CS231A CA Session
PSet4 Review

Congyue Deng
2024/05/24

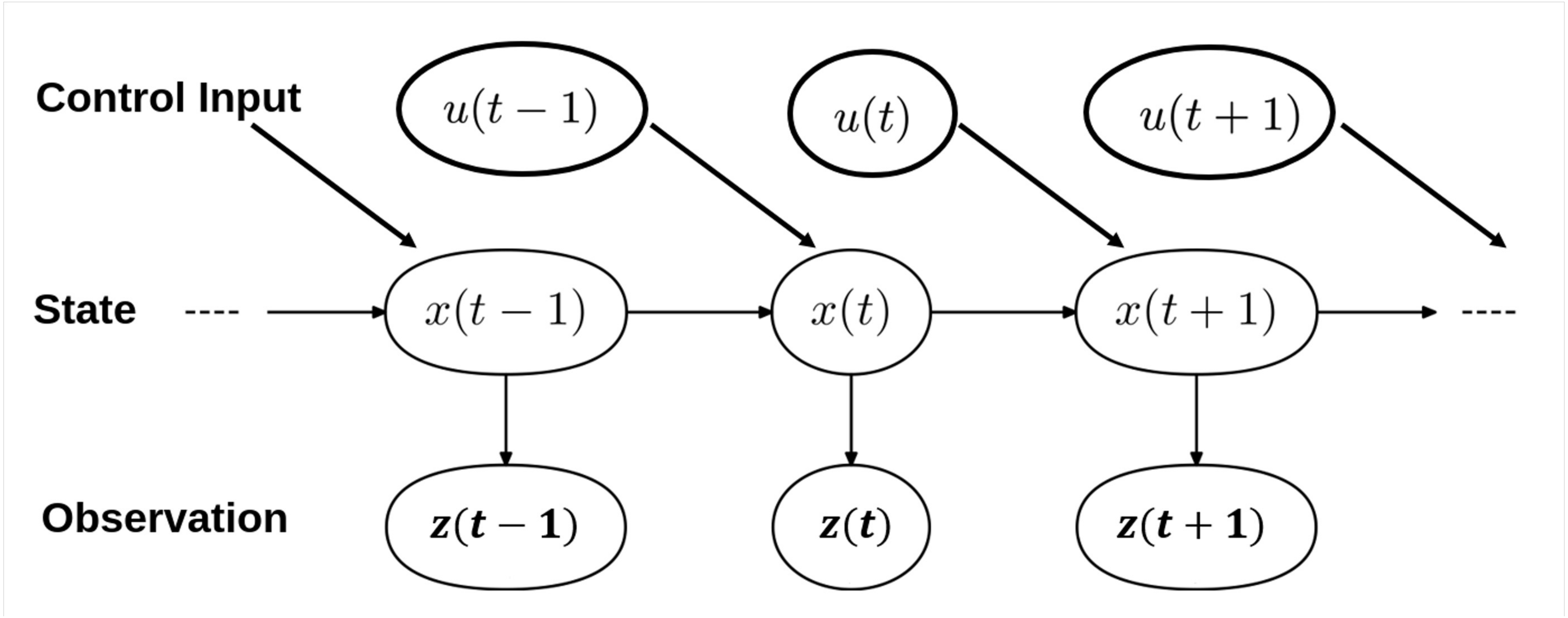
Outline

- Extended Kalman Filter
- Monocular depth estimation

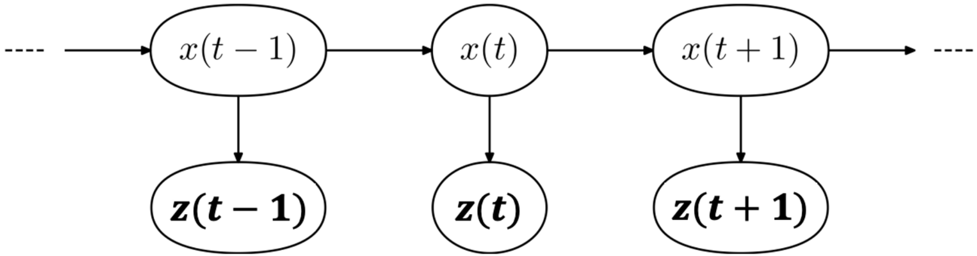
Outline

- **Extended Kalman Filter**
- Monocular depth estimation

Dynamical System



Notation



x State of dynamical system, dim n

x_t Instantiation of system state at time t

z Sensor Observation Vector, dim k

z_t Specific Observation at time t

u Robot action / control input, dim m

u_t Robot action / control input at time t

$p(x_t | z_{0:t}, u_{0:t})$ Probability distribution

Markov Assumption

State is complete $p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$

Kalman Filter

An algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.

