Developing a Probabilistic Post-Perception Module for Mobile Robotics

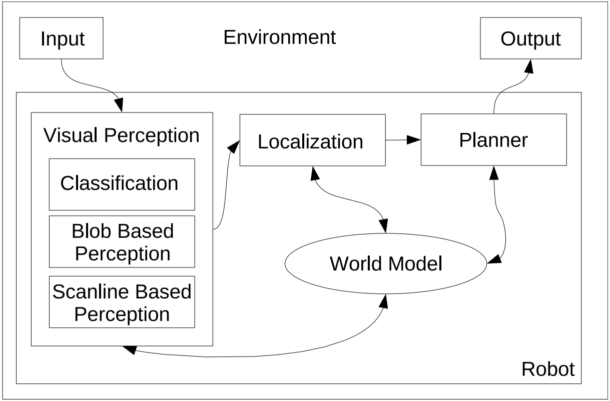
Can Kavaklıoğlu

Systems & Control Engineering
Boğaziçi University

January 23 2009

Outline

Typical Robot Design Review - Primary Perception Visual Input



Visual Perception Algorithms are NOT complete

- Even relatively small images have very large state space
- $(256^3)^{320*240} = 3.07 * 10^{554,858}$, to be specific for a 320x240 pixels image
- Methods exist for pruning the space
- *Classification* to just 10 possible colors reduces state space to $(10^3)^{320*240} = 10^{76,800}$ in the same image
- However all such methods introduce assumptions
- It is improbable to enumerate all cases if not impossible

ARTIFICIAL INTELLIGENCE LABORATORY
AILAB

Visual Perception Algorithms are NOT complete

- Even relatively small images have very large state space
- $(256^3)^{320*240} = 3.07 * 10^{554,858}$, to be specific for a 320x240 pixels image
- Methods exist for pruning the space
- *Classification* to just 10 possible colors reduces state space to $(10^3)^{320*240} = 10^{76,800}$ in the same image
- However all such methods introduce assumptions
- It is improbable to enumerate all cases if not impossible

Visual Perception Algorithms are NOT complete

- Even relatively small images have very large state space
- $(256^3)^{320*240} = 3.07 * 10^{554,858}$, to be specific for a 320x240 pixels image
- Methods exist for pruning the space
- *Classification* to just 10 possible colors reduces state space to $(10^3)^{320*240} = 10^{76,800}$ in the same image
- However all such methods introduce assumptions
- It is improbable to enumerate all cases if not impossible

Visual Perception Algorithms are NOT complete

- Even relatively small images have very large state space
- $(256^3)^{320*240} = 3.07 * 10^{554,858}$, to be specific for a 320x240 pixels image
- Methods exist for pruning the space
- *Classification* to just 10 possible colors reduces state space to $(10^3)^{320*240} = 10^{76,800}$ in the same image
- However all such methods introduce assumptions
- It is improbable to enumerate all cases if not impossible

Sanity Checks

- Hand coded sanity checks remove erroneous combinations
- But how?
- Based on the impossible cases in a known environment
- What about the unknown environments?
- Introduce more assumptions about the unknown environment
- More importantly state space problem makes complete testing almost impossible

Sanity Checks

- Hand coded sanity checks remove erroneous combinations
- But how?
- Based on the impossible cases in a known environment
- What about the unknown environments?
- Introduce more assumptions about the unknown environment
- More importantly state space problem makes complete testing almost impossible

Sanity Checks

- Hand coded sanity checks remove erroneous combinations
- But how?
- Based on the impossible cases in a known environment
- What about the unknown environments?
- Introduce more assumptions about the unknown environment
- More importantly state space problem makes complete testing almost impossible

Sanity Checks

- Hand coded sanity checks remove erroneous combinations
- But how?
- Based on the impossible cases in a known environment
- What about the unknown environments?
- Introduce more assumptions about the unknown environment
- More importantly state space problem makes complete testing almost impossible

