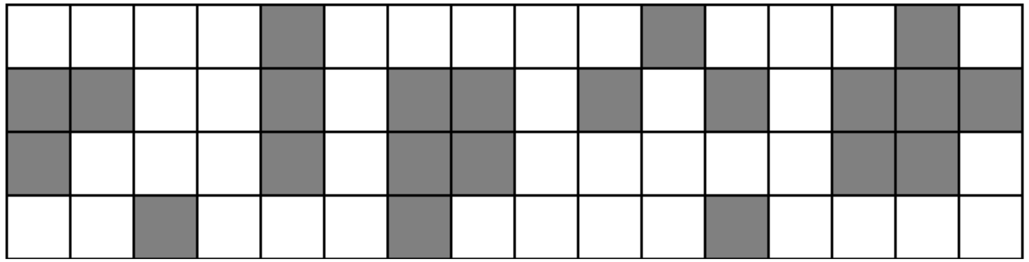
Agent Localization

Implemented with hidden Markov model

The Problem

- Agent is somewhere in the grid
  - Doesn’t know where (true state is HIDDEN), but knows the dimensions and where the obstacles are
- Agent has broken navigation system
  - Every time-slice it moves to a random (open) neighboring square
- Agent has cheap sensors that can report faulty readings
  - In this presentation’s graphics, an error rate of 20% is used



Review - Probabilistic Reasoning over Time

- Transition model $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})$
  - Read: Probability of the current state given the prior state
- Sensor model $\boldsymbol{P}(\boldsymbol{E}_t \mid \boldsymbol{X}_t)$
  - Read: Probability of the observed evidence given the current state
- Filtering - $\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{x_t} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t) \boldsymbol{P}(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$

Normalizing constant (so prob. sums to 1)

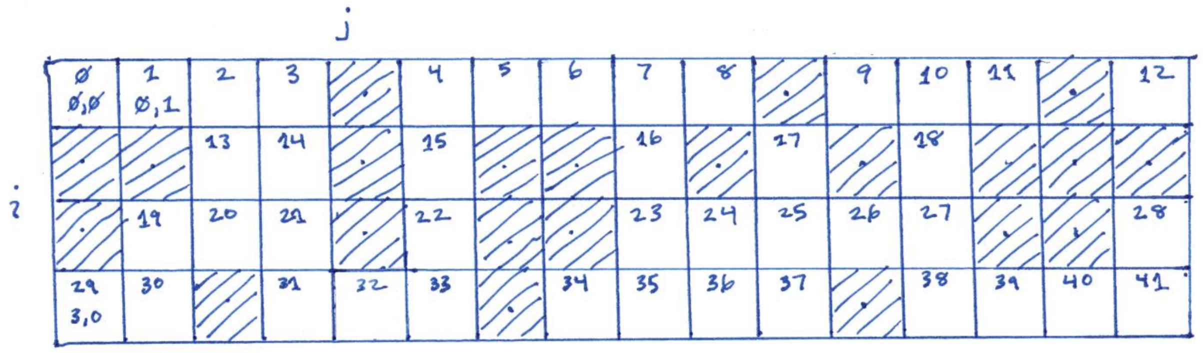
Sensor model

Transition model

Current state, aka the forward message $\boldsymbol{f}_{1:t}$ in subsequent slide (recursive)

Review – HMM Matrix Implementation

- Hidden Markov model (HMM) – state of the process is described by single discrete random variable
  - What is “hidden” is the true location of the agent
- Initial belief state = $\boldsymbol{P(X_0)}$ any occupiable square with equal probability, this is the first forward message



Transition matrix $P(X_{t+1}=j \mid X_t=i) = \pi_{ij} = (1/N(i)$ if $j \in$ Neighbors(i) else Ø$)$

42x42 Example: Square 3Ø, there is a 50% chance we move to 29 and 50% chance to 19.

So in our transition matrix, $\pi_{30,19} = .5$ and $\pi_{30,29} = .5$

Sensor matrix $P(E_t=e_t \mid X_t=i) = O_{tii} = (1-\epsilon)^{4-d_{it}}\,\epsilon^{d_{it}}$, all else Ø

42x42 @ each time t we get new evidence

Example: Square Ø, $\epsilon = .2$: $P(NSW \mid NSW) = O_{Ø,Ø} = (1-Ø.2)^{4-Ø}\, Ø.2^{Ø} = Ø.4Ø96$ (sensor, true)

Square 1: $P(NSW \mid NS) = O_{1,1} = (1-Ø.2)^{4-1}\, Ø.2^{1} = Ø.1Ø24$

$$\begin{array}{c|c} & 42 \rightarrow \\ \hline 42 \downarrow & \begin{bmatrix} Ø.4Ø96 & Ø & Ø & \cdots \\ Ø & Ø.1Ø24 & Ø & \cdots \\ Ø & Ø & & \cdots \\ \vdots & \vdots & & \end{bmatrix} \end{array}$$

