Probabilistic Robotics!

Marek Perkowski

International Forum on Intelligent Robotics, KAIST, 17 April 2003

Difficult and interesting problems.

- **Mapping**: automatic, manual, guided?
- **Probabilistic localization:** landmarks?, odometer!,
- **Route planning**: collision avoidance
- **Mine Mapping**?

How can we solve them?

….probabilistic robotics...

- Localization
- Mapping (2D, 3D)
- Humans detection
- Moving objects avoidance
- Language dialogs/generation
- Robot Vision
- Sound source detection
- Movement control - planning of paths
- Sensor integration
- Probabilistic Neural Nets
- Hidden Markov Models
- Probabilistic Finite State Machines
- Bayesian Nets
- Fault Diagnosis
- Case Based Reasoning
- Image Processing
- Pattern Recognition

The list goes on and on...

Top Research Teams

- **Sebastian Thrun**
- **Dieter Fox**
- **Wolfram Burgard**
- Maja Mataric
- Gourav Soukhatme
- Kris Konolidge
- Ilyah Nourbankhsh
- Judea Pearl
- Manuela Veloso
- Luis Enrique Sucar

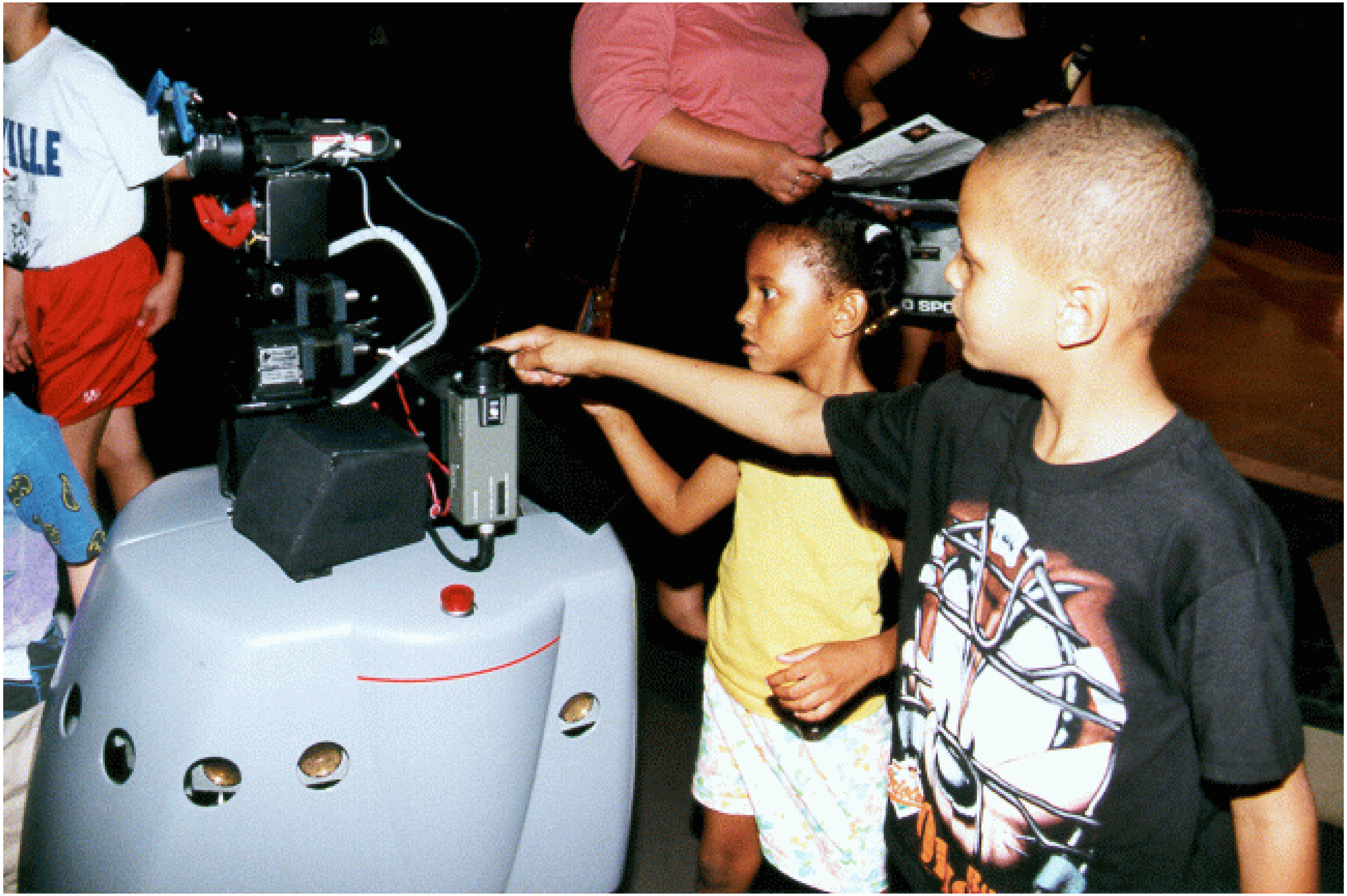
Probabilistic Robotics

1. Why should we use probabilistic techniques?

Key idea: Explicit representation of uncertainty (using the calculus of probability theory)

- Perception = state estimation
- Action = utility optimization
  - **Probabilistic State Estimation**
    - Localization
    - Mapping
  - **Probabilistic Decision Making**
    - Planning
    - Exploration

New robots are in dynamic complex real life environments such as museums or battlefields



System must be robust...

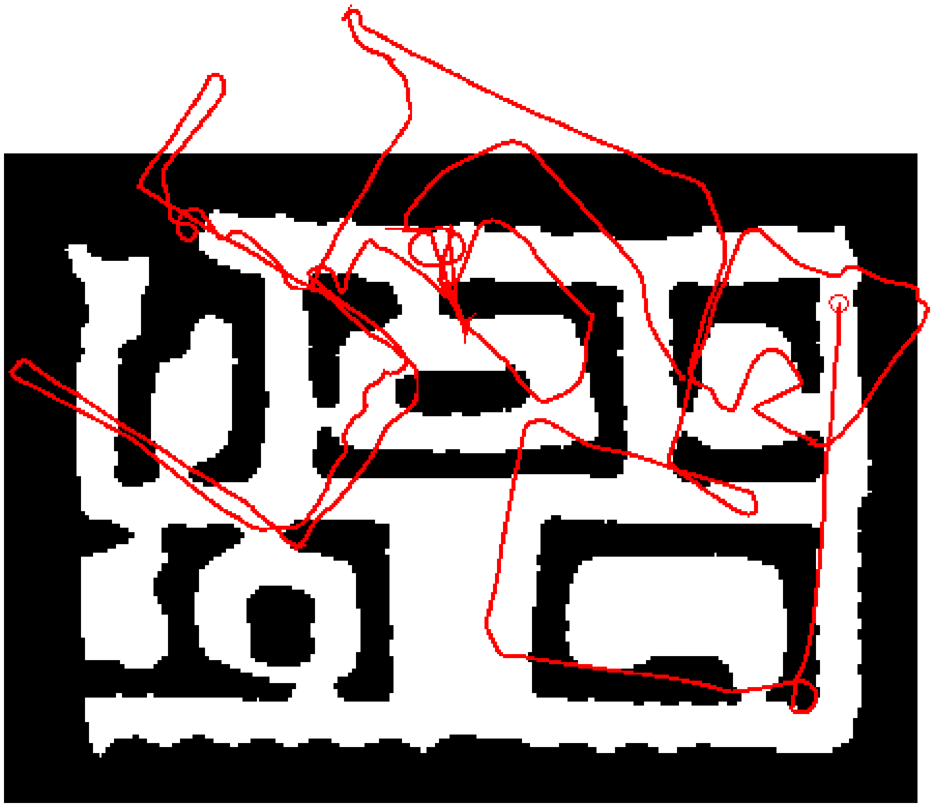
Robot Minerva

The Problems with Localization and Mapping

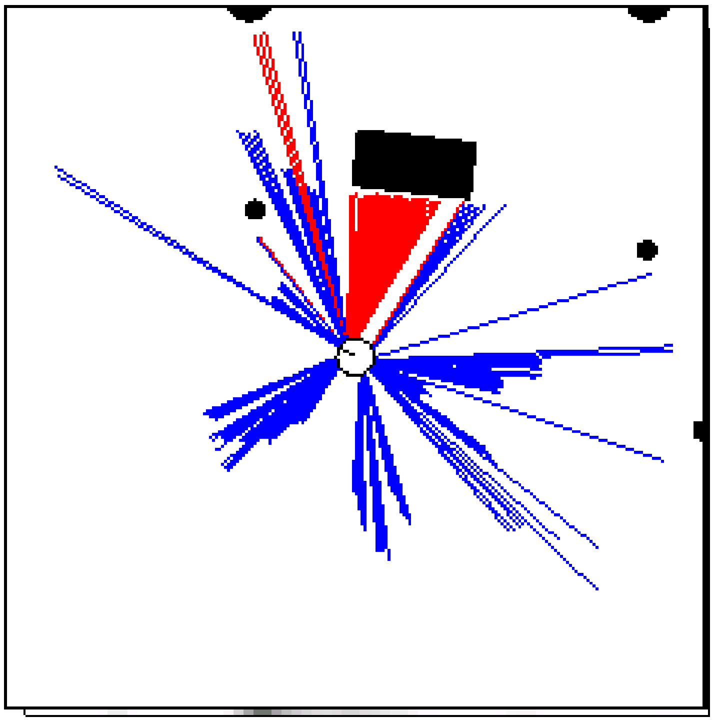
- Measurement noise
  - Sensor and Position noise is not independent
- Map size
  - High resolution maps can be very large
- Correspondence
  - Do multiple measurements at different times correspond to the same object?
- Dynamic environments
  - Most current algorithms assume a static environment

Command Noise

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



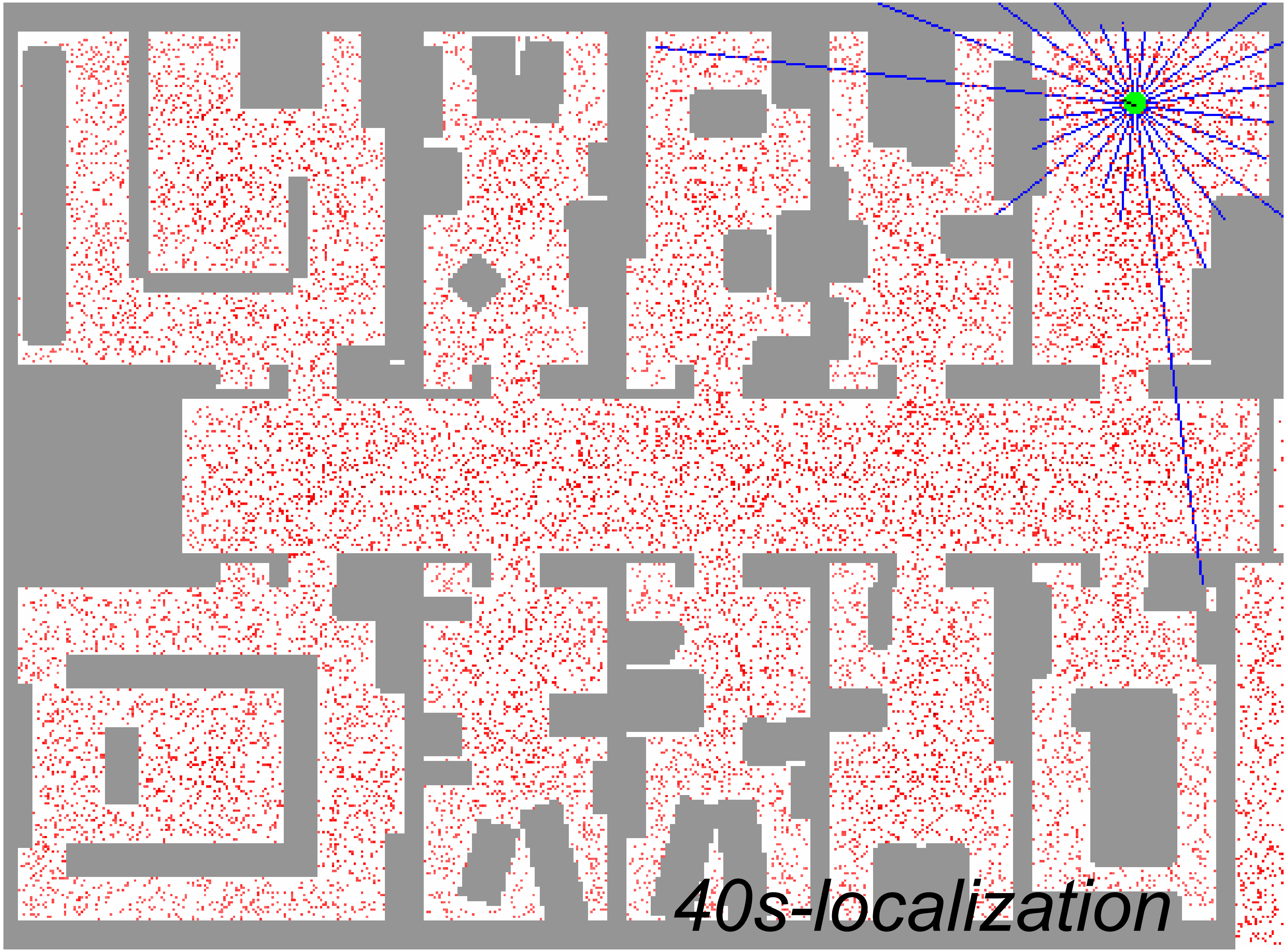
Odometry Data



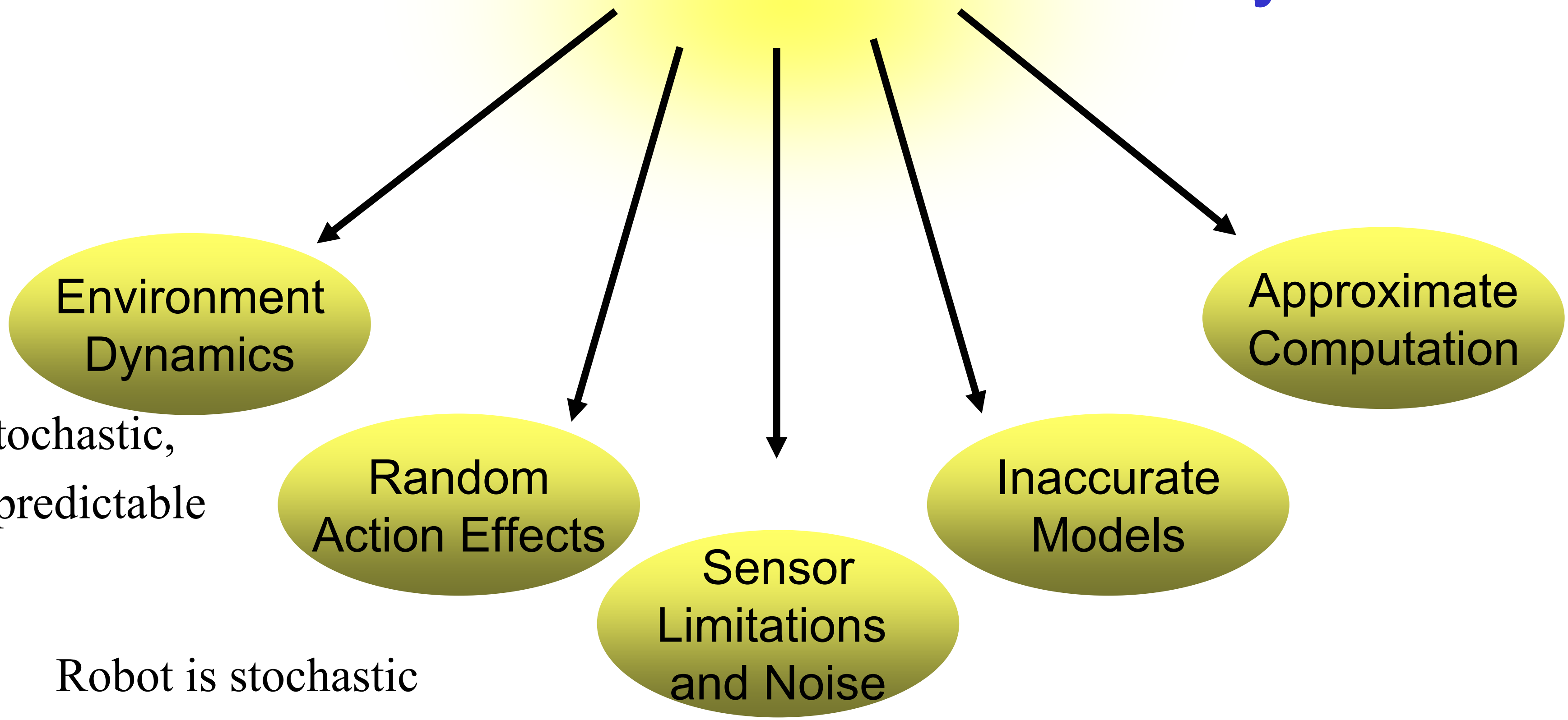
Range Data

Robots are Inherently Uncertain

Animation of Localization



Five Sources of Uncertainty



stochastic,
unpredictable

Robot is stochastic

Trends in Robotics

Classical Robotics (mid-70's)
- exact models
- no sensing necessary

Reactive Paradigm (mid-80's)
- no models
- relies heavily on good sensing

Hybrids (since 90's)
- model-based at higher levels
- reactive at lower levels

Probabilistic Robotics (since mid-90's)
- seamless integration of models and sensing
- inaccurate models, inaccurate sensors

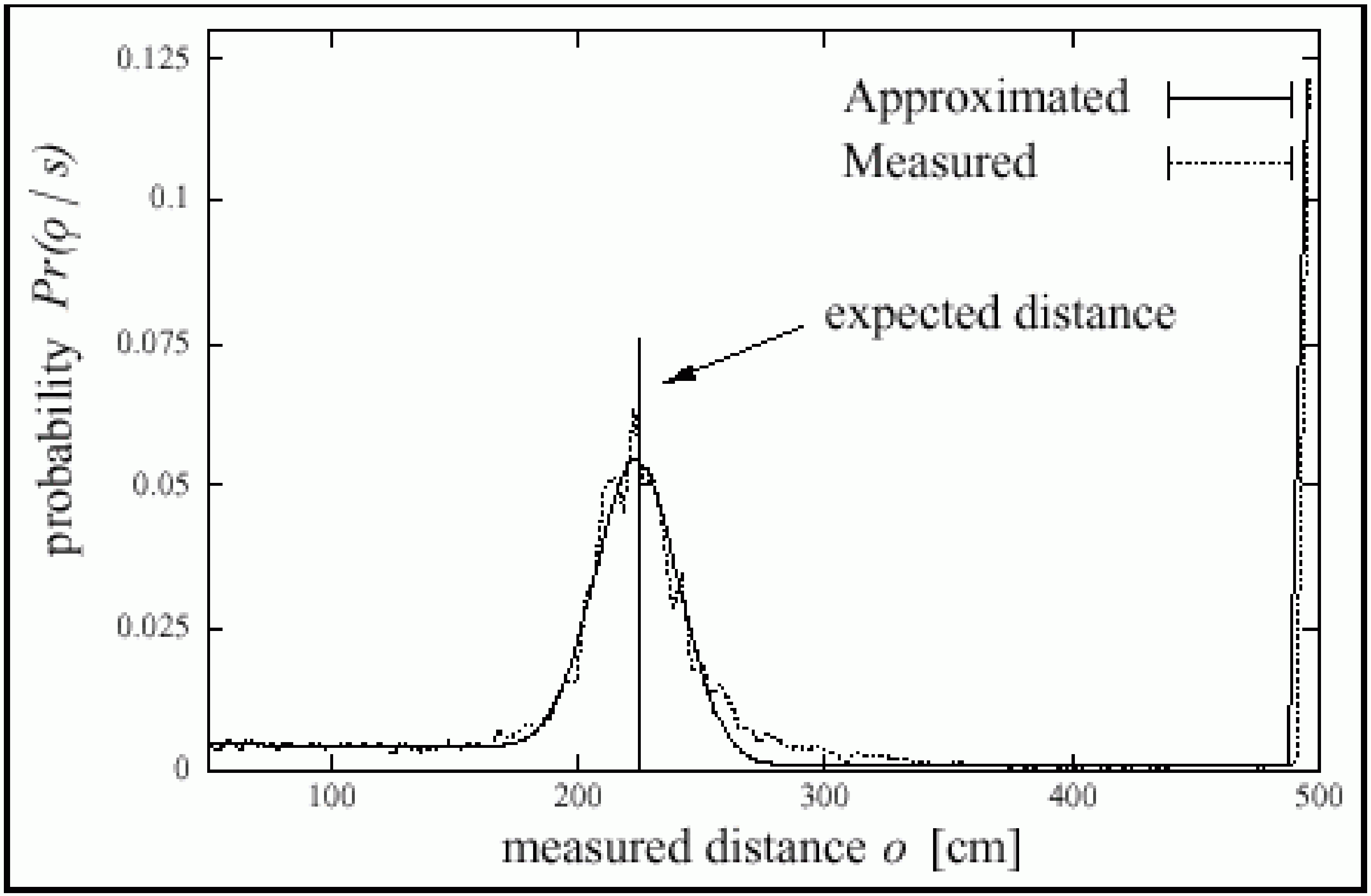
Advantages of Probabilistic Paradigm

- Can accommodate <u>inaccurate models</u>
- Can accommodate <u>imperfect sensors</u>
- <u>Robust</u> in real-world applications
- Best known approach to many <u>hard</u> robotics problems
- Pays Tribute to Inherent Uncertainty
  - Know your own ignorance
- Scalability
- No need for “perfect” world model
  - Relieves programmers

