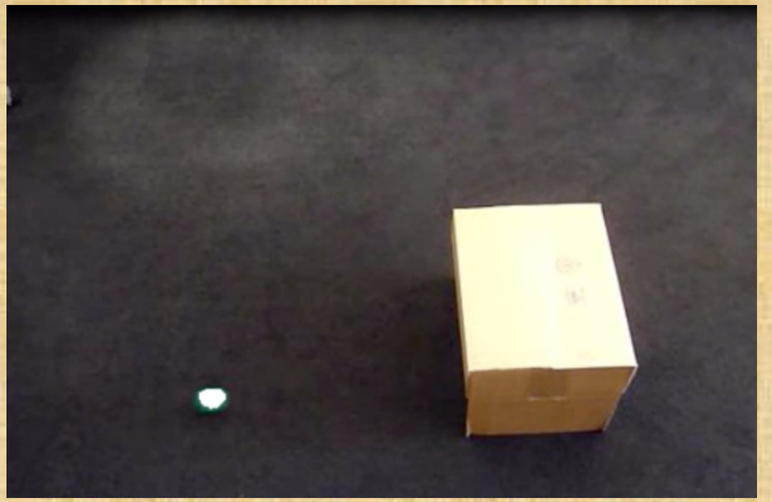
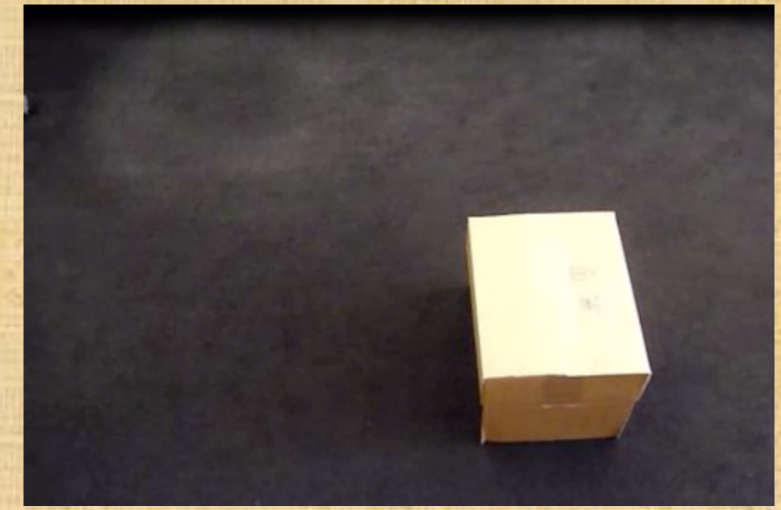
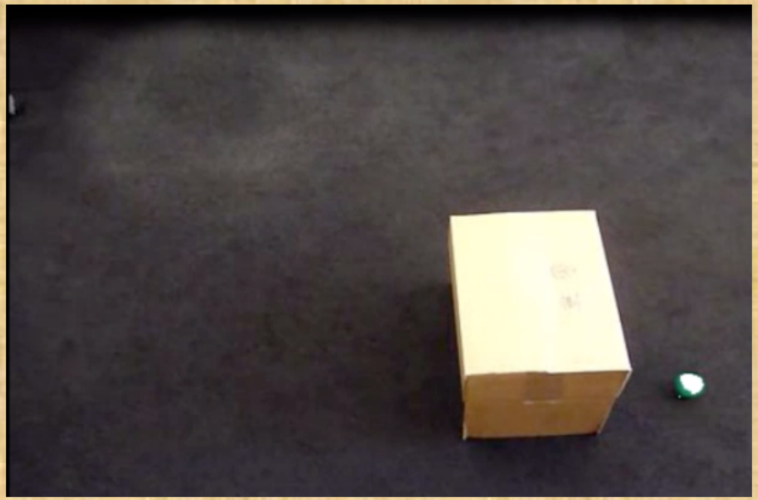
Kalman Filter

Computer Vision (CS0029)

One Tracking Approach

- You can detect objects at every frame independently
  - Ex: background subtraction
- But the it could be incorrect sometimes
- The object could be occluded sometimes

