Algorithm development using MATLAB

- Object tracking by particle filter -

MathWorks Japan

Application Engineer

EIJI OTA

Big Loop in Algorithm Development

Simulation

Mathematical Concept

Experiment of Concept

Result

Agenda

1. Introduction to particle filter

- What is particle filter ?
- How can we track an object in a movie ?

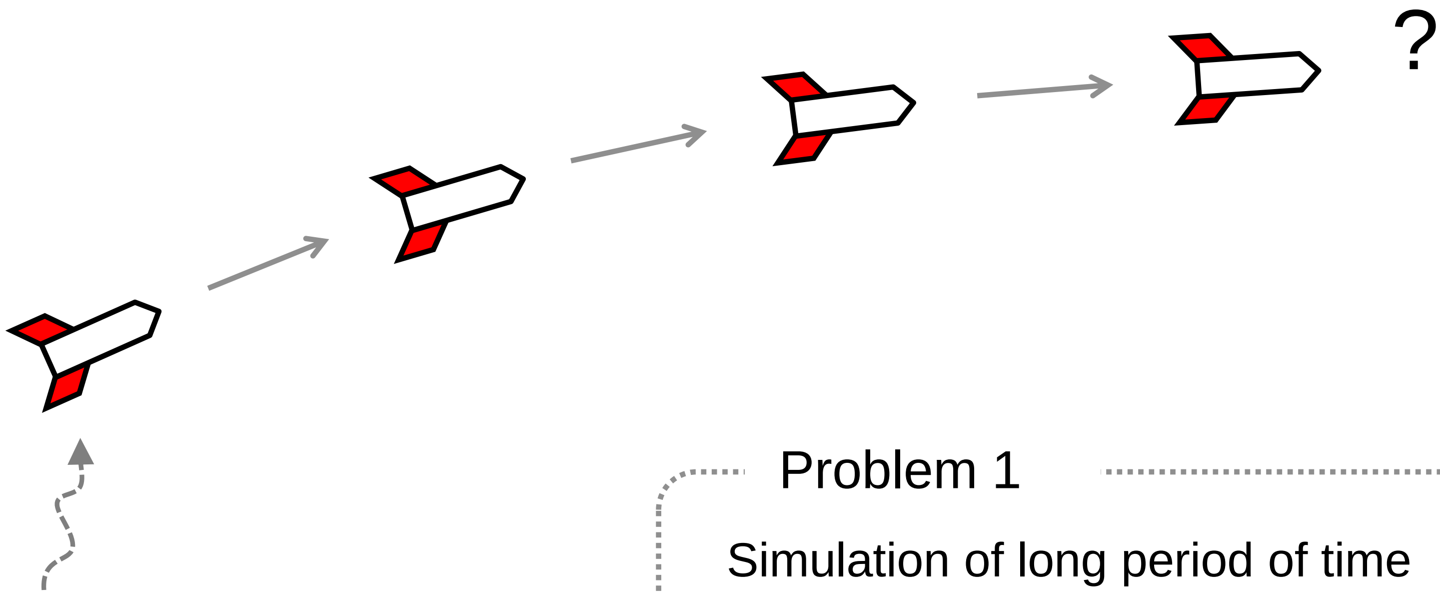
Mathematical Concept

2. Review of sample program

- What is MATLAB ?
- How can code particle filter ?

Experiment of Concept

How can we predict or estimate something we cannot see or touch?



