DD2410

Lecture slides

Localization

Remark:

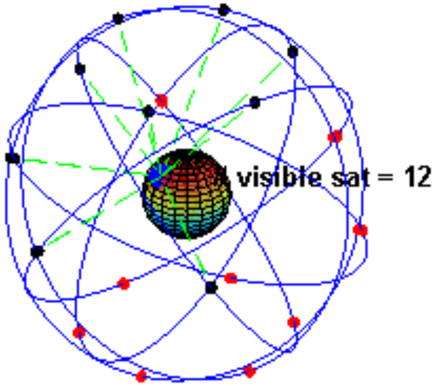
I will introduce concepts and give examples, but not details.

I link to material in Canvas for some more details.

Global Navigation Satellite System (GNSS)

Global Navigation Satellite System (GNSS)

- GPS the most well-known implementation
- Accuracy around 2-4m
- RTK GPS can improve accuracy down to cm

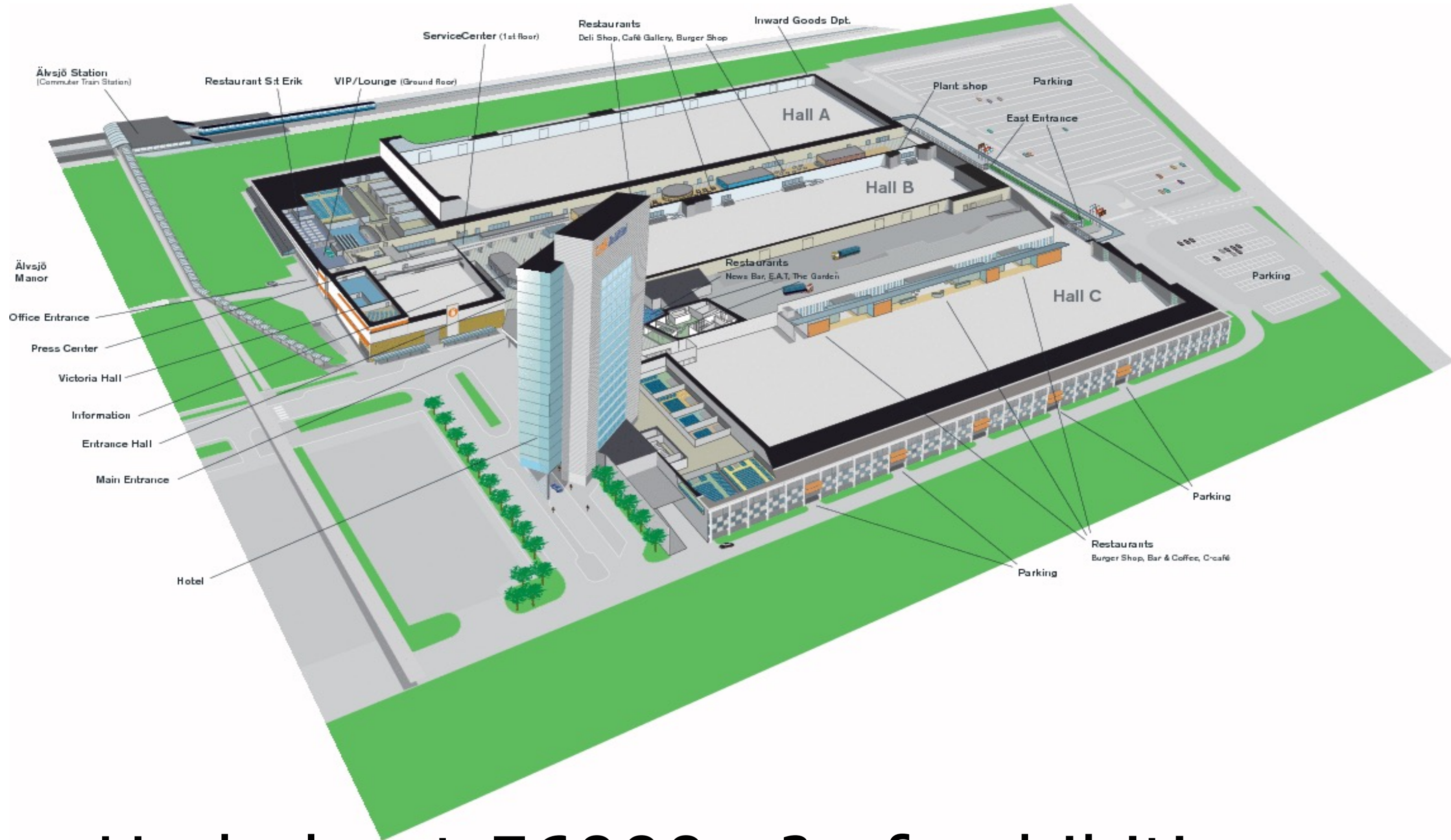


So why do we need anything else???

So why do we need anything else???

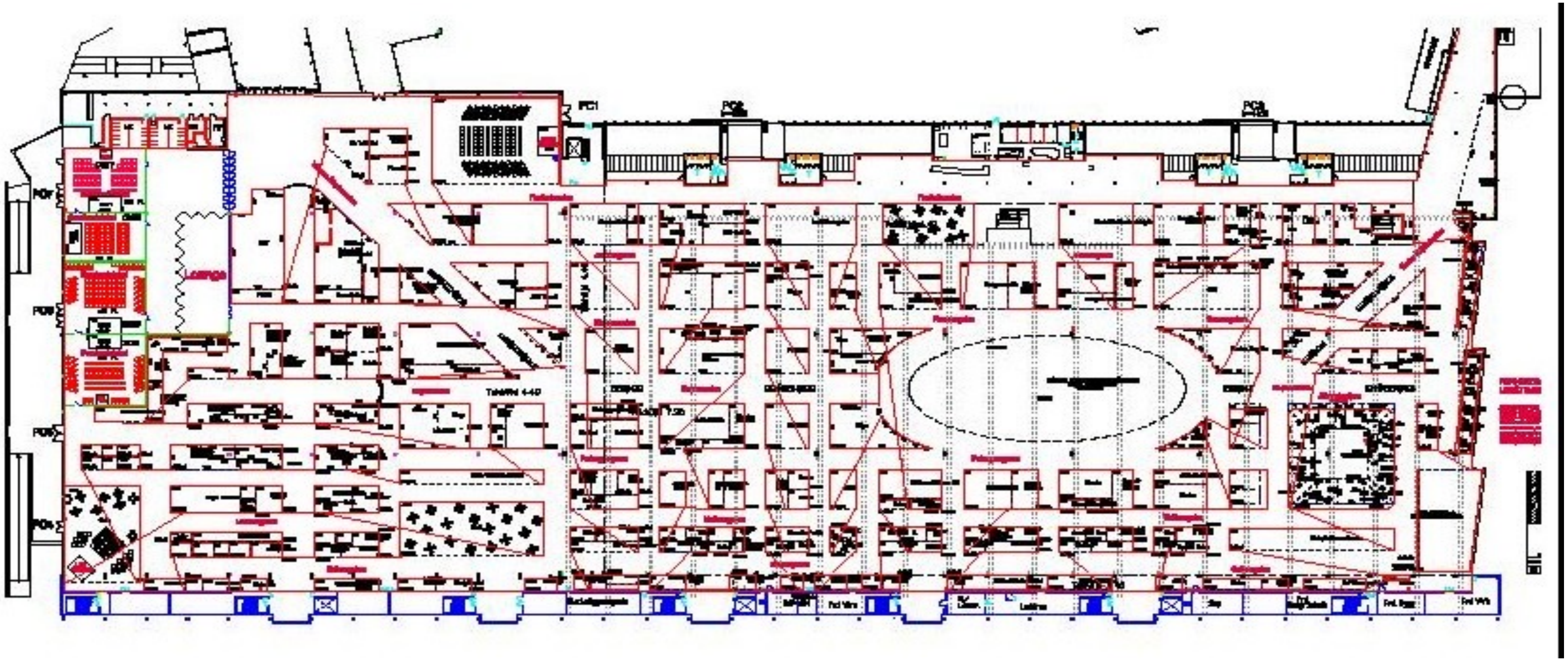
- At least 4 satellites need to be in line of sight.
  → Indoor,tunnels, etc GPS-denied
- Limited accuracy or more expensive with RTK
- The update rate is relatively limited (a few Hz).

Stockholm International Fairs



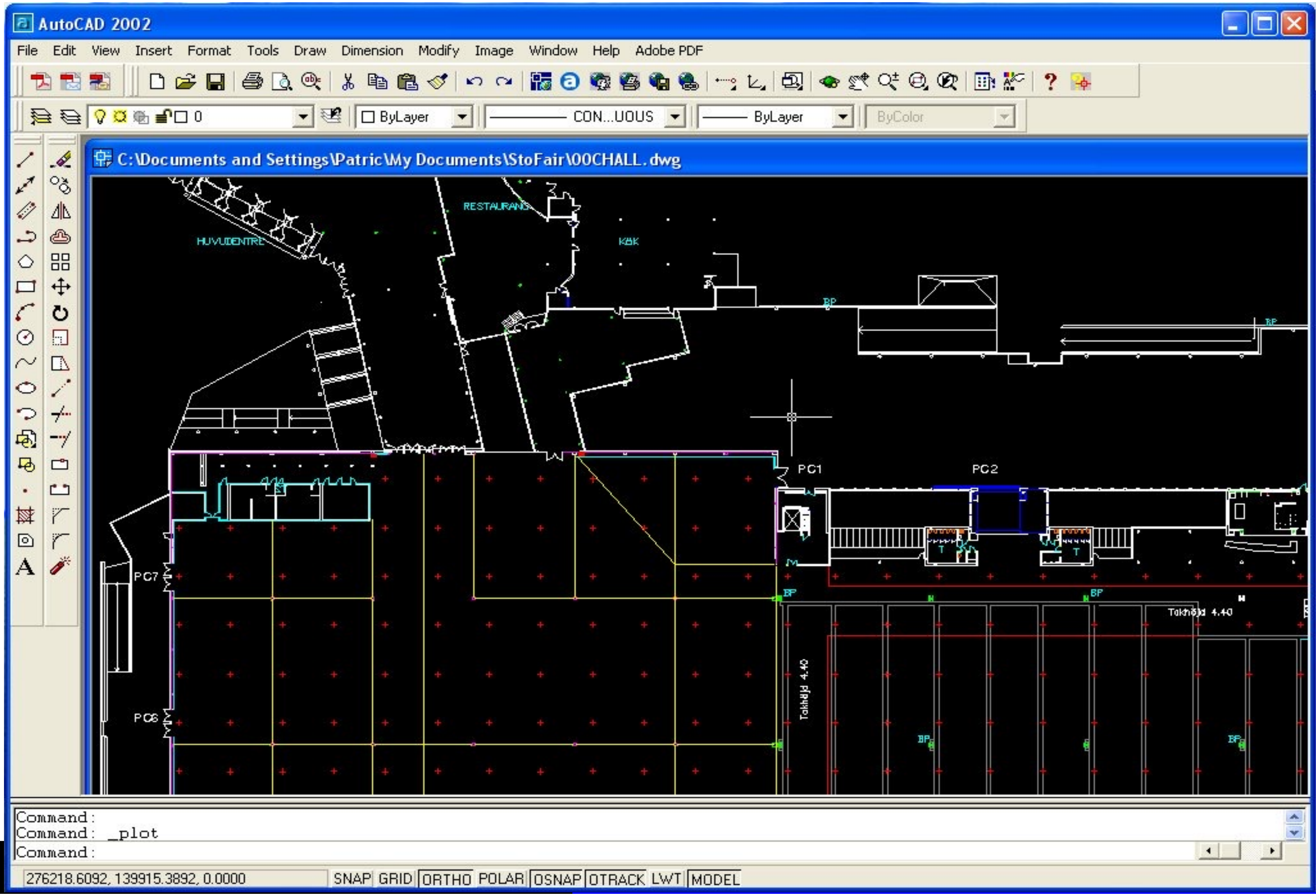
- Had about 56000m² of exhibition space
- How to automate the process of marking stands on the floor?