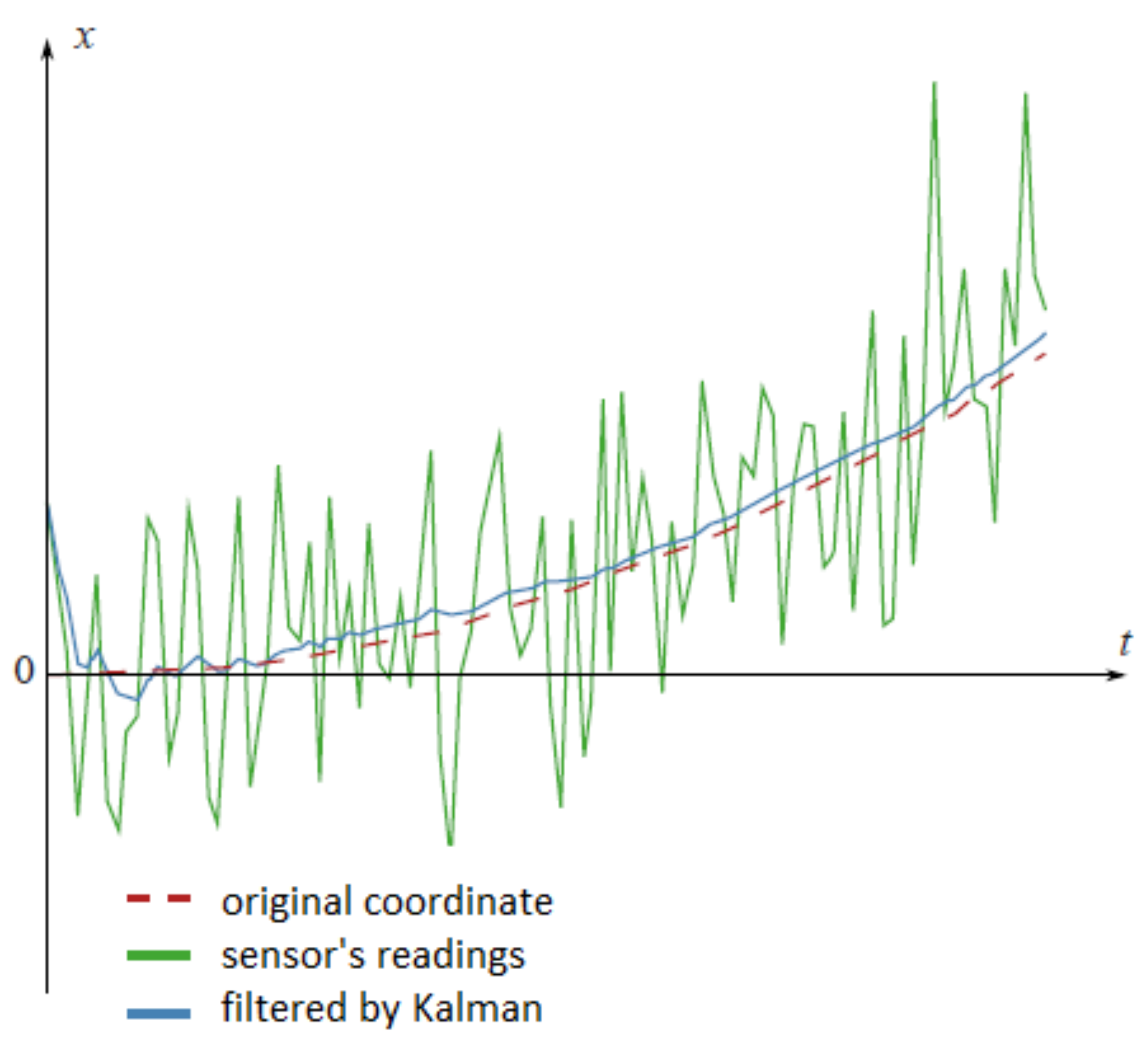
Tracking as induction

- Given corrected estimate for frame t:
  - Predict for frame t+1
  - Correct for frame t+1
- Recursive algorithm (Bayesian filter):

$$Bel(x_t) = p(z_t|x_t) \int p(x_t|u_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Why Filtering (Simple example)

- Sensor/measurement/tracking result from image always have errors
- Jitter could cause problem when high level inference is needed in the temporal domain



Prediction and Correction

- Prediction
- $P(X_t|y_0,\dots,y_{t-1}) = \int \underbrace{P(X_t|X_{t-1})}_{\textbf{Dynamics model}} \underbrace{P(X_{t-1}|y_0,\dots,y_{t-1})}_{\substack{\textbf{Corrected estimation} \\ \textbf{from previous step}}} dX_{t-1}$