Pitfalls

- Computationally inefficient
  - Consider entire probability densities
- False assumptions
- Approximate
  - Representing continuous probability distributions

Probabilistic Sensor Model



Probabilistic sensor model for laser range finders

Bayes Rule

$$p(a \mid b) = \frac{p(b \mid a)\; p(a)}{p(b)}$$

$$p(a \mid b, c) = \frac{p(b \mid a, c)\; p(a \mid c)}{p(b \mid c)}$$

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

Law of Total Probability

$$p(a) = \sum_i p(a \wedge b_i)$$

Discrete

$$= \sum_i p(a \mid b_i) p(b_i)$$

Continuous

$$p(a) = \int p(a \mid b) p(b) db$$

it follows that:

$$p(a \mid b) = \int p(a \mid b, c) p(c \mid b) dc$$

Markov Assumption

Future is Independent of Past Given Current State

"Assume Static World"

Probabilistic Model

$$Bel(s_t) \quad = \quad p(s_t \mid d_{0\ldots t})$$

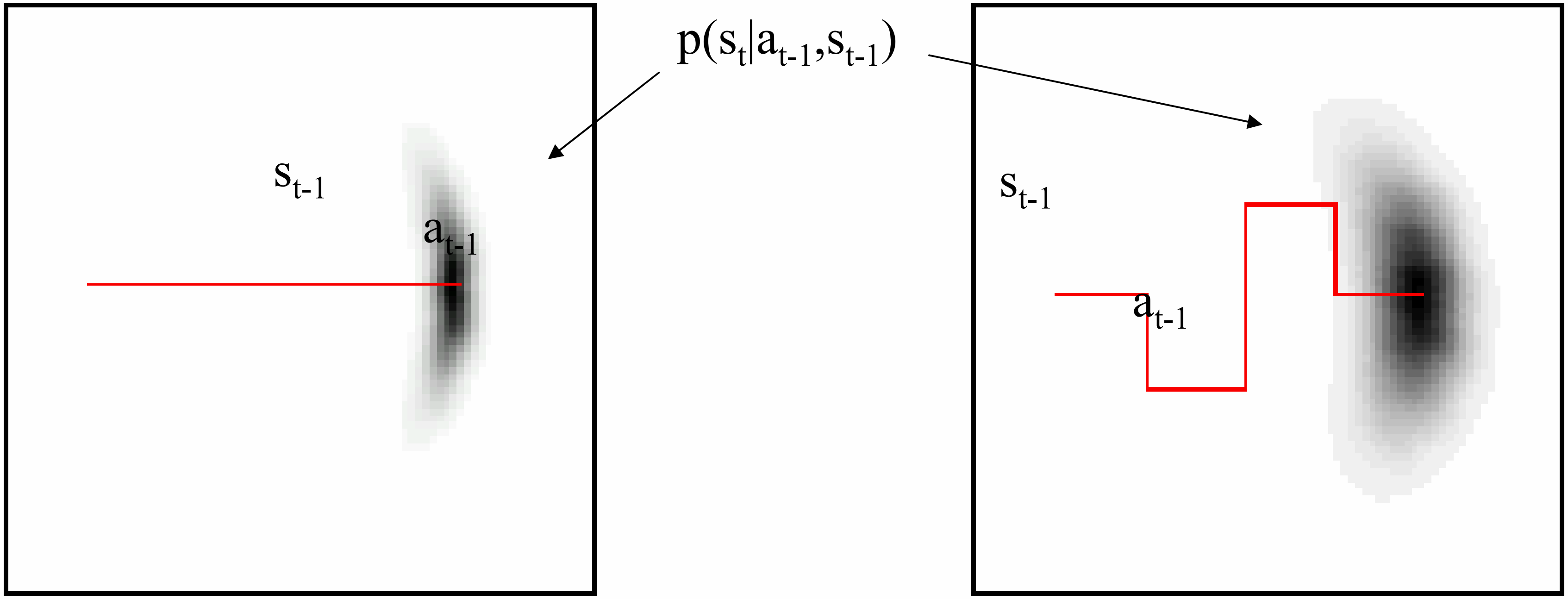
$$d_{0..t} = (a_0, o_0, a_1, o_1, \ldots, a_{t-1}, o_t)$$

Action Data

Observation Data

Probabilistic Action model

- Choose a motion model which considers the error in measuring the robots ego-motion (Odometry)

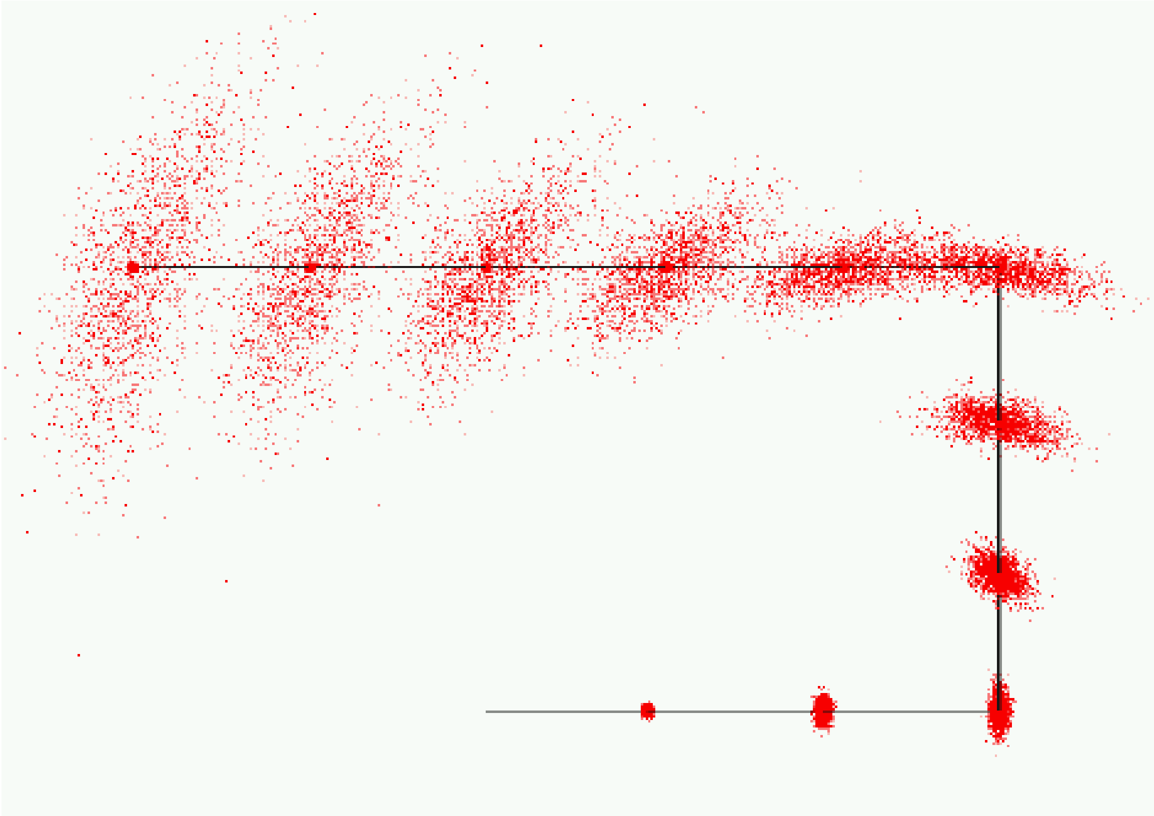


- Continuous probability density *$Bel(s_t)$* after moving 40m (left figure) and 80m (right figure). Darker area has higher probablity.

$$b(s_t \mid m) = \eta\, p(o_t \mid s_t, m) \int p(s_t \mid s_{t-1}, a_{t-1}, m)\, b(s_{t-1} \mid m)\, ds_{t-1}$$

Idea: Represent Belief Through Samples

$$b(s_t \mid m) = \eta\, p(o_t \mid s_t, m) \int p(s_t \mid s_{t-1}, a_{t-1}, m)\, b(s_{t-1} \mid m)\, ds_{t-1}$$



- Particle filters [Doucet 98, deFreitas 98]
- Condensation algorithm [Isard/Blake 98]
- Monte Carlo localization
- [Fox/Dellaert/Burgard/Thrun 99]

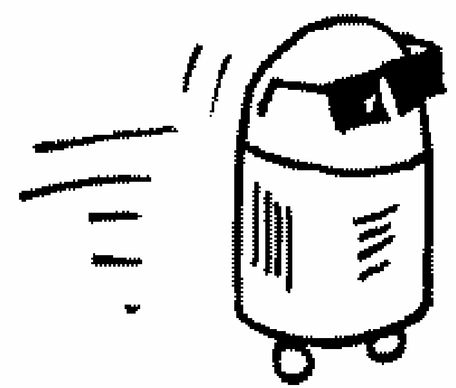
Sampling-based model of position belief

Sampling the Action Model

Localization

Where Am I ?

- Building a map with an accurate set of sensors – Easy!
- Localization with an accurate map – Simple!
- **Fact**: You start off with noisy sensors and no map



The Localization Problem

- Estimate robot's coordinates
- **s = (*x*,*y*,θ)** from sensor data
  - <u>Position tracking</u> (error bounded)
  - <u>Global localization</u> (unbounded error)
  - <u>Kidnapping </u>(recovery from failure)

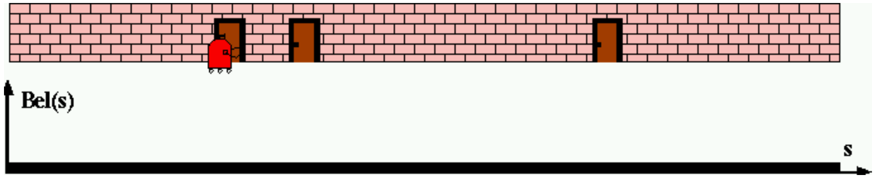
- Ingemar Cox (1991):

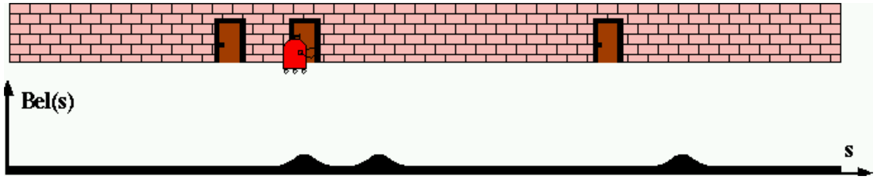
"Using sensory information to locate the robot in its environment is the most fundamental problem to provide a mobile robot with autonomous capabilities."

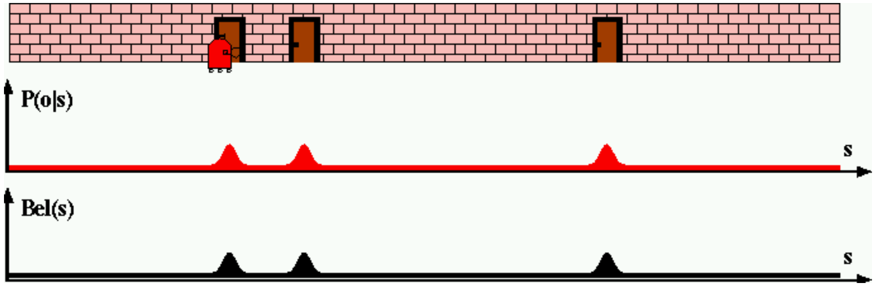
see also [Borenstein et al, 96]

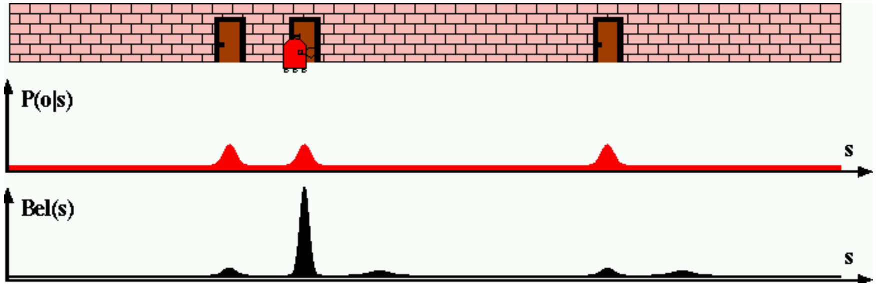
Bayes Filters in Localization

$$p(x_t \mid z_{1..t}, u_{1..t}) \;=\; \eta\, \boxed{p(z_t \mid x_t)} \int p(x_t \mid u_t, x_{t-1})\, p(x_{t-1} \mid z_{1..t-1}, u_{1..t-1})\, dx_{t-1}$$









[Simmons/Koenig 95]
[Kaelbling et al 96]
[Burgard, Fox, et al 96]

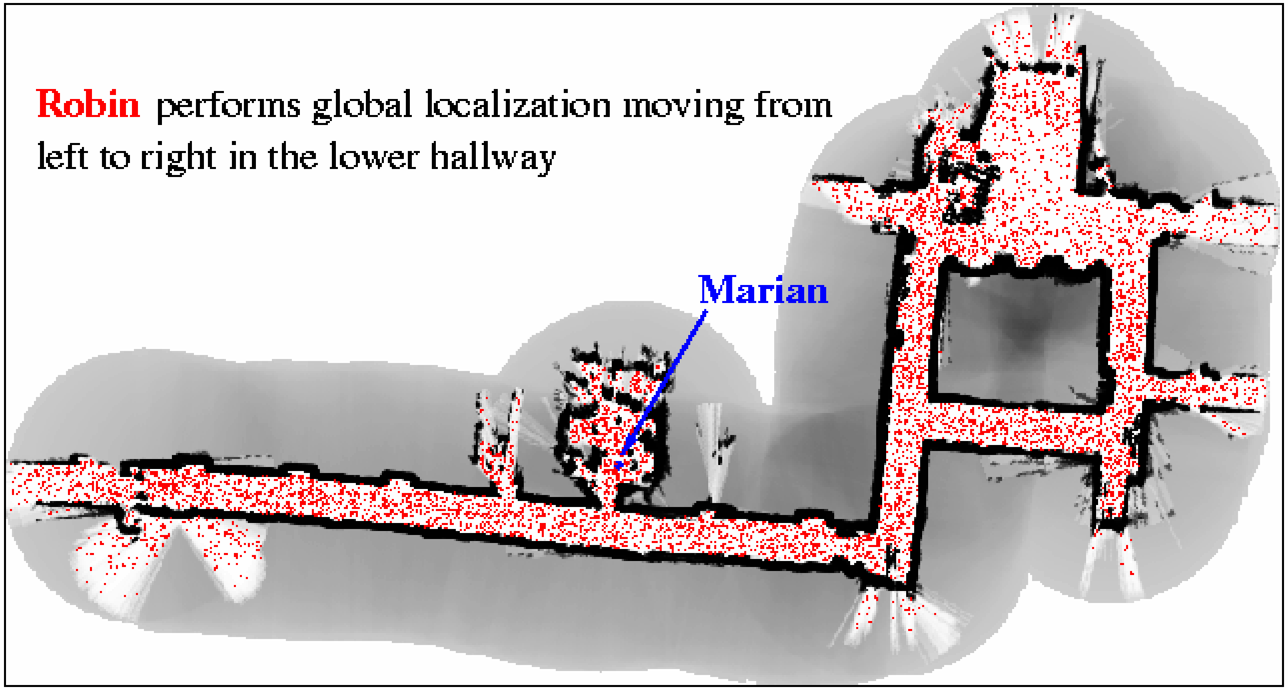
Florence
The Dancing
Robot

Nursebot Pearl

Assisting Nursing
Home Residents

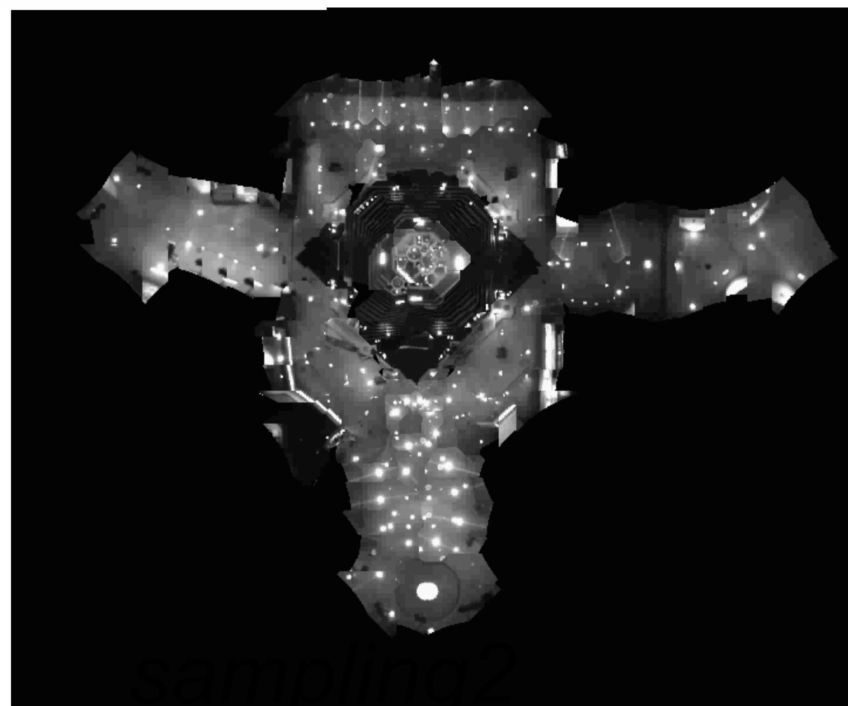
Longwood, Oakdale, May 2001
CMU/Pitt/Mich Nursebot Project

With: Greg Armstrong, Greg Baltus, Jacqueline Dunbar-Jacob, Jennifer Goetz, Sara Kiesler, Judith Matthews, Colleen McCarthy, Michael Montemerlo, Joelle Pineau, Martha Pollack, Nicholas Roy, Jamie Schulte



Animation of two robots for localization

Localization in Museum



Bayes Rule in time

- **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 Filters

- Assume static world (map *m* constant)
- $p(z_t|s_t, m)$ is the sensor model
- $p(s_t|u_t, s_{t-1})$ is the motion model
- $p(s_{t-1}, m|z^{t-1}, u^{t-1})$ is the probability we were where we thought we were last time
- Generally the sensor model and the motion model are static

Derivation : Markov Localization

- d =data
- o =observation
- a = action
- t = time
- s = state

$$Bel(s_t) = \quad p(s_t \mid o_t, a_{t-1}, o_{t-1}, \ldots, o_0)$$

Bayes

$$= \eta\, p(o_t \mid s_t, a_{t-1}, o_{t-1}, \ldots, o_0)\; p(s_t \mid a_{t-1}, o_{t-1}, \ldots, o_0)$$

Markov

$$= \eta\, p(o_t \mid s_t)\; p(s_t \mid a_{t-1}, o_{t-1}, \ldots, o_0)$$

Total Probability

$$= \eta\, p(o_t \mid s_t) \int p(s_t \mid s_{t-1}, a_{t-1}, \ldots, o_0)\; p(s_{t-1} \mid a_{t-1}, \ldots, o_0)\; ds_{t-1}$$

Markov

$$= \eta\, p(o_t \mid s_t) \int p(s_t \mid s_{t-1}, a_{t-1})\; p(s_{t-1} \mid o_{t-1}, a_{t-2} \ldots, o_0)\; ds_{t-1}$$

$$= \eta\, p(o_t \mid s_t) \int p(s_t \mid s_{t-1}, a_{t-1})\; p(s_{t-1} \mid d_{0 \ldots t-1})\; ds_{t-1}$$

$$\boxed{Bel(s_t) = \quad \eta\; p(o_t \mid s_t) \int p(s_t \mid s_{t-1}, a_{t-1})\; Bel(s_{t-1})\; ds_{t-1}}$$