Example fair layout

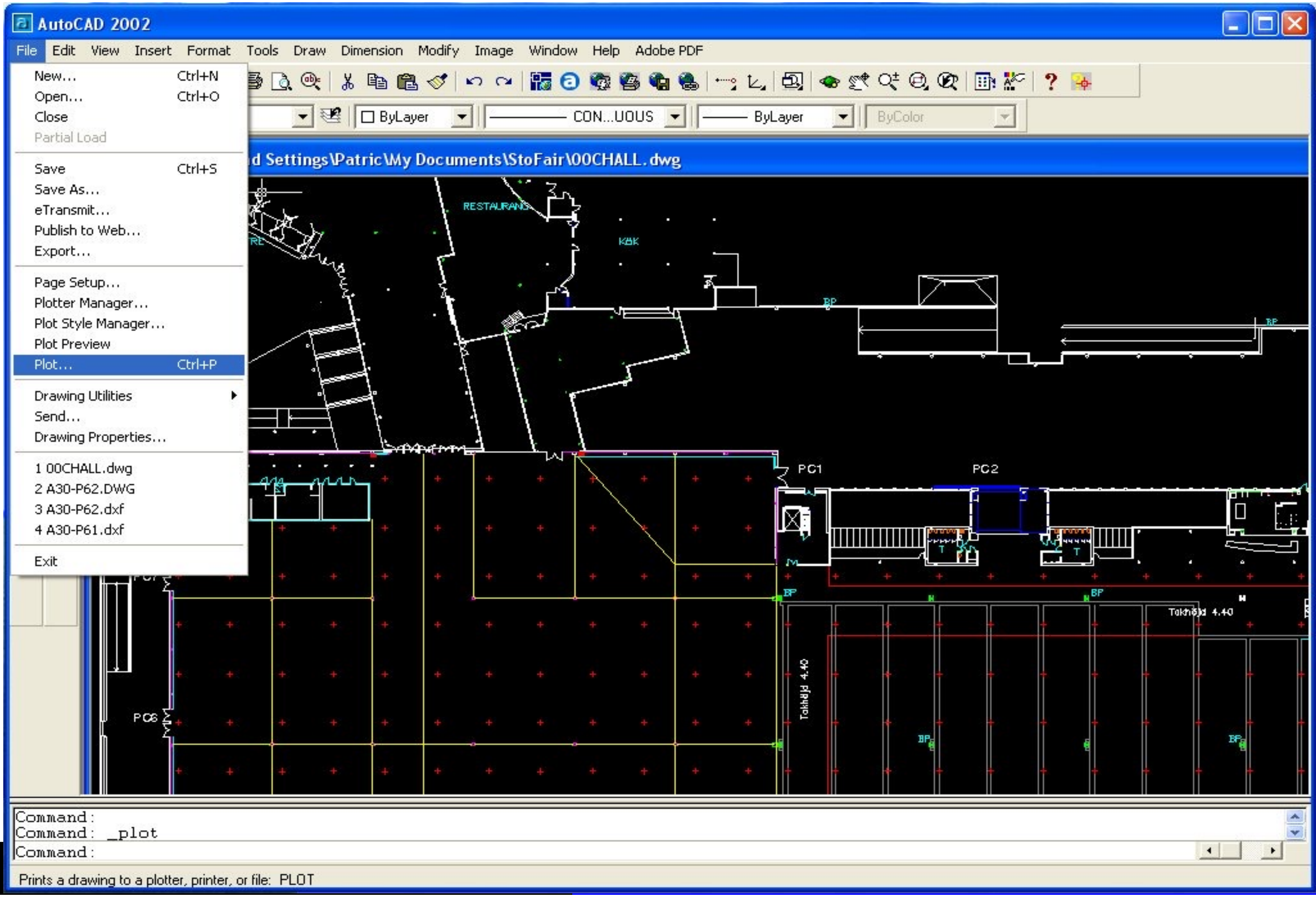


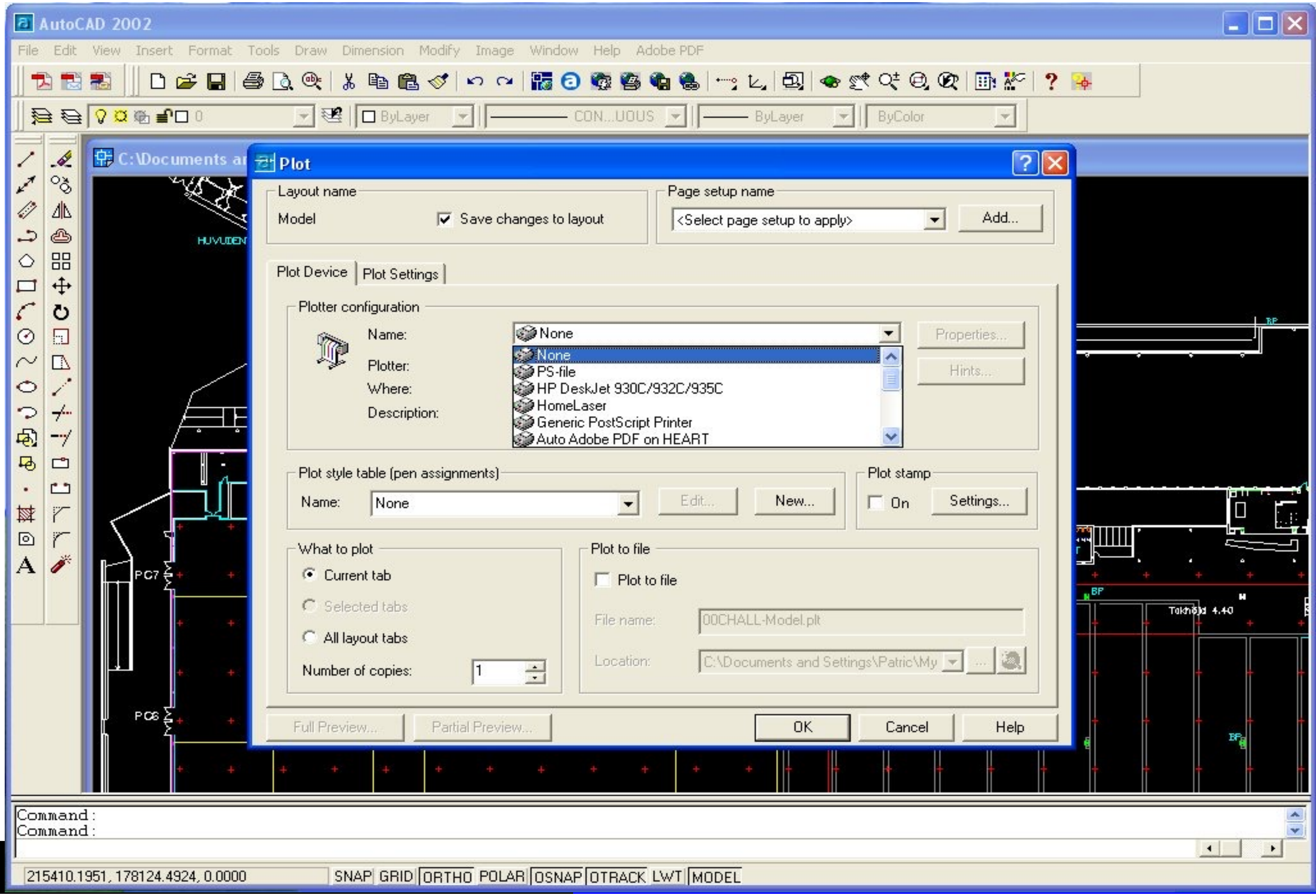
- Thousands of points to mark
- Very tedious job
- Time is money → want short time between fairs

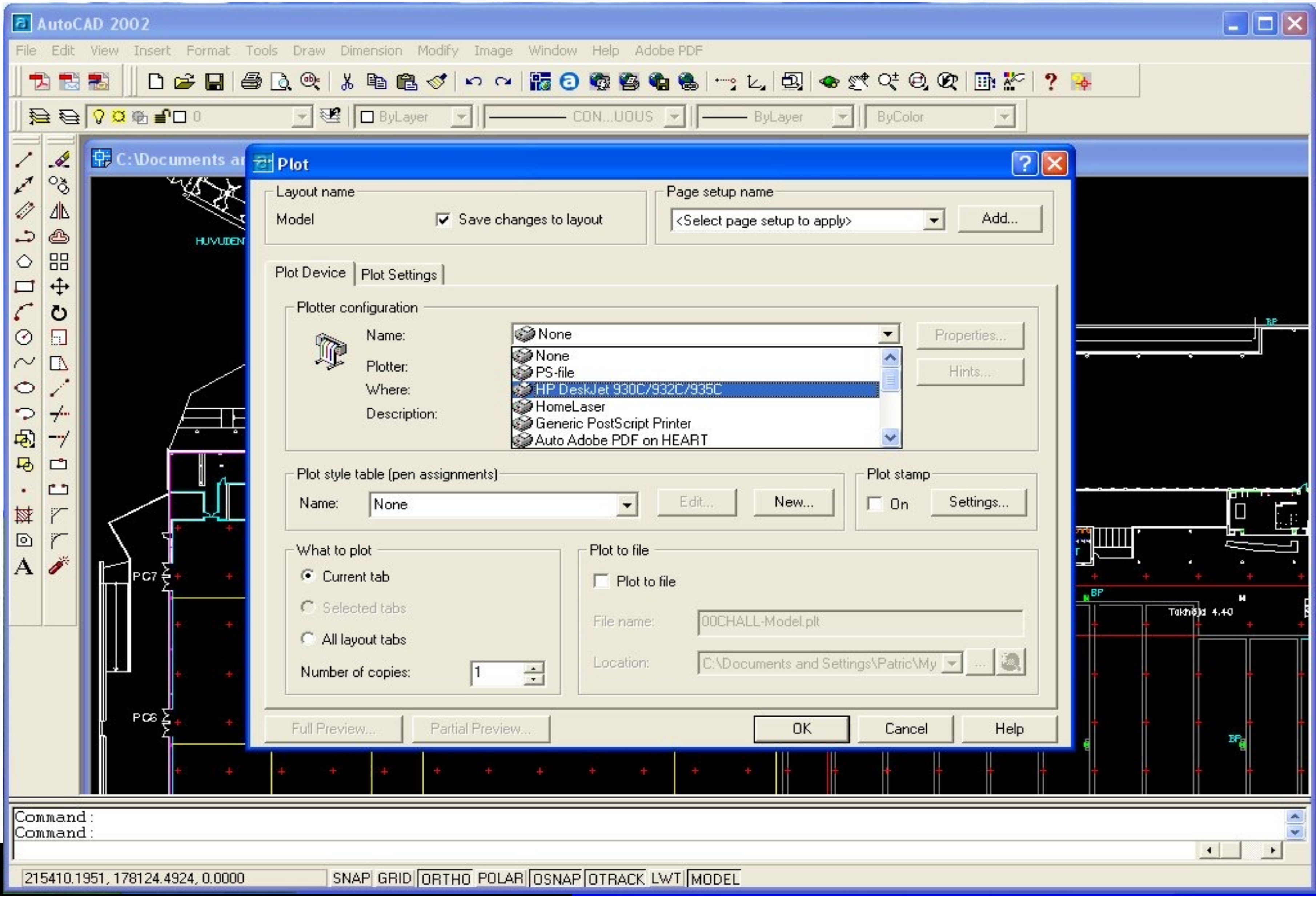
The Vision

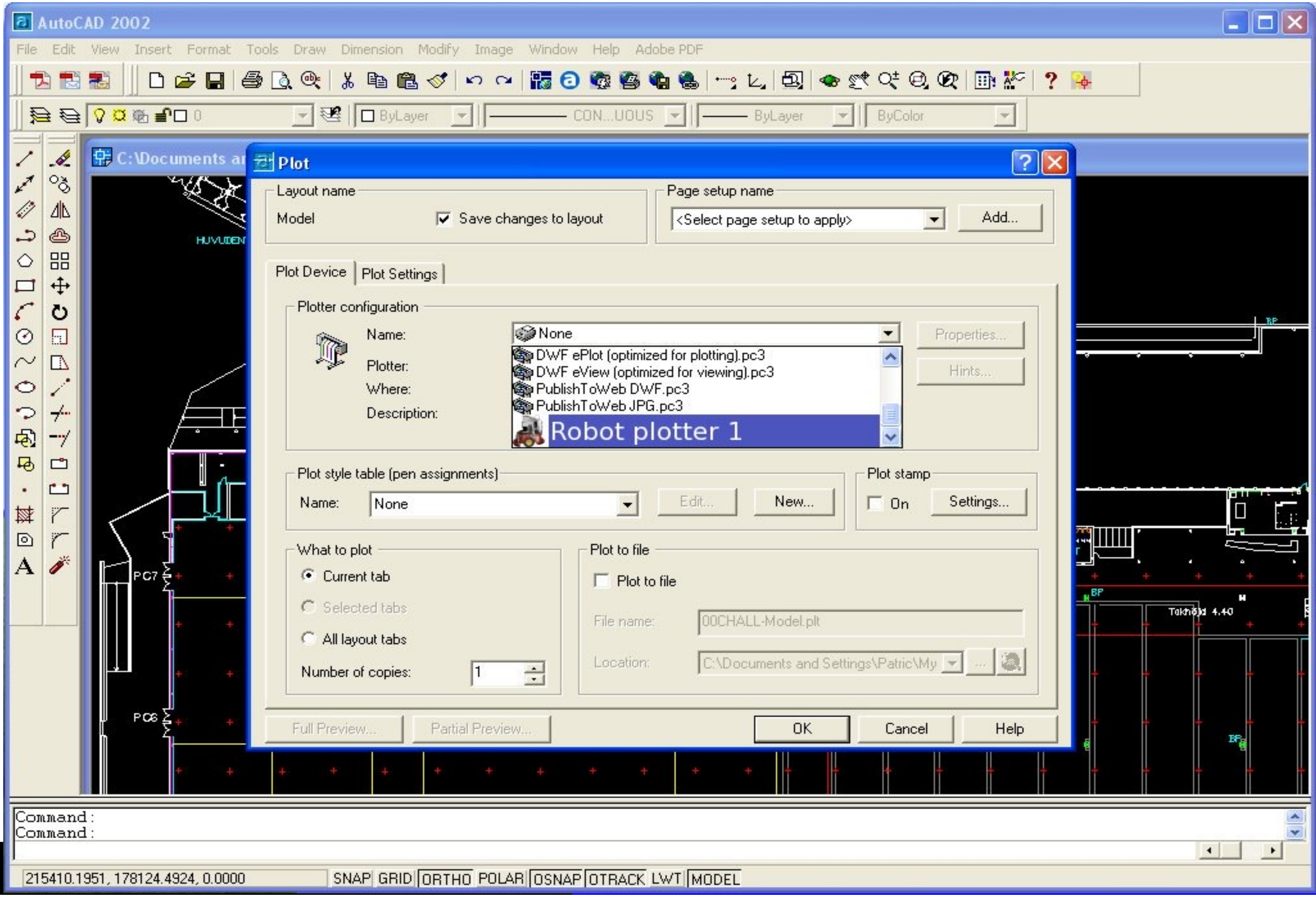


AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
ByLayer
CON...UOUS
ByLayer
ByColor
C:\Documents and Settings\Patric\My Documents\StoFair\00CHALL.dwg
RESTAURANG
HUVUDENTRE
KÖK
PC1
PC2
PC7
PC6
Command :
Command : _plot
Command :
276218.6092, 139915.3892, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

