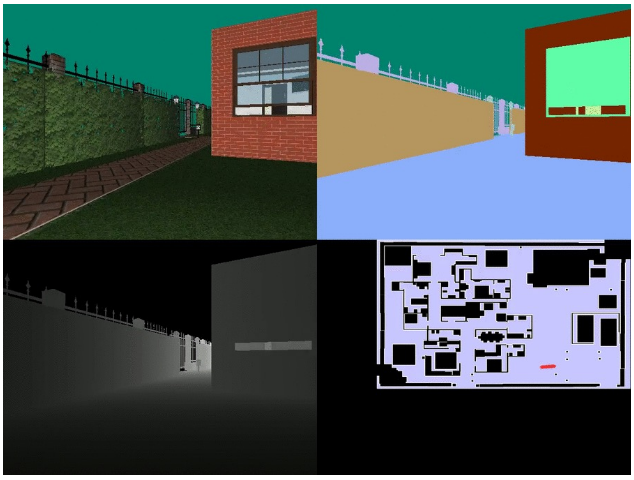
5

Embodied AI

Gibson Environment

Habitat Challenge

6



- House 3D (Facebook Research)
- Rich simulated environments
- Navigation, Perception, Visual Question Answering

7

Applications History - Robots in manufacturing/material handling

Manhattan project (1942) – handling and processing of radioactive materials – Telemanipulation

Manufacturing
- storage, transport delivery
- table top tasks, material sorting, part feeding – conveyor belt
- microelectronics, packaging
- harbor transportation
- construction (automatic cranes)

Suitable for hard repetitive tasks – heavy handling or fine positioning
Successful in restricted environments, limited sensing is sufficient – limited autonomy

Autonomous Robotic Systems
AGV's - automated guided vehicles
AUV's - automated unmanned vehicles

8

Applications History - Space Robotics

50-ties US space program, exploration of planets, collecting samples
Astronauts bulky space suits – difficult
NASA, JPL, DARPA – sponsoring agencies
Space programs, military application – surveillance, assistance
Planetary Rovers – initially controlled by humans
- large time delays,
- poor communication connections

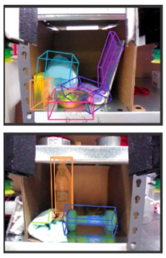
Need for (semi) – autonomy

Teleoperation – Mars Rover

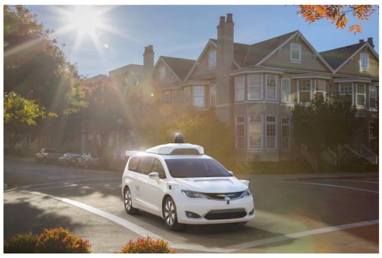
Human operator controls the robot
Local site – human views the sensory data, sends the commands
Remote site – sensors acquire the information

9

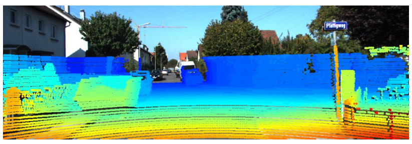
Dexnet

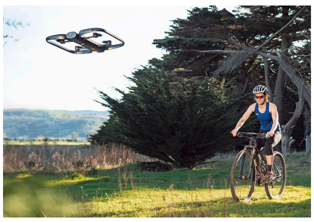


Amazon Picking Challenge



Waymo





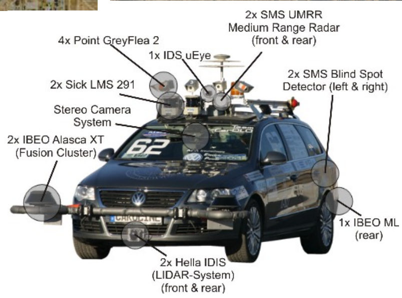
SKydio

Google Arm farm

10

Grand Challenges

DARPA

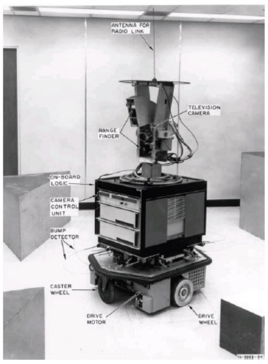


2004, 2005, 2007

- Lasers, camera, radar, GPS, compass, antenna, IMU,
- Steer by wire system, PC's with Ethernet for processing information from sensors

11

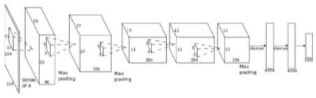
Shakey the Robot



12

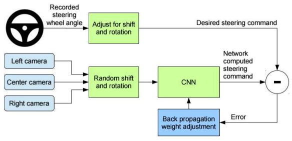
Learn policies

o_t



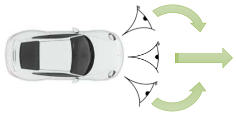
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

a_t



Supervised learning paradigm
training data o_t , a_t

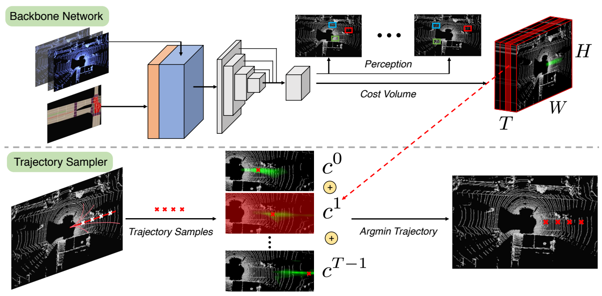
Learn the policy $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$



Bojarski '16 NVIDIA End to End Learning for Self-Driving Cars

13

End-to-end interpretable trainable motion planner



Optimizing perception, motion planning and control jointly
integrating map data, predictions of the object detectors

W. Zheng, W. Luo, S. Sua R. Urtasun et al.
End-to-end interpretable neural planner, CVPR 2019

14

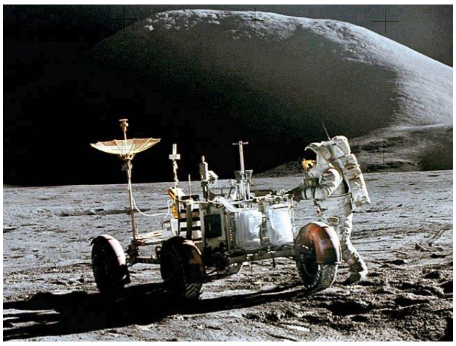
Example 1:Building Virtual Models of Mars

Example of stereo pipeline, from raw data, preprocessing, meshes, texture maps

See http://schwehr.org/photoRealVR/example.html

15

Apollo



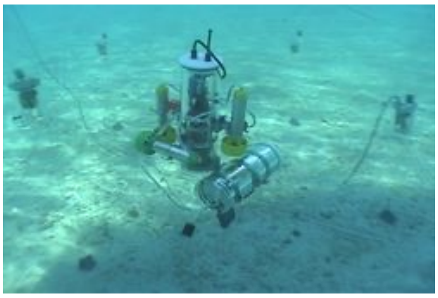
Lunar Rovers

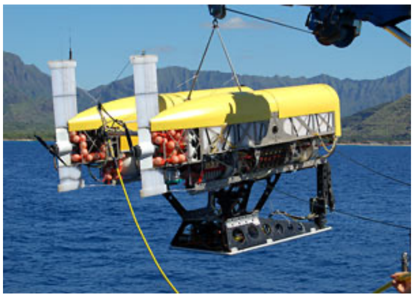
Current NASA Prototype

16

Applications: Underwater robotics

- Sensor network
- Remotely Operated robot for ocean exploration





18

Robots in the service of humans

- Robotic surgery - DaVinci robotic surgery robot – human assisted
- http://www.intuitivesurgical.com/products/da_vinci_video_overview.aspx
- Robotics in rehabilitation surgery (Hocomo Inc)