You can predict this rocket's trajectory by solving differential equation… but

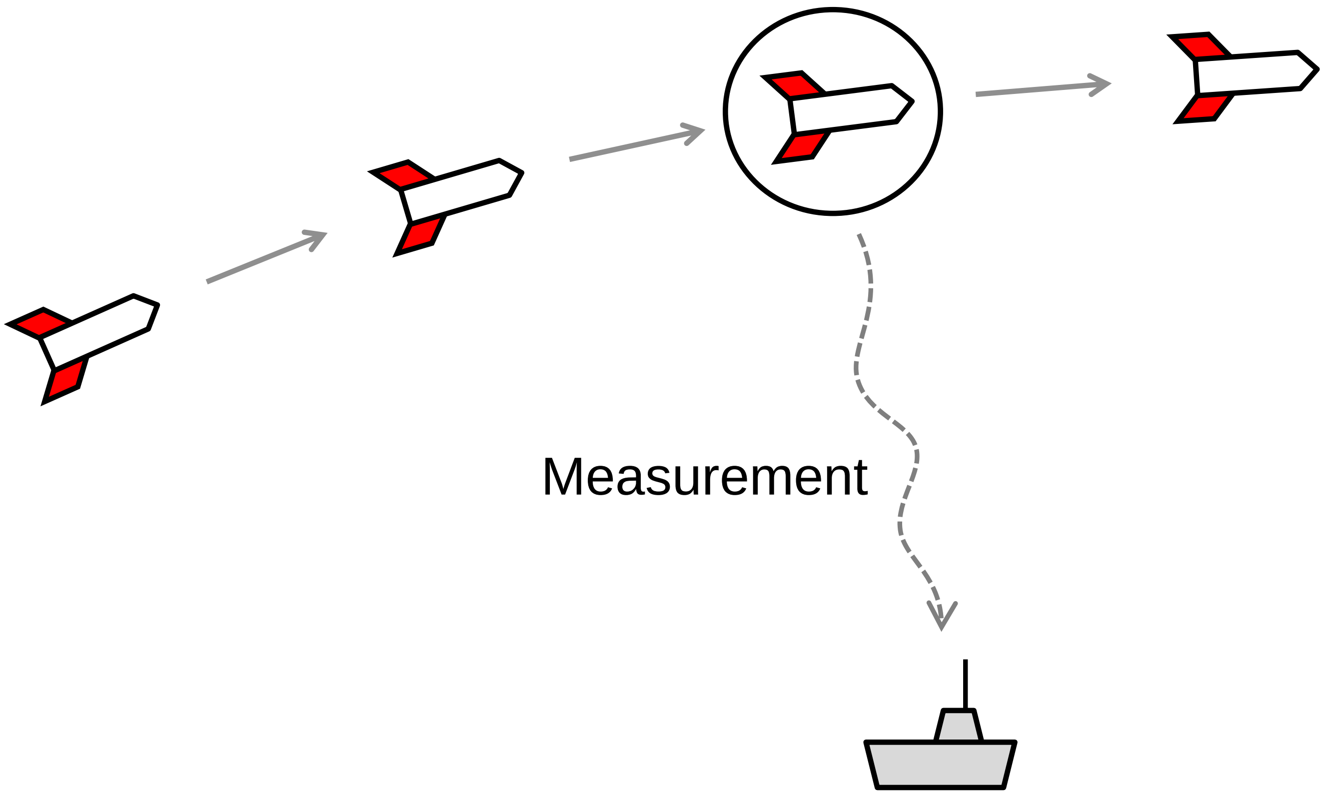
Problem 1

Simulation of long period of time might cause accumulation of error

Problem 2

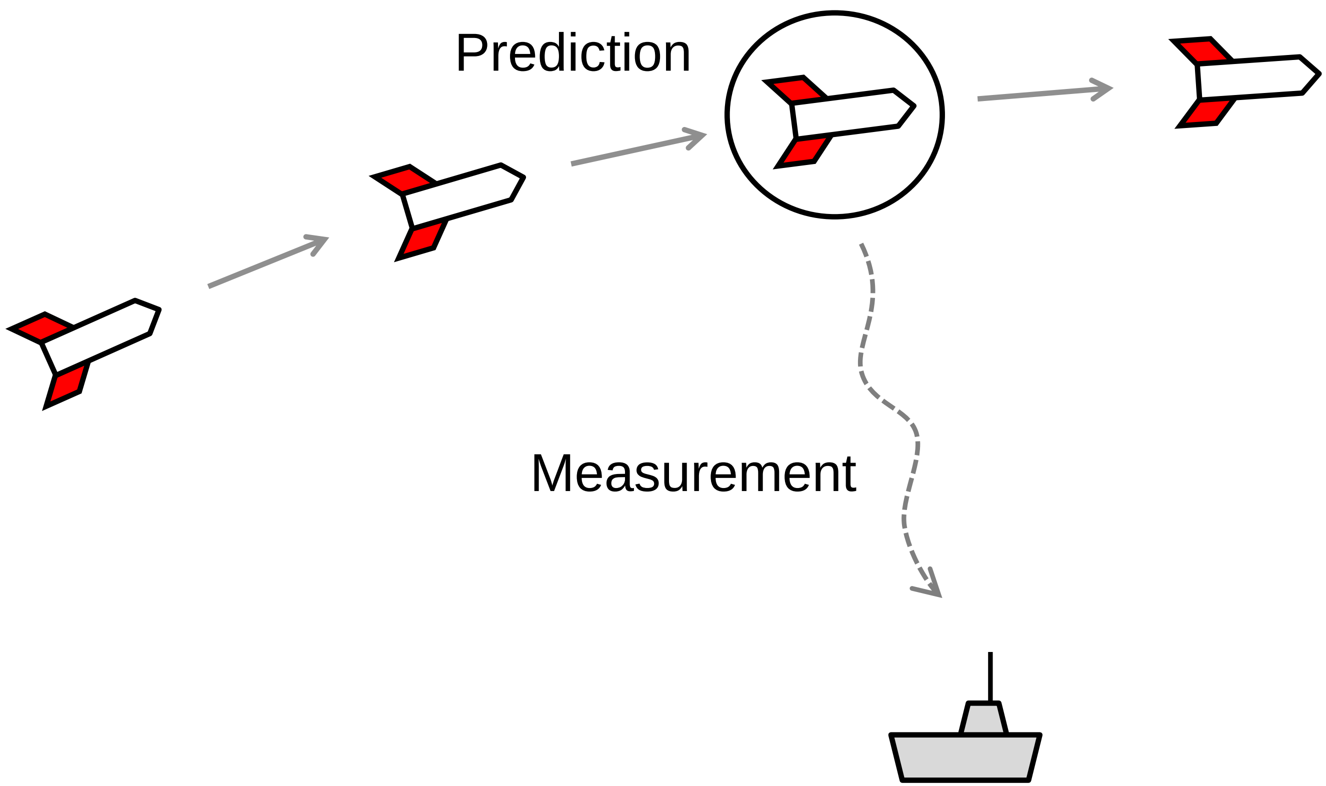
Smallest error of initial value might cause a drastic change of solution (Butterfly effects)

You might think a good measurement might solve this problem …



But single measurement might not be enough to estimate the location of rocket accurately ...

If neither perfect prediction nor perfect measurement exists …



Why don’t we combine these two methods ?

What is data assimilation ?

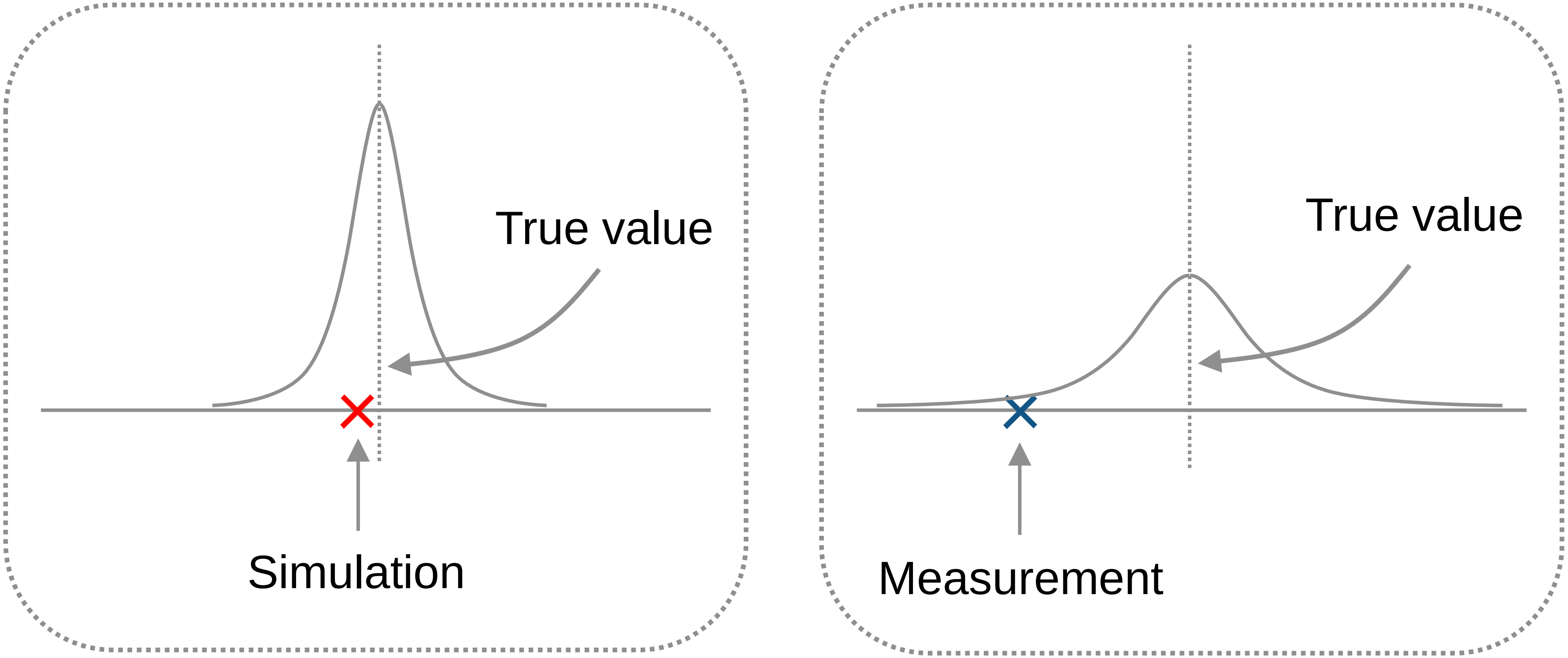
Simulation + Measurement

Mathematical technique combing simulation and measurement

Data Assimilation

Popular examples are karman filter, particle filter, and etc.

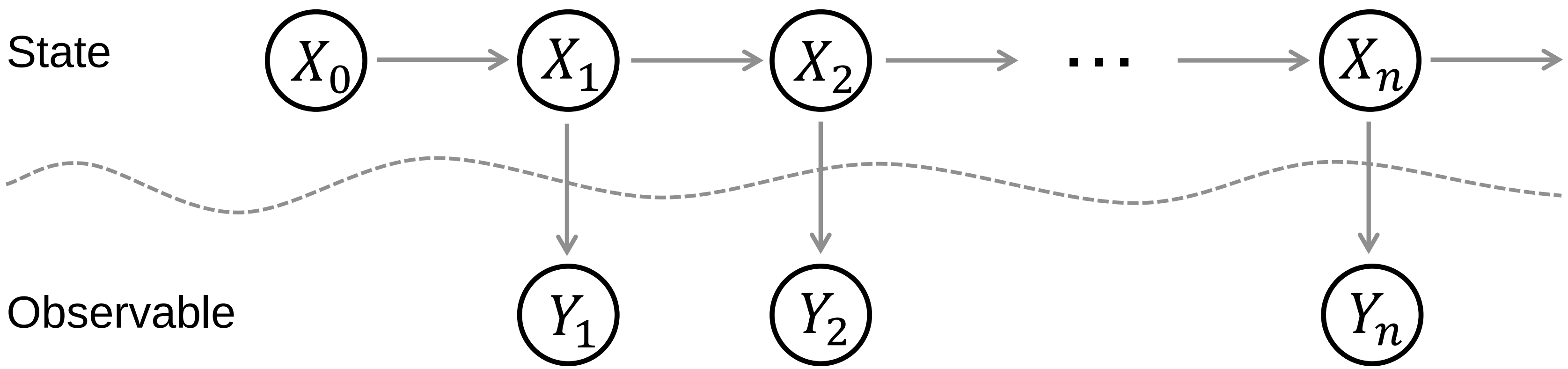
Combining simulation and measured data



Mean value of these 2 values is good enough? No!!

