Hierarchical Monte-Carlo Localization Balances Precision and Speed

Vladimir Estivill-Castro

Blair McKenzie

Australia

Outline

- Localization
  - Kalman filters
  - Markov approach
  - Monte Carlo methods
- Our method
- Details that are needed
- Experiment and Results
- Conclusion

Hierarchical Localization

Mi-PAL

griffith university

Hierarchical MC Localization

Localization

- Fundamental problem of mobile autonomous robots
- Use sensor information to determine the robot whereabouts

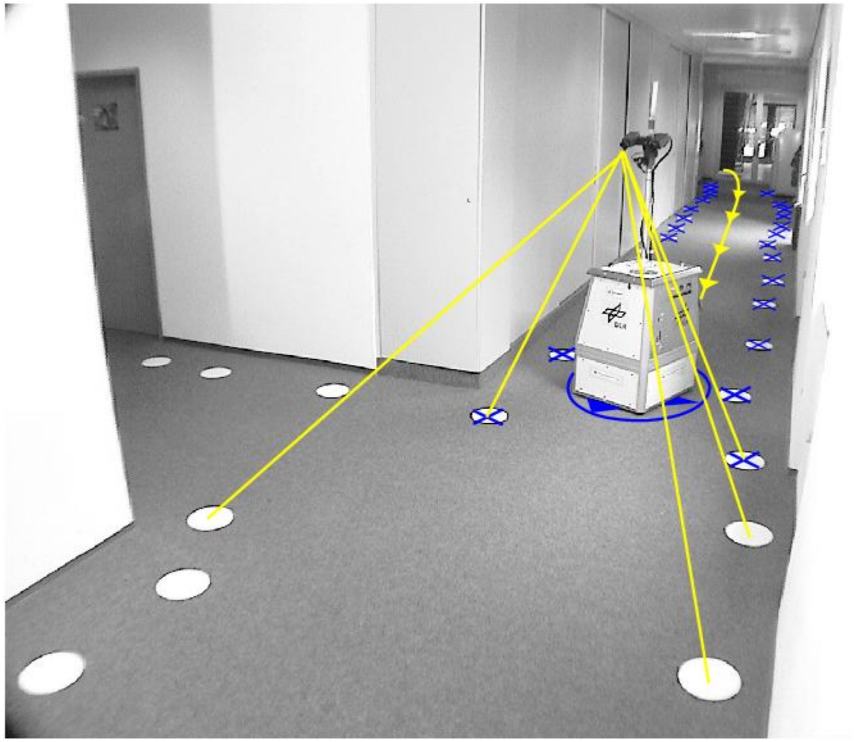


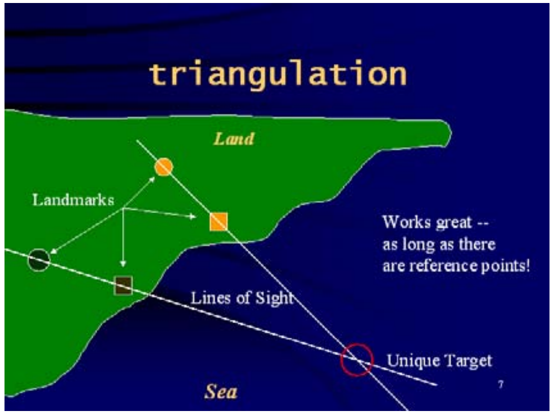
Photo: U. Bremen

Variants to localization

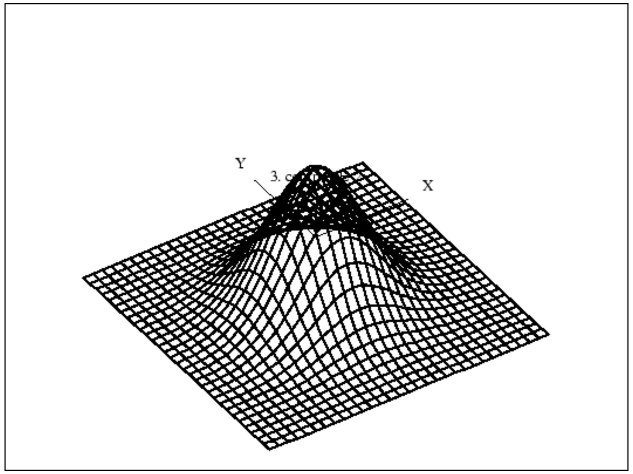
- Self-localization
  - Recognize where I am now on a previously described world
- Position tracking
  - Regularly monitoring position
- Kidnap problem
  - Recovering for being transported (uniformed)

Methods

- The classical triangulation
  - Sensors must be very accurate
  - Enough landmarks
  - No ambiguity