Source: Wikipedia

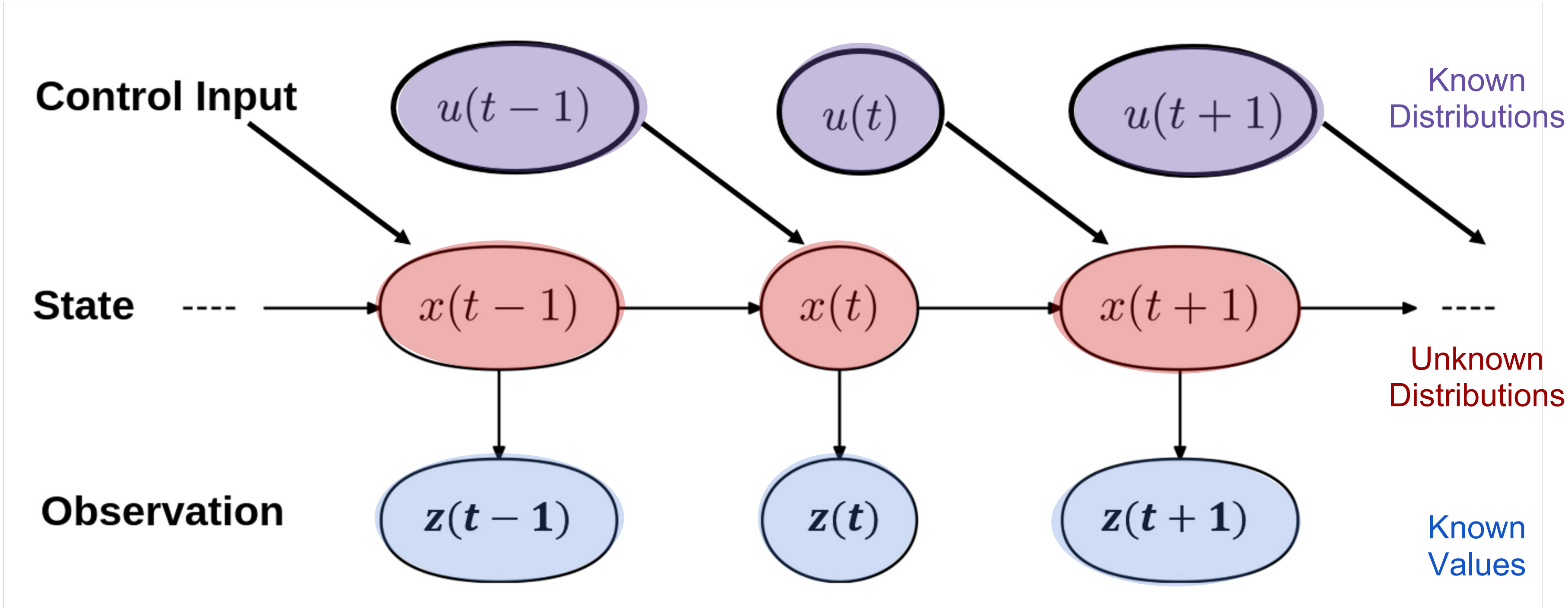
Kalman Filter

An algorithm that uses a series of **measurements observed over time**, containing **statistical noise** and other inaccuracies, and produces **estimates of unknown variables** that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.