[Kalman 60, Rabiner 85]

The desired posterior is calculated using recursive formula

Markov Assumption

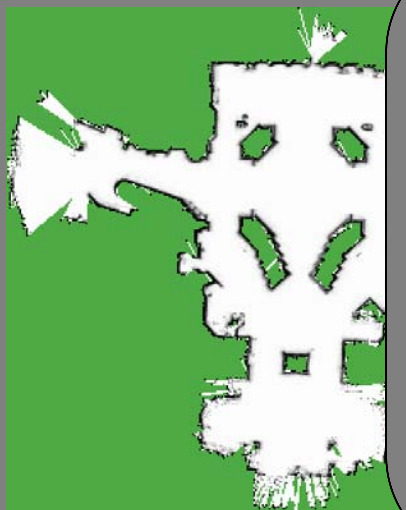
$$\left.\begin{aligned} p(o_t \mid s_t, a_{t-1}, o_{t-1}, \ldots, o_0) &= p(o_t \mid s_t) \\ p(s_t \mid s_{t-1}, a_{t-1}, o_{t-1}, \ldots, o_0) &= p(s_t \mid s_{t-1}, a_{t-1}) \end{aligned}\right\} \text{used above}$$

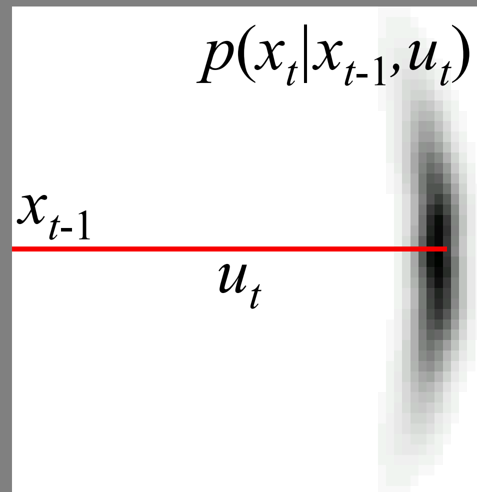
$$\Leftarrow \quad p(o_T, \ldots, o_t, a_{t-1} \ldots, o_0 \mid s_t) = p(o_T, \ldots, o_t \mid s_t)\, p(a_{t-1} \ldots, o_0 \mid s_t)$$

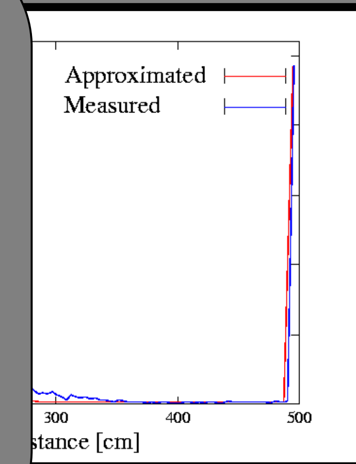
- Knowledge of current state renders past, future independent:
  - “Static World Assumption”
  - “Independent Noise Assumption”

Probabilistic Localization

x = state
t = time
m = map
z = measurement
u = control







$p($ … $x,m)$

Markov

$$= p \ldots p(x_t \mid z_{0\ldots t} \ldots_{\ldots t}, m)$$

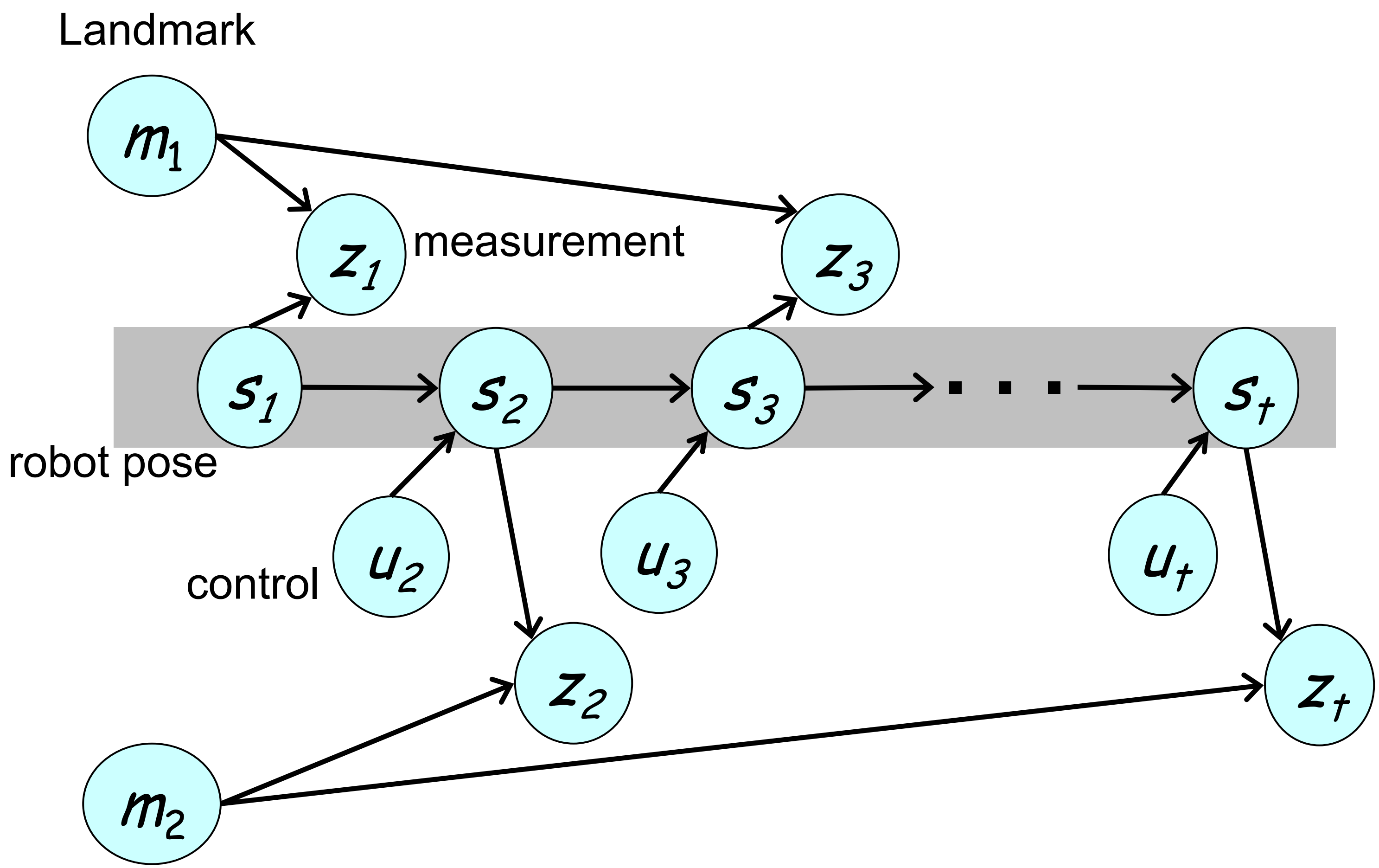
$$= p(z \ldots \int p(x_t \mid x_{t-1}, z_{0\ldots t-1}, u_{0\ldots t}, m)\, p(x_{t-1} \mid z_{0\ldots t-1}, u_{0\ldots t}, m)\, dx_{t-1}$$

Markov

$$= p(z_t \mid x_t, m) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{0\ldots t-1}, u_{0\ldots t-1}, m)\, dx_{t-1}$$

[Kalman 60, Rabiner 85]

Mapping: Structured Generative Model



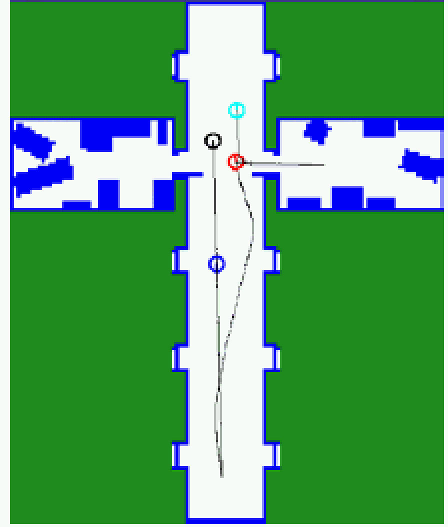
$$p(m, s_{0\ldots t} \mid z_{0\ldots t}, u_{0\ldots t}) = p(s_{0\ldots t} \mid z_{0\ldots t}, u_{0\ldots t}) \prod_{n=1}^{N} p(m_n \mid s_{0\ldots t}, z_{0\ldots t}, u_{0\ldots t})$$

With K. Murphy, B. Wegbreit and D. Koller

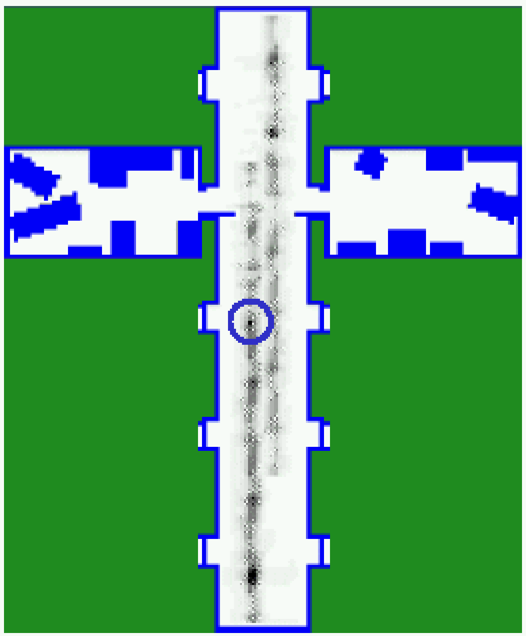
Markov localization

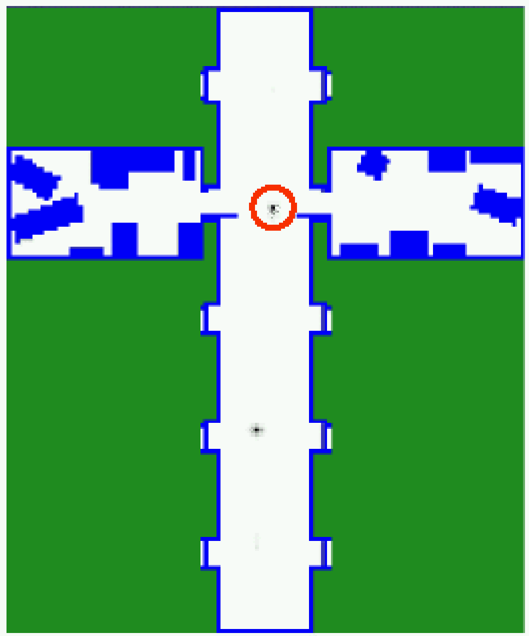
- It equally represents the basic update equation in:
  - Kalman filters,
  - Hidden Markov models,
  - dynamic belief networks.
- Kalman filter represents beliefs by Gaussians
- Vanilla Kalman Filter also assumes Gaussian noise and linear motion equations.
- Applied to tracking and mapping.
- Not good for global localization and kidnapped robot problem.

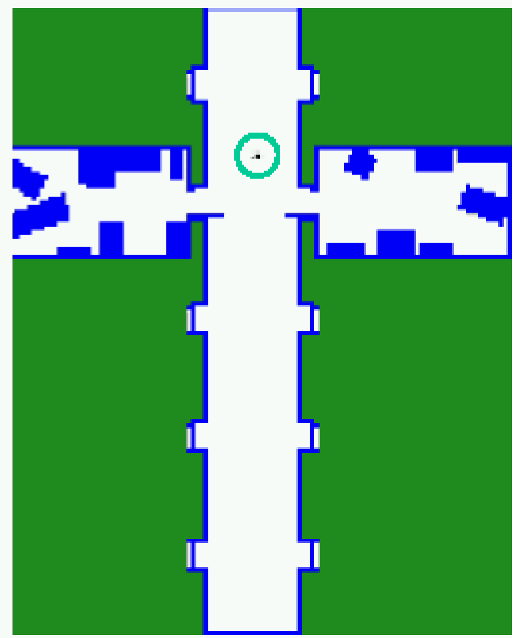
Robot Rhino - CMU: Example of grid-based Markov <u>localization</u> in a symmetric environment based on sonar measurements



Path, highlighting four robot poses







[Burgard et al 96] [Fox 99]

Posterior belief b at second pose

Belief b at third pose

Belief b at fourth pose (<u>robot is certain</u>)

Localization flashback

- **The Kalman Filter**
  - **Concise, closed form equations**
  - **Robust and accurate for tracking position**
  - **Does not handle non-Gaussian or non-linear motion and measurement models**
  - **Restricted sub-optimal extensions with varying success**
- **Topological Markov Localization**
  - **Feature-based localization**
  - **Bayesian Landmark Learning (BaLL)**
  - **Very coarse resolution**
  - **Low accuracy**
- **Grid-based Markov Localization**
  - **Fine resolution by discretizing state space**
  - **Very robust**
  - ***A priori* commitment to precision**
  - **Very high computational burden, with effects on accuracy Sampling-based methods**

Sampling-based Methods

- Invented in the 70's!
- Rediscovered independently in target-tracking, statistical and computer vision literature

- Bootstrap filter
- Monte-Carlo filter
- Condensation algorithm

Particle Filters

Monte Carlo Localization

- Probabilistic Localization = Bayes filters
- Particle filters: Approximate posterior by random samples

- Approximate Bayes Filtering
  - Full posterior estimation
  - Converges in O(1/□ã#samples)[Tanner'93]
  - **Robust:** multiple hypothesis with degree of belief
  - **Any-time:** by varying number of samples
  - Easy to implement

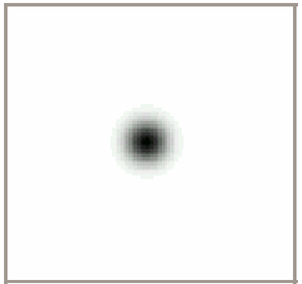
Monte-Carlo Localization

- Represent the probability density $Bel(s_t)$ by a set of randomly drawn samples
- From samples, we can always approximately reconstruct density (e.g. histogram)
- **Reason:** The discrete distribution defined by the sample will approximate the desired one.
- **Goal:** Recursively compute at each time instance *t* the set of samples S_t that is drawn from $Bel(s_t)$

Algorithm: Prediction phase

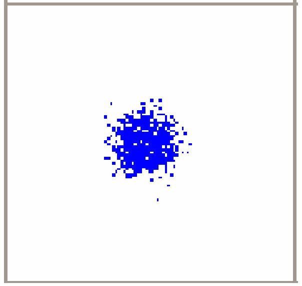
1. Draw a random sample S_{t-1} from the current belief $Bel(s_{t-1})$

continuous



$Bel(s_{t-1})$

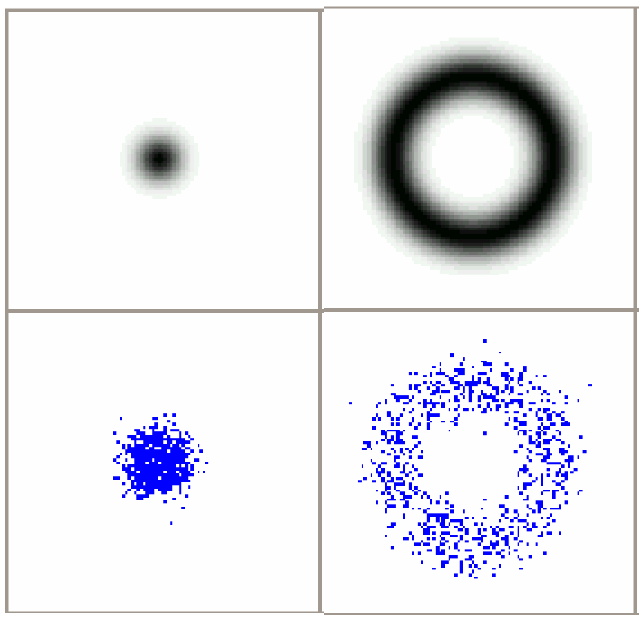
discrete



S_{t-1}

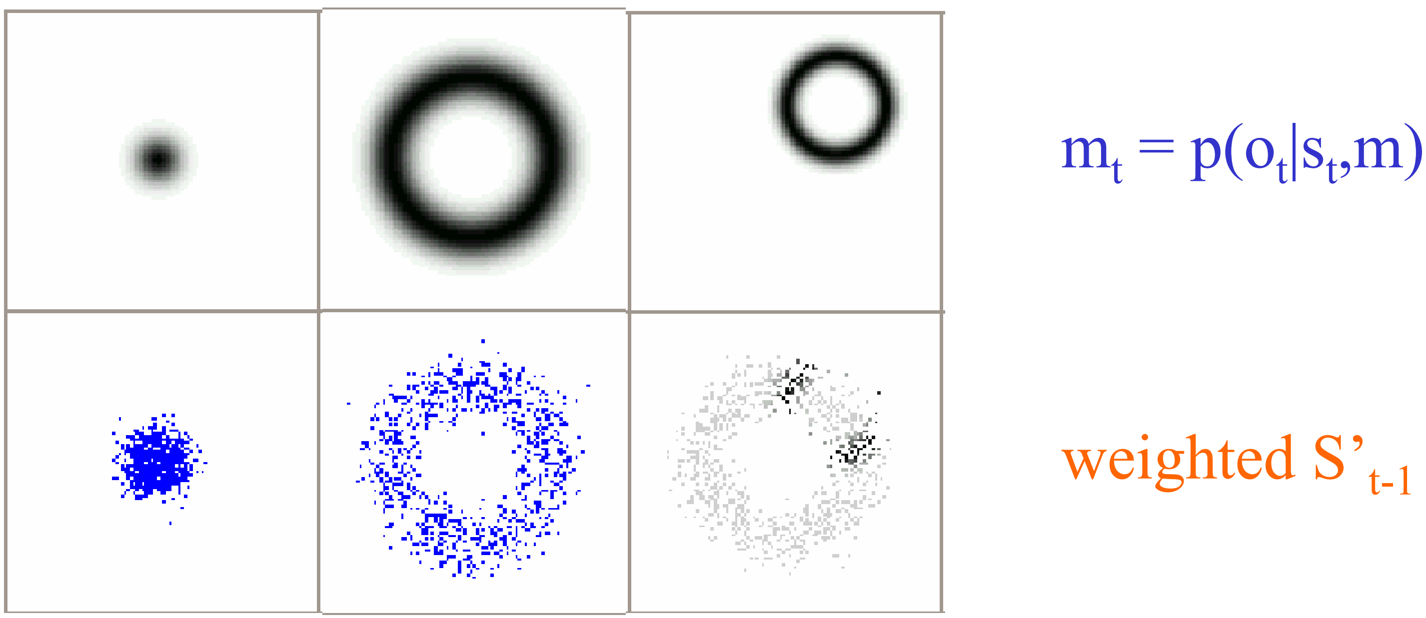
Algorithm: Update phase - I

2. For this S_{t-1}, guess a set of successor poses s_t, as per the distribution $p(s_t|a_{t-1},s_{t-1},m)$ to form S'_{t-1}



