## Advanced Artificial Intelligence

CS 687

Jana Kosecka, 4444 Research II
kosecka@gmu.edu ,  3-1876

## Logistics

- **Grading:** Homeworks/Projects 60% Exam 40%
- **Prerequisites:** Basic statistical concepts, geometry, linear algebra, calculus, CS 580
- **Course web page:** cs.gmu.edu/~kosecka/cs687/

- Homeworks/Projects every 1-2 weeks, Optional Final Project
- Late policy: budget of 3 late days

## Required Text

- S. Russell and P. Norvig: **Artificial Intelligence: A Modern Approach** (at least second edition)
- R. Sutton and A. G. Barto: **Introduction to Reinforcement Learning** (on-line materials see course www)
- Course goal – gain breadth in AI

## Relation to other courses

- CS 685 Intelligent Robotic Systems
- CS 682 Computer Vision
- CS 688 Pattern Recognition
- CS 780 Data Mining
- CS 782 Machine Learning
- CS 659 Theory and Applications of Data Mining
- SYS/STAT 664 Bayesian Inference and Decision Theory

- Advanced AI
- More in depth coverage: Probabilistic Graphical Models, Reinforcement Learning, Natural Language Processing, Markov Decision Processes, Robotics, Computer Vision

## Today's outline

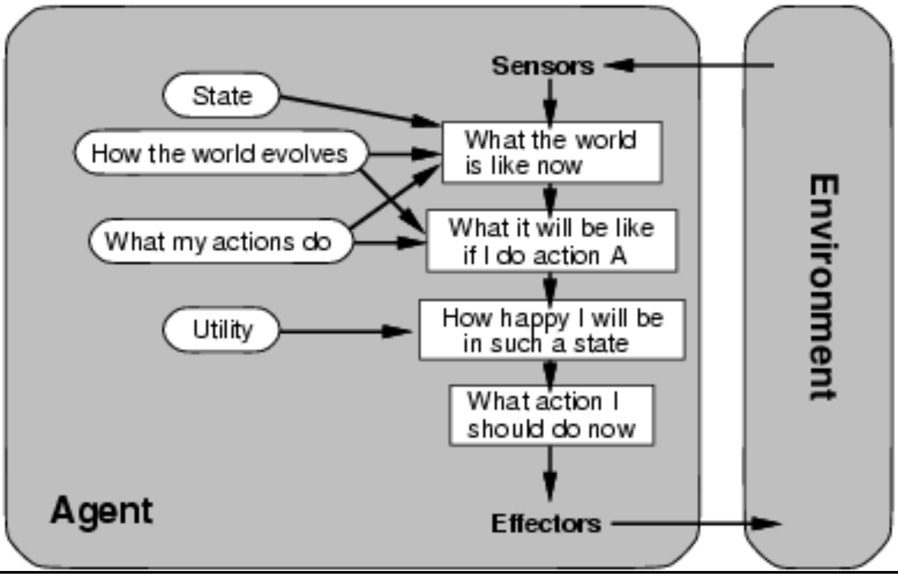
- History of AI
- AI approaches
- AI applications to intelligent agent design, robotics, computer vision, game playing, medical diagnosis
- Outline of course topics – Advanced AI in 10 slides

- Part I
- Supervised Learning
- Regression and Classification problems

## Intelligent Agents

- Agents – humans, robots, termostats, web applications
- Agent programs map percept histories to actions
- We focus on the design of rational agents, which will try to maximize the expected value of the performance measure given the percepts up to now
- Performance measure, environment, actuators, sensors
- Automated taxi
- Internet Shopping agent
- Environment types
- Observable, deterministic, episodic, static, discrete
- What are the environment types for different agents ?
- Environment type determines the type of agent

## Utility-based agent




## Robotics and AI

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

Computer Vision
- study of perception
- recognition, vision and motion, segmentation and grouping representation

Natural Language Processing
- provides better interfaces, parsing, understanding, machine translation language grounding problem

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

[Flakey robot video]

## Robotic Navigation

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

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

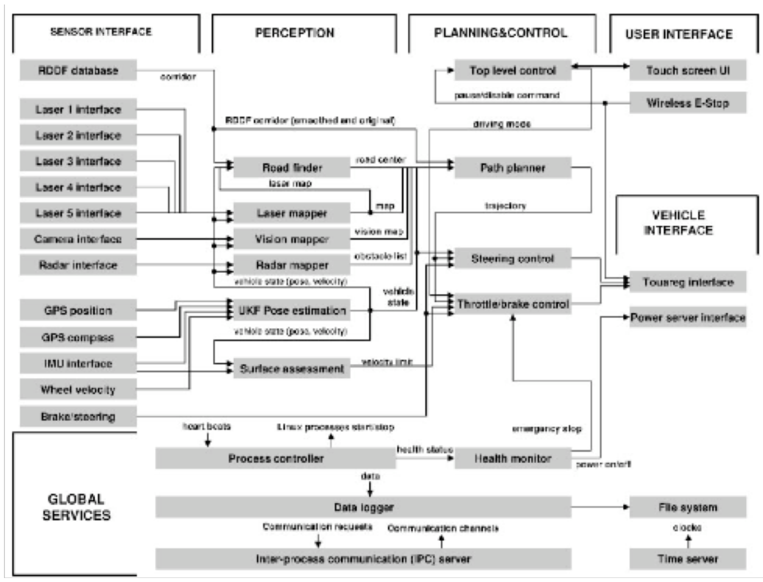


## Robot Components (Stanley)

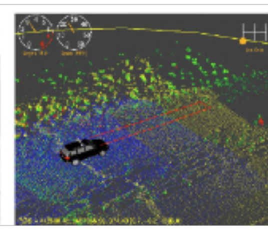
- Sensors
- Actuators-Effectors
- Locomotion System
- Computer system – Architectures – (the brain)

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

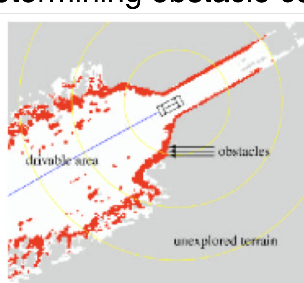
## Stanley Software System

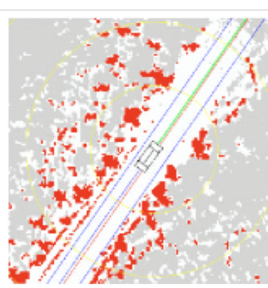



- Terrain mapping using lasers

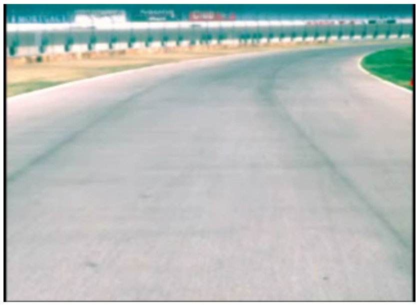


- Determining obstacle course



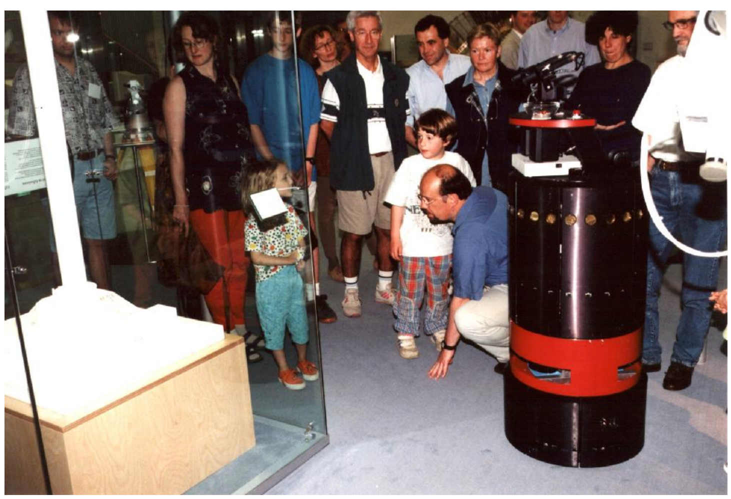



## Example 6: Classification



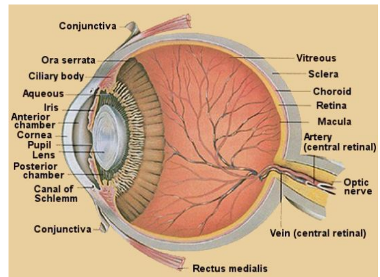
## Rhino – First Museum Tour giving robot

University of Bonn ('96)



## Computer Vision

Visual Sensing




Images I(x,y) – brightness patterns

- image appearance depends on structure of the scene
- material and reflectance properties of the objects
- position and strength of light sources