- Correction
- $P(X_t|y_0,\dots,y_{t-1},\boldsymbol{y_t}) = \dfrac{\overbrace{P(y_t|X_t)}^{\textbf{Observation}} \overbrace{P(X_t|y_0,\dots,y_{t-1})}^{\textbf{Predicted estimation}}}{\int P(y_t|X_t)P(X_t|y_0,\dots,y_{t-1})dX_{t-1}}$

Kalman Filter

- A way to implement Bayes filter
- Linear Dynamics
  - Predictions are based upon a linear function of the current belief
- Assume Gaussian noise about how things might change and the measurement
  - After prediction and having a measurement, add Gaussian noise

Linear Dynamics Model

- State undergoes linear transformation plus Gaussian noise
    - $X_t \sim \mathrm{N}(D_t X_{t-1}, \Sigma_{d_t})$
    - X_t: state at time t and it is a Gaussian distribution
    - $D_t X_{t-1}$: mean of the state is the linear function of previous belief
        - D_t: a scalar or matrix (transformation)
        - Usually D_t is constant over time
    - Σ_{d_t}: Covariance matrix/Gaussian noise

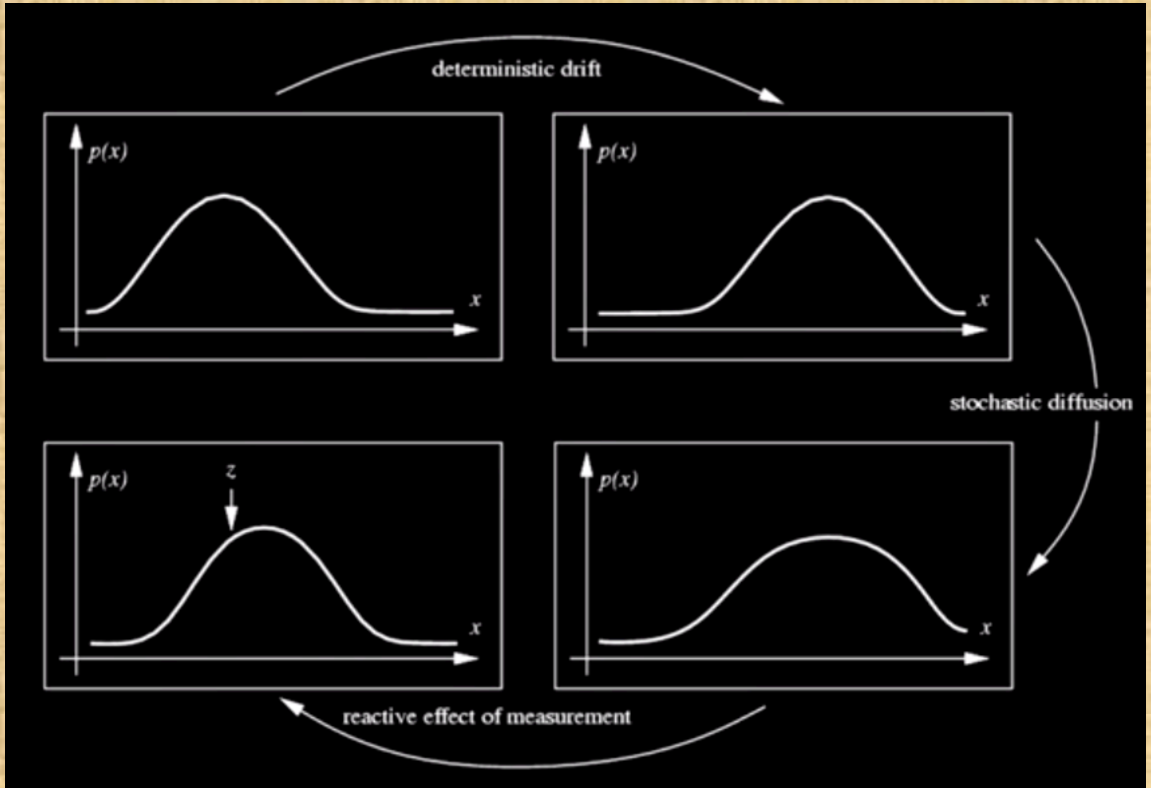
Linear Measurement Model

- Observation model: measurement is linearly transformed state plus noise
  - $y_t \sim \mathrm{N}(M_t X_t, \Sigma_{m_t})$
  - y_t: measurement is a Gaussian distribution
  - $M_t X_t$: mean of the measurement is the linear function of the state
    - M_t: a scalar or matrix (transformation)
  - Σ_{m_t}: Covariance matrix/Gaussian noise

