Visual tracking

Outline

- Visual tracking
- Example Applications

Detection vs. Tracking


t=1


t=2

...

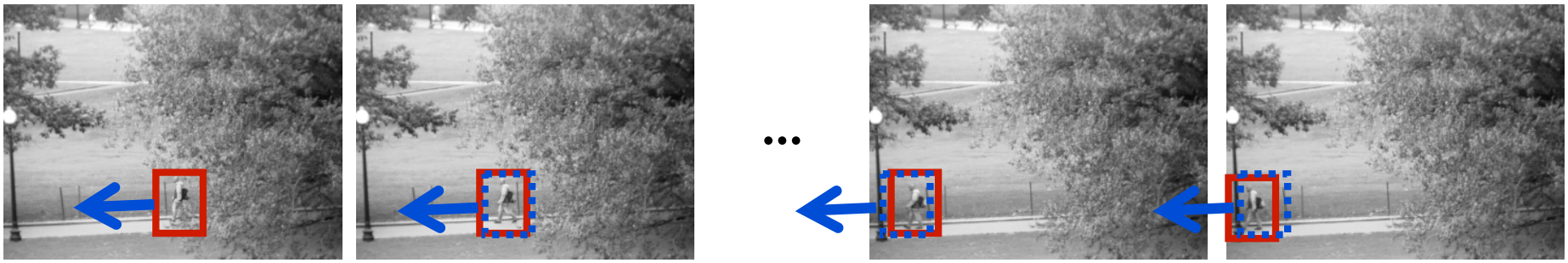

t=20


t=21

Detection vs. Tracking

- Detection
  - We detect the object independently in each frame and can record its position over time, e.g., based on blob's centroid or detection window coordinates.

