Localization

Carnegie Mellon University

October 20, 2003

Scott Lenser
Manuela Veloso

Localization

- Localization determines and tracks the location of the robot in the environment.
- Takes estimates of robot motion as input. The update related to motion estimates is sometimes called the process update.
- Takes sensations associated with locations in environment as input.

Localization - Motion Estimates

Examples of different sources of motion estimates:

- Motion commands attempting to be executed. This is used on the AIBOs.
- Wheel encoder readings (measures rotation of wheels). Used on most wheeled robots.
- Inertial sensors. These include accelerometers and rate gyros. These are used on most aerial robots.
- Optical flow (vision based movement estimates). Not commonly used because it is more expensive and complicated to calculate.

Localization - Sensor Readings

- Can use any sensor reading that varies with the robots position.
- Sensor readings with simpler relations to robot location are usually easier to use.
- AIBOs mainly use the bi-colored markers. Can also use the goals, walls of field, and stripes.

Localization - Sensor Readings

Example of different sensor readings used in localization:

- Vision of landmarks.
- Sonar distance readings.
- Laser scanner distance readings.
- GPS signals.
- Electronic compass readings.
- WLAN signal strength.

Object Tracking

- Object tracking determines and tracks the location of objects in the environment.
- Takes estimates of object motion as input. This is called the process update.
- Takes sensations associated with object locations as input.

Relation of Tracking Problems

- Localization tracks tracks position of robot relative to fixed landmarks.
- Object tracking tracks position of objects relative to robot.
- The problems are related by a simple change in reference frame.
- Each problem requires the same process update and snesor update steps.
- Same techniques are applicable for each problem.

Probability Review - Notation

- $P(A = a)$ — probability/likelihood of random variable A taking on value a.
- A random variable is simply a variable that can take on a variety of values.
- For discrete variables, $P(A = a)$ is the probability that A takes on the value a.
- For continuous variables, the probability of a random variable taking on a particular value is 0.
- For continuous variables, $P(A = a)$ is proportional to the probability of A taking a value in a small neighborhood around a.

Probability Review - Notation [cont.]

- $P(A)$ is called a probability distibution in the discrete case and a probability density in the continuous case (our case).
- $P(A)$ in continuous case is simply a function from values of A to likelihoods in small neighborhoods.
- A likelihood is simply proportional to a probability.
- The probability over a range of values is simply the integral of the probability density.
- $P(\vec{A})$ must sastisfy

$$\int_{\vec{a}=-\vec{\inf}}^{\vec{\inf}} P(\vec{A} = \vec{a}) d\vec{a} = 1$$

- This simply means that the total value of getting some value of A is 1.

Probability Review - Notation [cont.]

- $P(A|B) \equiv \frac{P(A,B)}{P(B)}$
- $P(A|B)$ is called a conditional probability density.
- At a particular value of $B = b$, $P(A|B = b)$ is a probability density.
- $P(A|B = b)$ must sastify

$$\int_{a=-\inf}^{\inf} P(A = a|B = b)da = 1$$

Probability Review

- $P(A = a|B = b) = \frac{P(A=a,B=b)}{P(B=b)}$ by definition.
- Note that $P(A|B)P(B)$ is a joint probability density over A and B.
- We can get a probability density over A by taking the integral over possible values of B. $\int_b P(A|B)P(B)db$ is a probability density over A.
- In general, the formulas that apply with random variables at specific values also apply for random variables with densities.
- Rearranging, we get

  $$P(A, B) = P(A|B)P(B)$$

  This rule can also be chained as

  $$P(A, B, C) = P(A|B, C)P(B|C)P(C)$$

- Bayes' rule

$$P(A|B) = \frac{P(A, B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

Probability Review

- If $P(A|B) = P(A)$, A and B are called independent.
- If A and B are independent, $P(A, B) = P(A)P(B)$.
- If $P(A|B, C) = P(A|C)$, A and B are called conditionally independent given C, or A is independent of B given C.
- If $P(A|B, C) = P(A|C)$,

$$P(B|A, C) = \frac{P(A|B,C)P(B|C)}{P(A|C)} =$$

$$= \frac{P(A|C)P(B|C)}{P(A|C)} = P(B|C)$$

- If A and B are conditionally independent given C, $P(A, B|C) = P(A|C)P(B|C)$

Probability Review

- If A and B are mutually exclusive, $P(A \vee B) = P(A) + P(B)$.
- Corrolary,

$$P(A = x, x \in [a_0, a_1]) = \int_{x=a_0}^{a_1} P(A = x)dx$$

- Corrolary,

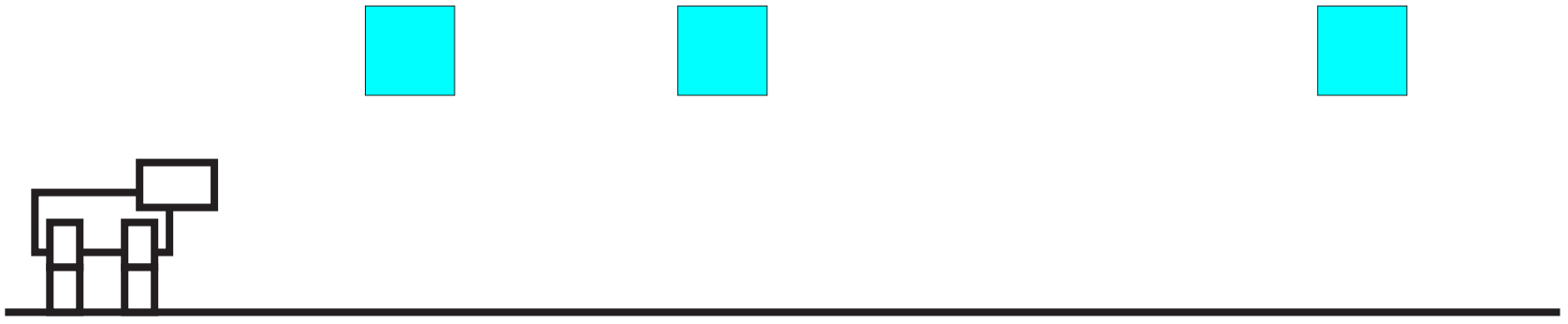
$$P(B = b, A = x, x \in [a_0, a_1]) = \int_{x=a_0}^{a_1} P(B = b, A = x)dx$$

Probabilistic Localization

- Represent robot location as a probability density over possible robot locations or poses (a pose is simply a location including orientation information).
- This probability density is known as the belief state because it represents the robot's belief in its current position. This probability density summarizes all the information the robot has about its current position.
- This probability density is updated for movements of the robot and for information from sensors.