- X_t: we do not know, hidden states

Kalman Filter

- Both prediction state and measurement state are represented by Gaussian
  - You only need to maintain them by means and covariances



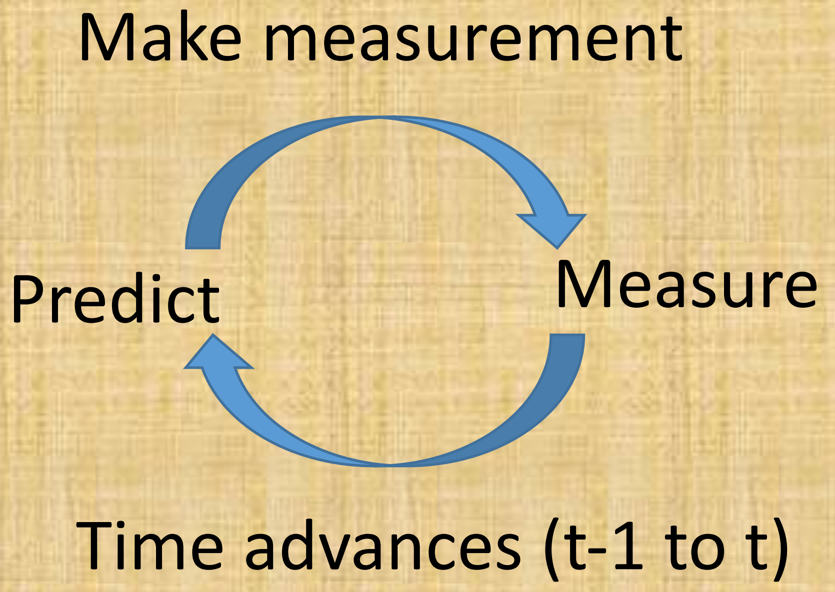
The Kalman Filter: 1D State

$P(X_t | y_0, \ldots, y_{t-1})$

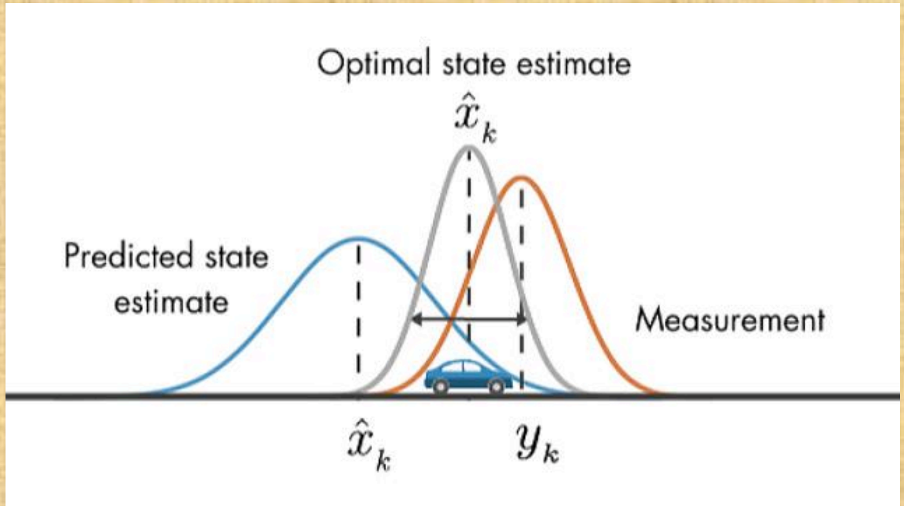
Mean and std. dev.
Of predicted state: μ_t^p, σ_t^p