Detection vs. Tracking

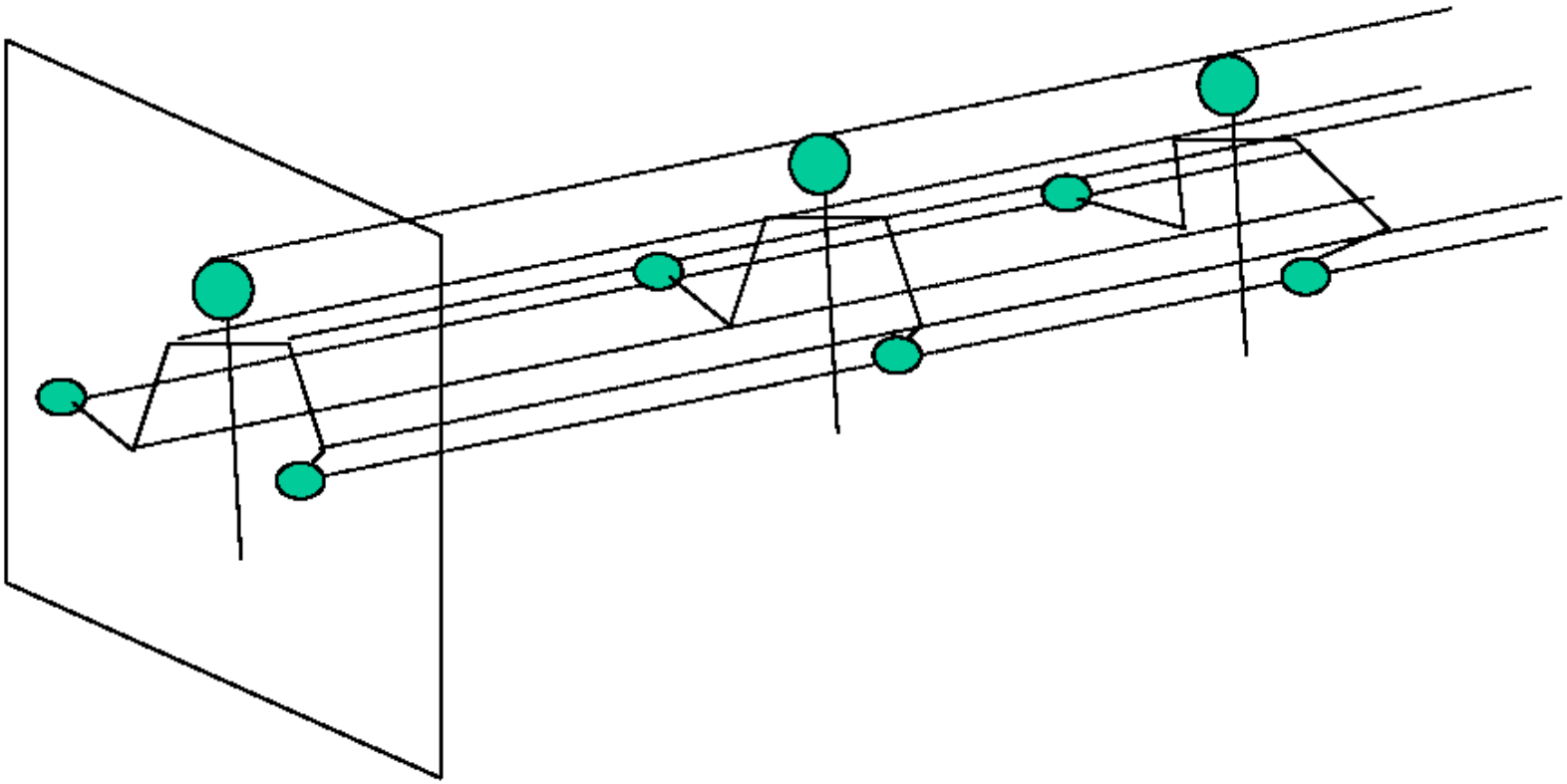


- Tracking with *dynamics*:
  - We use image measurements to estimate the object position, but also incorporate the position predicted by dynamics, i.e., our expectation of the object's motion pattern.

Why is it hard?

Why is it hard?