Sanity Checks

- Hand coded sanity checks remove erroneous combinations
- But how?
- Based on the impossible cases in a known environment
- What about the unknown environments?
- Introduce more assumptions about the unknown environment
- More importantly state space problem makes complete testing almost impossible

Higher Level Problems Related with Lower Level Perception

- Without reliable perception input *misperceptions* might occur
- Localization particles can diverge
- Cars can go out of the road
- Effects of *misperceptions* can be very severe on the overall performance of a robot
- *See movie localization error*

Higher Level Problems Related with Lower Level Perception

- Without reliable perception input *misperceptions* might occur
- Localization particles can diverge
- Cars can go out of the road
- Effects of *misperceptions* can be very severe on the overall performance of a robot
- *See movie localization error*

Higher Level Problems Related with Lower Level Perception

- Without reliable perception input *misperceptions* might occur
- Localization particles can diverge
- Cars can go out of the road
- Effects of *misperceptions* can be very severe on the overall performance of a robot
- *See movie localization error*

Higher Level Problems Related with Lower Level Perception

- Without reliable perception input *misperceptions* might occur
- Localization particles can diverge
- Cars can go out of the road
- Effects of *misperceptions* can be very severe on the overall performance of a robot
- *See movie localization error*

Problem Statement

- State space is too big for *complete* testing
- Sanity checks are error prone
- But we still need high quality lower level perception

Post-Perception Literature

- Does not exist
- Unlike localization or multi hypothesis tracking
- Due to sanity checks *incomplete* sufficiency?

Post-Perception Literature

- Does not exist
- Unlike localization or multi hypothesis tracking
- Due to sanity checks *incomplete* sufficiency?

Post-Perception Literature

- Does not exist
- Unlike localization or multi hypothesis tracking
- Due to sanity checks *incomplete* sufficiency?

Localization Problems

- How does a robot locate itself in an environment?
  - Triangulation?
  - Probabilistic methods?
- Is the environment known or unknown?
  - The field of RoboCup Standard Platform League
  - Path of an outdoor autonomous vehicle

Localization Problems

- How does a robot locate itself in an environment?
  - Triangulation?
  - Probabilistic methods?
- Is the environment known or unknown?
  - The field of RoboCup Standard Platform League
  - Path of an outdoor autonomous vehicle

Localization Problems

- How does a robot locate itself in an environment?
  - Triangulation?
  - Probabilistic methods?
- Is the environment known or unknown?
  - The field of RoboCup Standard Platform League
  - Path of an outdoor autonomous vehicle

Localization Problems

- How does a robot locate itself in an environment?
  - Triangulation?
  - Probabilistic methods?
- Is the environment known or unknown?
  - The field of RoboCup Standard Platform League
  - Path of an outdoor autonomous vehicle

Multi Hypothesis Tracking

- Tracking has been popular since early computer science
  - Kalman filters used with multi target tracking (1979)
  - Millitary applications with 64K words of memory
- Currently the problem is reduced to Data Association
  - Not necessarily Gaussian noise assumed by Kalman filters
  - Probabilistic Data Association Filter
  - Joint Probabilistic Data Association Filter

Multi Hypothesis Tracking

- Tracking has been popular since early computer science
  - Kalman filters used with multi target tracking (1979)
  - Millitary applications with 64K words of memory
- Currently the problem is reduced to Data Association
  - Not necessarily Gaussian noise assumed by Kalman filters
  - Probabilistic Data Association Filter
  - Joint Probabilistic Data Association Filter

Multi Hypothesis Tracking

- Tracking has been popular since early computer science
  - Kalman filters used with multi target tracking (1979)
  - Millitary applications with 64K words of memory
- Currently the problem is reduced to Data Association
  - Not necessarily Gaussian noise assumed by Kalman filters
  - Probabilistic Data Association Filter
  - Joint Probabilistic Data Association Filter