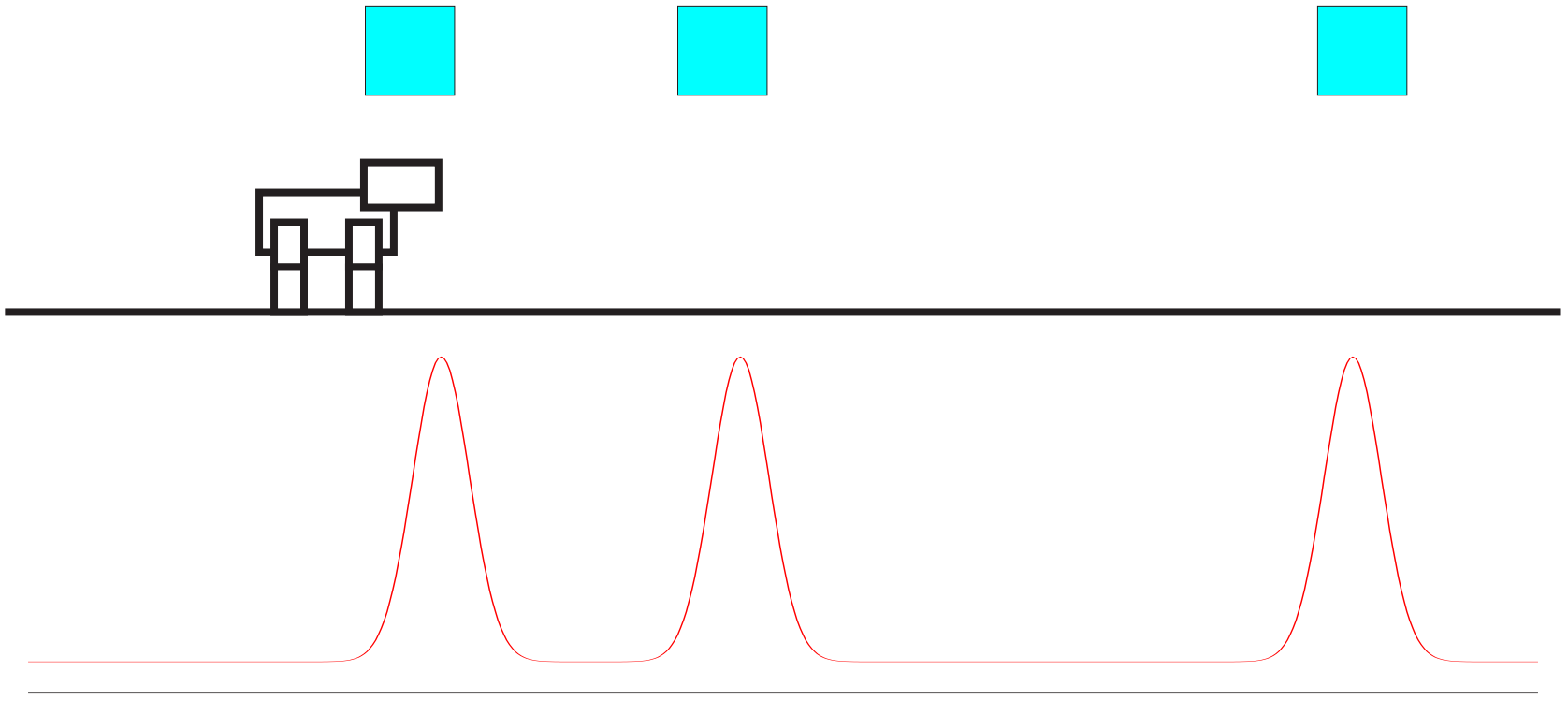
Probabilistic Localization in Pictures

Robot starts off lost.



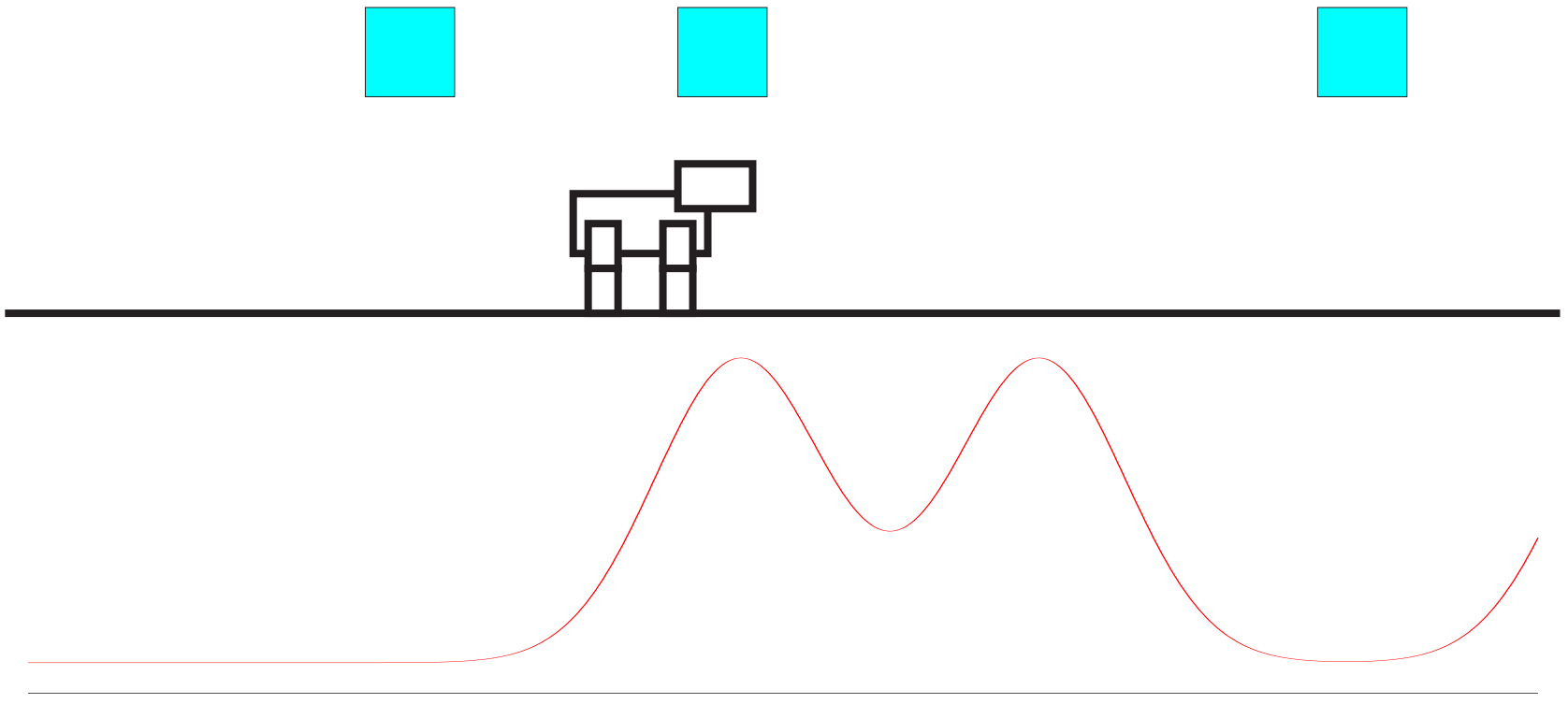
Probabilistic Localization in Pictures

Robot sees a marker to its side.