Geometrically under-constrained



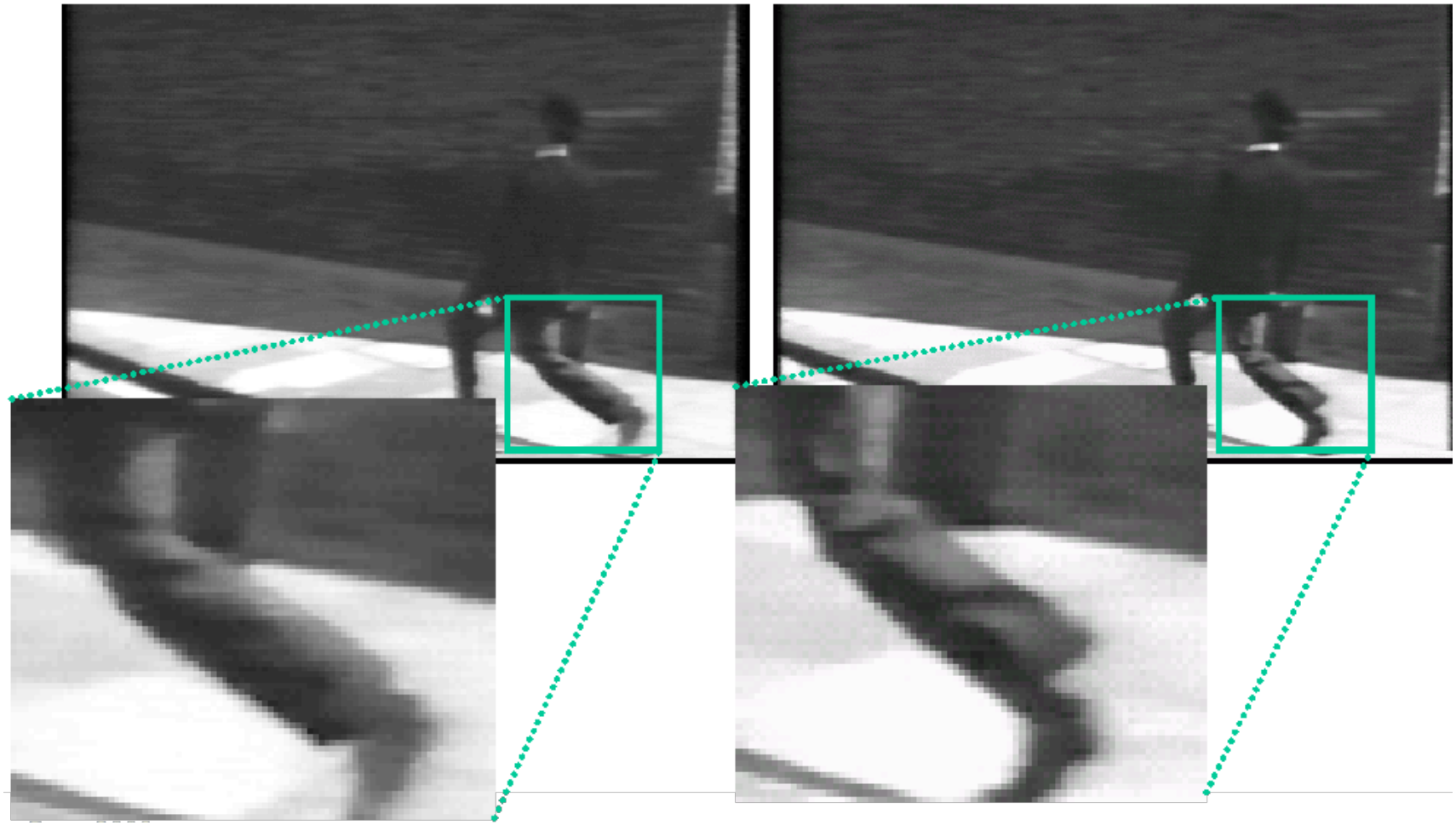
Why is it Difficult?

The appearance of people can vary dramatically

Bones and joints are unobservable (muscle, skin, clothing hide the underlying structure)

Why is it Difficult?

Clothing and Lighting



Large Motions

Long-range motions

(makes search and matching hard)

Motion Blur

(nothing to match)

Ambiguities

Ambiguous Matches

Self Occlusion

Visual Tracking: what are we tracking?

What are we tracking?

- position, velocity, acceleration
- shape, size, deformation
- 3d structure
- …

Which image properties should we track?

- intensity / colours
- region statistics
- contours (edges)
- shapes
- motion

Visual Tracking: simplifying the problem

What simplifies the problem/solution in practice

- known/stationary background (e.g. track blobs)
- distinct a priori colours (e.g. skin)
- multiple cameras (often 2 or 3)
- manual initialisation
- strong dynamics model
- prior knowledge of the number of objects and object types
- sufficient object size
- limited occlusion