Multi Hypothesis Tracking

- Tracking has been popular since early computer science
  - Kalman filters used with multi target tracking (1979)
  - Millitary applications with 64K words of memory
- Currently the problem is reduced to Data Association
  - Not necessarily Gaussian noise assumed by Kalman filters
  - Probabilistic Data Association Filter
  - Joint Probabilistic Data Association Filter

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

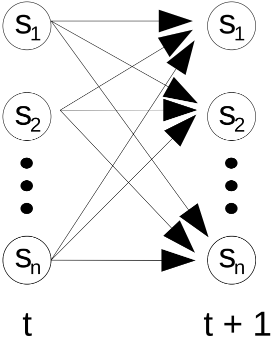


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

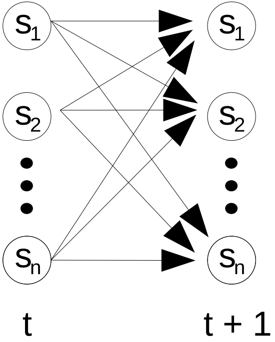


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
    - State definition
    - Transition model
    - Observation model
- Tracking an observation sequence:
    - Initial state
    - Prediction update
    - Correction update

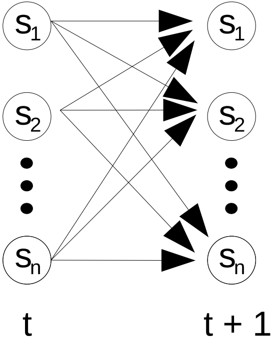


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

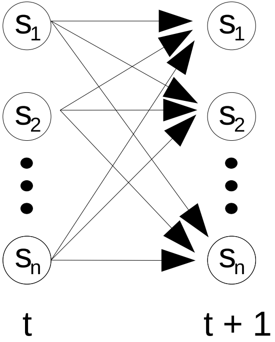


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
    - State definition
    - Transition model
    - Observation model
- Tracking an observation sequence:
    - Initial state
    - Prediction update
    - Correction update

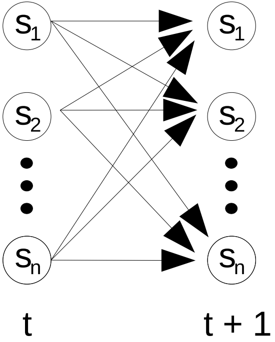


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

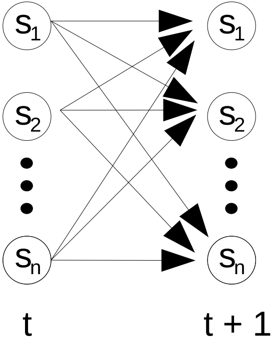


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

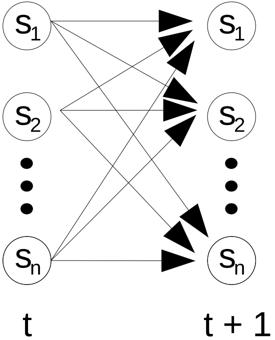


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

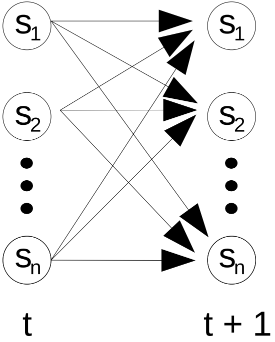


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

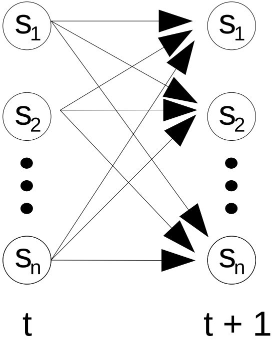


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

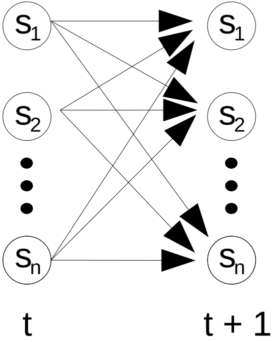


Figure: HMM state representation

Hidden Markov Model

- One of the most general tracking algorithms
- Can track all possible paths of a state space
- Parameters of the system:
  - State definition
  - Transition model
  - Observation model
- Tracking an observation sequence:
  - Initial state
  - Prediction update
  - Correction update