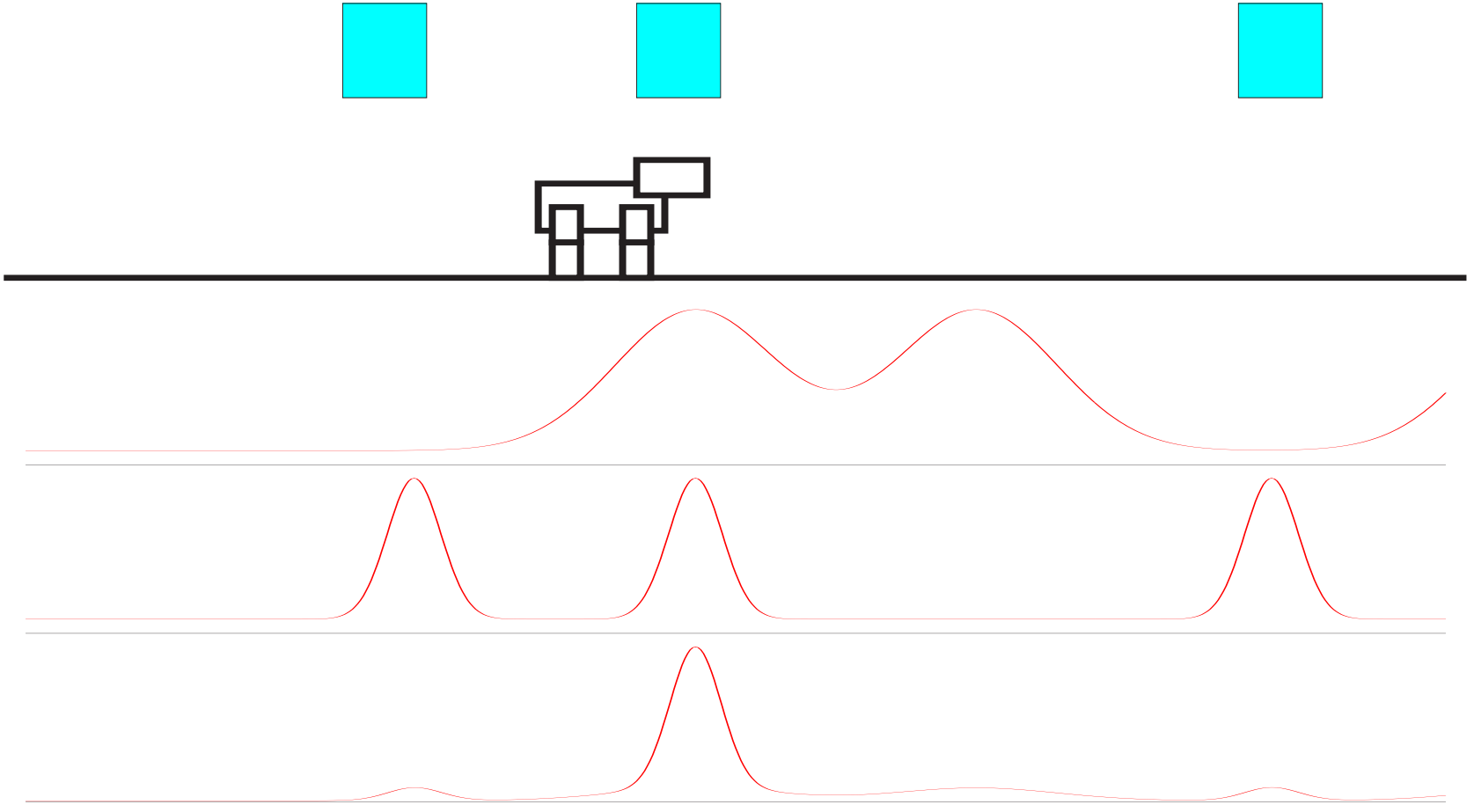
Probabilistic Localization in Pictures

Robot continues moving to right.



Probabilistic Localization in Pictures

Robot sees another marker to its side.



Probabilistic Localization in Formulas

- Probability density for location of robot L^t is

$$B(L^t) = \int_{l^0} P(L^t, L^0 | o^t, u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0)$$

- $B(L^t)$ represents the belief state at time t.
- o^k represents the observation at time k. o^k is shorthand for $O^k = o^k$. The variable z is often used for observations.
- u^k represents the motion command at time k (or other motion estimate information).
- Let

$$B_-(L^t) = \int_{l^0} P(L^t, L^0 | u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0)$$

- This is the belief at time t before the incorporation of the latest observation.

Probabilistic Localization in Formulas

- Need to make some assumptions to make progress.
- Basic idea is to summarize the relevant information in a probability density. In most formulations, the relevant information is just the previous robot location.
- We will make a couple of conditional indepence assumptions.
- Assume $P(o^k|L^k, u^{k-1}, L^{k-1}, o^{k-1}, \ldots o^1, u^0, L^0)) = P(o^k|L^k)$
- Assume $P(L^k|L^{k-1}, u^{k-1}, o^{k-1}, \ldots o^1, u^0, L^0)) = P(L^k|u^{k-1}, L^{k-1})$
- The first assumption just says that knowing where we are gives us all the information that is relevant to the sensor reading we obtain.
- The second assumption just says that if we know where we just were and how we moved, knowing history prior to that location doesn't tell us anything new.
- This is often referred to as the Markov assumption because we assume that the system is only dependent on the current robot state and not on the full history.

Probabilistic Localization in Formulas

We can rewrite the belief state equation as follows:

$$B(L^t) = \int_{l^0} P(L^t, L^0 | o^t, u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0) =$$

$$\int_{l^0} \frac{P(o^t|L^0,L^t,u^{t-1},o^{t-1},u^{t-2},\ldots,o^1,u^0) P(L^t,L^0|u^{t-1},o^{t-1},u^{t-2},\ldots,o^1,u^0)}{P(o^t|u^{t-1},o^{t-1},u^{t-2},\ldots,o^1,u^0)} =$$

$$\int_{l^0} \frac{P(o^t|L^t) P(L^t,L^0|u^{t-1},o^{t-1},u^{t-2},\ldots,o^1,u^0)}{P(o^t|u^{t-1},o^{t-1},u^{t-2},\ldots,o^1,u^0)} =$$

$$\frac{P(o^t|L^t)}{P(o^t|u^{t-1},o^{t-1},u^{t-2},\ldots,o^1,u^0)} \int_{l^0} P(L^t, L^0 | u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0) =$$

$$\frac{P(o^t|L^t)}{\beta} B_-(L^t) =$$

$$\eta P(o^t|L^t) B_-(L^t) \propto P(o^t|L^t) B_-(L^t)$$

Probabilistic Localization in Formulas

We can rewrite the pre-sensor update belief state equation as follows:

$$B_-(L^t) = \int_{l^0} P(L^t, L^0 | u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0)$$

$$\int_{l^0, l^{t-1}} P(L^t, L^{t-1}, L^0 | u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0) =$$

$$\int_{l^0, l^{t-1}} P(L^t | L^{t-1}, L^0, u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0) *$$
$$P(L^{t-1}, L^0 | u^{t-1}, o^{t-1}, u^{t-2}, \ldots, o^1, u^0) =$$

$$\int_{l^0, l^{t-1}} P(L^t | u^{t-1}, L^{t-1}) P(L^{t-1}, L^0 | o^{t-1}, u^{t-2}, \ldots, o^1, u^0) =$$

$$\int_{l^{t-1}} P(L^t | u^{t-1}, L^{t-1}) \int_{l^0} P(L^{t-1}, L^0 | o^{t-1}, u^{t-2}, \ldots, o^1, u^0) =$$

$$\int_{l^{t-1}} P(L^t | u^{t-1}, L^{t-1}) B(L^{t-1})$$

Probabilistic Localization in Formulas

- This gives us a recursive update rule for probability density representing the robot's location (the belief state).

- We can update for motion via

$$B_{-}(L^k) = \int_{l^k-1} P(L^k|u^{k-1}, L^{k-1})B(L^{k-1})$$

- We can update for sensor readings via

$$B(L^k) \propto P(o^k|L^k)B_{-}(L^k)$$

- We can initialize the system with $B(L^0) = P(L^0)$.

Probabilistic Localization in Formulas

- For these update equations to be useful, we need two conditional probability densities which act as models of the robot.

- The motion model relates the robot's current location to its location after moving.

$$P(L^k|u^{k-1}, L^{k-1})$$

- The sensor model relates the robot's position to its sensor readings.

$$P(o^k|L^k)$$

Probabilistic Localization in Formulas

- We now have two update rules for the belief state.
- The motion update

$$B_-(L^k) = \int_{l^{k-1}} P(L^k|u^{k-1}, L^{k-1})B(L^{k-1})$$

- This operation is known as a convolution.
- The sensor update

$$B(L^k) \propto P(o^k|L^k)B_-(L^k)$$

- This just corresponds to a multiplication by the likelihood of the sensor readings.

Probabilistic Localization in Practice

- We now have a way to update the probabiliry density of the robot's location (belief state) as we move and get sensor readings.
- We still need a way to represent these probability densities.
- In general, it is not possible to represent these probability densities in closed form. As more information gets incorporated, the probability densities take on more and more complex forms.