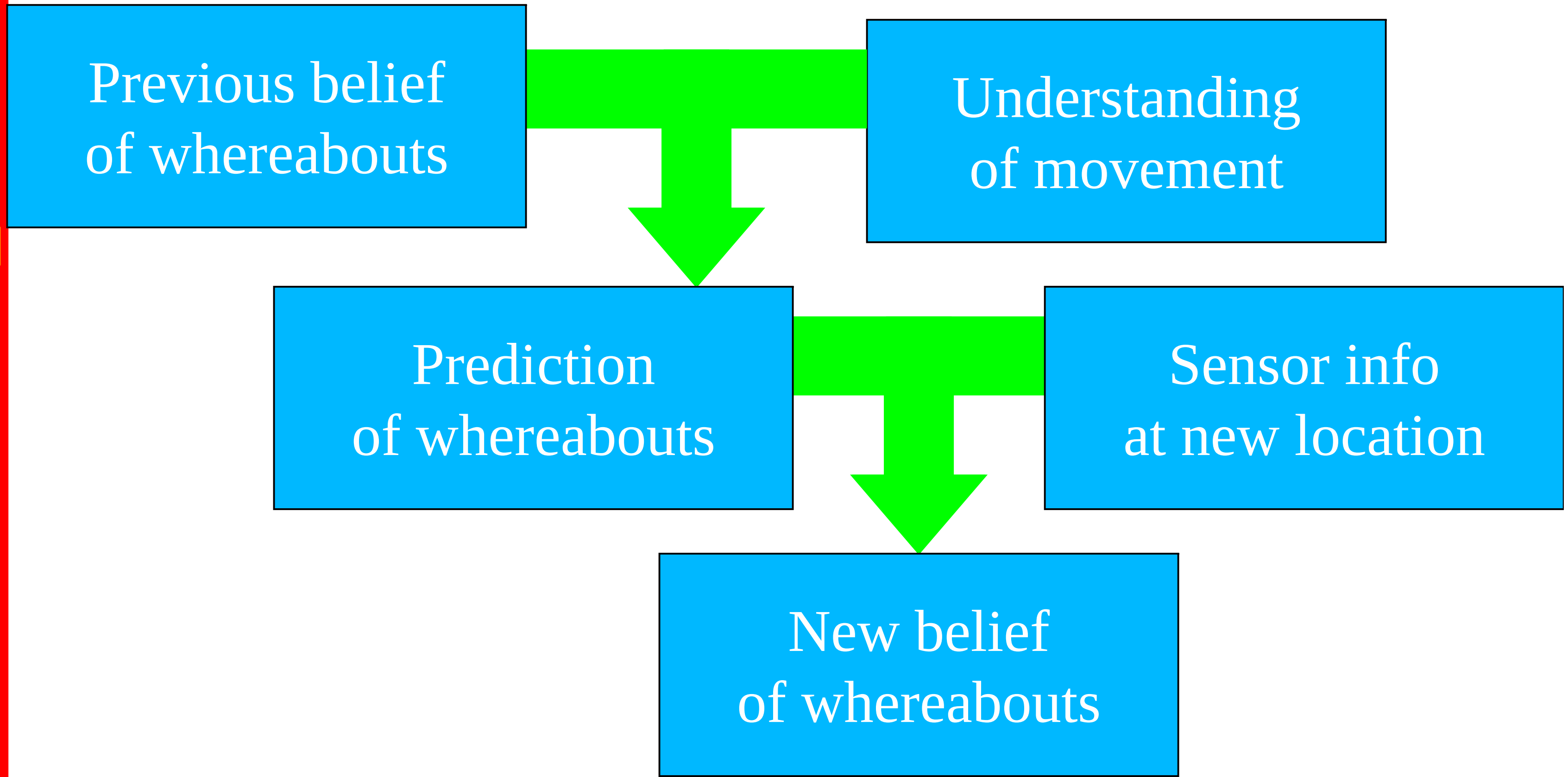


- The modern methods
  - Kalman filters
  - Markov Models
  - Monte-Carlo Localization



Hierarchical Localization

General framework



Mi-PAL

griffith university

Kalman filter

- Very popular for motion tracking
- Models whereabouts as probability distribution
  - A multivariate Gaussian
  - The estimated current position has a probability
- Can be interpreted as an application of Bayes Theory
- Difficulty with kidnap problem or self-localization problem
  - Some variants improve upon this
- Difficulty with ambiguous settings

Markov Models

- Allow for the representation of belief to be a piecewise linear function
- More flexible model of belief
  - And of sensor error and of motion modelling uncertainty
- Use Bayes rule to update belief
- Computational requirements are high

Monte-Carlo Localization

- Represent belief as a very large sample of potential postures (positions)
  - Also known as particles or marbles
- Shown to be superior to Extended Kalman Filter and Markov models[Gutmann and Fox, 2002]
- Shown to be effective for SONY Aibo league (Ambiguity of localizing on lines) [Röfer and Jüngel,2004]

More advantages Monte-Carlo Localization

- Belief does not need to be a parametric probabilistic model
  - Maintain ambiguous hypothesis
- Sensor (Noise) model can also very flexible
  - Several sensors (data fusion)
- Motion model can also be flexible
  - Robot skates, pushed, pick-ed up
- Simple to implement

Monte-Carlo localization working example

One dimensional example, with particles in {0,1,…,9}

0	1	2	3	4	5	6	7	8	9

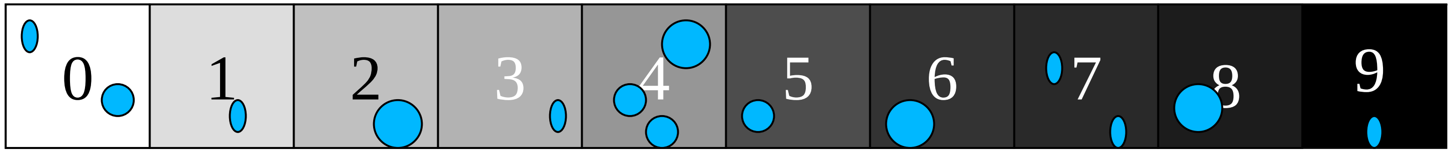
A particle has a weight attached to it

Small weight

Large weight

Initialization

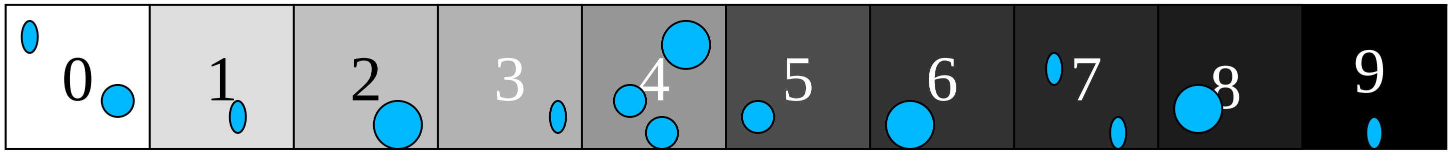
Random particles in {0,1,…,9} with random weights



