Master in Computer Vision Barcelona

**Module:** Video Analysis

**Lecture 5:** Bayesian tracking (I)

**Lecturer:** Ramon Morros

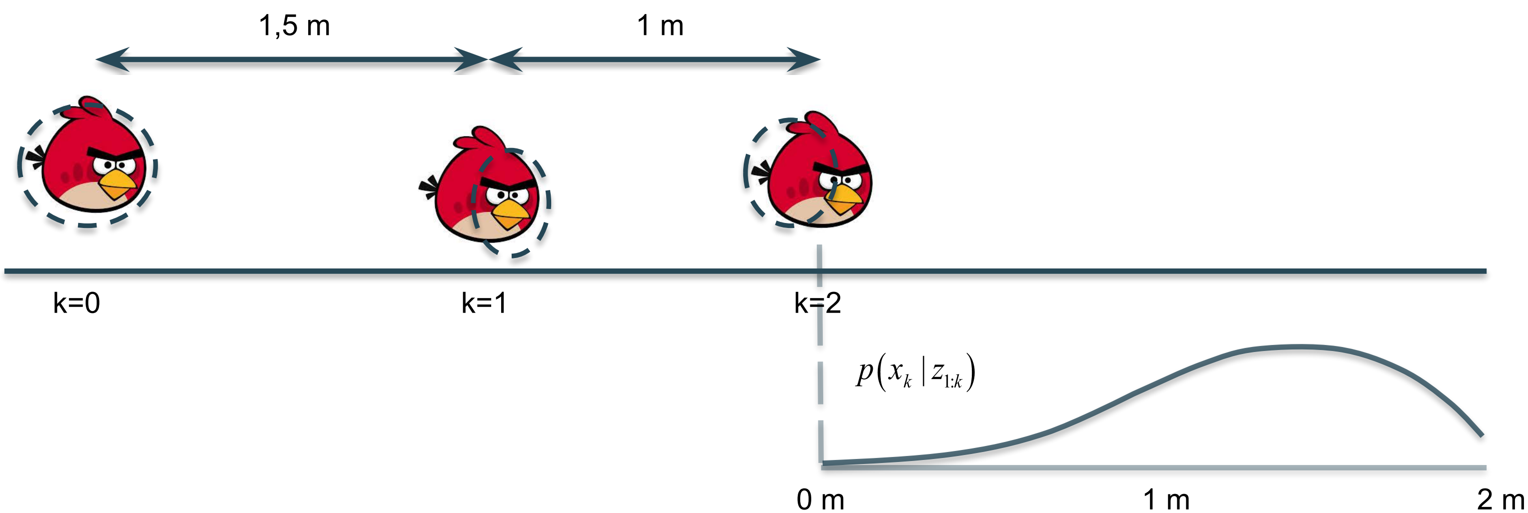
# Overview

- Introduction to the tracking problem
- Linear Dynamic Models
- Kalman Filter
- Particle Filters

# Tracking

## Bayesian estimation

- **Goal:** finding an **object position** over time in a video sequence.



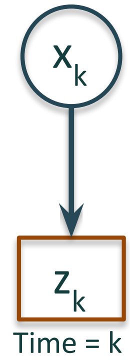

- **How to estimate this function iteratively?**
  - Kalman Filter
  - Particle Filter

# Tracking

## Observations and states

- **State (x):** true parameters that characterize the object being tracked
  - Position, velocity, etc.
  - Usually "hidden" or unknown

- **Measurement/observation (z)**: what we can measure on the image at a given time
  - Blob centroids, bounding boxes, etc.
  - Observations result from underlying states
  - Subject to noise



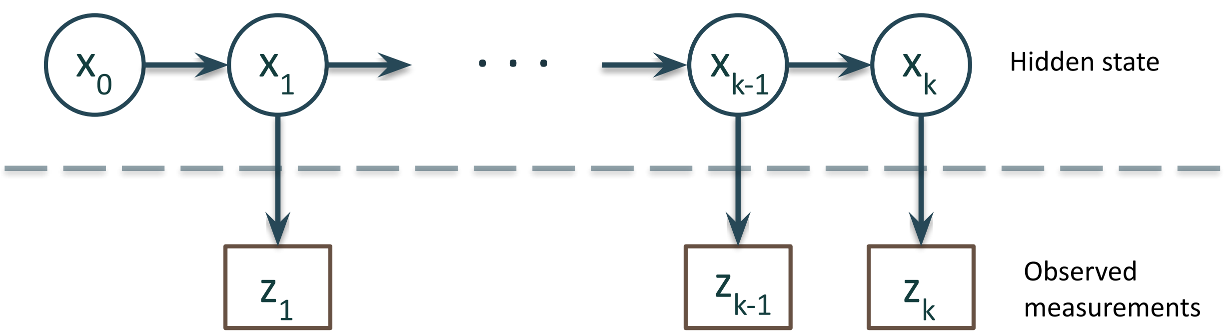
# Tracking

## Tracking as inference

- At each time step, **object state changes** (from $x_{k-1}$ to $x_k$ ) and we get a new observation $z_k$