$P(X_t | y_0, \ldots, y_t)$

Mean and std. dev.
Of predicted state: μ_t^c, σ_t^c



1D Karman Filter: Prediction



Linear dynamics model defines predicted state evolution with noise

$$X_t \sim \mathrm{N}(DX_{t-1}, \sigma_d^2)$$

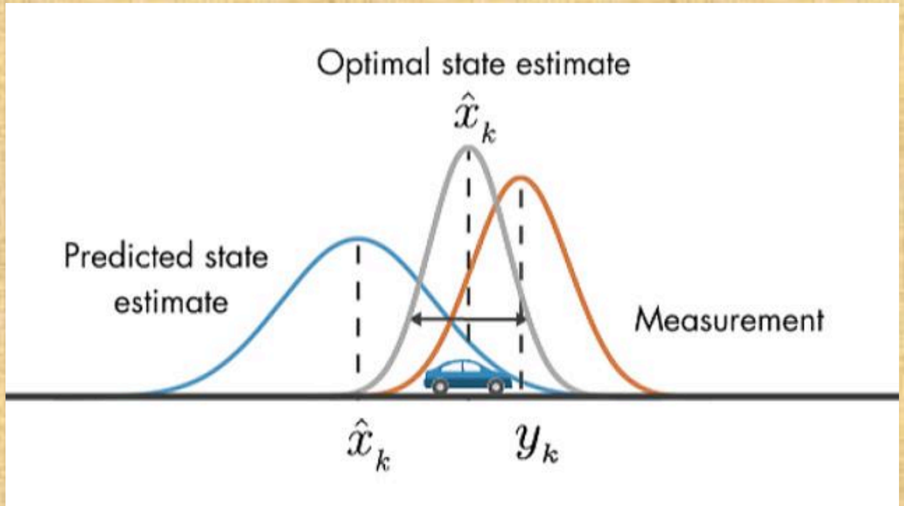
The distribution for next predicted state is also a Gaussian

$$P(X_t | y_0, \ldots, y_{t-1}) \sim \mathrm{N}(\mu_t^p, (\sigma_t^p)^2)$$

Predict the mean: $\mu_t^p = D\mu_{t-1}^c + Bu_{t-1}$

Predict the uncertainty :$(\sigma_t^p)^2 = \sigma_d^2 + D{\sigma_{t-1}^c}^2$

1D Karman Filter: Correction



With linear, Gaussian dynamics and measurements, the corrected distribution to be:

$$P(X_t|y_0, \ldots, y_t) = N(\mu_t^c, (\sigma_t^c)^2)$$

Update the mean:

1. calculate innovation: $\widehat{y_t} = y_t - \mu_t^p$

2. calculate Kalman gain, K_t: $(K_t)^2 = \dfrac{(\sigma_t^p)^2}{(\sigma_t^p)^2 + \sigma_m^2}$

3. update the mean: $\mu_t^c = \mu_t^p + K_t * \widehat{y_t}$

Update the uncertainty, σ_t^c: $(\sigma_t^c)^2 = (1 - K_t) * (\sigma_t^p)^2$

Example

Predict the mean: $\mu_t^p = D\mu_{t-1}^c$

Predict the uncertainty: $(\sigma_t^p)^2 = \sigma_d^2 + D{\sigma_{t-1}^c}^2$

Update the mean:

1. calculate innovation: $\widehat{y}_t = y_t - \mu_t^p$
2. calculate Kalman gain, K_t: $(K_t)^2 = \frac{(\sigma_t^p)^2}{(\sigma_t^p)^2 + \sigma_m^2}$
3. update the mean: $\mu_t^c = \mu_t^p + K_t * \widehat{y}_t$

Update the uncertainty, σ_t^c: $(\sigma_t^c)^2 = (1 - K_t) * (\sigma_t^p)^2$

- We want to estimate the temperature overtime
- Thermometer's uncertainty(standard deviation) is 4
- We assume the temperature does not change from t-1 to t, but the uncertainty (standard deviation) of this assumption is also 4
- ***If we want to estimate temperature at time t, we have estimated the temperature is 23 and the uncertainty (standard deviation) is 3. And the measurement at time t is 25.***

- Predict the mean:
  - D = 1
  - μ_{t-1}^c = 23
  - μ_t^p = 1*23=23
- Predict the uncertainty:
  - σ_d:4
  - σ_{t-1}^c: 3
  - $\sigma_t^p = \sqrt{4 + 1 * 3} = 5$

- Update the mean:
  - Innovation:
    - y_t = 25
    - μ_{t-1}^c = 23
    - $\widehat{y}_t$ = 25-23=2
  - Kalman gain:
    - σ_m: 4
    - $K_t = \sqrt{\frac{5^2}{5^2+4^2}} = 0.78$
  - Update the mean:
    - μ_t^c =23+0.78*2=24.56

- Update uncertainty:
  - $\sigma_t^c = \sqrt{(1 - 0.78) * 5^2}$=2.35

Now, we have the temperature estimation and its uncertainty at time t. The, if we get the temperature measurement of time $t + 1$ from the thermometer, we can calculate the temperature and uncertainty at $t + 1$ by the same process.