Apply an action

For as many as the total number of particles

Draw a particle using the distribution and apply motion model to the particle

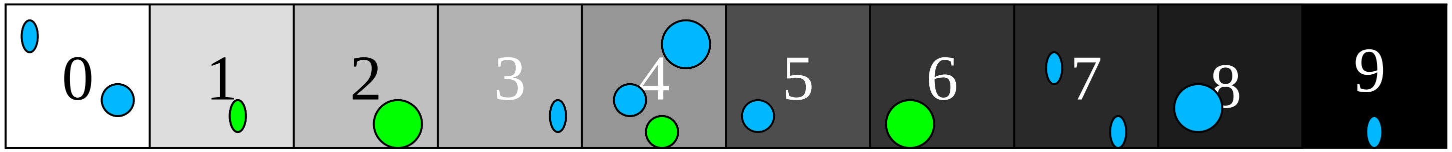


Move one square right

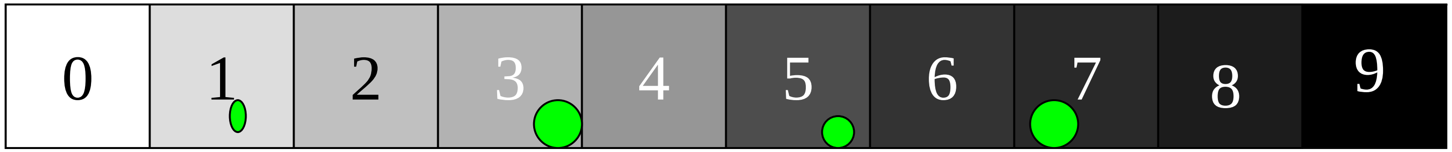
Apply an action

For as many as the total number of particles

Draw a particle using the distribution and apply motion model to the particle



Move one square right

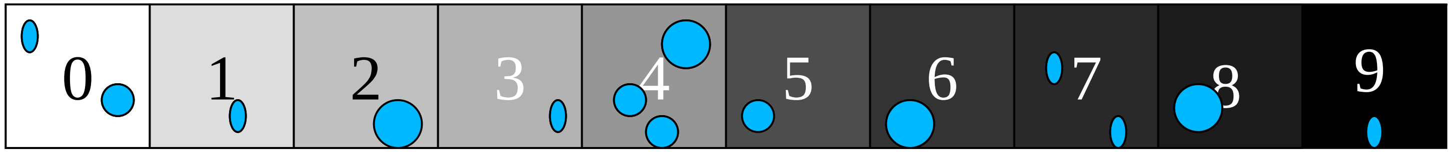


Motion model may include failure possibility (one in 4 moves does not happen)

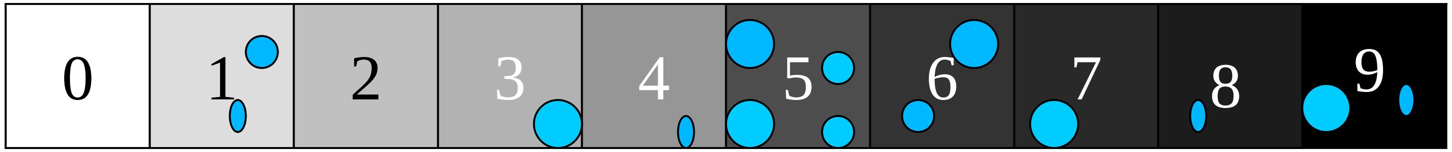
Apply an action

For as many as the total number of particles

Draw a particle using the distribution and apply motion model to the particle



Move one square right

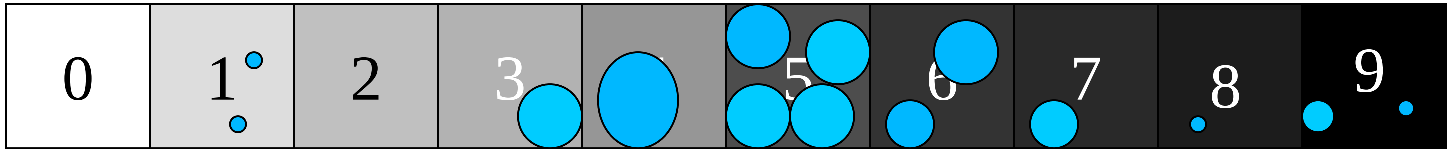


Read sensor information

For each particle,

modify weight as how likely is that such a sensor reading would have been resulted from that posture