Probabilistic Localization in Practice

- Two basic approached to this problem.
- First approach is to assume the sensor readings and process have a particular form.
- If we assume that the system is linear, the process noise is Gaussian, and the sensor noise is Gaussian, the problem can be solved in closed form. This leads to the Kalman filter which we will cover later.
- Second approach is to approximate the pose probability density (belief state).

Probabilistic Localization in Practice

Different approaches to approximation:

- Use a piecewise constant grid.
  - Computationally expensive unless carefully implemented.
  - Computation expense increases quickly with resolution.
  - Results in a poor aproximation of the pose probability density.
  - Difficult to implement.

- Approximate with a set of samples (particles).
  - This technique is called a particle filter.
  - Good approximation in high probability areas with reasonable computation.
  - This approach is generally recognized as superior to grid approximations.
  - We will cover it is much more detail next.

Particle Filters

- Idea is to represent pose probability density by a set of sample robot poses $(x_0, x_1, \ldots, x_{n-1})$.
- The number of samples n is usually between 50 and 400 samples.
- The lower sample numbers require a few special techniques which we will discuss next week. This is caused by the relatively poor approximation of the belief state they provide.
- Complex probability densities tend to require more points for an adequate approximation.

Particle Filters

- The density of the samples in a region of pose space is proportional to the probability that the robot is in that region.
- The count of the samples in a region of pose space takes the place of an intergral over pose space.
- Any desired summary statistics can be calculated as follows:

$$\int f(l)B(L=l)dl \approx \sum_i f(x_i)/n$$

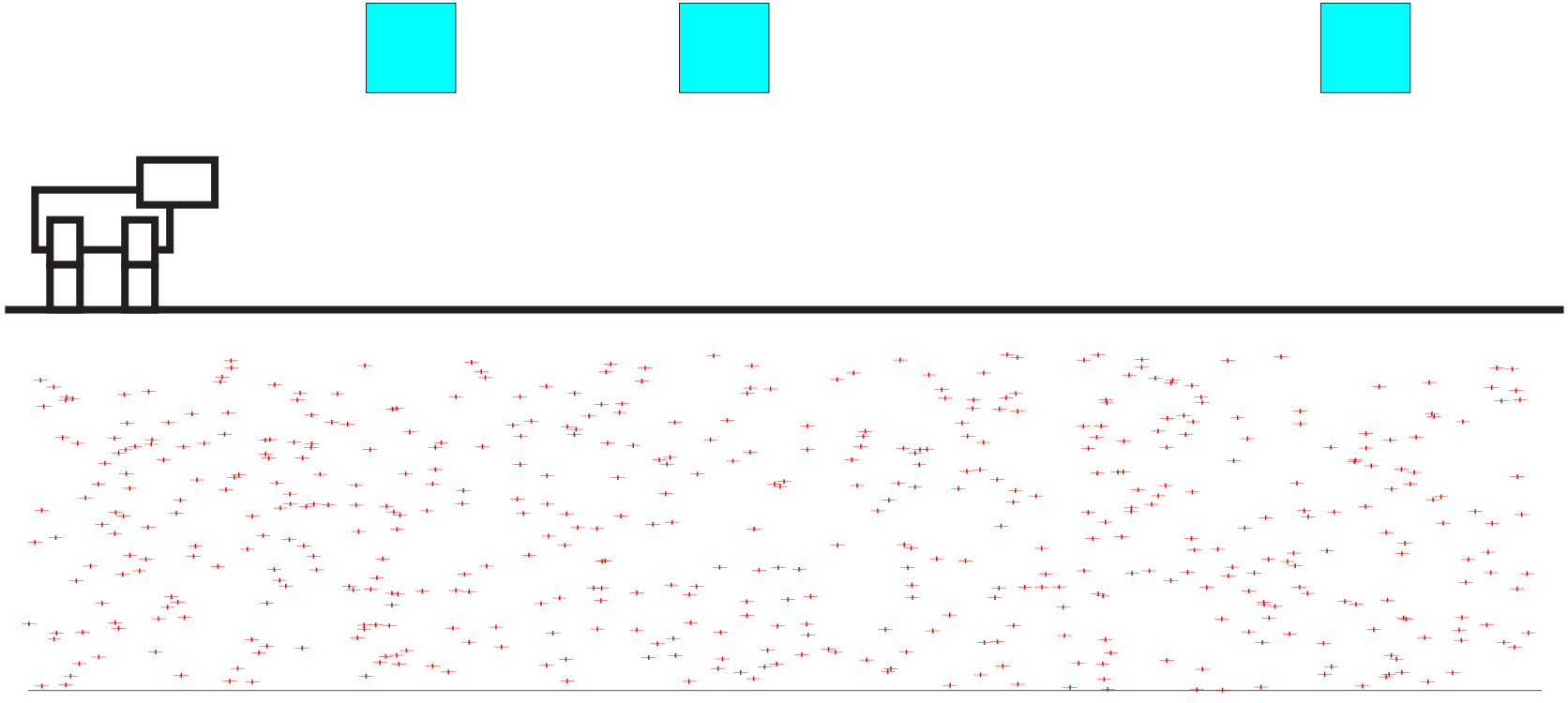
- Each sample also has an associated weight w_i. The sample is counted a number of times equal to its weight, so a sample with a weight of 3 counts as 3 samples, with a weight of .5 counts as .5 samples, etc.

Particle Filters - Initialization

- Remember, we need to initialize the localization with an initial belief state $B(L^0)$.
- Usually, the initial belief state is initialized to a uniform distribution over the entire environment the robot might be located in.
- A uniform distribution is easy to create in samples. We just need to create n samples randomly over the entire environment.

Particle Filters - Initialization

Robot starts off lost.



Particle Filters - Motion Update

Recall the motion update has the form

$$B_-(L^k) = \int_{l^k-1} P(L^k|u^{k-1}, L^{k-1})B(L^{k-1})$$

By the relationship between integrals over probability densities and sums over samples, this becomes

$$B_-(L^k) = \sum_i P(L^k|u^{k-1}, L^{k-1} = x_i^{k-1})/n$$

But $P(L^k|u^{k-1}, L^{k-1} = l)$ is itself a probability density that we will need to approximate.

Particle Filters - Motion Update

- One solution would be to reuse the sampling idea.
- Generate a whole bunch of samples of $P(L^k|u^{k-1}, L^{k-1} = l)$ and use them to form the next belief state.
- But we would like to keep the number of samples constant over time.
- Instead, we will generate a set of samples that have the same expected probability of occurance as the large sample set idea.
- We can do this simply by taking one sample from $P(L^k|u^{k-1}, L^{k-1} = l)$ and using that sample!
- This preserves all expectations over the belief state.

Particle Filters - Motion Update

In pseudocode, this looks like:

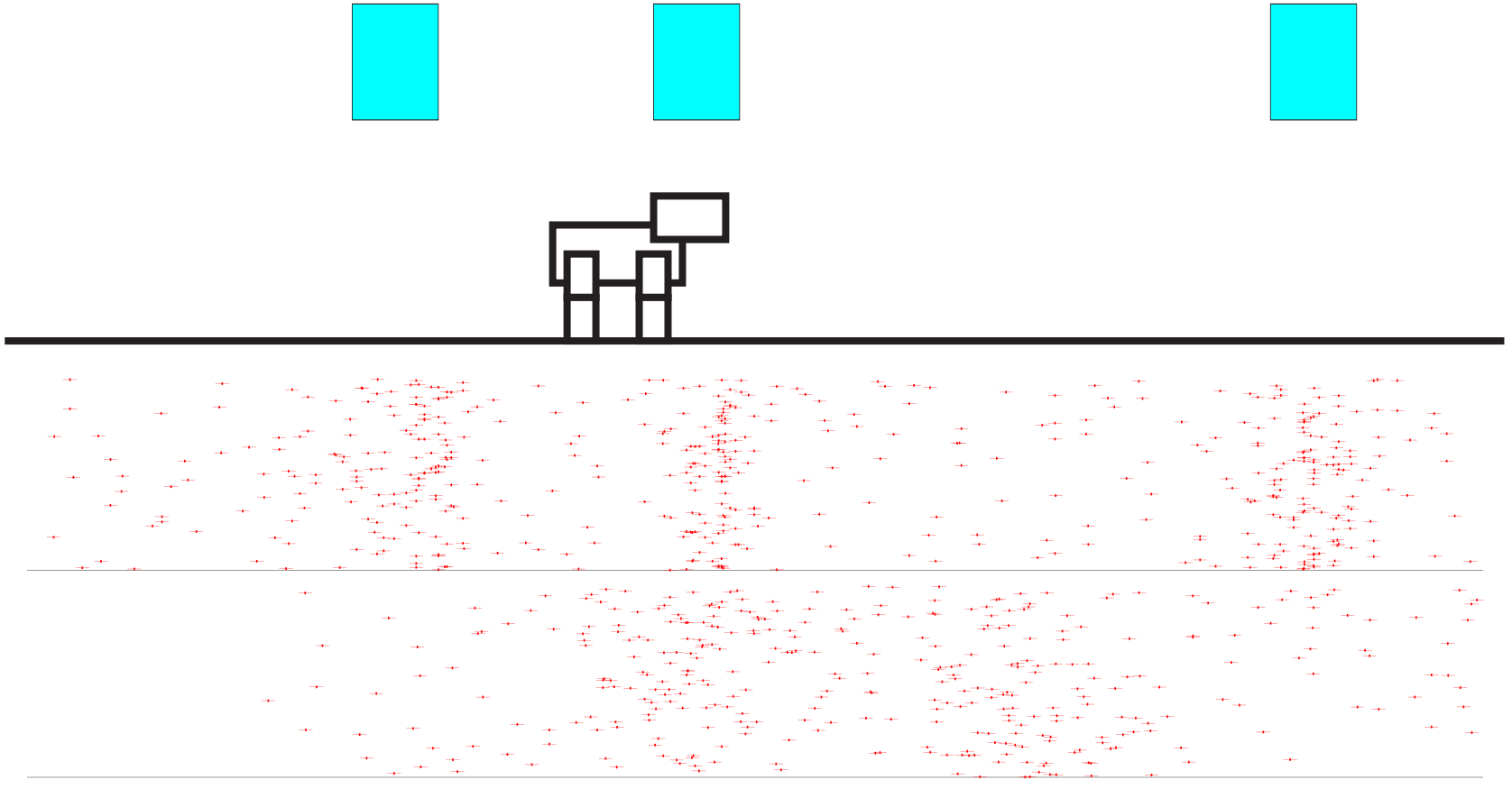
For each sample x_i^{k-1} in $B(L^{k-1})$

Generate sample $x_{-,i}^{k-1}$ from $P(L^k|u^{k-1}, L^{k-1} = x_i^{k-1})$

Add $x_{-,i}^{k-1}$ with weight w_i^{k-1} to $B_-(L^k)$

Particle Filters - Motion Update

Robot moves to right.



Particle Filters - Sensor Update

- Recall the sensor update has the form

$$B(L^k) \propto P(o^k|L^k)B_-(L^k)$$

- We can perform the same operation on the samples using only the weights

$$w_i^k = P(o^k|L^k = x_{-,i}^k) * w_{-,i}^k$$

- The new samples have the same positions as the old samples but different weights.

- The probability density is implicitly normalized by the sum of the weights.

Particle Filters - Sensor Update

In pseudocode, this looks like:

For each sample $x^k_{-,i}$ in $B_-(L^k)$

$w^k_i = P(o^k|L^k = x^k_{-,i}) * w^k_{-,i}$

$x^k_i = x^k_{-,i}$

Add x^k_i with weight w^k_i to $B(L^k)$

Particle Filters - Sensor Update

- This update rule works but over time the weights can become very uneven.
- This can lead to a poor approximation of the probabilty density.
- We would like the average level of the weights to remain constant over time.
- To solve this problem, we will introduce a renormalization step.
- This renormalization step is often called "Sample Importance Resampling" for reasons that will be apparent shortly.

Particle Filters - Sensor Update

- We need an operation that preserves the expected density of weights of the samples while renormalizing their weights.
- We will generate a new set of samples with equal weight but the same expected density of weights.
- Solution is to sample from the samples in proportion to their weights.

Particle Filters - Resampling

In pseudocode, this looks like:

For each sample y_i^k we want in $B'(L^k)$

Select sample x_j^k from $B(L^k)$ in proportion to their weights.