• Recovery of the properties of the environment from single or multiple views

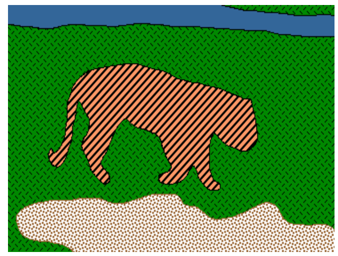
Vision problems

- Semantic Segmentation
- Recognition
- Reconstruction
- Vision Based Control - Action

Visual Cues

• Stereo, motion, shading, texture, contour, brightness

Segmentation – partition image into separate objects

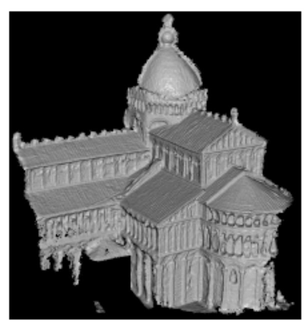


- Clustering and search algorithms in the space of visual cues
- Supervised and unsupervised learning strategies
- Object and Scene recognition/categorization

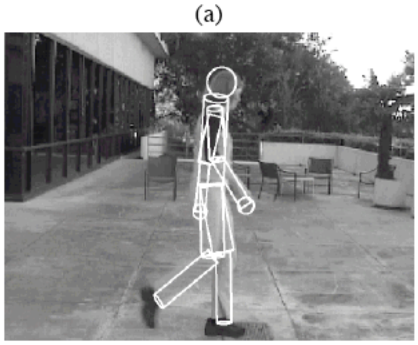
So what does object recognition involve?

Object categorization

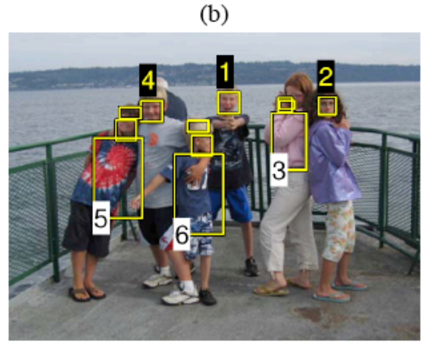

(a)



(b)



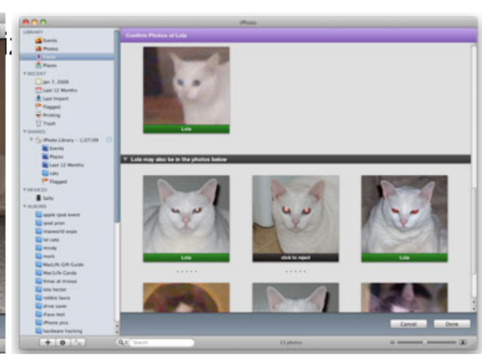
(c)




(d)

http://www.apple.com/ilife/iphoto/

## Consumer application: iPhoto 2009



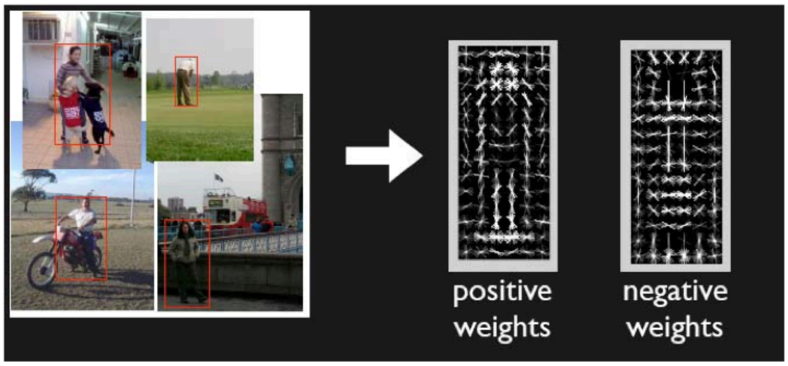
http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

## Consumer application: iPhoto 2009



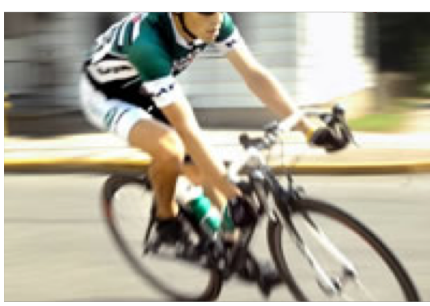
Dalal and Triggs, CVPR 2005

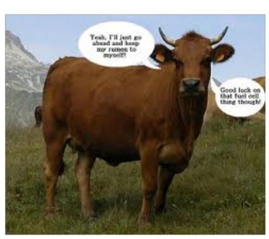
$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



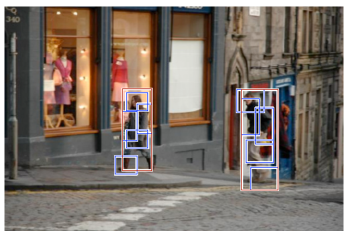

Slides from Deva Ra

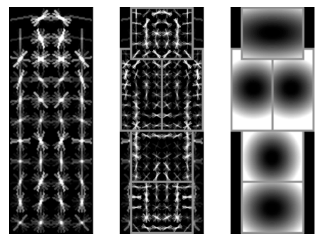
t c



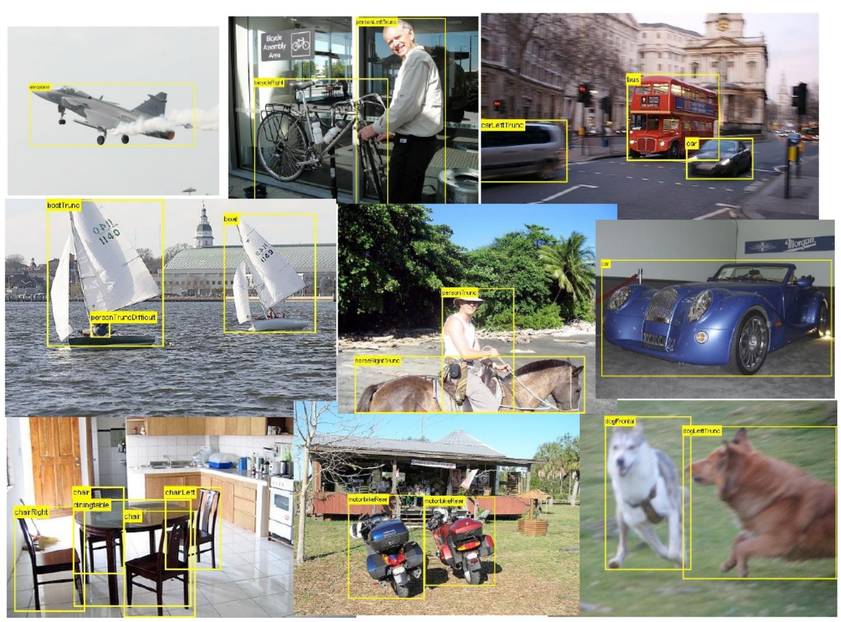



## More sliding window detection:
## Discriminative part-based models



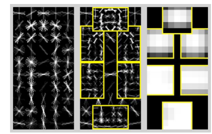


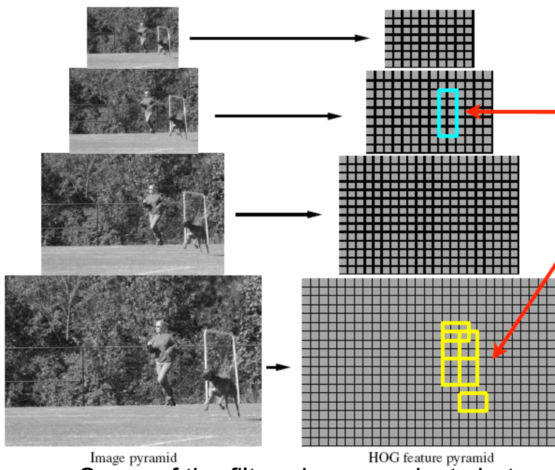
Many slides based on [P. Felzenszwalb]



## Object Hypothesis

- Multiscale model: the resolution of part filters is twice the resolution of the root






$z = (p_0, \ldots, p_n)$

$p_0$ : location of root

$p_1, \ldots, p_n$ : location of parts

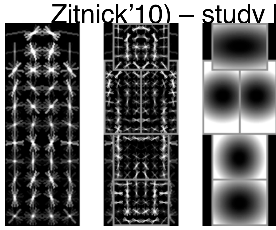
Score is sum of filter scores minus deformation costs

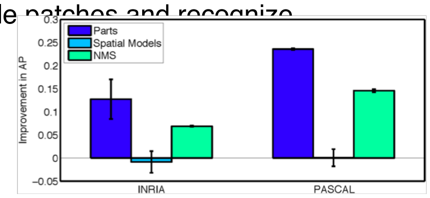
Score of the filter : inner products between the filter and features