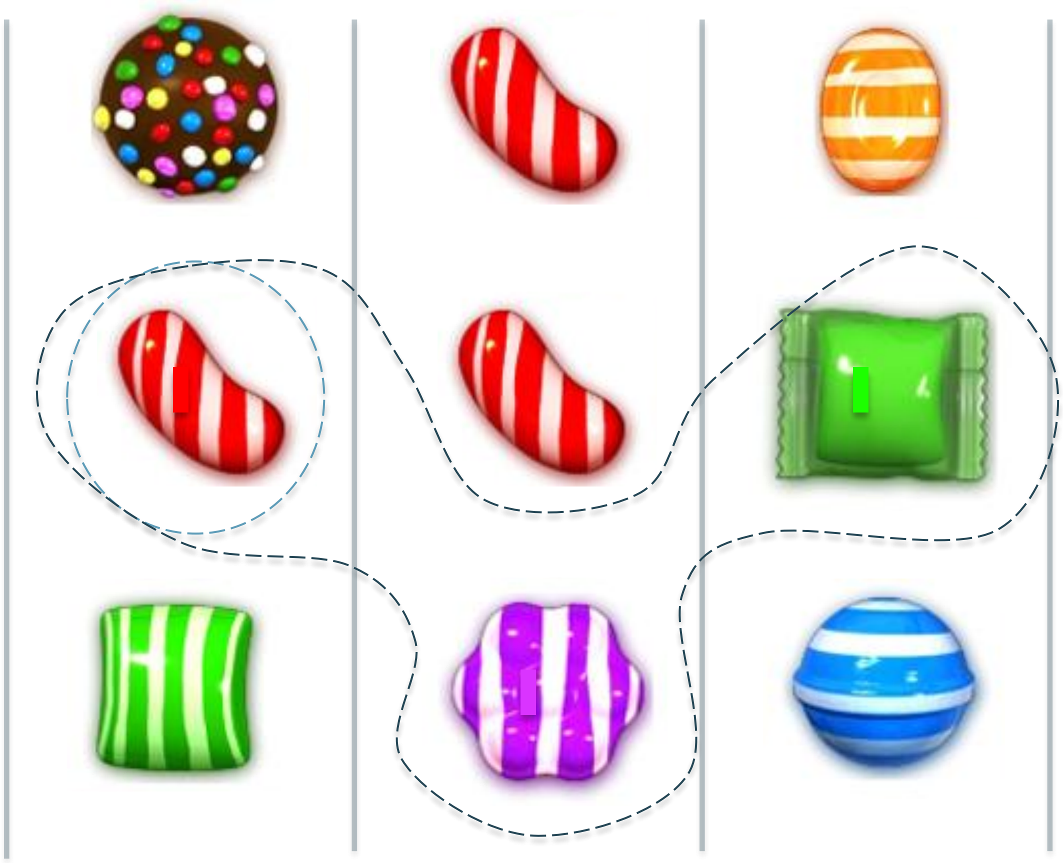
(k indicates discrete time)

- **Goal**: recover most likely state $x_k$ given:
  - All observations seen so far ($z_1$, …, $z_k$)
  - Knowledge about dynamics of state transitions.



# Tracking

## Steps of tracking



# Tracking

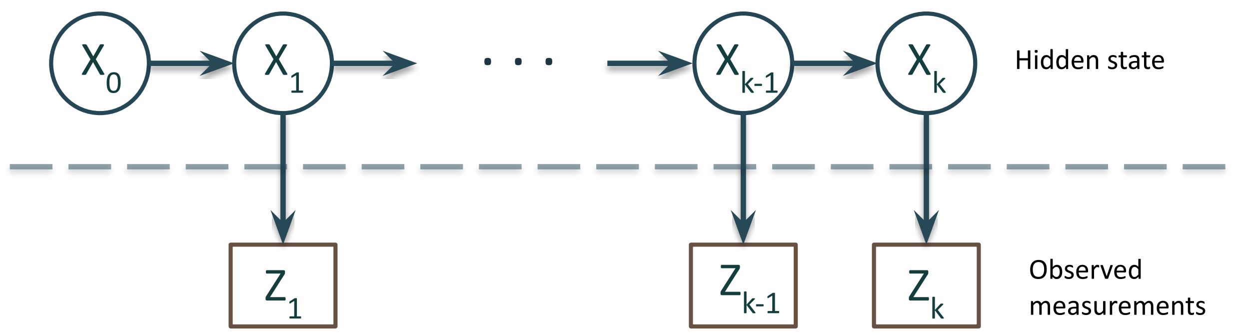
## Steps of tracking

- **Prediction:** What is the current state of the object given past measurements?

$$p(x_k \mid z_1, \ldots, z_{k-1}) = p(x_k \mid z_{1:k-1})$$

- **Update:** Compute an updated estimate of the state from prediction and measurements.