Suppose we read 4



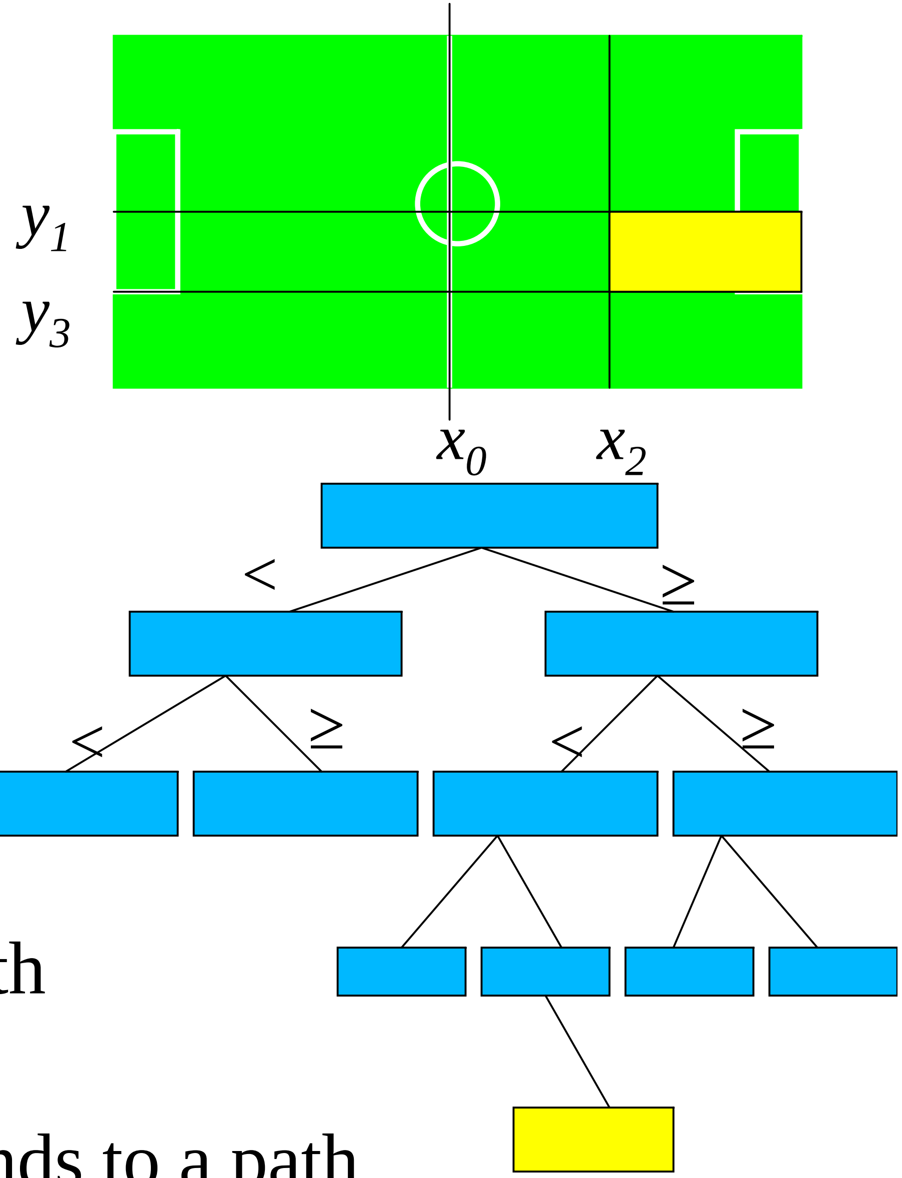
Monte Carlo disadvantages

- Stochastic nature of algorithm implies number of particles can not be small
- Must model the possibility of a kidnap by randomly introducing new particles to the space
- Slow to converge if the sensors are too accurate
- Little theoretical foundation for some of its fixes

Hierarchical MCL

- Organize the *k*-dimensional space for a pose by a *k*d-tree[Bentley,1975].
  - Partition the space at each level by an alternating hyper-planes
- Place a Vanilla MCL at each node to determine the section of the space for the whereabouts of the robot

kd-tree

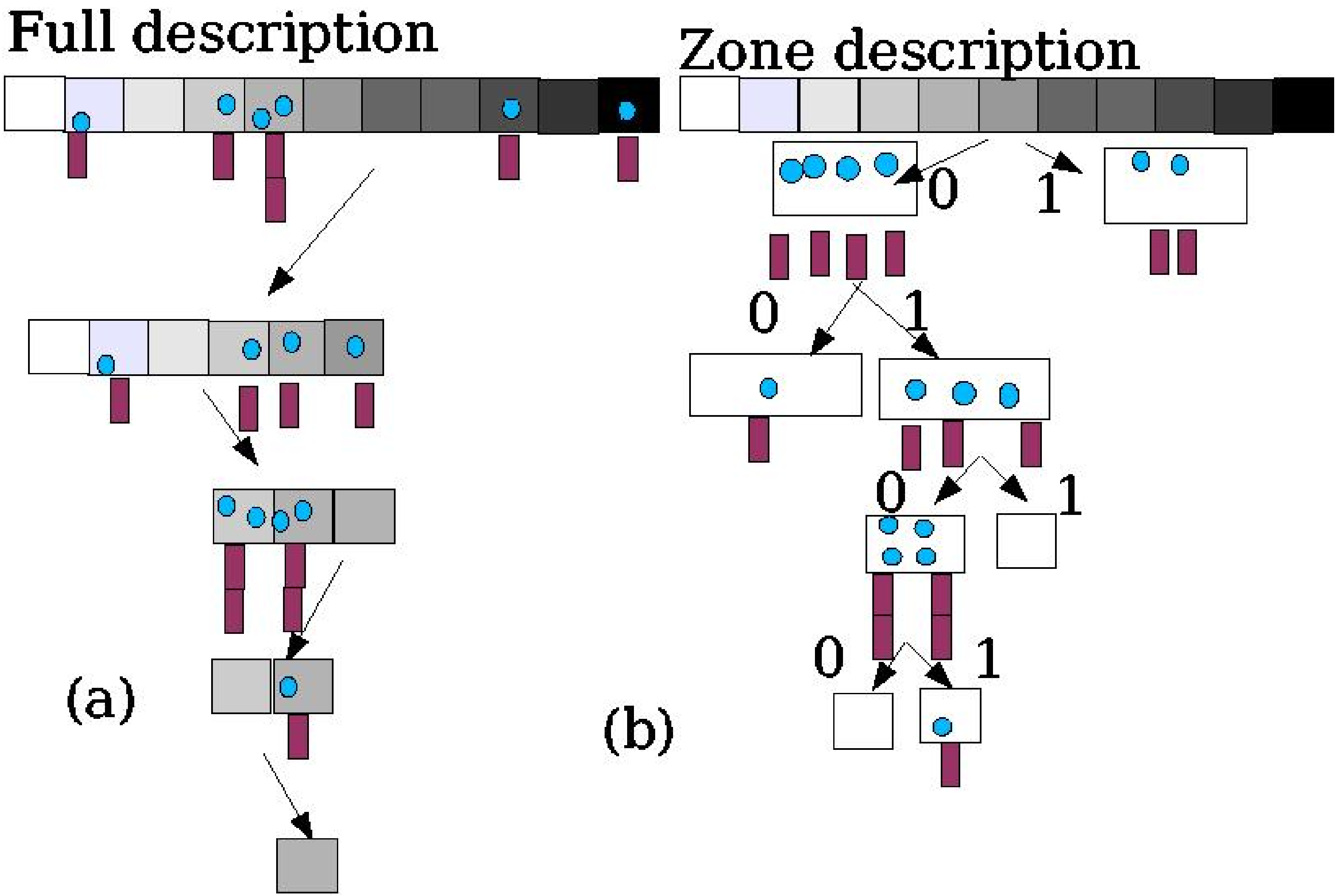


- Illustration in 2D
  - First division with respect to *x*
  - Second partition with respect to *y*
  - Third partition with respect to *x*
  - A region corresponds to a path in the tree