## State of the art

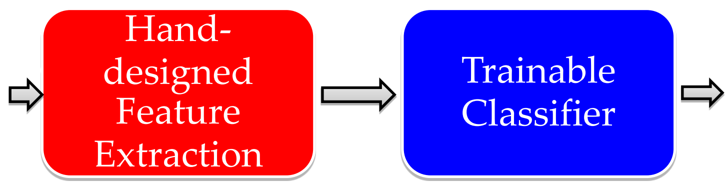
- Previous approaches
- Hand Designed Features (SIFT, HOG, GIST …)
- What is next ? Better Features ? More Training data ? Better classifiers ?
- Main factor compared to humans is better features (Parikh & Zitnick'10) – study look at little patches and recognize

ts are





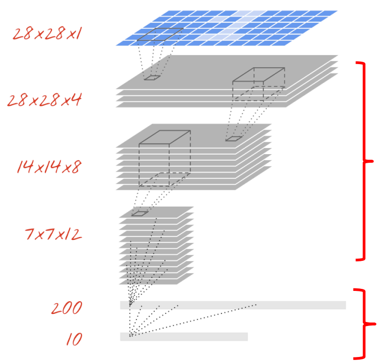
## Traditional Recognition Approach




## What about learning the features?

- Learn a *feature hierarchy* all the way from pixels to classifier
- Each layer extracts features from the output of previous layer
- Train all layers jointly

## Convolutional Neural Networks




**Convolutional Layer parameters**

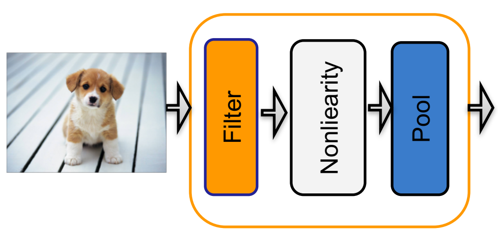
Feature Maps

- Spatial support of filters and stride
- Number of filters per layer = number of feature maps

**Fully connected layers**

- Learning non-linear combinations of features

## Single Layer Architecture

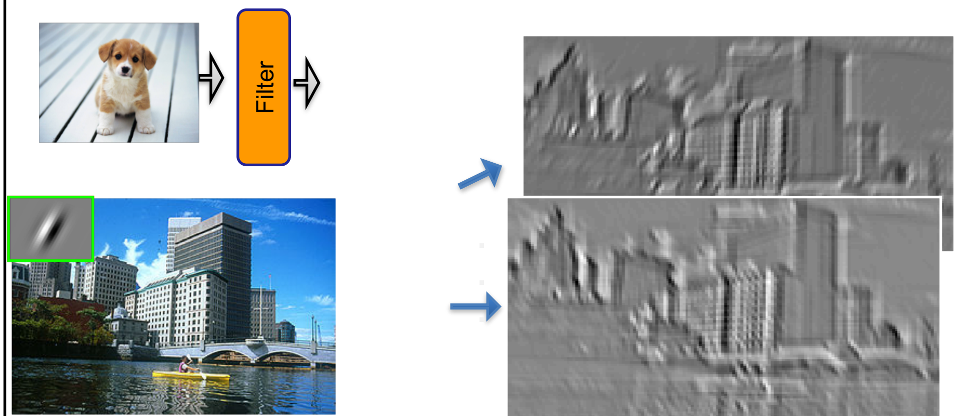



Output: Features / Classifier

## Filtering

Translation equivariance

- Tied filter weights (same at each position)

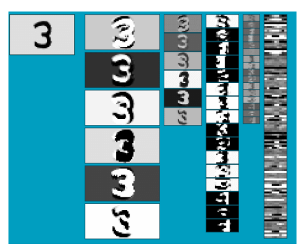


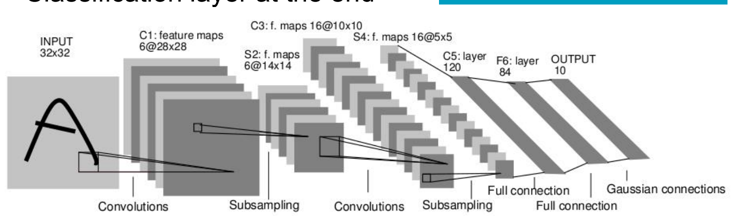

Slide courtesy of Deep Learning Tutorial, CVPR 2012 (Rob Fergus and others)

**Feature Map**

## Convolutional Neural Networks (CNN, Convnet)

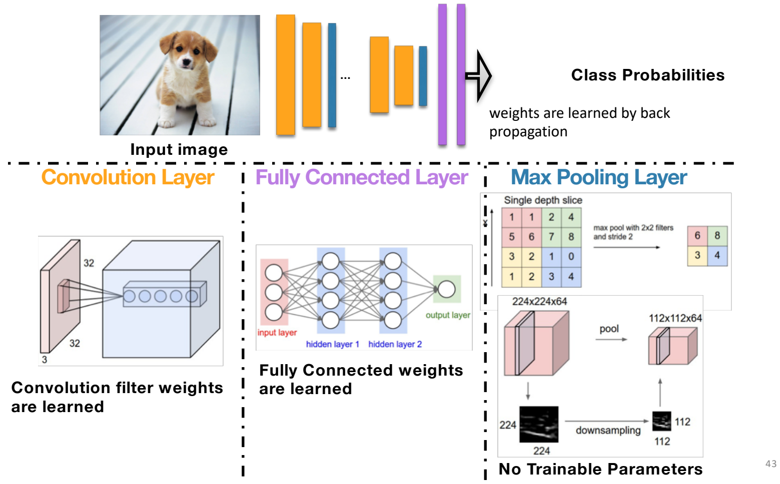
- Neural network with specialized connectivity structure
- Stack multiple stages of feature extractors
- Higher stages compute more global, more invariant features
- Classification layer at the end






Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,
Gradient-based learning applied to document recognition, Proceedings of the IEEE

# Convolutional Neural Networks (CNN)




# Application to ImageNet

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

**ImageNet Classification with Deep Convolutional Neural Networks** [NIPS 2012]


Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

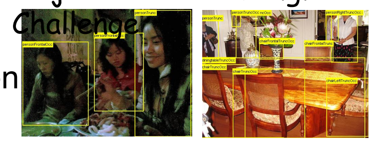
## Detection tasks

Object

Pedestrian detection

Car detection

Object detection: e.g. Pascal Challenge



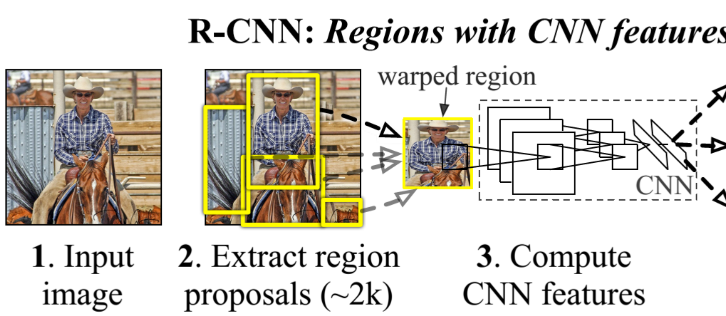
Pascal challenge Everingham'12

Pedestrian detection

www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

Caltech Pedestrian Detection Benchmark Dollar et al'12

## R-CNN Algorithm: Proposal boxes + CNN features




Girshick et al'14

Selective Regression Search

DNN training

SVM

## Overview of the topics

- Supervised learning
- Representation of uncertainty
- Bayesian Networks
- Inference and Learning in Bayesian networks
- Hidden Markov Models
- Bayes filters, Kalman filters
- Visual Perception
- Robot Perception and Control
- Reinforcement learning

- With applications to intelligent agent design, robotics, computer vision, game playing, medical diagnosis

## Supervised/Unsupervised learning

- Design of agents which learn from observations and improve performance on future tasks
- Regression and classification problems
- Regression - e.g. prediction of house prices
- Classification – disease/no disease
- Artificial neural networks

- Unsupervised learning
- Finding structure in the available data

## Representation of uncertainty

- Needs of agents to handle uncertainty due to non-determinism or partial observability
- How to represent uncertain knowledge
- Basis of probabilistic reasoning
- E.g. Bayes rule

## Bayes nets - Probabilistic Graphical Models

Graphical models offer several useful properties:

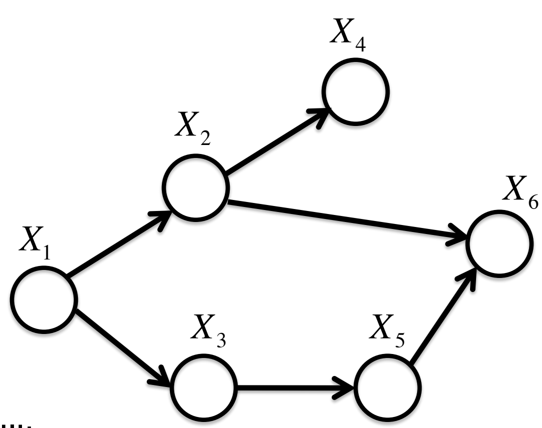
1. Models are descriptions of how parts of the world work
2. May not account for every variable
3. May not account for every interaction
4. Enable us to reason about unknown variables given some evidence
   - explanation (diagnostic reasoning)
   - prediction (causal reasoning)

## Probabilistic Graphical Models

Graphical models offer several useful properties:

1. They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models.
2. Insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph.
3. Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

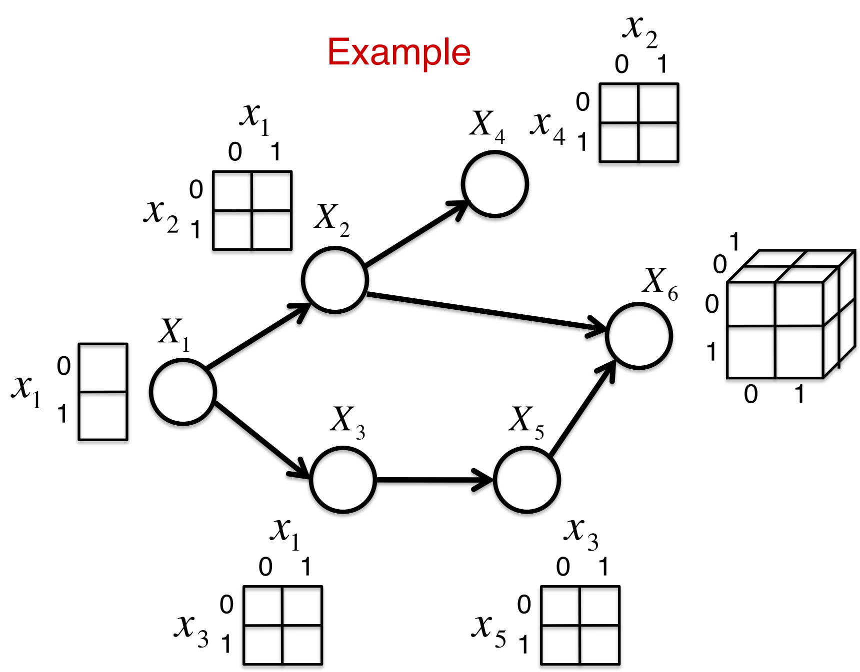
## Example




Joint Probability:

$$p(x_1,x_2x_3,x_4,x_5,x_6) =$$
$$p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2,x_5)$$

## Example




## Applications

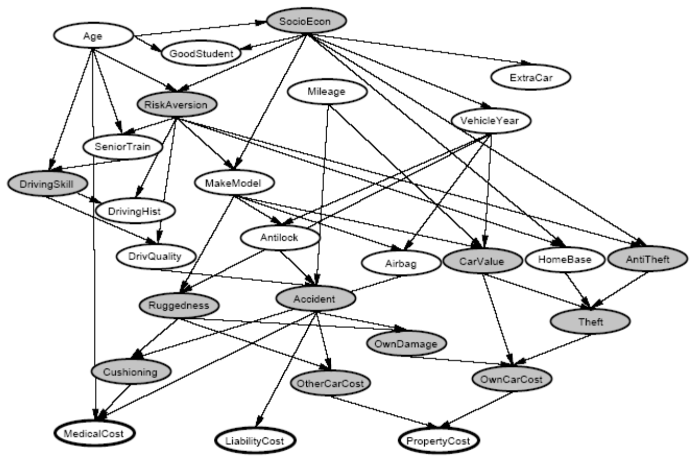
Implementations in real life :

- It is used in the Microsoft products (Microsoft Office)
- Medical applications and Biostatistics (BUGS)
- In NASA Autoclass project for data analysis
- Collaborative filtering (Microsoft – MSBN)
- Fraud Detection (ATT)
- Speech recognition (UC , Berkeley )

65

## Bayesian Networks

- Graphical models, efficient representation of joint probability distribution
- Credit card companies - Fradulent transaction detection



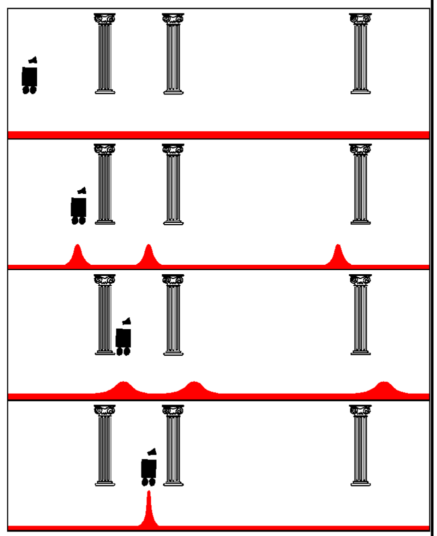

## Probabilistic Reasoning in Time

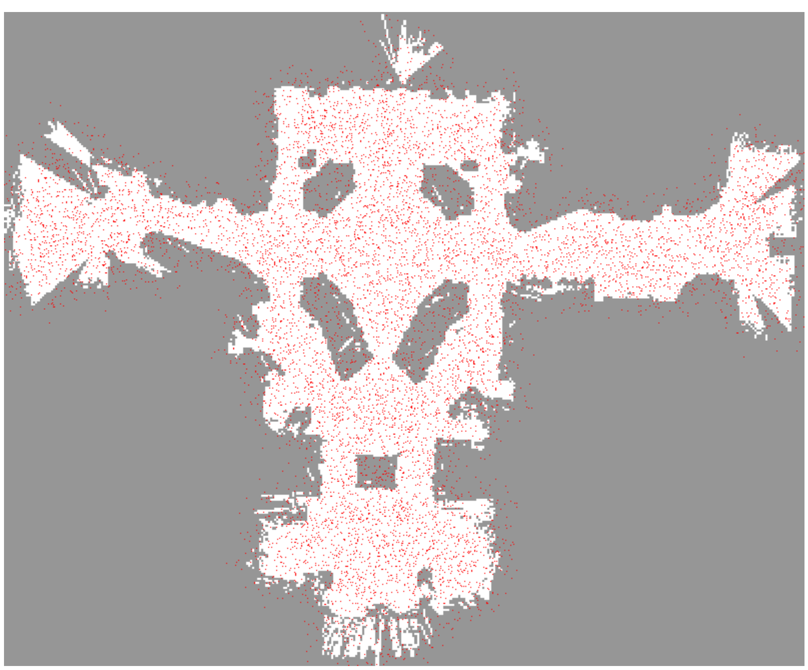
- Tracking
- Robotic localization
- Propagating beliefs
- Includes models of dynamics of the worlds

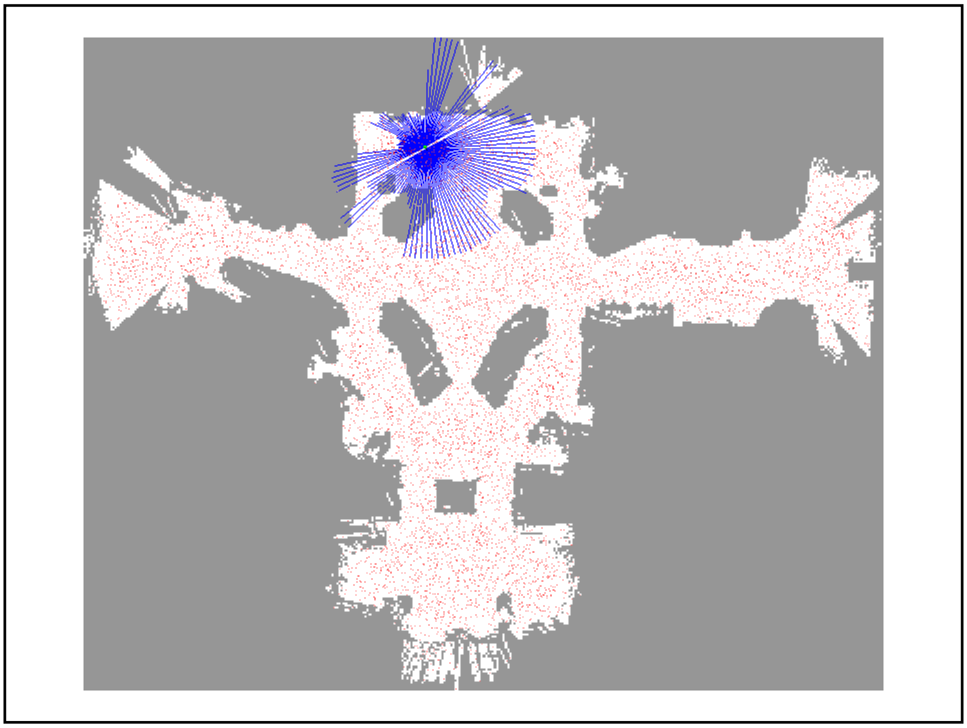
- Hidden Markov Model
- Natural Language Processing, Speech Analysis

