Object Tracking

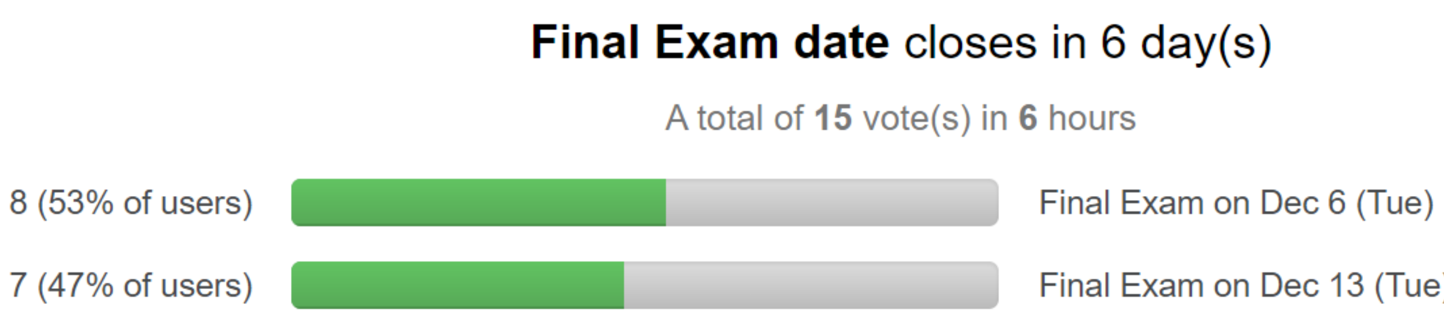
Computer Vision

Jia-Bin Huang, Virginia Tech

Many slides from D. Hoiem

Administrative stuffs

- HW 5 (Scene categorization)
  - Due 11:59pm on Wed, November 16
- Poll on Piazza –
  When should we have the final exam?
  - Dec 6
  - Dec 13



Today's class

- Explain HW 5 in detail
- Review/finish object detection
- Tracking Objects
    - Examples and Applications
    - Overview of probabilistic tracking
    - Kalman Filter
    - Particle Filter

HW 5

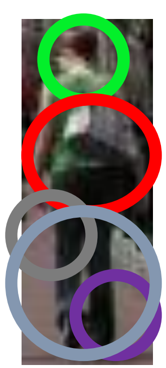
- Color histogram and k-nearest neighbor (kNN) classifier
- Bag of visual words model and nearest neighbor classifier
- Bag of visual words model and a discriminative classifier
- Spatial pyramid model and a discriminative classifier

Review: Statistical template

- Object model = log linear model of parts at fixed positions

$$+3 +2 -2 -1 -2.5 = -0.5 \overset{?}{>} 7.5$$

Non-object

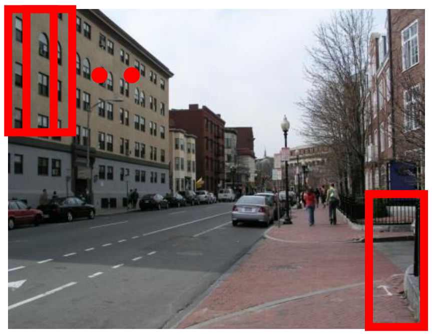


$$+4 +1 +0.5 +3 +0.5 = 10.5 \overset{?}{>} 7.5$$

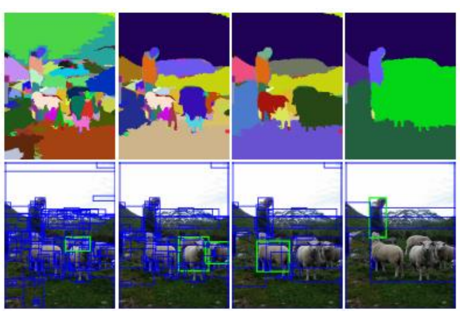
Object

Review: Statistical templates

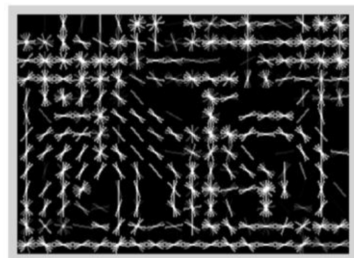
A. Propose Window → B. Extract Features → C. Classify → D. Post-process



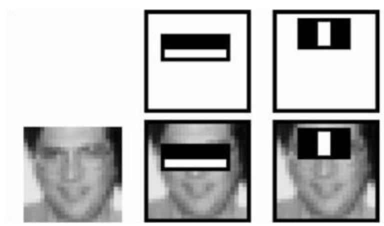
Sliding window: scan image pyramid



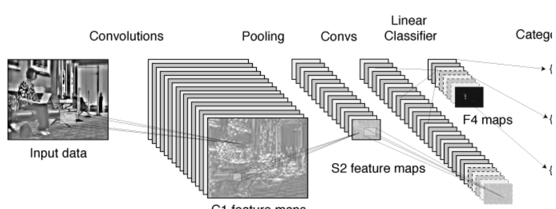
Region proposals: edge/region-based, resize to fixed window



HOG



Fast randomized features



CNN features

SVM

Boosted stubs

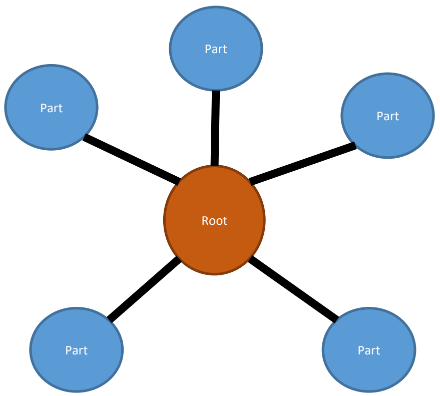
Neural network

Non-max suppression

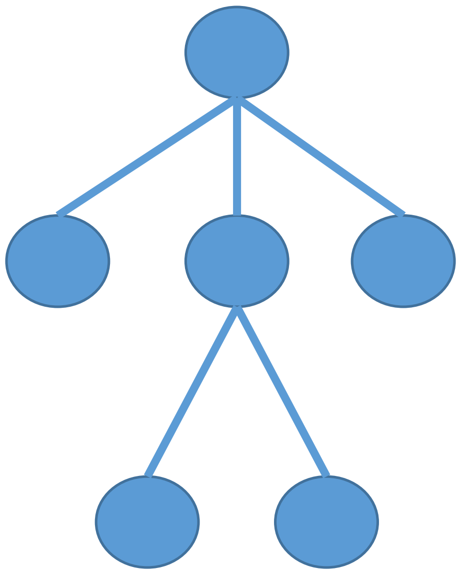
Segment or refine localization

Review: Part-based Models

1. Star-shaped model
   - Example: Deformable Parts Model
     - [Felzenswalb et al. 2010](#)



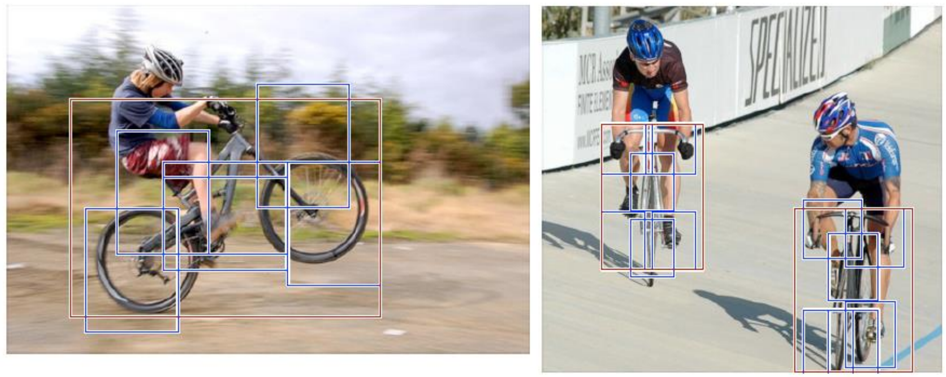
2. Tree-shaped model
   - Example: Pictorial structures
     - [Felzenszwalb Huttenlocher 2005](#)