Two variants of Hierarchical MCL

- Full description
  - At each node a Vanilla MCL with particles that represent a complete pose descriptor
    - A vector $\vec{x}$
- Zone descriptor
  - At each node a Vanilla MCL with particles that are in the discrete universe {0,1}
    - "Go left –0; Go right 1."

Hierarchical MCL
1 dimensional illustration



Intuition

- Less particles are necessary to determine which half of a region the robot is in
- Many times we just need more global information for decision making than very specific information
  - Which half of the field I am in can be answered by the root node

Two schemes for allocation of particles

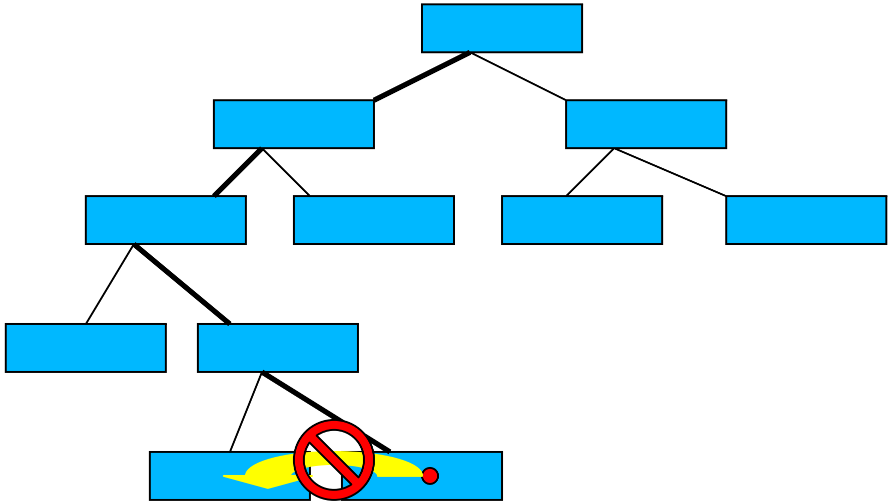
- Let m be the number of particles you would use in Vanilla MCL
- Schema 1, place $m_0=m/(depth+1)$ at each node and the complexity of Hierarchical MCL is the equivalent
- Schema 2 place $m_0=m(1\text{-}1/2^{depth})$ and then $m_i=2m_{i+1}$ and the space requirements of Hierarchical MCL are equivalent to Vanilla MCL
- (with equivalent time complexity)

Hierarchical MCL Localization

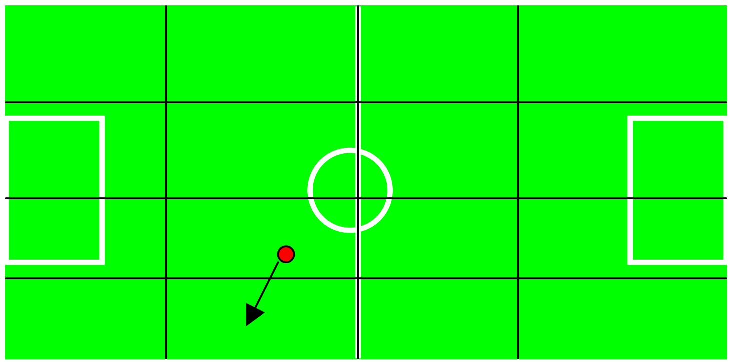
Important implementation aspects

- No migration of particles to siblings
- Particles in a node represent positions outside the region covered by the node
- Incorporation of high precision sensors and low precision sensors
  - Conversion of high precision sensors to virtual low resolution sensors
- Some percentage of particles are always random
- Conditional approach to the importance step

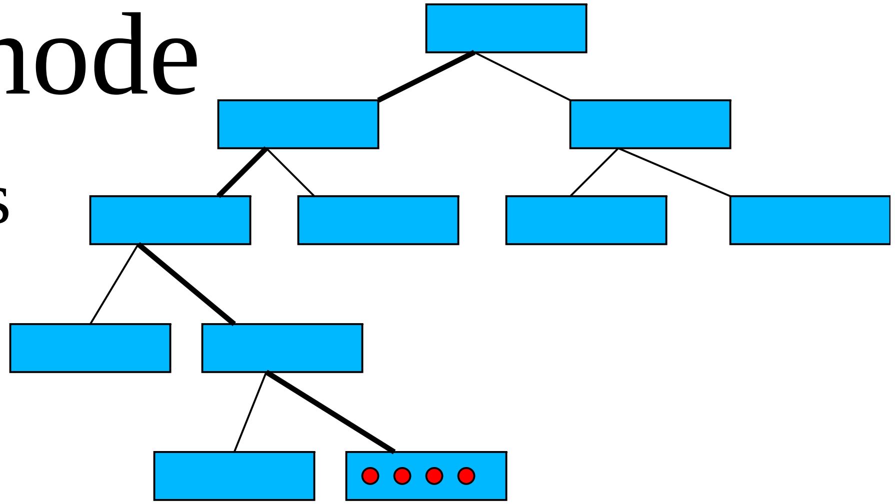
No migration of particles to siblings



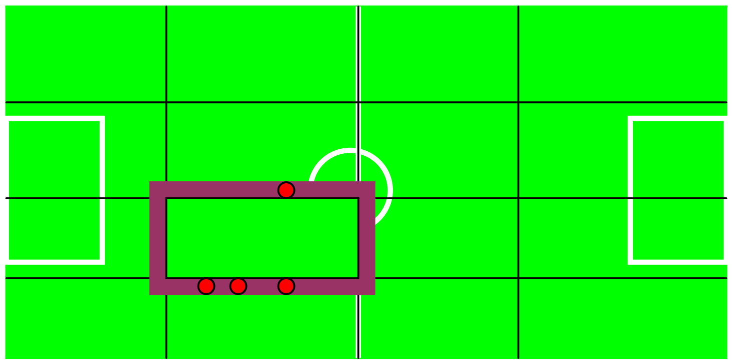
- A particle at a node receives the Motion Model modification
- Particle does not shift to sibling



