

# Lecture 02

## SmartPhone Sensing

# Announcements

- Have you started programming on Android?
  - Are you using the sample code in Github?
  - Have you formed groups in excel sheet
  - Novel twists, no resit
- 
- Soon:
  - Select your final evaluation slot.

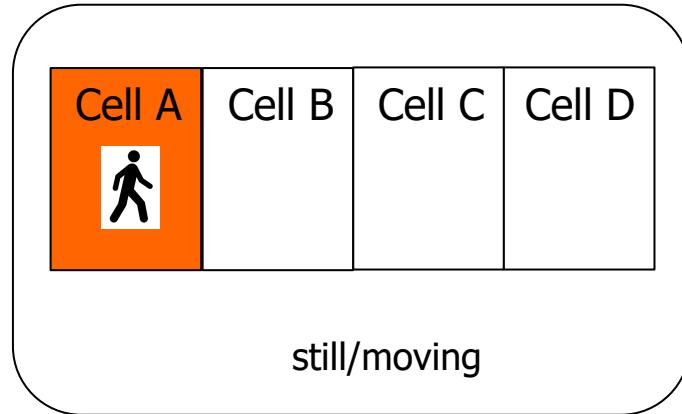
# Today's menu

- Recap/Finish KNN for localization
- Localization
  - Bayesian Inference (Bayesian Filters)
  - Particle Filters
- Parts of this lecture are based on content from
  - "Artificial Intelligence for Robotics"
    - <https://www.udacity.com/course/cs373>
    - This is an \*excellent\* course from Udacity.
  - "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking"
    - <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00978374>

# Step by Step guidelines for first report

- Step 1) Get rss and acc data
  - Use Android sample code
- Step 2) Identify features to **train** your data.
  - Divide the raw data in windows
  - For each window identify features
  - Label training data and save it in the phone
- Step 3) **test** your data
  - Get raw data, divide it windows, get features
  - Apply KNN to classify your data
  - Show the classified label in the phone

# Guidelines for App



- (1) Determine where you are among four zones and
- (2) Determine if you are still or walking

# First report: Option 1 (5% grade)

- Deadline: midnight on May 7<sup>th</sup>.
- Submission will be via feedbackfruits.
- You can perform your experiments at home or at the university.
- A detailed template for the report is posted in feedbackfruits.
- Max 1-page text (double/single-column ) + max 1-page for figures, tables, references.
- If you can't complete the first assignment, it would be wise to reconsider taking the course.

# Code & load sharing

*“Software engineers do not like to write code, they like to copy/paste it”, anonymous*

- Build on top of what it is there, but cite it in your report.
- By all means share knowledge and pointers amongst yourselves but give credit to your (peer) source in your report.
- Balance the work. Else one passes and the other doesn't.

wait ...

you can also do a bunch of other  
useful stuff with accelerometers

# **LET'S HAVE A BIRD'S EYEVIEW OF THE TECHNIQUES**

# The core of the course

App option 1: Localization

Basic localization &  
Activity monitoring

kNN

Advanced localization

**Bayesian filters      Particle filters**

Log normal  
shadowing

Visible Light  
Communication

# CONDITIONAL PROBABILITIES

# Quick test on conditional probabilities

- $P(A|B) = P(A \text{ and } B) / P(B)$   
=  $P(B|A) P(A)/P(B)$
- The probability a person is 1.90 m tall is 0.05, the probability a person is Dutch and is 1.90 m tall is 0.01. What is the probability that a person is Dutch given that the person is 1.90 m tall?