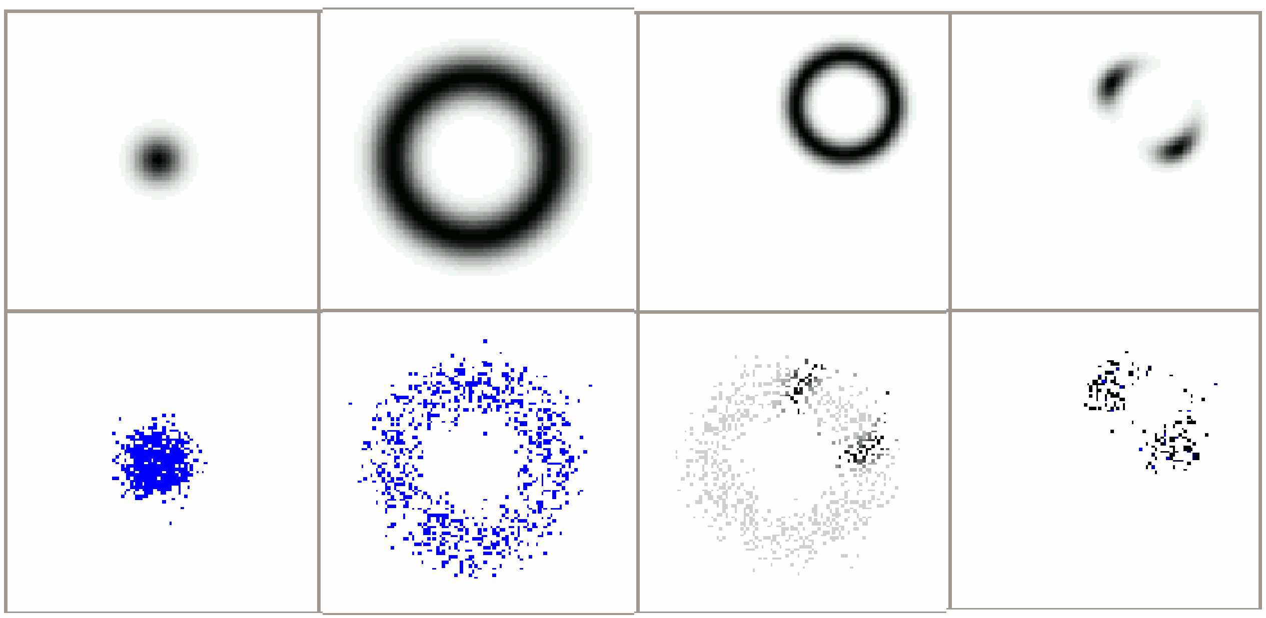
Algorithm: Update phase - II

3. Weight each sample in S'_{t-1} by $m_t = p(o_t|s_t,m)$, or what is called the importance factor.



Algorithm: Resampling

4. Draw each element s'^{j}_{t-1} in S'_{t-1} with probability equal to its weight m^j to form the new set S_t

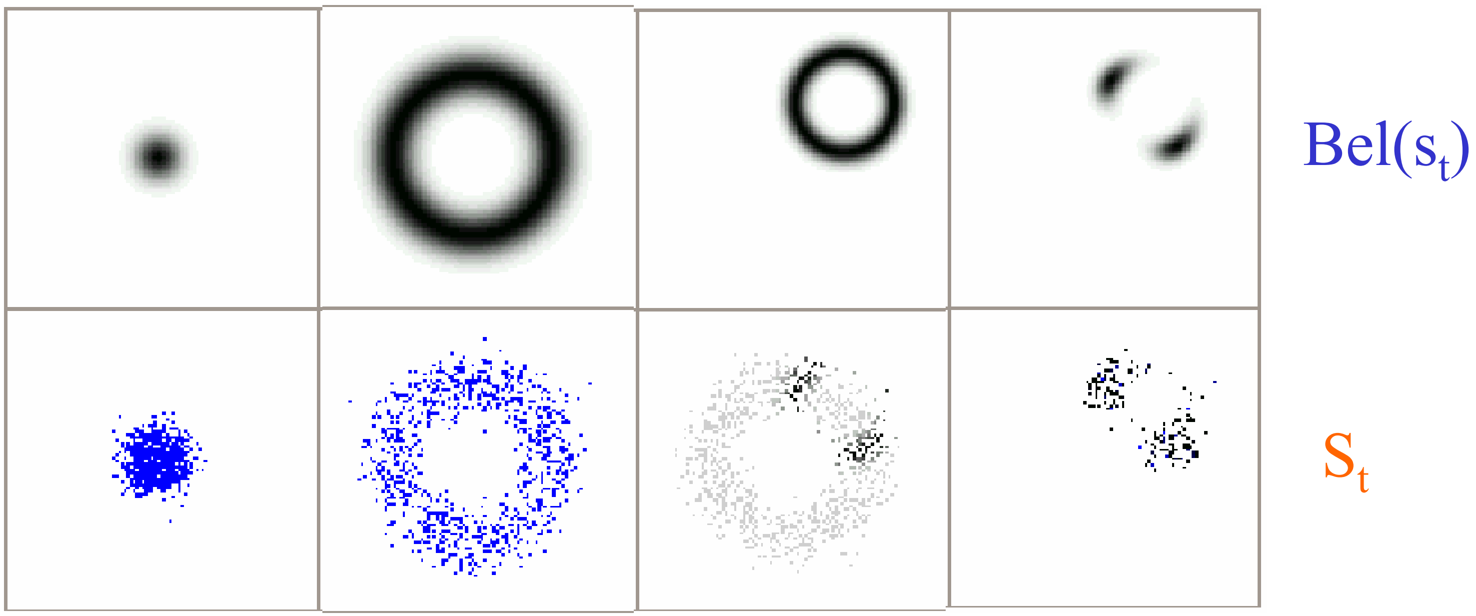


$Bel(s_t)$

S_t

Algorithm

5. Normalize the importance factors and repeat from (2).



Justification

- Predictive phase retrieves an empirical predictive density (stratified sampling) that approximates the real one.
- Update phase retrieves an empirical posterior density (importance sampling) by weighting more likely states.

The entire procedure is called Sampling / Importance Resampling (SIR)

Monte-Carlo Localization (MCL)

- Solves the global localization and kidnapped robot problem
- Multi-modal (unlike the Kalman filter)
- Drastic reduction in memory requirement
- More accurate than ML with a fixed cell size
- Easy to implement
- Fast

- References:
  - AAAI Tutorial on Probabilistic Robotics (Sebastian Thrun)
  - Probabilistic Algorithms in Robotics (Thrun)
  - Robust Monte Carlo Localization for Mobile Robots (Thrun, Fox, Burgard, Dellaert)
  - Monte Carlo Localization for Mobile Robots (Dellaert, Fox, Burgard, Thrun)

Monte Carlo Localization

- 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

SLAM Thrun, Sebastian. "Animation of On-line mapping with Monte Carlo Localization"

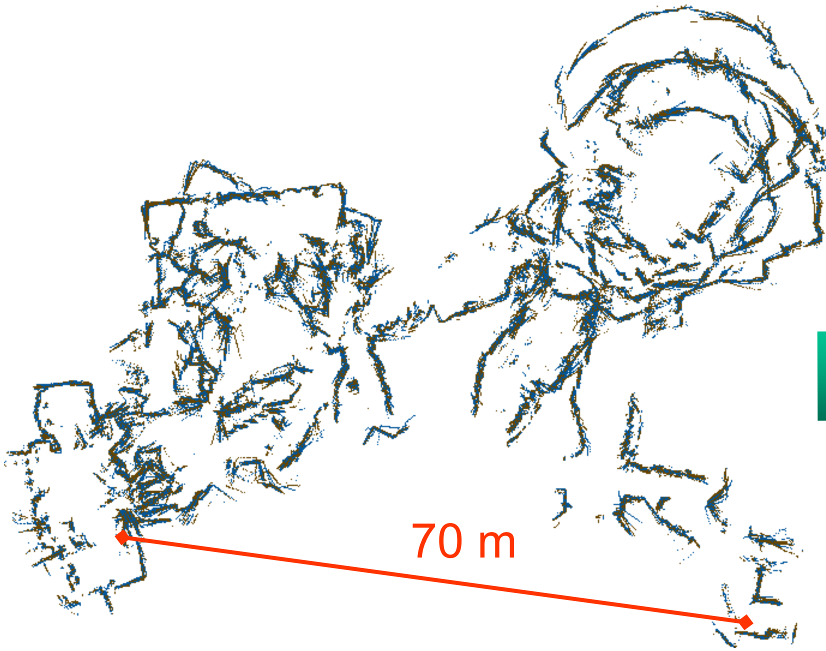
Thrun, Sebastian. "Animation of Monte Carlo Localization using laser range finders"

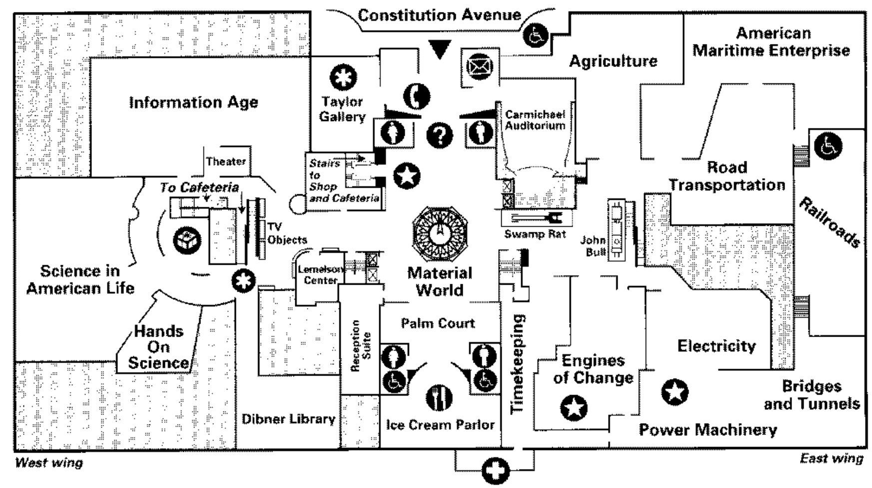
SLAM

- 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)

Learning Maps

aka Simultaneous Localization and Mapping (SLAM)

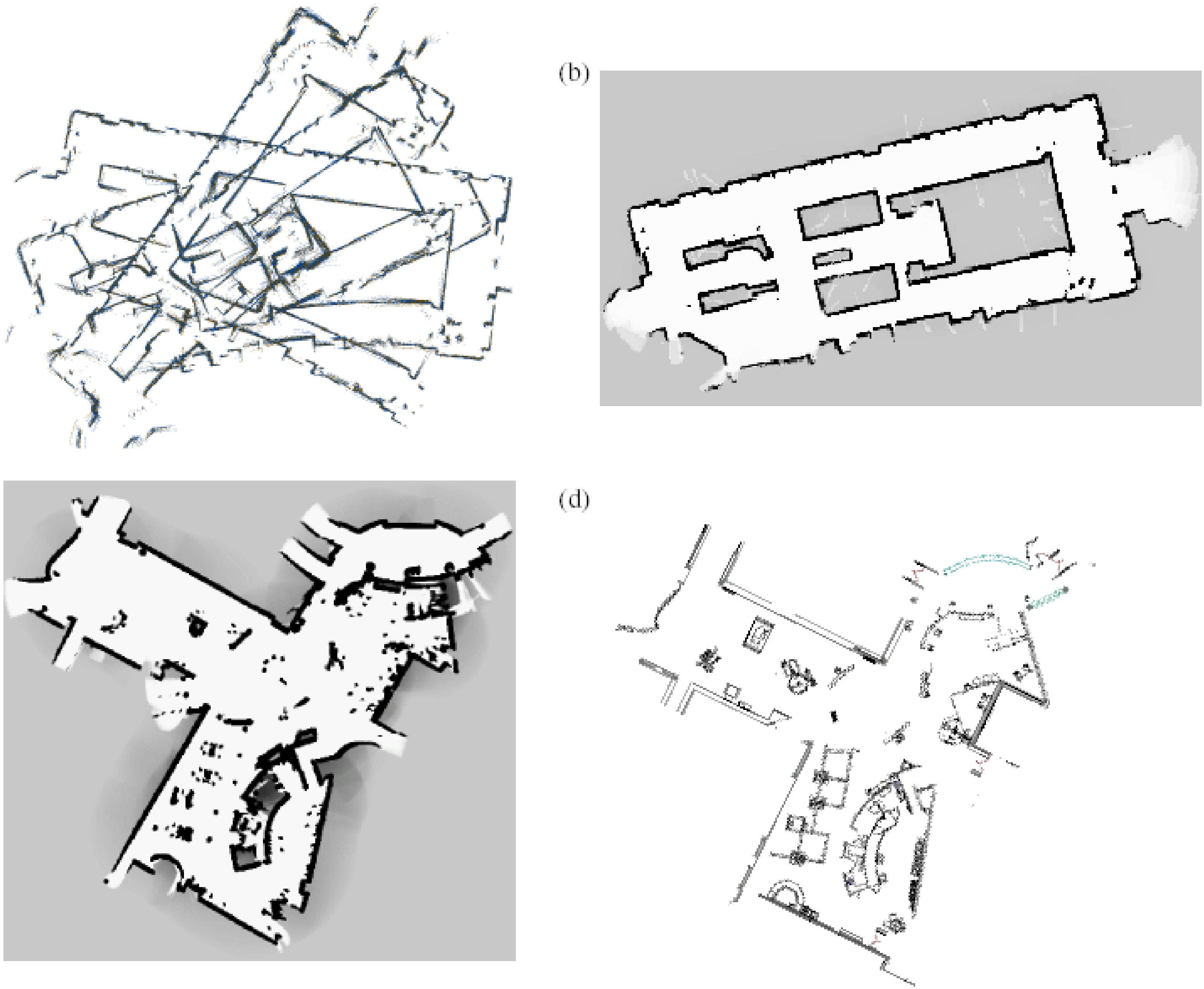




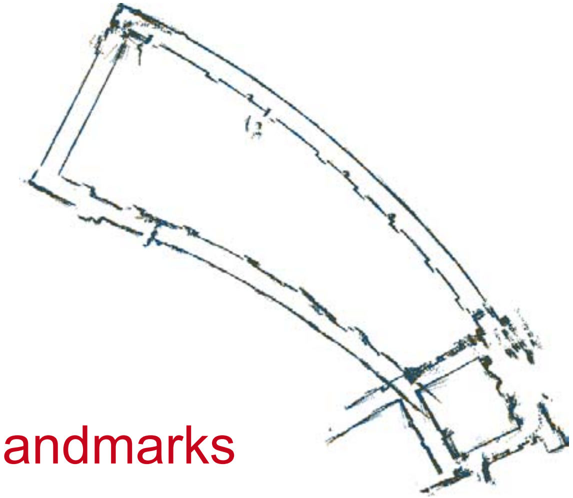
Mapping: The Problem

- Continuous variables
- High-dimensional (eg, 1,000,000+ dimensions)
- Multiple sources of noise
- Simulation not acceptable

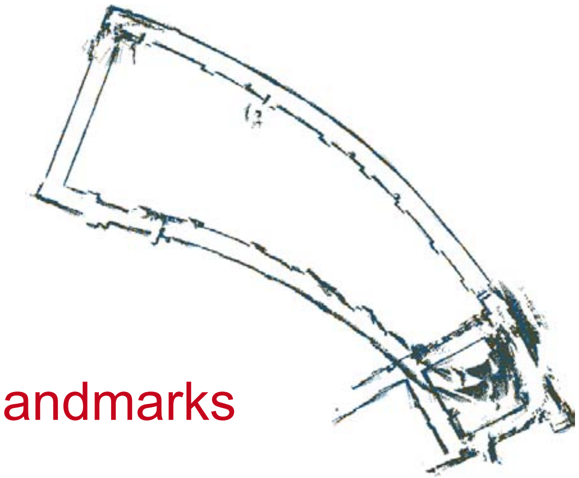
Results of Mapping



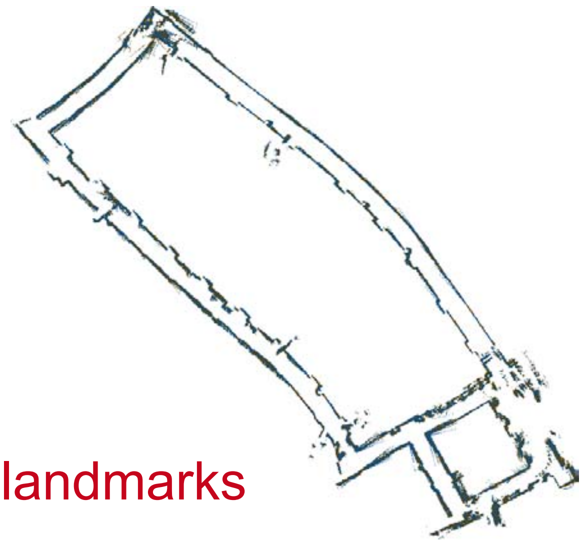
CMU’s Wean Hall (80 x 25 meters)