Particles in a node represent positions outside the region covered by the node

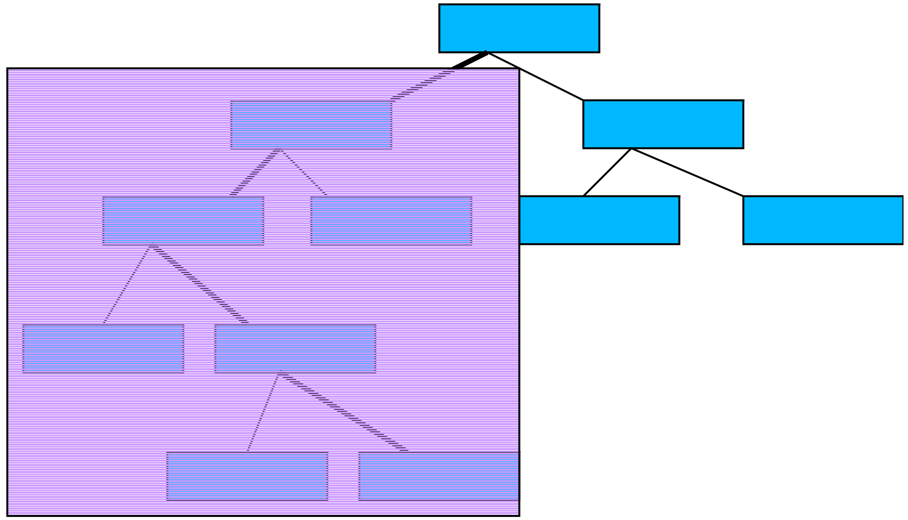


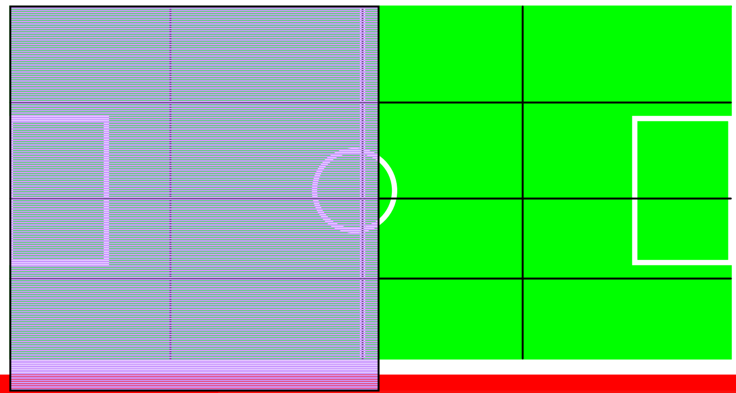
- Locally, this represents the robot is not at this node
- With in ε so that the Motion Model can place the particle if the robot enter the region of the node again
- Upper levels direct the belief



Incorporation of low precision sensors

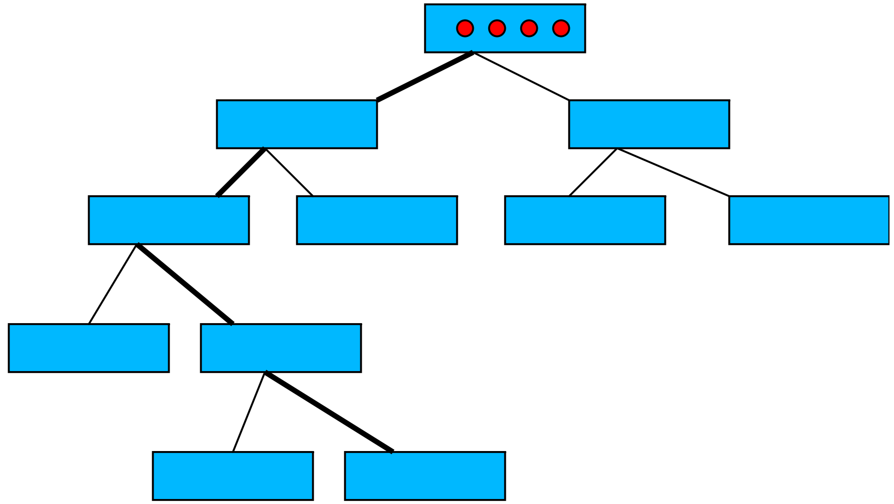
- A sensor of low accuracy does not modify nodes deep in the tree
  - Can be skipped

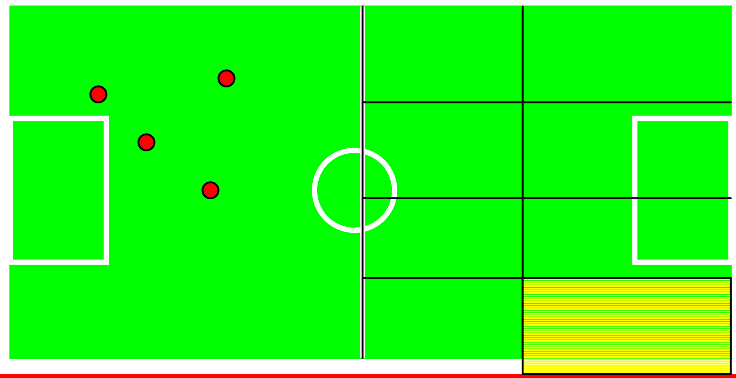




Incorporation of high precision sensors

- A sensor with high accuracy is made a virtual low resolution sensor on shallow nodes
  - Otherwise particles in one half are essentially modified by stochastic noise
    - The only hope is the particles included for the kidnap problem