# Solution

A = person is Dutch

B = person is 1.9m tall

$$P(A \text{ and } B) = 0.01$$

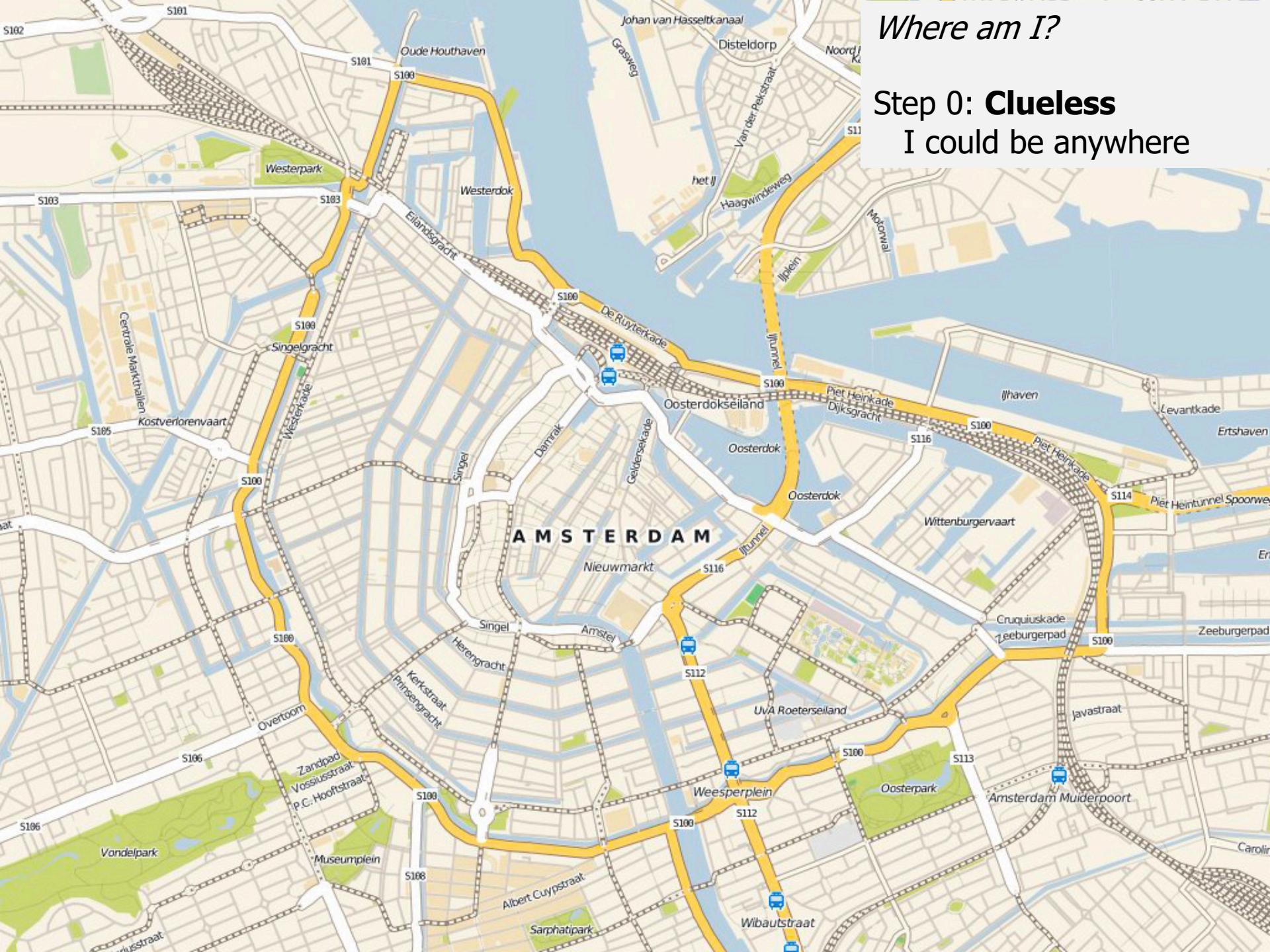
$$P(B) = 0.05$$

$$\begin{aligned} P(A|B) &= P(A \text{ and } B)/P(B) \\ &= 0.01/0.05 \\ &= 20\% \end{aligned}$$

# LOCALIZATION BASICS

*Where am I?*

**Step 0: Clueless**  
I could be anywhere



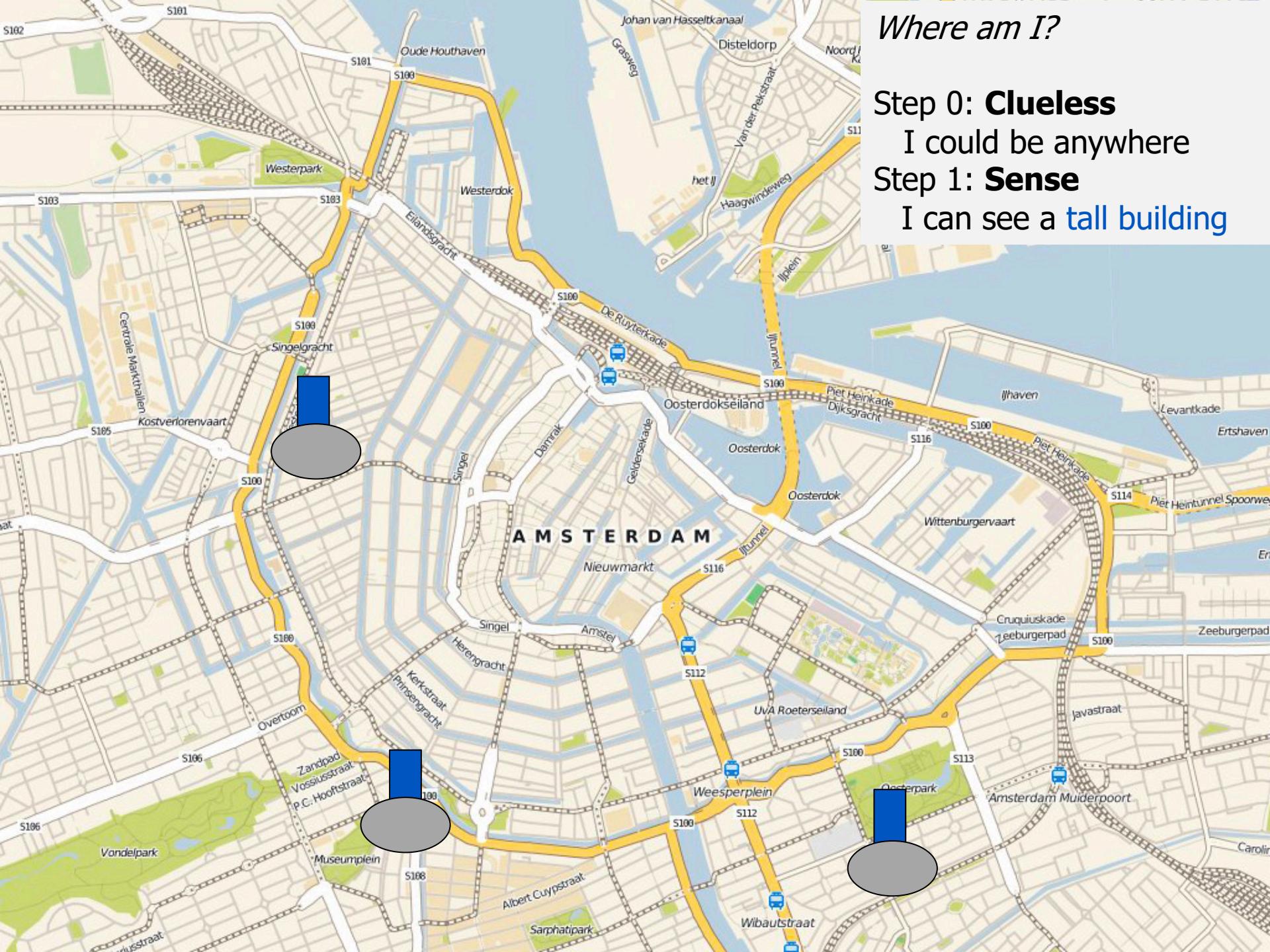
Where am I?