Review – HMM Matrix Implementation

- Now that we know what the $\mathbb{T}$ and $\mathbb{O}$ matrices look like, here is **filtering**:

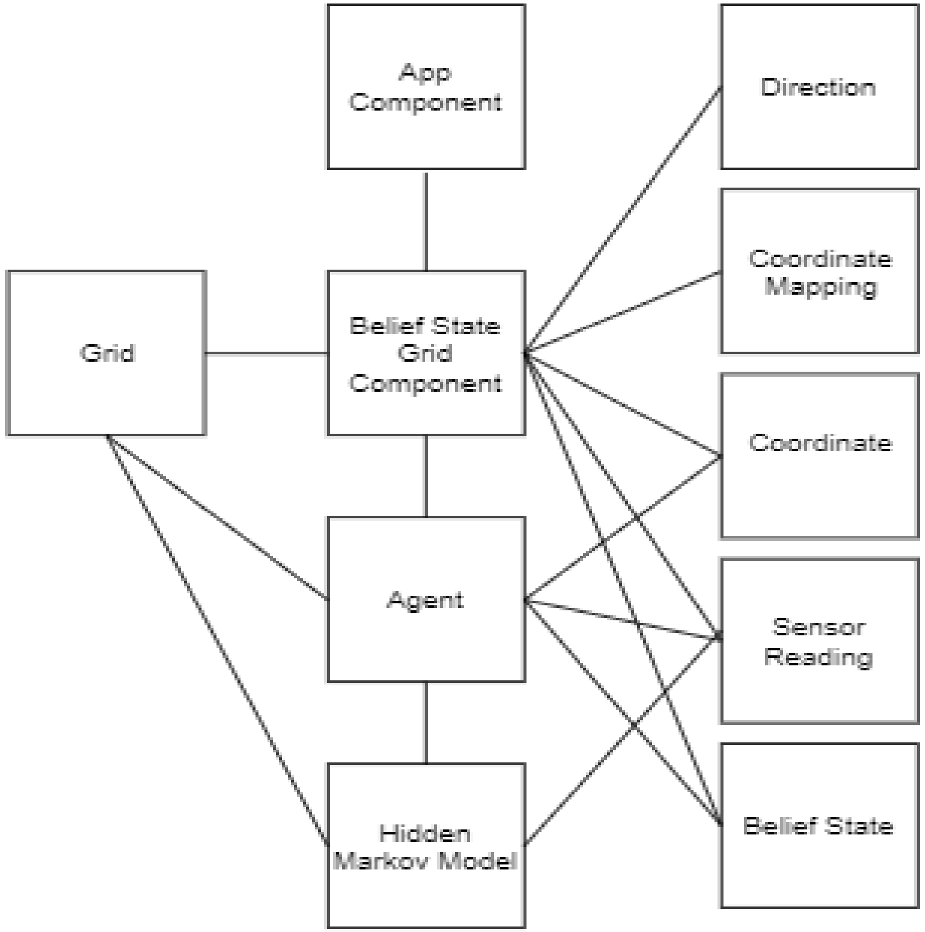
$$\boldsymbol{f}_{1:t+1} = \alpha \mathbb{O}_{t+1} \mathbb{T}^{\mathrm{T}} \boldsymbol{f}_{1:t}$$

Note: $\mathbb{T}$ $\mathbb{T}_{i,j}^{T} = \mathbb{T}_{j,i}$ (transpose - turn rows into columns and columns into rows)

Demo

- http://localhost:4200/#

Implementation



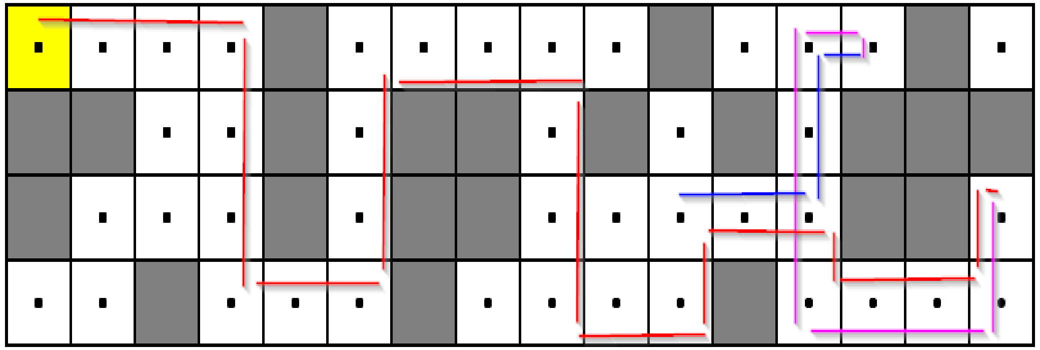
- **Belief state grid component** – draws grid based on belief state, allows stepping through time
- **Agent** – gets sensor readings, updates belief state using hmm
- **Hidden-markov-model** – the model the agent uses to update belief state
- **Grid** – things to do with the environment (neighbors,etc.)
- **Models** (right side)