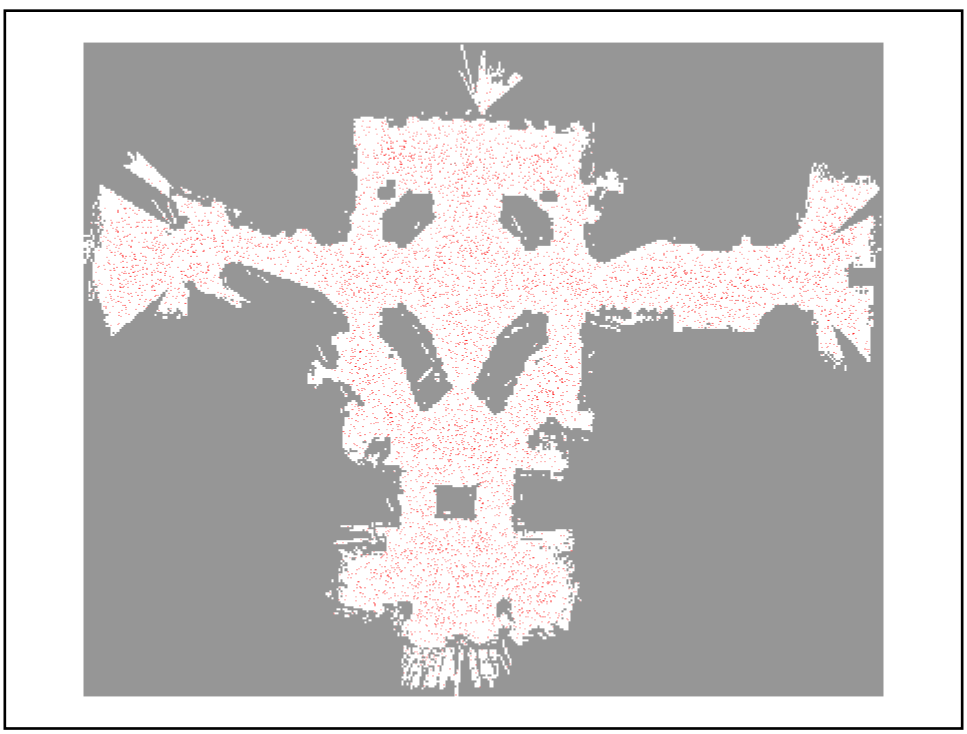
# Markov Localization

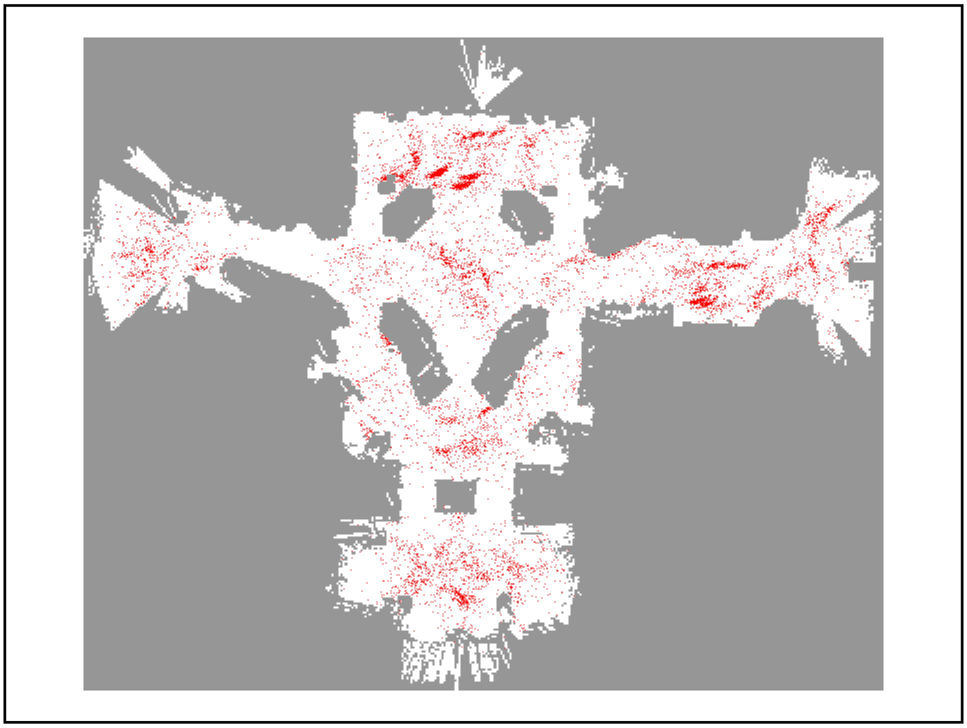
1. Start
   - No knowledge at start, thus we have an uniform probability distribution.
2. Robot perceives first pillar
   - Seeing only one pillar, the probability being at pillar 1, 2 or 3 is equal.
3. Robot moves
   - Action model enables to estimate the new probability distribution based on the previous one and the motion.
4. Robot perceives second pillar
   - Base on all prior knowledge the probability being at pillar 2
   - Becomes dominant