We need more delicate argument

General State Space Model

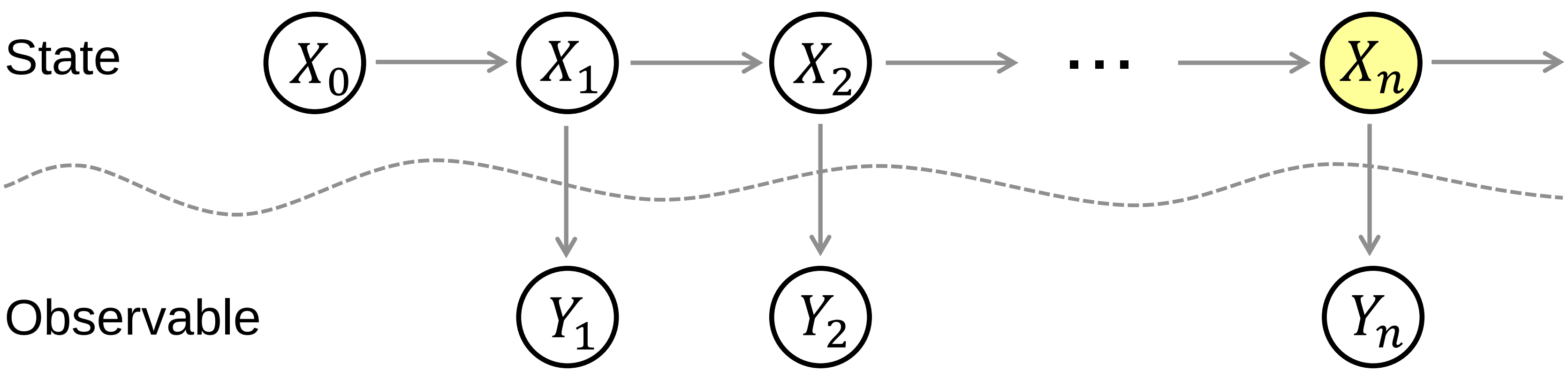


System Model : $X_n = f(X_{n-1}) + W_n$ — System Noise

Observation Model : $Y_n = g(X_n) + V_n$ — Observation Noise

What is the purpose of particle filter?

⇨ Generate samples which follows filter distribution

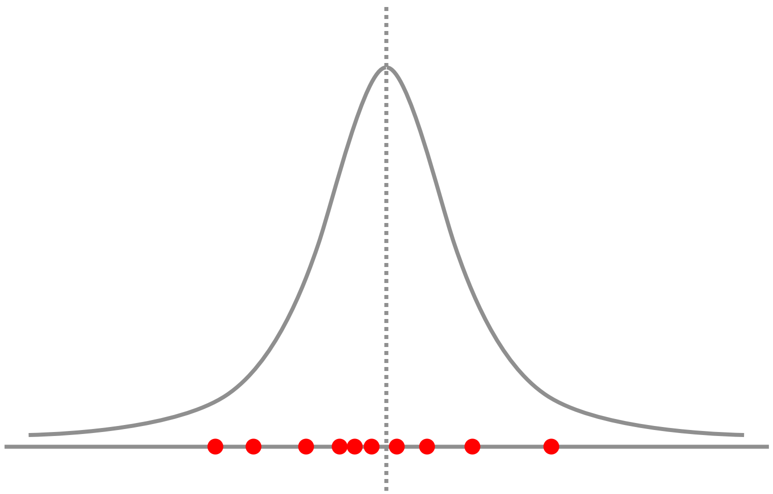


$P(X_n \mid Y_1, Y_2, \cdots, Y_n)$ ⟵ Filter distribution

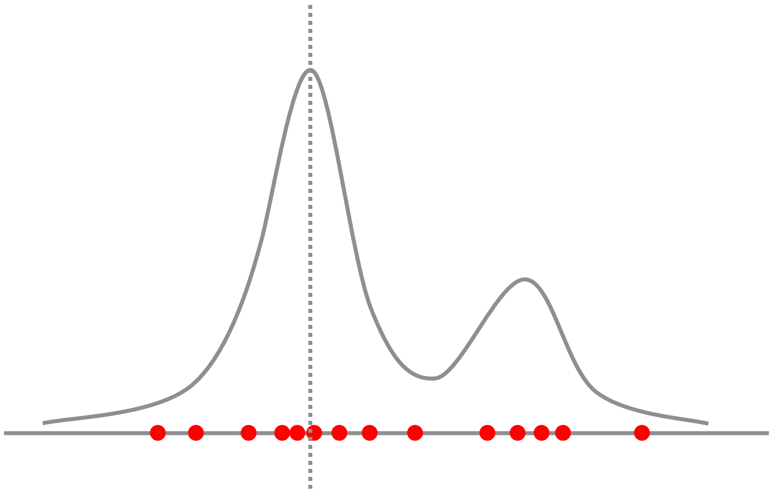
$$= \int \cdots \int P(X_0, X_1, \cdots, X_n \mid Y_1, Y_2, \cdots, Y_n)\, dX_{n-1} \cdots dX_0$$

If you have a lot of samples of filter distribution…

You can estimate the state by these methods



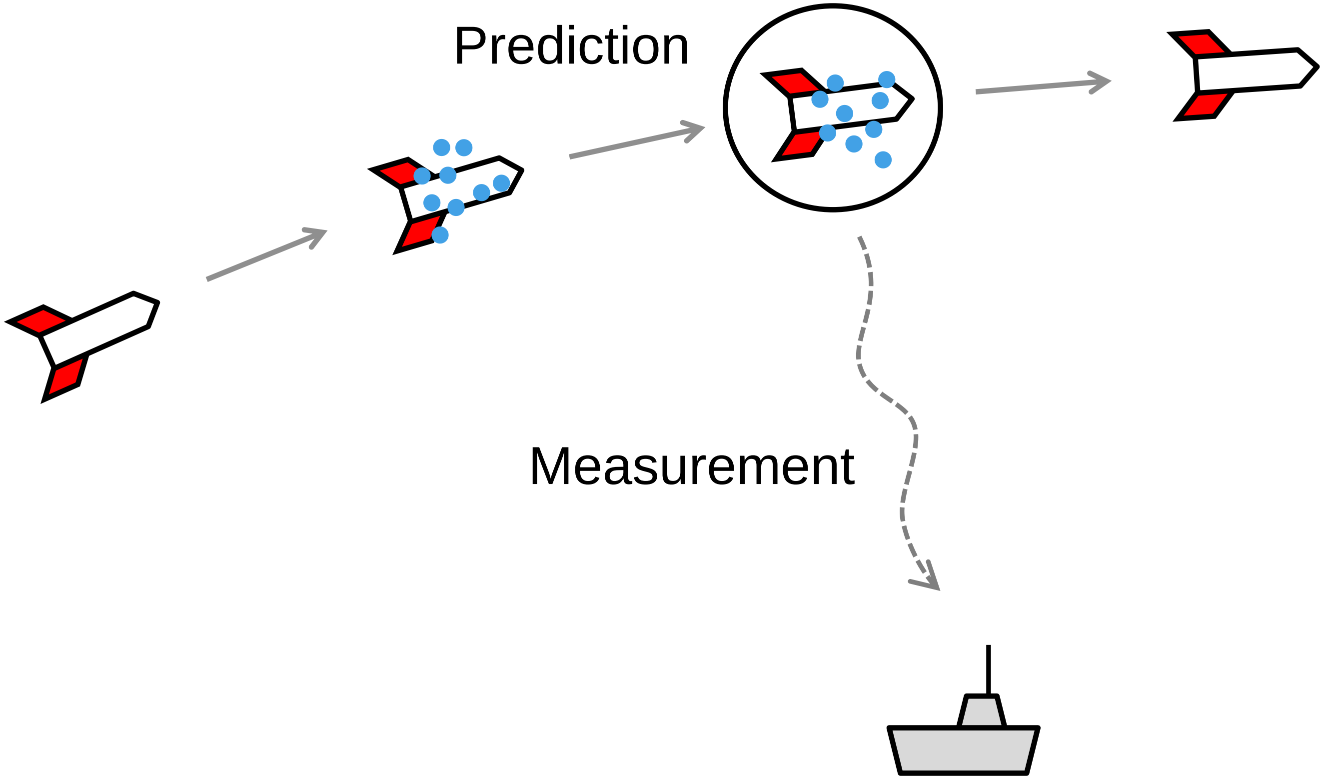
1. Using mean or median



2. Using empirical distribution

What is particle filter?