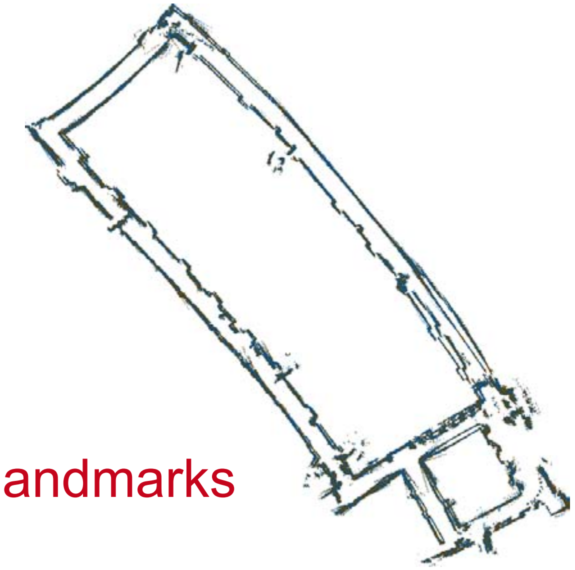
15 landmarks



16 landmarks

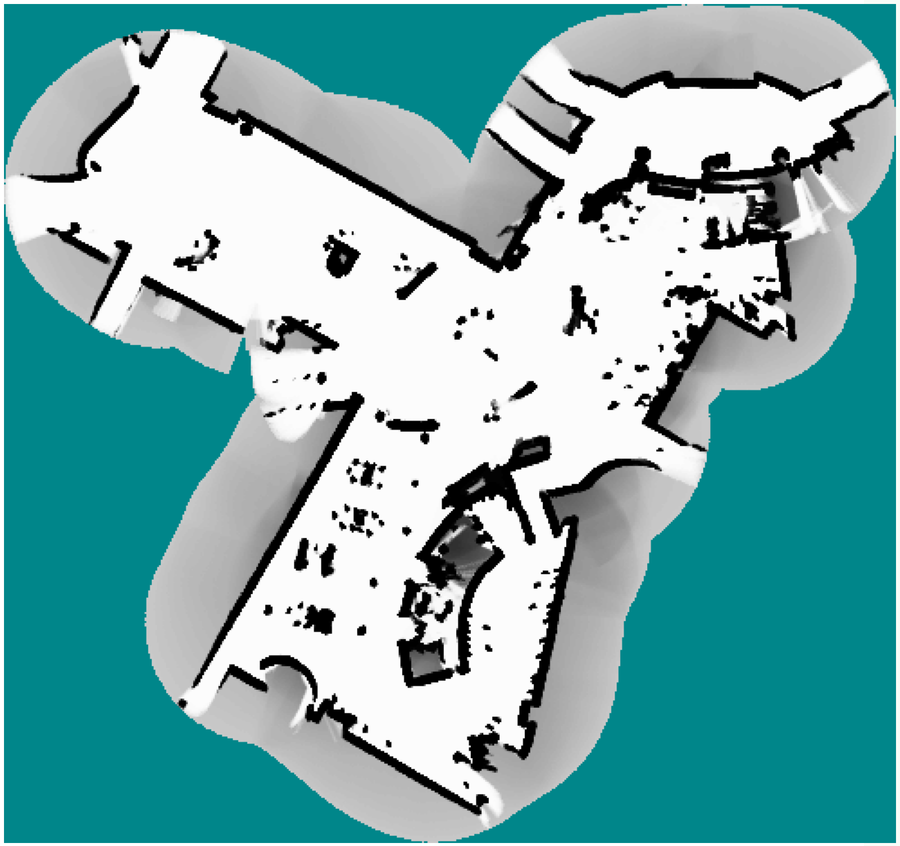


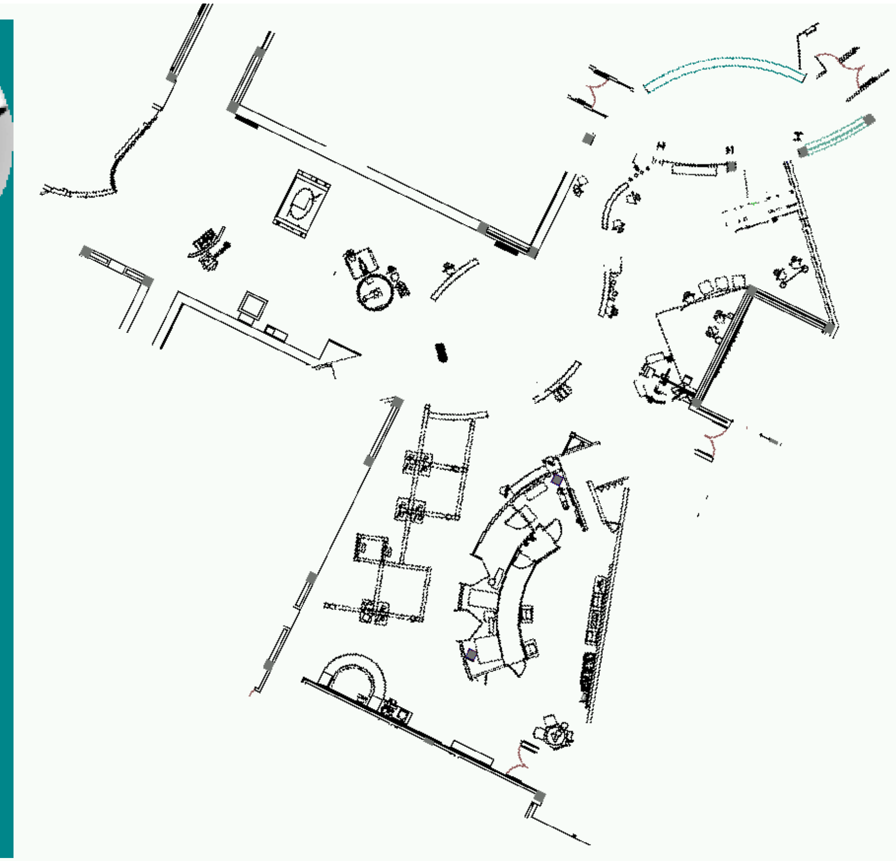
17 landmarks



27 landmarks

Accuracy: “The Tech” Museum, San Jose





2D Map, learned

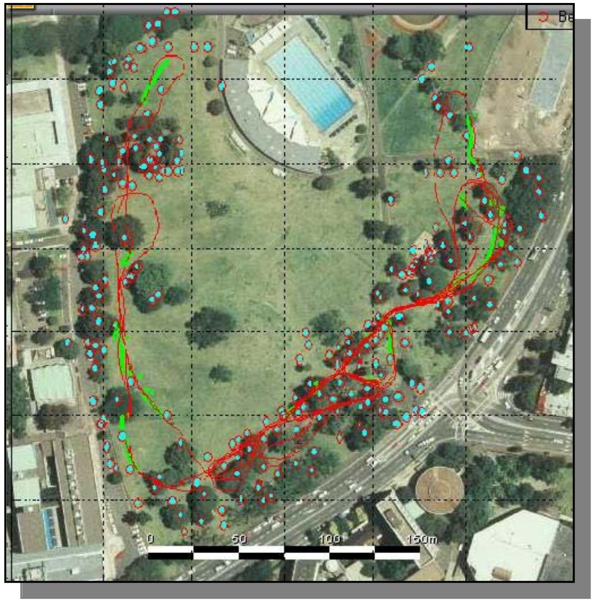
CAD map

Animation of On-line Mapping with Monte Carlo Localization

Multi-Robot Mapping -
Animation of 3D Map of Wean Hall

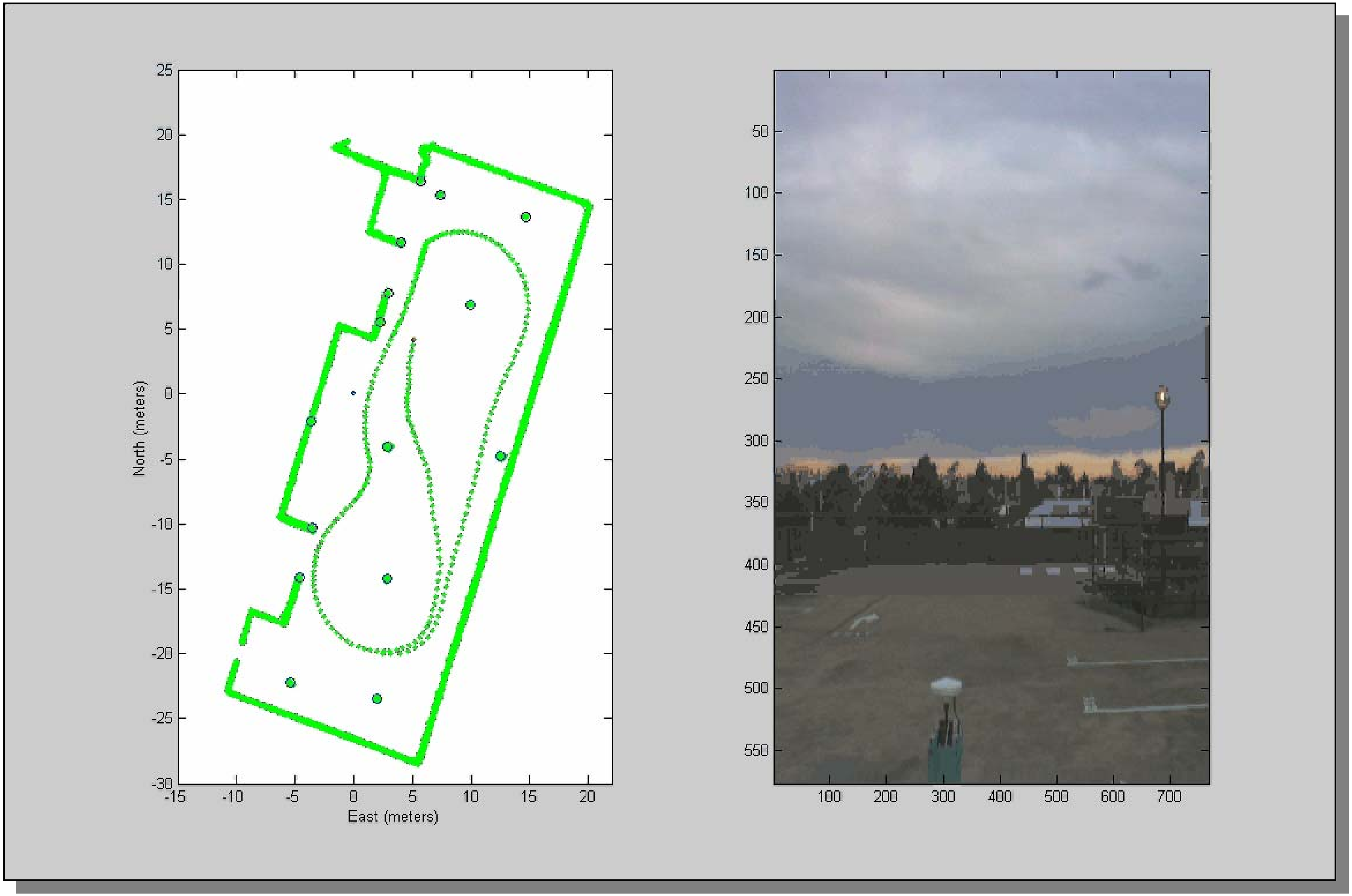
Multi-Robot Exploration with Monte Carlo Localization

Outdoor Mapping (no GPS)



With Juan Nieto, Jose Guivant, Eduardo Nebot, Univ of Sydney

Outdoor Mapping (no GPS)



Sebastian Thrun, Juan Nieto, Jose Guivant, Eduardo Nebot, Univ of Sydney

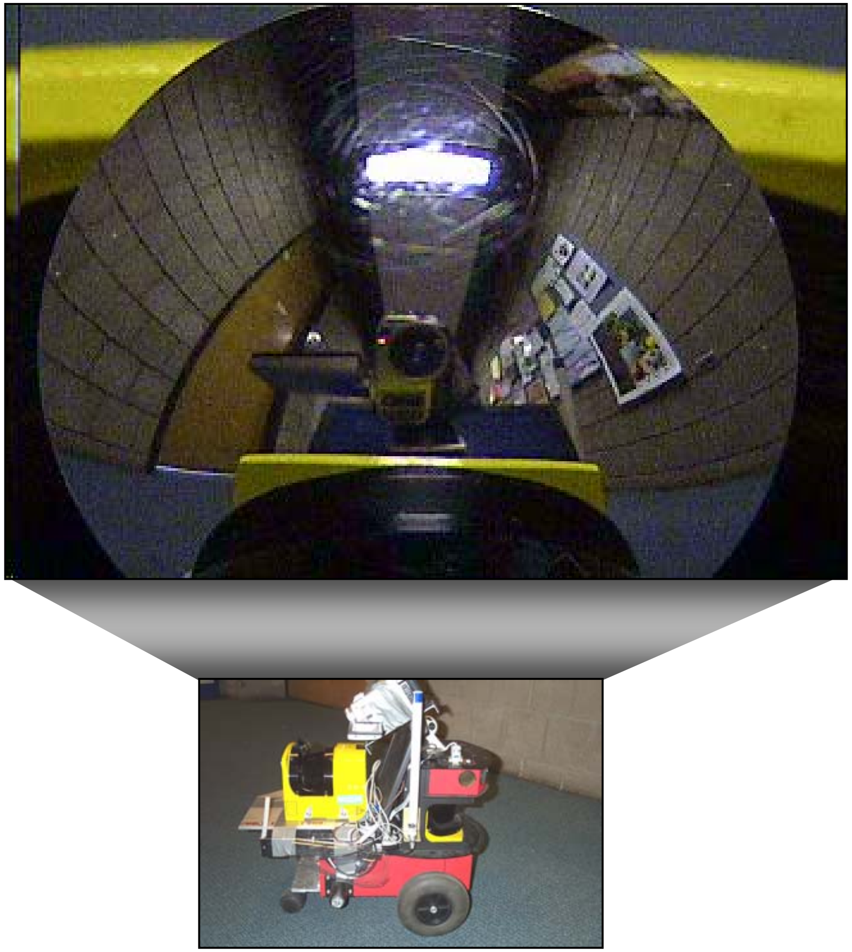
3D Mapping

Moravec et al, 2000

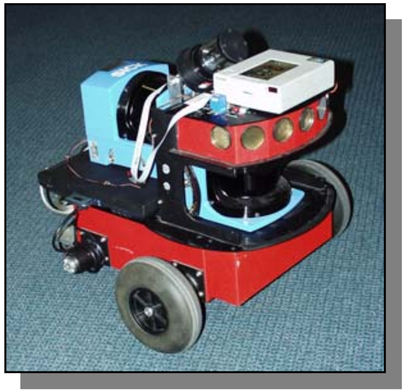
Konolige et al, 2001

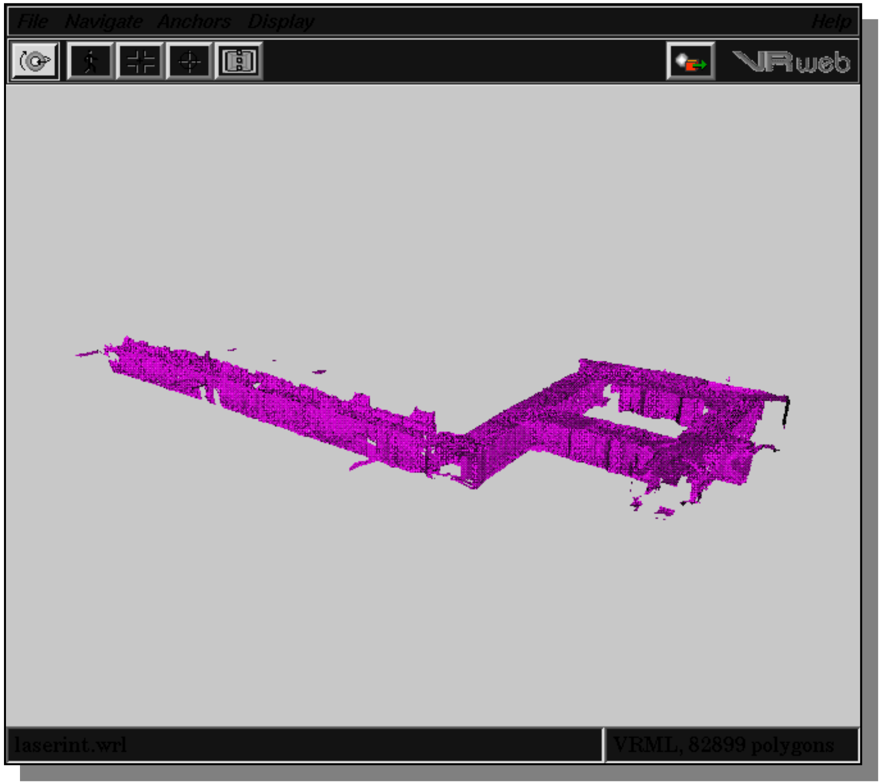
Teller et al, 2000

3D Volumetric Mapping

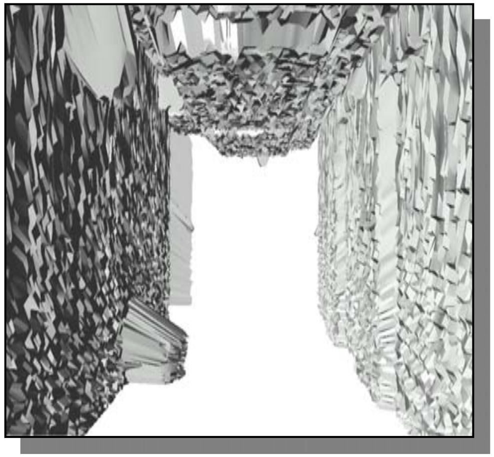


Learning Object Models





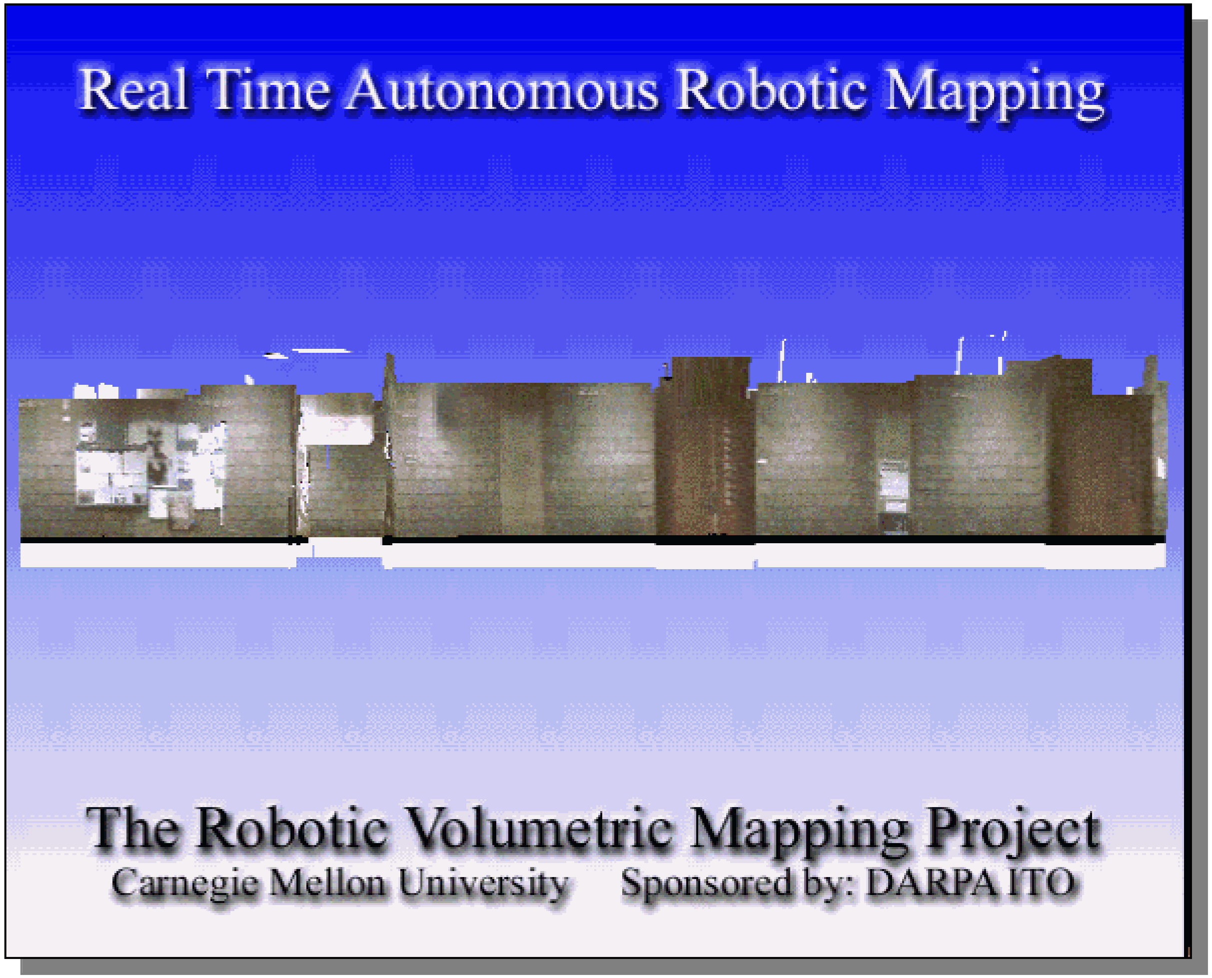
Nearly Planar Maps



Idea: Exploit fact that buildings posses many planar surfaces

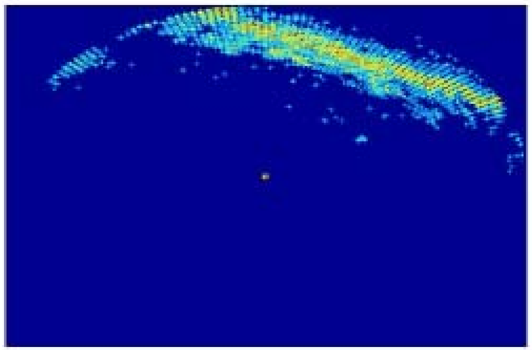
- Compacter models
- Higher Accuracy
- Good for capturing environmental change

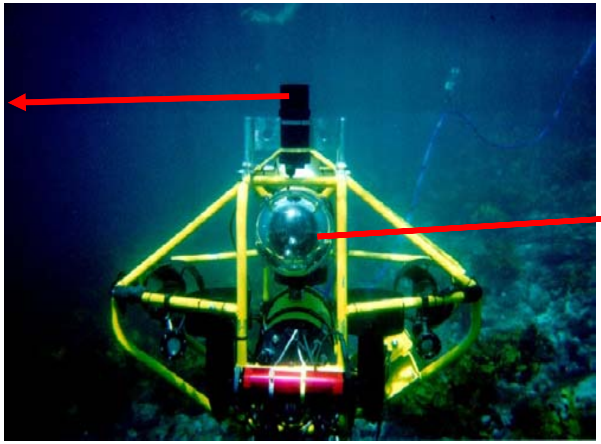
3D Mapping Result

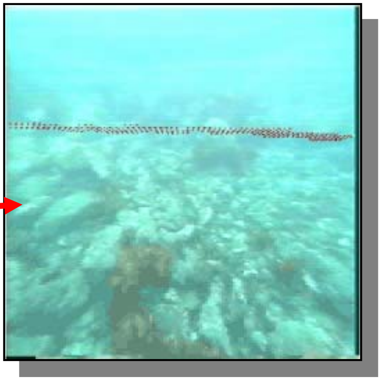


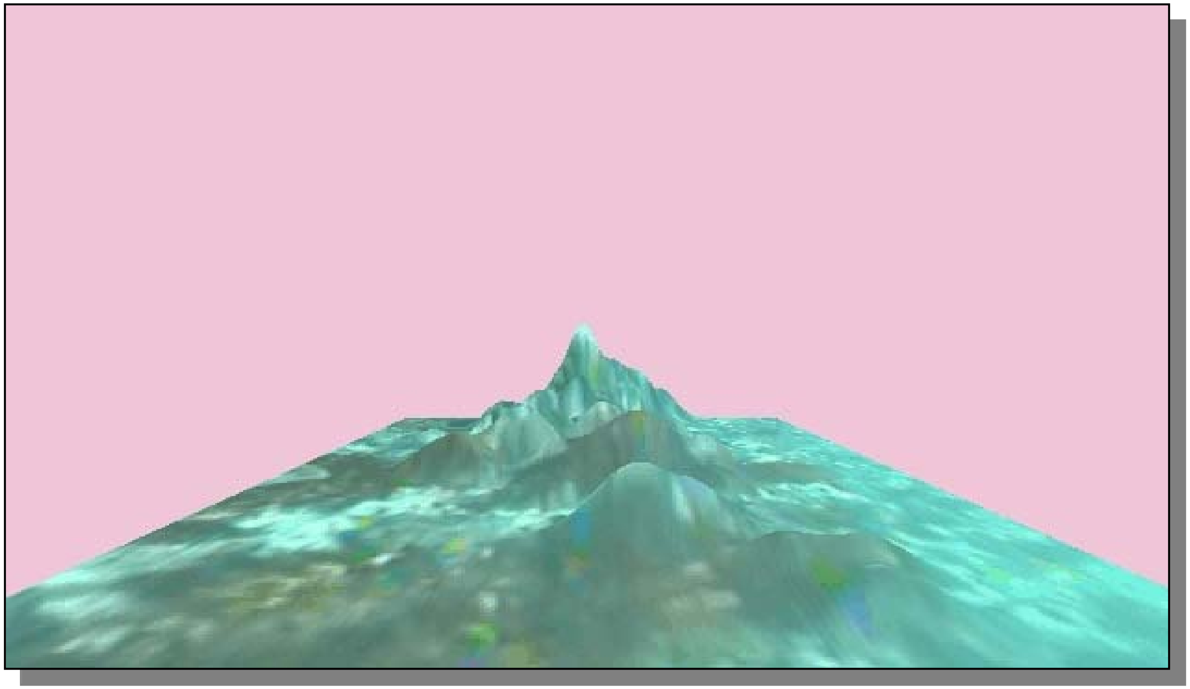
Sebastian Thrun, Christian Martin

Underwater Mapping (with University of Sydney)