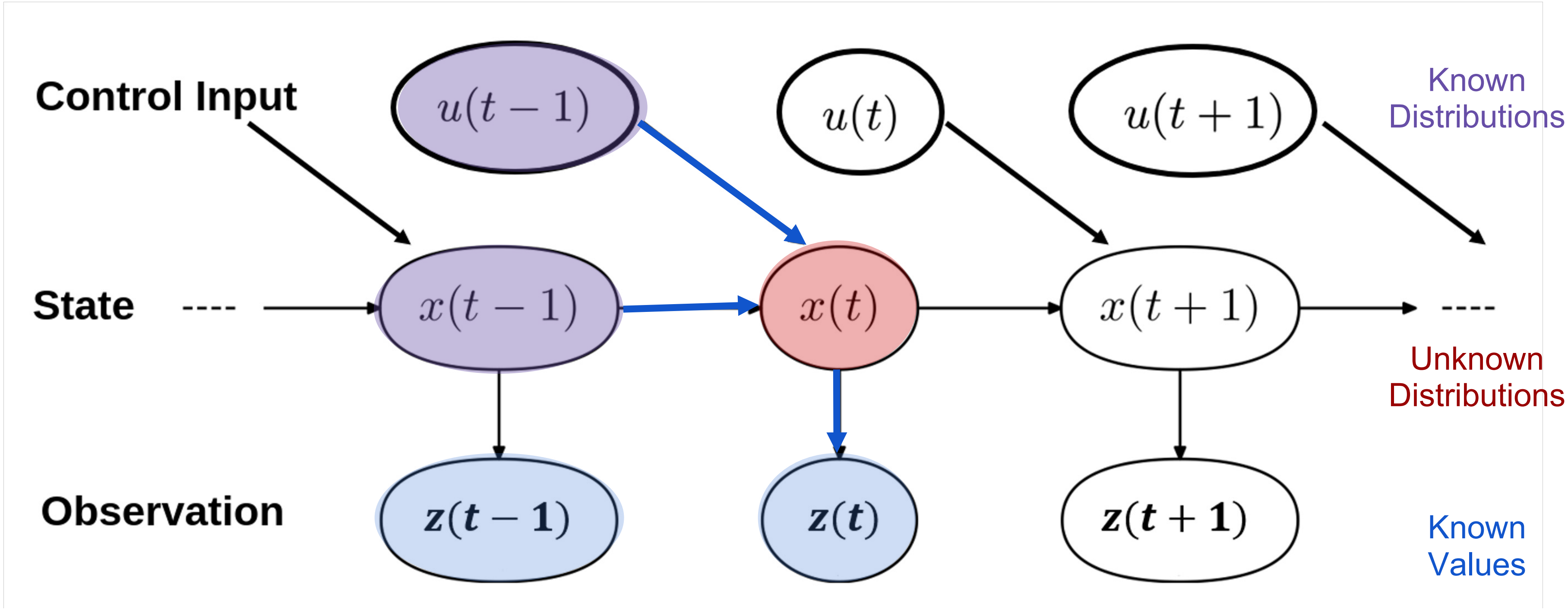
Source: Wikipedia

To make it even more illustrative ->

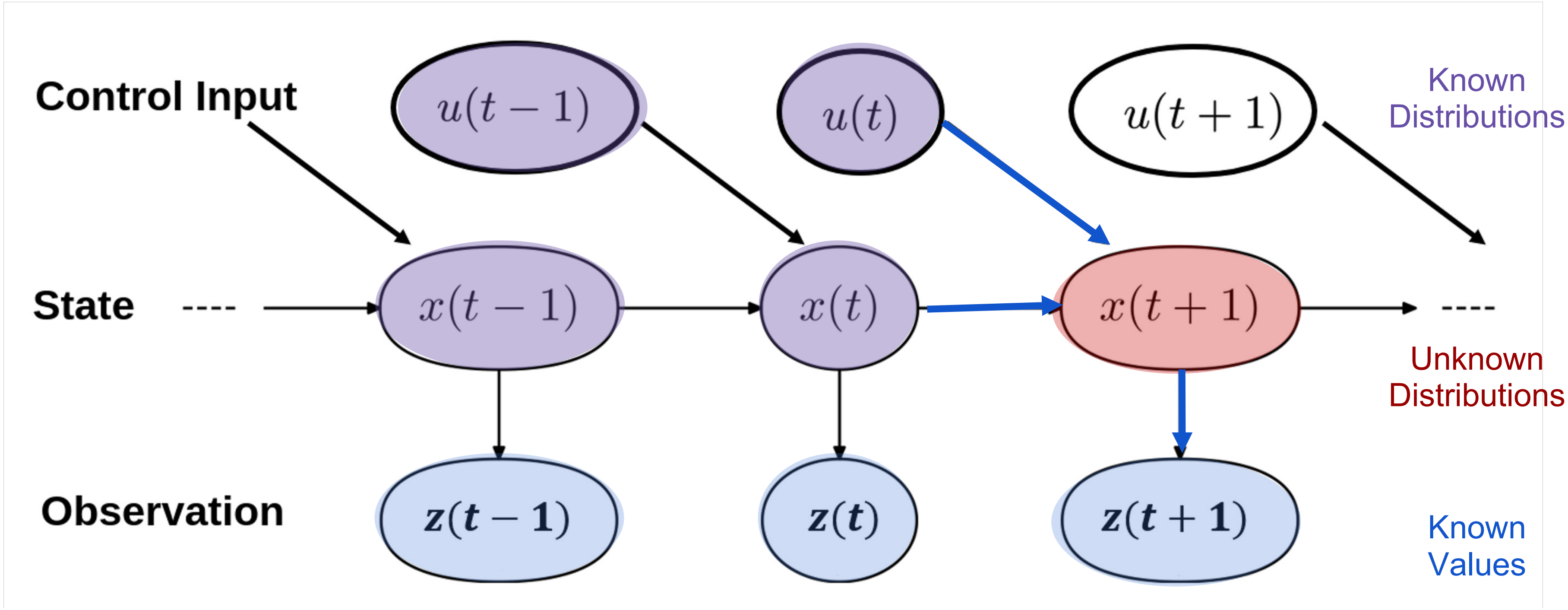
What does Kalman Filter do?