- Mobile Robots
- courier in buildings and hospitals, vacuum cleaners,

19

Variety of domains and tasks

20

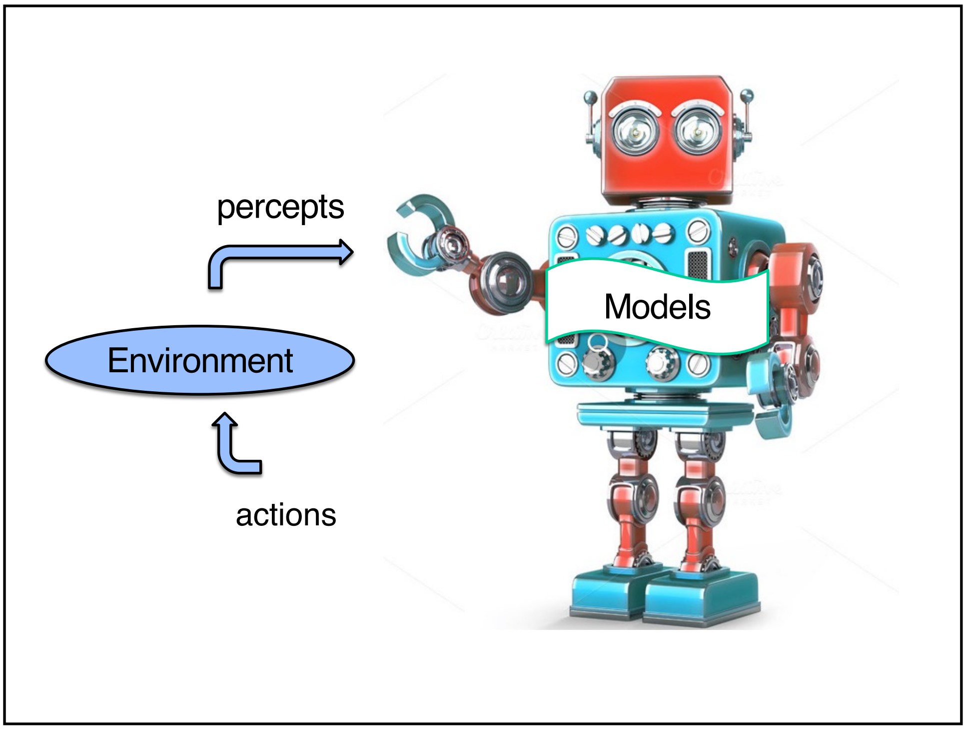
Games and Entertainment

Furbies

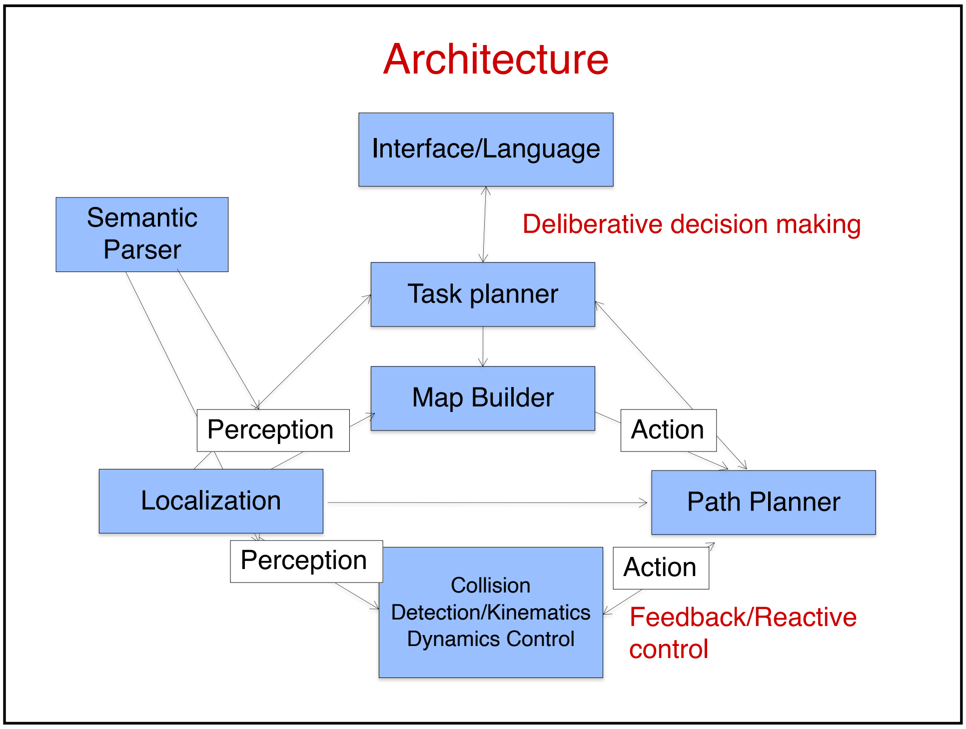
Aibos Latter & Macaron

Aibo soccer league - RoboCup

21

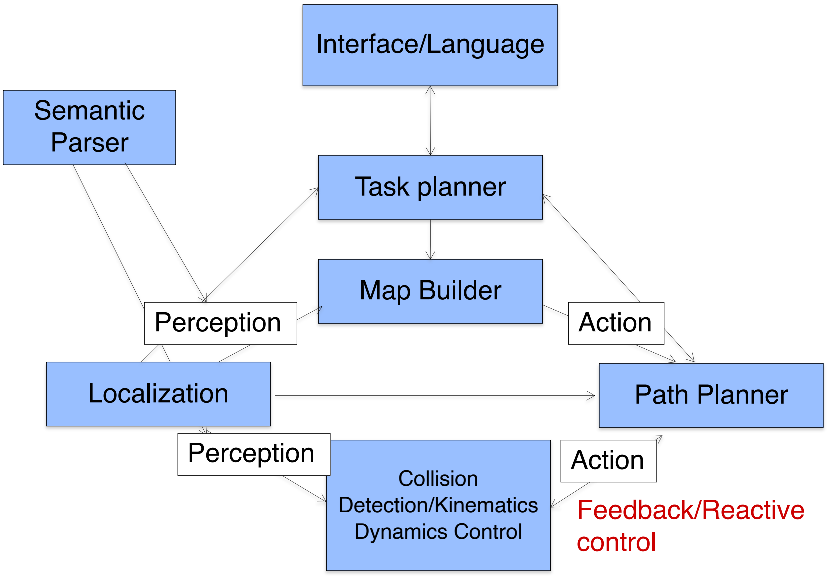

percepts
Environment
actions
Models

25


Architecture
Interface/Language
Semantic Parser
Deliberative decision making
Task planner
Map Builder
Perception
Action
Localization
Path Planner
Perception
Collision Detection/Kinematics Dynamics Control
Action
Feedback/Reactive control

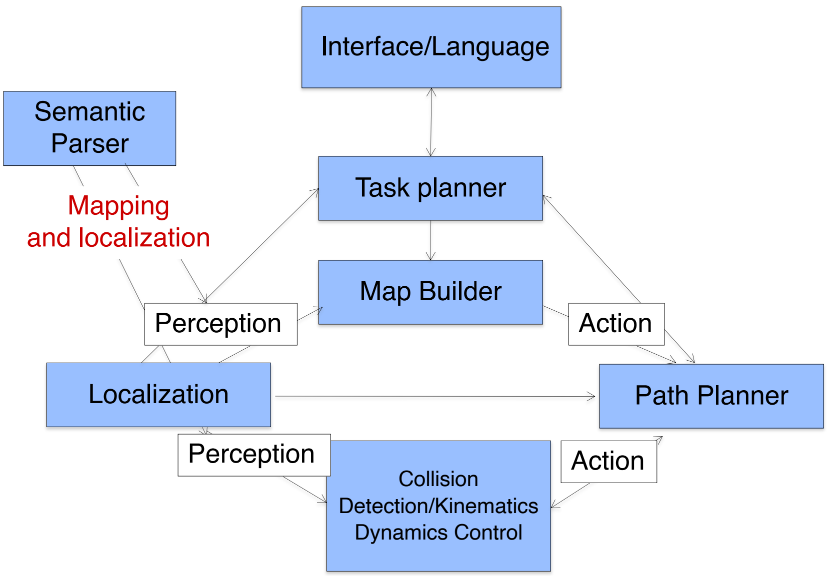
26

Architecture



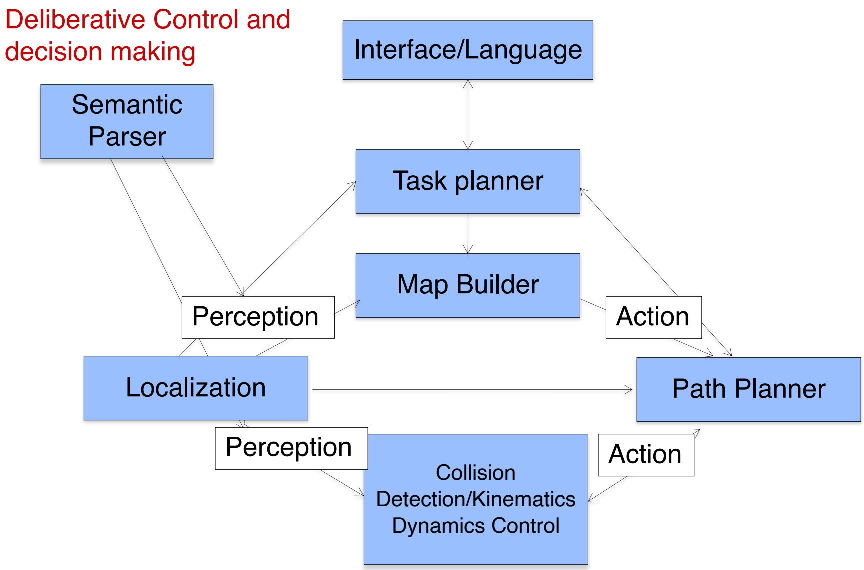
27

Architecture



28

Architecture



29

Robotics and AI

Knowledge representation
how to represent objects, humans, environments
symbol grounding problem

Computer Vision, Pattern Recognition, Perception
recognition, vision and motion, segmentation and grouping
representation

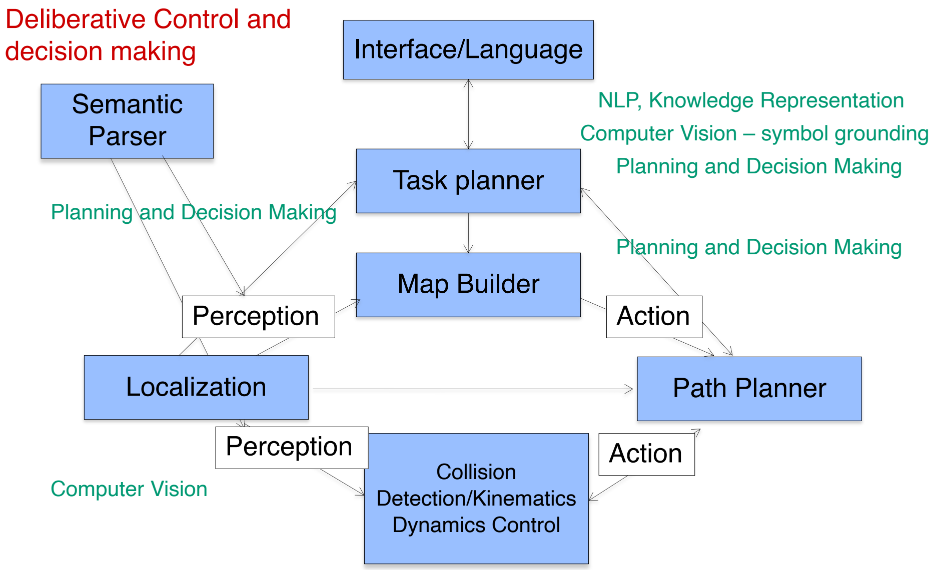
Natural Language Processing
provides better interfaces, symbol grounding problem

Planning and Decision Making
How to make optimal decision, actions give the current knowledge
of the state, currently available actions

Learning in Robotics
Learning to plan, learning to explore, learning to perceive, visual dialog
Learning to grasp, end-to-end learning, modular learning

30

Architecture



31

Autonomous Robotic System

- Three Basic Components of the Robotic System
- SENSE – process information from the sensors
- PLAN – compute the right commands/directives
- ACT – produces actuator commands

- Different organization of these functionalities gives rise to different robot architectures

32

Autonomous Driving

- DARPA Grand Challenge 2005
  2004 CMU vehicle drove 7.36 miles out of 150
  2005 5 teams finished, Stanford won
- DARPA Urban Challenge 2007
  urban environment other vehicles present
  6 teams finished
- Google Self-Driving Car
  by July 2015 1M miles, 14 minor accidents
- Ernst Dickmans / Mercedes Benz 1987
  1758 Km, 60 miles per hour
- Parking maneuvers, overtake maneuvers, skidding