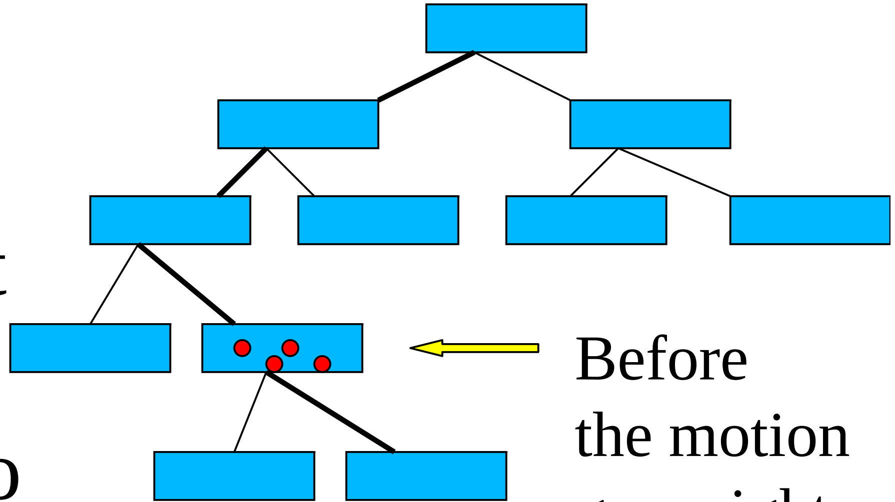
Some percentage of particles are always random

- At every node,
  - we take 90% of the particles drawn from the current representation of the probability distribution
  - We take 10% of the particles as random poses
- Protection for stochastic noise and kidnap problem

Hierarchical MC Localization

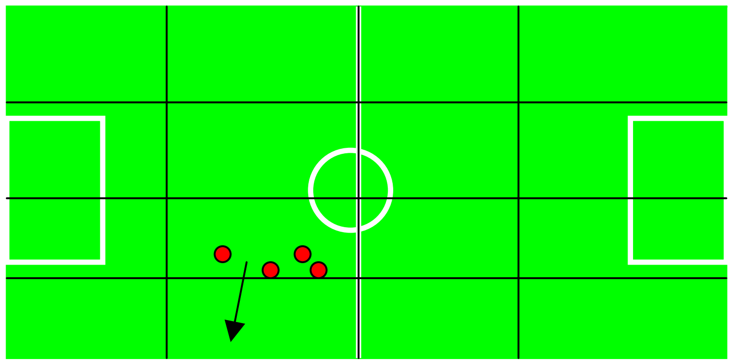
Conditional approach to the importance step

- Once the iteration loop is performed at the loop, the children is chosen to go down the tree and perform the iteration loop