$$p(x_k \mid z_1, \ldots, z_{k-1}, z_k) = p(x_k \mid z_{1:k})$$



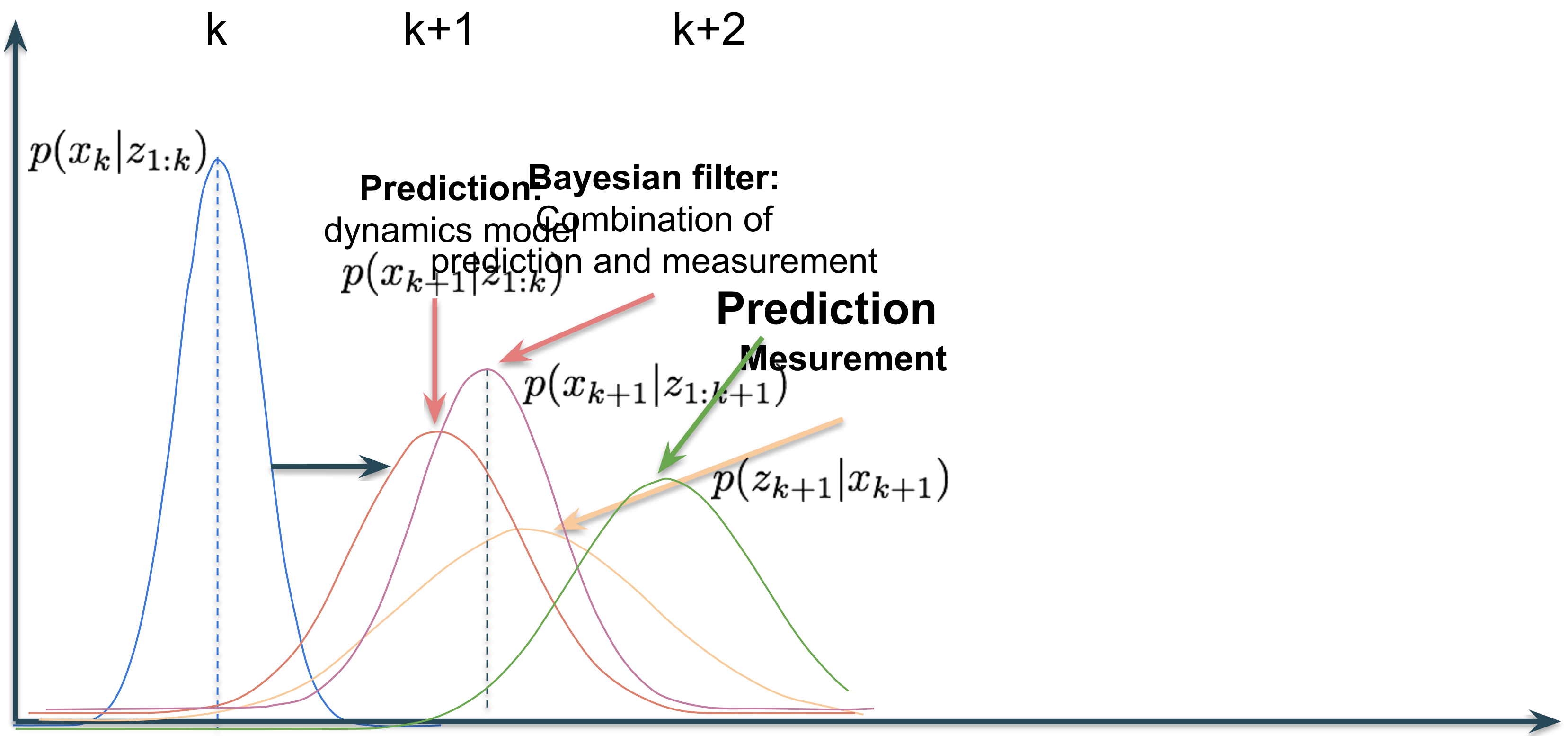
# Tracking

## Questions

- How to represent the dynamics model that govern the changes in the states?
- How to represent the relationship between state and measurements, plus our uncertainty in the measurements?
- How to compute each cycle of updates?
- How to combine prediction and correction?
  - If the dynamics model is too strong, will end up ignoring the data
  - If the observation model is too strong tracking is the observation model is too strong, tracking is reduced to repeated detection

Slide credit: Kristen Grauman

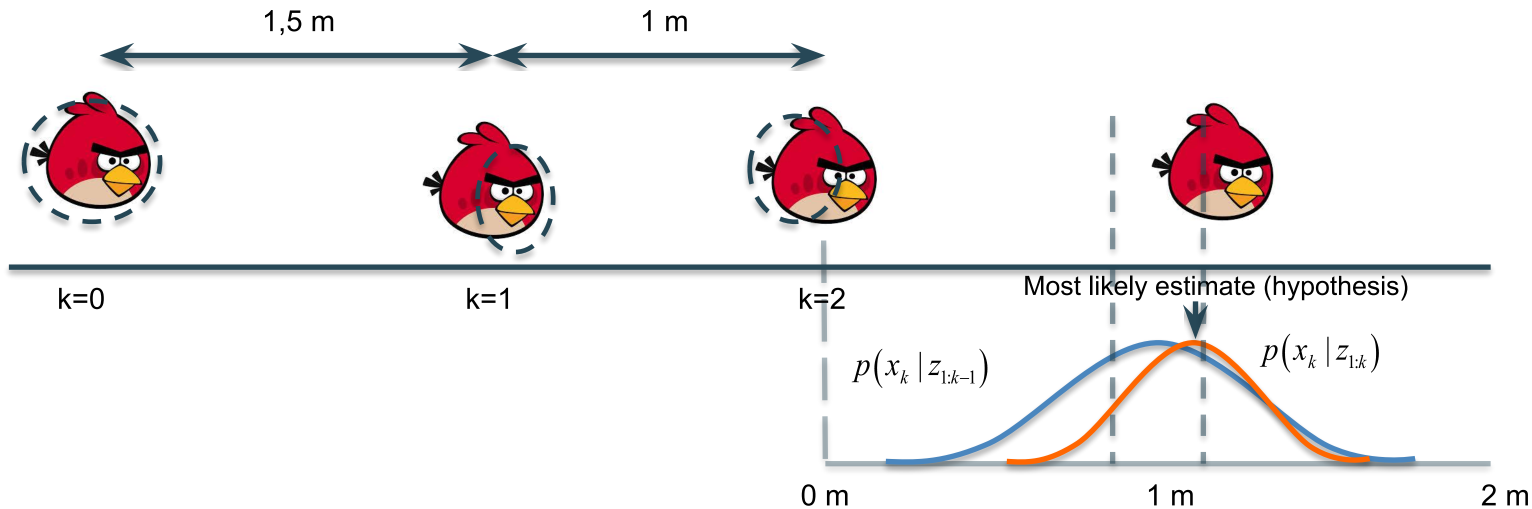
# Bayesian filtering



# Introduction

## Kalman Filter

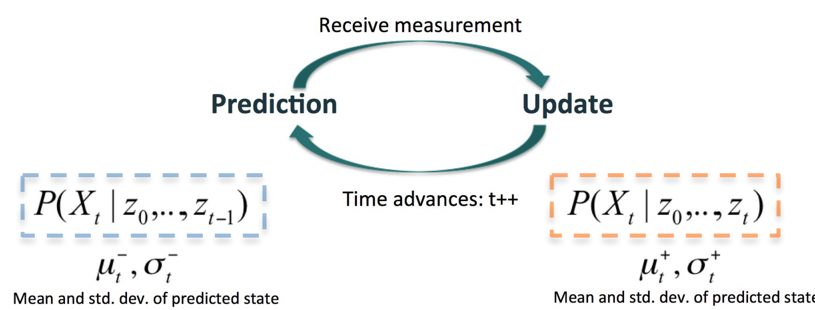
- **Definition:** Algorithm that processes measurements to deduce the **optimal estimate** of the state of a **linear system** using a set of measurements and a **statistical model** of the system.




$\phi$ ( $x_{k-1}$, $v_{k-1}$ ) = $Dx_{k-1}$ + $\Sigma_d$

$\psi$ ( $x_{t-1}$, $v_{t-1}$ ) = $Mx_{t-1}$ + $\Sigma_m$

$$\mathbf{x}_t \sim N(\mathbf{D}\mathbf{x}_{t-1}; \boldsymbol{\Sigma}_d)$$
$$z_t \sim N(\mathbf{M}\mathbf{x}_t; \boldsymbol{\Sigma}_m)$$



# Tracking

## Assumptions

- **Markov process:** Current state $X_k$ depends **only** on previous state $X_{k-1}$

$$p(x_k \mid x_0, \cdots, x_{k-1}) = p(x_k \mid x_{k-1})$$

Dynamics model

- **Independence:** Measurement depends only on current state

$$p(z_t \mid x_0, \cdots, x_k) = p(z_k \mid x_k)$$

Observation model

# Tracking

## Filtering framework

- Discrete-time state space filtering
- We want to **recursively** estimate the current state at every time that a measurement is received

- **Prediction**:
  - Propagate state pdf forward in time, taking noise into account (translate, deform, and spread the pdf)

- **Update**:
  - Use Bayes theorem to modify prediction pdf based on current measurement

pdf: the probability density function of a continuous random variable is a function that describes the relative likelihood for this random variable to take on a given value.