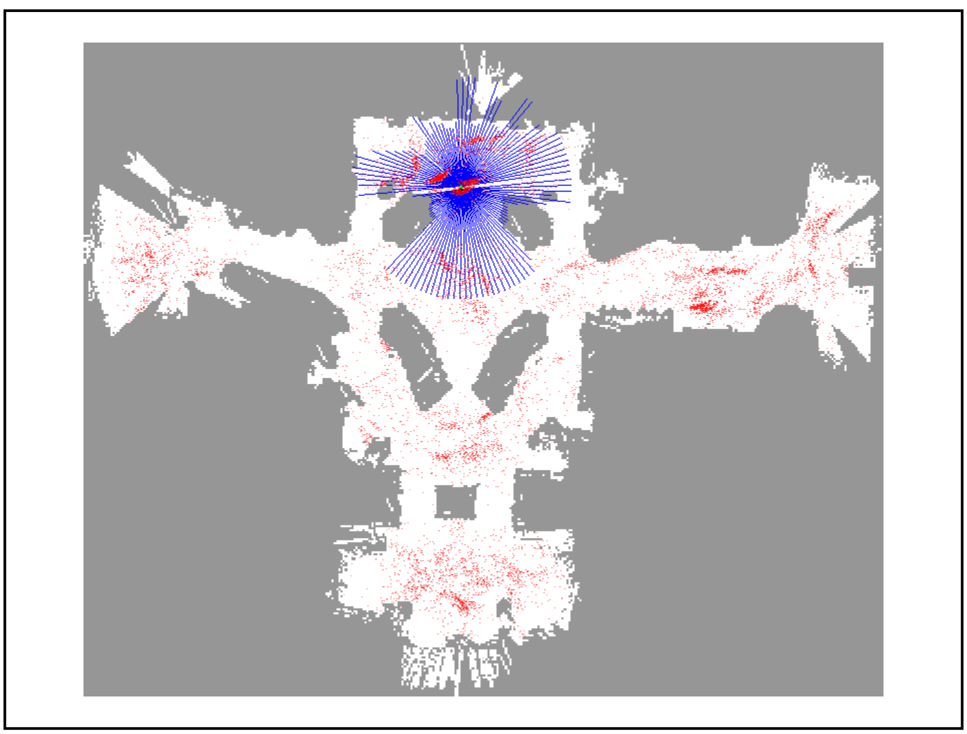


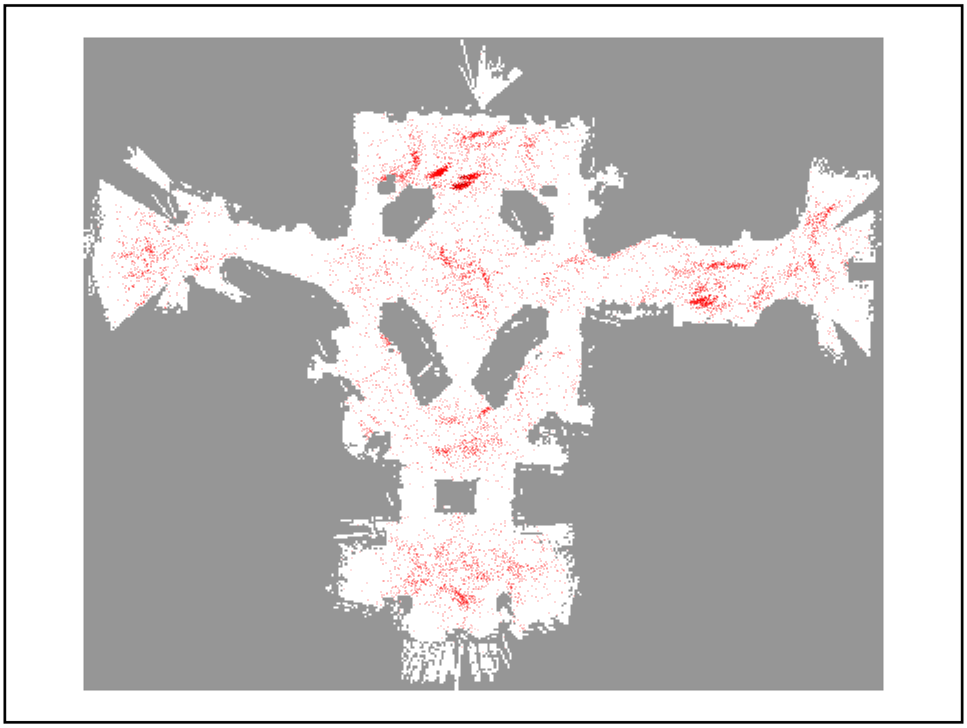


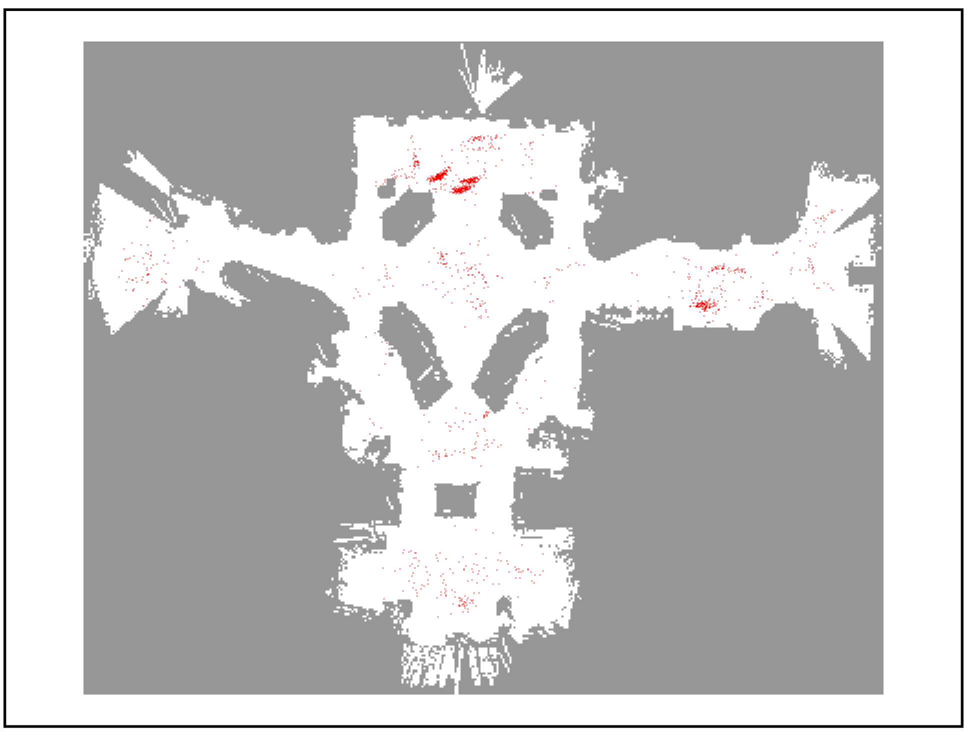


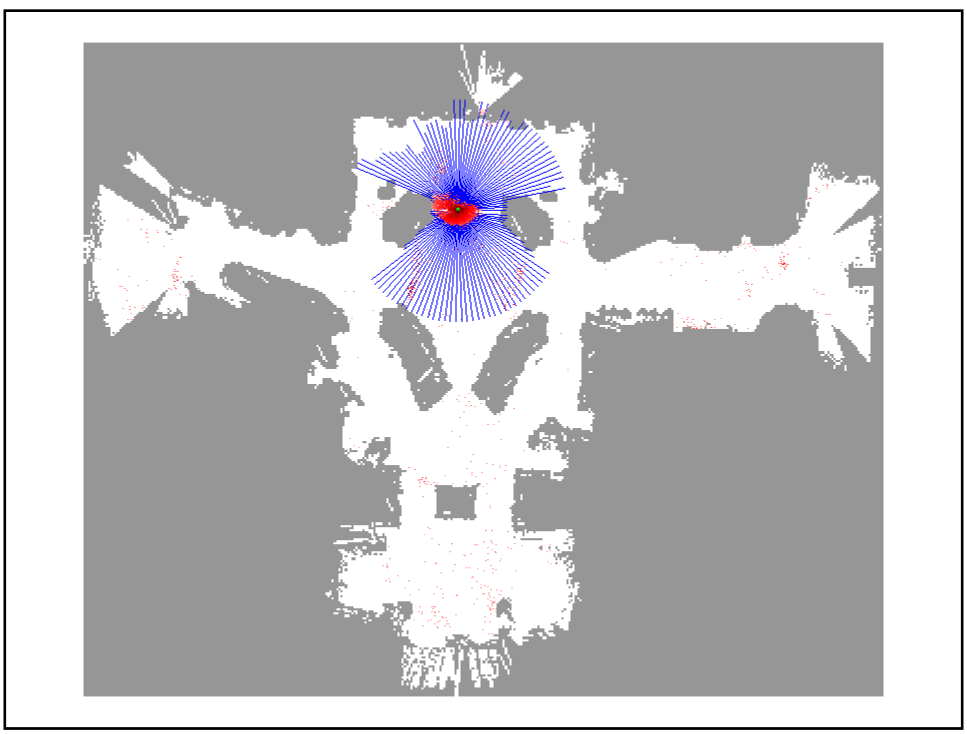


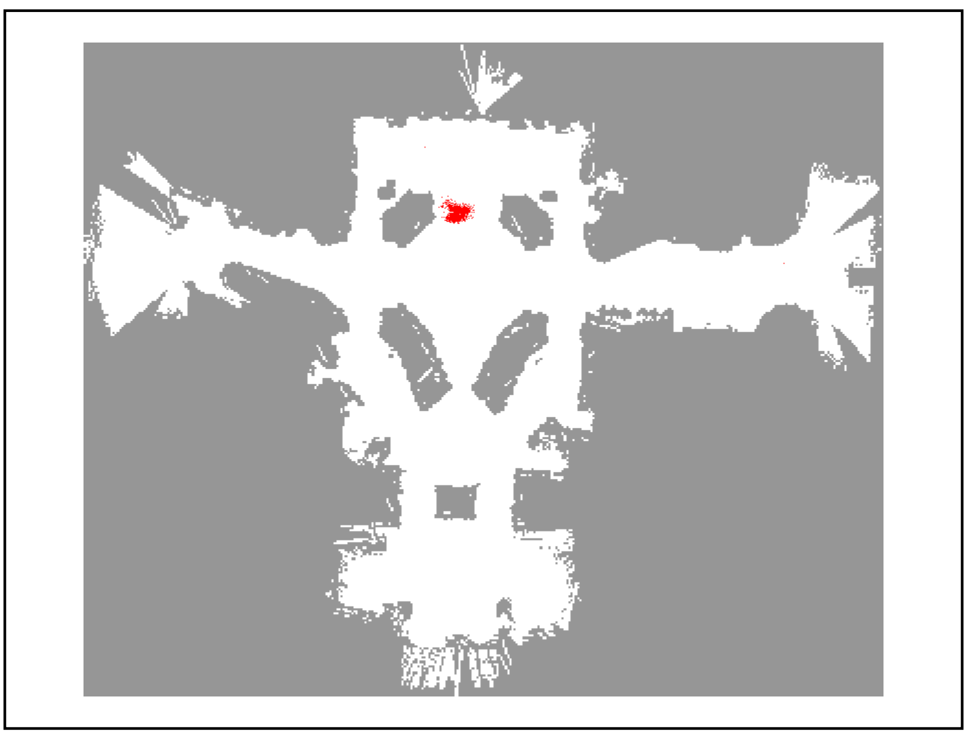


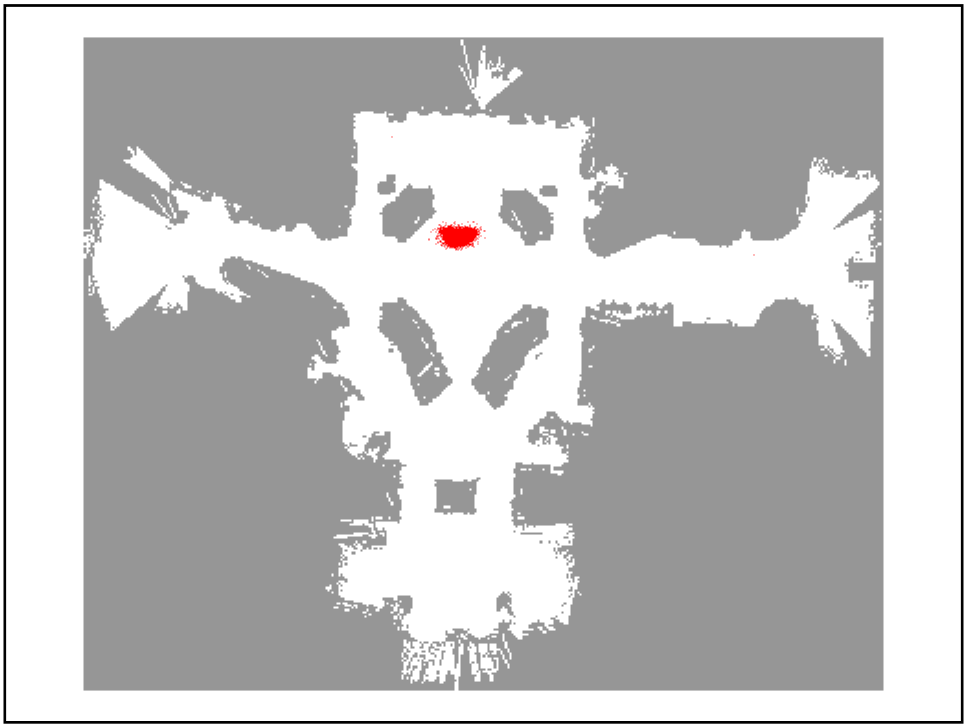