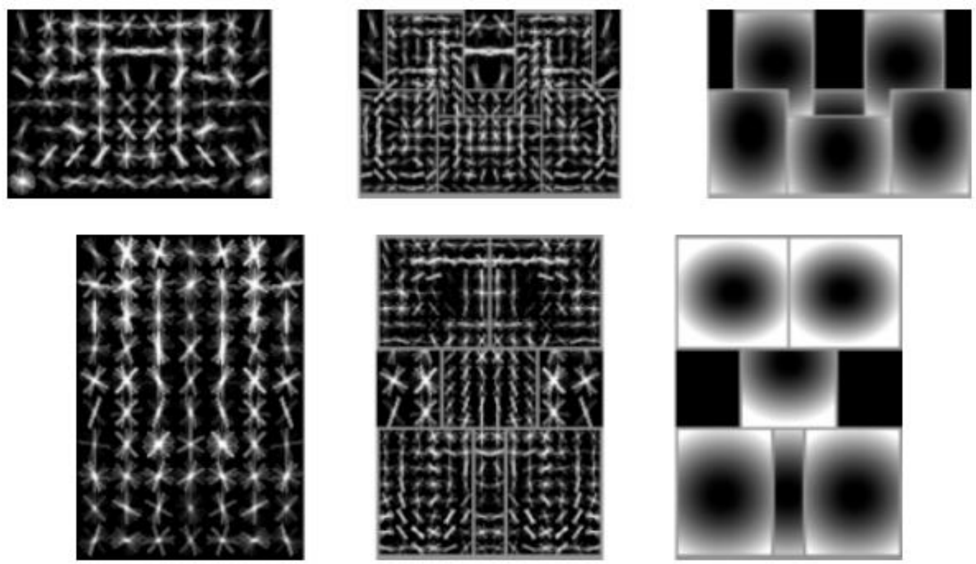
3. Sequential prediction models

Deformable Latent Parts Model (DPM)

Detections



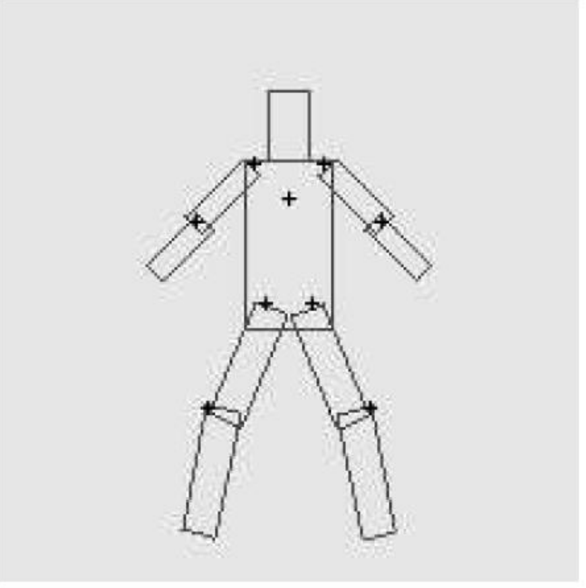
Template Visualization



root filters coarse resolution | part filters finer resolution | deformation models

Felzenszwalb et al. 2008, 2010

Pictorial Structures Model



$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right)$$

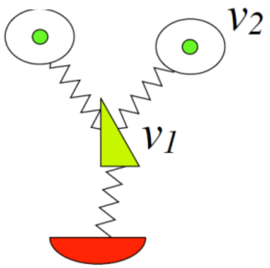
Appearance likelihood

Geometry likelihood

Pictorial Structures Model

Optimization is tricky but can be efficient

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$



- For each l_1, find best l_2:

$$\text{Best}_2(l_1) = \min_{l_2} \left[m_2(l_2) + d_{12}(l_1, l_2) \right]$$

- Remove v_2, and repeat with smaller tree, until only a single part
- For k parts, n locations per part, this has complexity of $O(kn^2)$, but can be solved in ~O(kn) using generalized distance transform

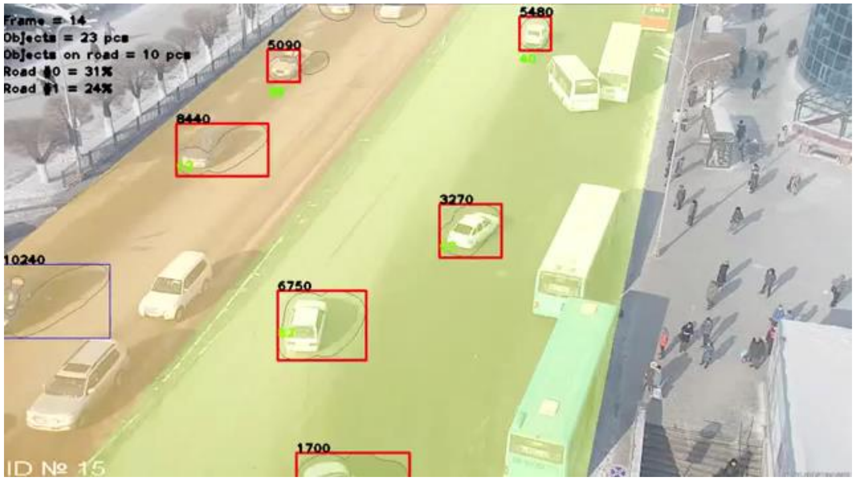
Tracking Objects

Goal:

Locating a moving object/part across video frames

- Examples and Applications
- Overview of visual tracking
- Motion models: probabilistic tracking
  - Kalman Filter
  - Particle Filter

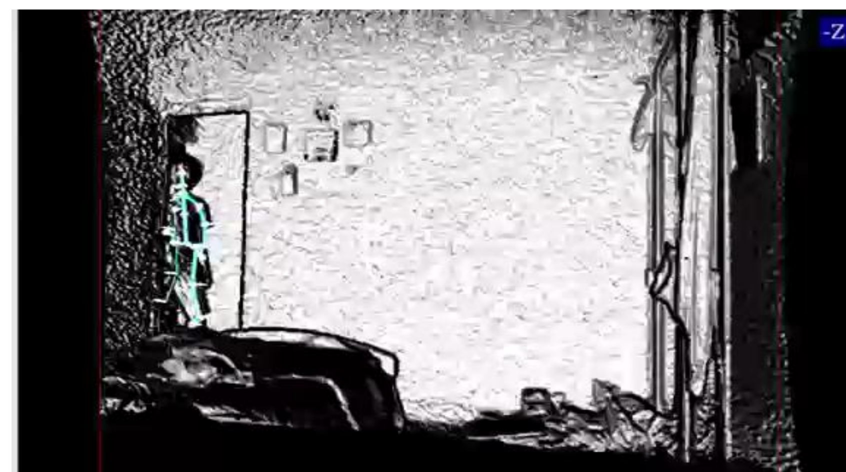
Tracking examples



Traffic

Sports

Face

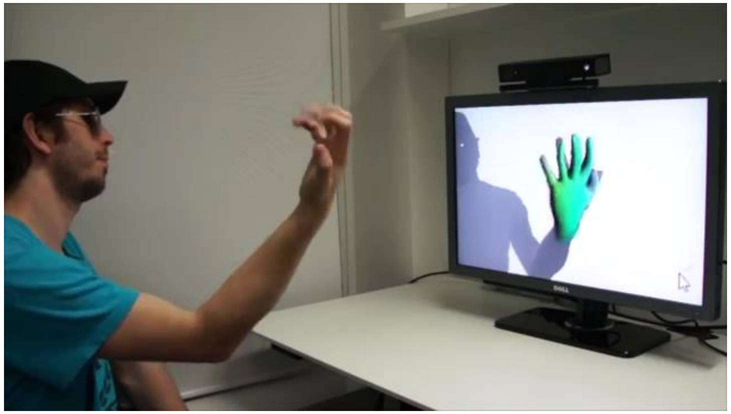


Body

Further applications

- Motion capture
- Augmented Reality
- Action Recognition
- Security, traffic monitoring
- Video Compression
- Human-computer interaction
- Video Summarization
- Medical Screening