Visual Tracking: simplifying the problem

What simplifies the problem/solution in practice

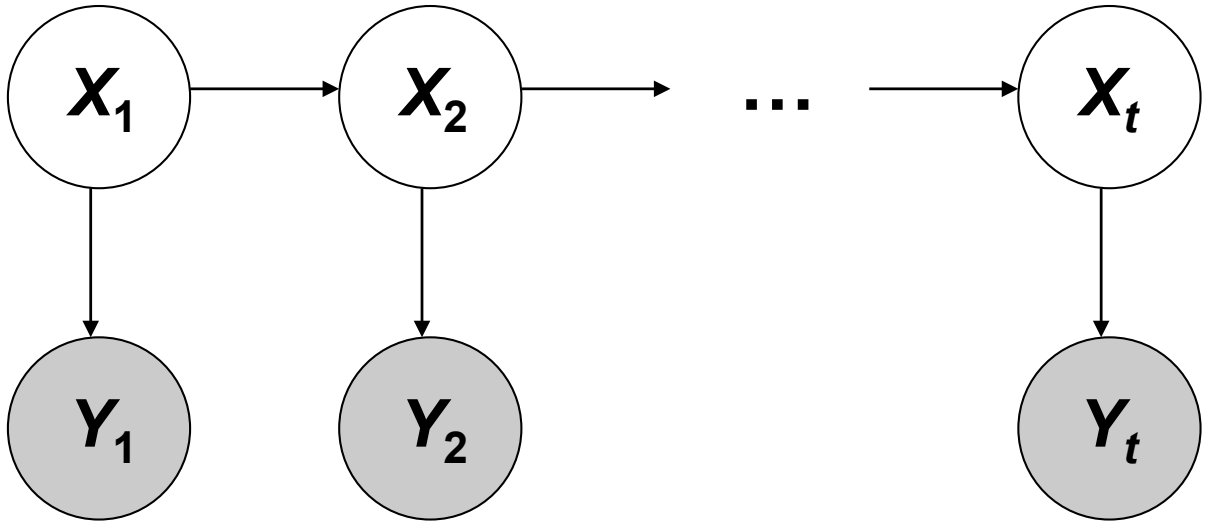
- known/stationary background (e.g. track blobs)
- distinct a priori colours (e.g. skin)
- multiple cameras (often 2 or 3)
- manual initialisation
- strong dynamics model
- prior knowledge of the number of objects and object types
- sufficient object size
- limited occlusion

Tracking with Dynamics

- Key idea
  - Given a model of expected motion, predict where objects will occur in next frame, even before seeing the image.
- Goals
  - Restrict search for the object
  - Improved estimates since measurement noise is reduced by trajectory smoothness.
- Assumption: continuous motion patterns
  - Camera is not moving instantly to new viewpoint.
  - Objects do not disappear and reappear in different places.
  - Gradual change in pose between camera and scene.

General Model for Tracking

- The moving object of interest is characterized by an underlying *state X*
- State *X* gives rise to *measurements* or *observations Y*
- At each time *t*, the state changes to X_t and we get a new observation Y_t



State vs. Observation

- Hidden state : parameters of interest
- Measurement : what we get to directly observe

Tracking as Inference

- The hidden state consists of the true parameters we care about, denoted X.
- The measurement is our noisy observation that results from the underlying state, denoted Y.
- At each time step, state changes (from X_{t-1} to X_t) and we get a new observation Y_t.
- Our goal: recover most likely state X_t given
  - All observations seen so far.
  - Knowledge about dynamics of state transitions.

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.

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

- Tracking can be seen as the process of propagating the posterior distribution of state given measurements across time.

Simplifying Assumptions

- Only the immediate past matters

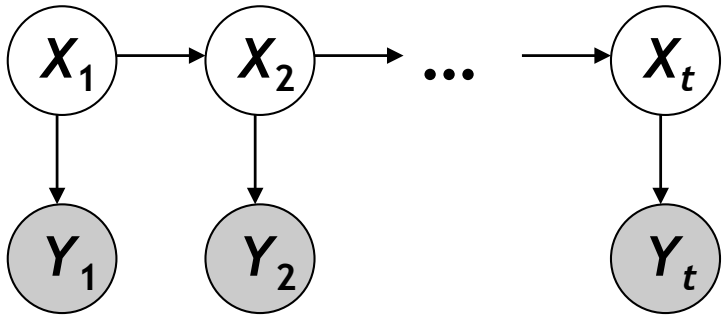
$$P(X_t | X_0, \ldots, X_{t-1}) = P(X_t | X_{t-1})$$