# Tracking

## Bayesian estimation

- Consider

$\{x_k\}_{k\geq 0}$: **Unobserved** process with transition density $x_k | x_{k-1} \sim f(\cdot | x_{k-1})$

$\{z_k\}_{k\geq 1}$: **Observations**, conditionally independent given $\{x_k\}_{k\geq 0}$, of marginal density $z_k | x_k \sim g(\cdot | x_k)$

$$\begin{cases} x_k = \phi(x_{k-1}, v_{k-1}) \\ z_k = \psi(x_k, w_k) \end{cases}$$

$\phi, \psi$ : Deterministic mappings

$\{v_k\}_{k\geq 2}, \{w_k\}_{k\geq 1}$ : Independent noises

# Tracking

## Tracking as induction

- Goal: Our goal is to obtain $p(x_k \mid z_{1:k})$

- Base case:
  - Assume we have an initial **prior** that predicts state in absence of any evidence: **p**(**x**$_0$)
  - At the first frame**, update** given the value of **z**$_0$

$$\underbrace{p(x_0 \mid z_0)}_{\substack{\text{Posterior prob.}\\ \text{of state given}\\ \text{measurement}}} = \frac{p(z_0 \mid x_0)p(x_0)}{p(z_0)} \propto \underbrace{p(z_0 \mid x_0)}_{\substack{\text{Likelihood of}\\ \text{measurement}}}\underbrace{p(x_0)}_{\substack{\text{Prior of}\\ \text{the state}}}$$

# Tracking

## Tracking as induction

- Prediction:

(Chapman-Kolmogorov equation)

$$p(x_k \mid z_{1:k-1}) = \int \underbrace{p(x_k \mid x_{k-1})}_{\text{Dynamics model}} \underbrace{p(x_{k-1} \mid z_{1:k-1})}_{\text{Corrected estimate from previous step}} dx_{k-1}$$

- Update:

$$p(x_k \mid z_{1:k}) = \frac{\overbrace{p(z_k \mid x_k)}^{\text{Observation model}} \overbrace{p(x_k \mid z_{1:k-1})}^{\text{Predicted estimate}}}{\int p(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k}$$

Slide credit: Svetlana Lazebnik

# LINEAR DYNAMIC MODELS

# Linear Dynamics Models (LDM)

## Linear dynamics models

- Dynamics model
  - State undergoes linear transformation **D** plus Gaussian noise

$$\phi\,(x_{k-1}, v_{k-1}) = Dx_{k-1} + \Sigma_d$$

$$x_k \sim N(Dx_{k-1}, \Sigma_d)$$

State at time t comes from a transformation (D) of previous state (k-1) plus a noise term

- Observation model
  - Measurement is the linearly transformed state plus Gaussian noise.

$$\psi\,(z_k, v_k) = Mx_k + \Sigma_m$$

$$z_k \sim N(Mx_k, \Sigma_m)$$

Measurement at time k comes from a transformation (M) of current state k plus a noise term

# Linear Dynamics Models (LDM)

## Notation

$$x_k \sim N(\mu, \Sigma)$$

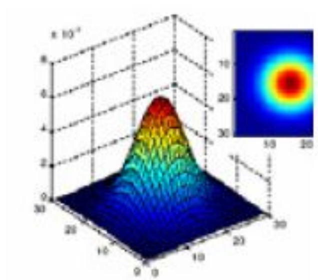
- Random variable with Gaussian probability distribution that has mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
  - $x$ and $\mu$ are $d$-dimensional, $\boldsymbol{\Sigma}$ is $d$x$d$.

- For the unidimensional case, $\boldsymbol{\mu}$ is a scalar and $\boldsymbol{\Sigma}$ is a 1x1 matrix → the variance $\sigma^2$

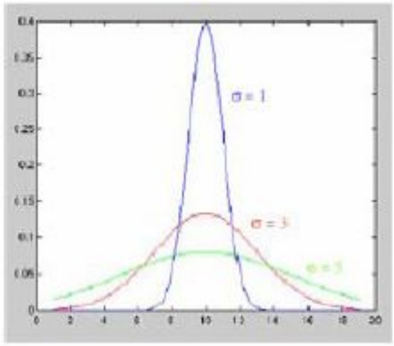
$$x_t \sim N(\mu, \sigma^2)$$

$$(2\pi)^{-\frac{k}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})},$$

***d*=2**



***d*=1**



# Linear Dynamics Models (LDM)

## Linear dynamics models

- Example: linear velocity (1D points)
  - **State vector:** position p and velocity v

$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} \qquad \begin{aligned} p_t &= p_{t-1} + (\Delta t) v_{t-1} + \varepsilon \\ v_t &= v_{t-1} + \xi \end{aligned}$$

(greek letters denote noise terms)

$$x_t = D_t x_{t-1} + noise = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + noise$$

  - **Measurement** is position only

$$z_t = M x_t + noise = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix} + noise$$

$$\mathbf{x}_t \sim N(\mathbf{D}\mathbf{x}_{t-1}; \boldsymbol{\Sigma}_d)$$
$$z_t \sim N(\mathbf{M}\mathbf{x}_t; \boldsymbol{\Sigma}_m)$$

# KALMAN FILTER

# Tracking with LDM case: Kalman Filter

## Kalman filter

- Optimal method for tracking **linear dynamical models** under the assumption of **Gaussian noise**

- The predicted/corrected state distributions are Gaussian
  - Only the mean and covariance should be estimated.
  - The calculations are easy (all the integrals can be done in closed form).

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D)

Know corrected state from k-1, and all measurements up to the current one.
→ Predict distribution over next state.

Know prediction of state, and next measurement
→ Update distribution over current state.