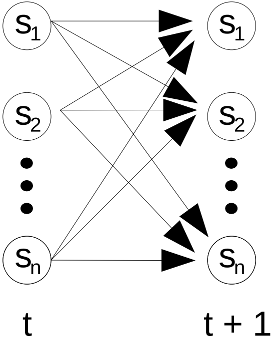
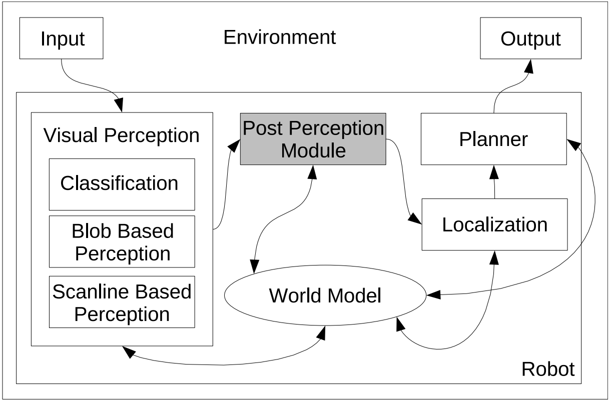


Figure: HMM state representation

Outline

Robot Design with Post-Perception Module in its Place



Employ Multiple Maps

- For every perception received check available map set if the received perception *fits*
- Update the fitting map
- Generate a new map if no map fits
- Pick the *best* map available and mark it as the current map
- Prune inactive maps

Employ Multiple Maps

- For every perception received check available map set if the received perception *fits*
- Update the fitting map
- Generate a new map if no map fits
- Pick the *best* map available and mark it as the current map
- Prune inactive maps

Employ Multiple Maps

- For every perception received check available map set if the received perception *fits*
- Update the fitting map
- Generate a new map if no map fits
- Pick the *best* map available and mark it as the current map
- Prune inactive maps

Employ Multiple Maps

- For every perception received check available map set if the received perception *fits*
- Update the fitting map
- Generate a new map if no map fits
- Pick the *best* map available and mark it as the current map
- Prune inactive maps

Employ Multiple Maps

- For every perception received check available map set if the received perception *fits*
- Update the fitting map
- Generate a new map if no map fits
- Pick the *best* map available and mark it as the current map
- Prune inactive maps

Motives of the Heuristic Solution

- Keep all maps clean of *misperceptions*
- Believe in the map with largest amount of perception information
- Achieve through sanity checking with the *fitness*

Motives of the Heuristic Solution

- Keep all maps clean of *misperceptions*
- Believe in the map with largest amount of perception information
- Achieve through sanity checking with the *fitness*

Motives of the Heuristic Solution

- Keep all maps clean of *misperceptions*
- Believe in the map with largest amount of perception information
- Achieve through sanity checking with the *fitness*

Problems of Heuristic Solutions

- The *perception fitness* is hard to define, either too specific or not expressive enough
- Implementations will be very environment dependent
- In the end fitness values will be result of unreliable lower level perception modules!
- Complexity is $O(perceptionnumber * mapnumber)$ linear but dependent on number of maps

Problems of Heuristic Solutions

- The *perception fitness* is hard to define, either too specific or not expressive enough
- Implementations will be very environment dependent
- In the end fitness values will be result of unreliable lower level perception modules!
- Complexity is $O(perceptionnumber * mapnumber)$ linear but dependent on number of maps

Problems of Heuristic Solutions

- The *perception fitness* is hard to define, either too specific or not expressive enough
- Implementations will be very environment dependent
- In the end fitness values will be result of unreliable lower level perception modules!
- Complexity is $O(perceptionnumber * mapnumber)$ linear but dependent on number of maps