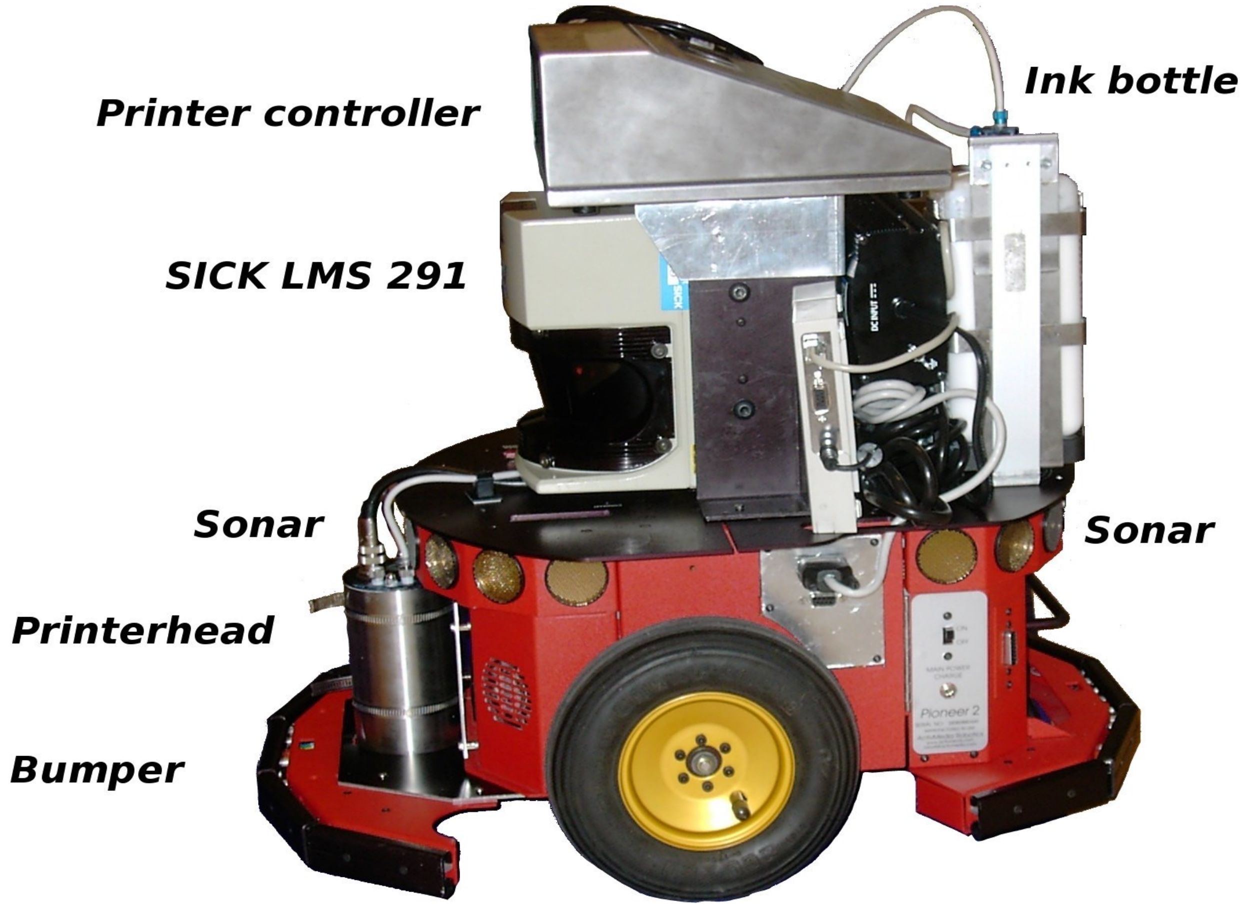

AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
New... Ctrl+N
Open... Ctrl+O
Close
Partial Load
Save Ctrl+S
Save As...
eTransmit...
Publish to Web...
Export...
Page Setup...
Plotter Manager...
Plot Style Manager...
Plot Preview
Plot... Ctrl+P
Drawing Utilities
Send...
Drawing Properties...
1 00CHALL.dwg
2 A30-P62.DWG
3 A30-P62.dxf
4 A30-P61.dxf
Exit
ByLayer
CON...UOUS
ByLayer
ByColor
d Settings\Patric\My Documents\StoFair\00CHALL.dwg
PC1
PC2
PC6
Command :
Command : _plot
Command :
Prints a drawing to a plotter, printer, or file: PLOT


AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
ByLayer
CON...UOUS
ByLayer
ByColor
C:\Documents ar
Plot
Layout name
Model
Save changes to layout
Page setup name
<Select page setup to apply>
Add...
Plot Device
Plot Settings
Plotter configuration
Name:
None
Properties...
Plotter:
Where:
Description:
Hints...
None
PS-file
HP DeskJet 930C/932C/935C
HomeLaser
Generic PostScript Printer
Auto Adobe PDF on HEART
Plot style table (pen assignments)
Name:
None
Edit...
New...
Plot stamp
On
Settings...
What to plot
Current tab
Selected tabs
All layout tabs
Number of copies:
1
Plot to file
Plot to file
File name:
00CHALL-Model.plt
Location:
C:\Documents and Settings\Patric\My
Full Preview...
Partial Preview...
OK
Cancel
Help
Command :
Command :
215410.1951, 178124.4924, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL


AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
Plot
Layout name
Model
Save changes to layout
Page setup name
<Select page setup to apply>
Add...
Plot Device
Plot Settings
Plotter configuration
Name:
None
Plotter:
Where:
Description:
None
PS-file
HP DeskJet 930C/932C/935C
HomeLaser
Generic PostScript Printer
Auto Adobe PDF on HEART
Properties...
Hints...
Plot style table (pen assignments)
Name:
None
Edit...
New...
Plot stamp
On
Settings...
What to plot
Current tab
Selected tabs
All layout tabs
Number of copies:
1
Plot to file
Plot to file
File name:
00CHALL-Model.plt
Location:
C:\Documents and Settings\Patric\My
Full Preview...
Partial Preview...
OK
Cancel
Help
Command :
Command :
215410.1951, 178124.4924, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

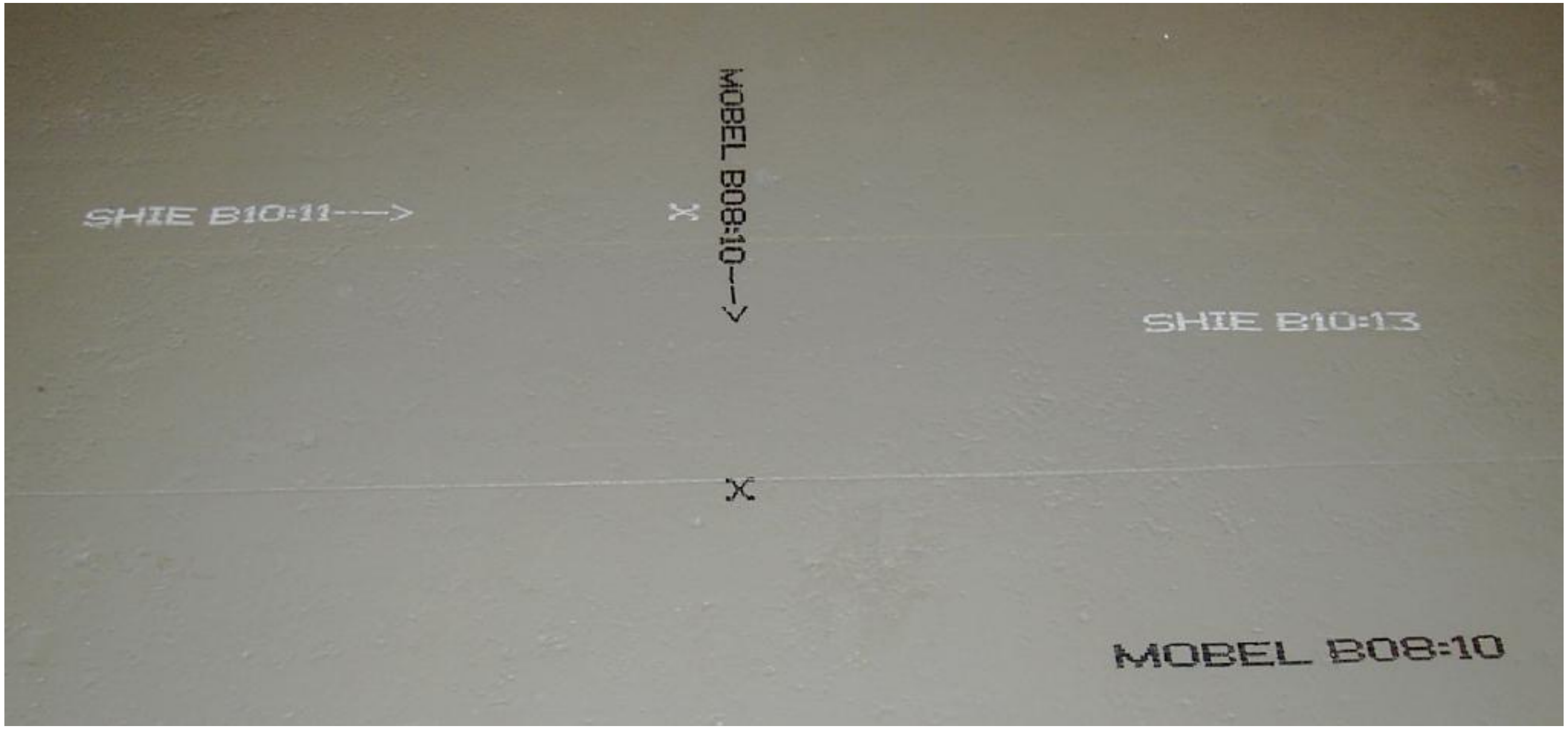

AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
Plot
Layout name
Model
Save changes to layout
Page setup name
<Select page setup to apply>
Add...
Plot Device
Plot Settings
Plotter configuration
Name:
Plotter:
Where:
Description:
None
DWF ePlot (optimized for plotting).pc3
DWF eView (optimized for viewing).pc3
PublishToWeb DWF.pc3
PublishToWeb JPG.pc3
Robot plotter 1
Properties...
Hints...
Plot style table (pen assignments)
Name:
None
Edit...
New...
Plot stamp
On
Settings...
What to plot
Current tab
Selected tabs
All layout tabs
Number of copies:
1
Plot to file
Plot to file
File name:
00CHALL-Model.plt
Location:
C:\Documents and Settings\Patric\My
Full Preview...
Partial Preview...
OK
Cancel
Help
Command :
Command :
215410.1951, 178124.4924, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

Meet Harry Plotter!



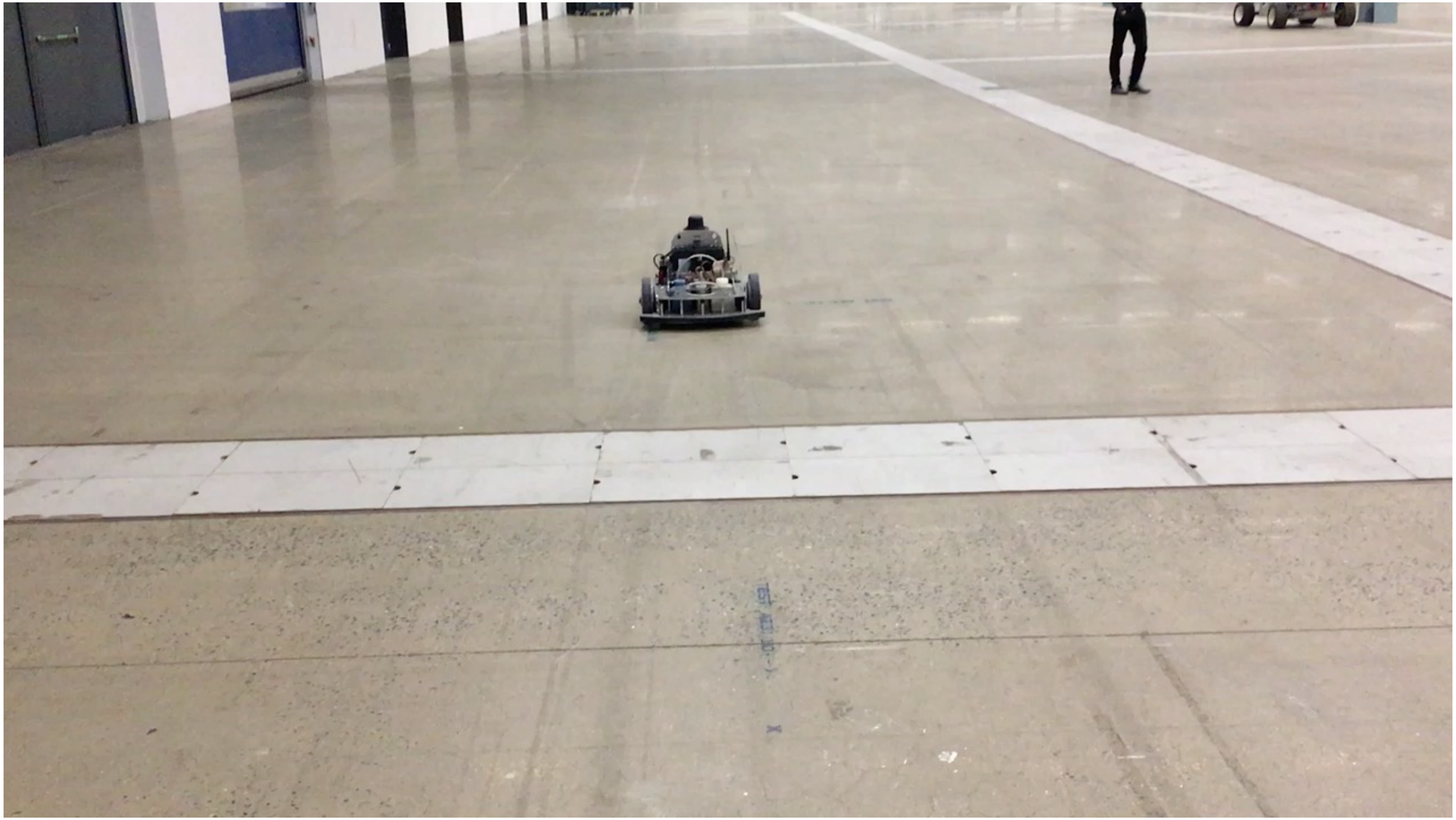
P. Jensfelt, E. Förell and P. Ljunggren,
“Automating the Marking Process for Exhibitions and Fairs”,
Robotics and Autonomous Magazine, 14:3, 2007

Example marks



- Harry got a sister, Hermione
- System in operation since 2003

Current version of the hardware



Localization at work
IJCAI ECAI 2018

Two steps in localization

- Dead reckoning – **prediction step**
- Map based **update step**

Dead reckoning

- Use relative measurements to estimate how the robot is moving

Dead reckoning

- Use relative measurements to estimate how the robot is moving
- Examples
  - Odometry using wheel encoders
  - Motor commands
  - Visual odometry

Dead reckoning

- Use relative measurements to estimate how the robot is moving
- Pros?
- Cons?