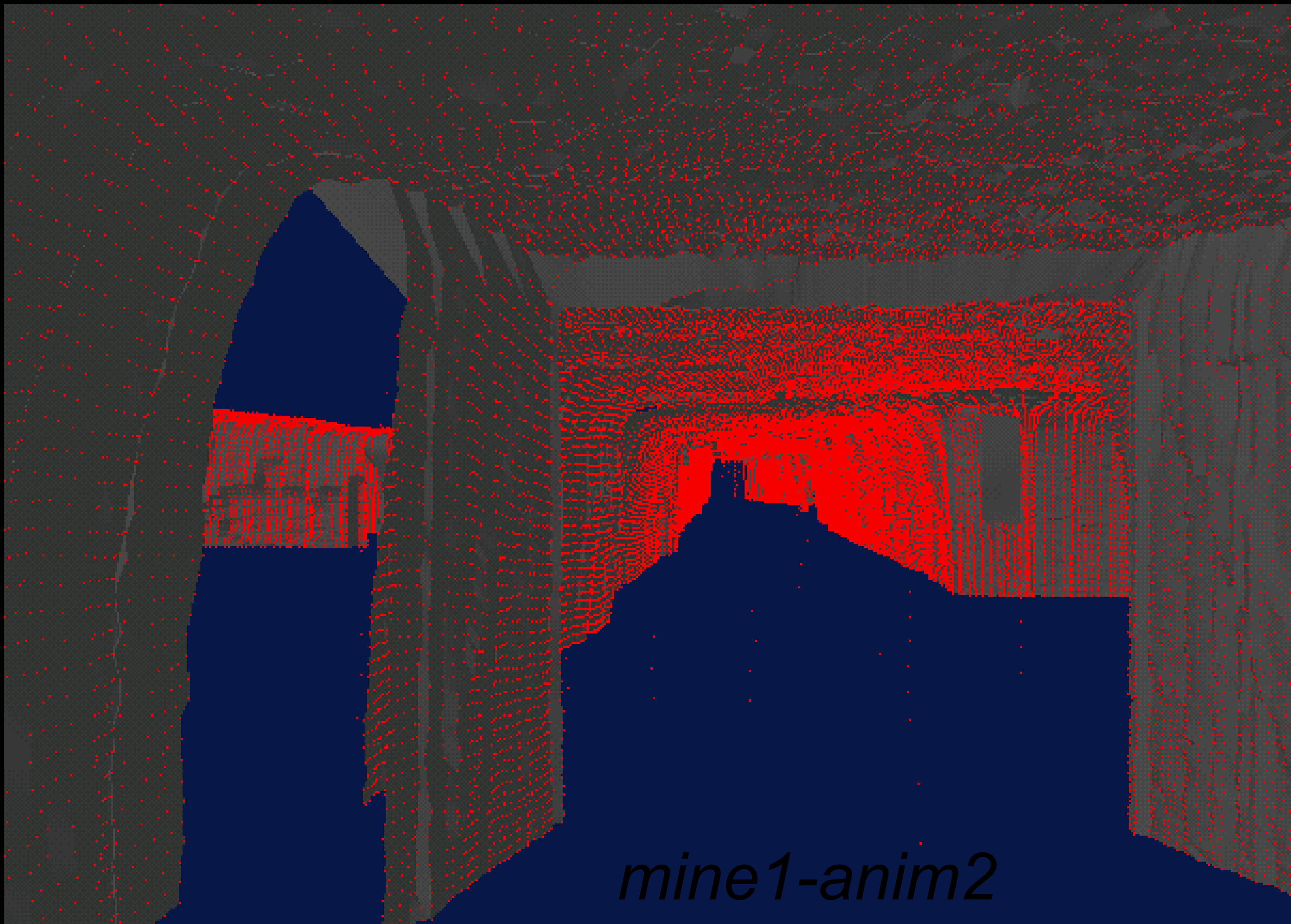
Sebastian Thrun, Hugh Durrant-Whyte, Somajyoti Majunder, Marc de Battista, Steve Scheding

Robotic Mine Mapping Project

mine1-video

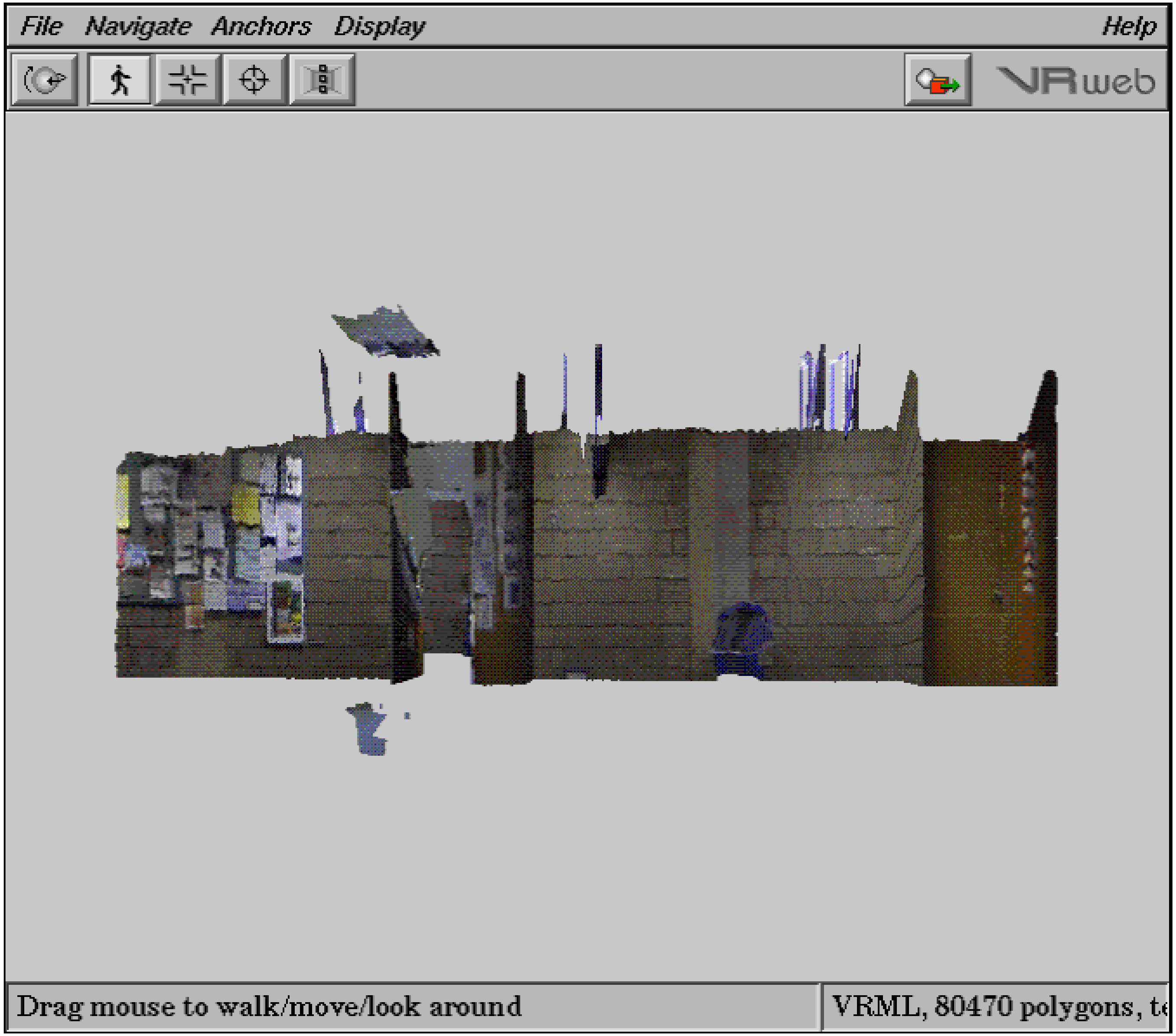
Mine Mapping

Mine Mapping Project

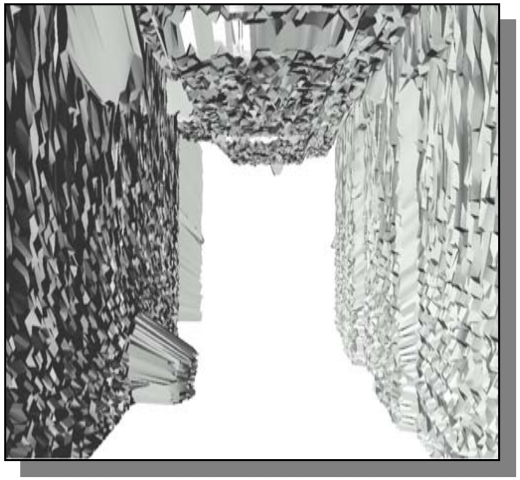


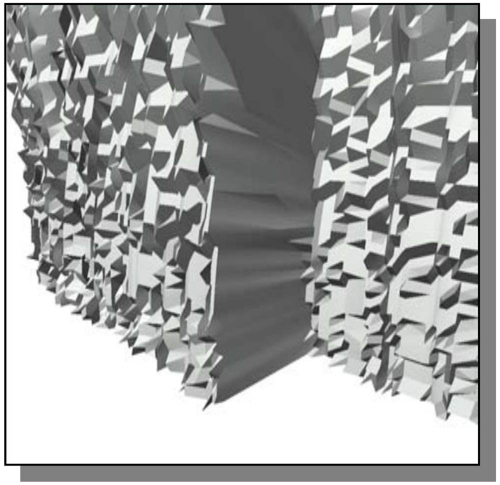
Mine Mapping

3D Texture Mapping



Fine-Grained Structure:
Can We Do Better?

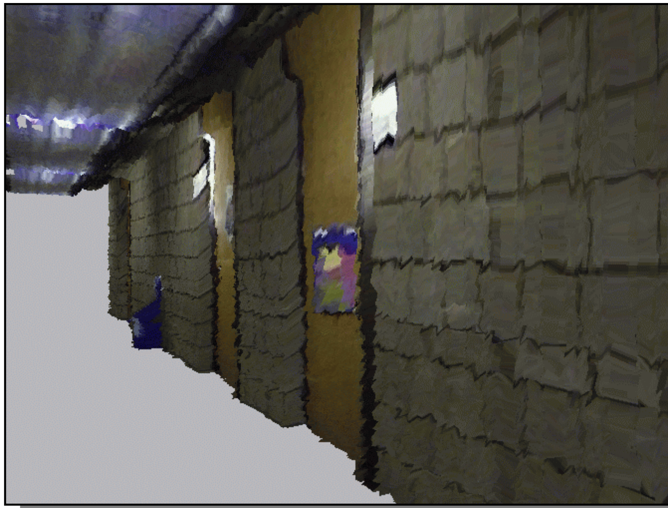


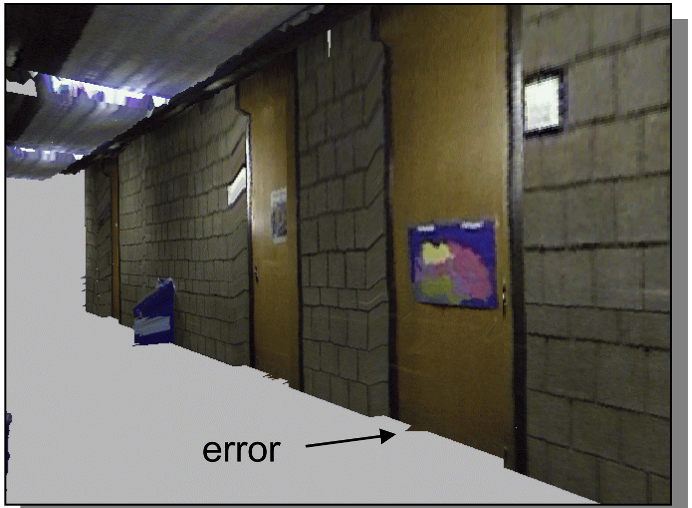


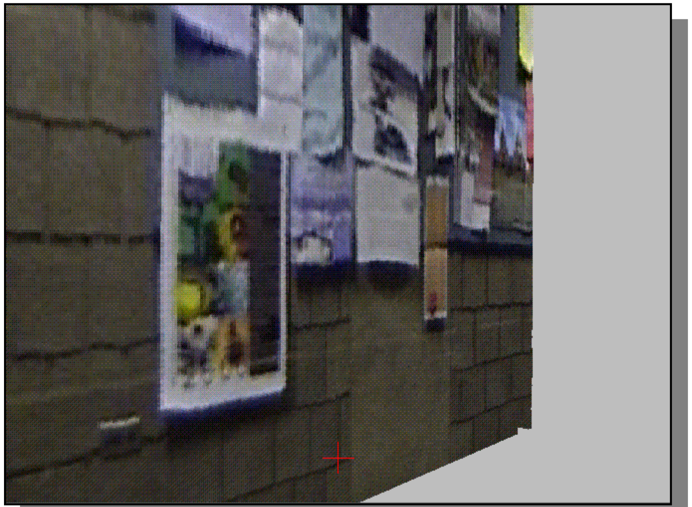
Results

Without EM

With EM
(95% of data explained by 7 surfaces)