Problems of Heuristic Solutions

- The *perception fitness* is hard to define, either too specific or not expressive enough
- Implementations will be very environment dependent
- In the end fitness values will be result of unreliable lower level perception modules!
- Complexity is $O(perceptionnumber * mapnumber)$ linear but dependent on number of maps

Problems of Heuristic Solutions

- The *perception fitness* is hard to define, either too specific or not expressive enough
- Implementations will be very environment dependent
- In the end fitness values will be result of unreliable lower level perception modules!
- Complexity is $O(perceptionnumber * mapnumber)$ linear but dependent on number of maps

Outline

ARTIFICIAL INTELLIGENCE LABORATORY
AILAB

State Definition: The *meta pose*

- Higher level algorithms (ie. localization, planning) do not need very precise information, even half a meter of error can be tolerated by the localization module
- Thus grid world can be employed
- But grid world is not very robust or useful
- What could be the next step in generalization?
- *Meta pose* is defined as all possible physical arrangements resulting with the received observation

State Definition: The *meta pose*

- Higher level algorithms (ie. localization, planning) do not need very precise information, even half a meter of error can be tolerated by the localization module
- Thus grid world can be employed
- But grid world is not very robust or useful
- What could be the next step in generalization?
- *Meta pose* is defined as all possible physical arrangements resulting with the received observation

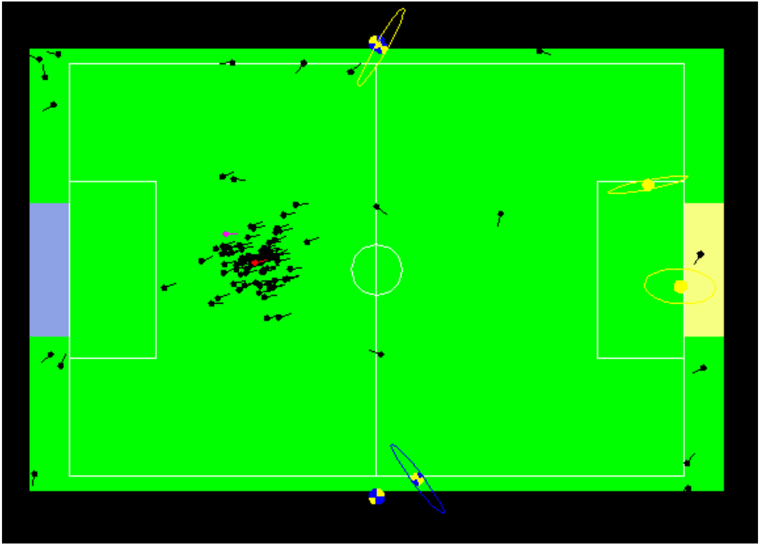
State Definition: The *meta pose*

- Higher level algorithms (ie. localization, planning) do not need very precise information, even half a meter of error can be tolerated by the localization module
- Thus grid world can be employed
- But grid world is not very robust or useful
- What could be the next step in generalization?
- *Meta pose* is defined as all possible physical arrangements resulting with the received observation

State Definition: The *meta pose*

- Higher level algorithms (ie. localization, planning) do not need very precise information, even half a meter of error can be tolerated by the localization module
- Thus grid world can be employed
- But grid world is not very robust or useful
- What could be the next step in generalization?
- *Meta pose* is defined as all possible physical arrangements resulting with the received observation

Meta Pose Illustration From Aibo Field



Transition Model: Transition Model in an Environment

- Consider 4 possible observations in the environment
- 4 states for each perception + 1 for no observation state
- Resultant *Gaussian transition model* is given below

	1	2	3	4	5
1	0.3	0.1	0.01	0.1	0.2
2	0.1	0.3	0.1	0.01	0.2
3	0.01	0.1	0.3	0.1	0.2
4	0.1	0.01	0.1	0.3	0.2
5	0.5	0.5	0.5	0.5	0.2