What does Kalman Filter do?



What does Kalman Filter do?



Extended Kalman Filter

- Extended Kalman filter (EKF) is heuristic for nonlinear filtering problem.
- Often works well (when tuned properly), but sometimes not.
- Widely used in practice.

Based on

- Linearizing dynamics and output functions at current estimate.
- Propagating an approximation of the conditional expectation and covariance.

Source: EE363

Extended Kalman Filter

- Extended Kalman filter (EKF) is heuristic for **nonlinear** filtering problem.
- Often works well (when tuned properly), but sometimes not.
- **Widely used in practice**.

Based on

- **Linearizing dynamics** and output functions at current estimate.
- Propagating an approximation of the conditional expectation and covariance.

Source: EE363

Implementing Extended Kalman Filter

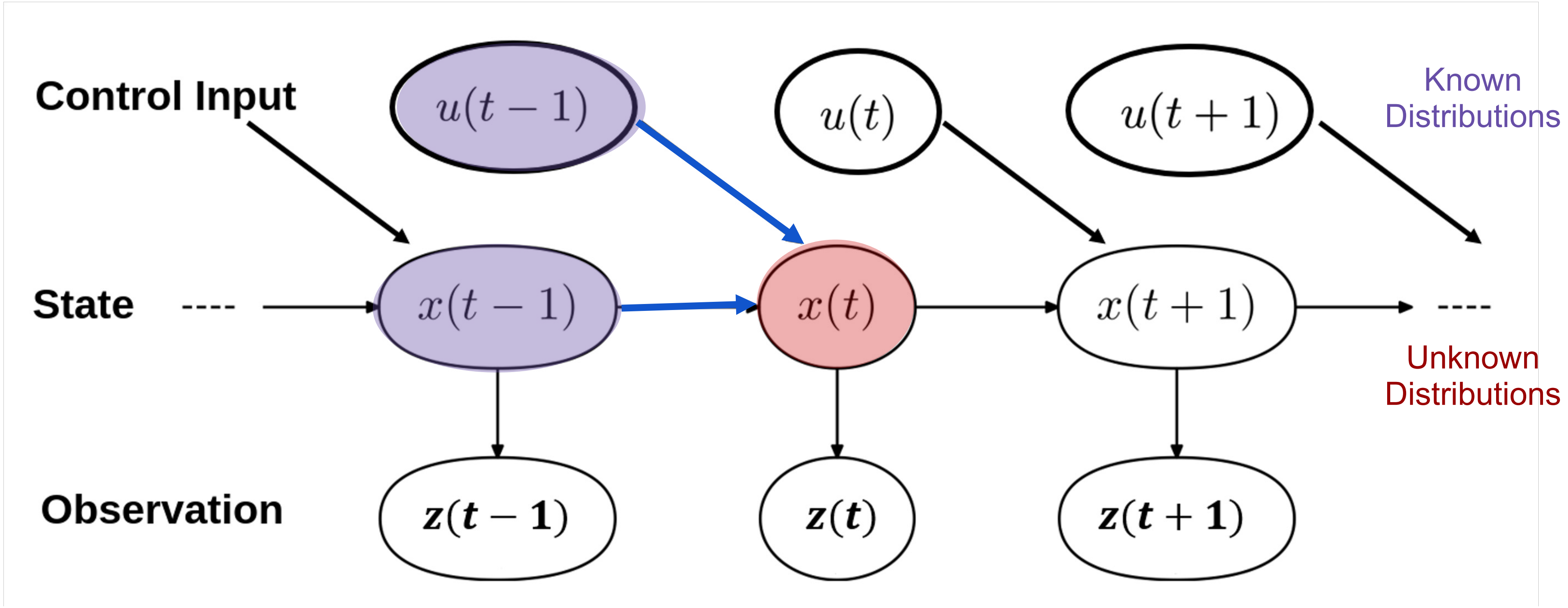
- Define the state, the control, and the noise
- Derive the system and the observation
- Compute the current Jacobian matrix (*linearizing dynamics*)
- Compute the distribution of the current state
- Iterate this process across time