Receive measurement

**Prediction**

**Update**

Time advances: k++

$$p(x_k \mid z_{1:k-1})$$

$$\mu_k^-, \sigma_k^-$$

Mean and std. dev. of predicted state:

$$p(x_k \mid z_{1:k})$$

$$\mu_k^+, \sigma_k^+$$

Mean and std. dev. of predicted state:

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D): prediction

- Linear dynamic model defining predicted state evolution, with noise

$$x_k = N(d \cdot x_{k-1}, \sigma_d^2)$$

- Estimate the predicted distribution for next state

$$p(x_k \mid z_{0:k-1}) = N\left(\mu_k^-, (\sigma_k^-)^2\right)$$

- Update mean and variance

$$\boxed{\mu_k^- = d \cdot \mu_{k-1}^+} \qquad \boxed{(\sigma_k^-)^2 = \sigma_d^2 + (d \cdot \sigma_{k-1}^+)^2}$$

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D): correction

- Linear model defining the mapping of state to measurements:

$$z_k = N(m \cdot x_k, \sigma_m^2)$$

- Want to estimate corrected distribution given latest measurement:

$$P(x_k \mid z_{1:k}) = N\left(\mu_k^+, (\sigma_k^+)^2\right)$$

- Update mean & variance

$$\mu_k^+ = \frac{\mu_k^- \cdot \sigma_m^2 + m \cdot z_k \cdot (\sigma_k^-)^2}{\sigma_m^2 + m^2 \cdot (\sigma_k^-)^2}$$

$$(\sigma_t^+)^2 = \frac{\sigma_m^2 \cdot (\sigma_k^-)^2}{\sigma_m^2 + m^2 \cdot (\sigma_k^-)^2}$$

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D): prediction vs. update

$$\mu_k^+ = \frac{\mu_k^- \cdot \sigma_m^2 + m \cdot z_k \cdot (\sigma_k^-)^2}{\sigma_m^2 + m^2 \cdot (\sigma_k^-)^2} \qquad (\sigma_k^+)^2 = \frac{\sigma_m^2 \cdot (\sigma_k^-)^2}{\sigma_m^2 + m^2 \cdot (\sigma_k^-)^2}$$

- If there is no prediction uncertainty: $\sigma_k^- = 0$
  - **Measurement is ignored!**

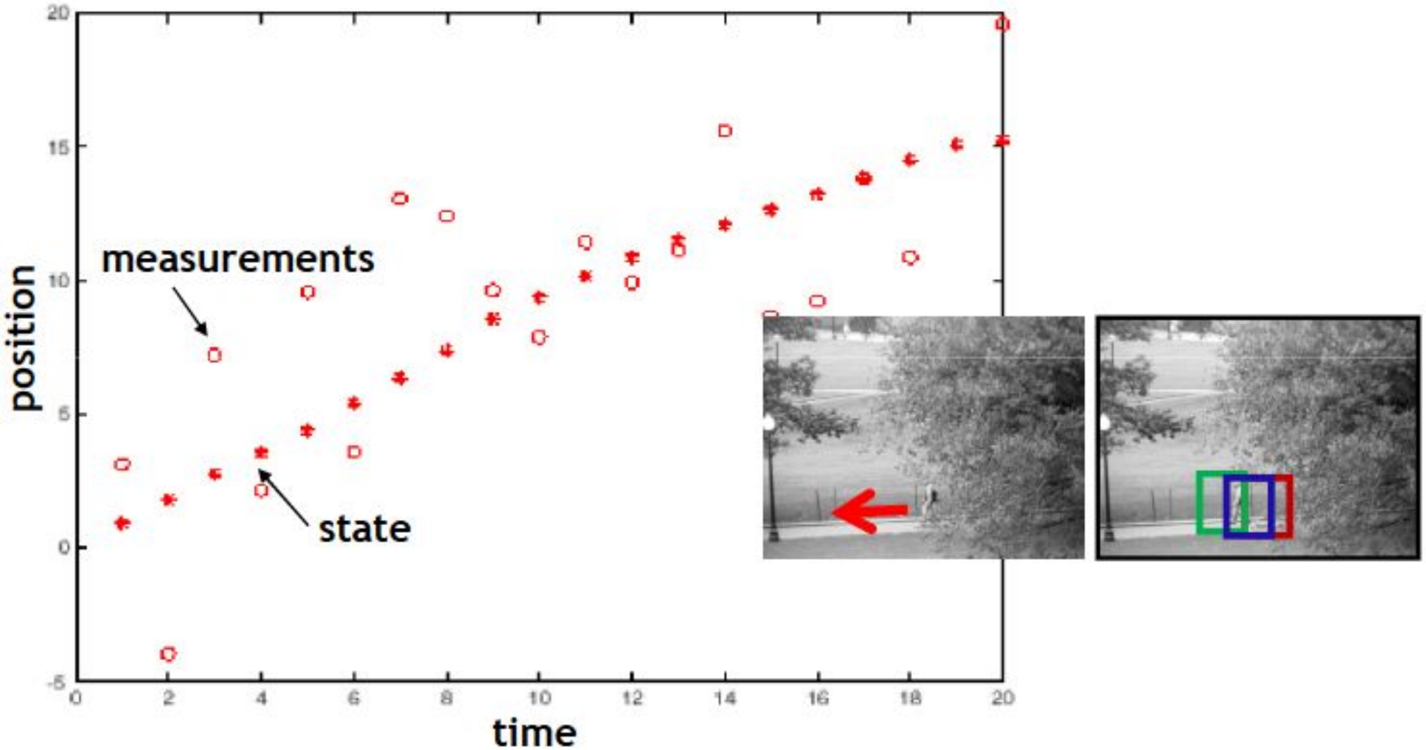
$$\mu_k^+ = \mu_k^- \qquad (\sigma_k^+)^2 = 0$$

- If there is no measurement uncertainty: $\sigma_m = 0$
  - **Prediction is ignored!**