- Six of the Aibo environment landmarks are used
- Autonomous vehicle experiments have 5 and 10 states

Observation Model

- Characteristics of the sensors defined the observation model
- Better sensor results in more belief in readings
- A similar matrix to the transition model with respective probabilities of sensors firing at observations

Comparison of Solutions

- Problematic definition of *perception fitness* is resolved with the *meta pose* state definition
- Unexpected perception information can be selected using the model
- Model is much more generic and parametrized
- Complexity can be lowered to $O(1)$ with precalculation of possible all matrix calculations

Outline

Aibo Scenarios

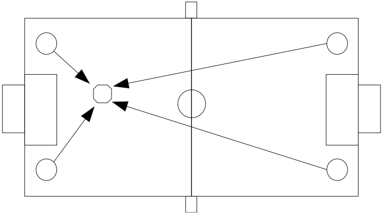


Figure: False perception elimination performance experiments

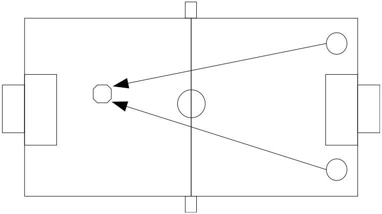


Figure: Localization performance experiments

- Localization experiments have; two set of misplaced landmarks,
- active / passive localization,
- are repeated 5 times from each side,
- error is calculated with the overhead camera, built as a part of this thesis

Aibo Environment Misperception #1

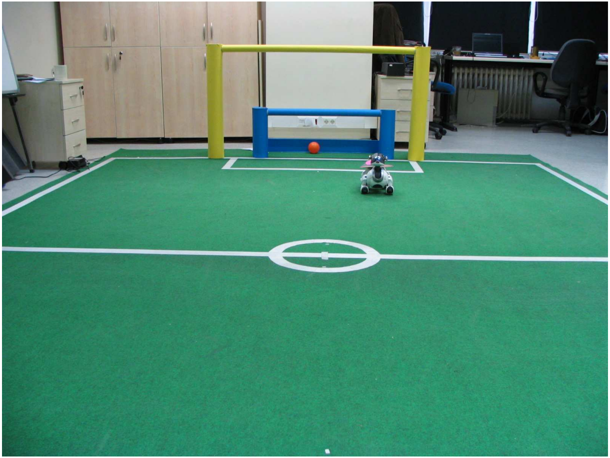


Figure: Misplaced goal

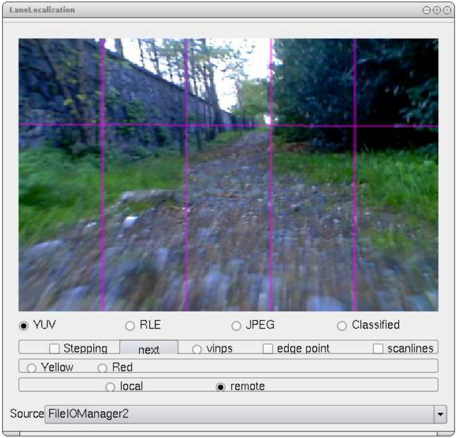
Aibo Environment Misperception #2

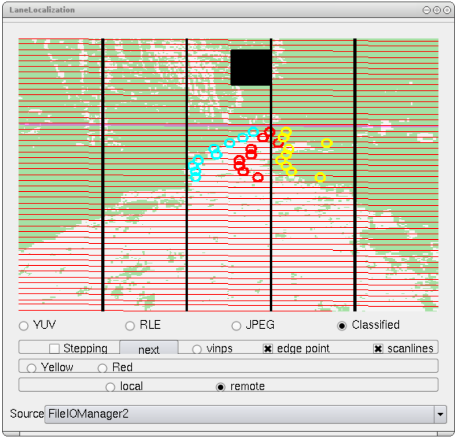
Figure: Misplaced right beacon

Aibo Environment Misperception #3

Figure: Misplaced left beacon

Autonomous Vehicle Scenario / Perception Method #1





Autonomous Vehicle Scenario / Perception Method #2

- Ten stripes instead of five
- Visual features are extracted from multiple sections of each stripe
- Images are hand labeling for depth on each stripe
- Labeled images are trained to match depth values with feature characteristics
- Vehicle moves to the *deepest* area ahead