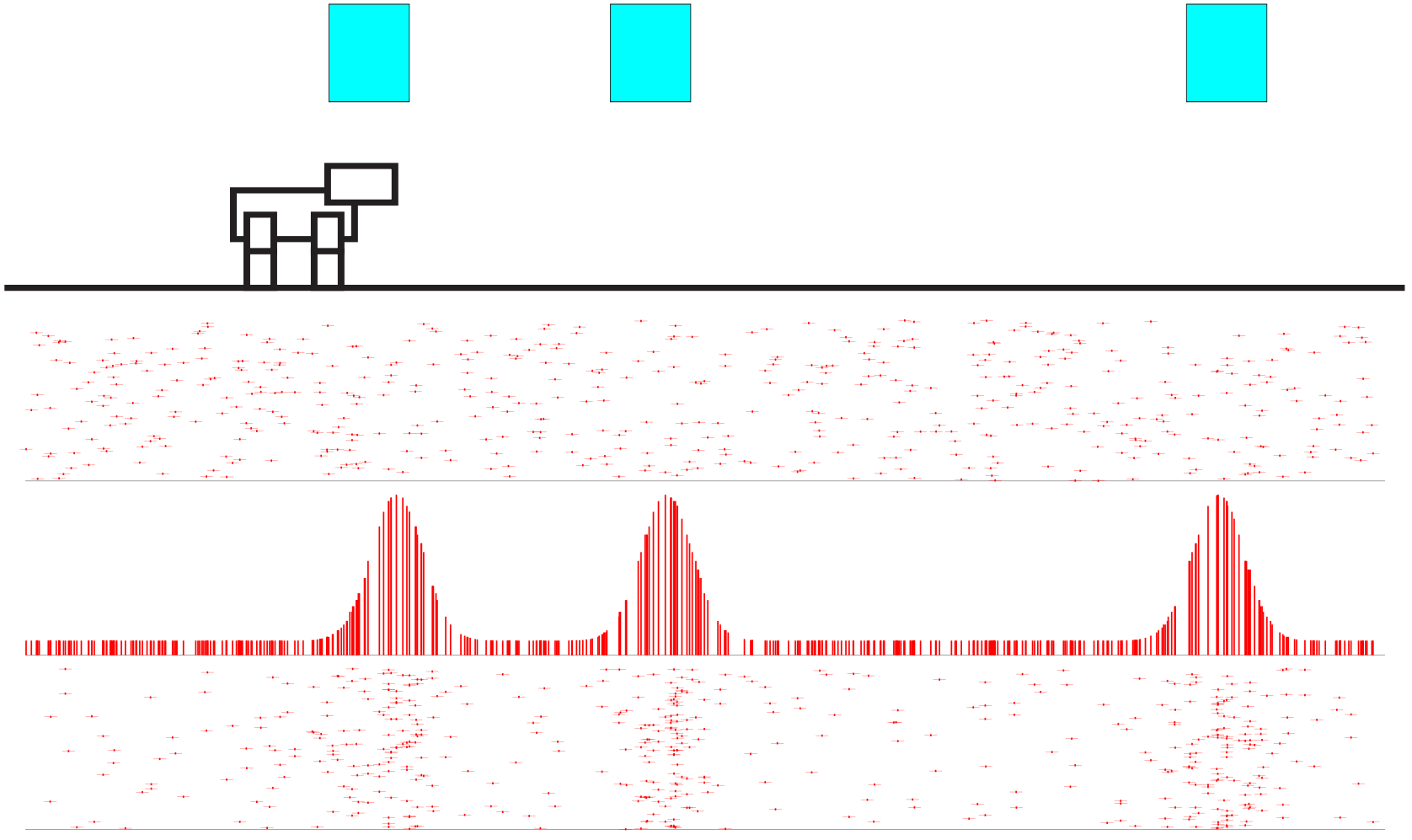
$y_i^k = x_j^k$

Add y_i^k with weight 1 to $B'(L^k)$

Replace $B(L^k)$ with $B'(L^k)$

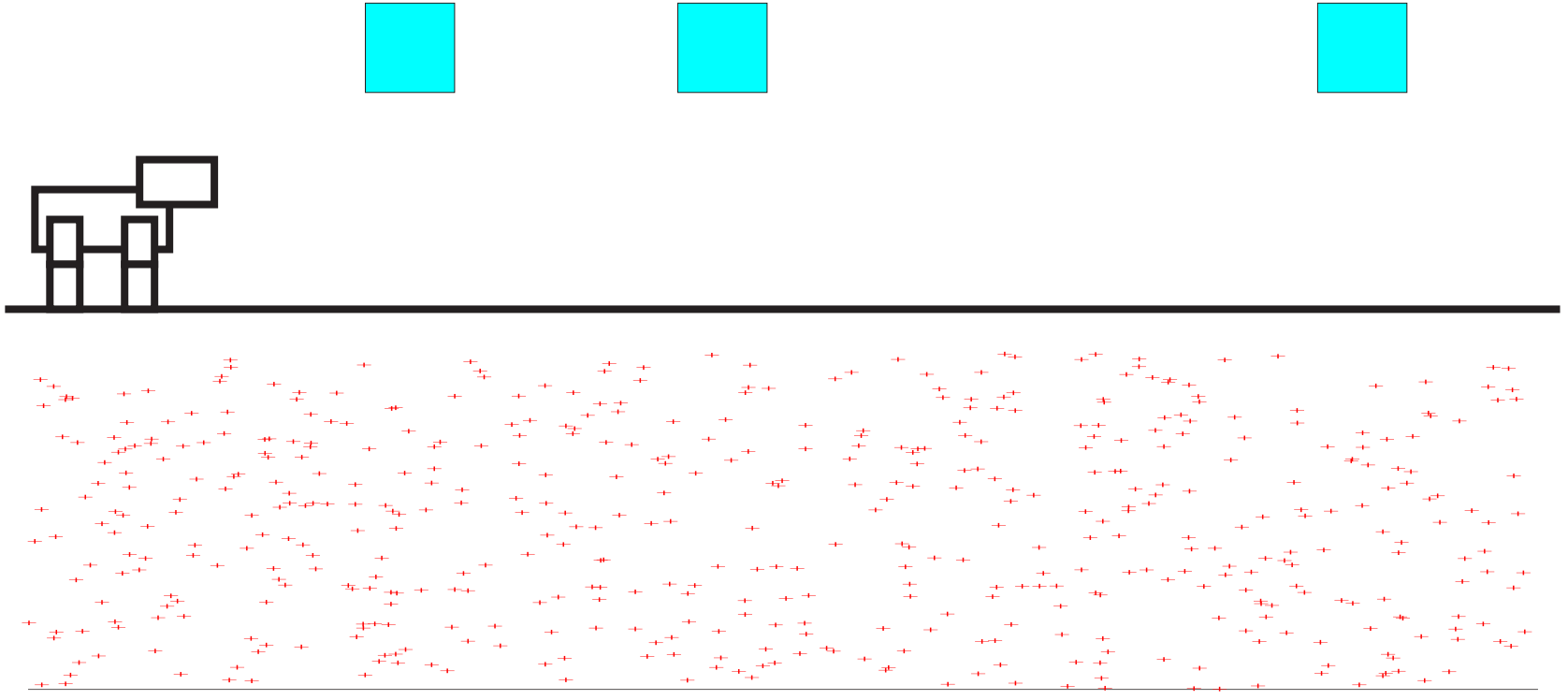
Particle Filters - Sensor Update

Robot sees a marker to its side.



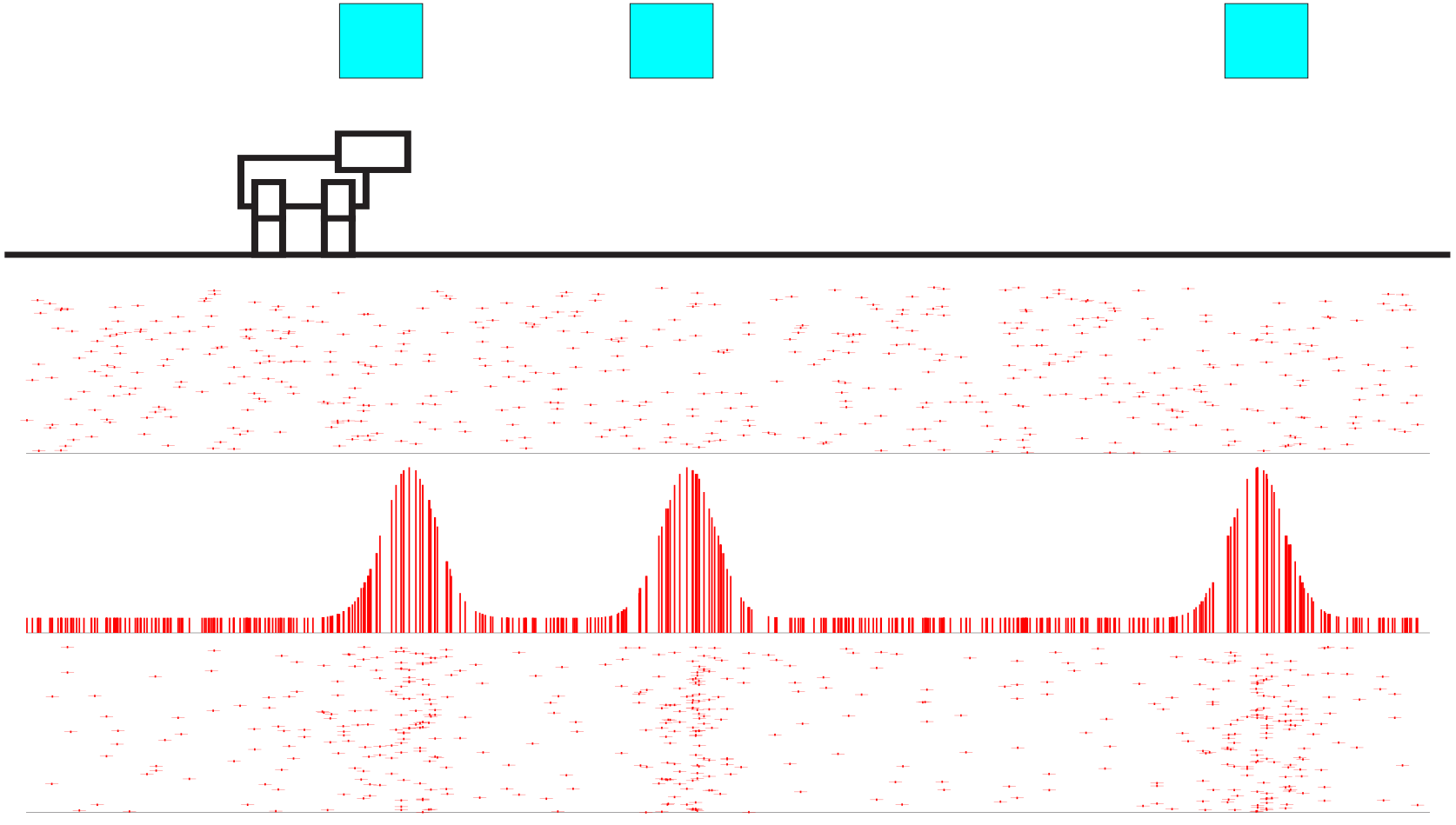
Particle Filters in Pictures

Robot starts off lost.