Practice

- We want to estimate the temperature overtime
- Thermometer's uncertainty(standard deviation) is 4 (σ_m)
- We assume the temperature does not change from t-1 to t, but the uncertainty (standard deviation) of this assumption is also 4 (σ_d)
- ***So, we have the temperature estimation 24.56(μ_t^c) and its uncertainty 2.35 (σ_t^c) at time t***
- ***We have temperature measurement at t+1 is 25.3***
- ***We have temperature measurement at t+2 is 25.5***
- ***What is the temperature estimation and uncertainty at t+2?***

Predict the mean: $\mu_t^p = D\mu_{t-1}^c$

Predict the uncertainty: $\left(\sigma_t^p\right)^2 = \sigma_d^2 + D{\sigma_{t-1}^c}^2$

Update the mean:

1. calculate innovation: $\hat{y}_t = y_t - \mu_t^p$
2. calculate Kalman gain, K_t: $(K_t)^2 = \frac{(\sigma_t^p)^2}{(\sigma_t^p)^2 + \sigma_m^2}$
3. update the mean: $\mu_t^c = \mu_t^p + K_t * \hat{y}_t$

Update the uncertainty, σ_t^c: $(\sigma_t^c)^2 = (1 - K_t) * (\sigma_t^p)^2$

N-Dimensional Kalman Filter

Predict (1-D)

$$\mu_t^p = D\mu_{t-1}^c$$

$$\left(\sigma_t^p\right)^2 = \sigma_d^2 + D{\sigma_{t-1}^c}^2$$

Predict (1-D)

$$\hat{y}_t = y_t - \mu_t^p$$

$$(K_t)^2 = \frac{(\sigma_t^p)^2}{(\sigma_t^p)^2 + \sigma_m^2}$$

$$\mu_t^c = \mu_t^p + K_t * \hat{y}_t$$

$$(\sigma_t^c)^2 = (1 - K_t) * (\sigma_t^p)^2$$

Predict (N-D)

$$X_t^p = DX_{t-1}^c$$

$$\Sigma_t^p = \Sigma_d + D_t \Sigma_{t-1}^c D_t^T$$

Predict (N-D)

$$\hat{y}_t = y_t - X_t^p$$

$$K_t = \Sigma_t^p \left(\Sigma_t^p + \Sigma_m\right)^{-1}$$

$$X_t^c = X_t^p + K_t\left(\hat{y}_t - X_t^p\right)$$

$$\Sigma_t^c = (I - K_t)\Sigma_t^p$$

Parameters

- Parameters you have to choose to use Kalman Filter
  - D: linear dynamics
  - σ_d: noise/error of the prediction
  - σ_m: noise/error of the measurement
  - μ : state vector

Predict (1-D)

$$\mu_t^p = D\mu_{t-1}^c$$

$$\left(\sigma_t^p\right)^2 = \sigma_d^2 + D{\sigma_{t-1}^c}^2$$

Predict (1-D)

$$\hat{y}_t = y_t - \mu_t^p$$

$$(K_t)^2 = \frac{(\sigma_t^p)^2}{(\sigma_t^p)^2 + \sigma_m^2}$$

$$\mu_t^c = \mu_t^p + K_t * \hat{y}_t$$

$$(\sigma_t^c)^2 = (1 - K_t) * (\sigma_t^p)^2$$

Parameters

- Temperature example
  - $\mu = [t]$
  - $D = [1]$

- A truck moves on a straight rails.
  - $\mu = \begin{bmatrix} x \\ v \end{bmatrix}$
  - $D = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$

- If the truck move on a 2D plane

Tracking with Kalman Filters

- Background subtraction and Kalman Filter

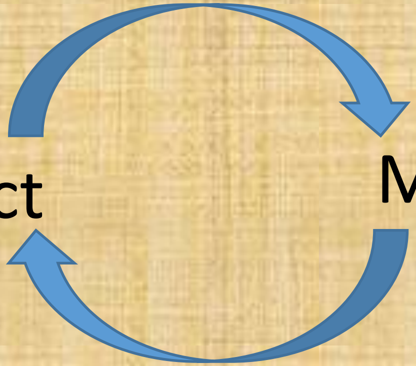
- https://www.youtube.com/watch?v=MxwVwCuBEDA

Summary

- Kalman filter
  - Represent the state using Gaussian distribution (mean is the estimation and standard deviation is the uncertainty)
  - Predict the state at time t+1 from state at time t by linear dynamics assumption
  - Combine the prediction with uncertainty at time t+1 and measurement with uncertainty at time t+1 to estimate the state at time t+1

Make measurement

Predict



Measure