$$\mu_k^+ = \frac{z_k}{m} \qquad (\sigma_k^+)^2 = 0$$

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D): example

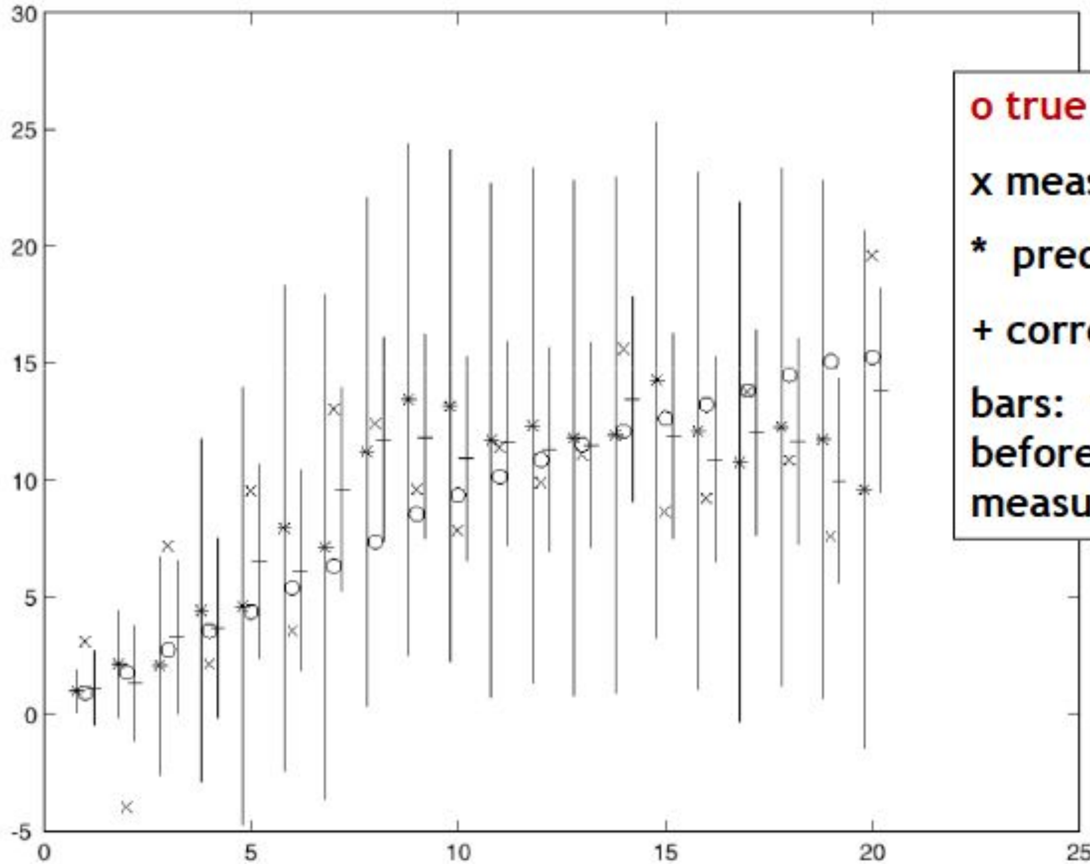



State is 2D: position + velocity
Measurement is 1D: position

Slide credit: Kristen Grauman

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D): example




o true state

x measurement

* predicted mean estimate

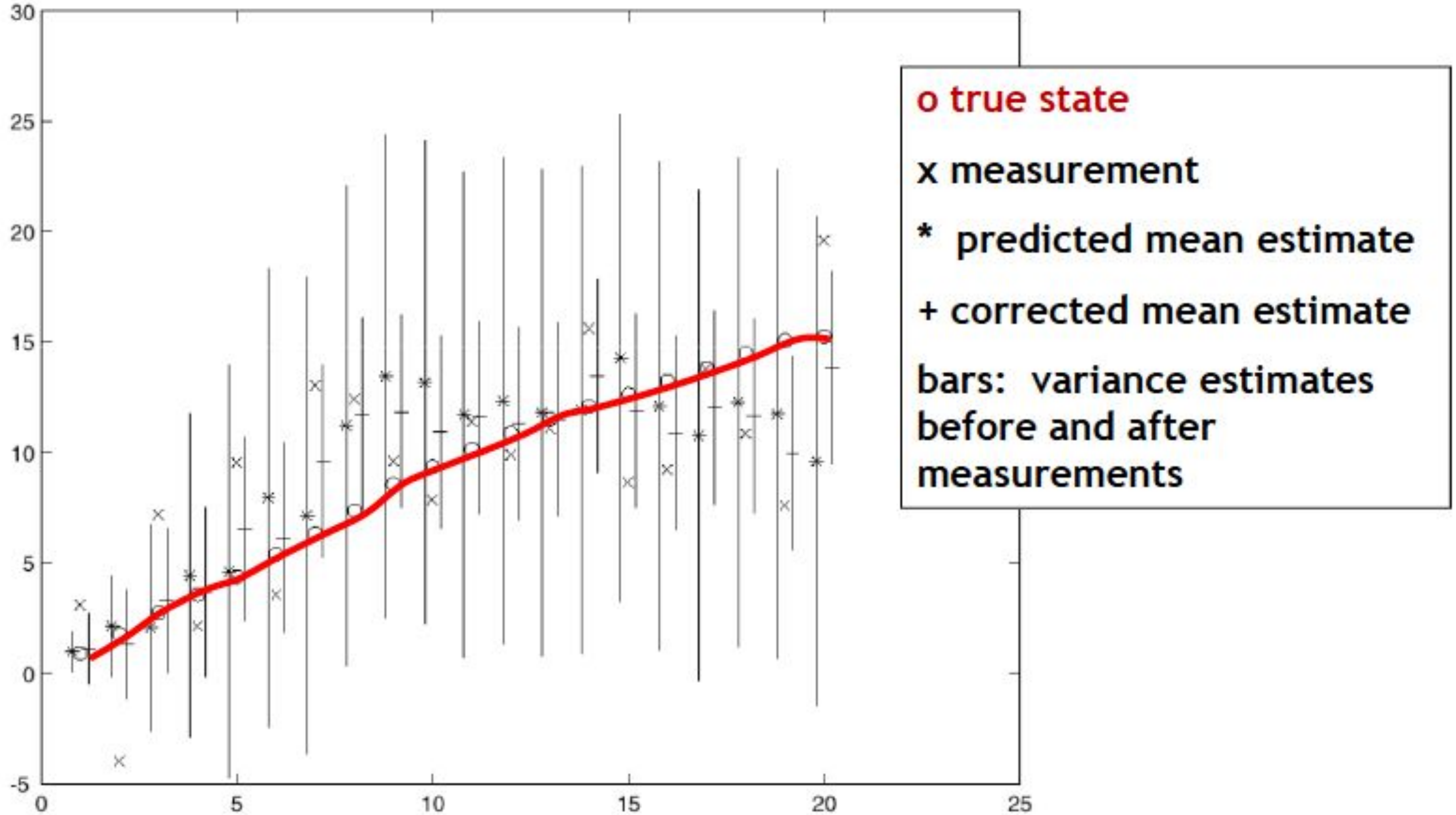
+ corrected mean estimate

bars: variance estimates before and after measurements

Slide credit: Kristen Grauman

# Tracking with LDM case: Kalman Filter

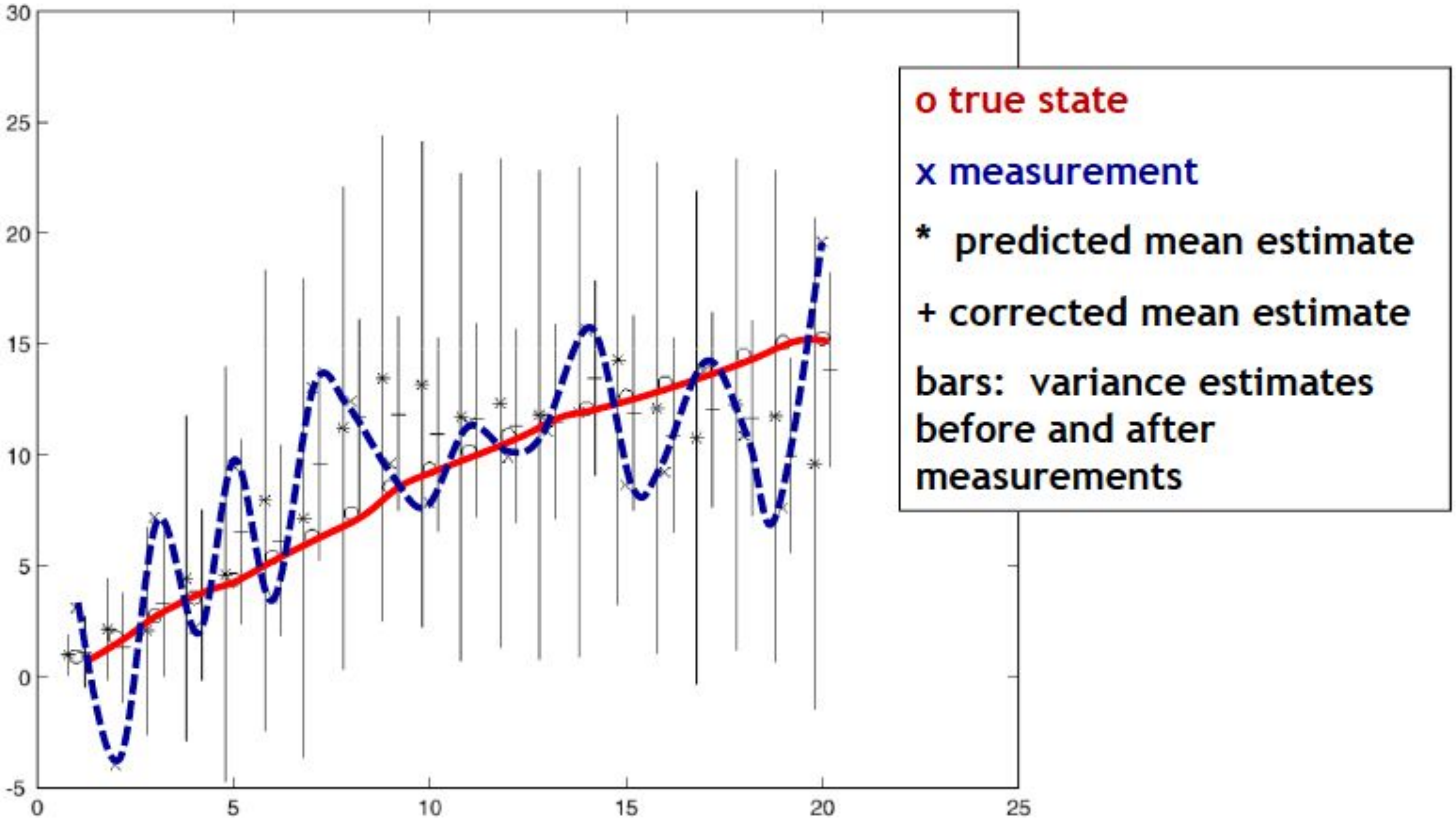
## Kalman filter (1D): example




Slide credit: Kristen Grauman

# Tracking with LDM case: Kalman Filter

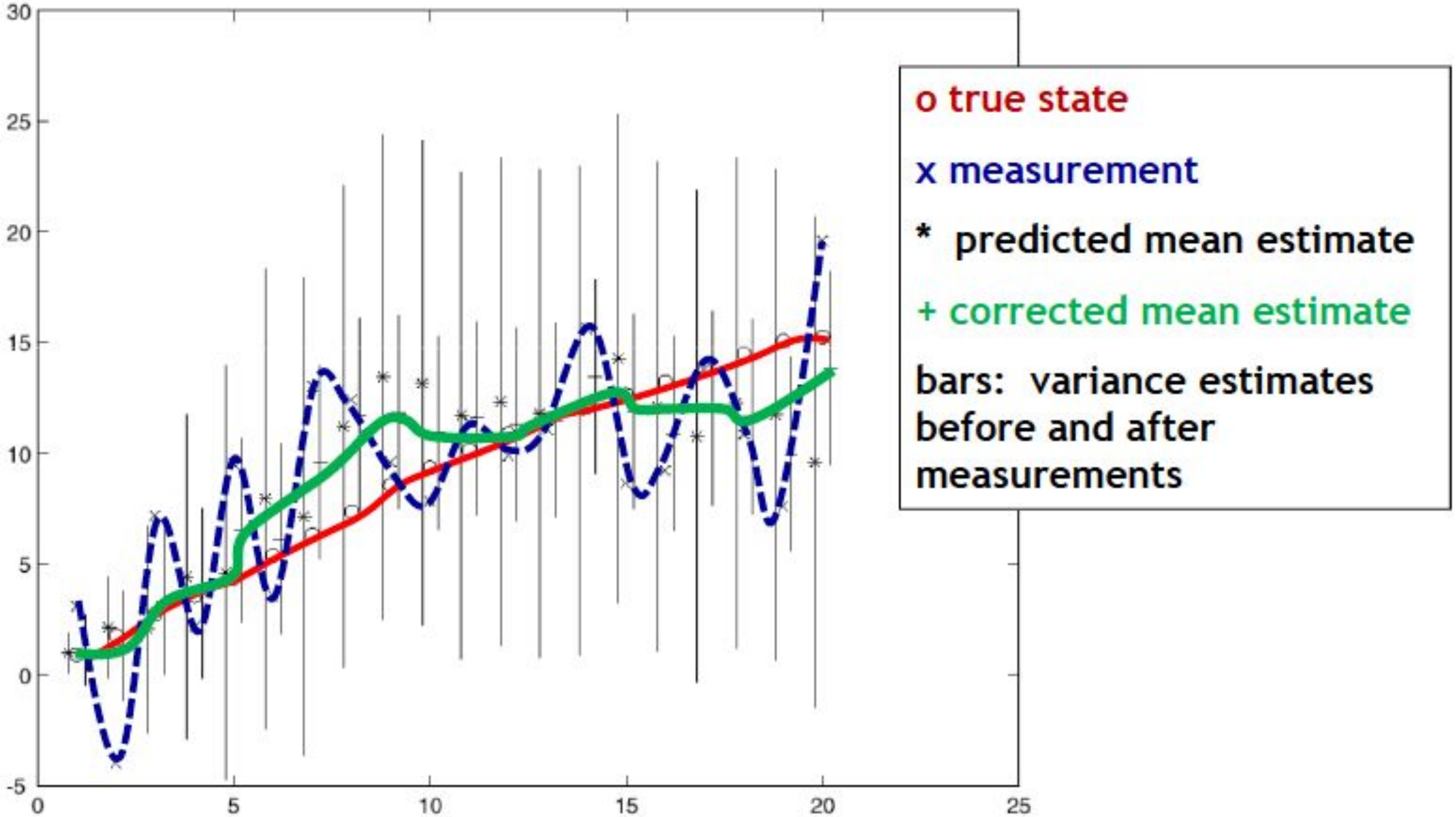
## Kalman filter (1D): example




Slide credit: Kristen Grauman

# Tracking with LDM case: Kalman Filter

## Kalman filter (1D): example




Slide credit: Kristen Grauman

# Tracking with LDM case: Kalman Filter

## Kalman filter: generic case

- Vectors with more than one dimension:

**Prediction**

$$x_k^- = Dx_{k-1}^+$$

$$\Sigma_k^- = D\Sigma_k^+ D^T + \Sigma_d$$