33

Robotic Navigation

- Stanford Stanley Grand Challenge
- Outdoors unstructured env., single vehicle

- Urban Challenge
- Outdoors structured env., mixed traffic, traffic rules

34

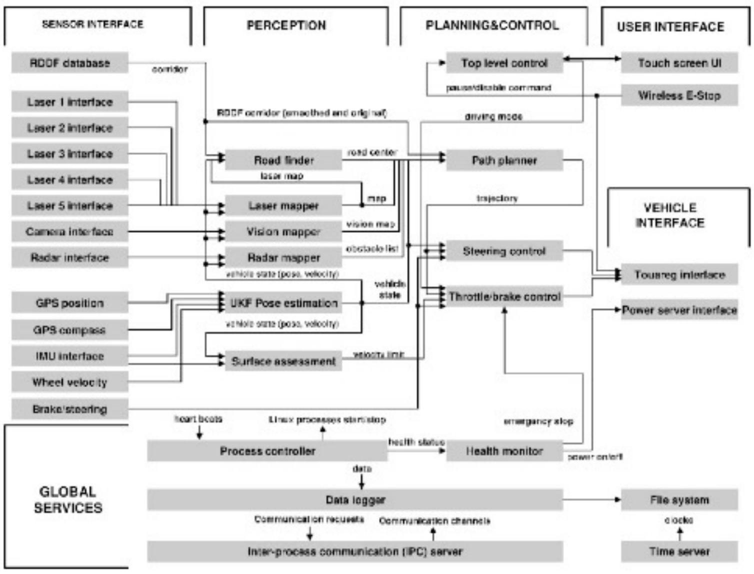
Robot Components (Stanley)

- Sensors
- Actuators-Effectors
- Locomotion System
- Computer system – Architectures – (the brain)

- Lasers, camera, radar, GPS, compass, antenna, IMU,
- Steer by wire system
- Rack of PC's with Ethernet for processing information from sensors

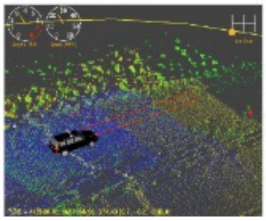
35

Stanley Software System

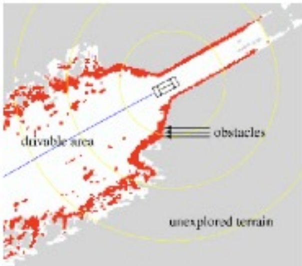


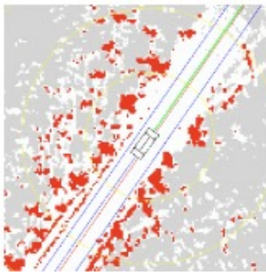
36

- Terrain mapping using lasers



- Determining obstacle course





37

Autonomous Driving

- Reduction of the number of accidents
- 5 million accidents and
- > 30,000 fatalities due to traffic accidents)
- Time recovered due to commuting,
- Improved parking in the cities,
- New models of personal mobility

This Was Supposed to Be the Year Driverless Cars Went Mainstream

Perfecting the technology has taken longer than expected. The coronavirus pandemic has made it even more difficult.

May 2020, New York Times

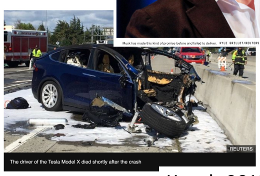
Elon Musk Promises a Really Truly Self-Driving Tesla in 2020

The CEO says his Autopilot system will be "feature-complete" this year, and ready to ferry snoozing passengers by the end of next year.

Musk has made this kind of promise before and failed to deliver. KYLE GRILLOT/REUTERS

Tesla in fatal Califc Autopilot

31 March 2018

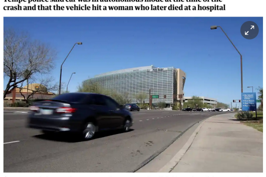


The driver of the Tesla Model X died shortly after the crash

March 2018, BBC News

Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian

Tempe police said car was in autonomous mode at the time of the crash and that the vehicle hit a woman who later died at a hospital



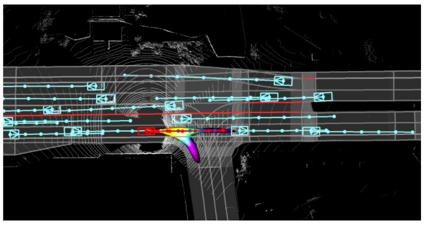
A car passes the location where a woman pedestrian was struck and killed by an Uber self-driving sport utility vehicle in Tempe, Arizona, on Monday. Photograph: Rick Scuteri/Reuters

March 2018, Guardian

38

Mapping, Control, Planning for autonomous driving

- Navigation strategies
- trajectory following, planning, (overtake, lane change)



Data Driven, Machine Learning Techniques

39

Autonomous Helicopter Flight

[Abbeel, Coates & Ng]

Kalman filtering, model-predictive control, LQR, system ID, trajectory learning

Slide courtesy P. Abbeel

40

Four-legged locomotion

[Kolter, Abbeel & Ng]

value iteration, receding horizon control, motion planning, inverse reinforcement learning, nolearning, learned

Slide courtesy P. Abbeel

41

Mobile Manipulation

[Maitin-Shepard, Cusumano-Towner, Lei, Abbeel, 2010]

localization, motion planning for navigation and grasping, grasp point selection, visual recognition

Slide courtesy P. Abbeel

42

Visuomotor Learning

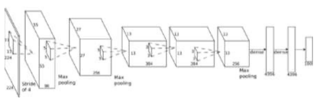
[Levine*, Finn*, Darrell, Abbeel, 2015]

Slide courtesy P. Abbeel

43

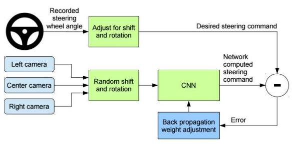
Learn policies

o_t



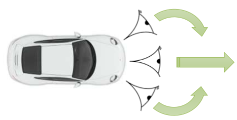
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

a_t



Supervised learning paradigm
training data o_t a_t

Learn the policy $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

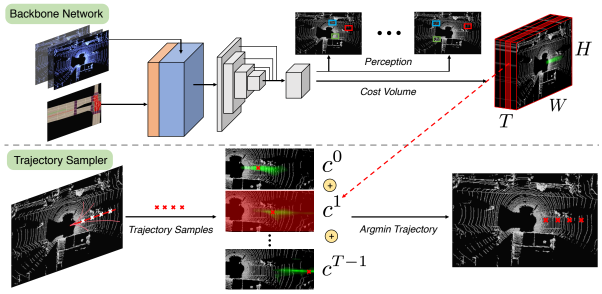


Bojarski ‘16 NVIDIA End to End Learning for Self-Driving Cars

4

44

End-to-end interpretable trainable motion planner

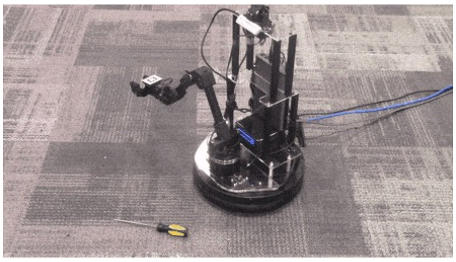


optimizing perception, motion planning and control jointly
integrating map data, predictions of the object detectors

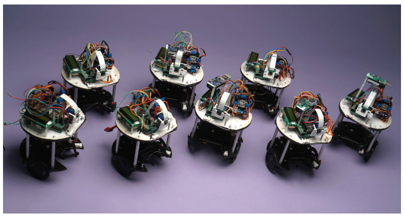
W. Zheng, W. Luo, S. Sua R. Urtasun et al.
End-to-end interpretable neural planner, CVPR 2019

45

Robots @ GMU

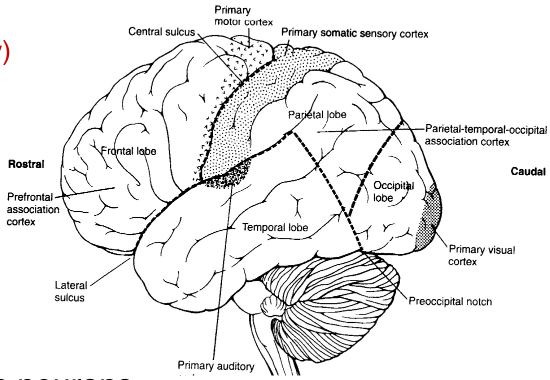


Pioneer, Pybot, Flockbots, RoboPatriots



49

The Brain (analogy)



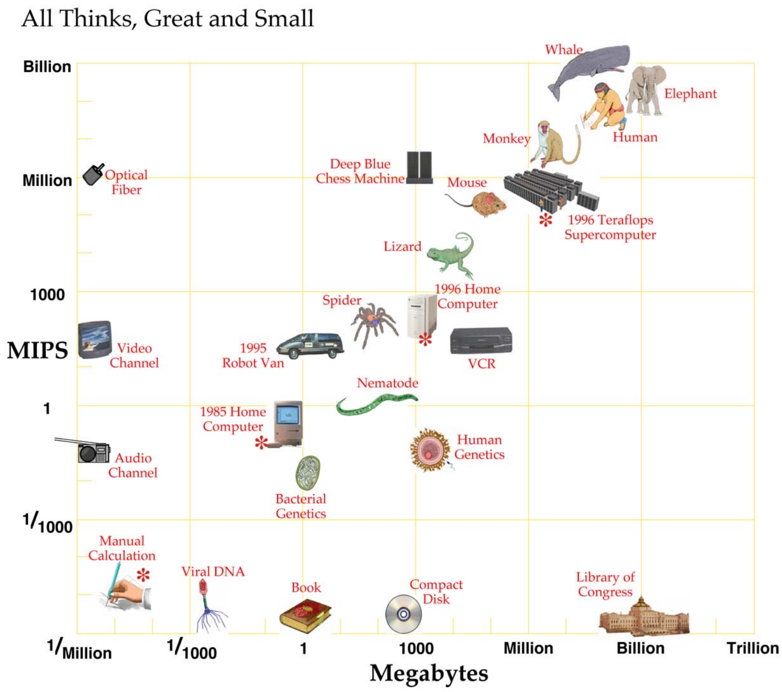
- 100 Billion neurons
- On average, connected to 1 K others
- Neurons are slow. Firing rates < 100 Hz.
- Can be classified into
  - Sensory – vision, somatic, audition, chemical
  - Motor – locomotion, manipulation, speech
  - Central – reasoning and problem solving