Tracking Examples

Traffic: https://www.youtube.com/watch?v=DiZHQ4peqjg

Soccer: http://www.youtube.com/watch?v=ZqQIItFAnxg

Face: http://www.youtube.com/watch?v=i_bZNVmhJ2o

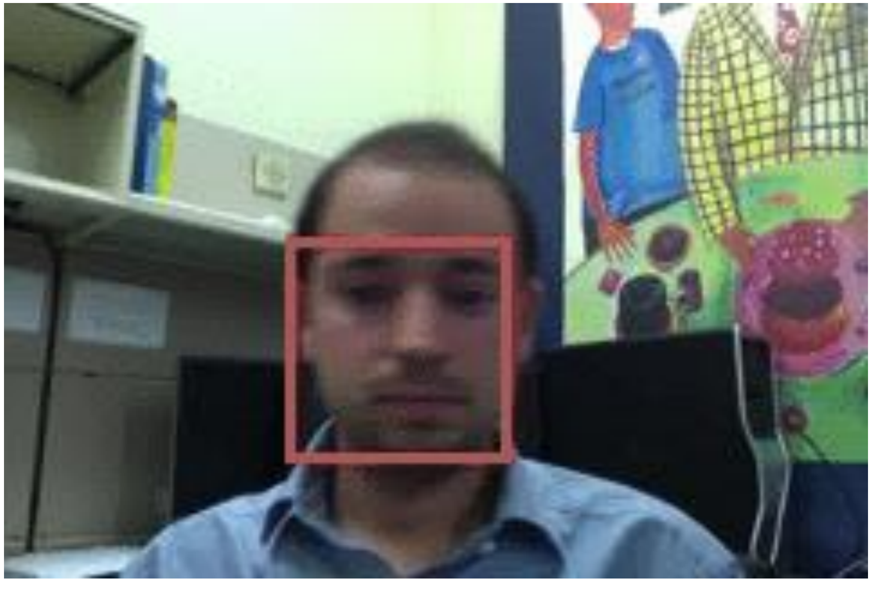
Body: https://www.youtube.com/watch?v=_Ahy0Gh69-M

Eye: http://www.youtube.com/watch?v=NCtYdUEMotg

Gaze: http://www.youtube.com/watch?v=-G6Rw5cU-1c

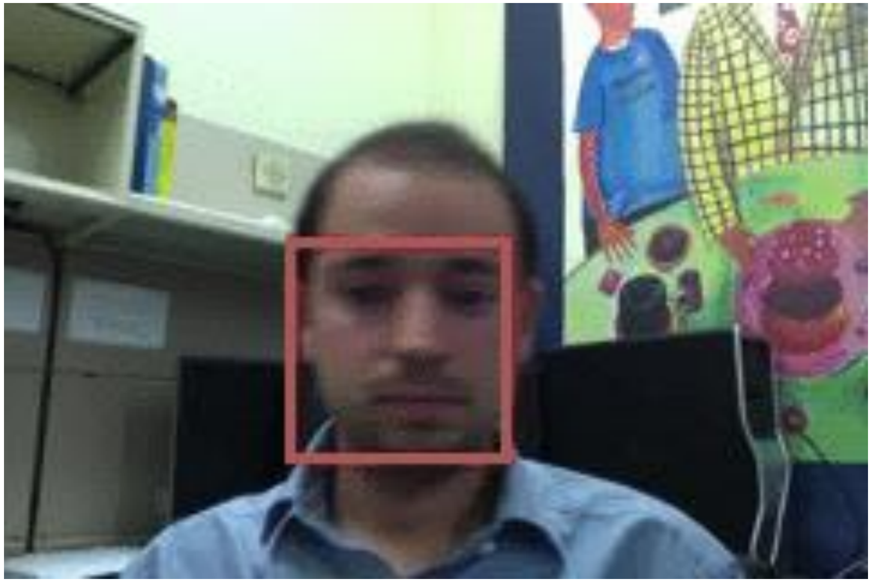
Things that make visual tracking difficult

- Small, few visual features
- Erratic movements, moving very quickly
- Occlusions, leaving and coming back
- Surrounding similar-looking objects



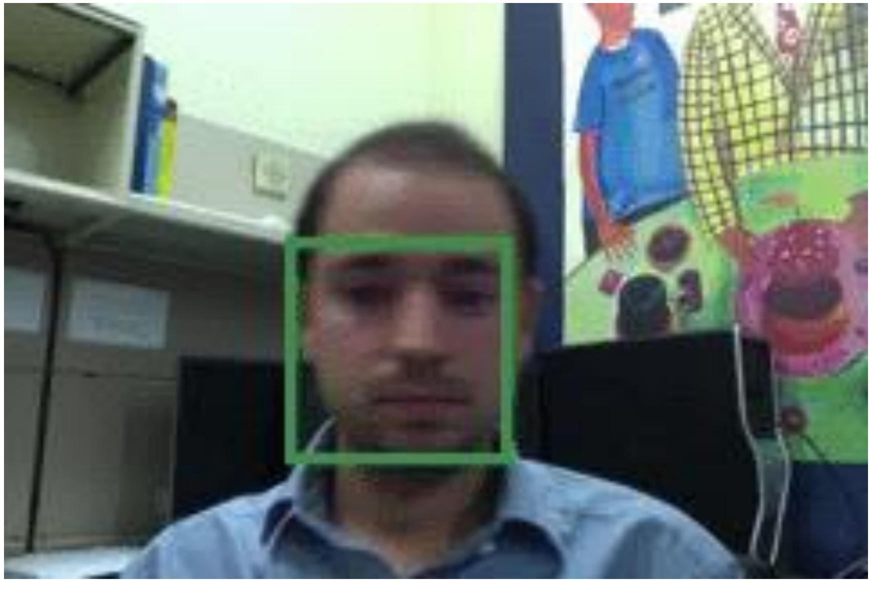
Strategies for tracking

- Tracking by repeated detection
  - Works well if object is easily detectable (e.g., face or colored glove) and there is only one
  - Need some way to link up detections
  - Best you can do, if you can't predict motion



Tracking with dynamics

- Key idea: Based on a model of expected motion, predict where objects will occur in next frame, before even seeing the image
  - Restrict search for the object
  - Measurement noise is reduced by trajectory smoothness
  - Robustness to missing or weak observations