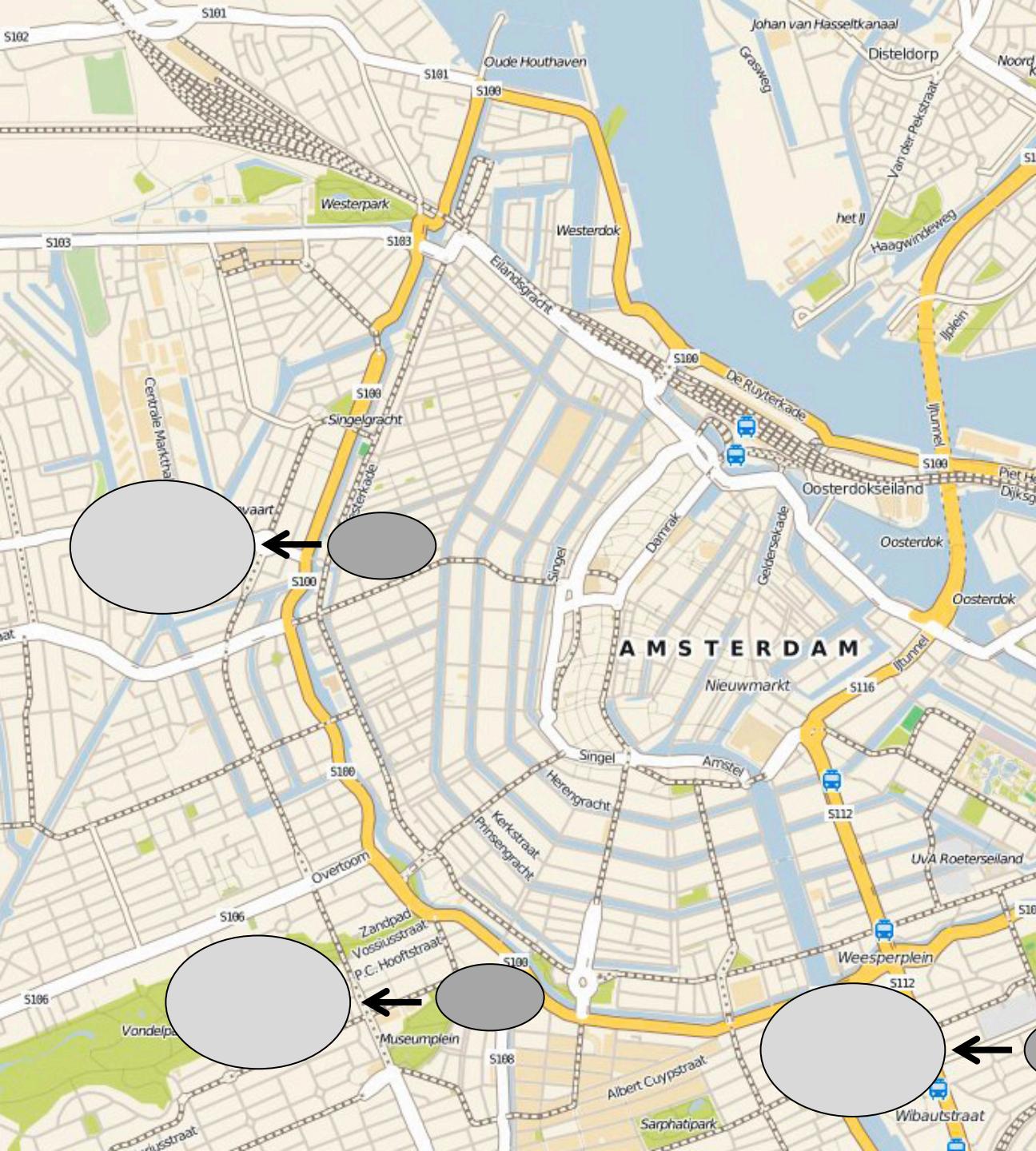
Step 0: **Clueless**

I could be anywhere

Step 1: **Sense**

I can see a **tall building**





## *Where am I?*

# Step 0: Clueless

# I could be anywhere

## Step 1: Sense

I can see a tall building

## Step 2: Move

-



Where am I?

