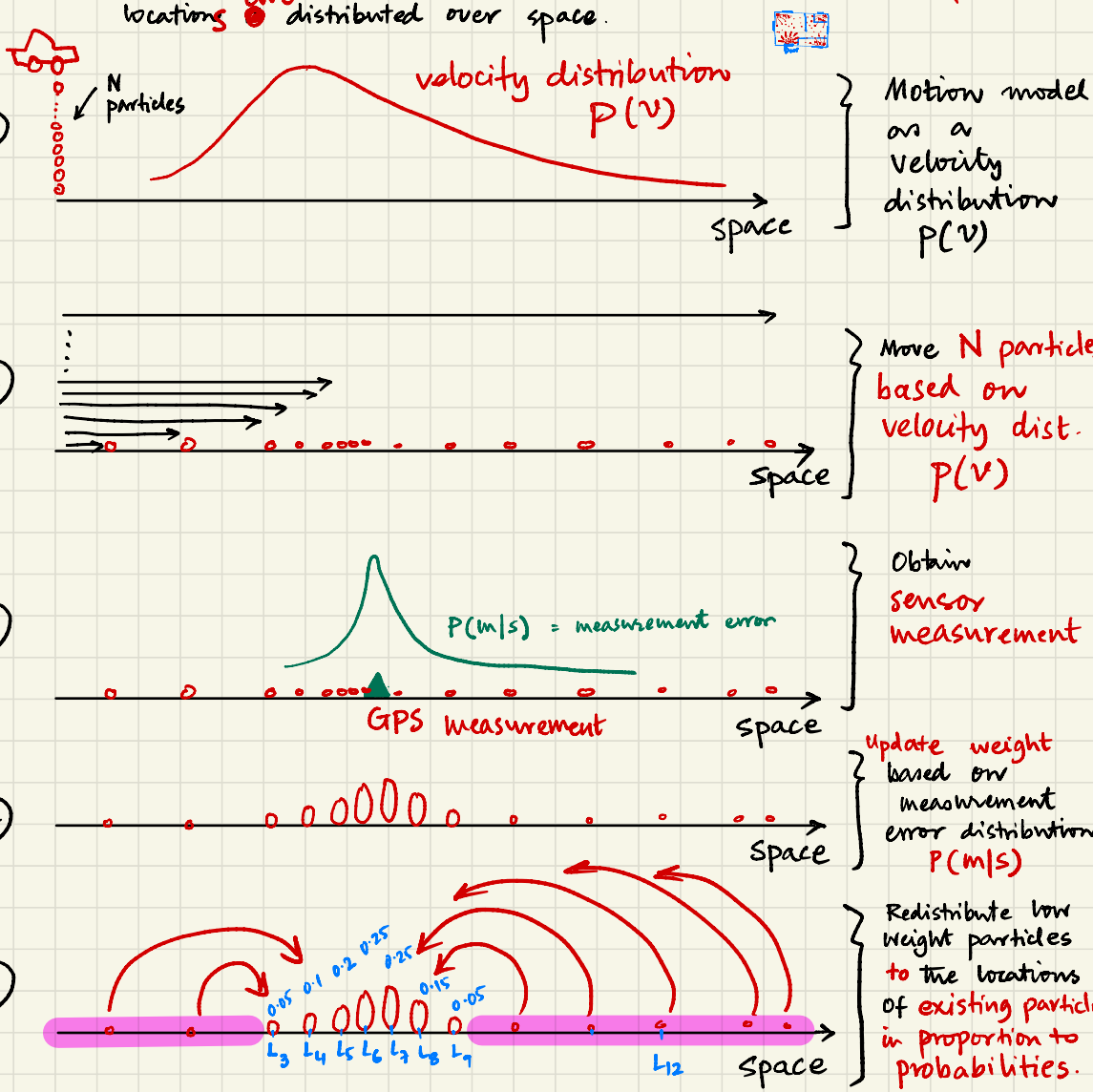
Particle Filter (just intuitions)