GitHub

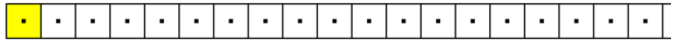
- https://github.com/jacob-barna/MarkovSensorLocation

Experiments

- Two environments… In each environment, use a set path



16 columns, 4 rows, 22 obstacles



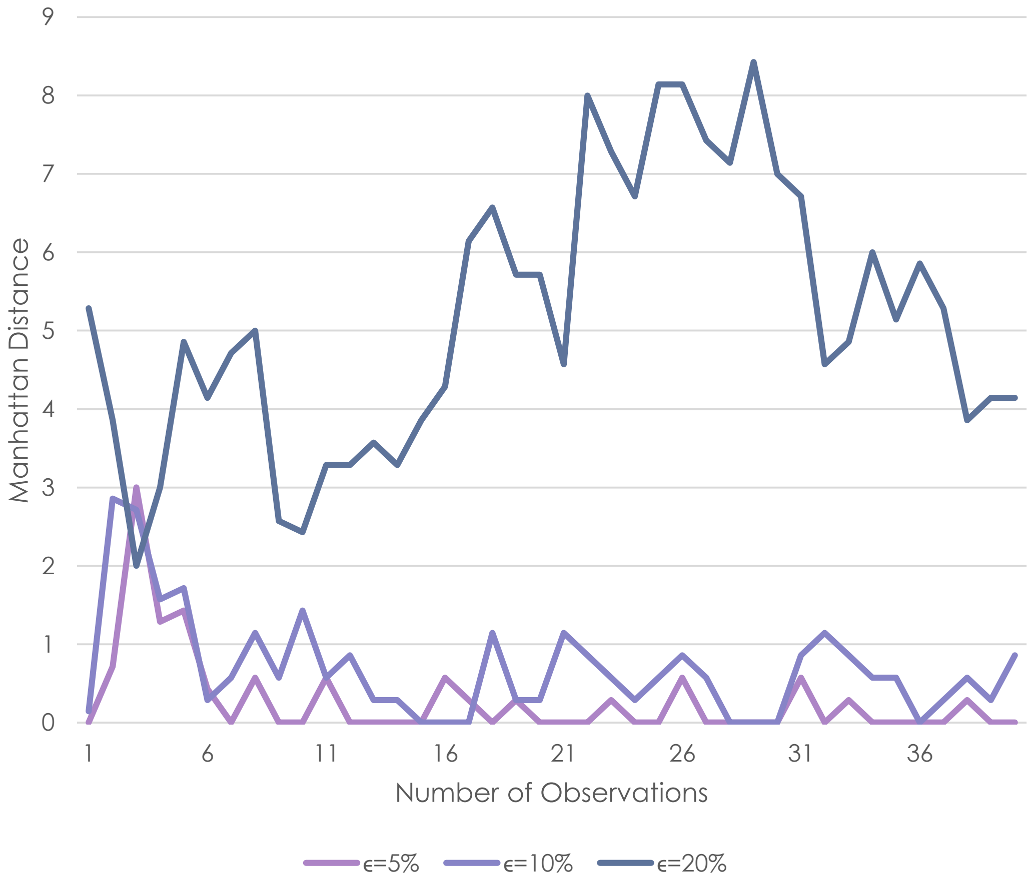
64 columns, 1 rows, no obstacles

- In each environment, run same path for 7 runs at 3 different error rates

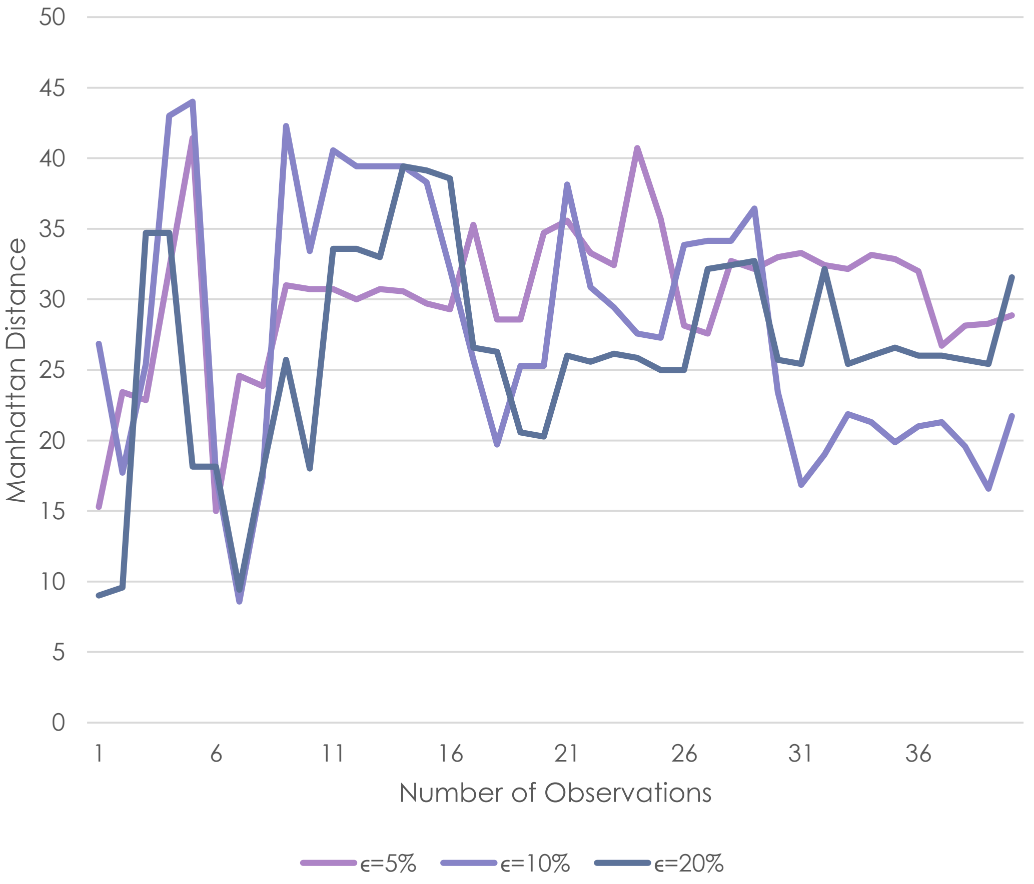
Error Rate	Number of Runs
5%	7
10%	7
20%	7

Results

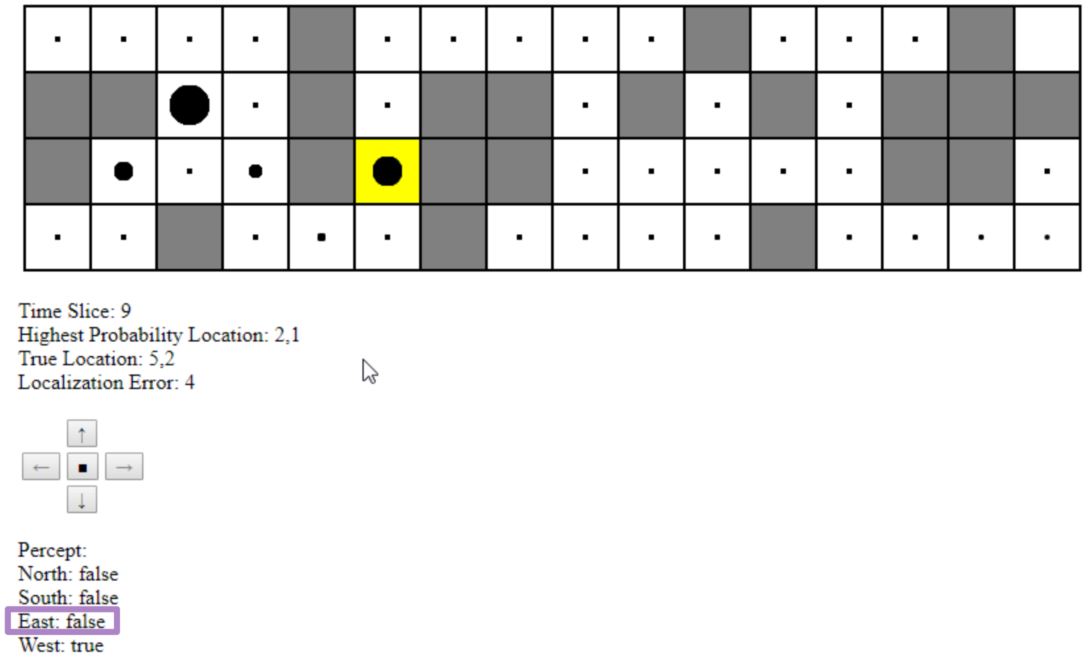
Experiment 1 – Average distance; 7 runs



Experiment 2 – Average distance; 7 runs



Some Observations



- With high error rate, even after many moves, this environment 1 quite often had the true location and perceived location on opposite ends of the grid
- Environment 2 – the agent had no idea where it was
- Environment matters – starting position matters

Future Work

- Randomize or allow user to set starting location to study how this affects early estimation
- Viterbi algorithm – path accuracy – can we work out where we started and what path we took?

The End