State estimation method using a lot of particles



Algorithm

1. Prediction

Time evolution of particles according to system model

2. Filtering

Reselection particles according to their likelihood

How we describe variables

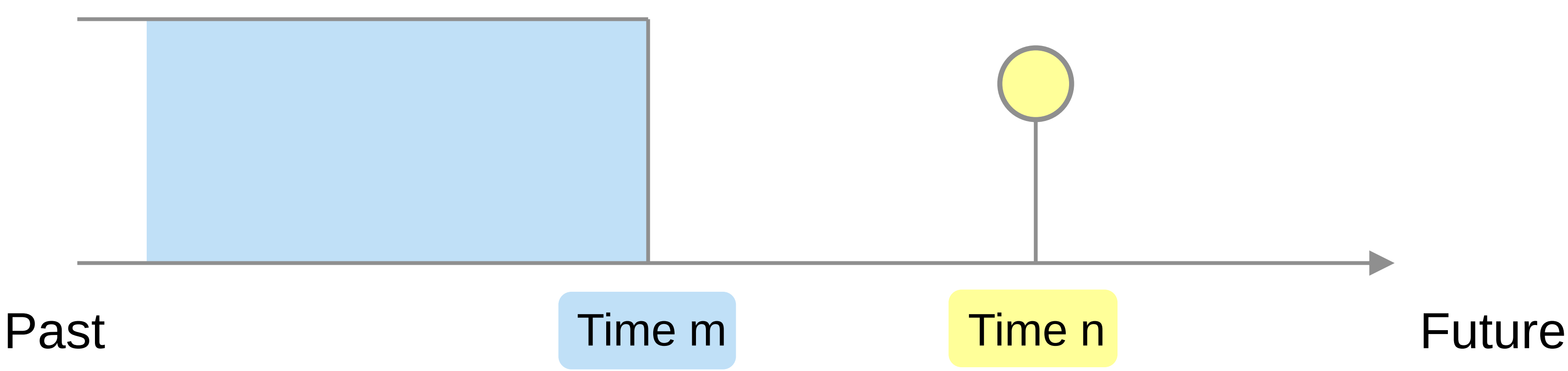
X_n : Estimator of state at time n

$X_{n\,|\,m}$: Estimator of state at time n

(when you emphasize X_n is using information up to time m)

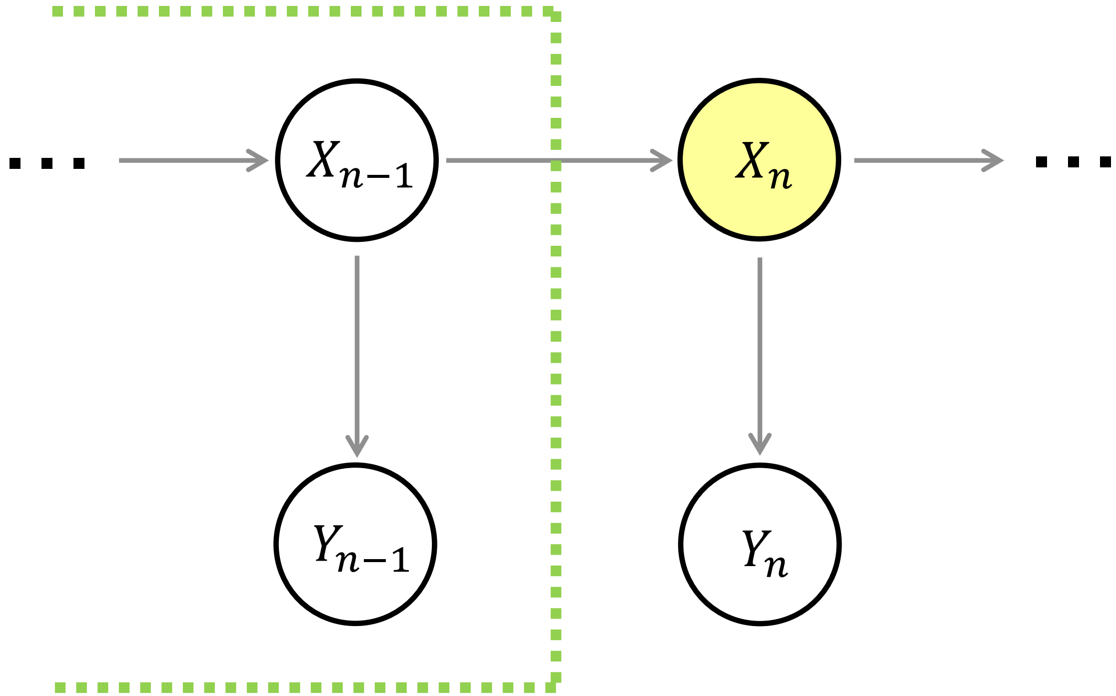
$X^{(k)}_{n\,|\,m}$: Estimator of state at time n

(when you emphasize $X_{n\,|\,m}$ is from k-th paritlce)



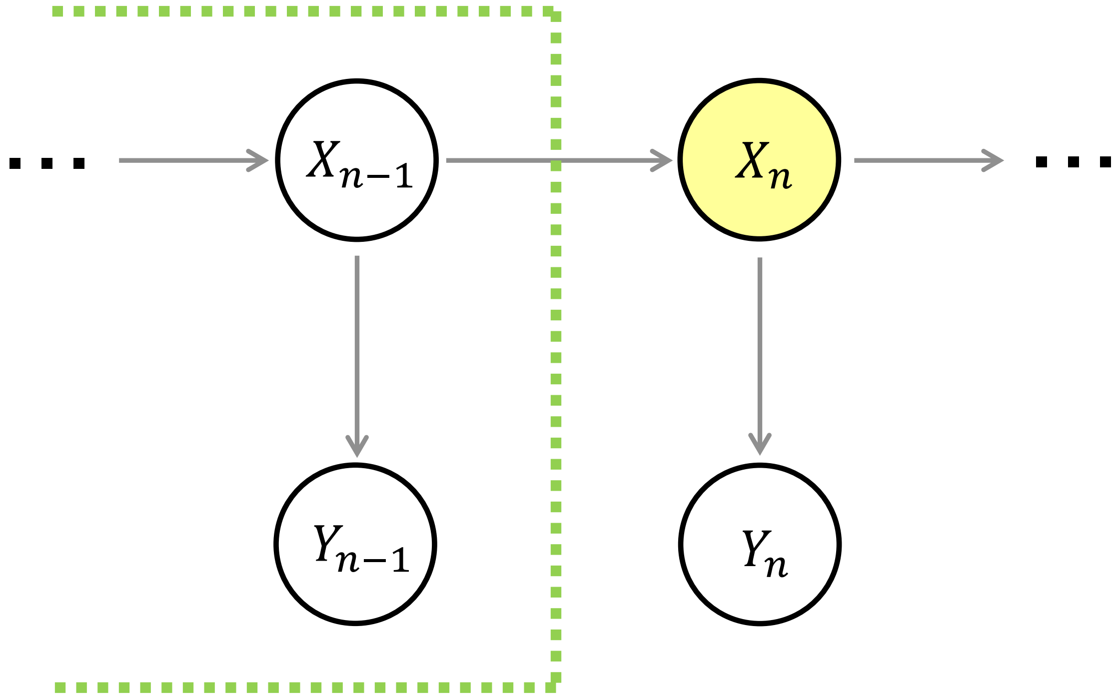
EX) $X_{n\,|\,n-1}$ means

"Estimator of state at time n using information up to time n-1"