Define the State

$$x_t = \begin{bmatrix} p_t^x \\ p_t^y \\ p_t^z \\ v_t^x \\ v_t^y \\ v_t^z \end{bmatrix}$$

State: 6-dimensional vector (position, velocity)

Define the System Matrix

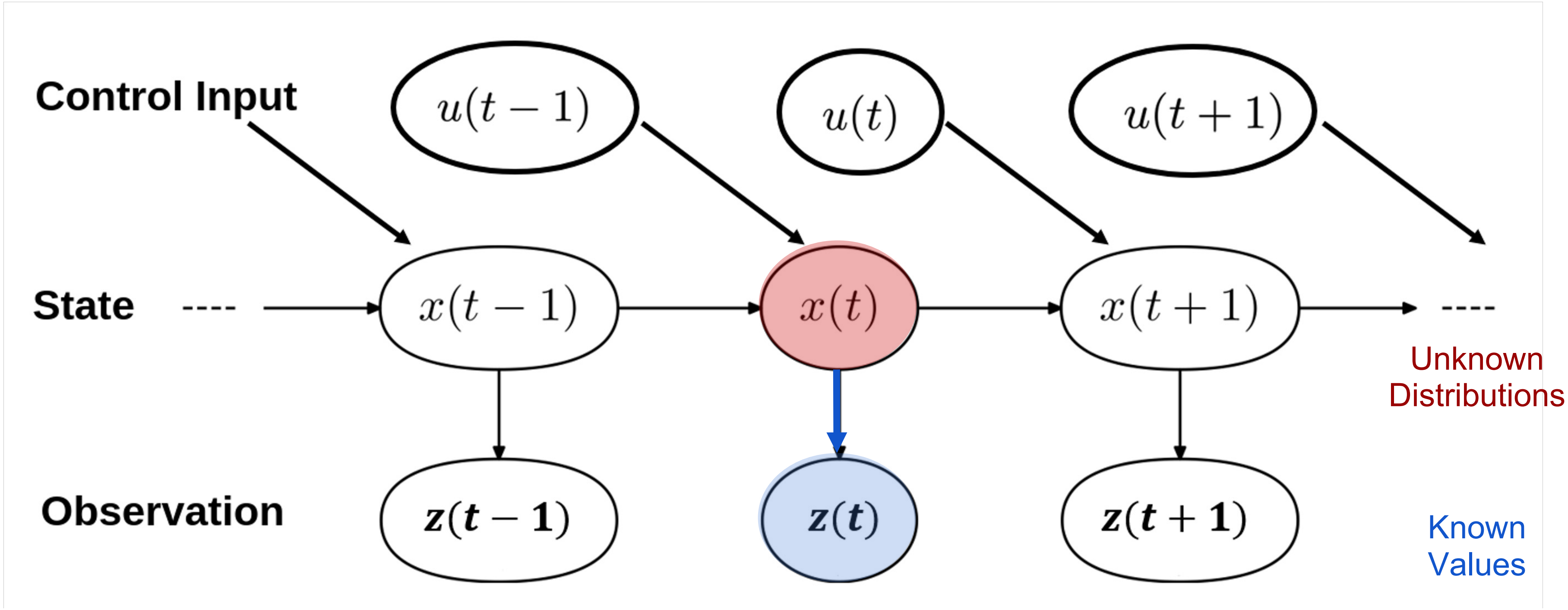


Define the System Matrix

$$x_t = \begin{bmatrix} p_t^x \\ p_t^y \\ p_t^z \\ v_t^x \\ v_t^y \\ v_t^z \end{bmatrix}$$