Outline

Aibo Results - False Perception

Table: Results from five trials of the false perception experiment

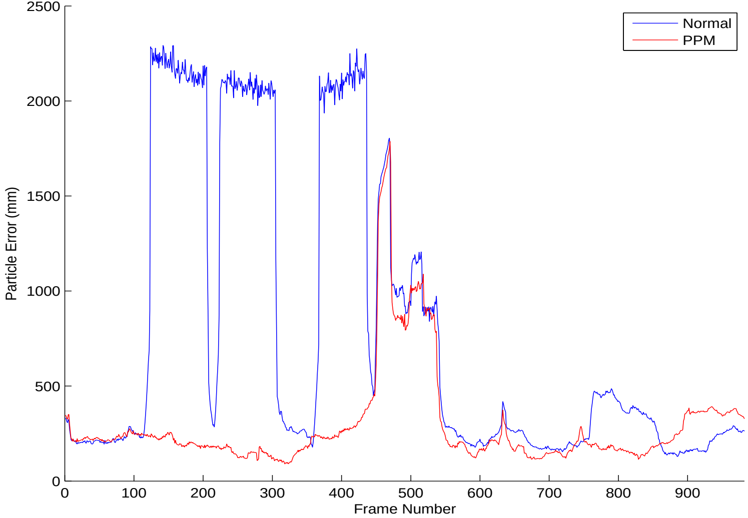
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Avg.
# of Misperceptions	4	2	19	26	23	15
# of Misperception Reports	77	91	102	118	103	98
Frames With Observation	392	463	340	383	351	386
Frames With No Observation	318	346	517	487	492	432
# of Total Frames	710	809	857	870	843	818
Misperception Elimination	50%	100%	89%	81%	78%	80%

Aibo Results - Localization Performance

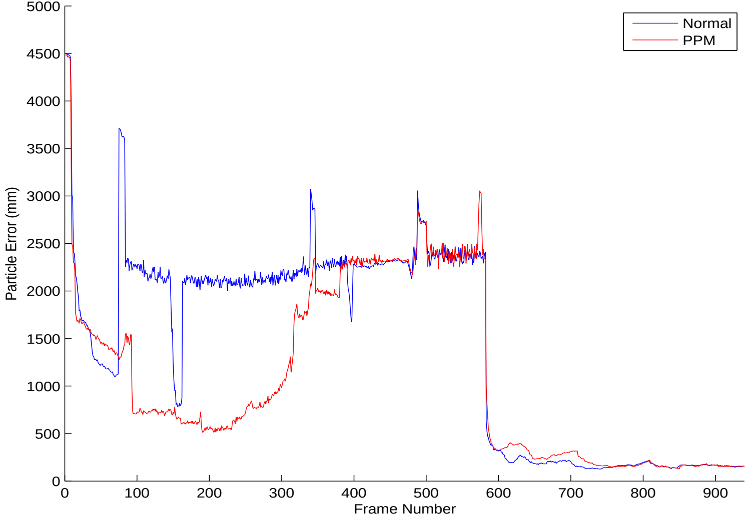
	Normal-1	PPM-1	Normal-2	PPM-2	Normal-4	PPM-4
Mean OE	317	279	675	715	699	542
Std d. of OE	32	9.2	91	35	34	29
Mean OE std d.	302	381	577	790	601	449
Mean AE	854	516	1679	1281	1166	690
Std d. of AE	13	18	27	24	27	33
Mean AE std d.	530	348	807	633	547	332