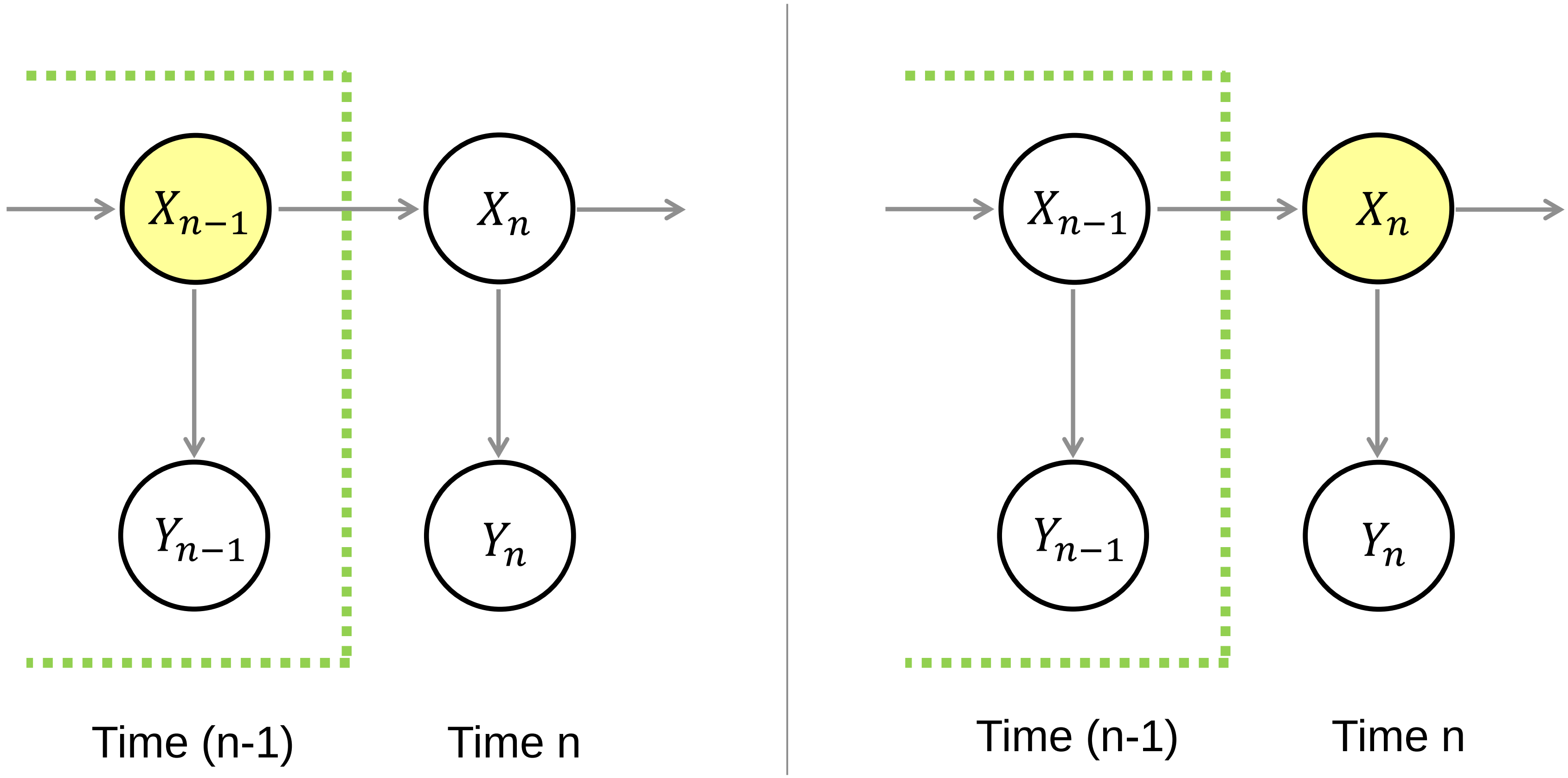
EX) $X_{n\,|\,n}$ means

"Estimator of state at time n using information up to time n"



Prediction

$X^{(k)}_{n-1\,|\,n-1}$ $\rightsquigarrow$ $X^{(k)}_{n\,|\,n-1}$



Prediction

Time (n-1)

$$X^{(k)}_{n-1\,|\,n-1}$$

Time n

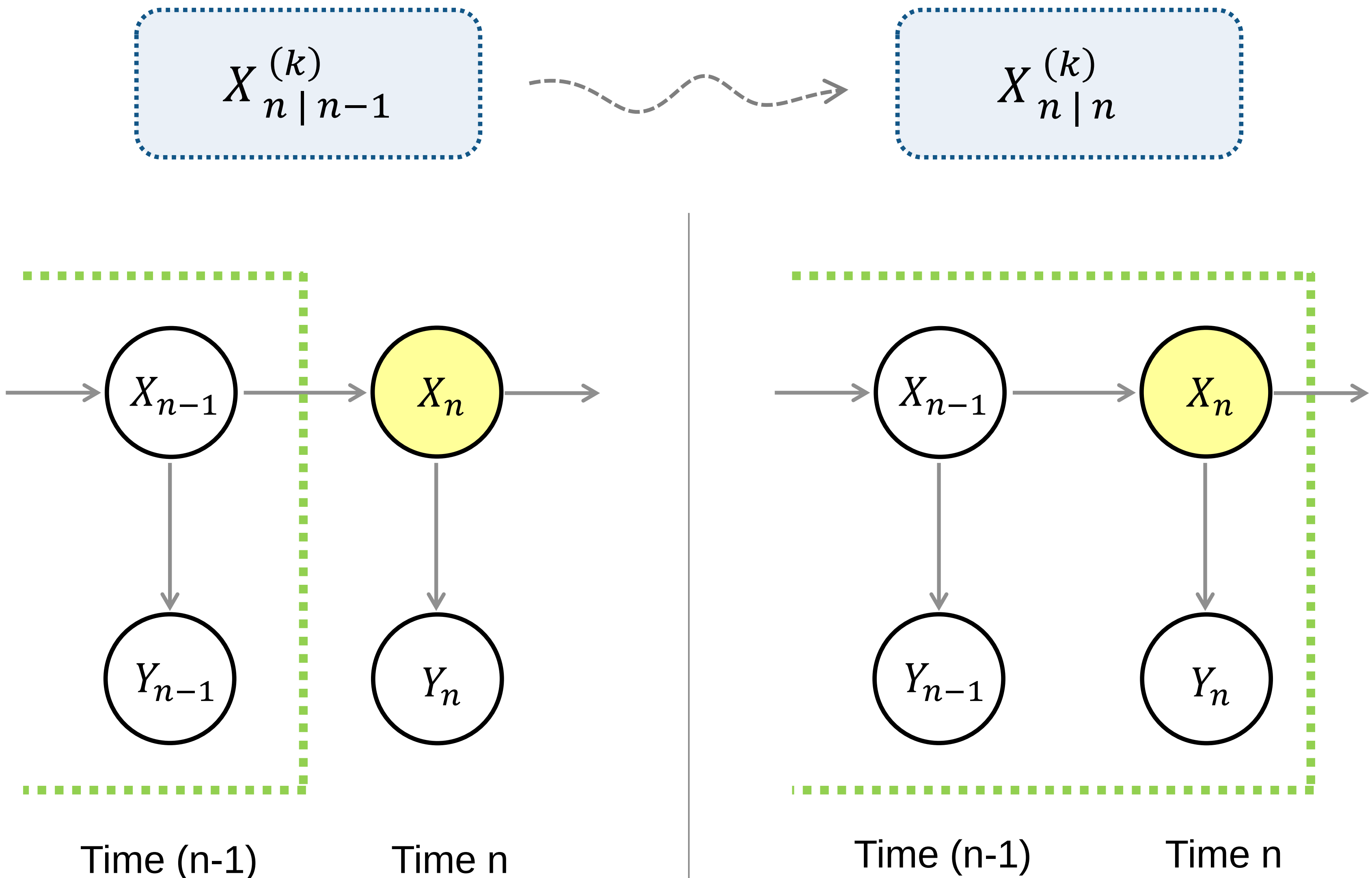
$$f\left(X^{(k)}_{n-1\,|\,n-1}\right) + W_n$$

Prediction

Random Number

Gap from mathematical model

Filtering



$X^{(k)}_{n\,|\,n-1}$
$X^{(k)}_{n\,|\,n}$
X_{n-1}
X_n
Y_{n-1}
Y_n
Time (n-1)
Time n
X_{n-1}
X_n
Y_{n-1}
Y_n
Time (n-1)
Time n