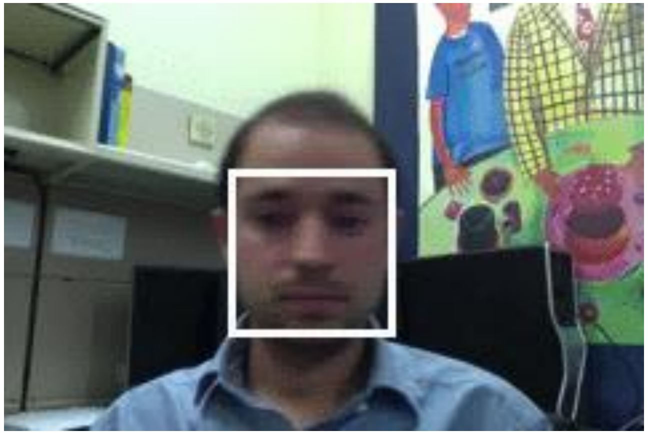


Strategies for tracking

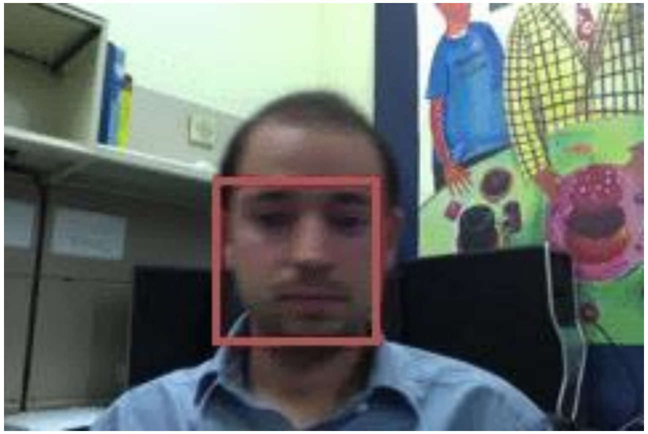
- Tracking with motion prediction
  - Predict the object’s state in the next frame
  - **Kalman filtering**: next state can be linearly predicted from current state (Gaussian)
  - **Particle filtering**: sample multiple possible states of the object (non-parametric, good for clutter)

General model for tracking

- ***state X***: The actual state of the moving object that we want to estimate
  - State could be any combination of position, pose, viewpoint, velocity, acceleration, etc.



- ***observations Y***: Our actual measurement or observation of state X. Observation can be very noisy



- At each time t, the state changes to X_t and we get a new observation Y_t

Steps of tracking

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

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}\right)$$

Steps of tracking

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

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}\right)$$

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

$$P\left(X_t \middle| Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, Y_t = y_t\right)$$

Simplifying assumptions

- Only the immediate past matters

$$P(X_t | X_0, \ldots, X_{t-1}) = \boxed{P(X_t | X_{t-1})}$$

dynamics model

Simplifying assumptions

- Only the immediate past matters

$$P(X_t | X_0, \ldots, X_{t-1}) = \boxed{P(X_t | X_{t-1})}$$

dynamics model

- Measurements depend only on the current state

$$P(Y_t | X_0, Y_0 \ldots, X_{t-1}, Y_{t-1}, X_t) = \boxed{P(Y_t | X_t)}$$

observation model

Simplifying assumptions

- Only the immediate past matters

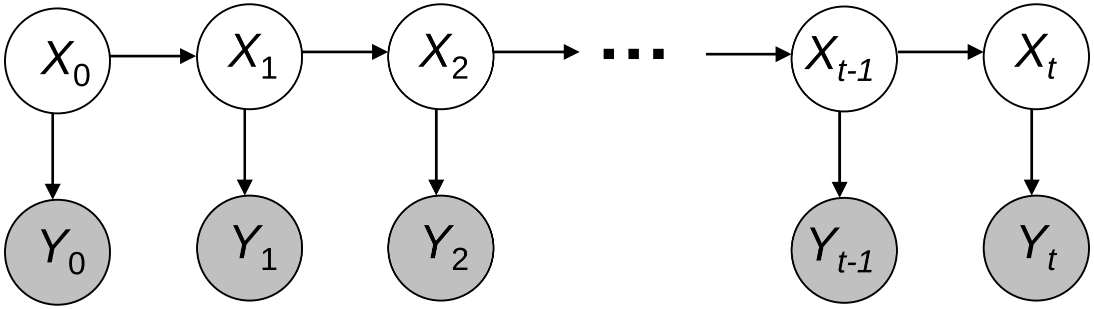
$$P(X_t | X_0, \ldots, X_{t-1}) = \boxed{P(X_t | X_{t-1})}$$

dynamics model

- Measurements depend only on the current state

$$P(Y_t | X_0, Y_0 \ldots, X_{t-1}, Y_{t-1}, X_t) = \boxed{P(Y_t | X_t)}$$

observation model



Problem statement

- We have models for

  Likelihood of next state given current state: $P\left(X_t \middle| X_{t-1}\right)$

  Likelihood of observation given the state: $P\left(Y_t \middle| X_t\right)$

- We want to recover, for each t: $P\left(X_t \middle| y_0, \ldots, y_t\right)$

Probabilistic tracking

- Base case:
  - Start with initial prior that predicts state in absence of any evidence: $P(X_0)$
  - For the first frame, *correct* this given the first measurement: $Y_0 = y_0$

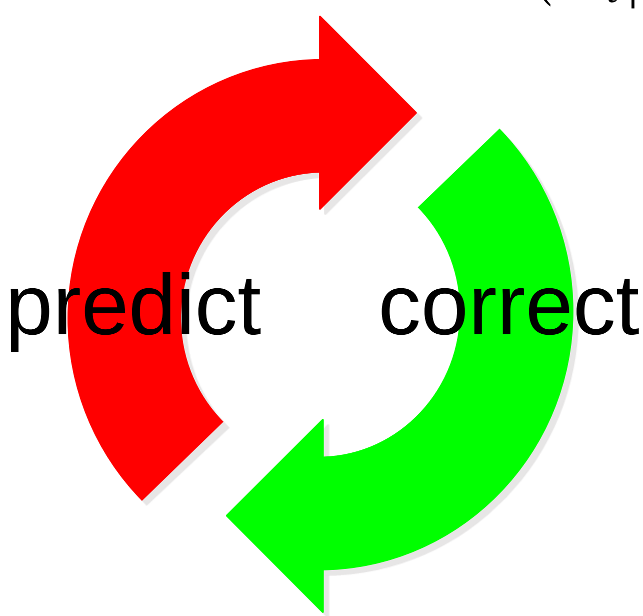
Probabilistic tracking

- Base case:
  - Start with initial prior that predicts state in absence of any evidence: $P(X_0)$
  - For the first frame, *correct* this given the first measurement: $Y_0=y_0$

$$P(X_0 \mid Y_0 = y_0) = \frac{P(y_0 \mid X_0)P(X_0)}{P(y_0)} \propto P(y_0 \mid X_0)P(X_0)$$

Probabilistic tracking

- Base case:
  - Start with initial prior that predicts state in absence of any evidence: $P(X_0)$
  - For the first frame, *correct* this given the first measurement: $Y_0 = y_0$
- Given corrected estimate for frame *t-1*:
  - Predict for frame *t* ➔ $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$
  - Observe y_t; Correct for frame *t* ➔ $P\left(X_t \middle| y_0, \ldots, y_{t-1}, y_t\right)$



Prediction

- Prediction involves representing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

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

$$= \int P\left(X_t, X_{t-1} \middle| y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

Law of total probability

Prediction

- Prediction involves representing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

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

$$= \int P\left(X_t, X_{t-1} \middle| y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

$$= \int P\left(X_t \mid X_{t-1}, y_0, \ldots, y_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

Conditioning on X_{t-1}

Prediction

- Prediction involves representing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

$$
\begin{aligned}
&P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) \\
&= \int P\left(X_t, X_{t-1} \middle| y_0, \ldots, y_{t-1}\right) dX_{t-1} \\
&= \int P\left(X_t \mid X_{t-1}, y_0, \ldots, y_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1} \\
&= \int P\left(X_t \mid X_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1}
\end{aligned}
$$

Independence assumption

Prediction

- Prediction involves representing $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$ given $P\left(X_{t-1} \middle| y_0, \ldots, y_{t-1}\right)$