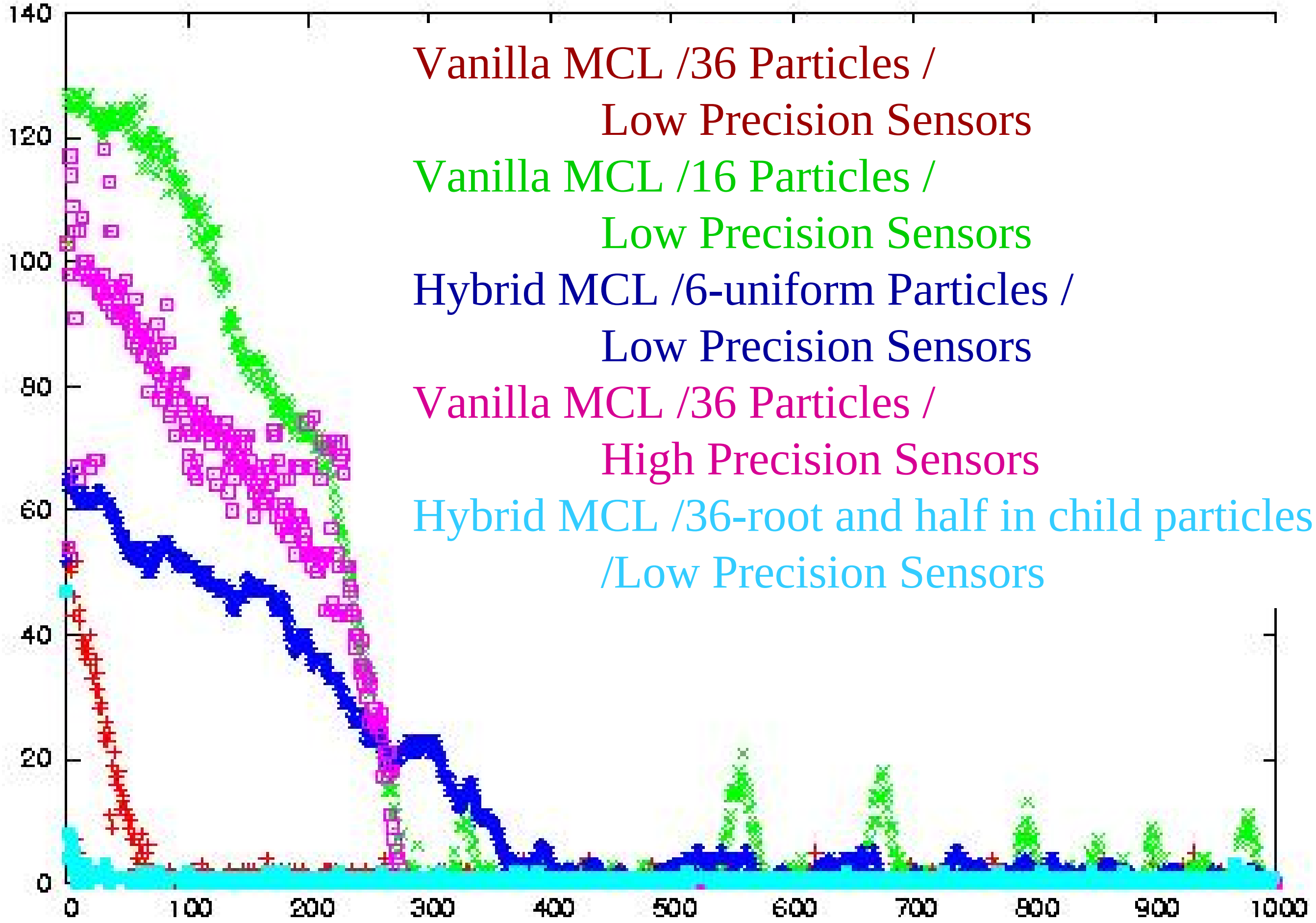
Before the motion goes right

After the motion goes left



Experiments

Self-localization problem

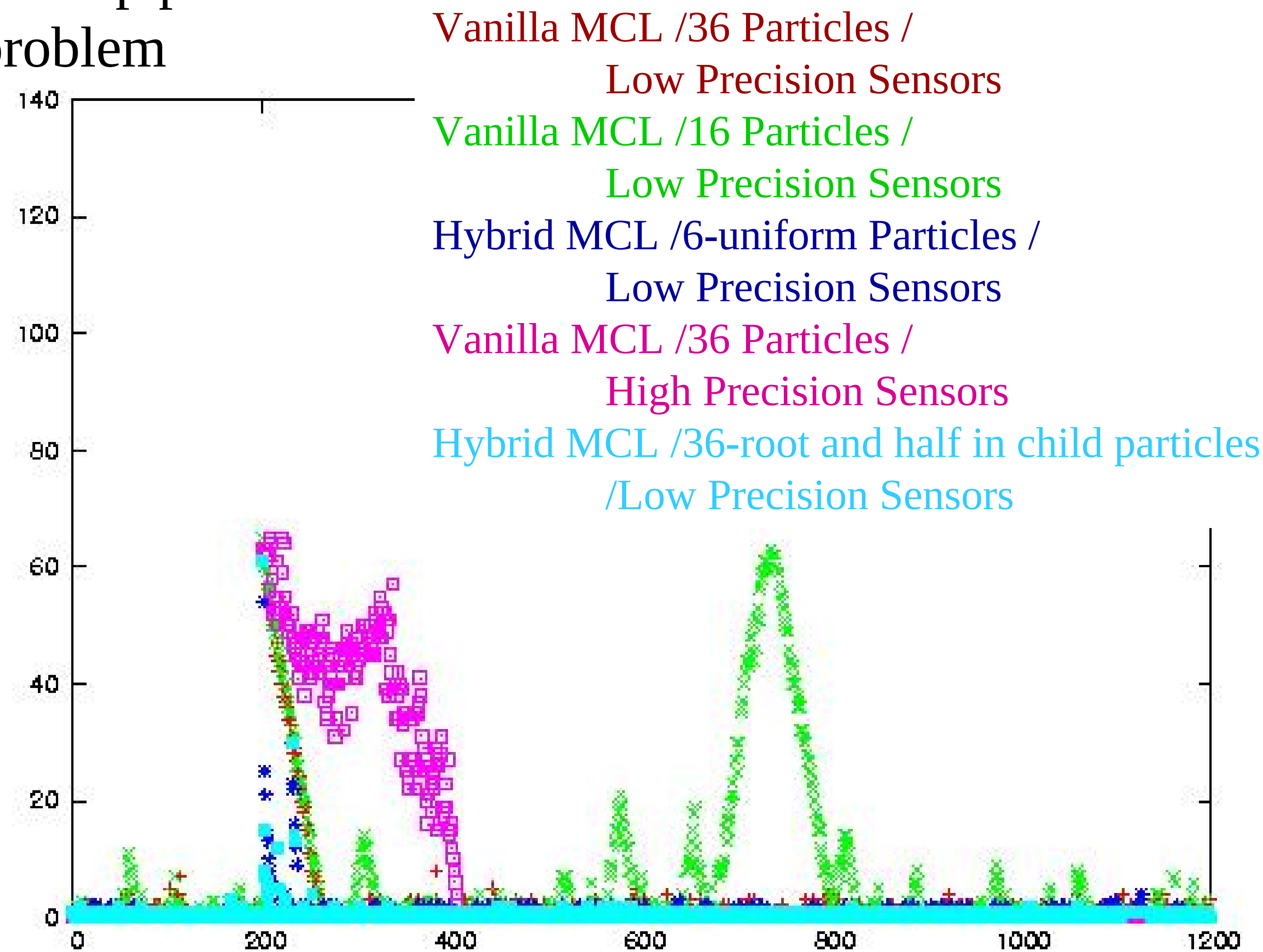


Hierarchical MCL Localization

Experiments

Kidnap problem

problem



Discussion

- The approach called *kd*-trees [Guttmann and Fox, 2002] and [Thrun et al, 2001] is truly a kernel density tree approach to represent piece-wise linear distributions
  - Not a mechanism to structure particles efficiently
- The improvements to Markov Localization (pre-computation of sensor model and selective update)[Fox et al 1991] demand high memory requirements
  - Closest work is Octrees approach [Burgard et al, 1998] but dynamically upgrading the tree is not trivial
  - Work is still proportional to nodes in the tree
    - Our work is proportional to path to the leaf in the tree
- [Fox,2003] discusses managing particles efficiently
  - Our two schemes for particle allocation handle this

Conclusion

- Hierarchical MCL allows incorporation of sensors of different precision at the right level of information content.
- Method is computationally competitive with Vanilla MCL and faster to answer global / regional queries

Hierarchical MC Localization

THANK YOU