Step 0: **Clueless**

I could be anywhere

Step 1: **Sense**

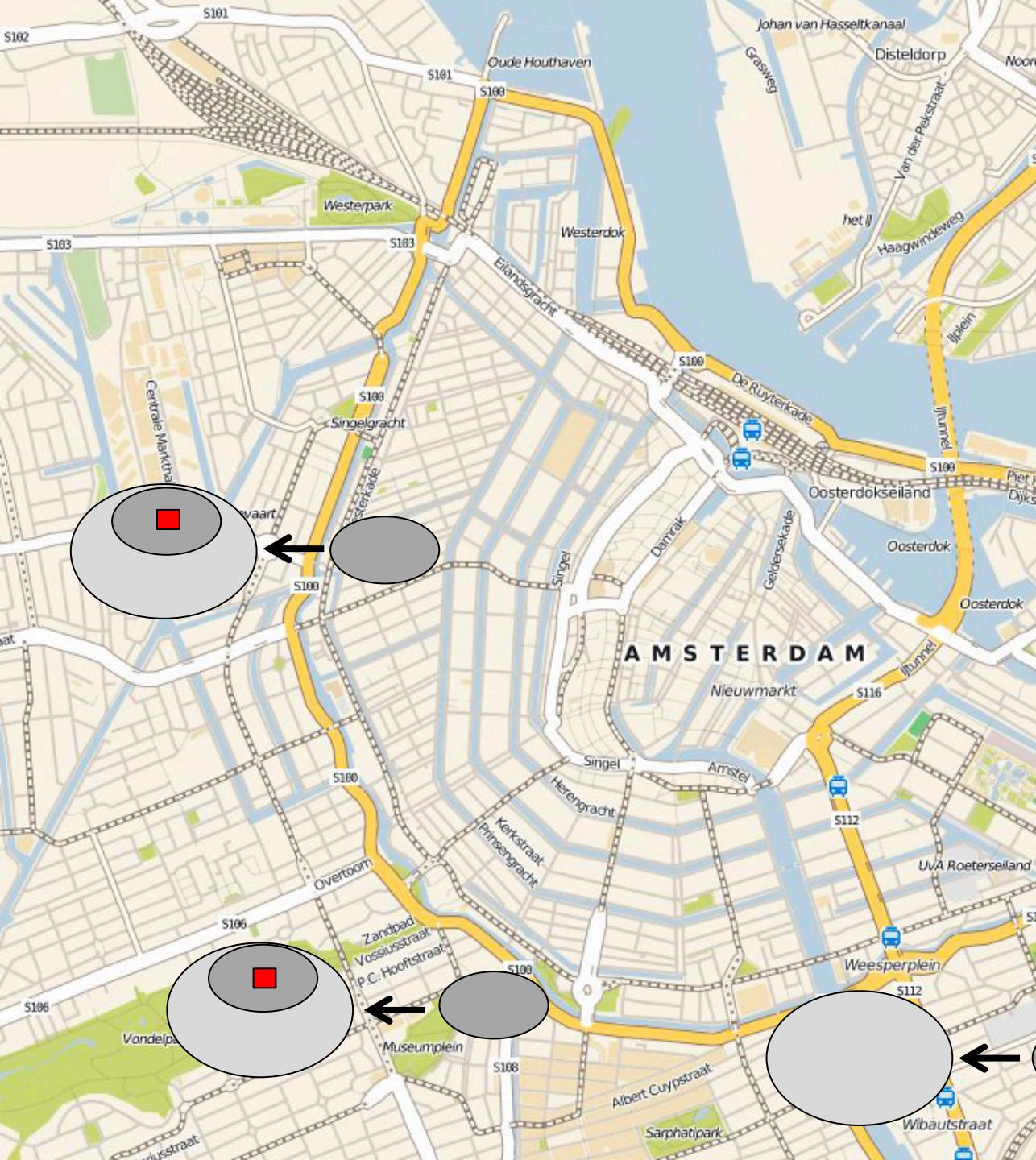
I can see a **tall building**

Step 2: **Move**

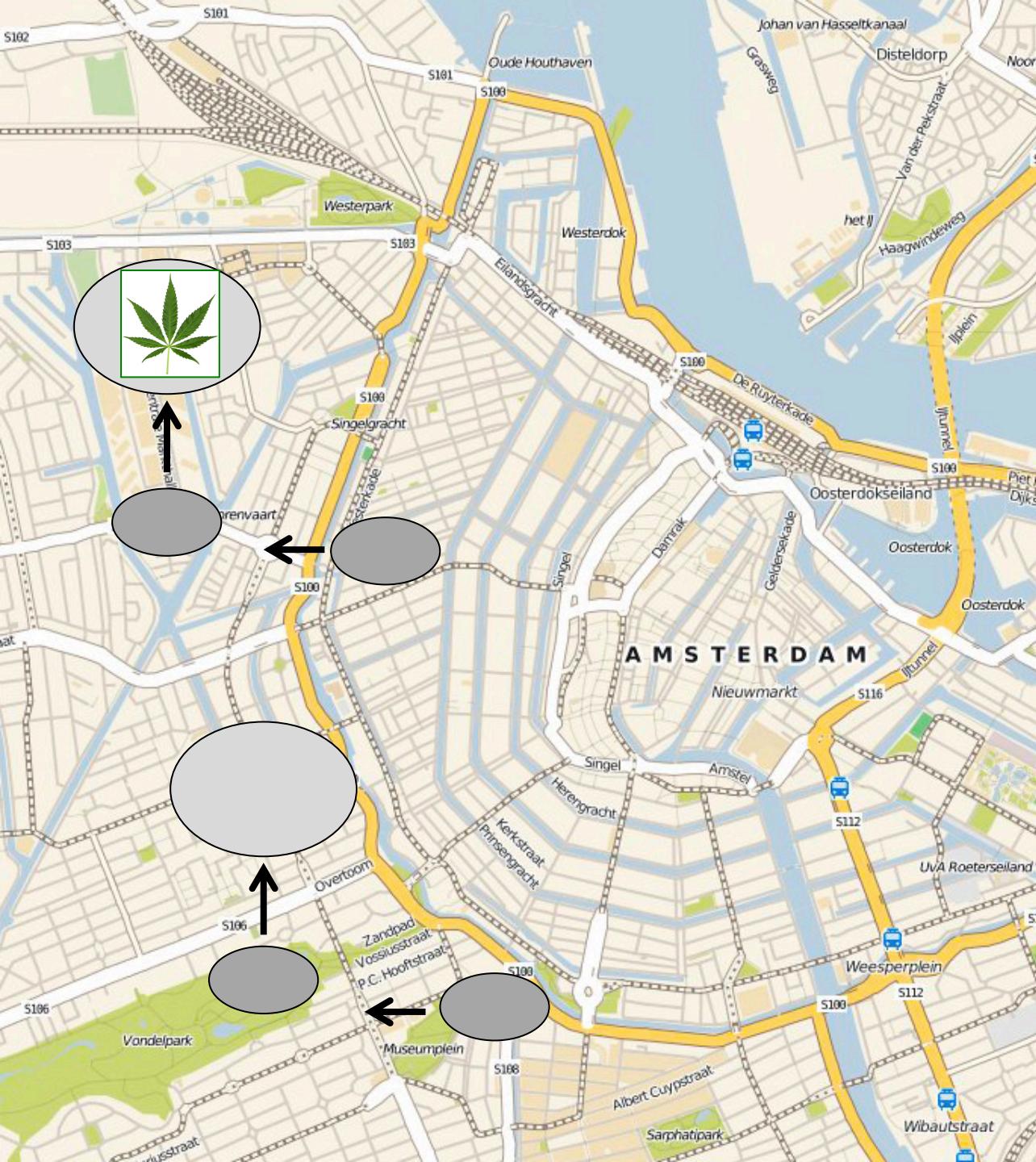
oops!

Step 3: **Sense (again)**

I can see a **super market**



*Where am I?*



**Step 0: Clueless**

I could be anywhere

**Step 1: Sense**

I can see a **tall building**

**Step 2: Move**

oops!