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

$$= \int P\left(X_t, X_{t-1} \middle| y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

$$= \int P\left(X_t \mid X_{t-1}, y_0, \ldots, y_{t-1}\right) P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right) dX_{t-1}$$

$$= \int \underbrace{P\left(X_t \mid X_{t-1}\right)}_{\text{dynamics model}} \underbrace{P\left(X_{t-1} \mid y_0, \ldots, y_{t-1}\right)}_{\text{corrected estimate from previous step}} dX_{t-1}$$

Correction

- Correction involves computing $P\left(X_t \middle| y_0, \ldots, y_t\right)$ given predicted value $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$

Correction

- Correction involves computing $P\left(X_t \middle| y_0, \ldots, y_t\right)$ given predicted value $P\left(X_t \middle| y_0, \ldots, y_{t-1}\right)$

$$P\left(X_t \middle| y_0, \ldots, y_t\right)$$

$$= \frac{P\left(y_t \mid X_t, y_0, \ldots, y_{t-1}\right)}{P\left(y_t \mid y_0, \ldots, y_{t-1}\right)} P\left(X_t \mid y_0, \ldots, y_{t-1}\right)$$

Bayes' Rule

Correction

- Correction involves computing $P(X_t | y_0, \ldots, y_t)$ given predicted value $P(X_t | y_0, \ldots, y_{t-1})$

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

$$= \frac{P(y_t \mid X_t, y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})} P(X_t \mid y_0, \ldots, y_{t-1})$$

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})}$$

Independence assumption
(observation y_t directly depends only on state X_t)

Correction

- Correction involves computing $P(X_t | y_0, \ldots, y_t)$ given predicted value $P(X_t | y_0, \ldots, y_{t-1})$

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

$$= \frac{P(y_t \mid X_t, y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})} P(X_t \mid y_0, \ldots, y_{t-1})$$

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})}$$

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{\int P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1}) dX_t}$$

Conditioning on X_t

Correction

- Correction involves computing $P(X_t | y_0, \ldots, y_t)$ given predicted value $P(X_t | y_0, \ldots, y_{t-1})$

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

$$= \frac{P(y_t \mid X_t, y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})} P(X_t \mid y_0, \ldots, y_{t-1})$$

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{P(y_t \mid y_0, \ldots, y_{t-1})}$$

observation model

predicted estimate

$$= \frac{P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1})}{\int P(y_t \mid X_t) P(X_t \mid y_0, \ldots, y_{t-1}) dX_t}$$

normalization factor

Summary: Prediction and correction

Prediction:

$$P\left(X_t \mid y_0,\ldots,y_{t-1}\right) = \int \underbrace{P\left(X_t \mid X_{t-1}\right)}_{\text{dynamics model}} \underbrace{P\left(X_{t-1} \mid y_0,\ldots,y_{t-1}\right)}_{\text{corrected estimate from previous step}} dX_{t-1}$$

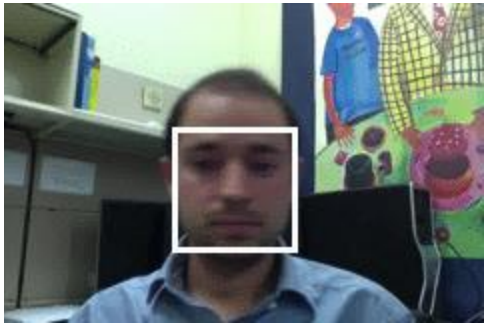
Correction:

$$P\left(X_t \mid y_0,\ldots,y_t\right) = \frac{\overbrace{P\left(y_t \mid X_t\right)}^{\text{observation model}} \overbrace{P\left(X_t \mid y_0,\ldots,y_{t-1}\right)}^{\text{predicted estimate}}}{\int P\left(y_t \mid X_t\right) P\left(X_t \mid y_0,\ldots,y_{t-1}\right) dX_t}$$

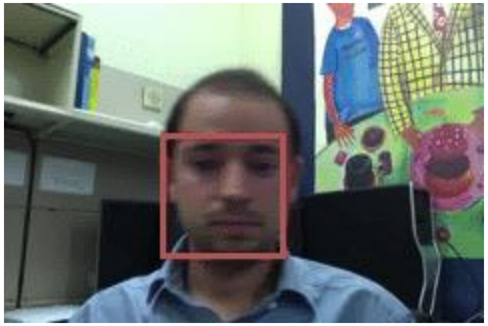
The Kalman filter

- Linear dynamics model: state undergoes linear transformation plus Gaussian noise
- Observation model: measurement is linearly transformed state plus Gaussian noise
- The predicted/corrected state distributions are Gaussian
  - You only need to maintain the mean and covariance
  - The calculations are easy (all the integrals can be done in closed form)

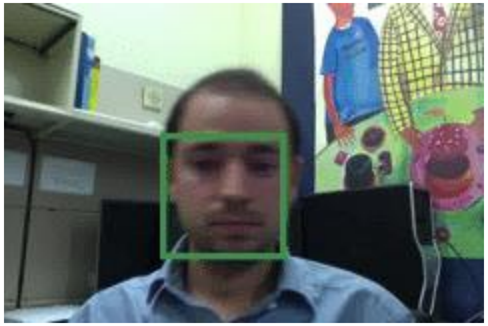
Example: Kalman Filter



Ground Truth



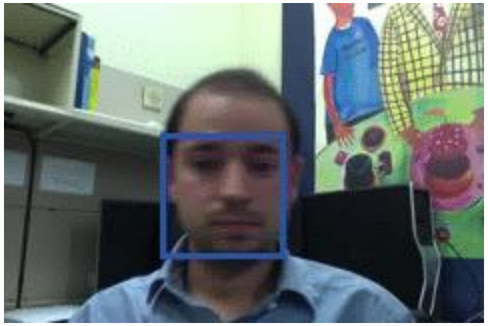
Observation



Correction

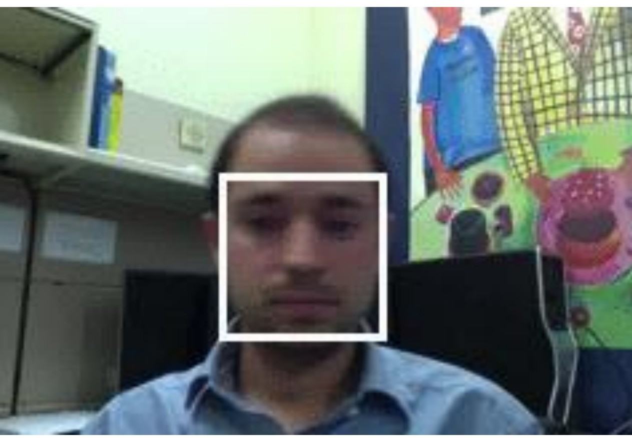
Update Location, Velocity, etc.