- Proposed module runs offline, all sanity checks are off
- OE: pose error , AE: particle error, values are in milimeters
- Experiment 1: two misplaced landmarks, active localization
- Experiment 2: four misplaced landmarks, active localization
- Experiment 4: two misplaced landmarks, passive localization

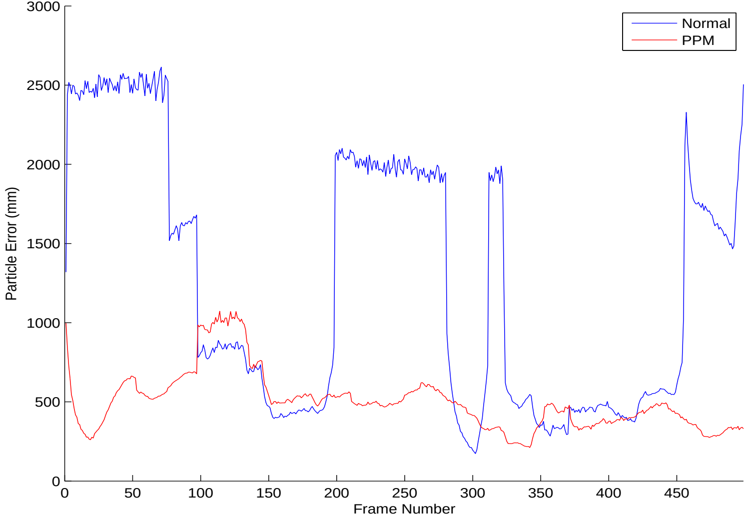
Aibo Experiments Localization Graphs - Experiment 1



Aibo Experiments Localization Graphs - Experiment 2



Aibo Experiments Localization Graphs - Experiment 4



Aibo Experiments Localization Video

- *See the video with the post-perception module*

Outline

Autonomous Vehicle Experiments Results

	Experiment #1				Experiment #2			
	Tr.1	Tr.2	Tr.3	Avg.	Tr.1	Tr.2	Tr.3	Avg.
Total frames	862	556	321	579	71	60	66	66
True positive	52	63	26	47	14	10	4	9
False positive	37	35	13	28	4	1	0	2

- False positive ratio is 18% less on average in the second experiment
- More complicated lower level perception method helps reduce false positive results
- Proposed method can be applied to other domains

ARTIFICIAL INTELLIGENCE LABORATORY
AILAB

Summary

- Due to state space related problems visual detection algorithms can not be *completely* tested
- Probabilistic methods can be applied to predict impossible perception information
- Experiments indicate a generic Hidden Markov Model implementation can be used in multiple domains

- Future Work
  - Improve the probabilistic model with further components for example aiming at producing expected perceptions
  - Sensor fusion applications can benefit from the proposed method in estimating less reliable sensors using reliable ones
  - Learning the Hidden Markov Model parameters to autonomously adapt to an environment to perform a specific task

Questions / Comments

Thanks for listening!

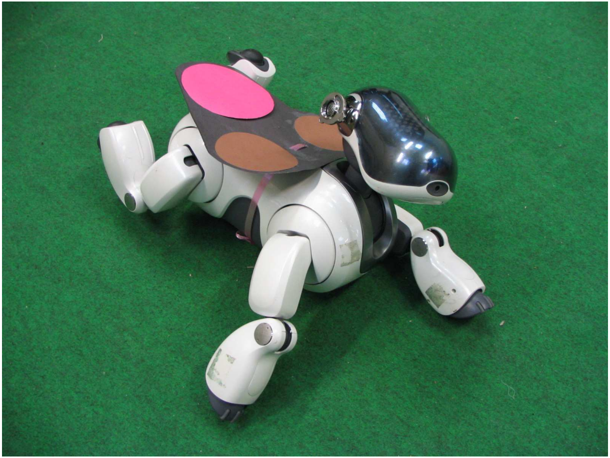


Figure: Aibo tired of carrying the marker