Likelihood

$$P\left(Y_n \,\middle|\, X_{n\,|\,n-1}^{(k)} \right)$$

Probability : Observable Y_n occurs when state is $X_{n\,|\,n-1}^{(k)}$

Filtering

$$X^{(k)}_{n\,|\,n-1}$$

$$X^{(k)}_{n\,|\,n}$$

Estimator at time n

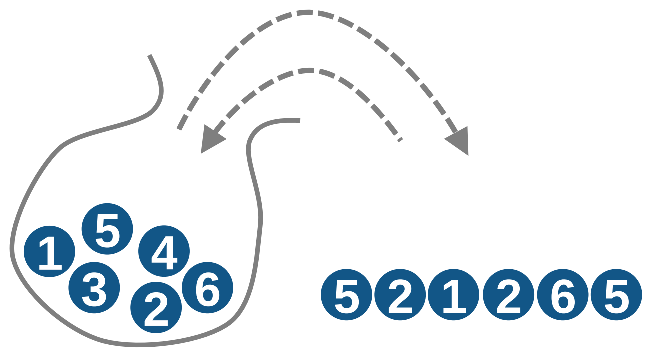
using information up to time (n-1)

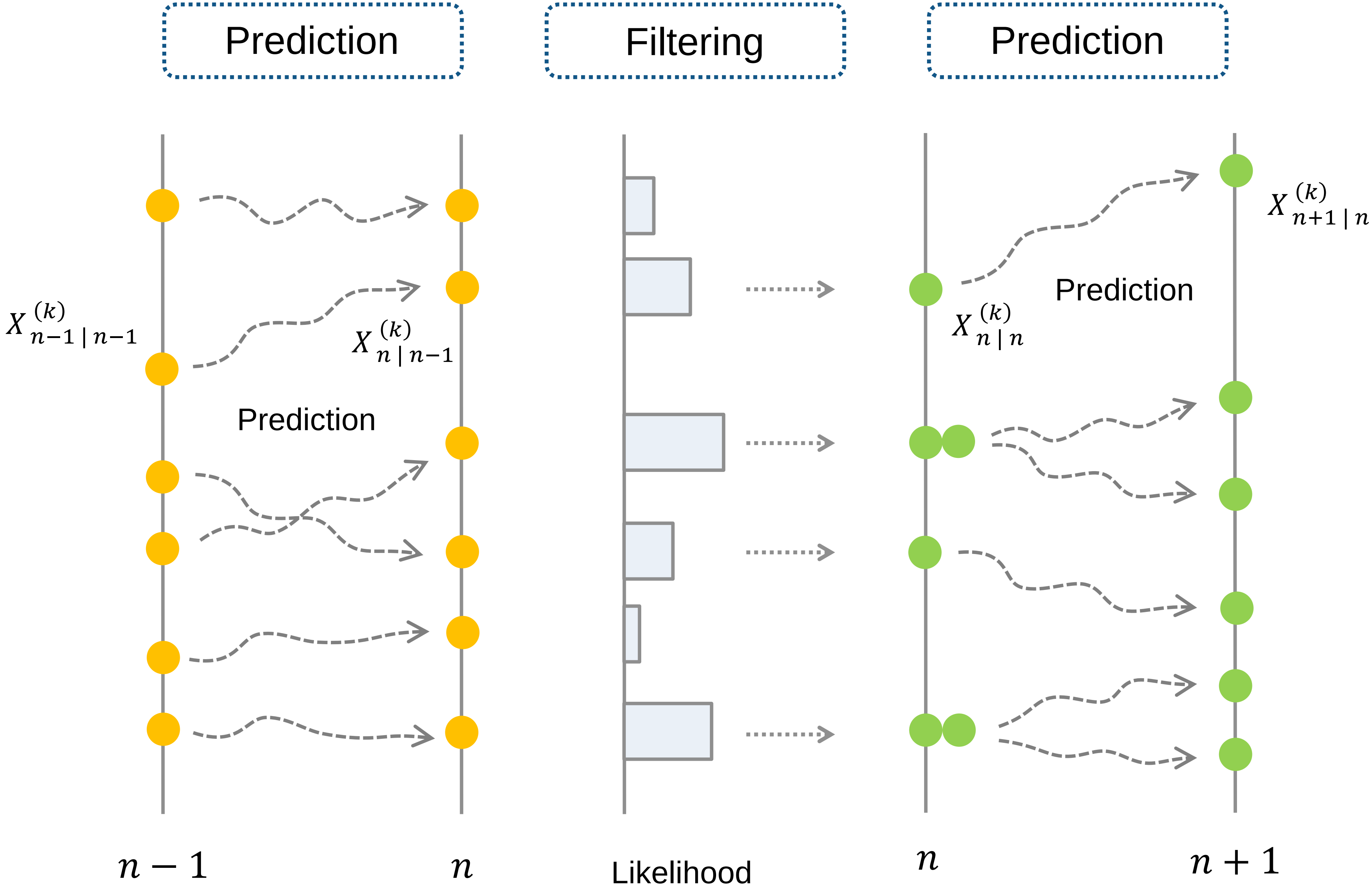
Estimator at time n

using information up to time n

Execute resampling with replacement according to likelihood ratio

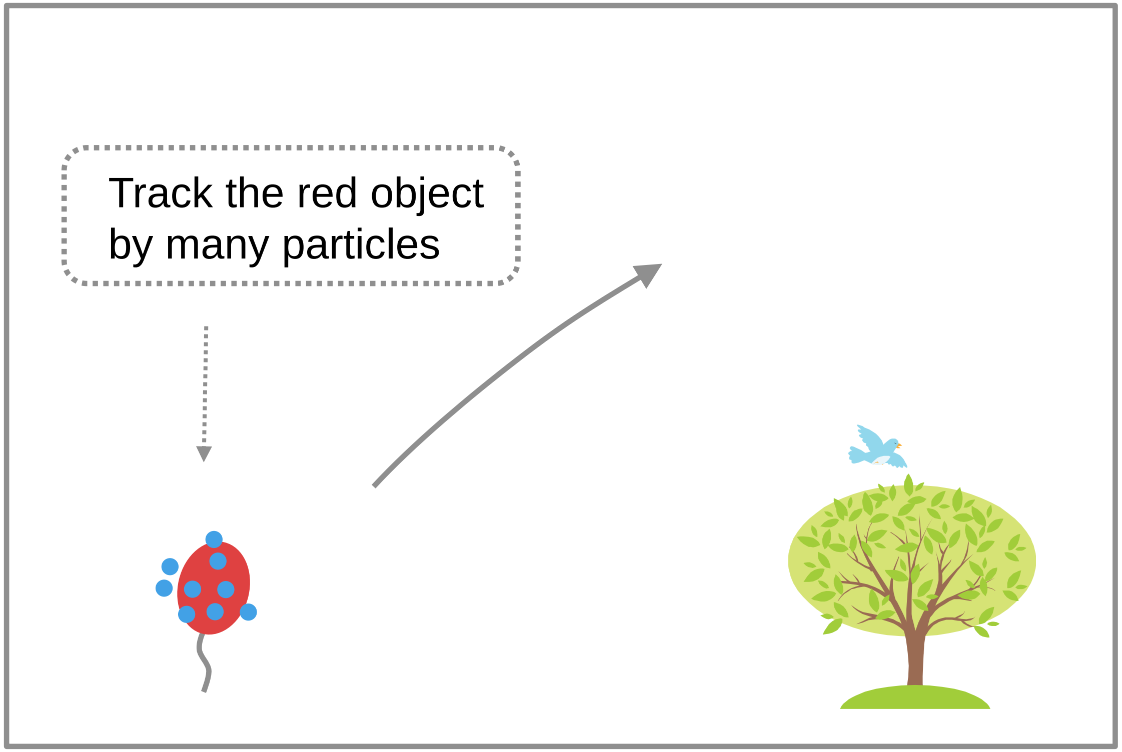
$$\frac{P\left(Y_n \,\middle|\, X^{(k)}_{n\,|\,n-1}\right)}{\sum_k P\left(Y_n \,\middle|\, X^{(k)}_{n\,|\,n-1}\right)} \quad (k = 1, \cdots, N)$$