50

Trends in biological and machine evolution
Hans Moravec: Robot

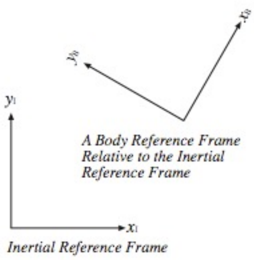
- 1 neuron = 1000 instructions/sec
- 1 synapse = 1 byte of information
- Human brain then processes 10^14 IPS and has 10^14 bytes of storage
- In 2000, we have 10^9 IPS and 10^9 bytes on a desktop machine
- In 25 years, assuming Moore's law we obtain human level computing power

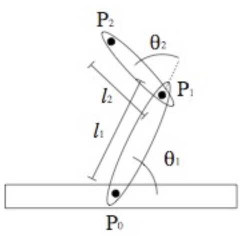
51

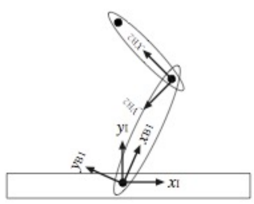


52

Course Overview – PART I
Modeling Geometric transformation







- Modeling Rigid Body Motion
- Modeling Kinematic Chains

53

Modelling motion

- Notion of state, state evolution
- Systems view vector $\mathbf{x}$ denotes the state of the system, vector $\mathbf{u}$ types of controls/actions the system can carry out we will discuss ways of characterizing the motion of the system

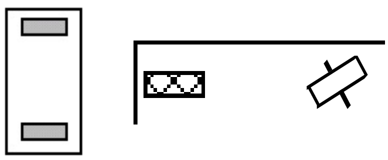
$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$$

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$$

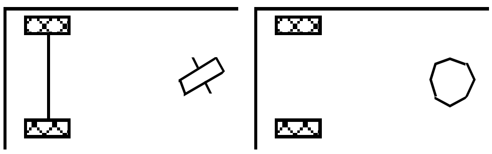
54

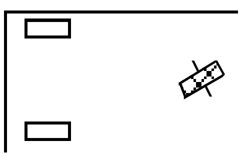
Mobile Robot Kinematics

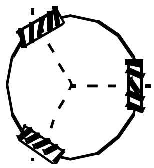
- Two wheels



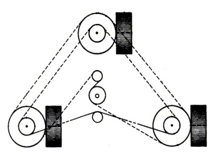
- Three wheels







Omnidirectional Drive

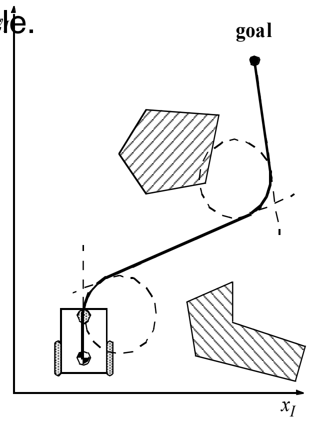


Synchro Drive

55

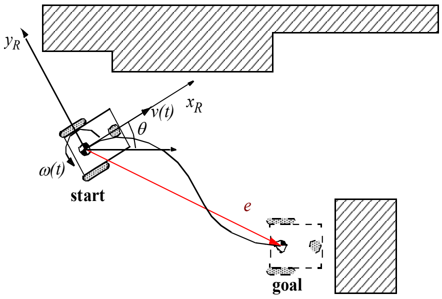
Motion Control: Open Loop Control

- trajectory (path) divided in motion segments of clearly defined shape:
  - straight lines and segments of a circle.
- control problem:
  - pre-compute a smooth trajectory based on line and circle segments



56

Motion Control: Feedback Control, Problem Statement



- Find a control matrix K, if exists

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$

- with $k_{ij}=k(t,e)$
- such that the control of v(t) and ω(t)

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \cdot {}^{R}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- drives the error e to zero.

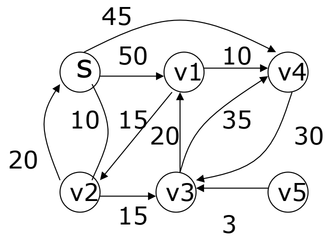
$$\lim_{t \to \infty} e(t) = 0$$

57

Motion Planning: Graph Based Methods

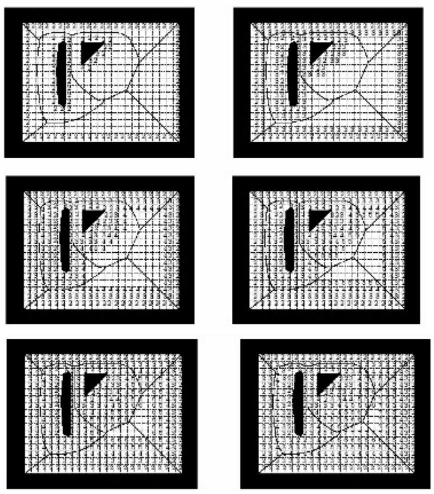
Single shortest path – single destination t (single source)
Given pair of vertices – what is the shortest path from
u to v

Example:



58

Motion Planning: Grassfire algorithm



- Discretize the space

Create a queue Q of all pixels at the boundary of obstacles
For each, set the boundary to 1
And the free space to 0.

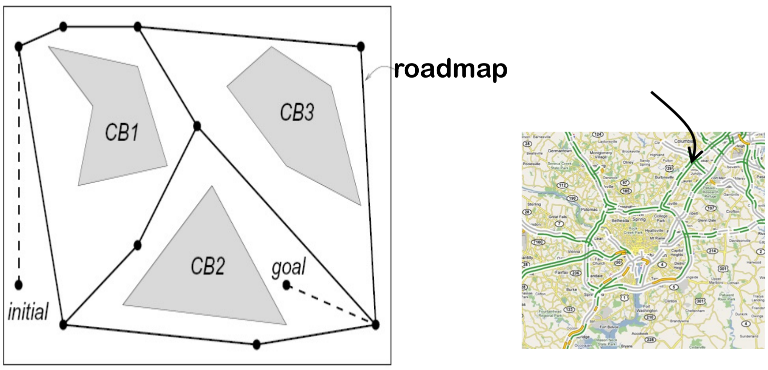
- For each element in the Q
- If d(q) = 0 set d(q) = 1+min d(q') of the neighbours which differ from 0
- Add all neighbours to the Q with d(q) = 0

Resulting map – distance to the nearest obstacle

59

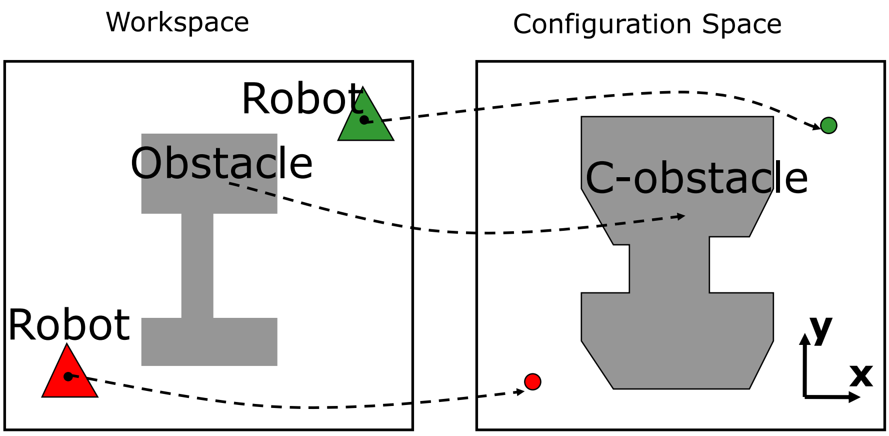
Motion Planning: Roadmap Methods

Capture the connectivity of C_{free} with a roadmap (graph or network) of one-dimensional curves



60

Configuration Space



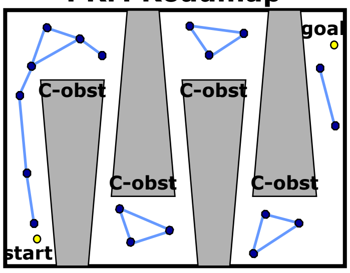
- C-obstacle is a polygon.

61

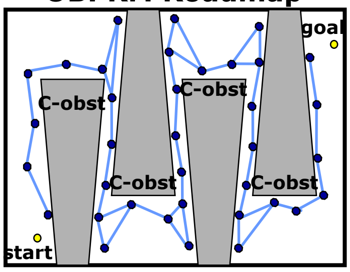
Probabilistic Roadmap Methods

To Navigate Narrow Passages we must sample in them
• most PRM nodes are where planning is easy (not needed)

PRM Roadmap



OBPRM Roadmap



Idea: Can we sample nodes near C-obstacle surfaces?

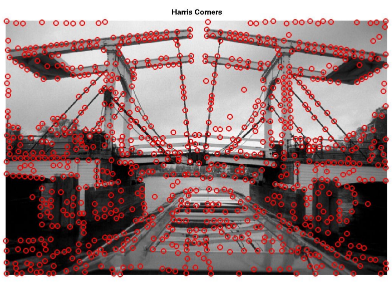
- we cannot explicitly construct the C-obstacles...
- we do have models of the (workspace) obstacles...