Dead reckoning

- Use relative measurements to estimate how the robot is moving

- Pros
  - High frequency and low cost
- Cons
  - Error unbounded and only relative position

Odometry dead-reckoning differential drive

- Odometry
  - $x(k+1) = x(k) + v*dt*\cos(\theta) \quad = x(k) + D*\cos(\theta)$
  - $y(k+1) = y(k) + v*dt*\sin(\theta) \quad = y(k) + D*\sin(\theta)$
  - $\theta(k+1) = \theta(k) + \omega*dt \quad = \theta(k) + \Delta\theta$

$$pose(k+1) = f_1(pose(k),v,\omega,dt) \quad = f_2(pose(k),D,\Delta\theta)$$

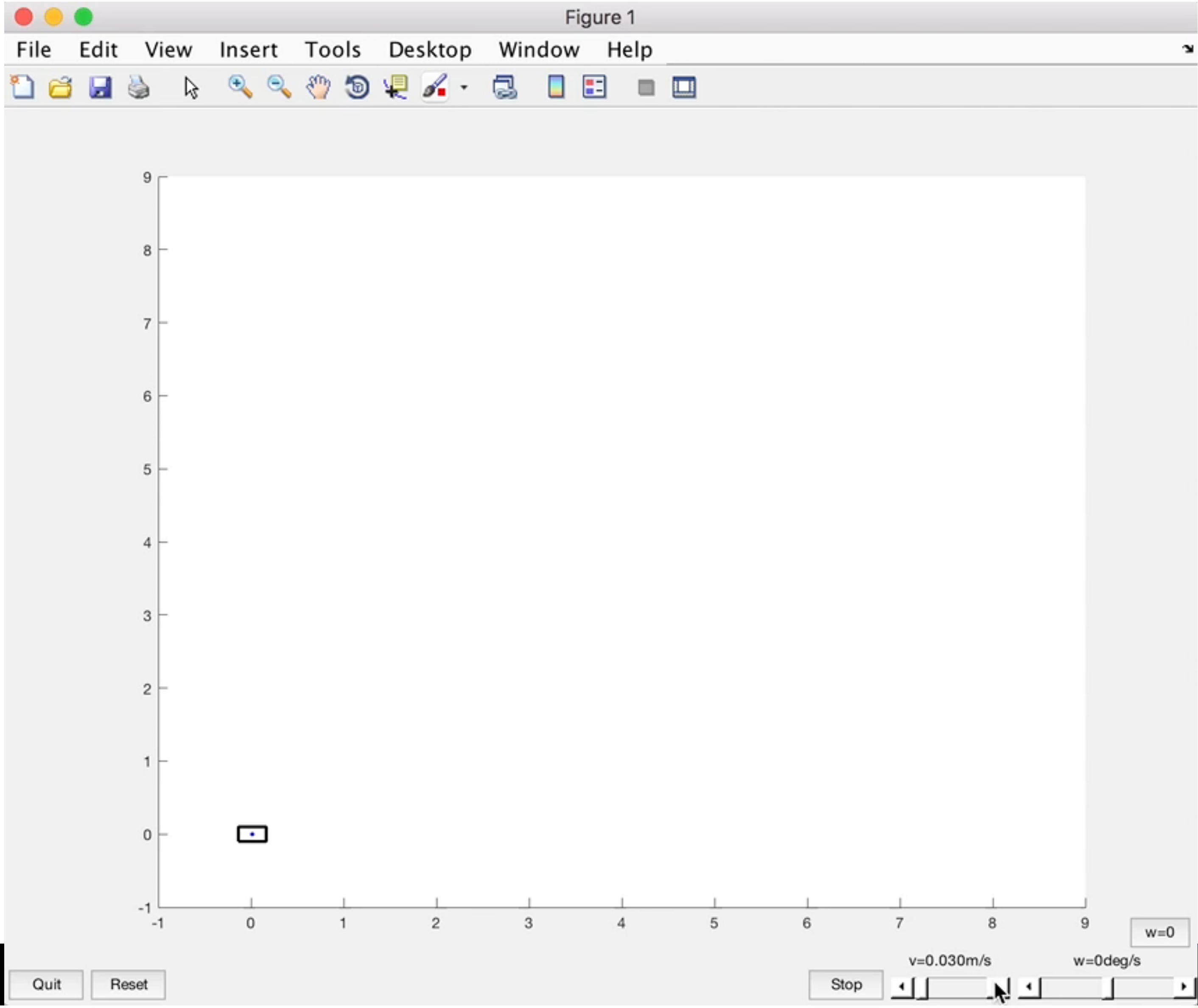
Odometry dead-reckoning differential drive with noise

- Odometry with noise (one possible model)
  - $x(k+1) = x(k) + (v*dt + \vartheta_D)*\cos(\theta)$
  - $y(k+1) = y(k) + (v*dt + \vartheta_D) *\sin(\theta)$
  - $\theta(k+1) = \theta(k) + (\omega*dt + \vartheta_{\theta,\omega}) + \vartheta_{\theta,v}$

  - Where ϑ_D, $\vartheta_{\theta,v}$ and $\vartheta_{\theta,\omega}$ are typically assumed to be zero-mean Gaussian i.e. $N(0,\sigma^2)$

  - Integrating the noise leads to drift!

Visualization of drift in odometry

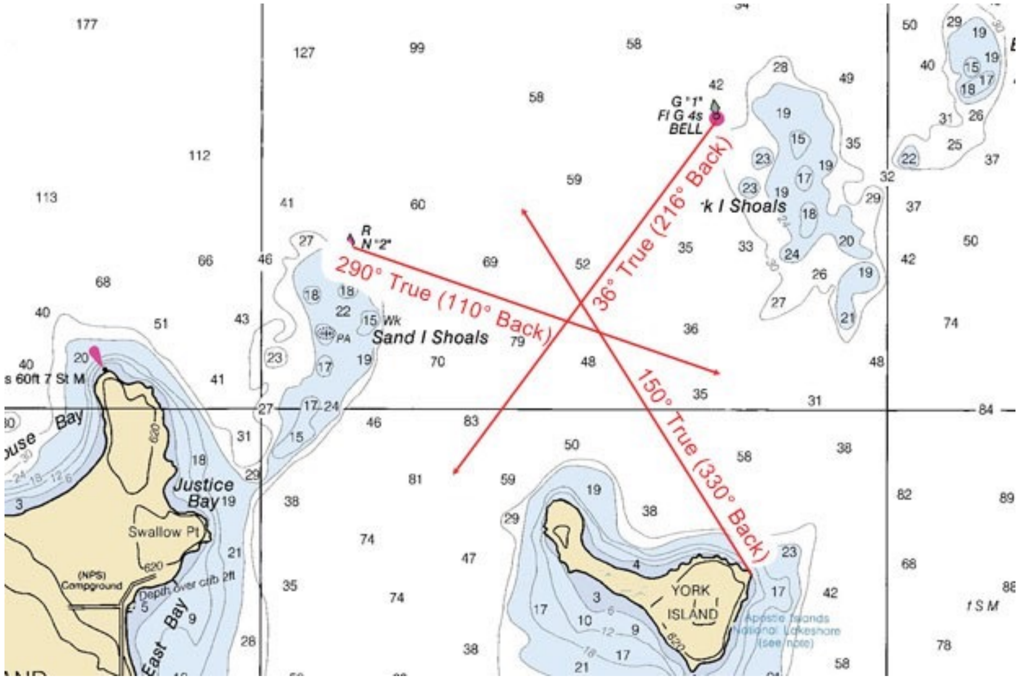


Map based update

- Measure distance, bearing, etc to “objects” with known locations

Map based update

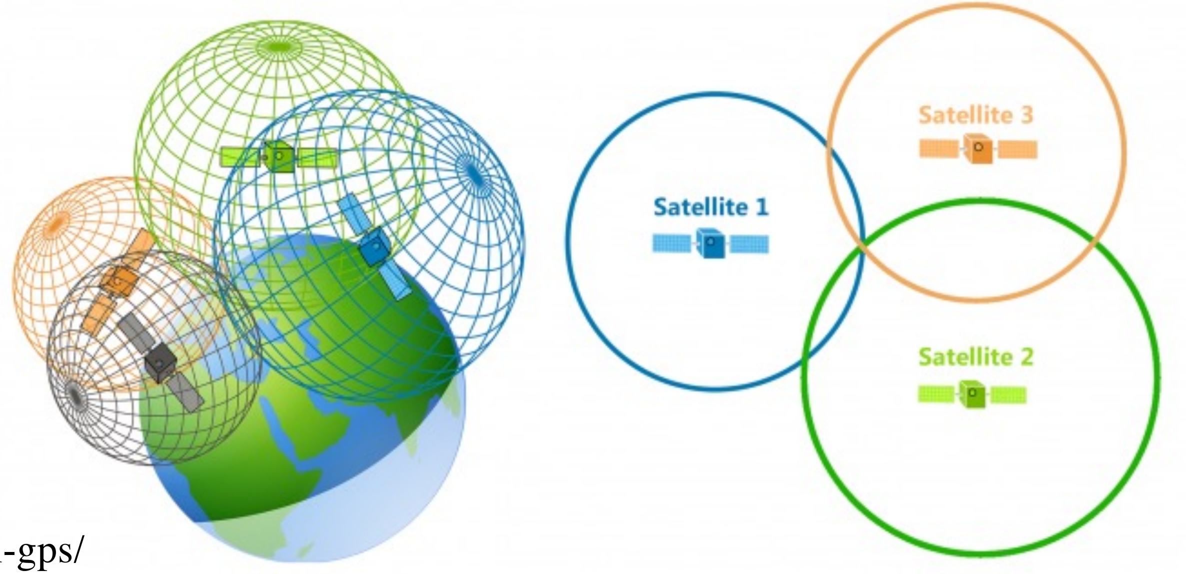
- Measure distance, bearing, etc to “objects” with known locations
- Examples:
  - Triangulation at sea



https://www.paddlinglight.com/articles/navigation-fixes-and-triangulation/

Map based update

- Measure distance, bearing, etc to "objects" with known locations
- Examples:
  - Triangulation at sea
  - Trilateration in GPS system



https://gisgeography.com/trilateration-triangulation-gps/

Map based update

- Measure distance, bearing, etc to “objects” with known locations
- Pros?
- Cons?

Map based update

- Measure distance, bearing, etc to “objects” with known locations

- Pros
  - No drift, position in world frame
- Cons
  - Need to correctly associate measurement with part of map, (typically) lower frequency

Localization

- Two step process
  - Prediction step
    - Dead reckoning estimation
    - Motion model: $x_{k+1} = f(x_k|u_{k+1})$
      $\rightarrow p(x_{k+1} \mid x_k , u_{k+1})$
    - Increases uncertainty

Localization

- Two step process
  - Prediction step
    - Dead reckoning estimation
    - Motion model: $x_{k+1} = f(x_k | u_{k+1})$
      $\rightarrow p(x_{k+1} \mid x_k , u_{k+1})$
    - Increases uncertainty
  - Update step
    - Correct estimate with map based position
    - Measurement model: $z_{k+1} = h(x_{k+1})$
      $\rightarrow p(z_{k+1} \mid x_{k+1})$
    - Decrease uncertainty

Bayesian filter formulation

Prediction based on control input u_k:

$$\mathbf{p(x_{k+1}|Z_k,U_{k+1}) = \int p(x_{k+1}|u_{k+1},x_k)\ p(x_k|Z_k,U_k)\ dx_k}$$

→ distribution smeared out (uncertainty increases)

Update with new measurement z_{k+1}:

$$\mathbf{p(x_{k+1}|Z_{k+1},U_{k+1}) = \eta\ p(z_{k+1}|x_{k+1})p(x_{k+1}|Z_k,U_{k+1})}$$

→distribution more peaked (uncertainty decreases)

Bayesian filter formulation

motion model often given by odometry

Prediction based on control input u_k:

$$\mathbf{p(x_{k+1}|Z_k,U_{k+1}) = \int \boxed{p(x_{k+1}|u_{k+1},x_k)}\ p(x_k|Z_k,U_k)\ dx_k}$$

Update with new measurement z_{k+1}:

$$\mathbf{p(x_{k+1}|Z_{k+1},U_{k+1}) = \eta\ p(z_{k+1}|x_{k+1})p(x_{k+1}|Z_k,U_{k+1})}$$

Bayesian filter formulation

motion model often given by odometry

Prediction based on control input u_k:

$$\mathbf{p(x_{k+1}|Z_k,U_{k+1}) = \int \boxed{p(x_{k+1}|u_{k+1},x_k)}\, p(x_k|Z_k,U_k)\, dx_k}$$

measurement model relating the state x to what you measure z

Update with new measurement z_{k+1}:

$$\mathbf{p(x_{k+1}|Z_{k+1},U_{k+1}) = \eta\, \boxed{p(z_{k+1}|x_{k+1})}\, p(x_{k+1}|Z_k,U_{k+1})}$$