Dynamics model

- Measurements depend only on the current state

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

Observation model



Tracking as Induction

- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$

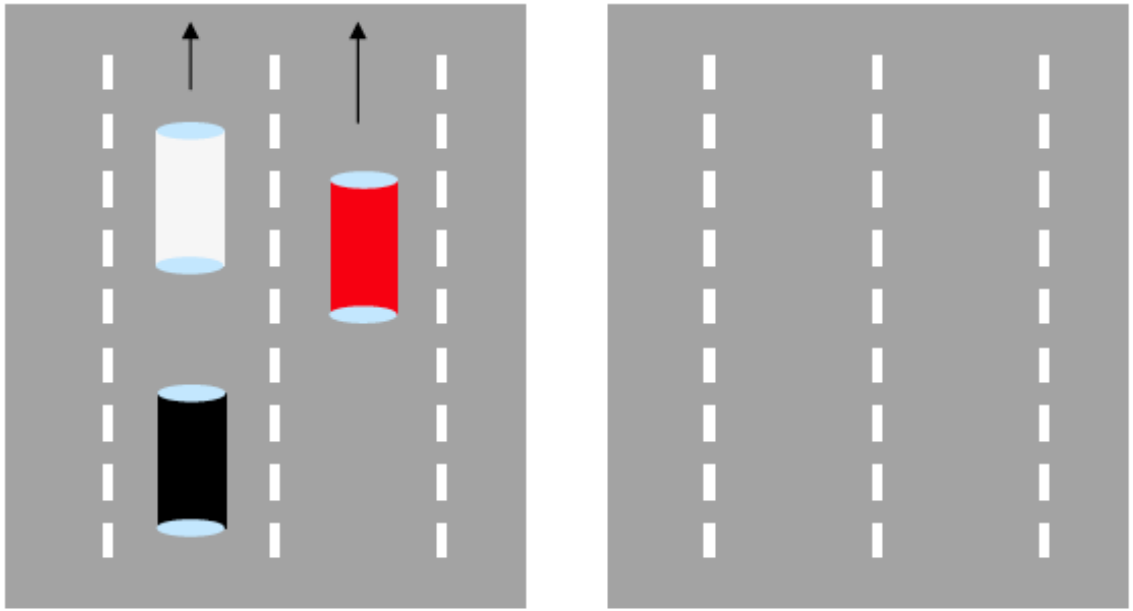
$$P(X_0 \mid Y_0 = y_0) = \frac{P(y_0 \mid X_0)P(X_0)}{P(y_0)} \propto P(y_0 \mid X_0)P(X_0)$$