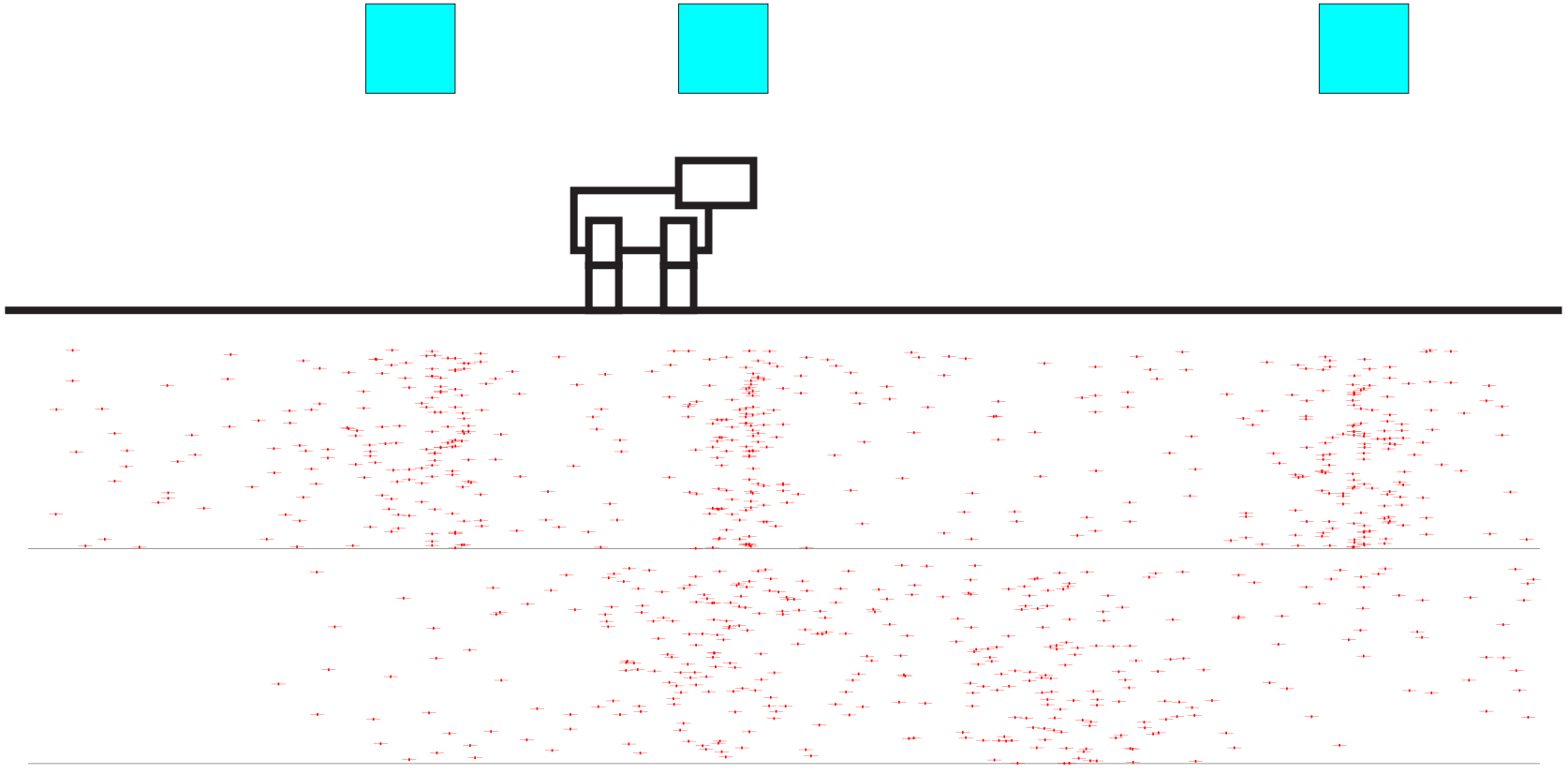
Particle Filters in Pictures

Robot sees a marker to its side.
The first density is before the update and the last is after the update.



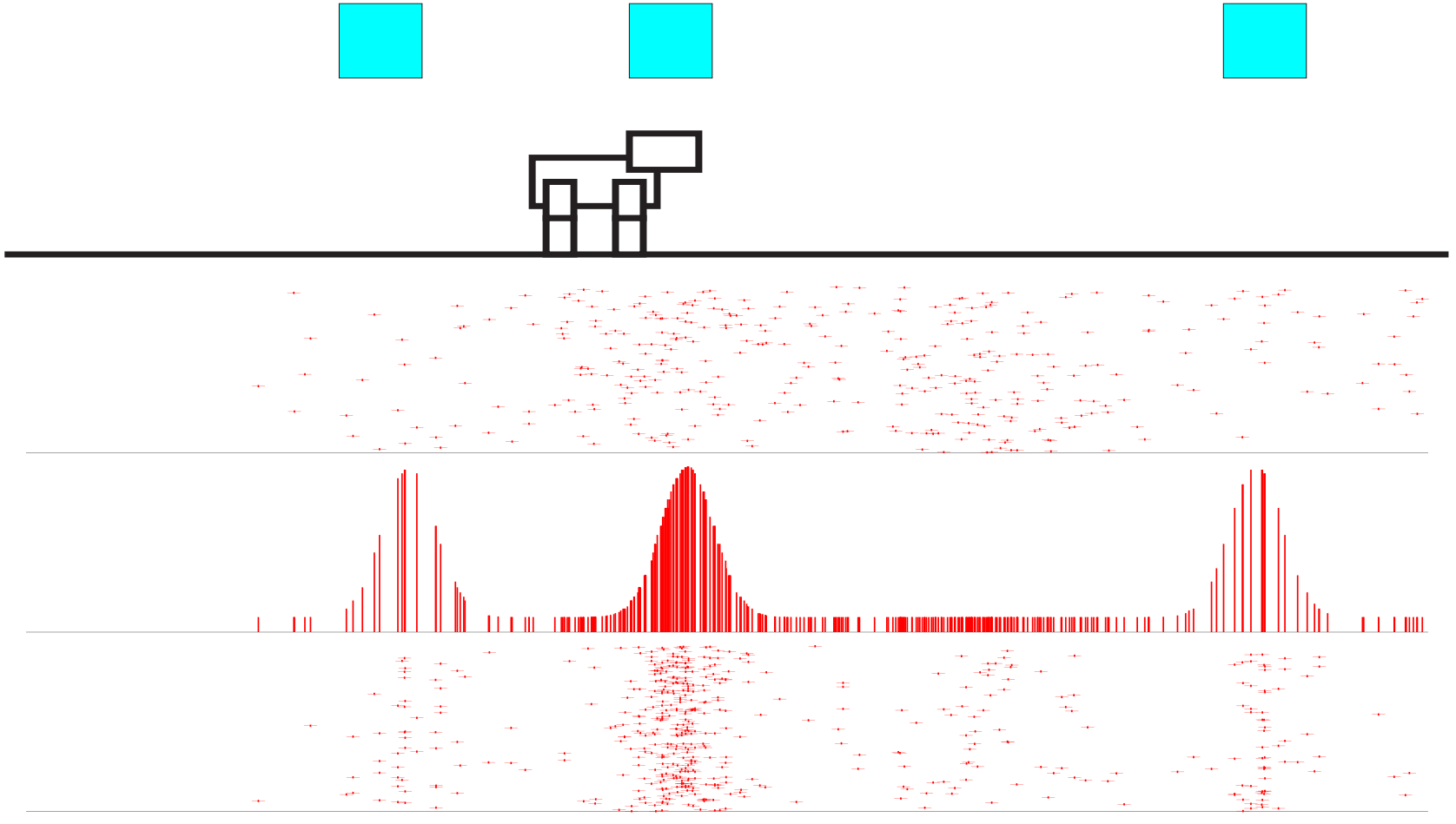
Particle Filters in Pictures

Robot continues moving to right.
The first density is before the update and the last is after the update.