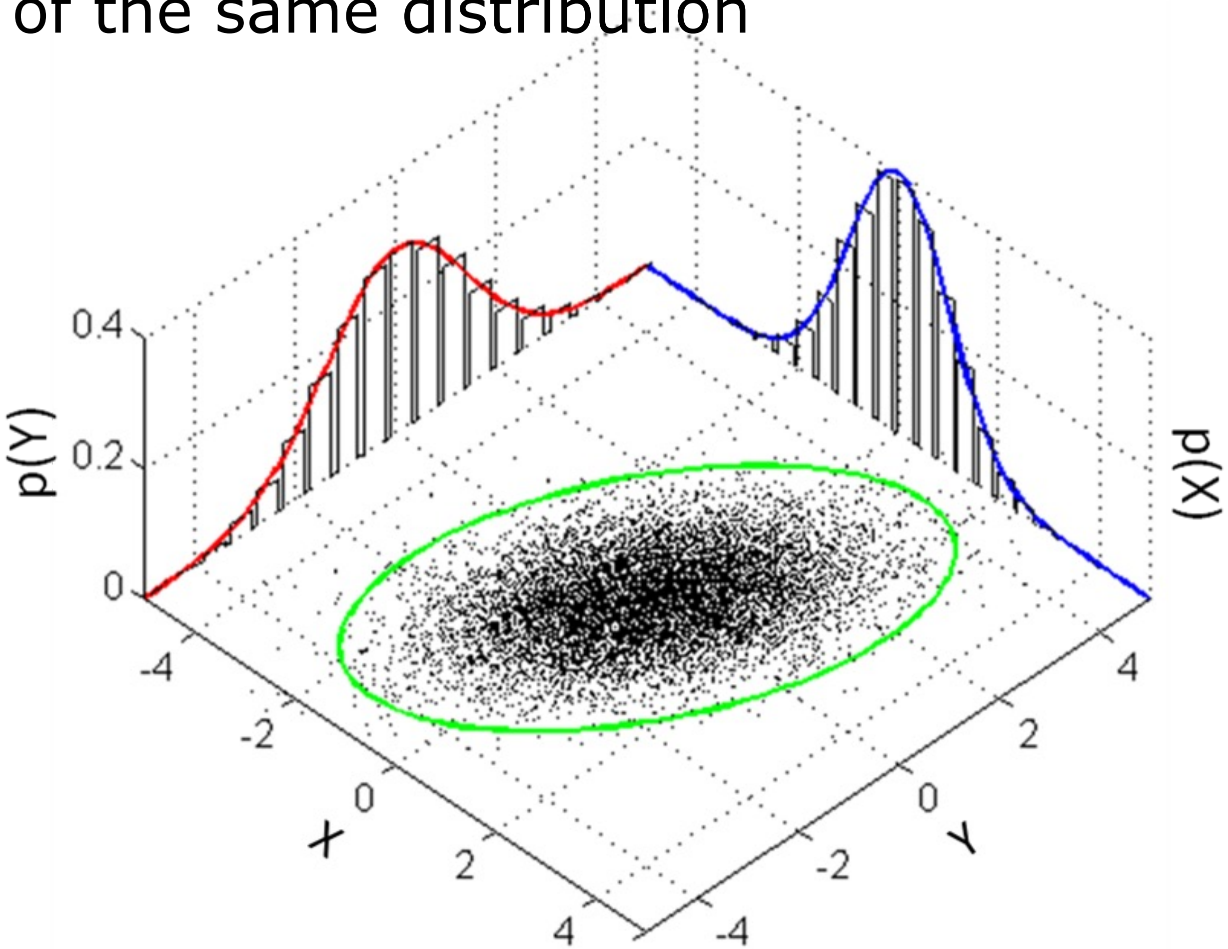
Representing probabilistic information

- What we described so far is general.
- To implement this we need to determine how to represent the distributions, such as $p(x_k|Z_k,U_k)$

- Two most common choices
  - Gaussian distribution
  - Set of particles (sampled based representation)

Gauss vs particle set for 2D example

- Green ellipse: Gaussian
- Black dots: Samples of the same distribution



https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Algorithms for localization

- The two most frequently used algorithms
  - Particle filter ←→ Sample based representation
  - Kalman filter ←→ Gaussian

Particle filter

The particle filter represents probability distributions using a set of particles, p_k, sampled from the distribution.

Each particle represents one “hypothesis” about the state.

Each particle also has a weight, initialized as $\pi=1/N$.

$$p_k=\{x_k, \pi_k\}$$

particle state weight

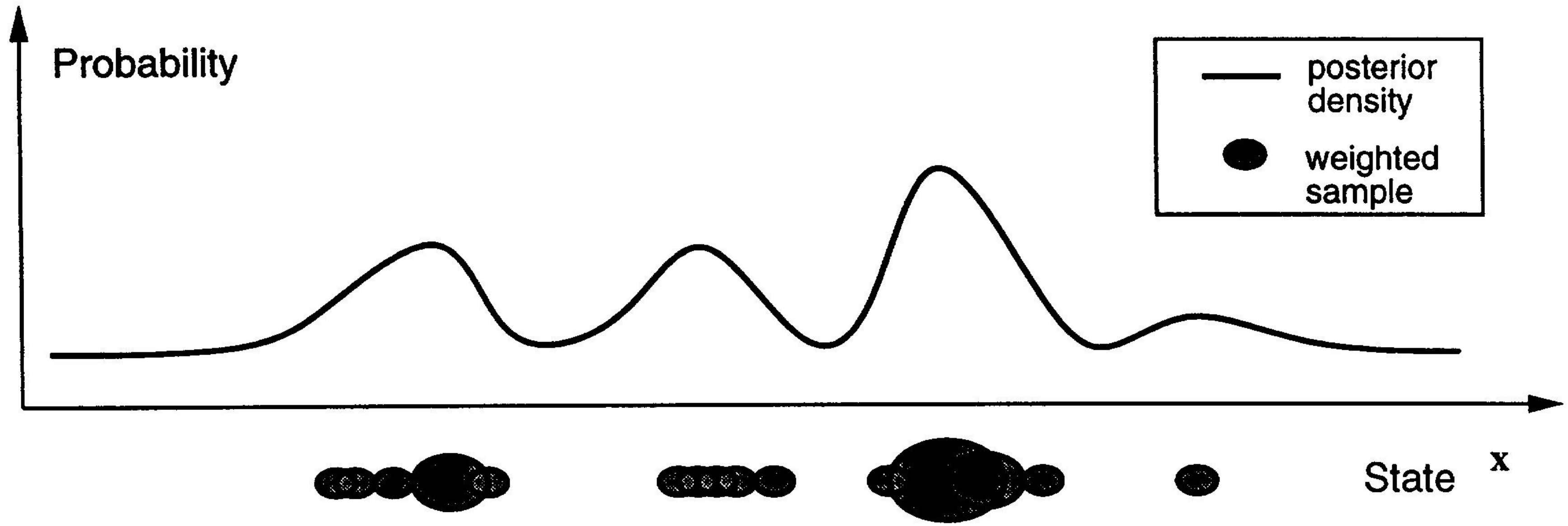
Particle filter

$p_k=\{x_k, \pi_k\}$

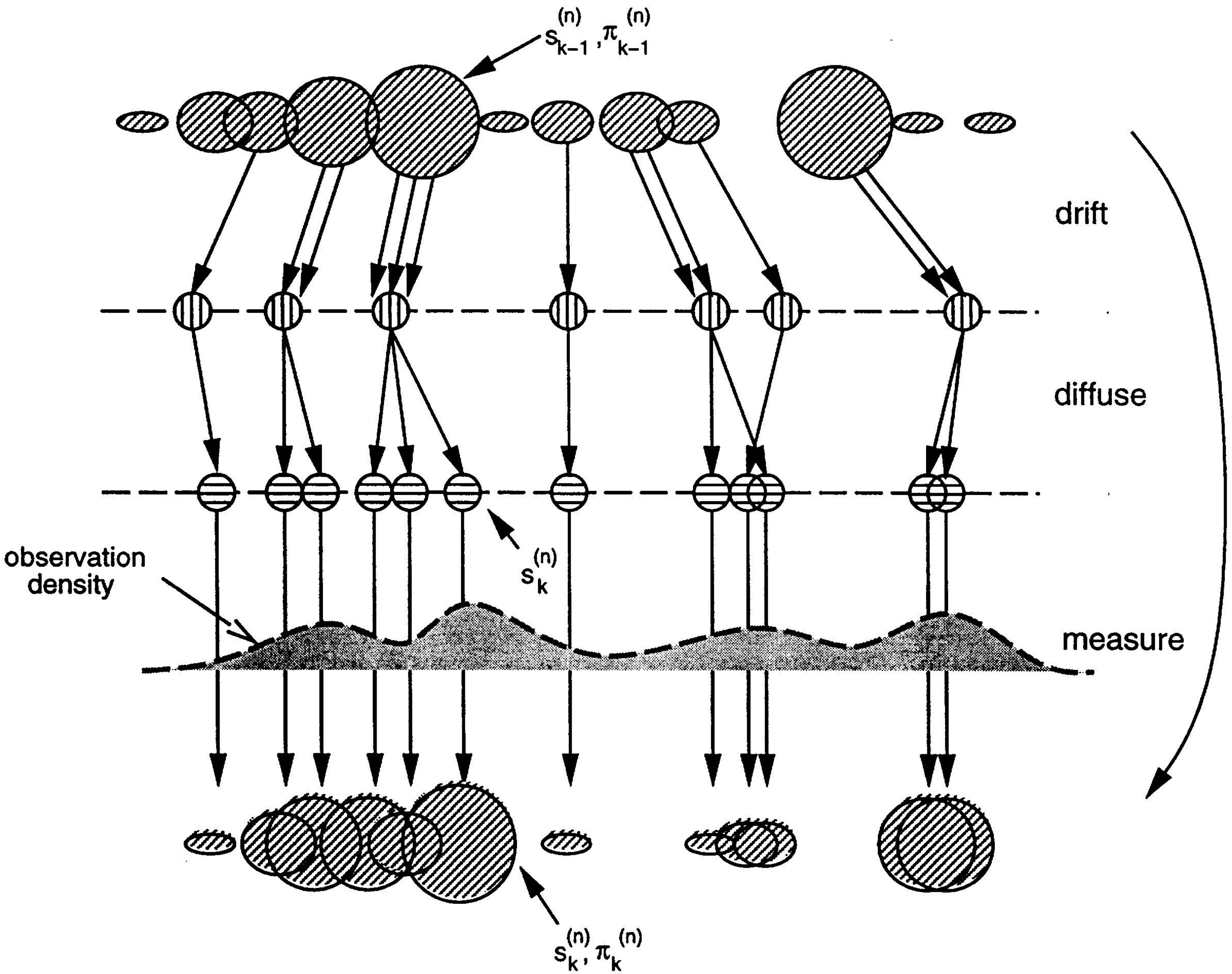
particle state weight

The weight allows us to use fewer particles.