Next Frame

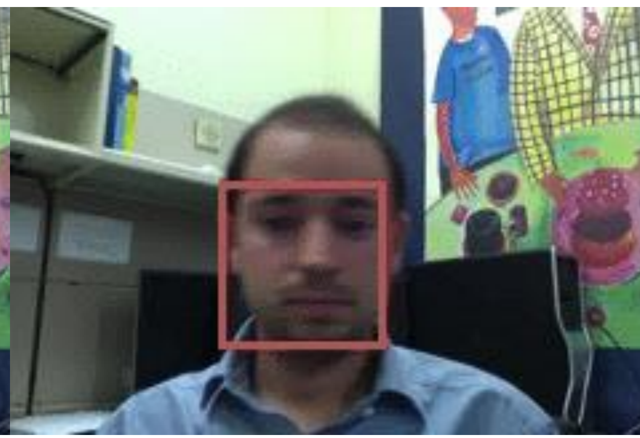


Prediction

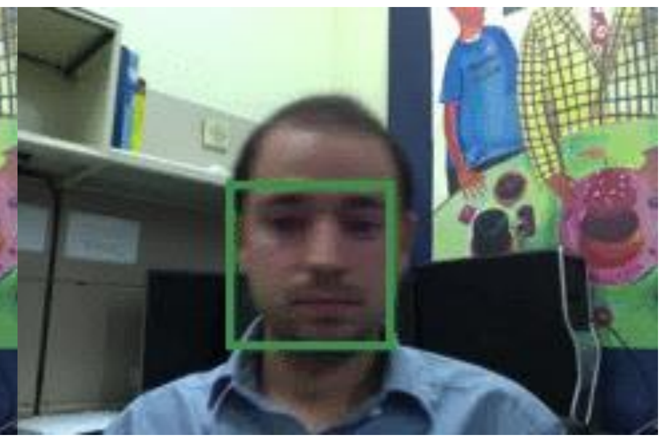
Comparison



Ground Truth

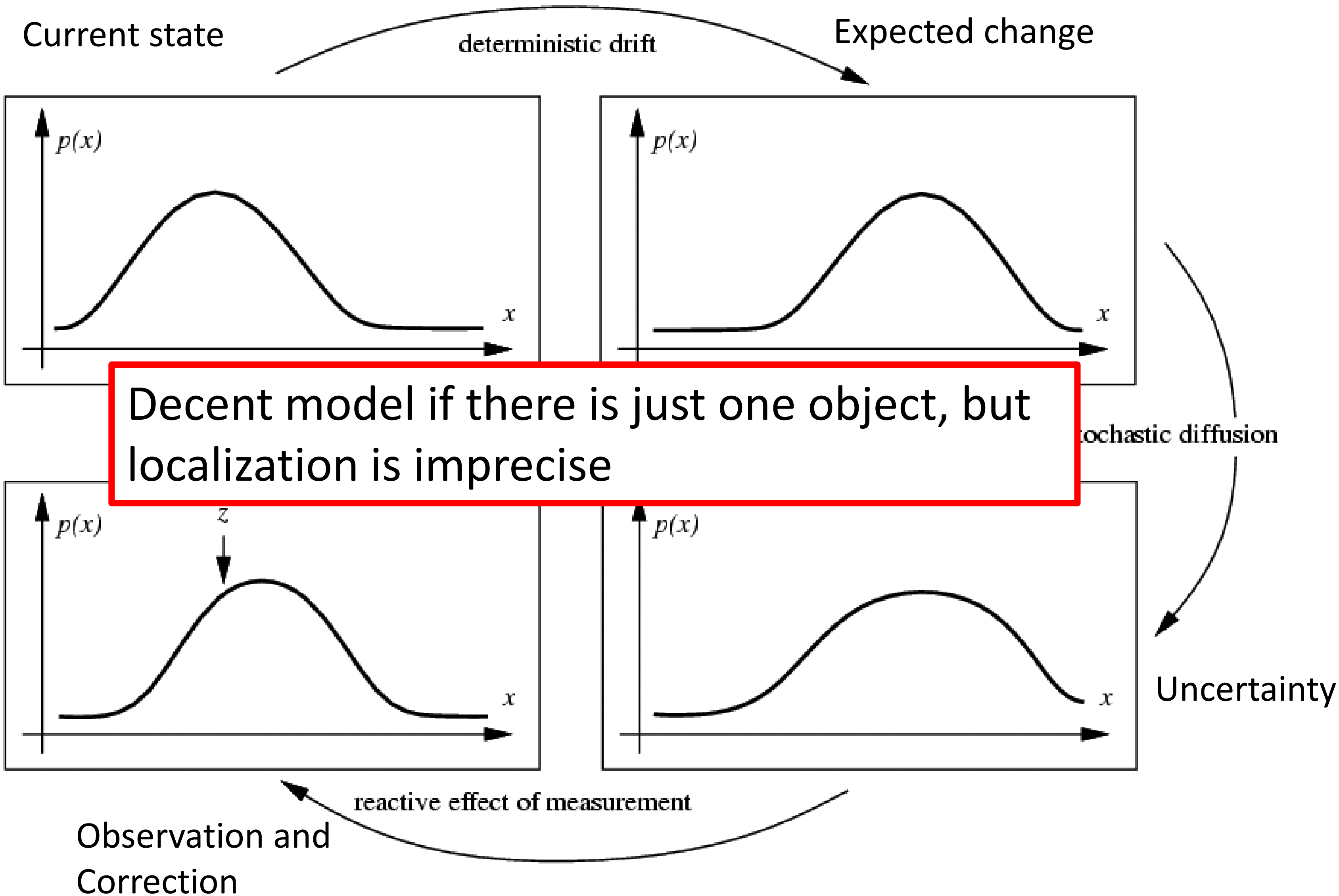


Observation



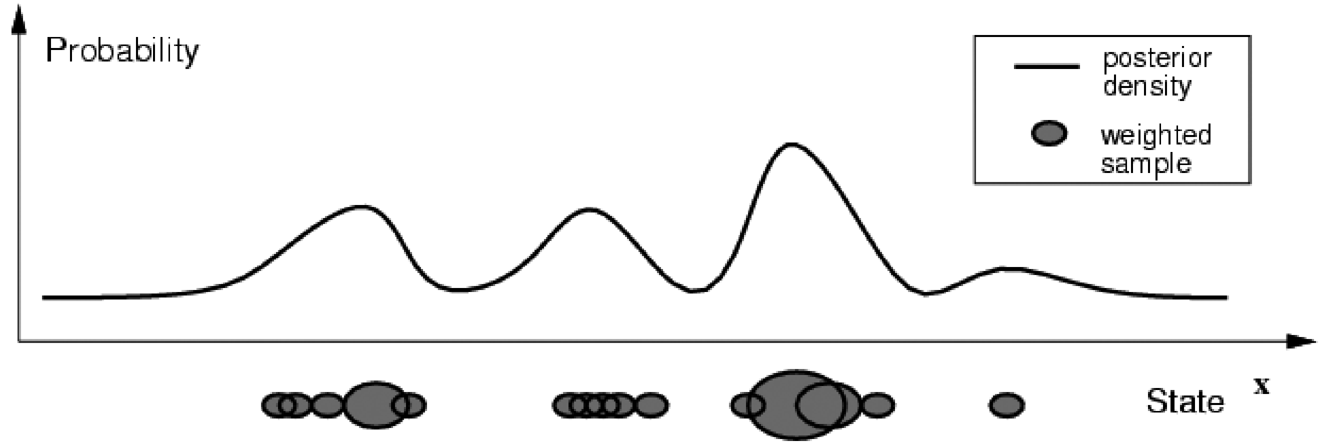
Correction

Propagation of Gaussian densities



Decent model if there is just one object, but localization is imprecise

Particle filtering

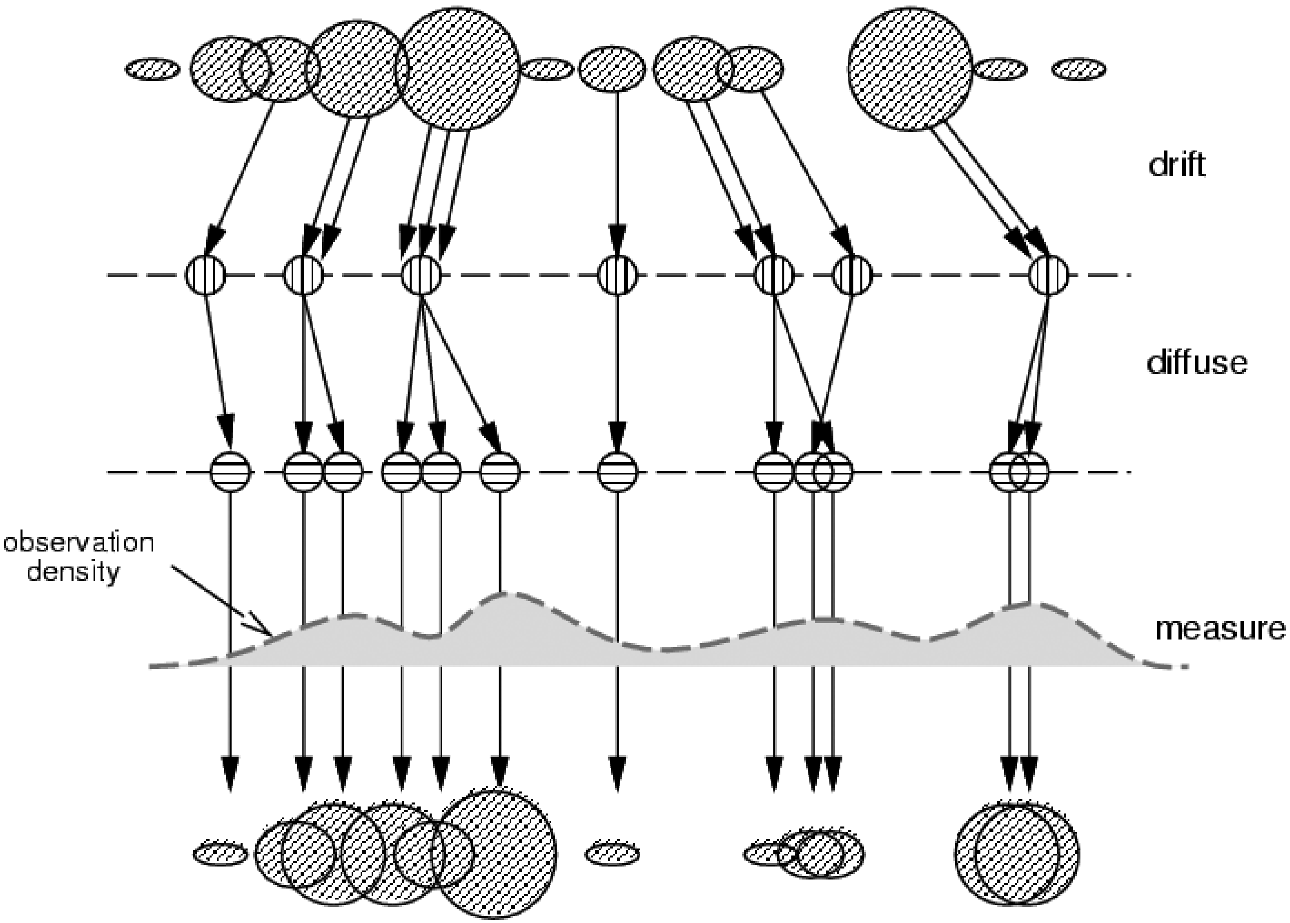


Represent the state distribution non-parametrically

- Prediction: Sample possible values X_{t-1} for the previous state
- Correction: Compute likelihood of X_t based on weighted samples and $P(y_t|X_t)$

M. Isard and A. Blake, [CONDENSATION -- conditional density propagation for visual tracking](), IJCV 29(1):5-28, 1998

Particle filtering



Start with weighted samples from previous time step