Posterior prob. of state given measurement

Likelihood of measurement

Prior of the state

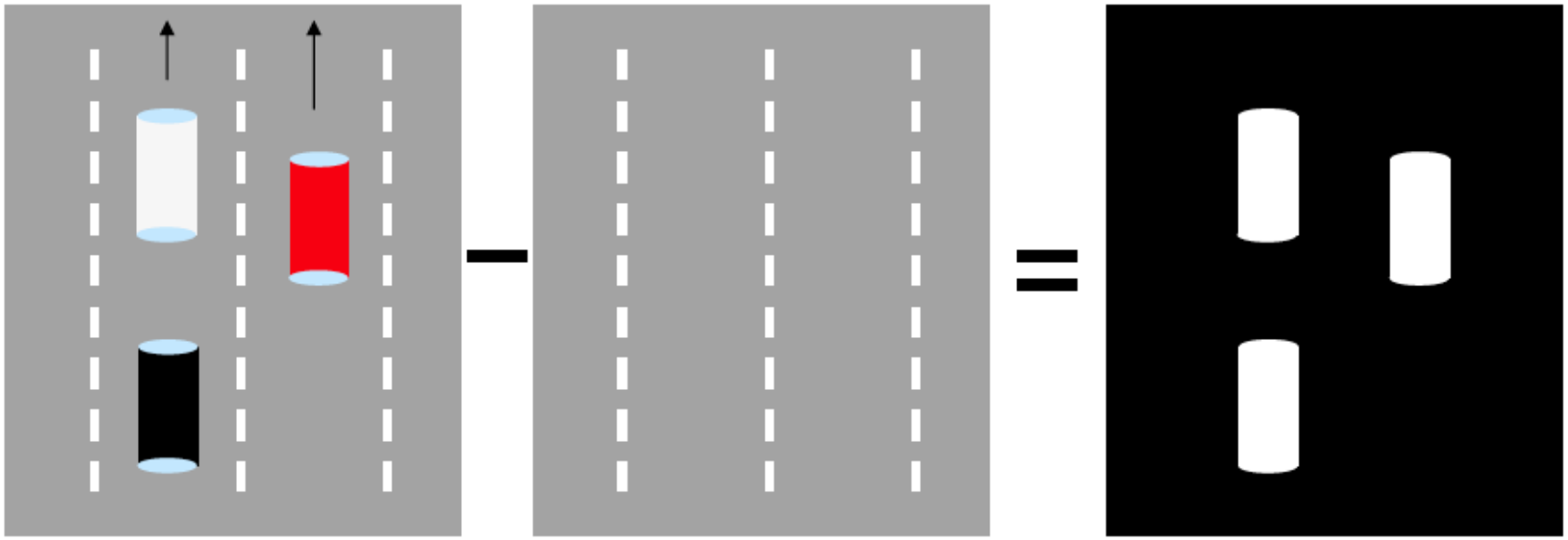
Problem Formulation



Goal: estimate car positions at each time instant

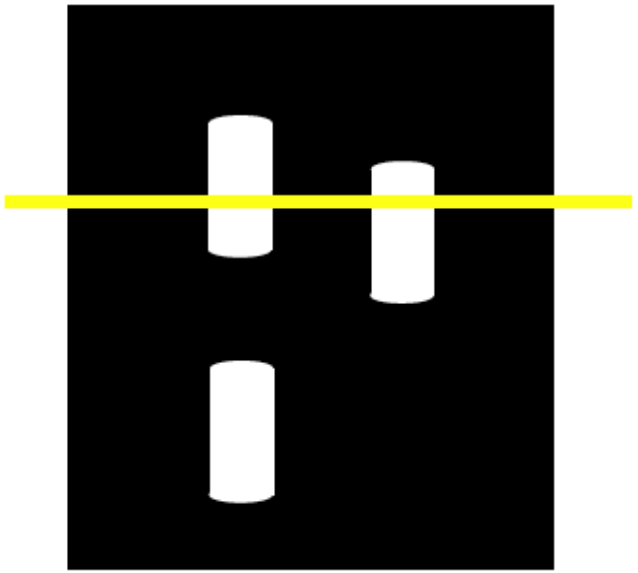
Observations: image sequences and known background