**Update**

$$x_k^+ = x_k^- + \underbrace{\boxed{K_k}}_{\text{Kalman gain}} \cdot \underbrace{\boxed{(z_k - M \cdot x_k^-)}}_{\text{Residual}}$$

$$\Sigma_k^+ = (I - K_k M)\Sigma_k^-$$

$$\boxed{K_k = \Sigma_k^- M^T (M\Sigma_k^- M^T + \Sigma_m)^{-1}}$$

- More weight on residual when measurement error covariance approaches 0.
- Less weight on residual as a priori estimate error covariance approaches 0.

Slide credit: Kristen Grauman

# Tracking with LDM case: Kalman Filter

## Kalman filter

- Pros:
  - Gaussian densities everywhere
  - Simple updates, compact and efficient
  - Very established method, very well understood
  - Optimal for linear dynamic models under Gaussian noise
- Cons:
  - Unimodal distribution, only single hypothesis
  - Restricted class of motions defined by linear model

**Modern visual trackers using Kalman Filter:**

- Bewley, Alex et al. “Simple Online and Realtime Tracking.” 2016 IEEE International Conference on Image Processing (ICIP)
- Nicolai Wojke et al. “Simple Online and Realtime Tracking with a Deep Association Metric”, arXiv. 1703.07402, 2017