For example, can replace 5 particles by one with 5 times the weight



The particle filter

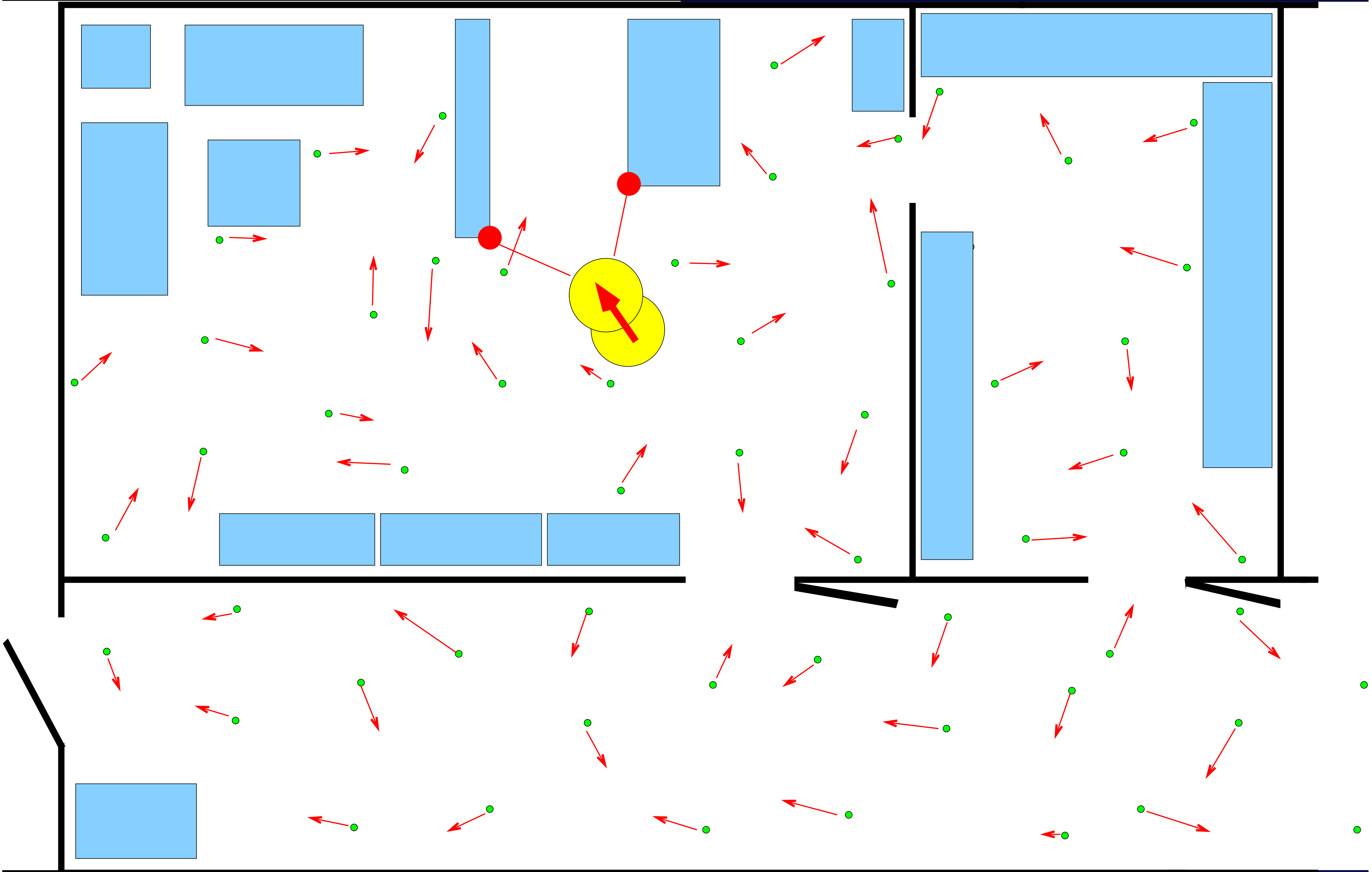


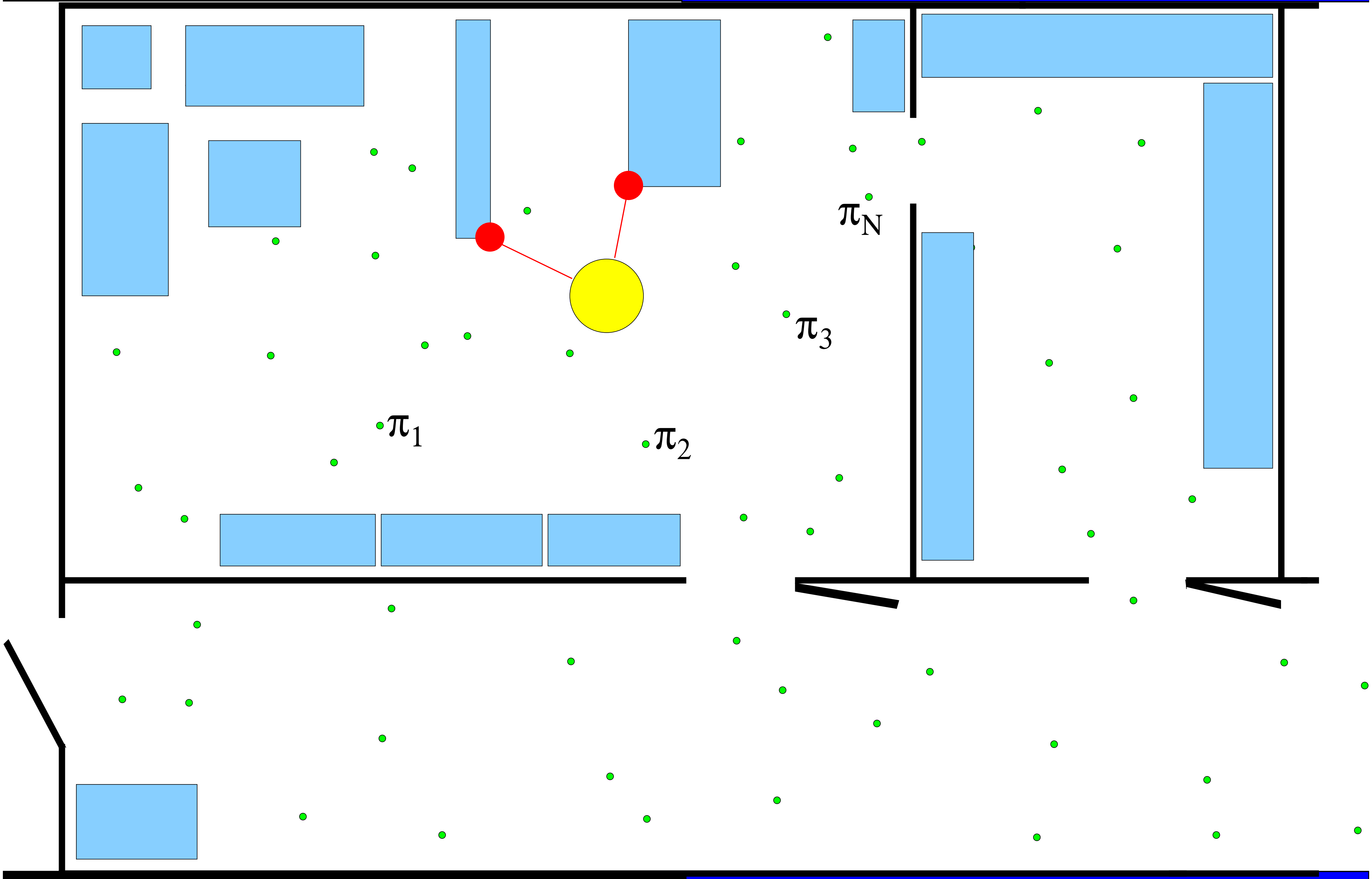
Prediction step

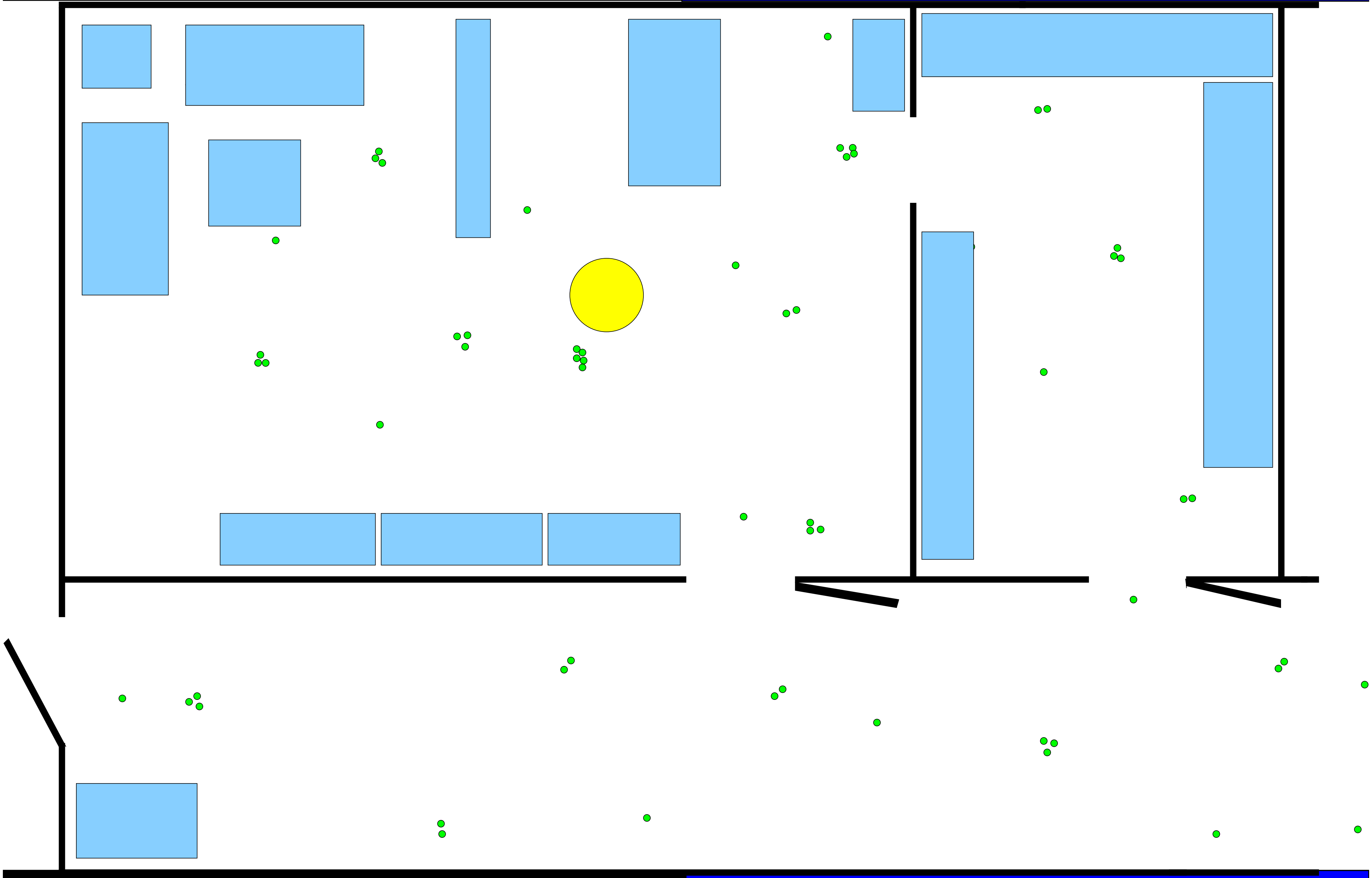
drift = noise free odometry prediction
diffuse = add noise

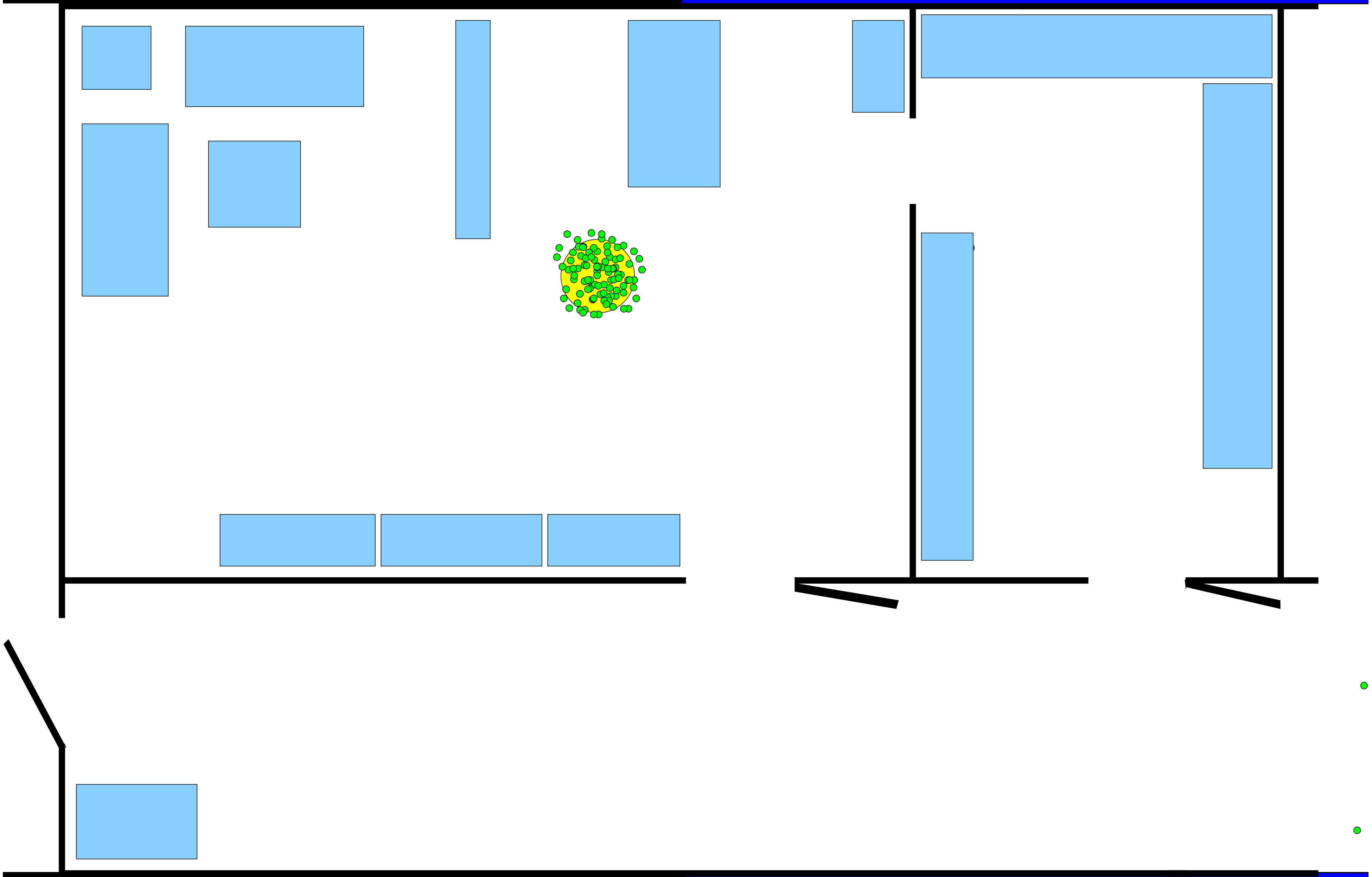
Particle filter algorithm

- 0. Initialize the particles given what you know to start with and with weight 1/N.
  - If you know nothing → start with uniform distribution
  - Know a lot → start with very peaked distribution
- 1. Use odometry to update all poses of particles and perturb each particle according to odomety noise
  - each particle will experience different noise (independent)
- 2. Use measurements and multiply the weight of each particle, i, with $p(z_k|x^i_k)$, ie the likelihood
  - This is the measurement model
- 3. Re-sample "if needed" and then return to 1.




π_N
π_3
π_1
π_2





Play with particle filter

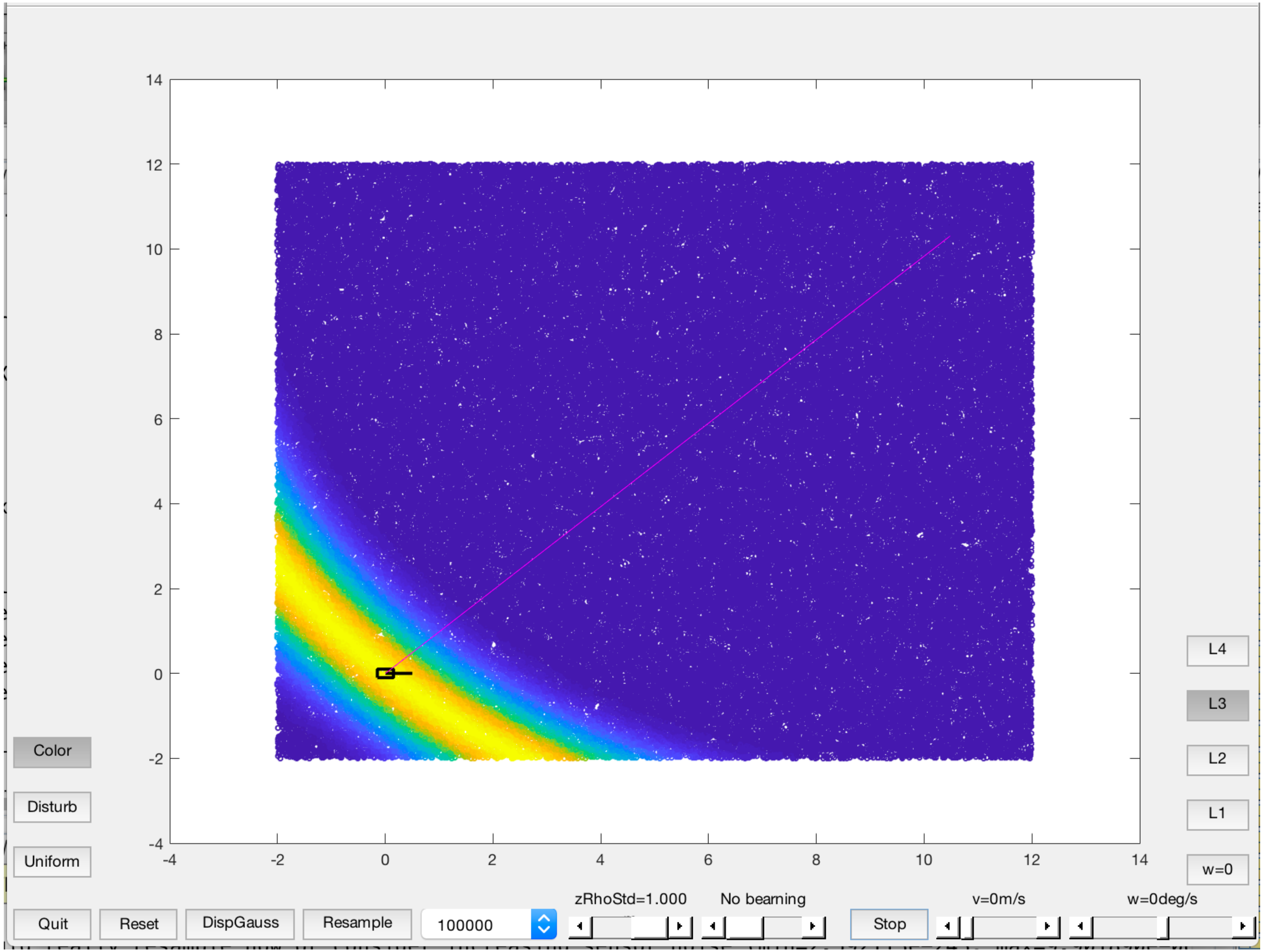
- Prediction (what dictates the spread of particles?)
- Tracking
- Global localization

- Non-Gaussian distributions

- Start from uniform distribution and measure range to point landmark. What does the position distribution look like?
- What if you have two points?
- What if you measure bearing to a point?