**Step 3: Sense (again)**

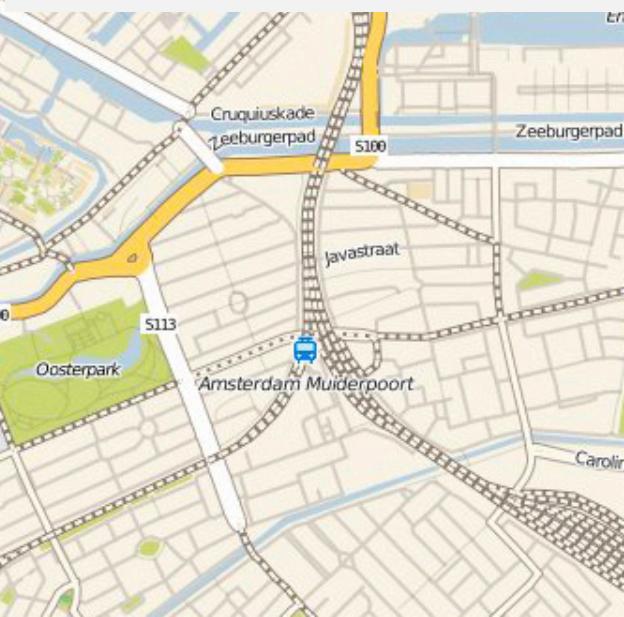
I can see a **super market**

**Step 4: Move (again)**

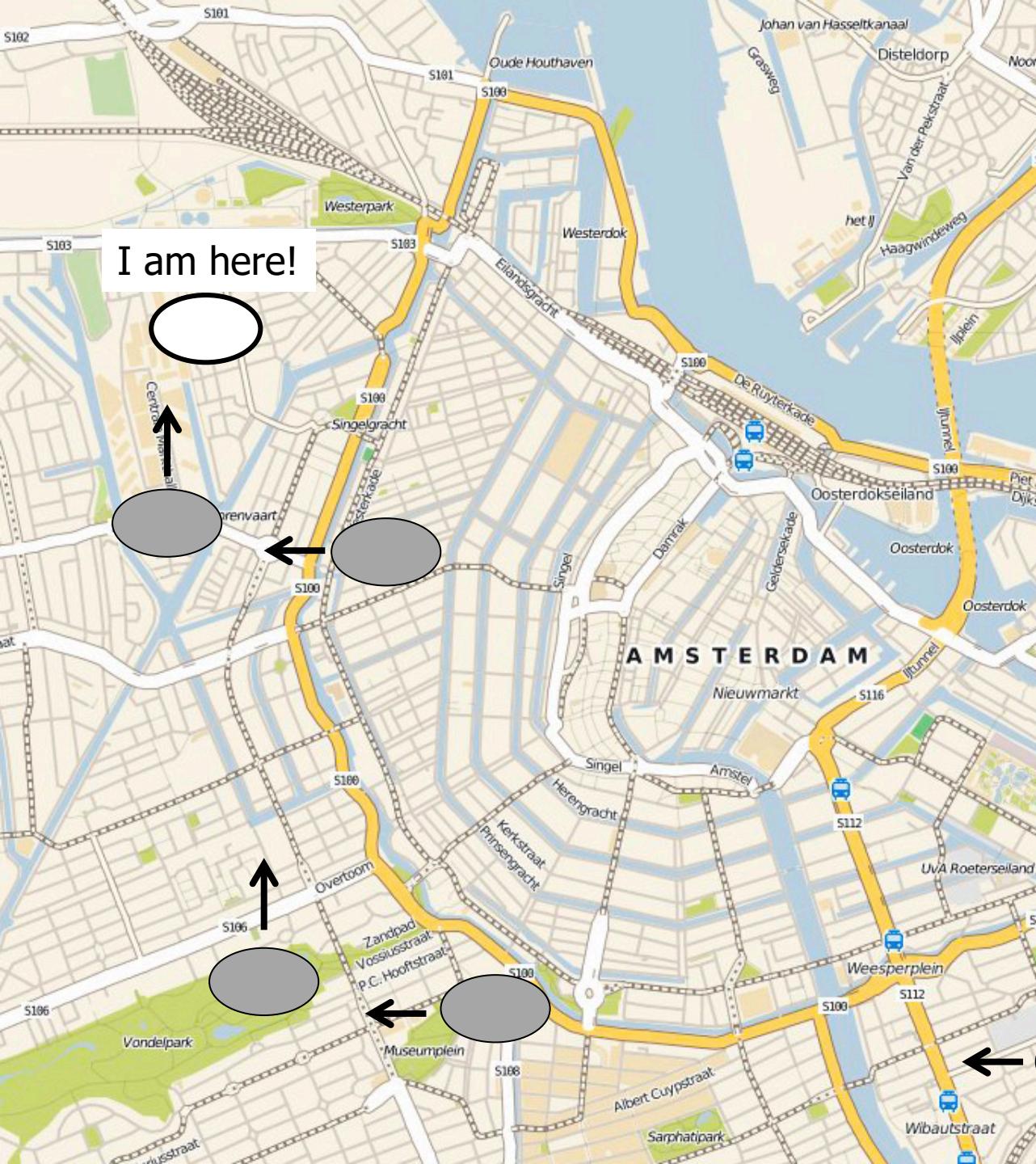
oops!

**Step 5: Sense (again)**

I can see a "**coffee shop**"



*Where am I?*



I am here!

**Step 0: Clueless**

I could be anywhere

**Step 1: Sense**

I can see a **tall building**

**Step 2: Move**

oops!

**Step 3: Sense (again)**

I can see a **super market**

**Step 4: Move (again)**

oops!

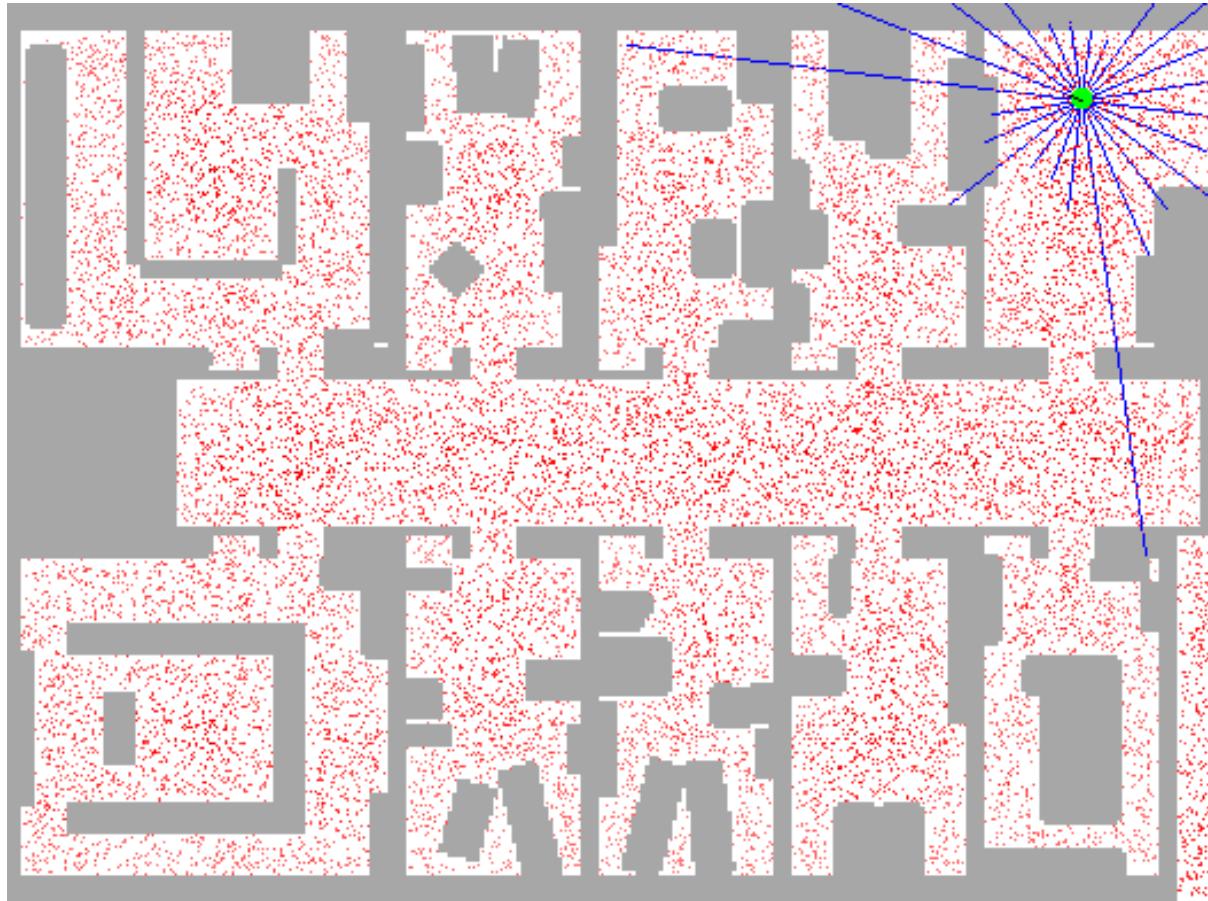
**Step 5: Sense (again)**

I can see a "**coffee shop**"