Sample and shift according to dynamics model

Spread due to randomness; this is predicted density $P(X_t|Y_{t-1})$

Weight the samples according to observation density

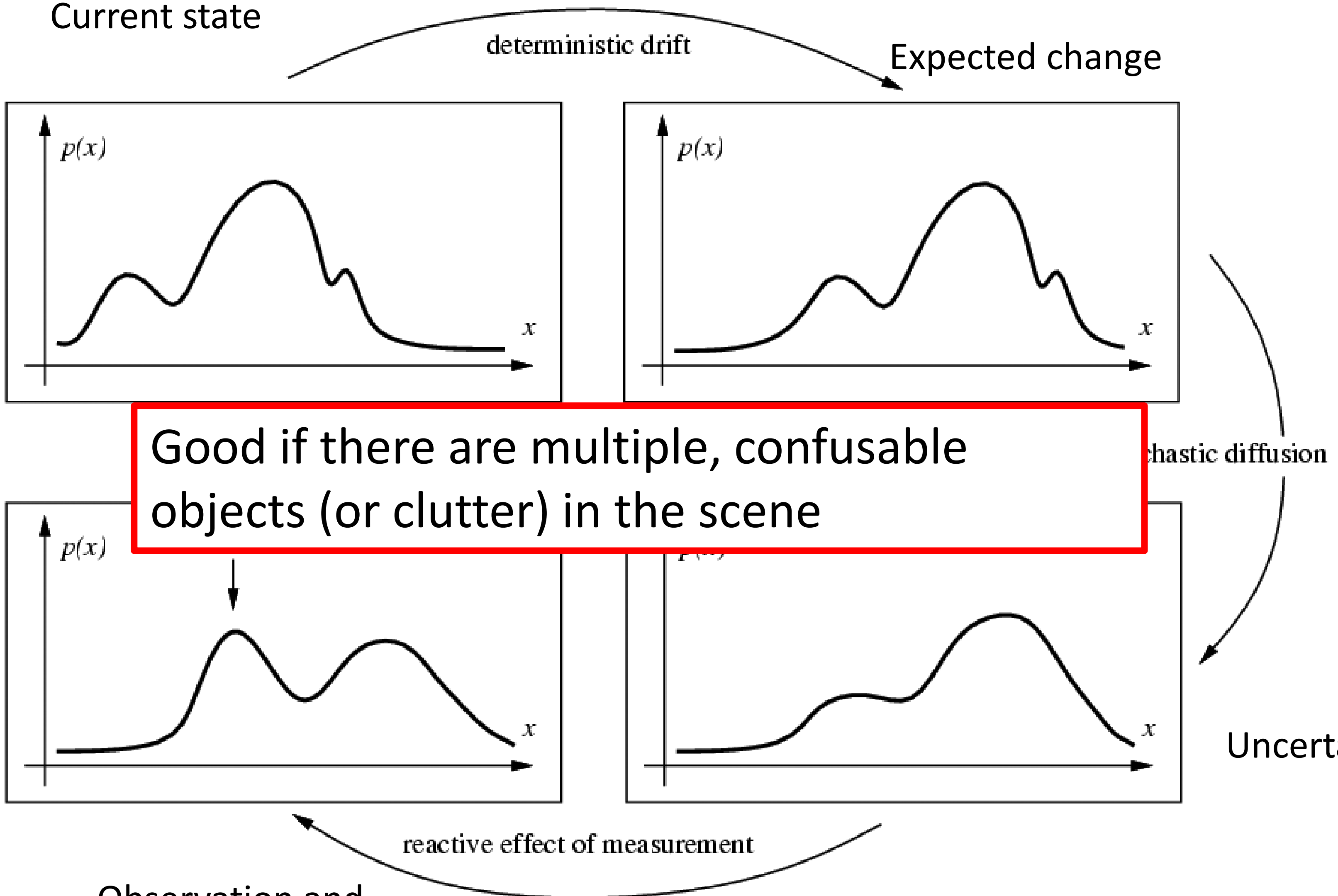
Arrive at corrected density estimate $P(X_t|Y_t)$

M. Isard and A. Blake, CONDENSATION -- conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998

Propagation of non-parametric densities

Current state

Expected change



Good if there are multiple, confusable objects (or clutter) in the scene

Uncertainty

Observation and Correction

Particle filtering results

People: http://www.youtube.com/watch?v=wCMk-pHzScE

Hand: http://www.youtube.com/watch?v=tIjufInUqZM

Localization (similar model): https://www.youtube.com/watch?v=rAuTgsiM2-8
http://www.cvlibs.net/publications/Brubaker2013CVPR.pdf