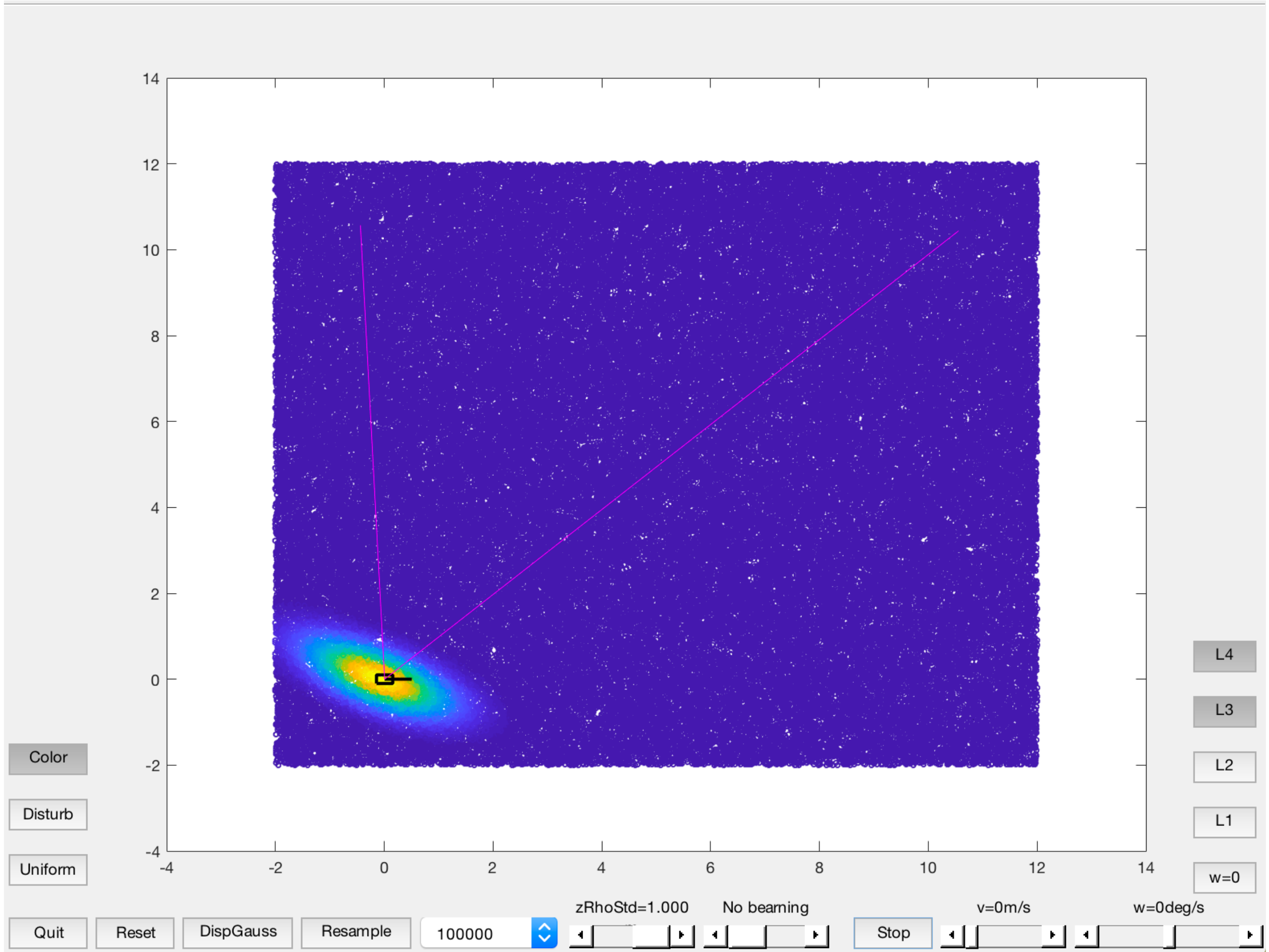
Update with range to singe landmark

- Clearly not Gaussian!



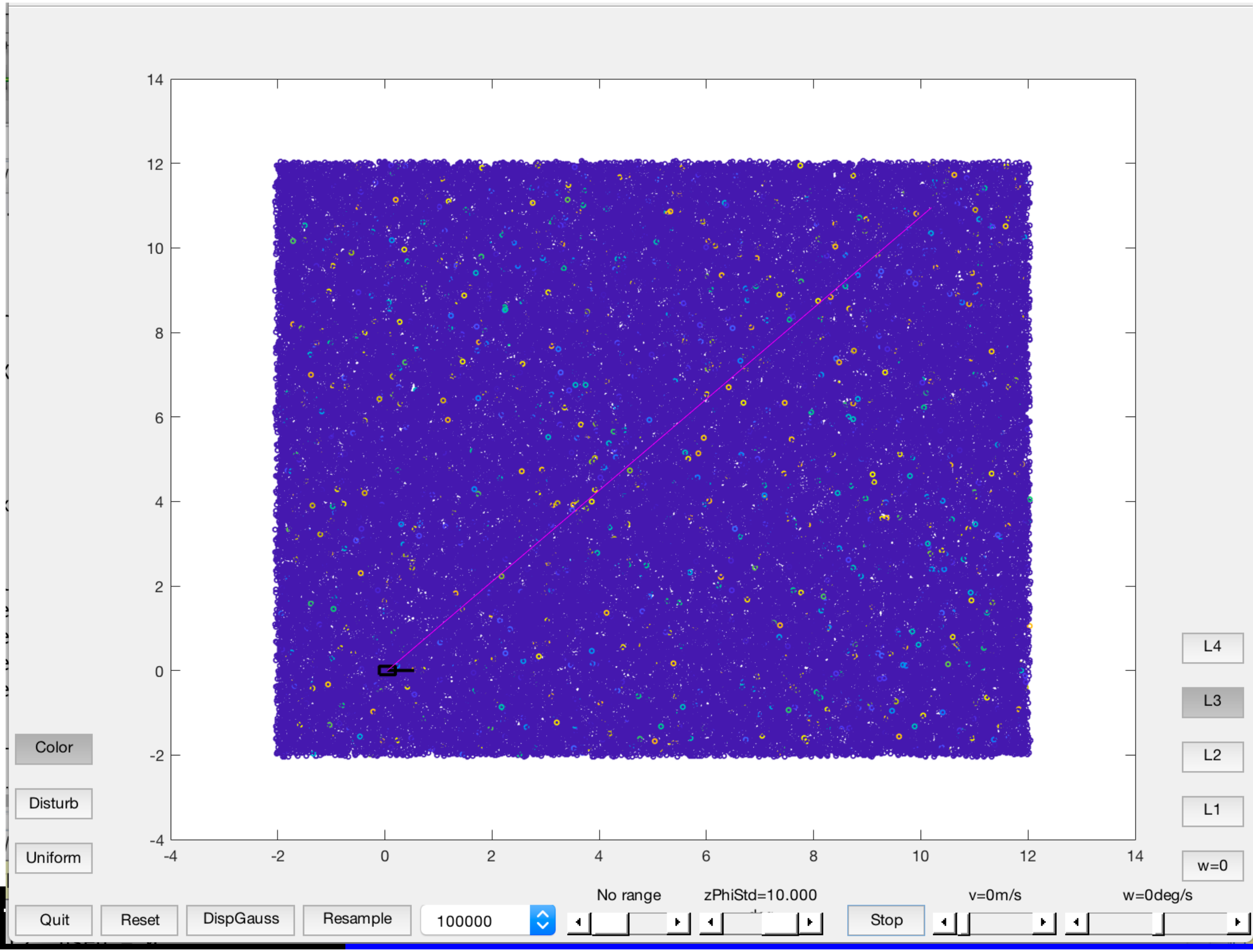
Update with range to two landmarks

- Smaller uncertainty
- Now closer to Gaussian



Update with angle to single landmark

- Why do we not see a clear peak?



High level summary: particle filter

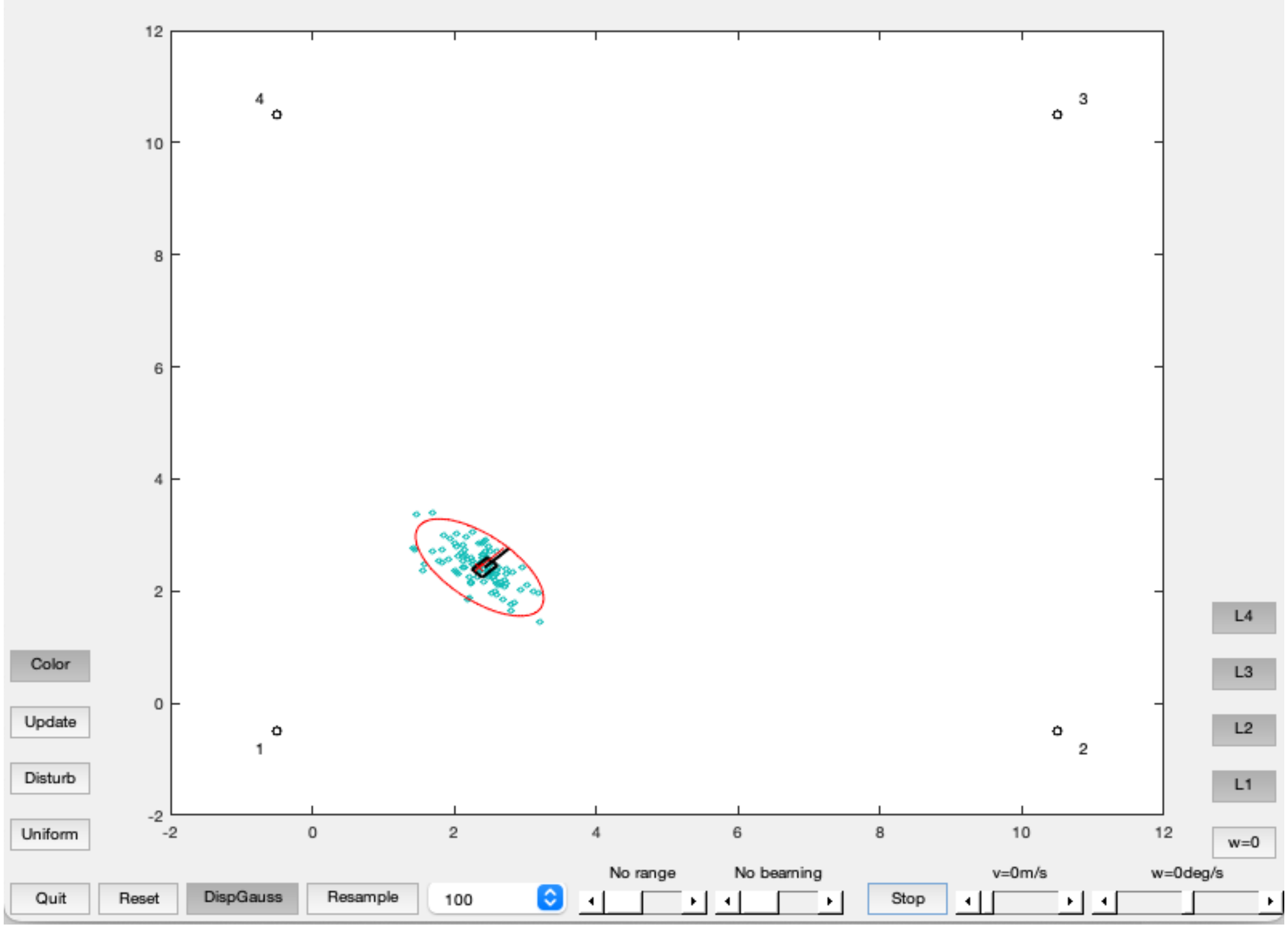
- Very powerful framework
  - Can be used for much more than localization
- Can represent any distribution (given enough particles)
  - Multi-modal no problem!
  - Can solve global localization, i.e. starting from uniform distribution
- Can handle non-linearities
- High dimensionality in the state a challenge
  - To sample the state space, the number of particles required grows exponentially with the number of dimensions.
    "the curse of dimensionality"
    - Too computationally expensive for some applications
- Accuracy depending on number of particles
- Requires some processing to get the "answer", i.e. what is the state? (often fits a Gaussian to distribution)

Gaussian approximation

- The Gaussian approximation to the probability distribution is quite good when the uncertainty is kept low
- Much more efficient representation than particle set
  - N(x,P) where x is the state estimate (the pose) and P is the estimate error covariance of that estimate.
    - 9 parameters for 3D statespace (x:3 + P:6) compared to 3N, where N>>1 for particle filter
  - BUT limited to uni-modal distributions

- Updated by the Kalman Filter
  - When there are non-linearities use
    - Extended Kalman Filter (EKF) or
    - Unscented Kalman Filter (UKF)

Gaussian approximation



Extended Kalman Filter
EKF Localization

- System dynamics ("motion model")