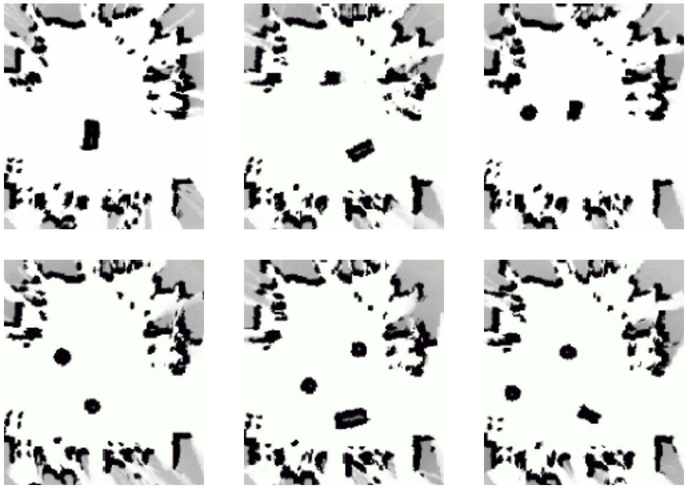


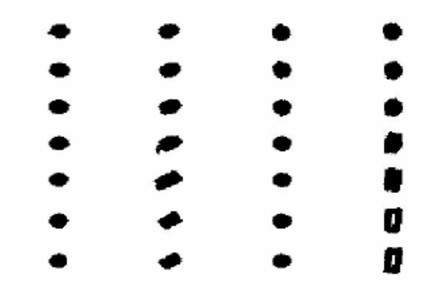


With: Deepayan Chakrabarti, Rosemary Emery, Yufeng Liu, Wolfram Burgard, ICML-01

Dynamic Environments

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





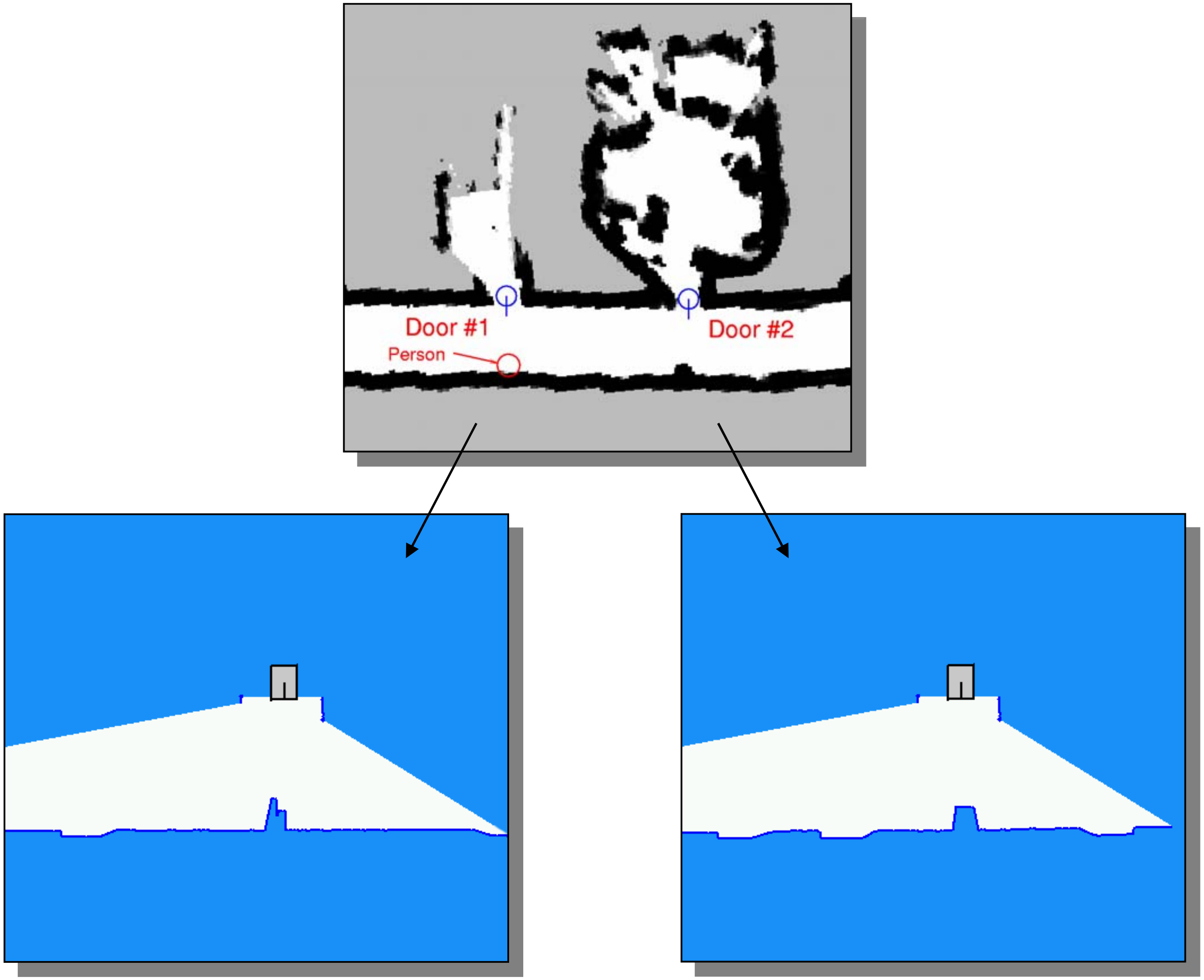
Combining Tracking and Mapping

With Dirk Hähnel, Dirk Schulz and Wolfram Burgard

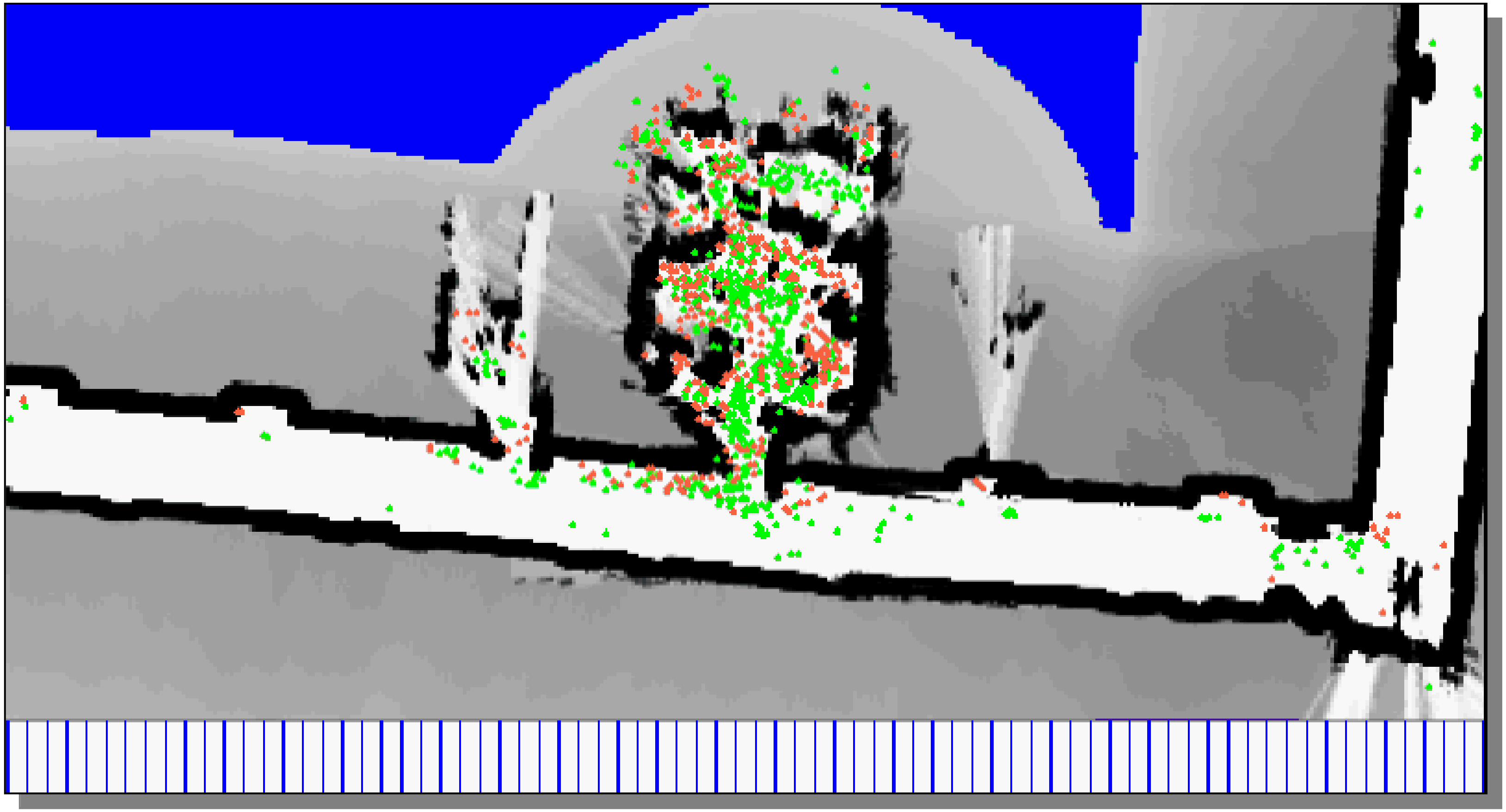
Tracking Moving Features

With: Michael Montemerlo

Tracking Moving Entities Through Map Differencing

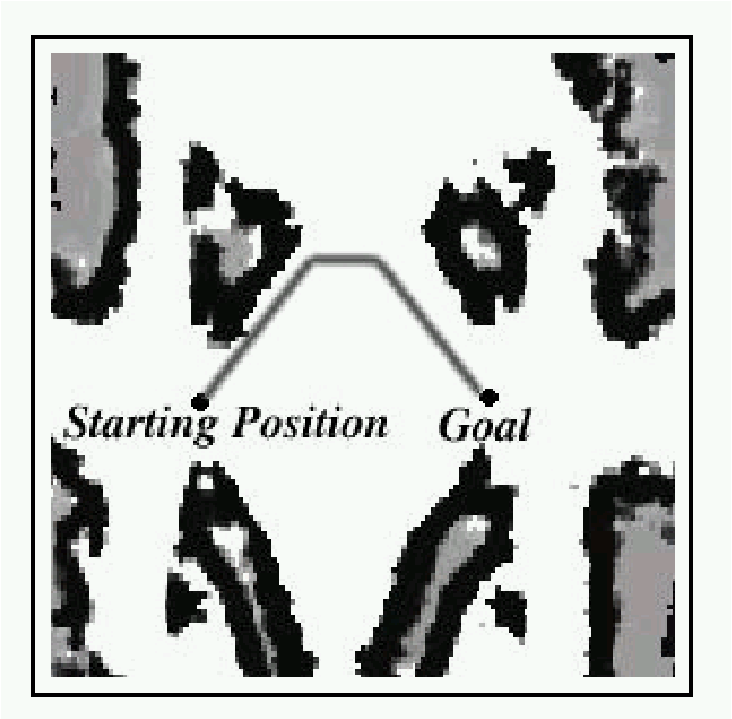


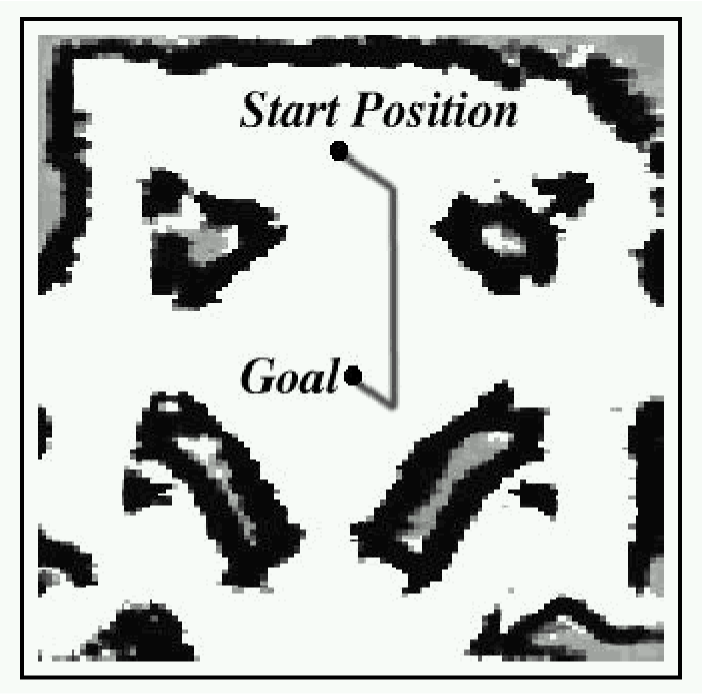
Map-Based People Tracking



With: Michael Montemerlo

Robot Control





- Coastal plans: the robot actively seeks the proximity of obstacles to improve its localization.
- The large open area in the center of this Smithsonian museum is approximately 20 meters wide and usually crowded with

Robot Puppets - Natural Language Dialogs and Emotional Behavior Animation

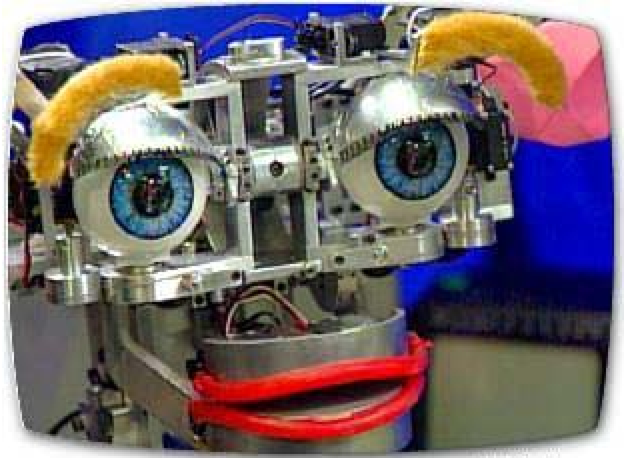
Dog.com from Japan

- Many talking toys exist, but they are still very primitive
- Actors for robot theatre, agents for advertisement, education and entertainment.
- Designing inexpensive natural size humanoid caricature and realistic robot heads

Machine Learning techniques used to teach robots behaviors, natural language dialogs and facial gestures.

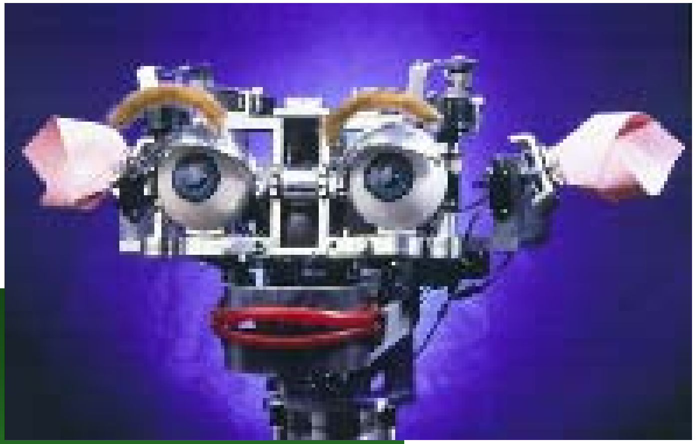
Probabilistic techniques

Robot with a Personality?



- Future robots will interact closely with non-sophisticated users, children and elderly, so the question arises, how they should look like?
- If human face for a robot, then what kind of a face?
- Handsome or average, realistic or simplified, normal size or enlarged?

•The famous example of a robot head is Kismet from MIT.



• Why is Kismet so successful?

•We believe that a robot that will interact with humans should have some kind of "personality" and Kismet so far is the only robot with "personality".

Behavior, Dialog and Learning

Words communicate only about 35 % of the information transmitted from a sender to a receiver in a human-to-human communication.

The remaining information is included in <u>para-language.</u>

Emotions, thoughts, decision and intentions of a speaker can be recognized earlier than they are verbalized.

Probabilistic techniques

Robot face should be friendly and funny

The Muppets of Jim Henson are hard to match examples of puppet artistry and animation perfection.

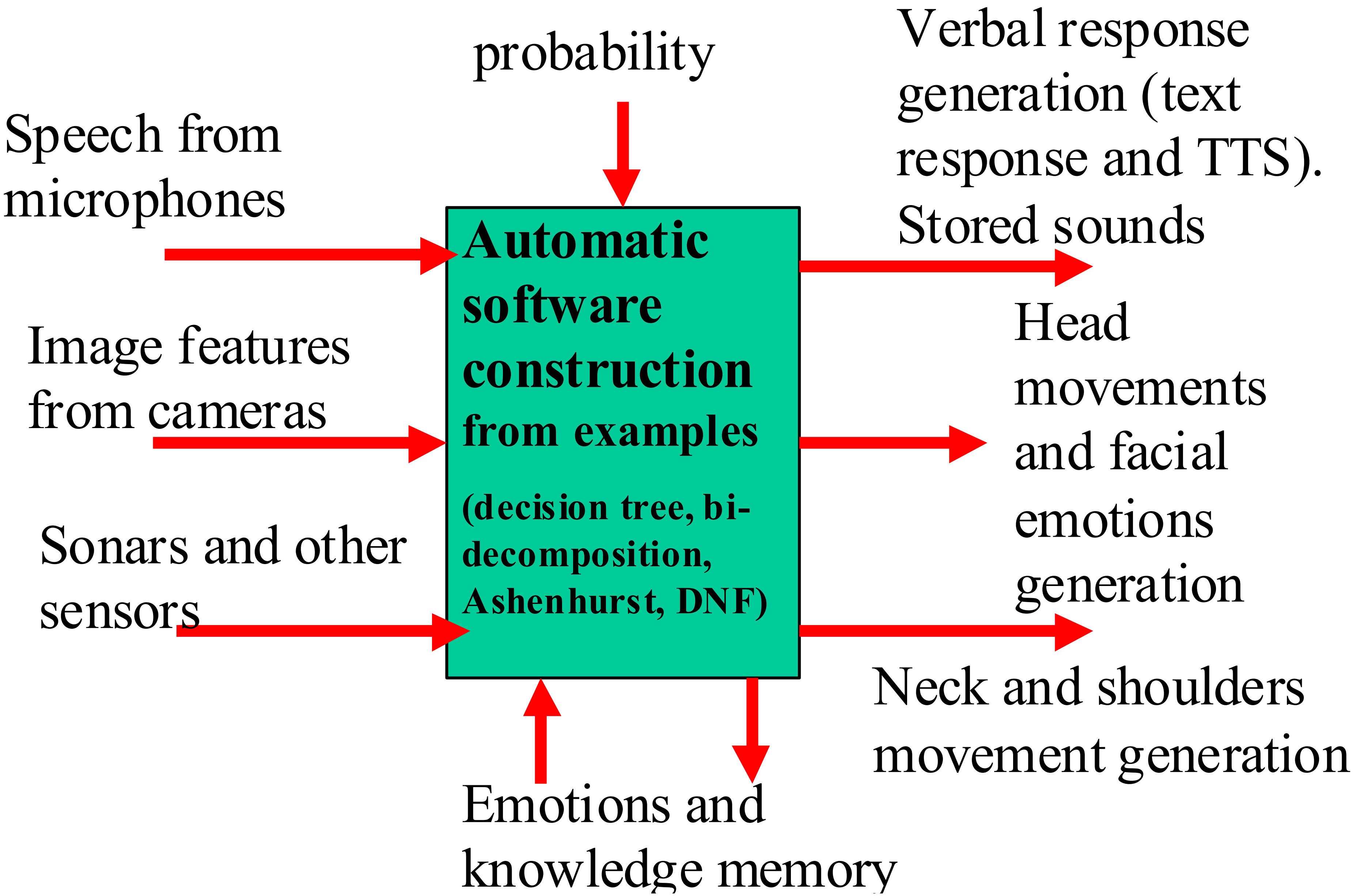
We are interested in robot's personality as expressed by its:

- behavior,
- facial gestures,
- emotions,
- learned speech patterns.

- Robot activity as a ***mapping*** of the sensed environment and internal states to behaviors and new internal states (emotions, energy levels, etc).
- Our goal is to ***uniformly integrate*** verbal and non-verbal robot behaviors.

Portland State University and KAIST: work in progress

Learning Behaviors as Mappings from environment's features to interaction procedures



Professor Perky

1 dollar latex
skin from

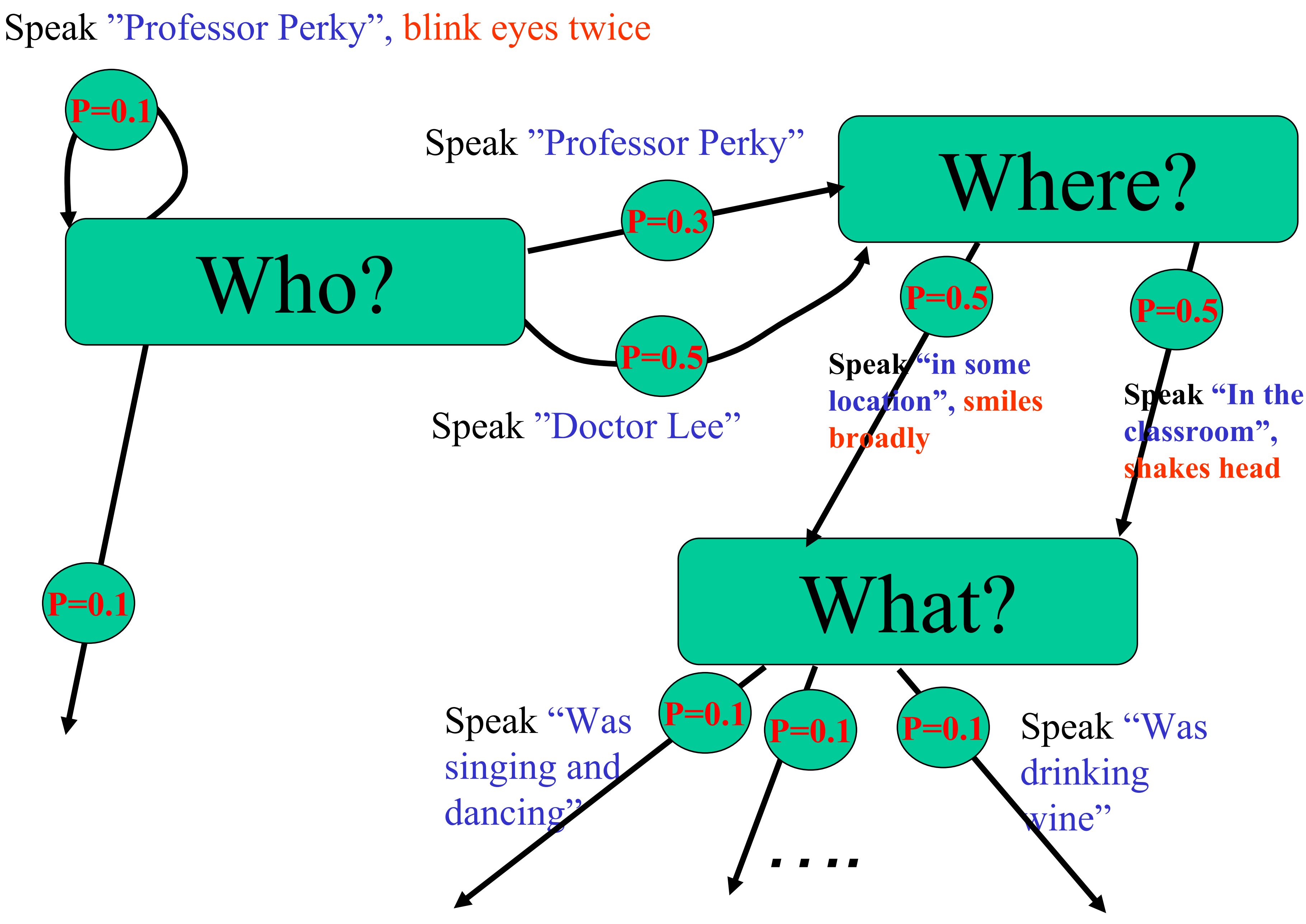
Professor Perky with automated speech recognition (ASR) and text-to-speech (TTS) capabilities

- We compared several commercial speech systems from Microsoft, Sensory and Fonix.

- Based on experiences in highly noisy environments and with a variety of speakers, we selected Fonix for both ASR and TTS for Professor Perky and Maria robots.

- We use microphone array from Andrea Electronics.