Good informal explanation: https://www.youtube.com/watch?v=aUkBa1zMKv4

Tracking issues

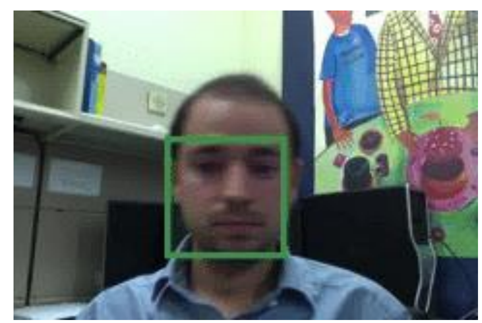
- Initialization
  - Manual
  - Background subtraction
  - Detection

Tracking issues

- Initialization
- Getting observation and dynamics models
  - Observation model: match a template or use a trained detector
  - Dynamics model: usually specify using domain knowledge

Tracking issues

- Initialization
- Obtaining observation and dynamics model
- Uncertainty of prediction vs. correction
  - If the dynamics model is too strong, will end up ignoring the data
  - If the observation model is too strong, tracking is reduced to repeated detection

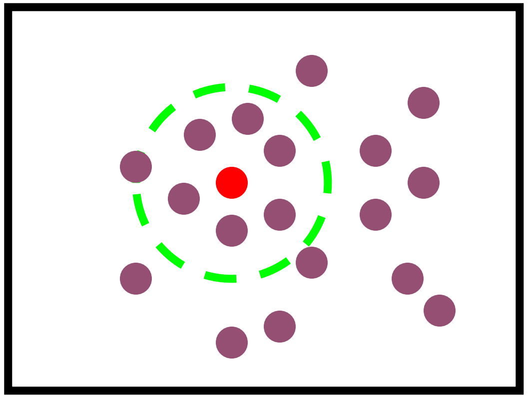


Too strong dynamics model

Too strong observation model

Tracking issues

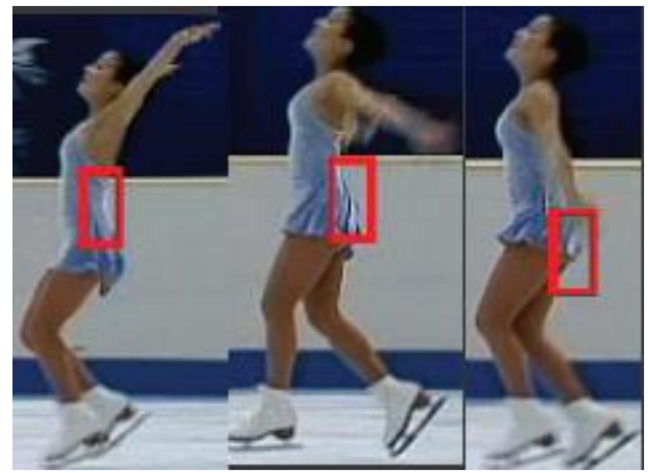
- Initialization
- Getting observation and dynamics models
- Prediction vs. correction
- Data association
  - When tracking multiple objects, need to assign right objects to right tracks (particle filters good for this)



Tracking issues

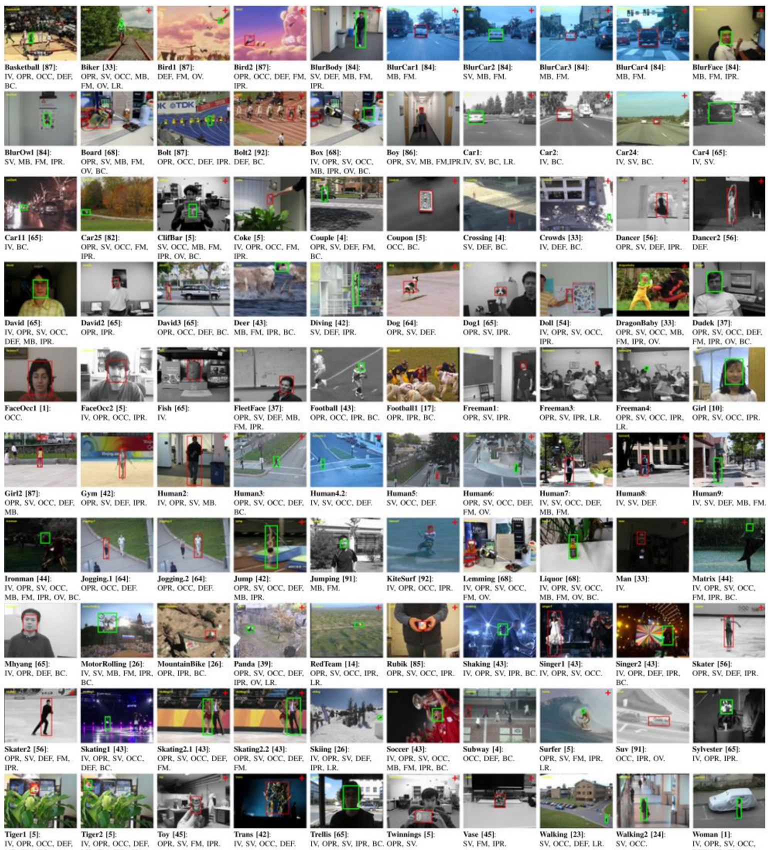
- Initialization
- Getting observation and dynamics models
- Prediction vs. correction
- Data association
- Drift
  - Errors can accumulate over time

Drift



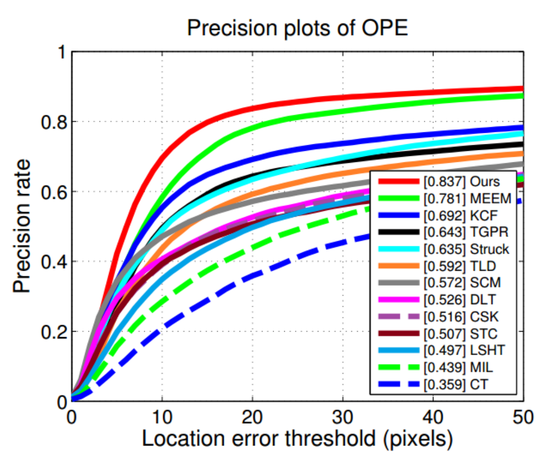
D. Ramanan, D. Forsyth, and A. Zisserman. [Tracking People by Learning their Appearance](). PAMI 2007.

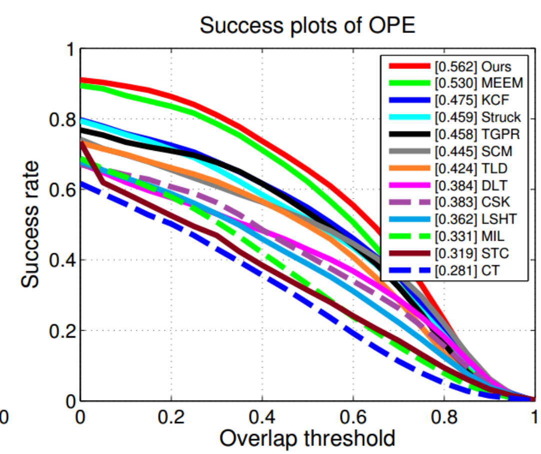
State-of-the-art object tracking



Object tracking benchmark, PAMI15

VOT 2016





Things to remember

- Tracking objects = detection + prediction

- Probabilistic framework
  - Predict next state
  - Update current state based on observation

- Two simple but effective methods
  - Kalman filters: Gaussian distribution
  - Particle filters: multimodal distribution