62

Robot Perception: Feature Matching

Original image

Strong + connected weak edges

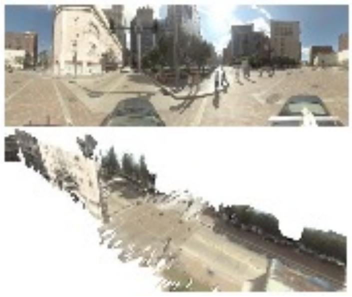


Interest points

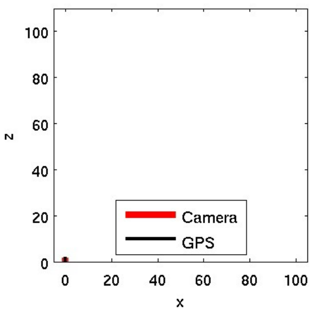
63

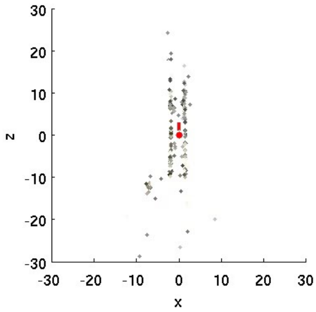
Perception: Mapping and localization

- Visual odometry
- 3D reconstruction



64





65

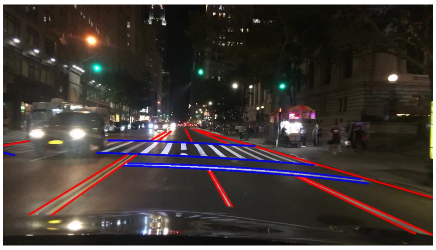
Perception: 3D mapping

http://www.cs.unc.edu/Research/urbanscape

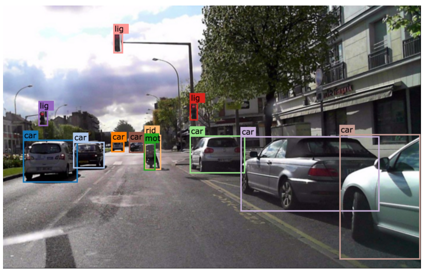
66

Perception for Autonomous Driving

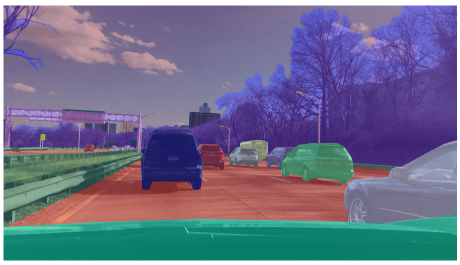
Drivable Areas



Lane Markings



Car detections

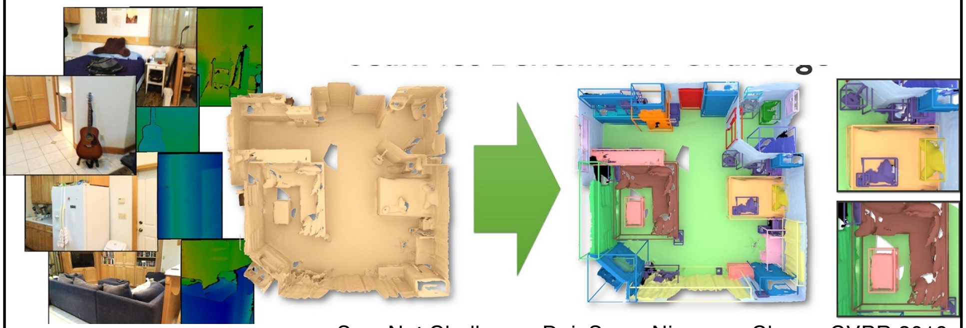


Semantic Segmentation

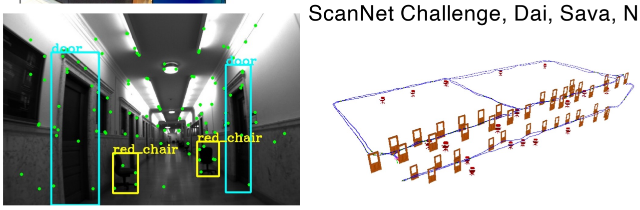
68

68

Perception - Spatial environment representations

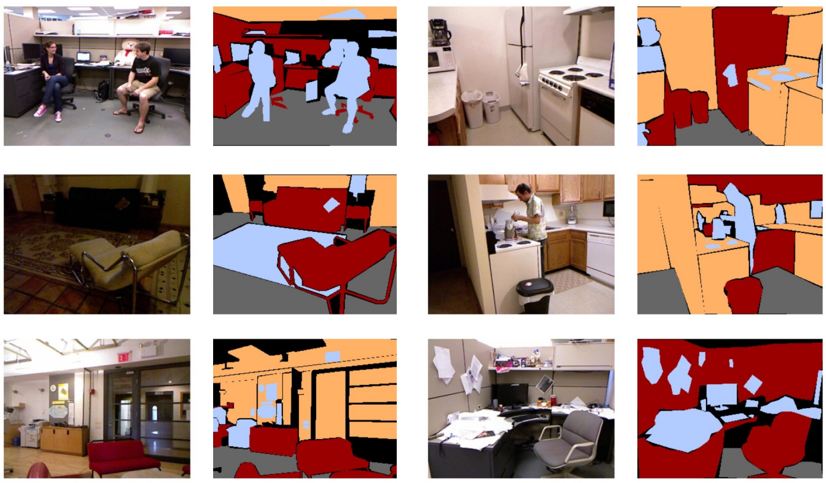


ScanNet Challenge, Dai, Sava, Niessner, Chang, CVPR 2019



Localization from Semantic Observations via the Matrix Permanent N. Atanasov, M. Zhu, K. Daniilidis and G. Pappas

Semantic Segmentation



Probabilistic Robotics

- Taking into account uncertainty of sensors and actions
- Localization in the presence of uncertainty,
- Map building

Robot Perception

- How to process information from sensors
- Visual Sensing
- Range Sensing

- MDP's
- POMDP's

71

Markov Localization :

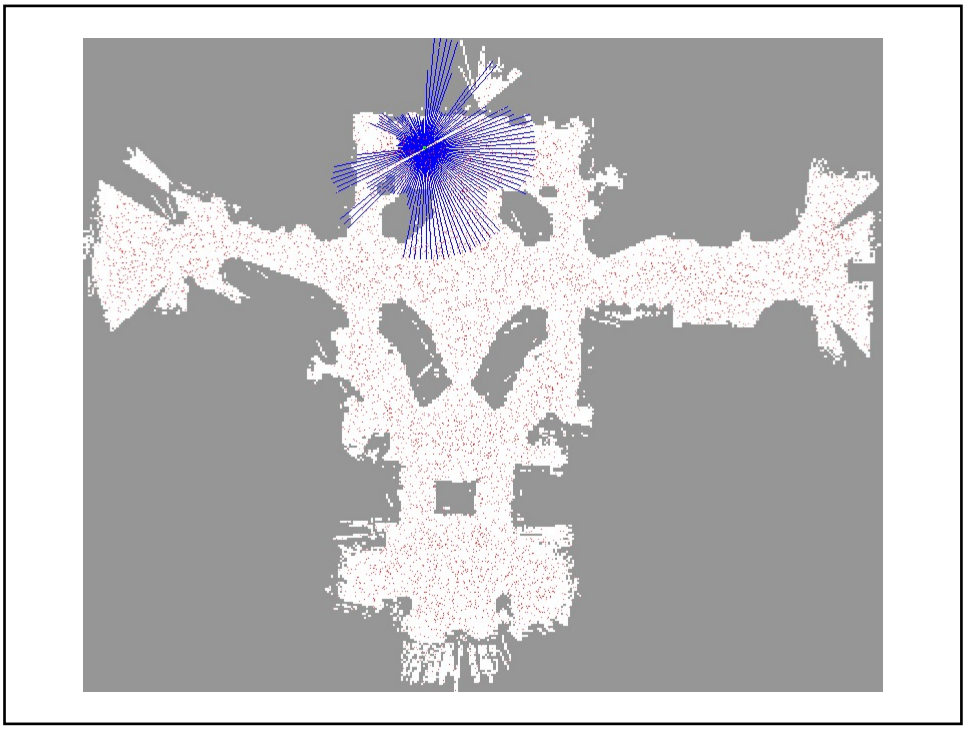
Applying probability theory to robot localization

- Bayes rule: $$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$
  - Map from a belief state and a action to new belief state (ACT):

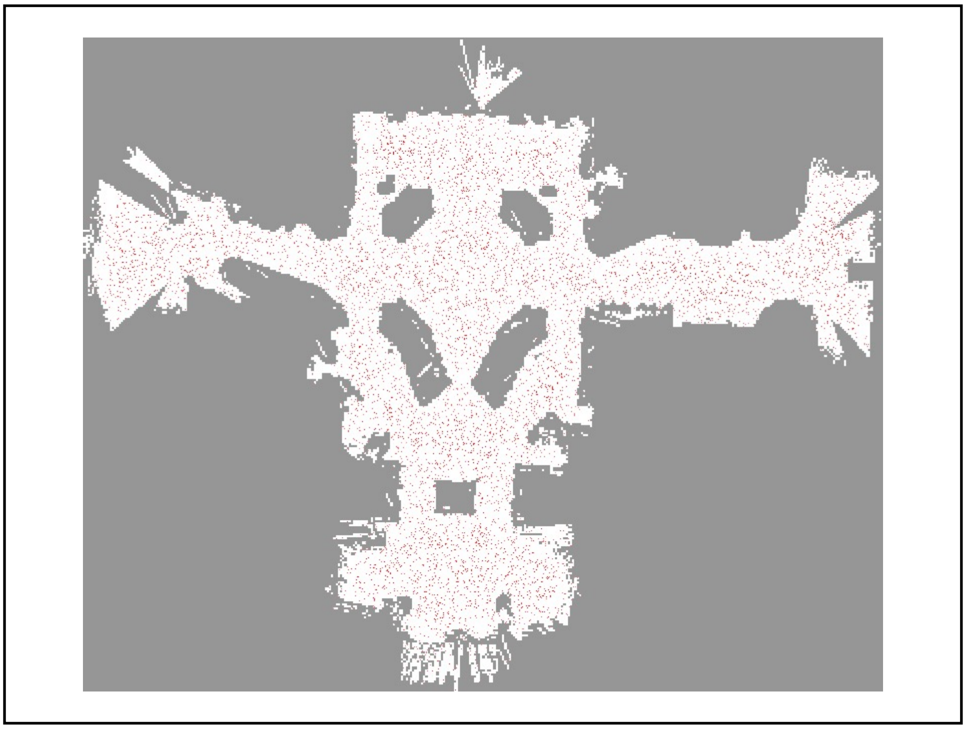
$$p(l_t|o_t) = \int p(l_t|l'_{t-1}, o_t)p(l'_{t-1})dl'_{t-1}$$

  - Summing over all possible ways in which the robot may have reached l.
- Markov assumption: Update only depends on previous state and its most recent actions and perception.