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}),$$

- Measurement model

Make noise explicit

$$z_k = h(x_k, v_k)$$

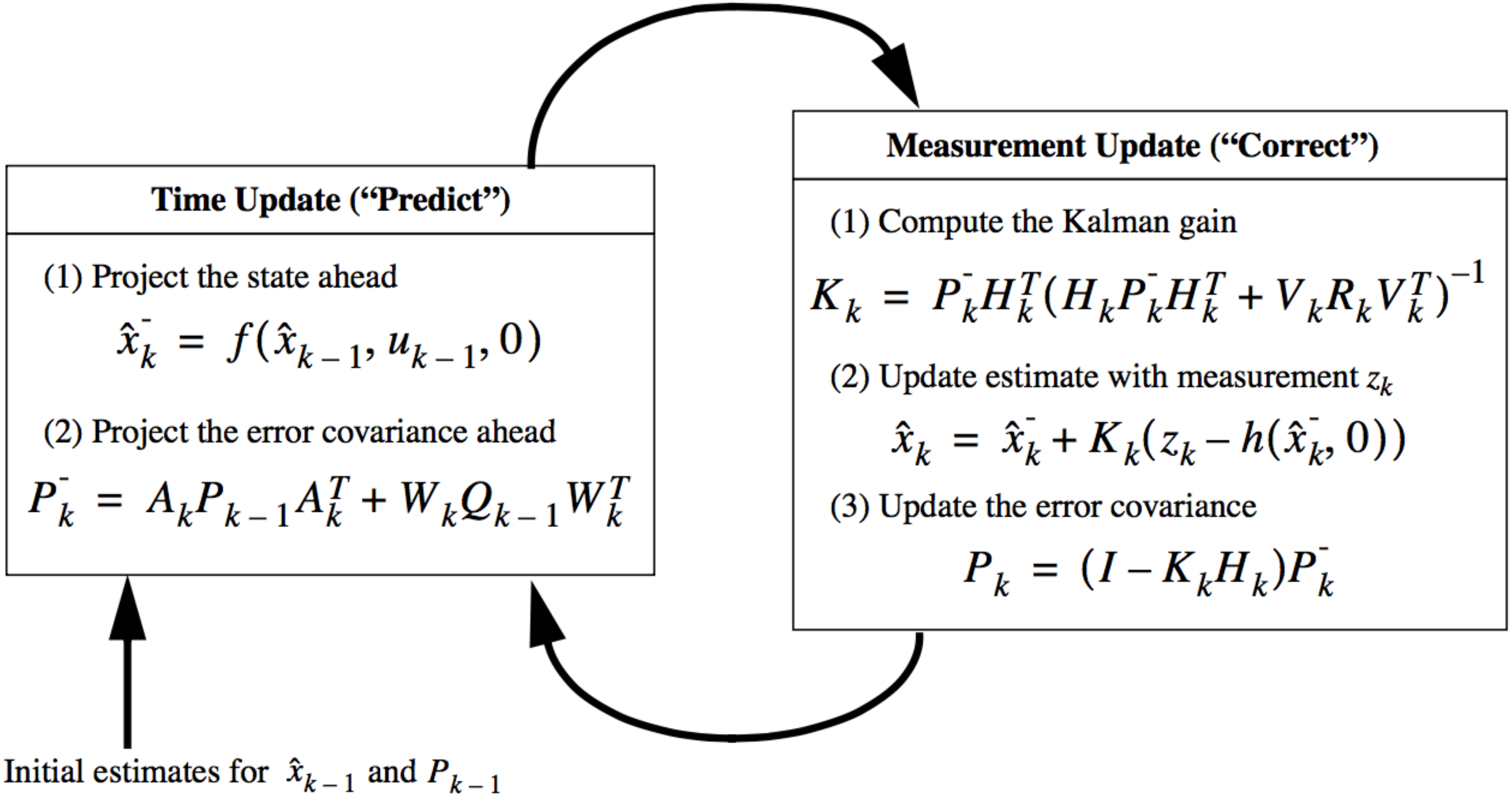
- Assumption on process and measurement noise (zero-mean and Gaussian)

$$p(w) \sim N(0, Q),$$

$$p(v) \sim N(0, R).$$

Extended Kalman Filter (EKF)

- A prediction step and an update step like for the particle filter
- K is the Kalman gain, weights process vs measurement noise



EKF Localization

- Motion model is the same as for particle filter
- Measurement model the same
- Need to calculate Jacobians
- Need to set noise covariance matrices Q and R
  - You can think of these as tuning values (just like in the case for the particle filter)
  - Start at values derived from “reasoning” about errors and then adjust for good performance.

Play with EKF

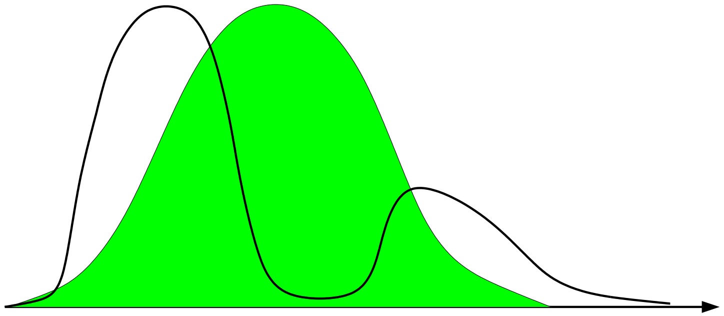
- Pure prediction (effect of noise?)
- Incorporate measurements (effect of noise?)
- Disturbances ("kidnapped robot")
- Global localization
  - What about large uncertainty and non-linearities

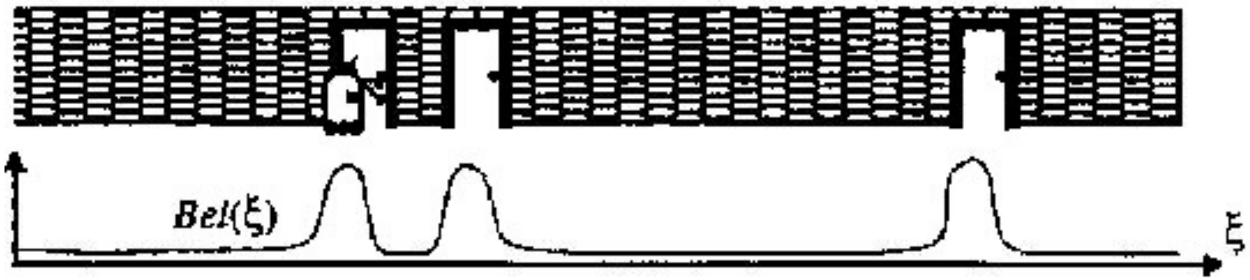
High level summary: EKF Localization

- Computationally very efficient
- Has been used extensively “everywhere” for a long time
  - The workhorse in target tracking for example
- Scales well to high dimensions
- Cannot handle multi-modal distributions
- Linearization problematic in cases of large uncertainty w.r.t to the non-linearities.
  - Dynamics very different between estimated (around which we linearize) and true state (where the system actually is) → bad estimates
- Need “good enough” initial guess to converge which depends on the type of use case.

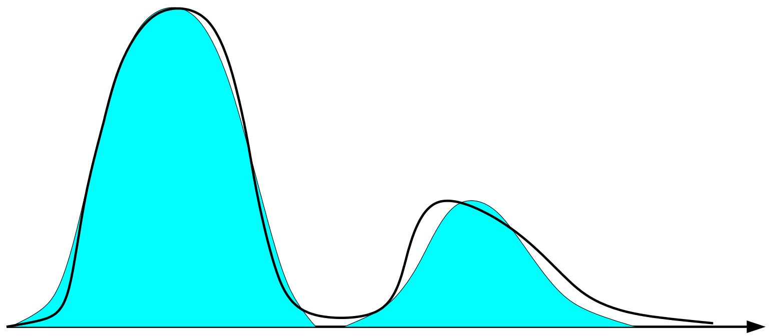
Limitations of a Gaussian

- A Gaussian cannot represent multi modal distributions!





- "Active global localization for a mobile robot using multiple hypothesis tracking", P. Jensfelt and S. Kristensen, IEEE Transactions on Robotics and Automation, 2001.
  - Shows that you can handle multimodal distributions using a mixture of Gaussians



What about those measurements?

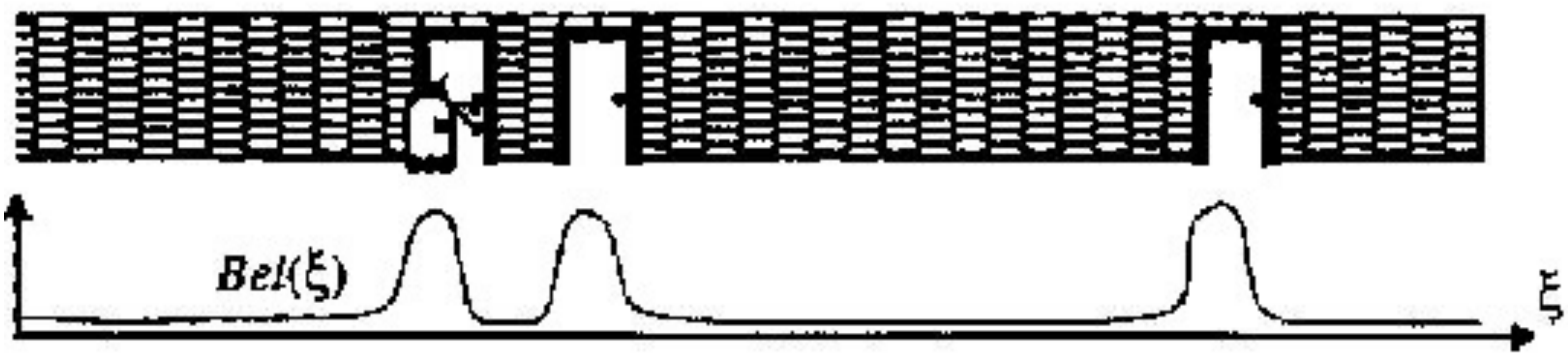
- What can we measure in practice
  - Range and bearing to a landmark (e.g. a corner)
  - Lines/planes → distances and angles
  - Transformation between images/laser scans
  - Ray trace in a map to get likelihood of measurement
  - …
- What we need:
  - Provide likelihood of the measurement given state $p(z|x)$

Additional real-world challenges

- Data association is super hard!
  - Need to associate a measurement with a landmark / part of a map
  - Ex: What parts of the environment did the M measurements from the laser come from??

Topological localization

- Discrete positions ("places").
  - Ex: A room, a subway station, near landmark X
- Pure topological localization depends on place recognition
  - Need to recognize the place and at least be able to tell all places connected to current place apart so that we know how we move
- Often combined with coarse metric information
- Ex: Discretize corridor into segments

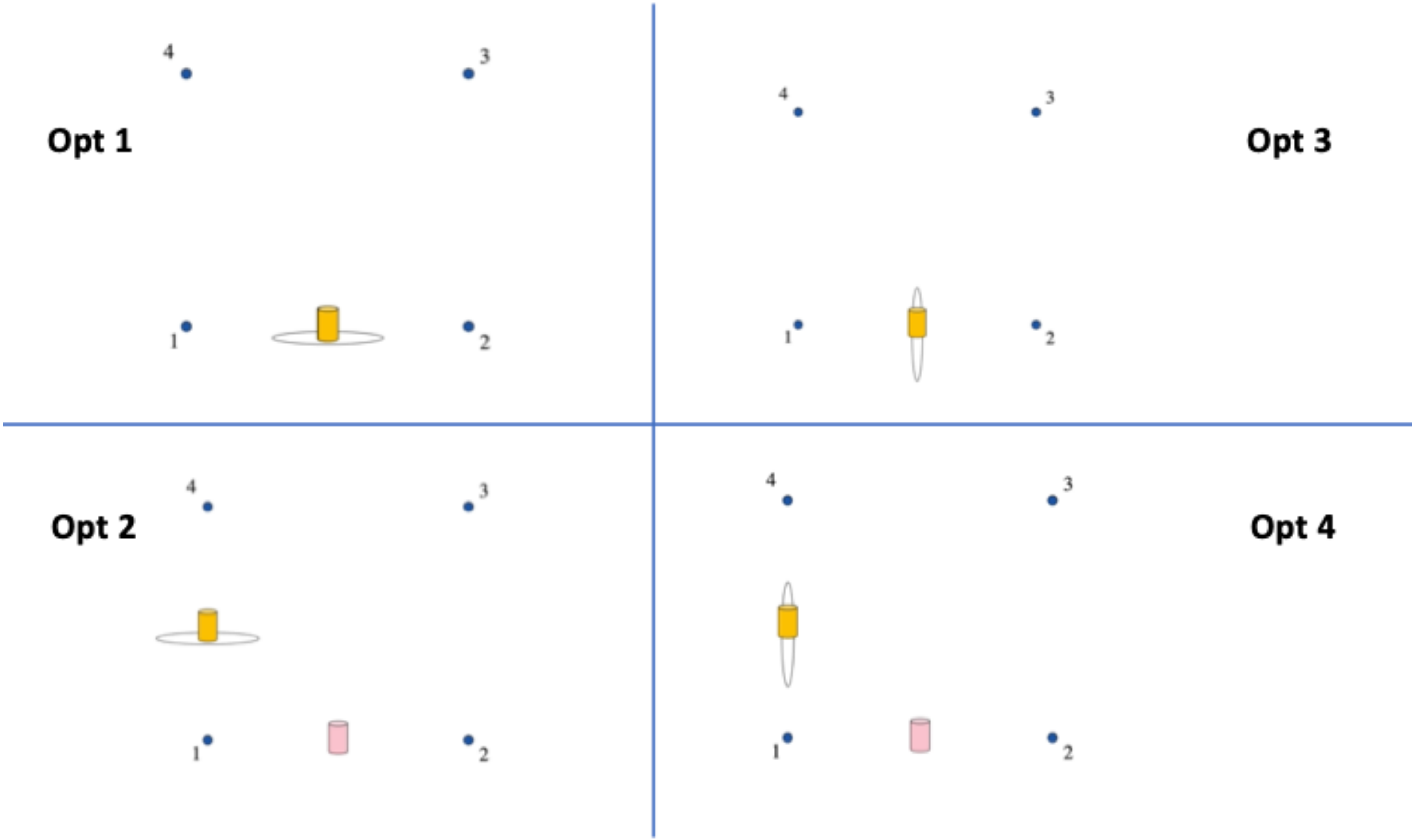


So much more to say

- Localisation is an extremely heavily researched area
- Needed by all (more or less) systems that move
- Localiztion using
  - Signal strength of WiFi signals
  - BLE
  - UWB
  - ...

If there is time

- Menti



4
3
Opt 1
1
2
4
3
Opt 3
1
2
4
3
Opt 2
1
2
4
3
Opt 4
1
2