$$x_{t+1} = Ax_t + \epsilon_t$$

Define the Observation



Define the Observation

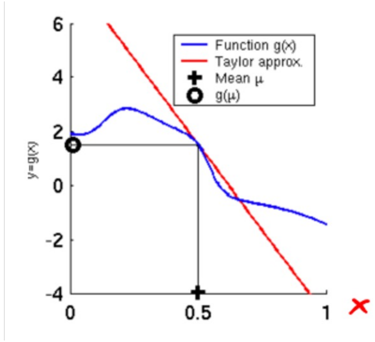
$$z_t = h(x_t) + v_t$$

Observation in Q2: 2-dimensional vector (pixel location)

Observation in Q3: 3-dimensional vector (pixel location, disparity)

$h(x_t)$ can be derived using the camera model we learned from previous lectures.

Computing the Jacobian

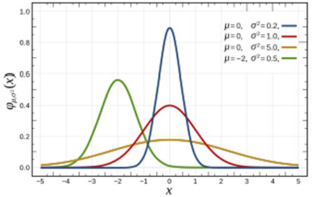


$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Source: Wikipedia

The Kalman Filter

Kalman Filter



- Initial Belief $\mathbf{x_0} \sim N(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$

$$bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}(x_0-\mu_0)^T\Sigma_0^{-1}(x_0-\mu_0)\}$$

- Distribution over next state

$$p(x_t \mid u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\}$$

Process Noise

Transition Model

- Likelihood of Measurement

Measurement Noise

$$p(z_t \mid x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}(z_t - C_t\, x_t)^T Q_t^{-1}(z_t - C_t\, x_t)\}$$

Measurement Model

The Kalman Filter

The Kalman Filter Algorithm

Algorithm Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

$\bar{\mu}_t = A_t \, \mu_{t-1} + B_t \, u_t$

$\bar{\Sigma}_t = A_t \, \Sigma_{t-1} \, A_t^T + R_t$

$K_t = \bar{\Sigma}_t \, C_t^T (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \, \bar{\mu}_t)$

$\Sigma_t = (I - K_t \, C_t) \, \bar{\Sigma}_t$

return μ_t, Σ_t

Uncertainty increases

K = Kalman Gain $K \approx \frac{R}{Q}$

Uncertainty decreases

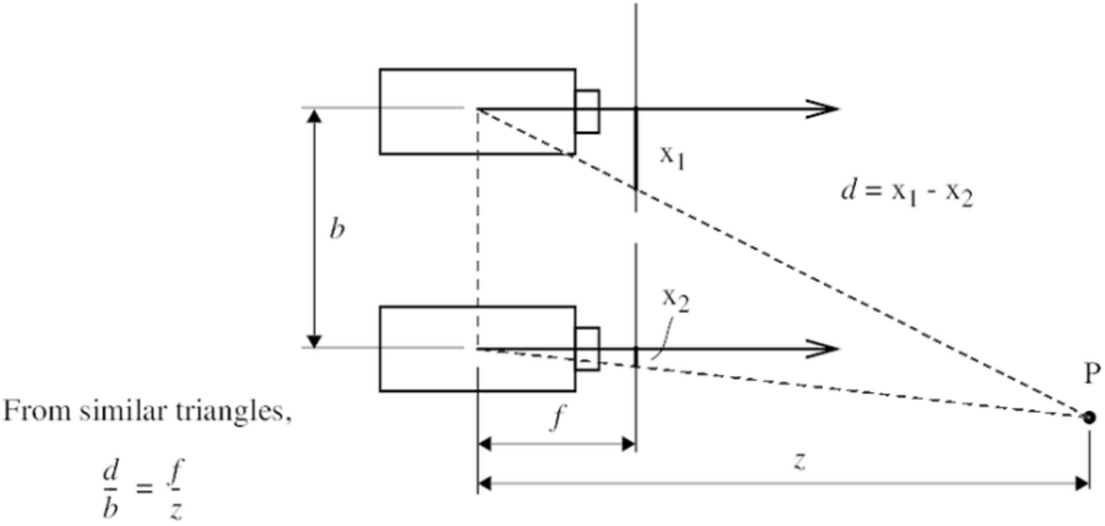
If R large, then K is large. Update dominated by innovation.