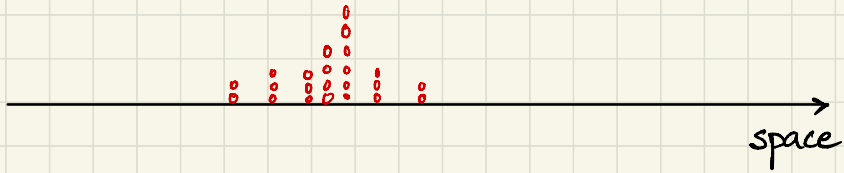
Model the location of the object with a particle at that location.

Since we know the object's location probabilistically, let's scatter N particles at most likely initial loc. and then compute a weight for each particle based on its agreement with motion models & measurement.

At any time, the particles capture ~~how~~ my belief about how the object locations are distributed over space.

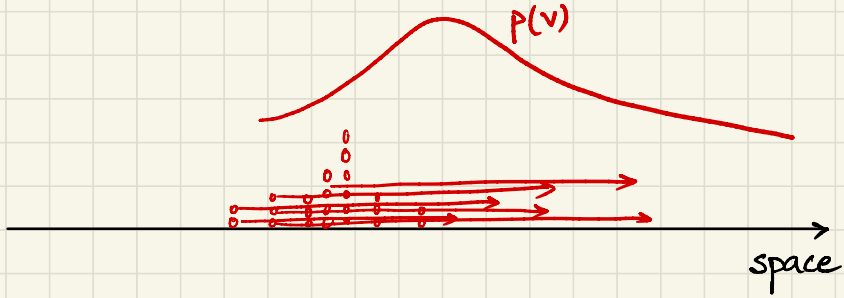


6



Still N particles in total. But more particles where belief is greater.

7



Apply motion model p(v) on each particle and repeat

⋮

Observe :

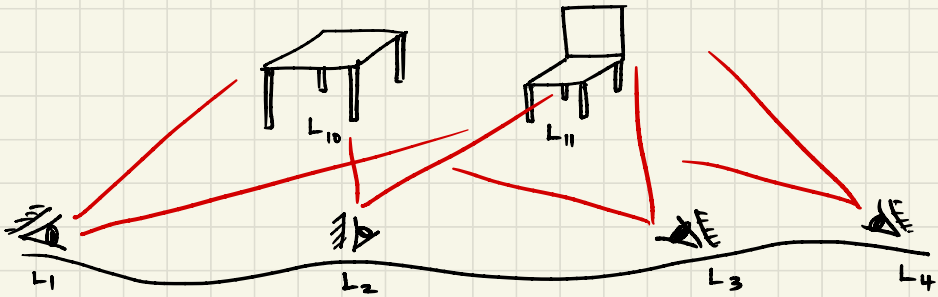
- Always N particles in the system
- Scattering of N particles captures my current belief of s_k
- Particles are like magnifying lens (or clones) → you allocate more particles to places/trajectories that you want to search with high resolution.

Observe :

1. P.F. operating on continuous space.
2. Any kind of motion model can be incorporated
3. Noise can be modeled
4. $s_k = f(s_{k-1}) + n$, $m = g(s_k) + n$ } any f and g possible even if non-linear
5. CPU load is function of N, so you can choose N based on system resources.

Kalman Filter : When f and g are linear AND noise is gaussian } simple algorithm in that case.

SLAM : Simultaneous localization & mapping



Main goal :

Can I compute both the $\underbrace{\text{sensor locations}}_{\langle L_1\ L_2\ L_3\ L_4 \rangle}$ and the visible $\underbrace{\text{objects' locations}}_{\langle L_{10}\ L_{11} \rangle}$

Basically, I want : $P(\overset{\text{camera}}{S^{(c)}_{1:n}}\ \overset{\text{object}}{S^{(o)}_{1:n}} \mid m_{1:n})$