# Tracking with LDM case: Kalman Filter

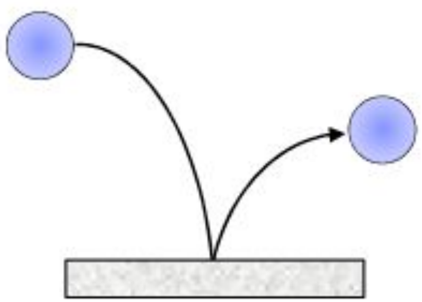
## Kalman filter

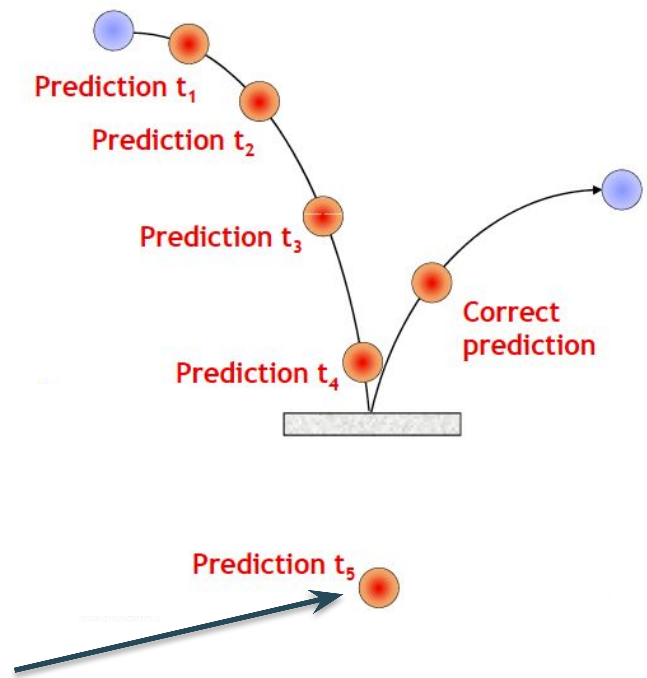

Source: https://github.com/Team-Neighborhood/Kalman-Filter-Image

# Tracking with LDM case: Kalman Filter

## Kalman filter

- Linear model often does not describe accurately the dynamics of the problem
  - E.g. bouncing ball






Slide credit: B. Leibe

# Tracking with LDM case: Kalman Filter

## Tracking issues

- Initialization
  - Often done manually
  - Background subtraction, detection can also be used
- Data association, multiple tracked objects
  - Occlusions, clutter
- Deformable and articulated objects
- Constructing accurate models of dynamics
  - E.g., Fitting parameters for a linear dynamics model
- Drift
  - Accumulation of errors over time

# Tracking without LDM assumption

## Other methods

- Extended Kalman filter
  - Removes linearity constraint on the state transition and observation models
- Mean-shift
  - Non-parametric technique
- Particle filter
  - Uses a sequential Montecarlo method
  - Multimodal distribution.
  - Removes linearity and gaussianity constraints