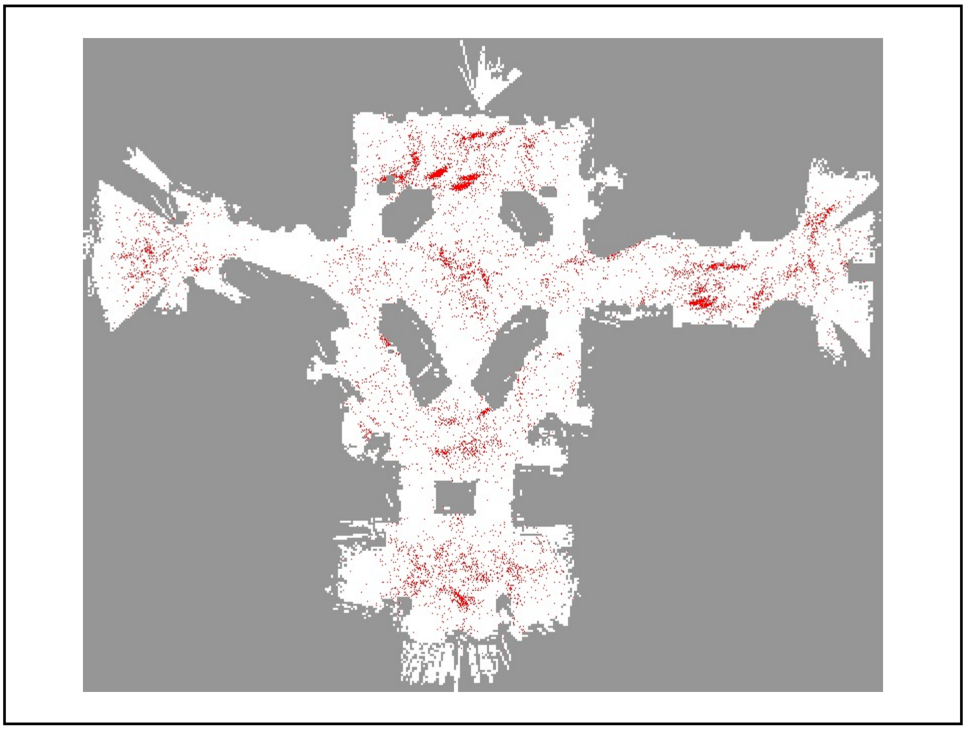
75



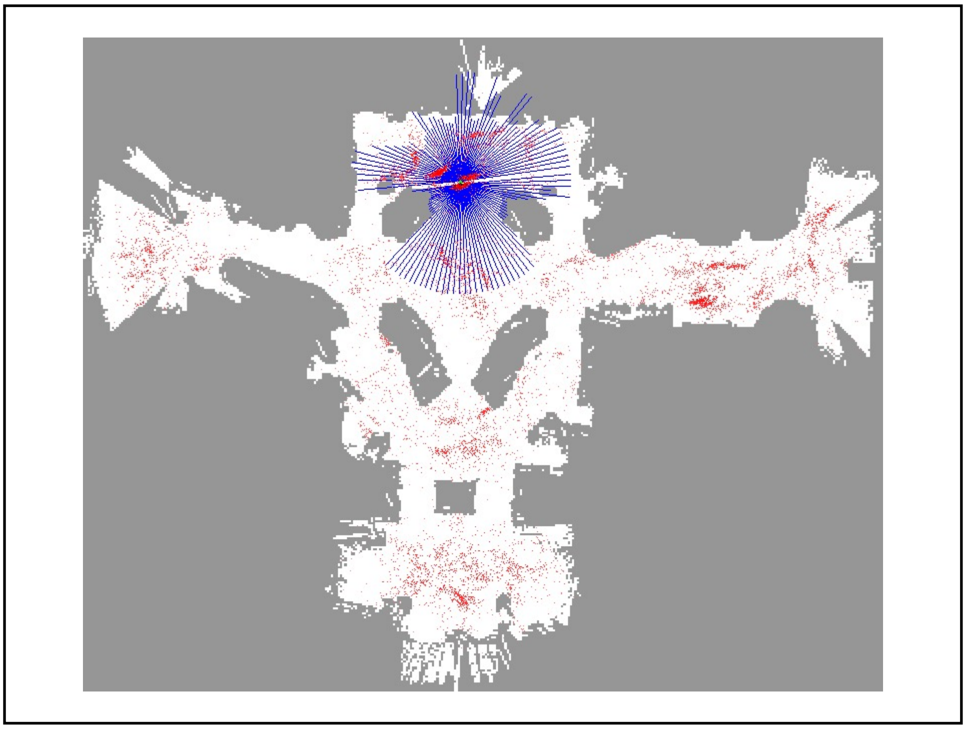
77



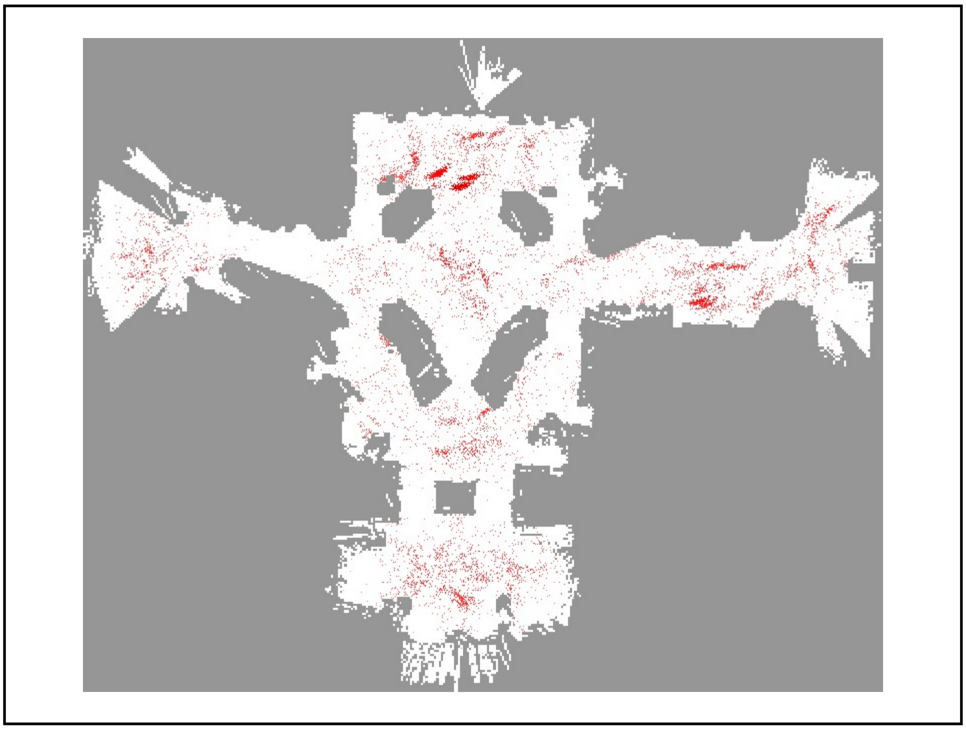
78



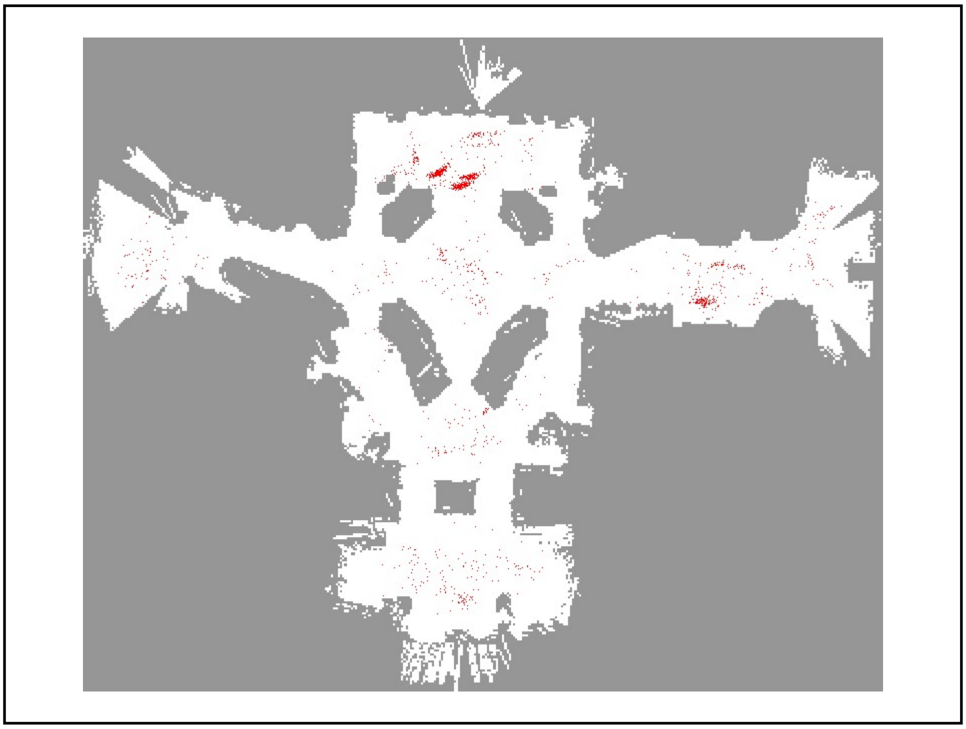
79



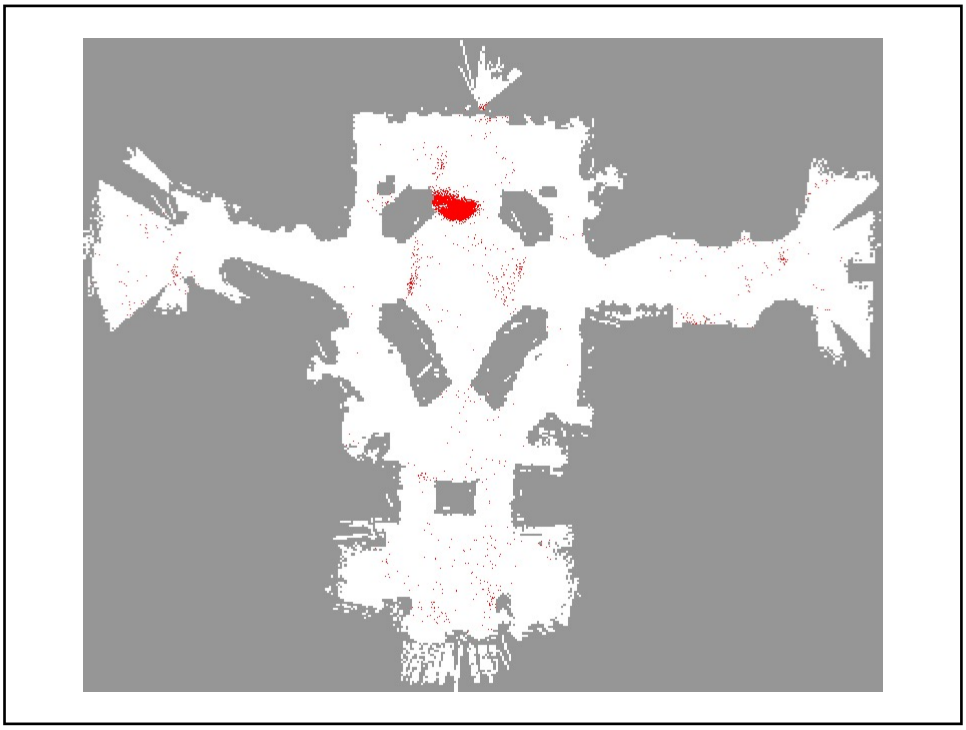
80



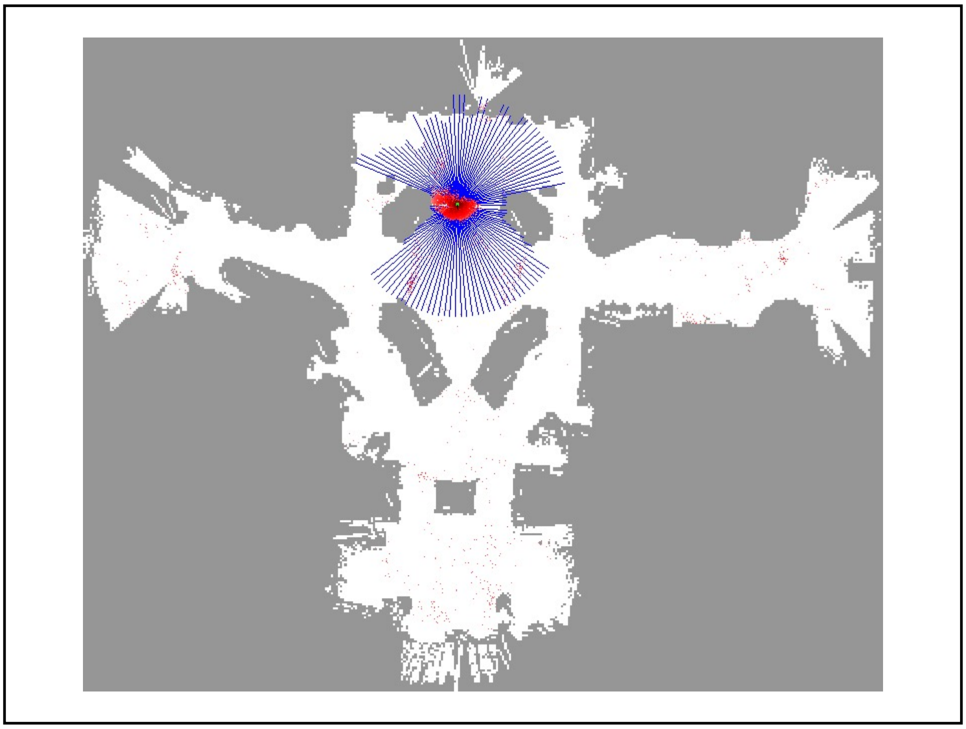
81



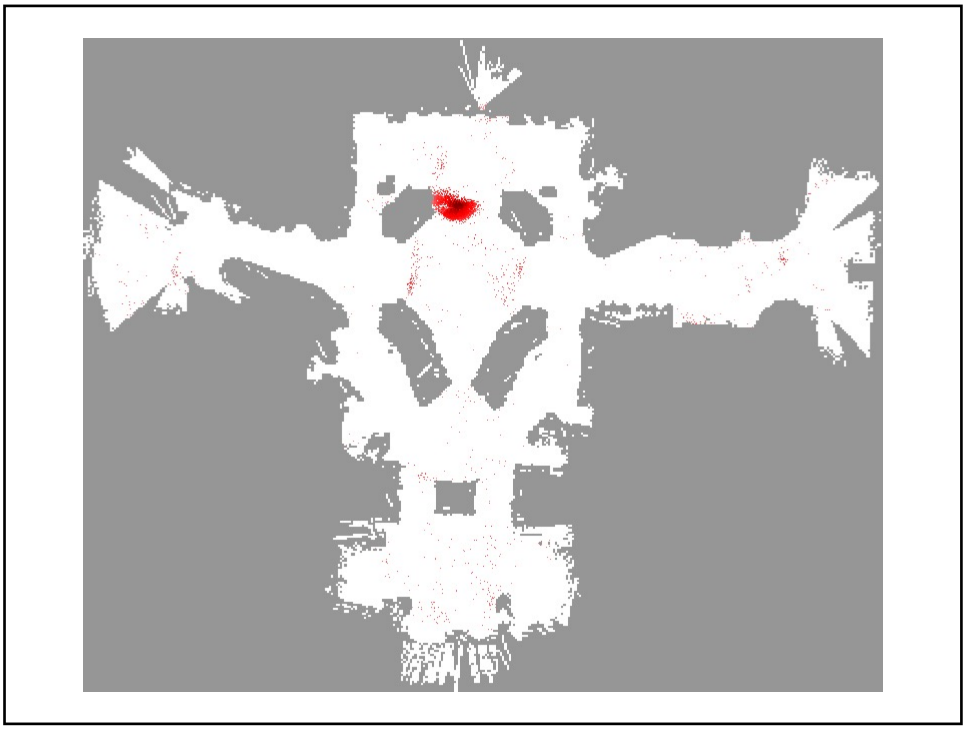
82



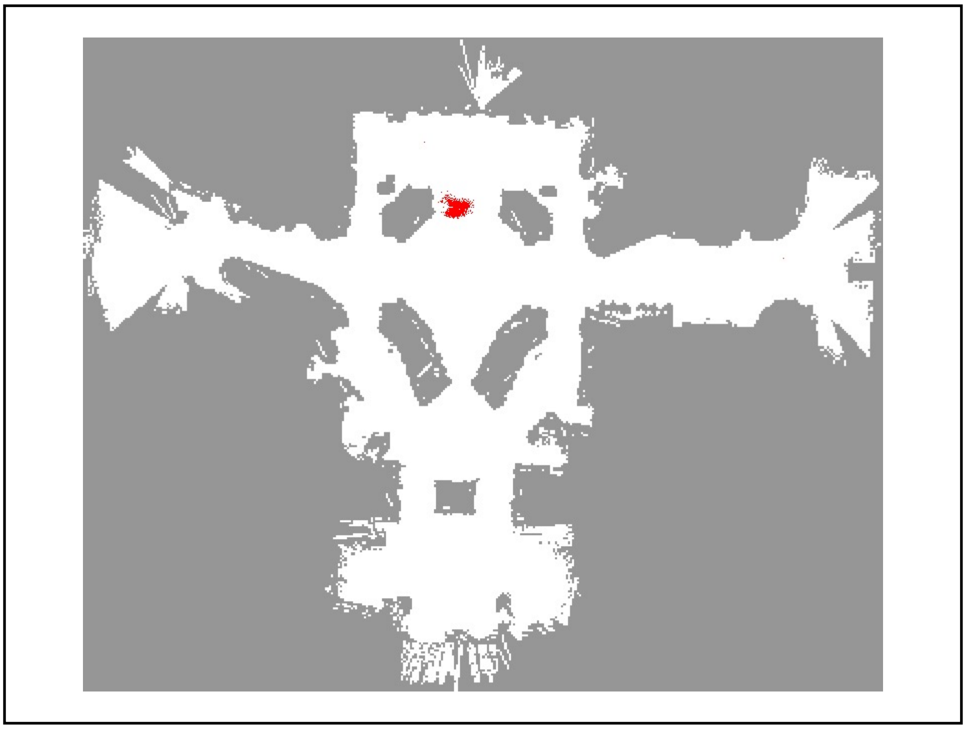
83



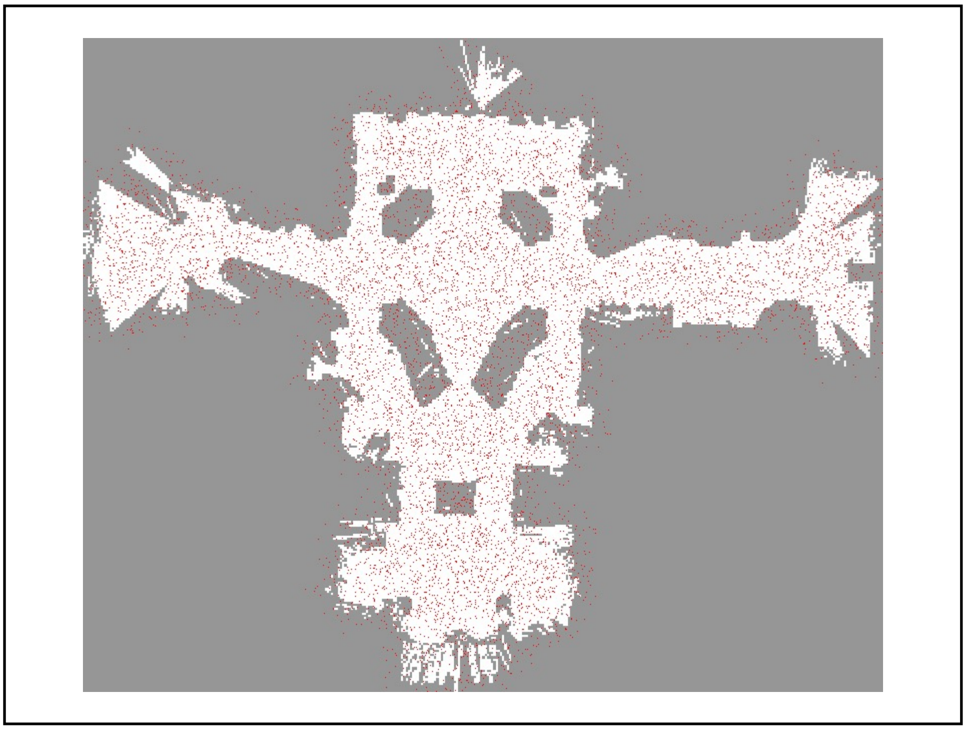
84



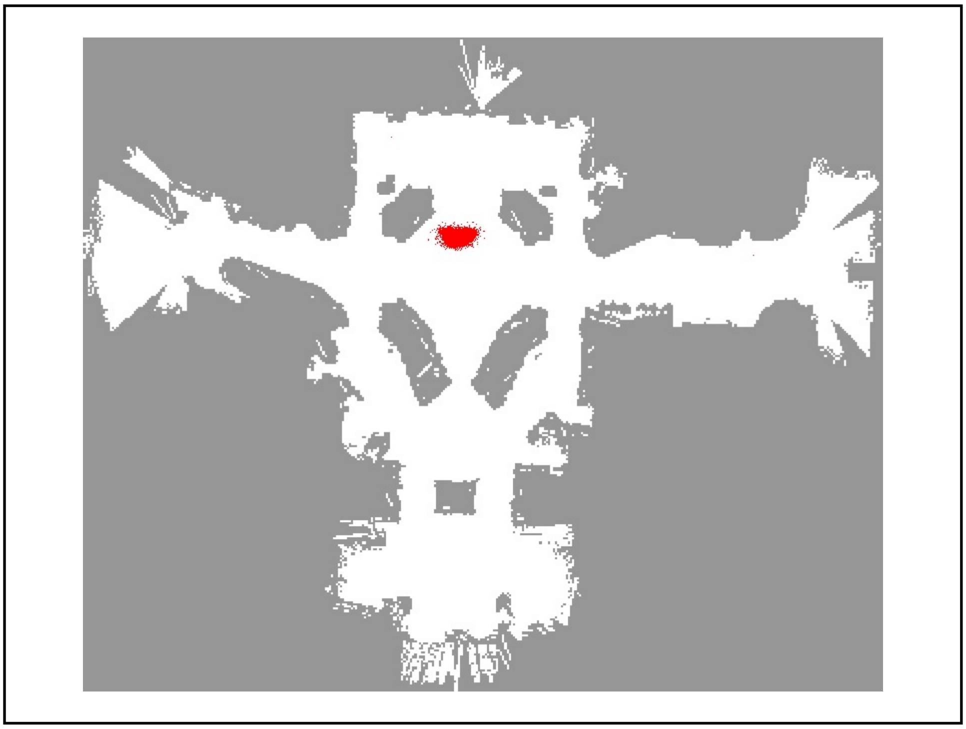
85



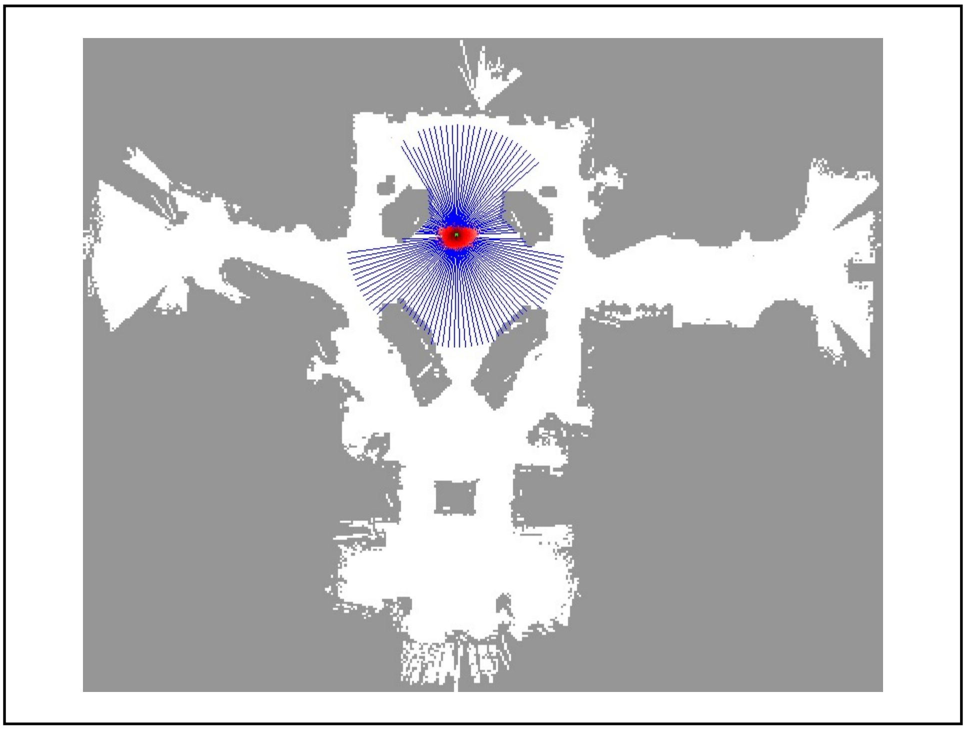
86



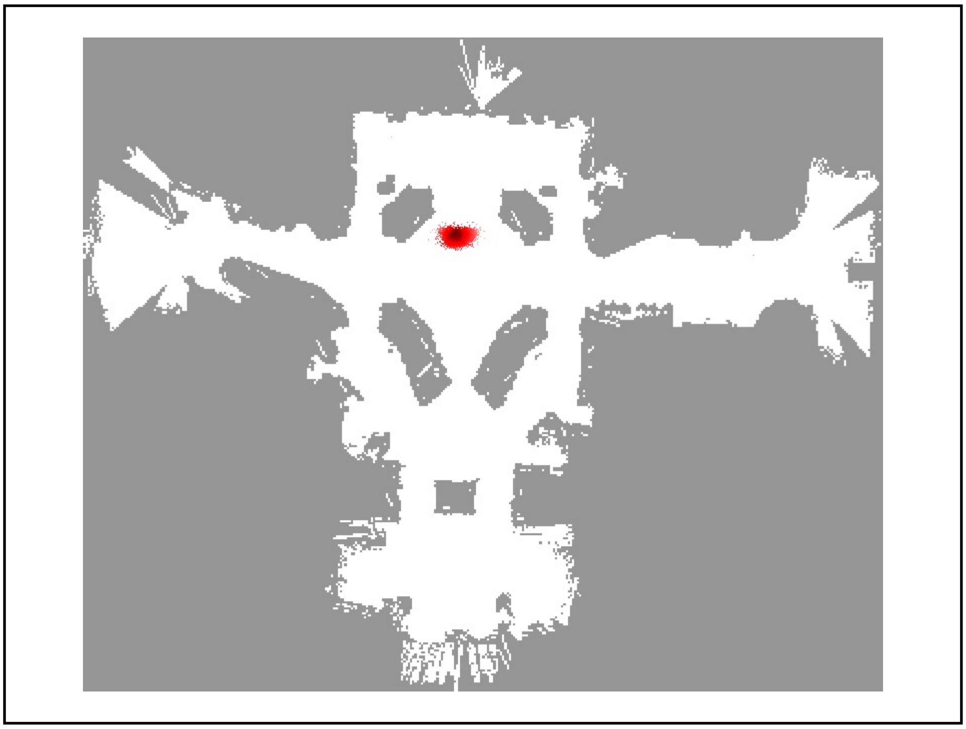
87



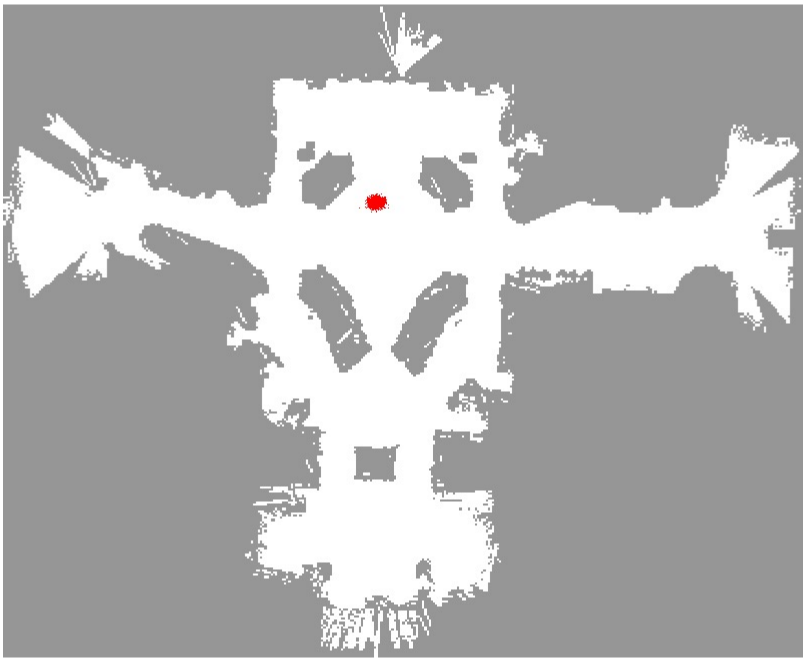
88



89



90



91

Probabilistic Robotics: MDP

- "Markov" generally means that given the present state, the future and the past are independent