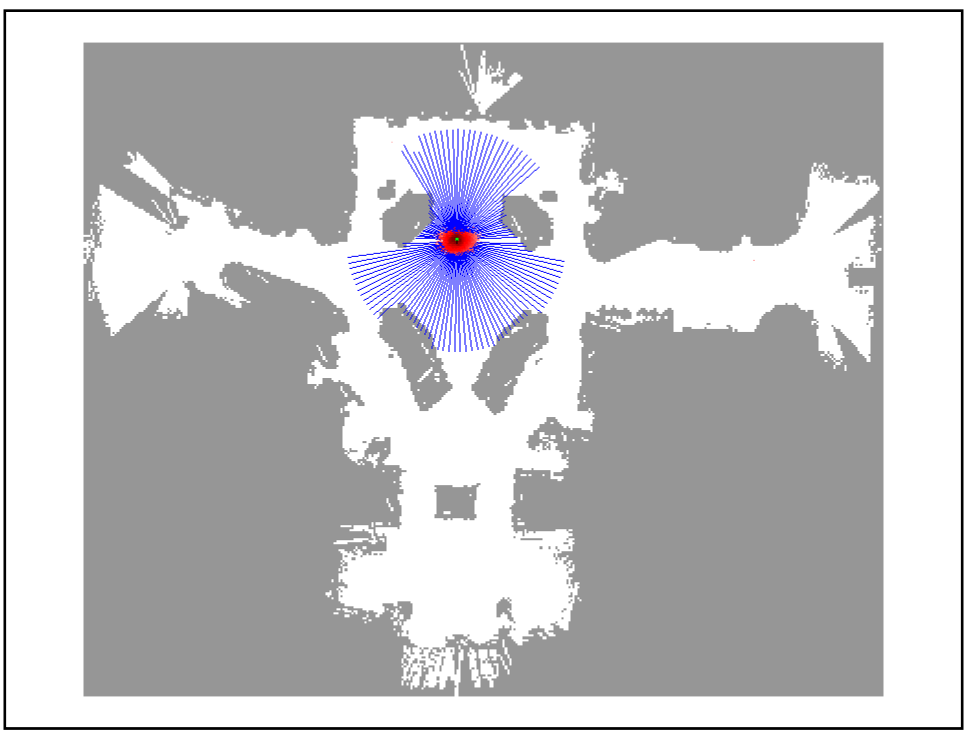


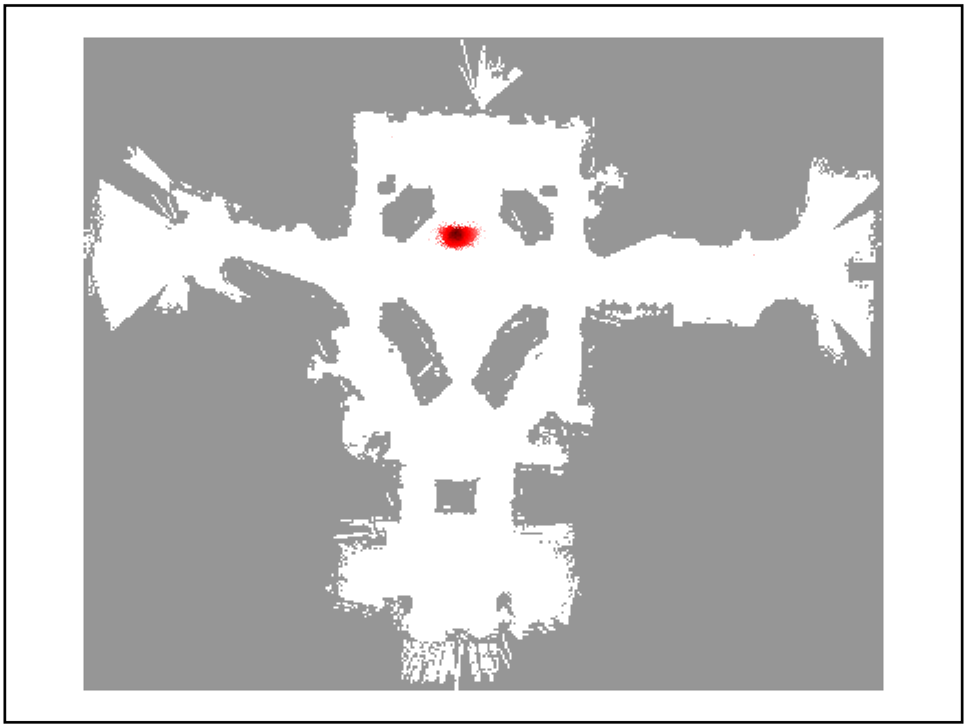


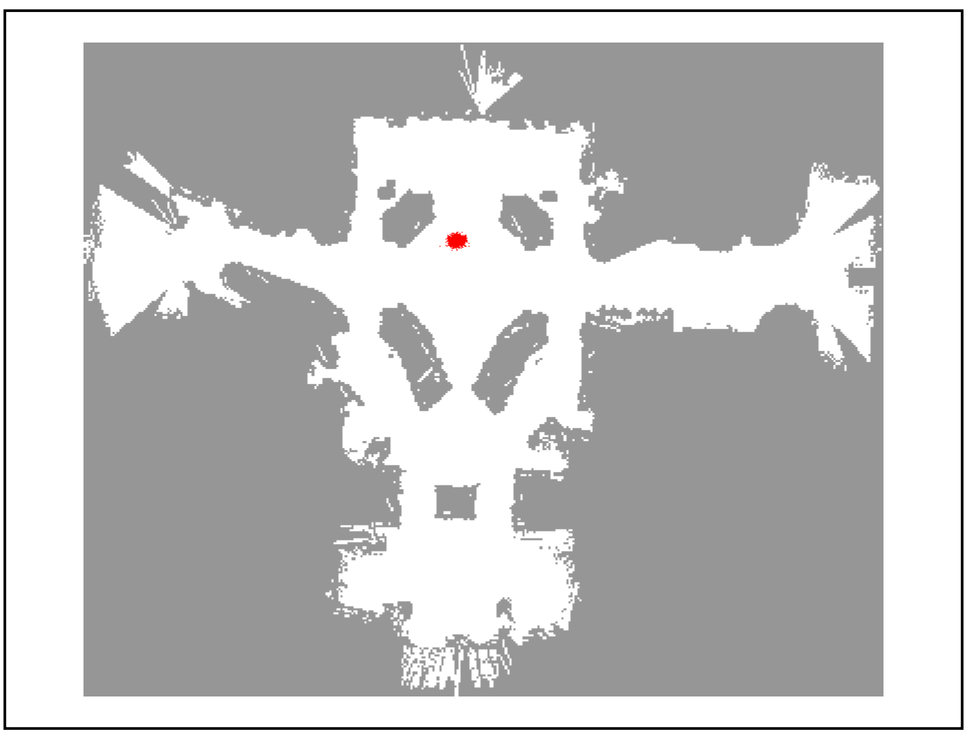


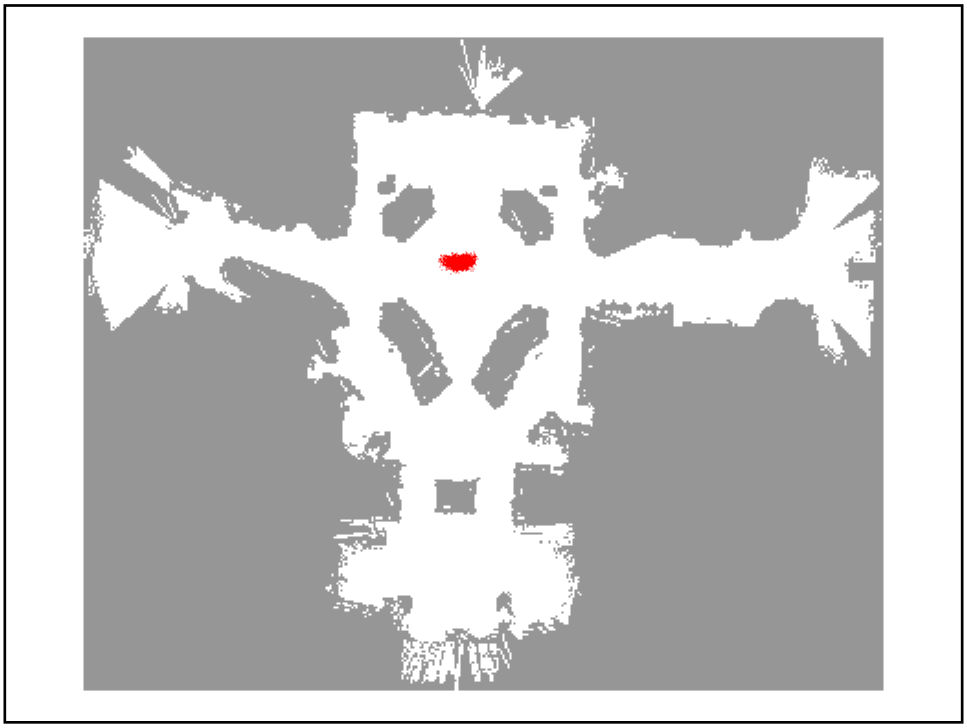


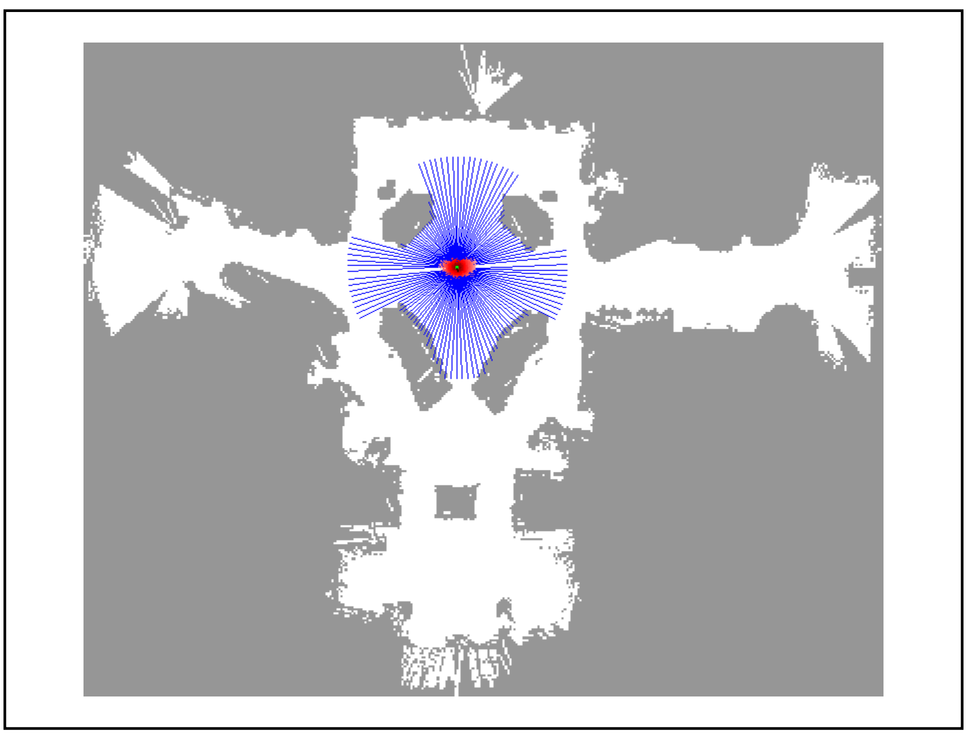


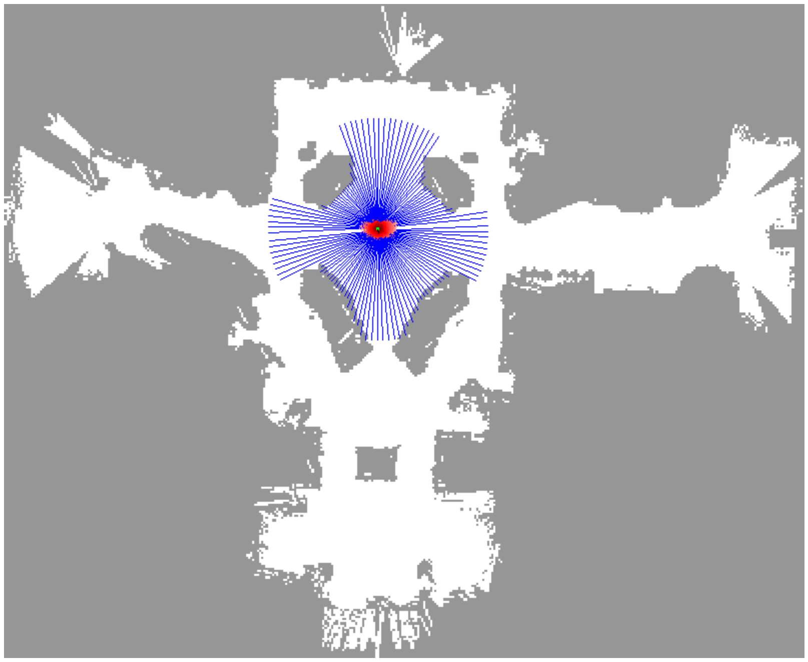