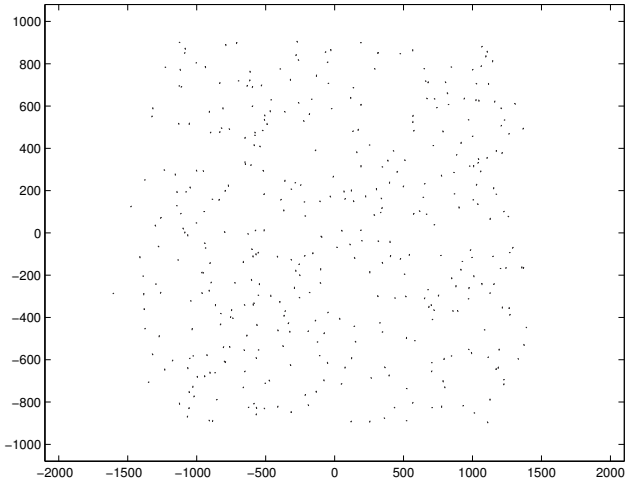


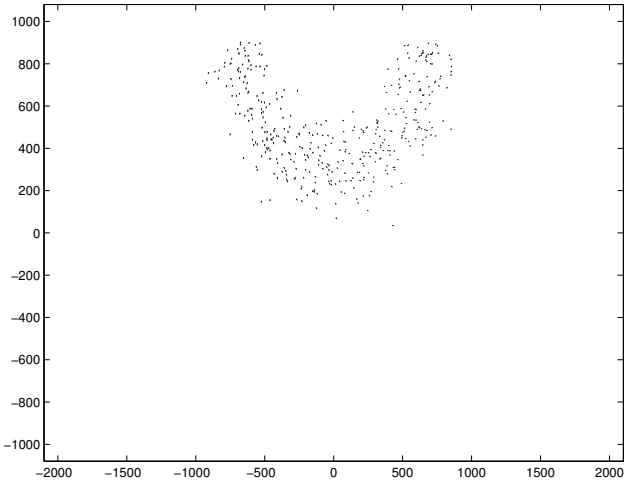
Particle Filters in Pictures

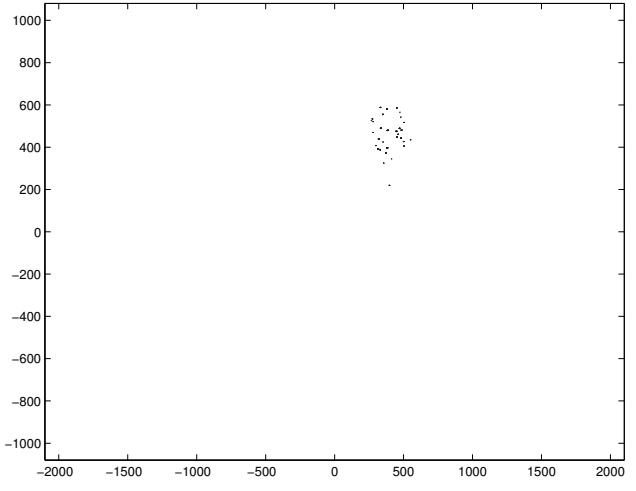
Robot sees another marker to its side.
The first density is before the update and the last is after the update.



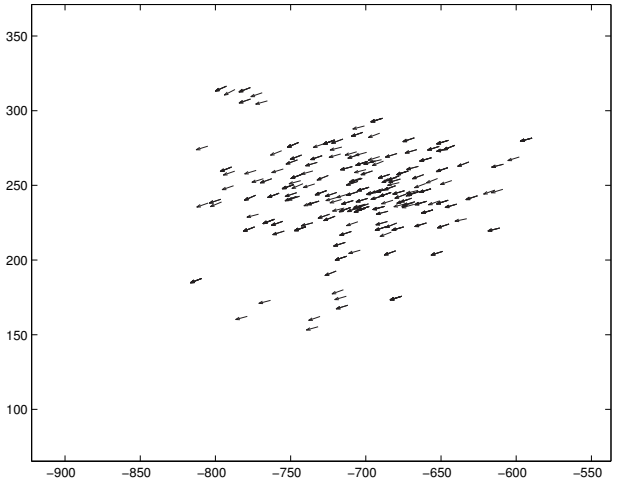
Particle Filters - Example Densities

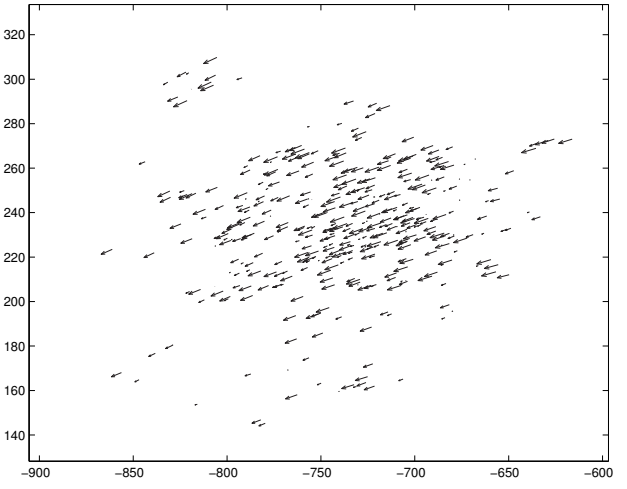


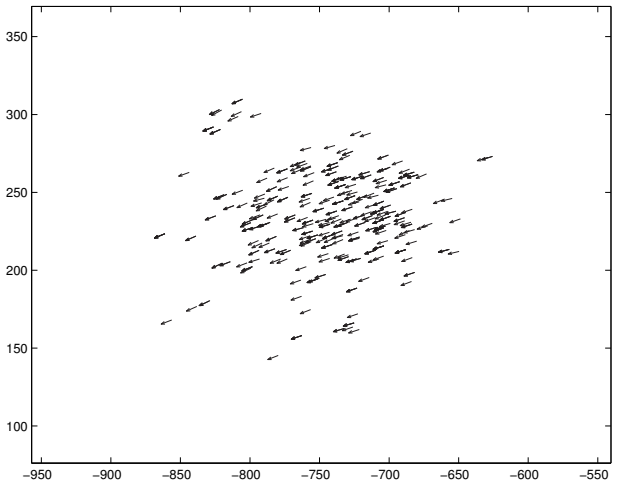




Particle Filters - Example Densities







Particle Filters - Information Extraction

- Now that we can track a belief state about the robot's pose, we need to get this information to behaviors in a useful form.
- Behaviors are interested in knowing the most likely pose of the robot.
- We can approximate this by calculating the mean pose of the robot.
- Each position coordinate of this mean pose can be calculated independently from the sample set as a simple weighted average.

$$\bar{x} = \sum_i x_i.x * w_i / \sum_i w_i$$

- The heading of this mean pose can be calculated using a weighted average over heading vectors computed from each sample.

Particle Filters - Information Extraction

- Behaviors also need a way to judge the reliability of the estimate.
- The covariance matrix provides a convenient way to summarize the uncertainty in the result.
- This reports the variance in each coordinate and their correlation.
- The covariance matrix can be easily calculated from the samples using the standard statistics formulas.
- This information can be used to actively trigger localization amongst other uses.