*I know where I am!*



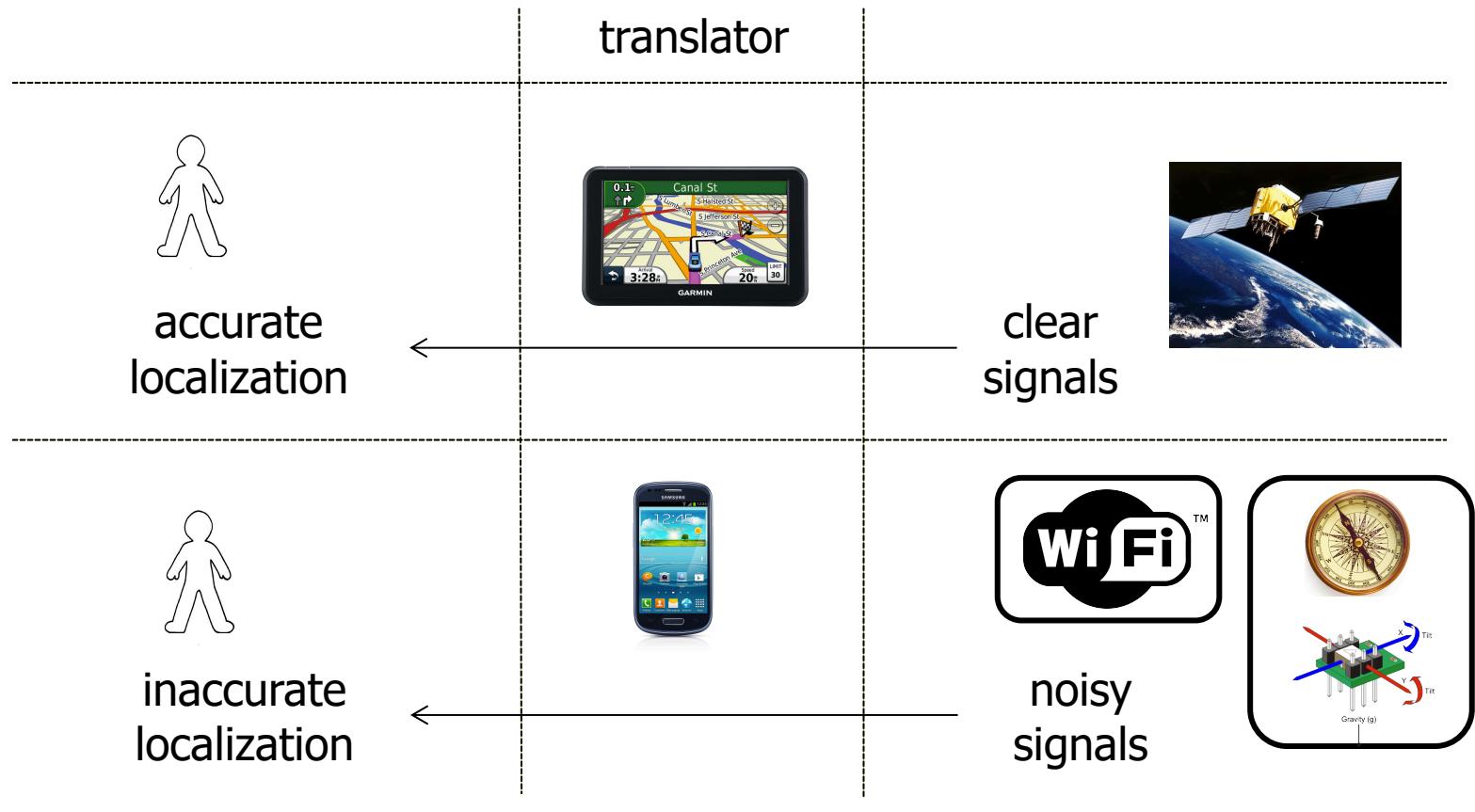
# Now, a real evaluation of particle filters



movie link: <http://robots.stanford.edu/movies/sca80a0.avi>

# The problem

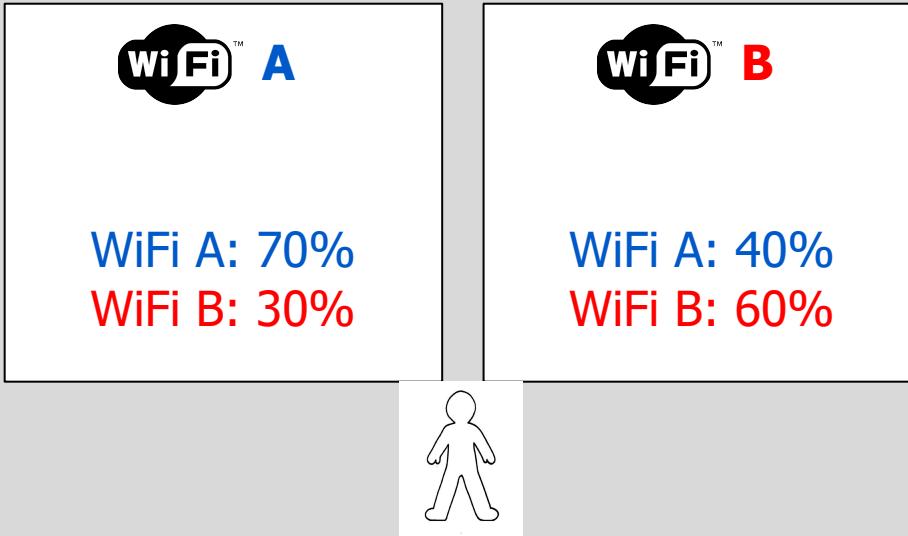
You only get indirect noisy information of what you want



# Teaser

Room A

Room B



$$\begin{aligned} P(A/B) &= P(A \text{ and } B) / P(B) \\ &= P(B/A) P(A)/P(B) \end{aligned}$$

Initially the person can be anywhere.  
Either on cell A or cell B. What is the  
probability that the person is in cell A,  
given that it hears WiFi A?