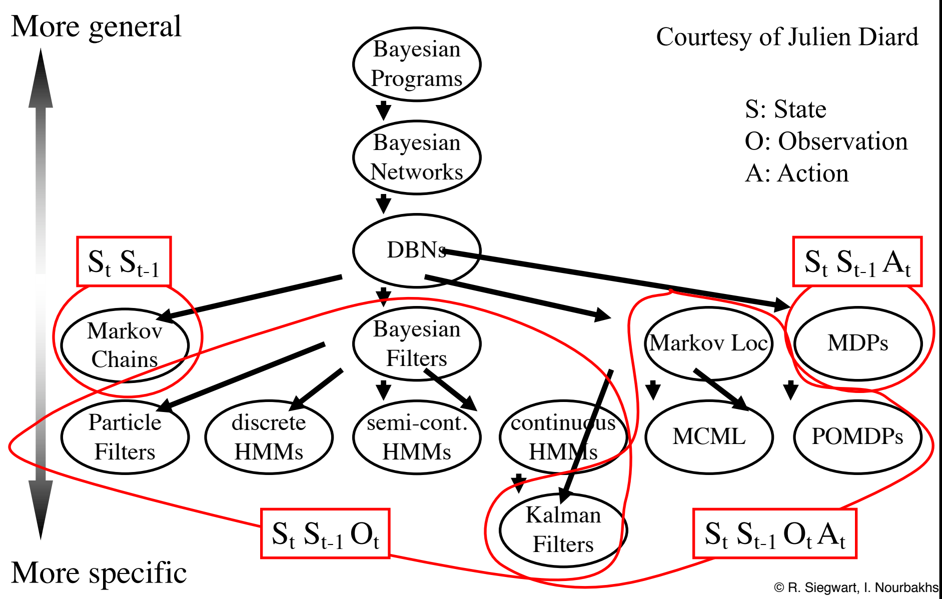






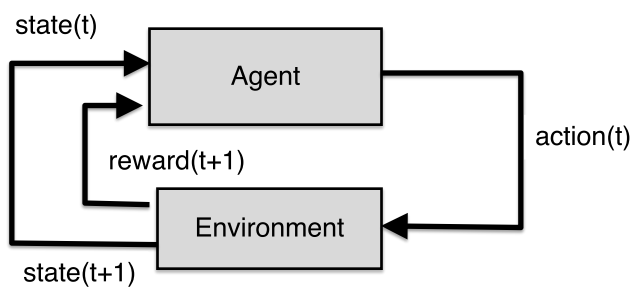


## Probabilistic models



## 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
- http://www.youtube.com/user/stanfordhelicopter




## Supervised Learning

- Blackboard Notes