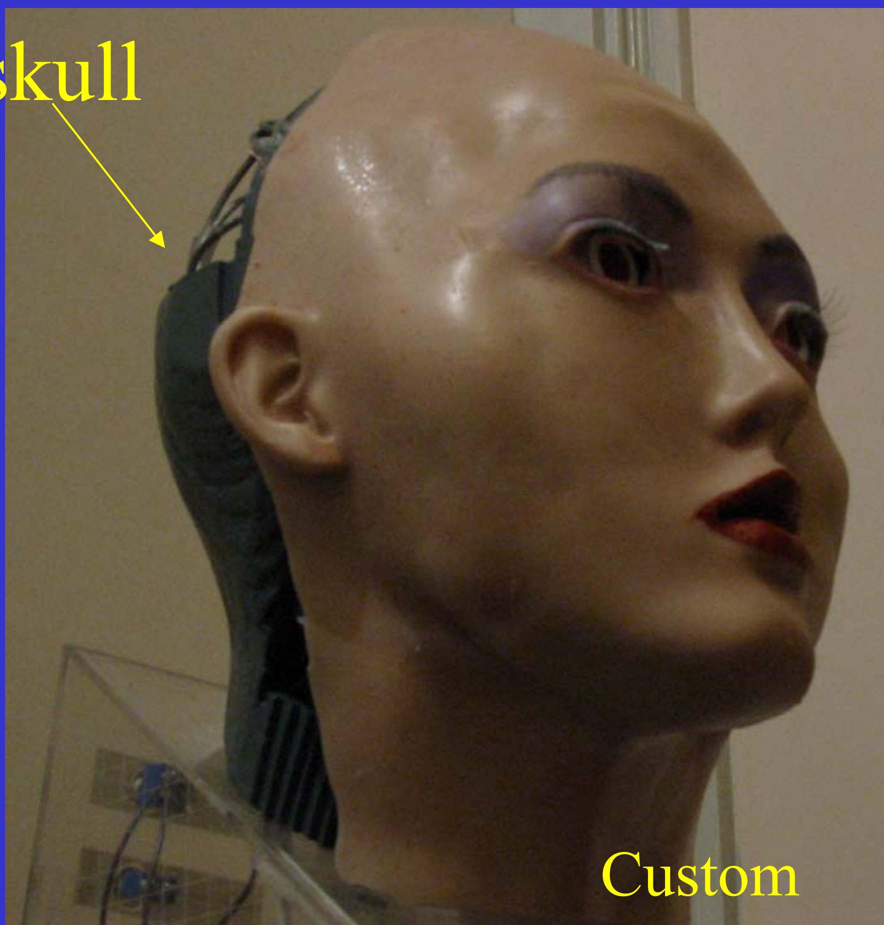
Probabilistic Grammars for performances

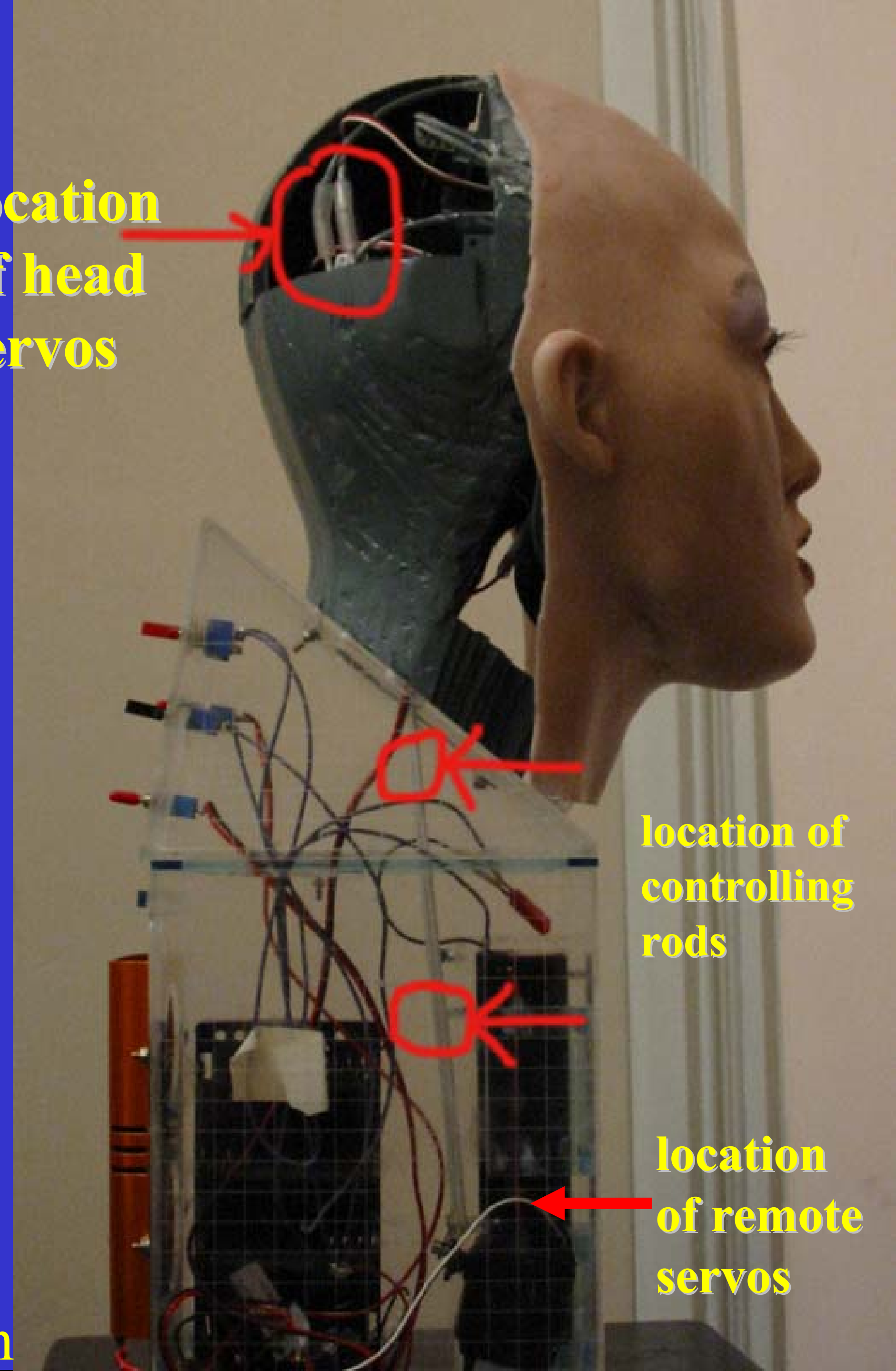


Maria

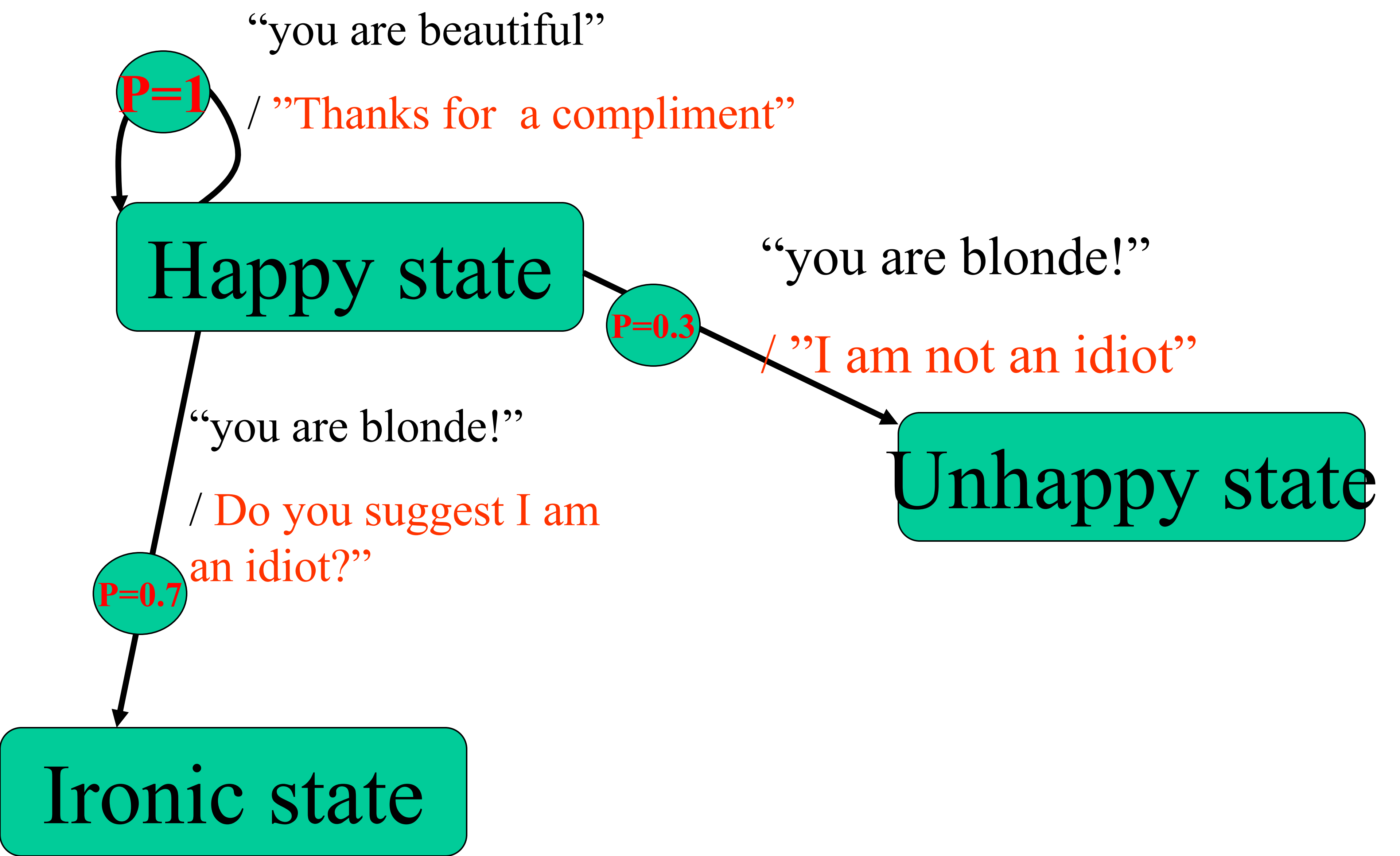
20 DOF

Construction details of Maria





Probabilistic State Machines to describe emotions of MARA



Facial Behaviors of Maria

Maria asks: *Do I look like younger than twenty three?*

Response:

"yes"

"no"

0.3

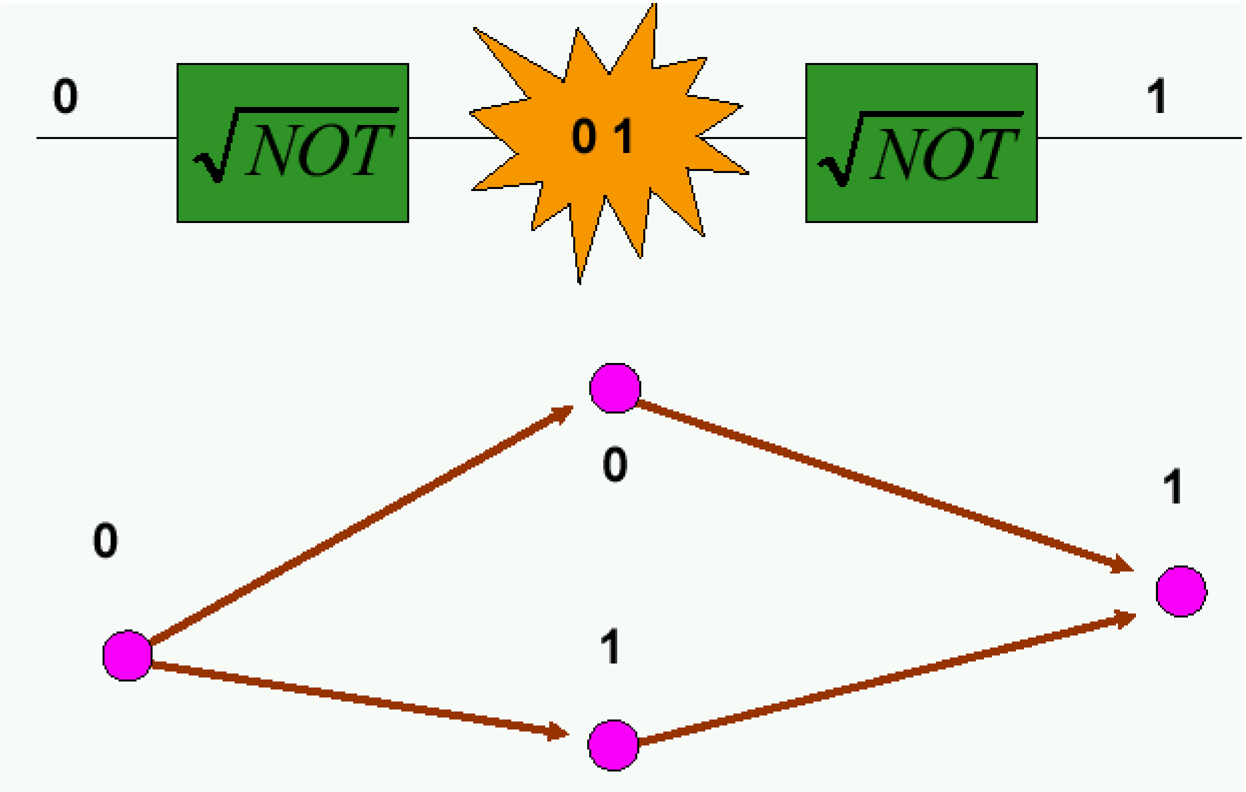
"no"

0.7

Maria smiles

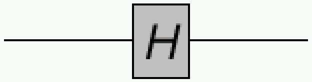
Maria frowns

Quantum Gates and "generalized Probability "



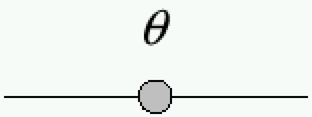
We cannot achieve this in standard or probabilistic logic

Hadamard and Phase gates



$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

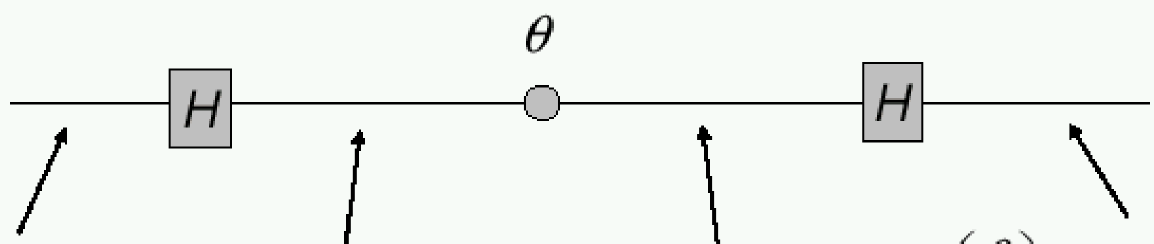
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$



$$|0\rangle \rightarrow |0\rangle$$

$$|1\rangle \rightarrow e^{i\theta}|1\rangle$$

Quantum Circuit



$|0\rangle$

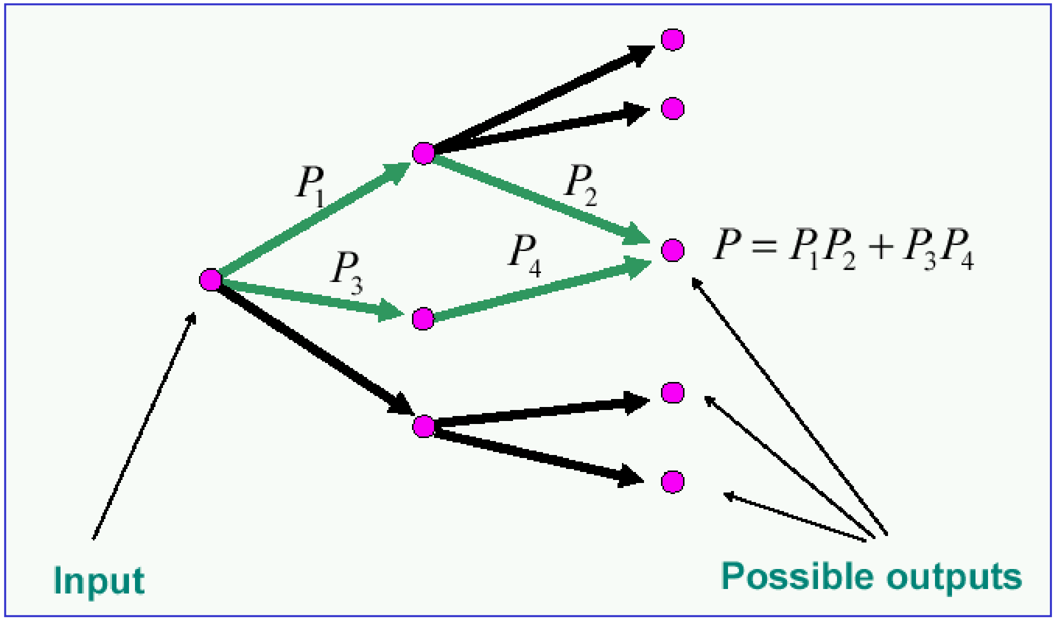
$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$

$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\theta}|1\rangle\right)$

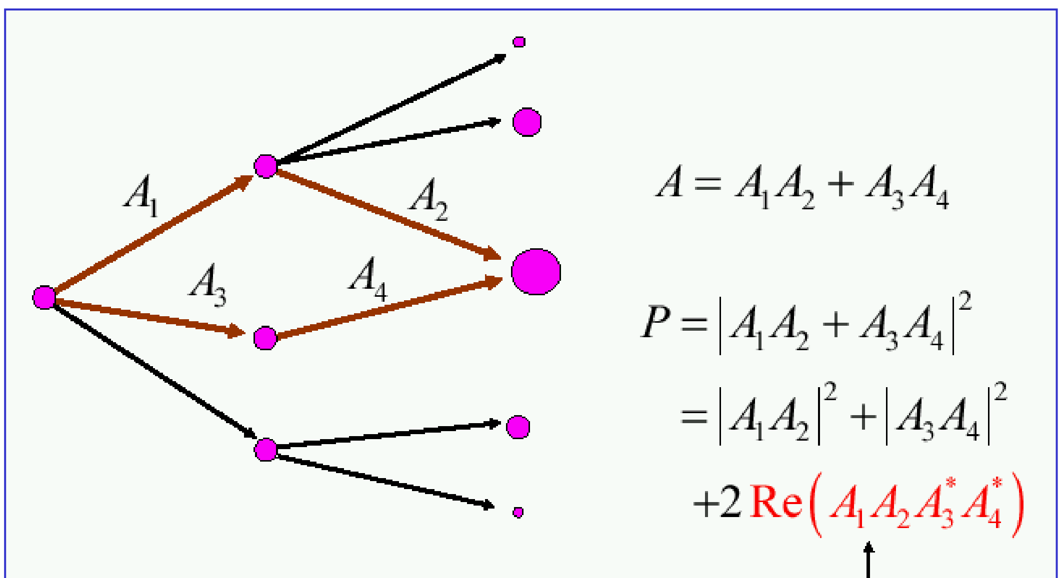
$\cos\left(\frac{\theta}{2}\right)|0\rangle + i\sin\left(\frac{\theta}{2}\right)|1\rangle$

Single particle interference

Probabilistic versus Quantum Computation



Probabilistic



Quantum

Generalizations of Concepts

- Discrete Mapping
- Finite State Machine
- Algorithm
- Turing Machine
- Cellular Automaton
- Neural Net

- Probabilistic Mapping
- Probabilistic State Machine
- Probabilistic Algorithm
- Probabilistic Turing Machine
- Probabilistic Cellular Automaton
- Probabilistic Neural Net

- Quantum Mapping
- Quantum State Machine
- Quantum Algorithm
- Quantum Turing Machine
- Quantum Cellular Automaton
- Quantum Neural Net

Current and Future Research

Representations.

- The choice of representation is crucial in the design of any probabilistic algorithm
- It determines its robustness, efficiency, and accuracy.

Learning.

- The probabilistic paradigm lends itself naturally to learning.
- Very little work has been carried out on automatically learning models (or behaviors) in real-world robotic applications using probabilistic representations.
- Many of today's best learning algorithms are grounded in statistical theory similar to the one underlying the current approach.

High-Level Reasoning and Programming.

Theory of Robotics.

Innovative Applications

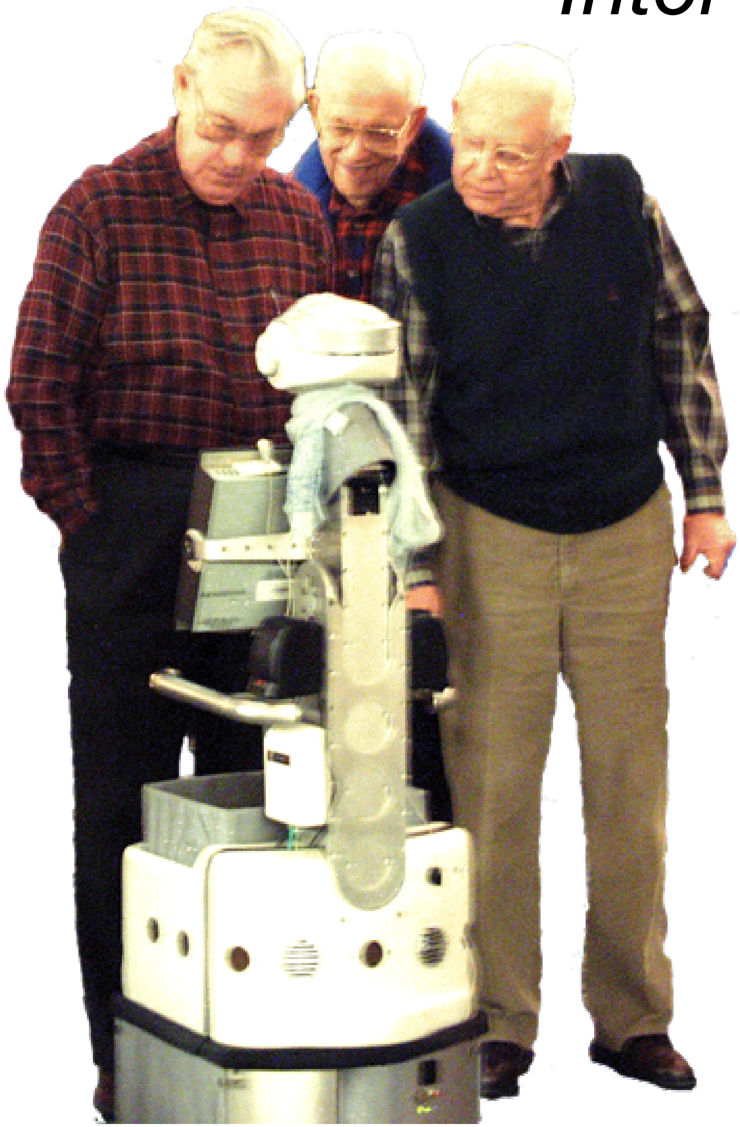
Lecture Summary

- In Robotics, there is no such thing as
  - A perfect sensor
  - A deterministic environment
  - A deterministic robot
  - An accurate model
- Therefore:
  - Uncertainty inherent in robotics
- New Approach, creatively combines model-based and model-less approaches
- Many opportunities possible

- Sources
  - Sebastian Thrun http:// www. cs. cmu.edu/~ thrun/
  - Dieter Fox http:// www. cs. washington.edu/ homes/ fox/
  - Wolfram Burgard http:// www. informatik.uni- freiburg. de/~ burgard/
  - Ranjith Unnikrishnan
  - Marc Zinck
  - David Black-Schaffer
  - Kristof Richmond
  - Ekert, Oxford Univ.

…And Can We Actually Do Something *Useful*?

Intel



Sebastian Thrun