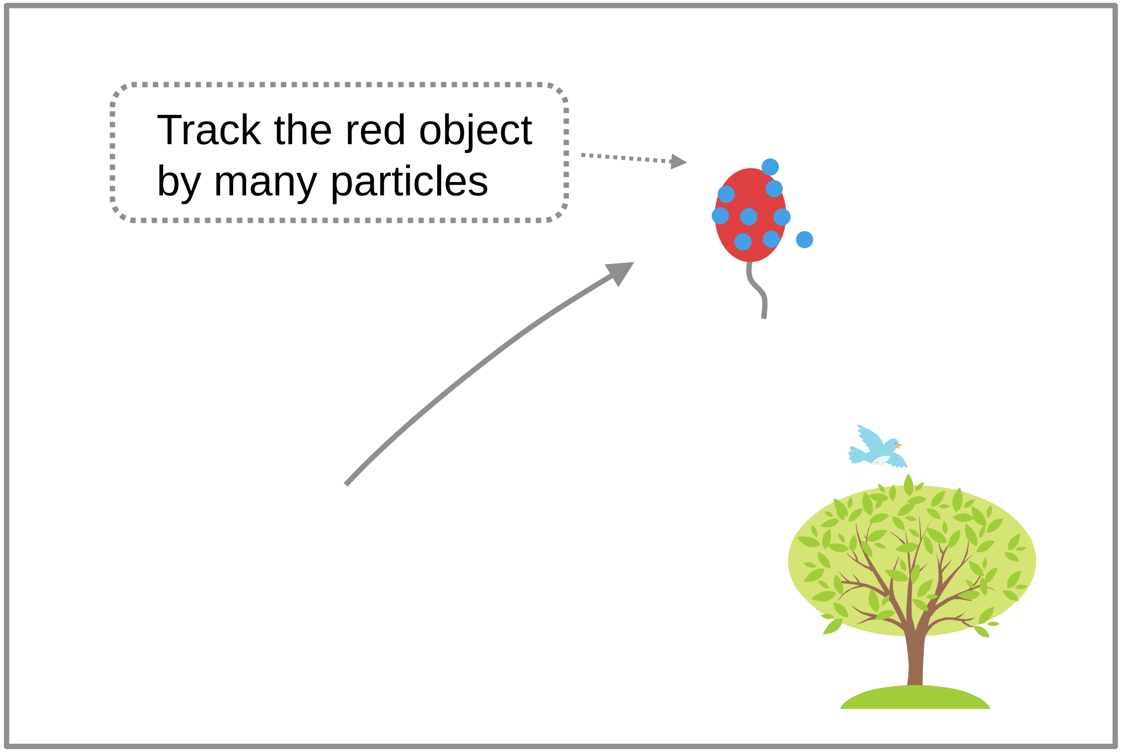
Prediction
Filtering
Prediction
$X^{(k)}_{n-1\,|\,n-1}$
$X^{(k)}_{n\,|\,n-1}$
Prediction
$X^{(k)}_{n\,|\,n}$
Prediction
$X^{(k)}_{n+1\,|\,n}$
$n-1$
n
Likelihood
n
$n+1$

Example of Particle Filter (Object tracking)



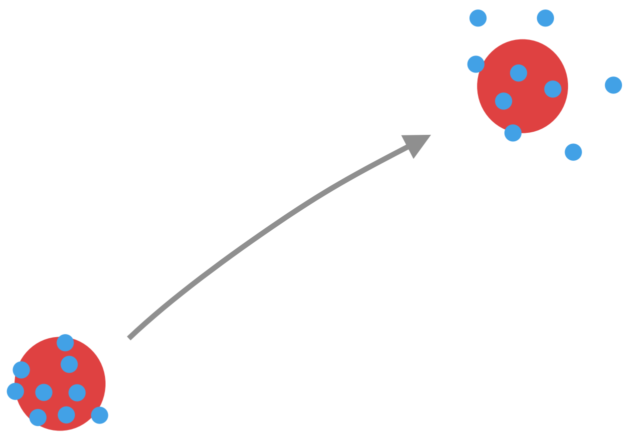
We are going to track the red object in the movie

Example of Particle Filter (Object tracking)



We are going to track the red object in the movie

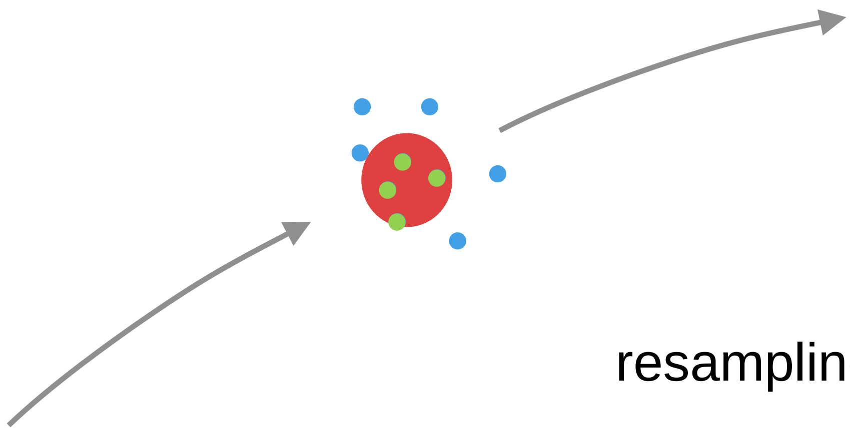
Prediction (move particles)



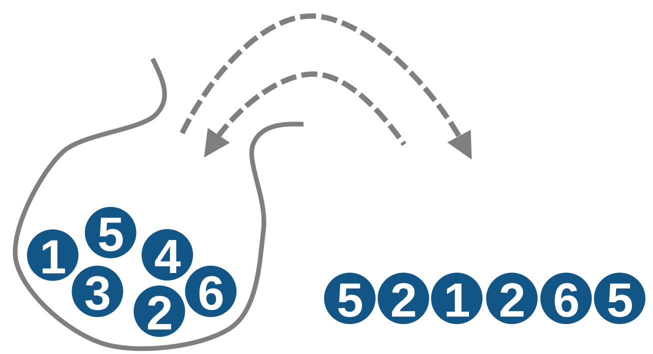
Location / Speed   Noise

$$\begin{pmatrix} x_n \\ y_n \\ \dot{x}_n \\ \dot{y}_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \dot{x}_{n-1} \\ \dot{y}_{n-1} \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_{\dot{x}} \\ v_{\dot{y}} \end{pmatrix}$$

Filtering (resample particles)