# Solution

$A$  = person is in cell A

$B$  = person is in cell B

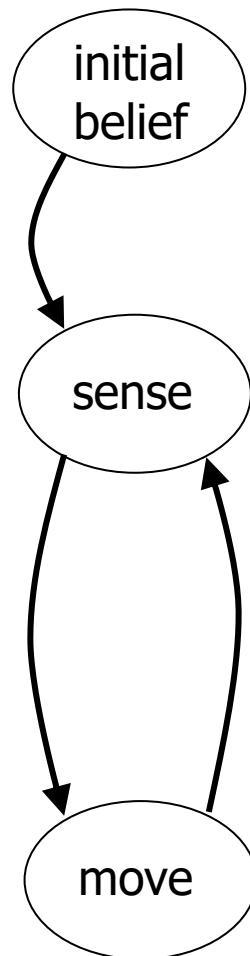
$Wifi\_A$  = person hears WiFi A

$$\begin{aligned} P(A/Wifi\_A) &= P(Wifi\_A/A) * P(A) / P(Wifi\_A) \\ &= 0.7 * 0.5 / P(Wifi\_A) \end{aligned}$$

$$\begin{aligned} P(Wifi\_A) &= P(Wifi\_A/A) * P(A) + P(Wifi\_A/B) * P(B) \\ &= 0.7 * 0.5 + 0.4 * 0.5 = 0.55 \end{aligned}$$

$$P(A/Wifi\_A) = 0.35 / 0.55 = 7/11$$

You may have not realized it, but you already know the gist of Particle (and Bayesian) filters! Below,  $\mathcal{X}$  represents location and  $\mathcal{Z}$  measurements



current pdf  
**(posterior)**

$p(\mathcal{X}_k | \mathcal{Z}_{1:k}) = \frac{p(\mathcal{Z}_k | \mathcal{X}_k) p(\mathcal{X}_k | \mathcal{Z}_{1:k-1})}{p(\mathcal{Z}_k | \mathcal{Z}_{1:k-1})}$

perception model  
(sense)

pdf from last time step  
(prior)

$$p(\mathcal{X}_k | \mathcal{Z}_{1:k}) = \frac{p(\mathcal{Z}_k | \mathcal{X}_k) p(\mathcal{X}_k | \mathcal{Z}_{1:k-1})}{p(\mathcal{Z}_k | \mathcal{Z}_{1:k-1})}$$

normalization

current pdf  
**(posterior)**

$p(\mathcal{X}_k | \mathcal{Z}_{1:k-1}) = \int p(\mathcal{X}_k | \mathcal{X}_{k-1}) p(\mathcal{X}_{k-1} | \mathcal{Z}_{1:k-1}) d\mathcal{X}_{k-1}$

motion model  
(move)

pdf from last time step  
(prior)

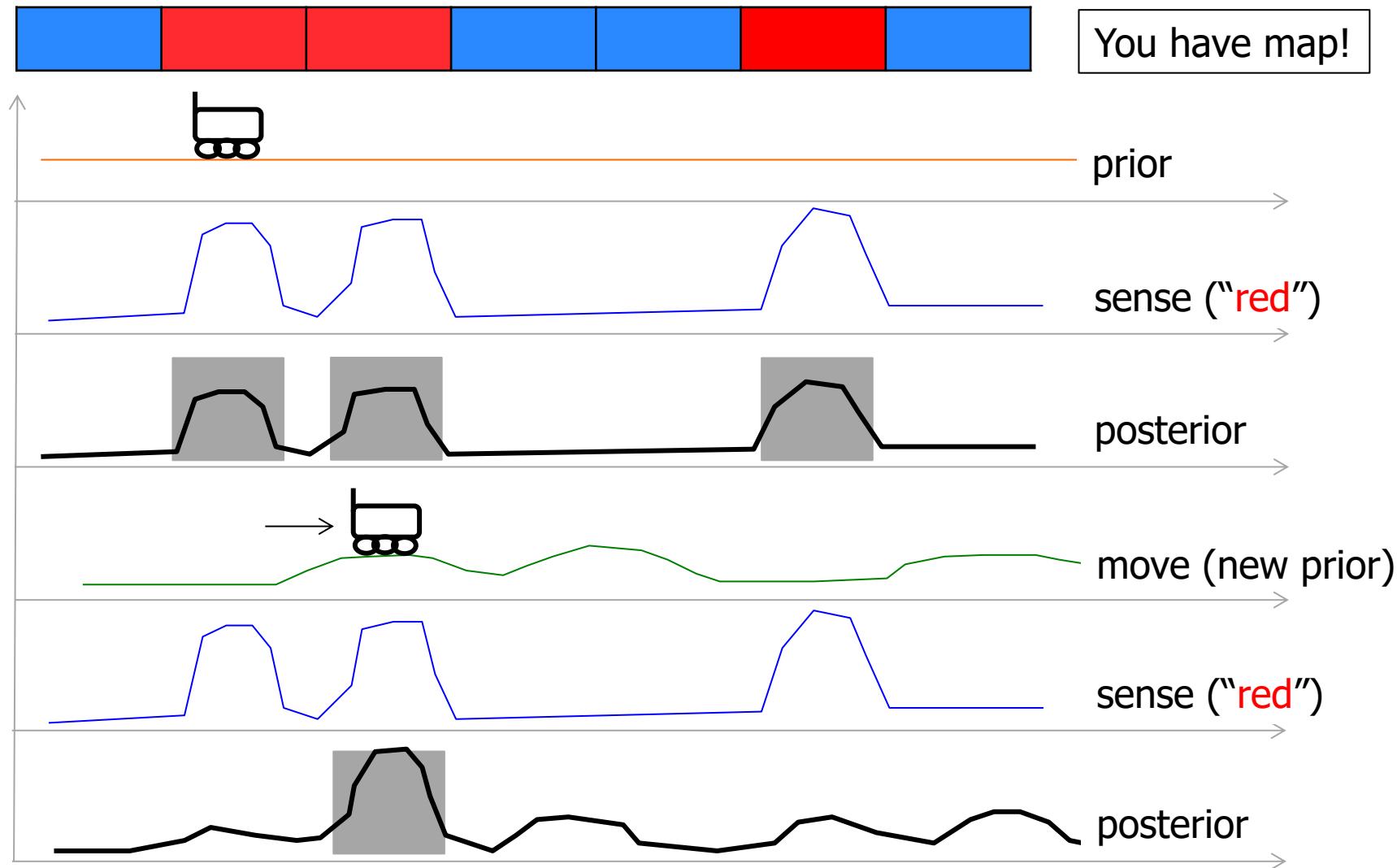
Paper: "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE Trans. Signal Processing, 2002.