If Q large, then K is small. Update dominated by prediction.

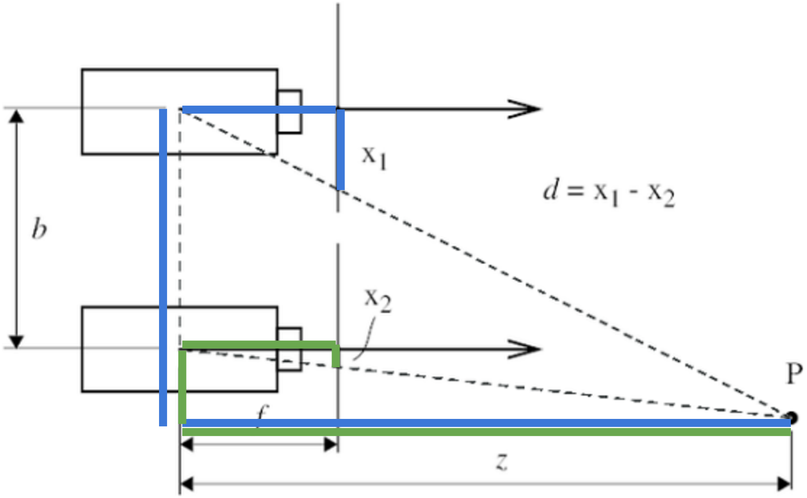
Outline

- Extended Kalman Filter
- **Monocular depth estimation**

Disparity inverse proportional to depth



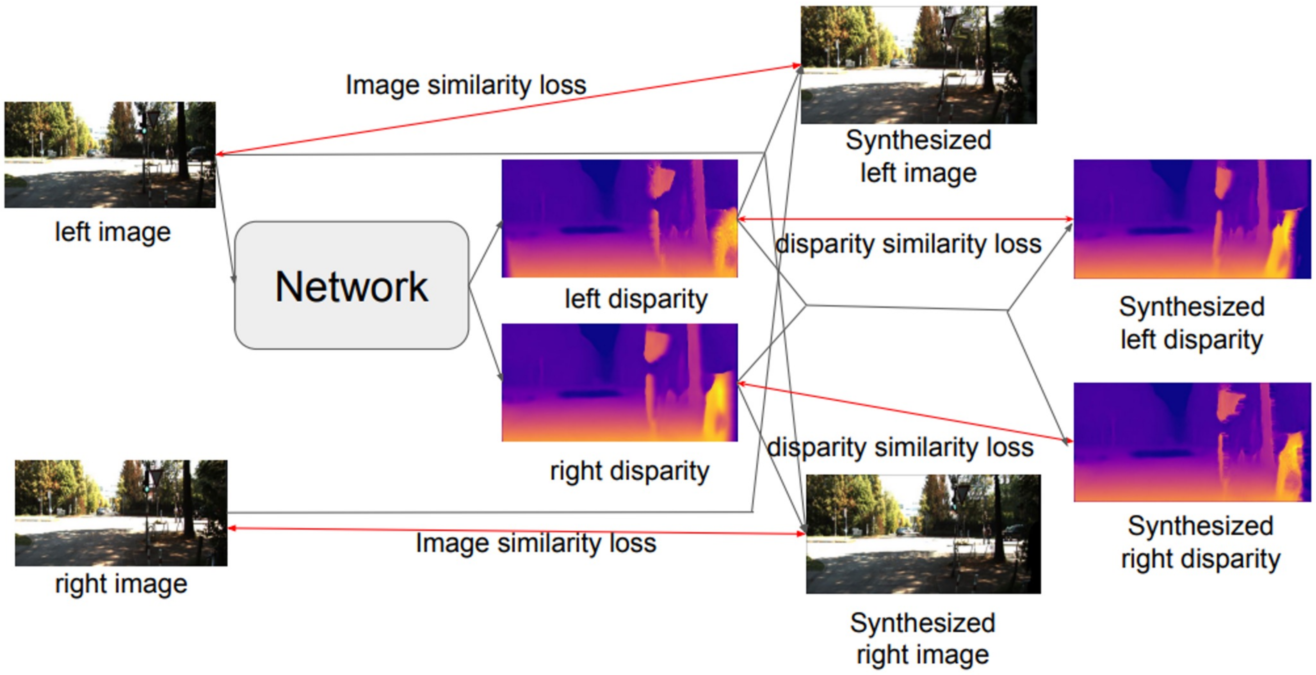
Disparity inverse proportional to depth



From similar triangles,

$$\frac{d}{b} = \frac{f}{z}$$

Unsupervised monocular depth estimation



Unsupervised monocular depth estimation

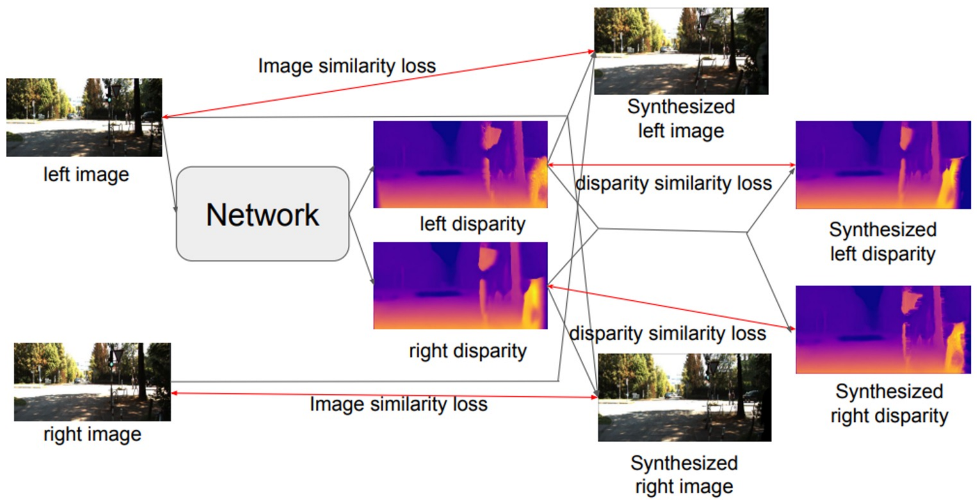


Image prediction:

$$img'_l = \text{generate\_image\_left}(img_r, disp_l)$$
$$img'_r = \text{generate\_image\_right}(img_l, disp_r)$$

Disparity prediction:

$$disp'_l = \text{generate\_image\_left}(disp_r, disp_l)$$
$$disp'_r = \text{generate\_image\_right}(disp_l, disp_r)$$

Unsupervised monocular depth estimation

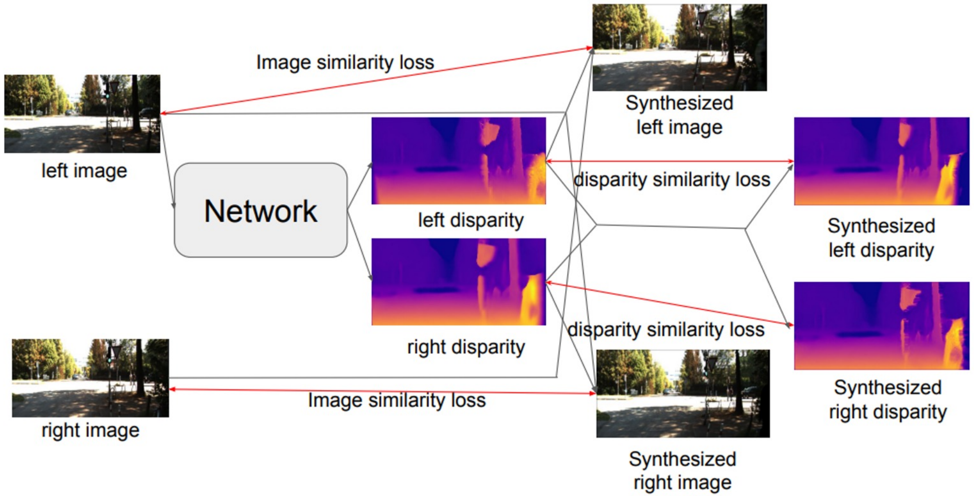


Image prediction:

$$L_{img} = \text{compare}_i(img'_l, img_l) + \text{compare}_i(img'_r, img_r)$$

Disparity loss

$$L_{disp} = \text{compare}_d(disp'_l, disp_l) + \text{compare}(disp'_r, disp_r)$$

Thank you!