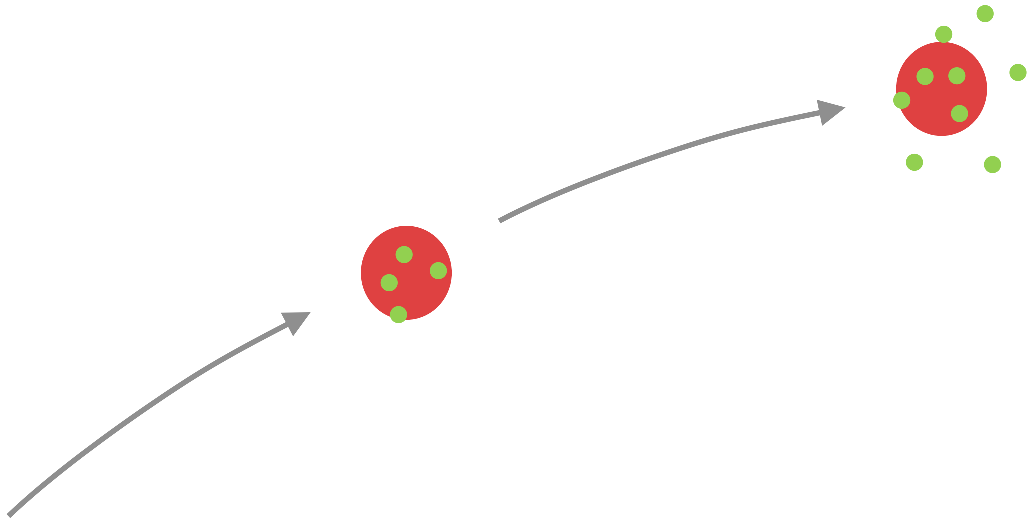


resampling with replacement



Particles with high “likelihood”
are more likely to be picked up

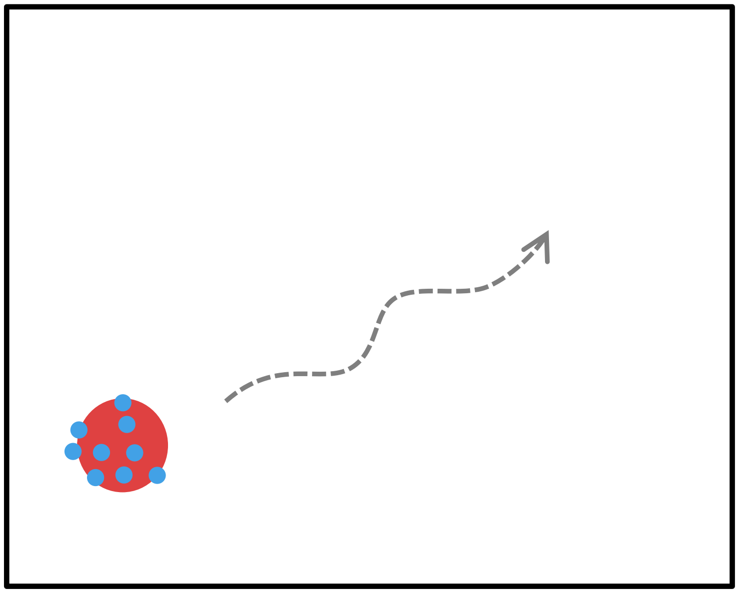
Prediction (move particles)



Location / Speed Noise

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \dot{x}_{n+1} \\ \dot{y}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ \dot{x}_n \\ \dot{y}_n \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_{\dot{x}} \\ v_{\dot{y}} \end{pmatrix}$$

State and Observables in Object Tracking



State

Location and speed of red object

$$X = (x, y, \dot{x}, \dot{y})$$

Observables

Color of pixel on which particles exist

$$Y = (r, g, b)$$

Calculation of “likelihood”

1) Get the color of pixel on which particle exists (observation)

Location of particle → Observe → Color of pixel

(x, y) (r, g, b)

2) Compare the RGB of pixel with red (255, 0, 0)

Color of pixel ↔ Compare ↔ Red

(r, g, b) $(255, 0, 0)$

Calculation of “likelihood”

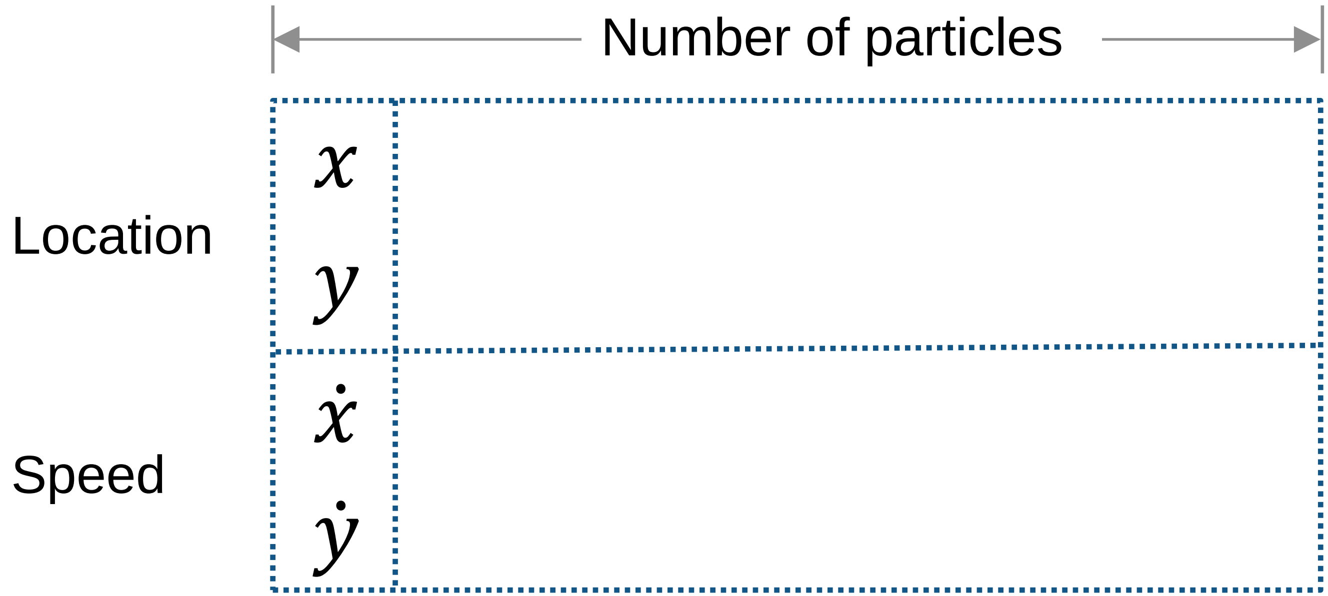
$$P\left(Y_n \,\middle|\, X_{n|n-1}^{(k)}\right) = \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp\left(-\frac{d^2}{2\sigma^2}\right)$$ ← Likelihood

$$d = \sqrt{(r-255)^2 + g^2 + b^2}$$

We are supposing gauss distribution for simplicity

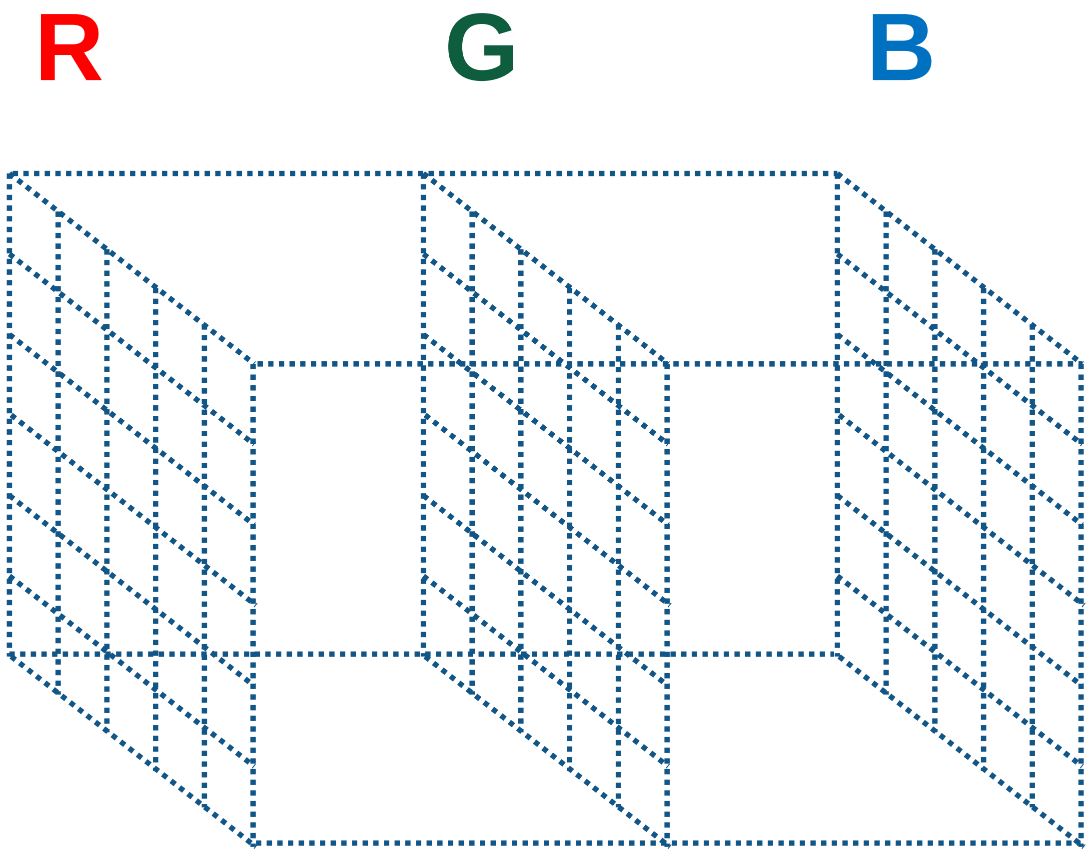
The closer the color we observe becomes red, the bigger the likelihood of particle becomes.

Explanation of variable (particles)



When you multiple matrix $F = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ from left, particles move

Explanation of variable (color photo)



480 X 640 X 3