# Bayesian and particle filters are \*very\* popular and useful

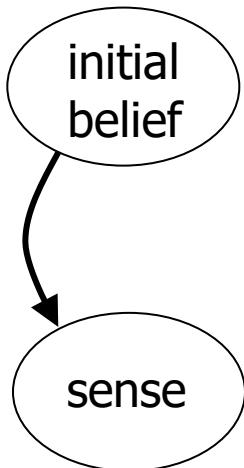
- Job interviews at companies and universities
- Self driving cars
  - In deserts: DARPA challenge winner
  - In cities: GPS is not accurate enough
- Particle filters are a general tool (1990's)
  - The problem: Tracking the state of a system as it evolves over time
  - We have: Sequentially arriving (noisy or ambiguous) observations
  - We want to know: Best possible estimate of the hidden variables

# Localization, the discrete case **BAYESIAN INFERENCE**

# Bayesian Filter in one slide



# First let's look at the sensing part



current pdf  
**(posterior)**

perception model  
**(sense)**

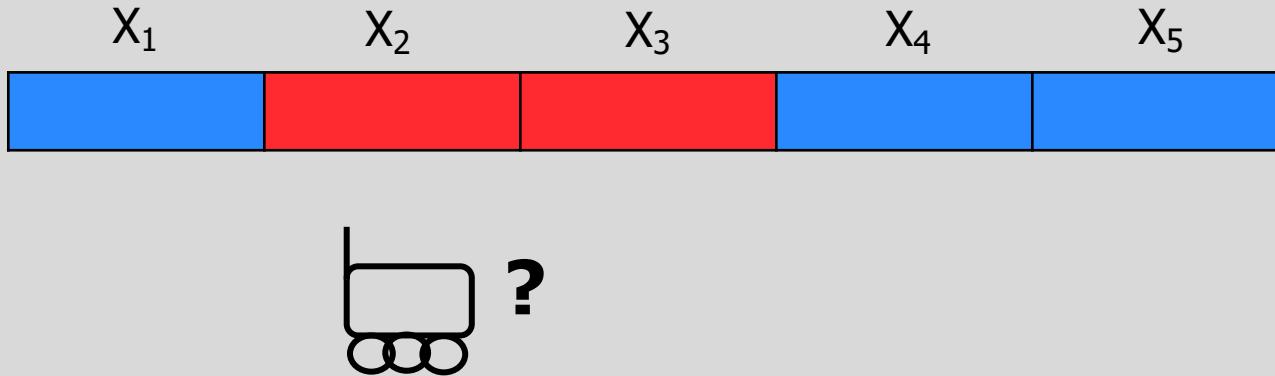
pdf from last time step  
**(prior)**

$$p(X_k \mid \mathcal{Z}_{1:k}) = \frac{p(\mathcal{Z}_k \mid X_k) p(X_k \mid \mathcal{Z}_{1:k-1})}{p(\mathcal{Z}_k \mid \mathcal{Z}_{1:k-1})}$$

normalization

# Initial belief

a robot *has a map* of the grid below, but it doesn't know where it is, what is its initial belief?

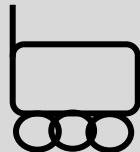


# Sense

*perception model*

$$\begin{array}{ll} p(z=\text{"red"} | X_i \text{ is "red"}) = 0.6 & p(z=\text{"blue"} | X_i \text{ is "blue"}) = 0.8 \\ p(z=\text{"blue"} | X_i \text{ is "red"}) = 0.4 & p(z=\text{"red"} | X_i \text{ is "blue"}) = 0.2 \end{array}$$

X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>
<b>0.2</b>	<b>0.2</b>	<b>0.2</b>	<b>0.2</b>	<b>0.2</b>



Exercise: 'X<sub>i</sub>' represents cell i.  
'z' is the sensing measurement

Question: the observation z is "red",  
what is the new belief?

# Sense

*perception model*

$$\begin{array}{ll} p(z=\text{"red"} \mid X_i \text{ is "red"}) = 0.6 & p(z=\text{"blue"} \mid X_i \text{ is "blue"}) = 0.8 \\ p(z=\text{"blue"} \mid X_i \text{ is "red"}) = 0.4 & p(z=\text{"red"} \mid X_i \text{ is "blue"}) = 0.2 \end{array}$$

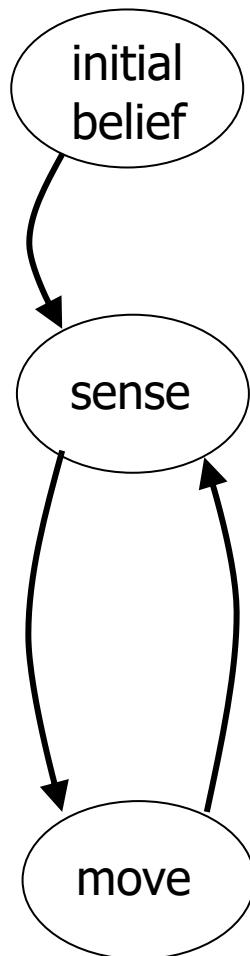
0.2	0.2	0.2	0.2	0.2	p(x) [prior]
0.2	0.6	0.6	0.2	0.2	$p(z x)$
0.04	0.12	0.12	0.04	0.04	$p(z x)*p(x)$
0.04/0.36	0.12/0.36	0.12/0.36	0.04/0.36	0.04/0.36	normalize
<b>1/9</b>	<b>1/3</b>	<b>1/3</b>	<b>1/9</b>	<b>1/9</b>	<b>p(x) [posterior]</b>

current pdf  
**(posterior)**      perception model  
**(sense)**      pdf from last time step  
**(prior)**

$$p(X_k \mid \mathcal{Z}_{1:k}) = \frac{p(\mathcal{Z}_k \mid X_k) p(X_k \mid \mathcal{Z}_{1:k-1})}{p(\mathcal{Z}_k \mid \mathcal{Z}_{1:k-1})}$$

normalization

# Now let's look at the movement part



current pdf  
**(posterior)**

$$p(X_i^t | X_j^{t-1}) = \sum p(X_i | X_j) \text{ motion model (move)} p(X_j^{t-1}) \text{ pdf from last time step (prior)}$$

# Movement Model

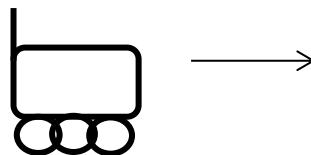
*movement model*

$$p(X_i \mid X_i) = 0.1$$

$$p(X_{i+1} \mid X_i) = 0.8$$

$$p(X_{i+2} \mid X_i) = 0.1$$

X1	X2	X3	X4	X5
0	0.5	0.5	0	0



Exercise: the robot is moving right, what is the new distribution?

# Movement Model

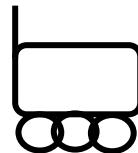
*movement model*

$$p(X_i | X_i) = 0.1$$

$$p(X_{i+1} | X_i) = 0.8$$

$$p(X_{i+2} | X_i) = 0.1$$

X1	X2	X3	X4	X5
0	0.5	0.5	0	0
0	0.05	0.45	0.45	0.05



When using RF, you don't want to see people teletransporting in your localization app. Movement models (any model) help a ton!