- For Markov decision processes, "Markov" means action outcomes depend only on the current state

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \dots S_0 = s_0)$$
$$=$$
$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

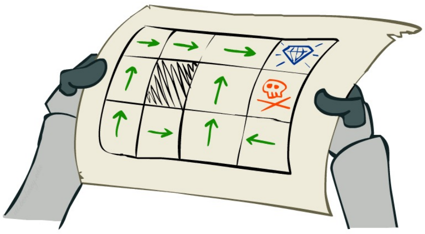
- This is just like search, where the successor function could only depend on the current state (not the history)

Andrey Markov (1856-1922)

92

Probabilistic Robotics: Policies

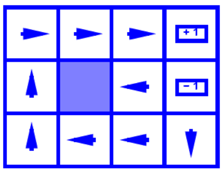
- In deterministic single-agent search problems, we wanted an optimal plan, or sequence of actions, from start to a goal
- For MDPs, we want an optimal policy $\pi^*: S \rightarrow A$
  - A policy π gives an action for each state
  - An optimal policy is one that maximizes expected utility if followed
  - An explicit policy defines a reflex agent
- Expectimax didn't compute entire policies
  - It computed the action for a single state only



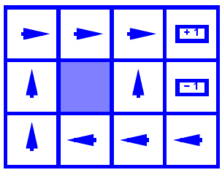
Optimal policy when R(s, a, s') = -0.03 for all non-terminals s

93

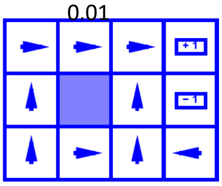
Optimal Policies



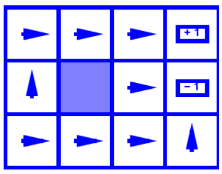
R(s) = -0.01



R(s) = -0.03



R(s) = -0.4

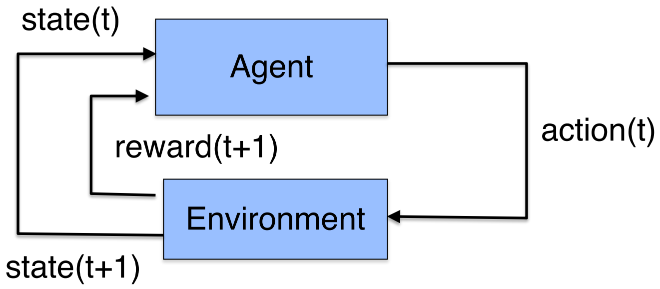


R(s) = -2.0

94

Robot Learning, Reinforcement Learning

- How to improve performance over time from our own/systems experience
- Goal directed learning from interaction
- How to map situations to action to maximize reward



95

96