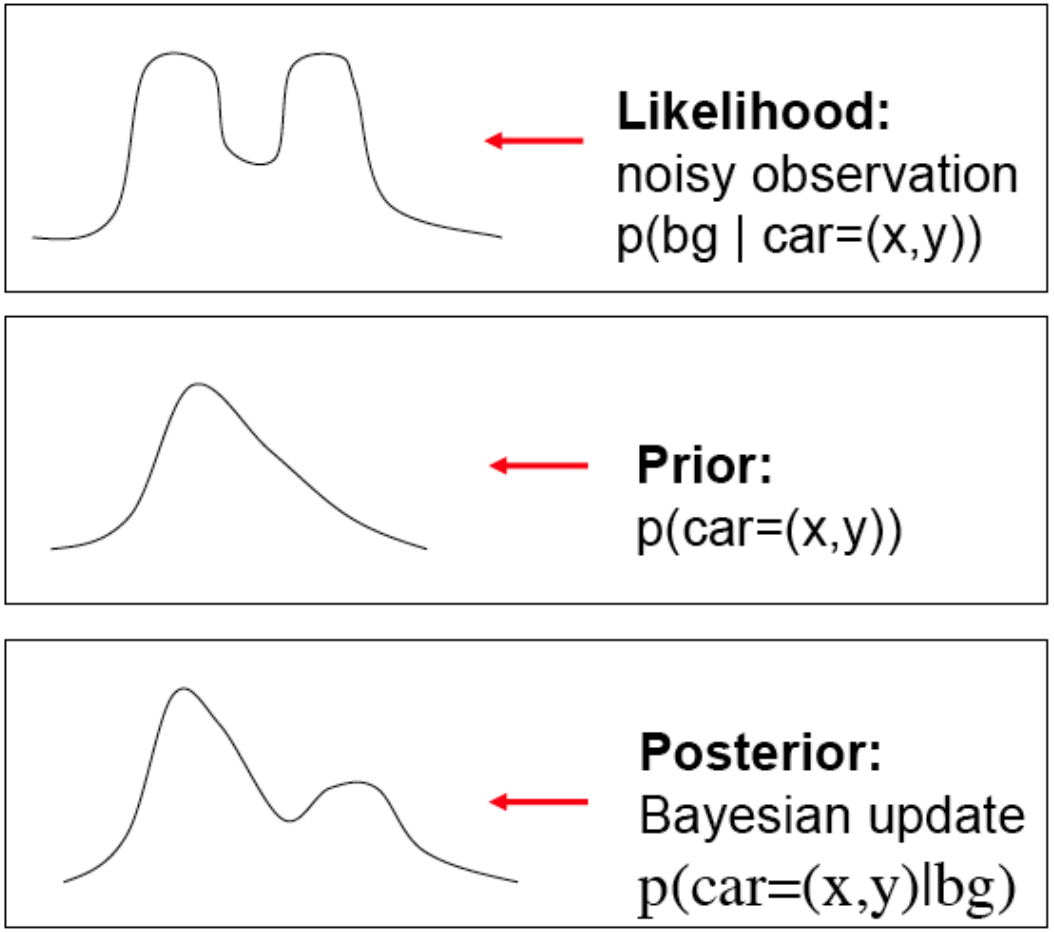
Problem Formulation



Define image likelihood: p(bg | car=(x,y))

Problem Formulation





system states: car positions

observations: images

Tracking as Induction

- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$

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

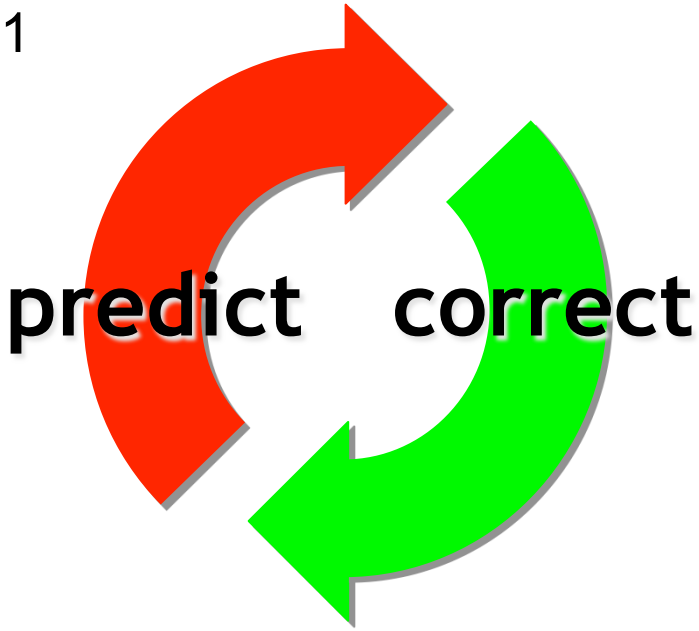
Posterior prob. of state given measurement

Likelihood of measurement

Prior of the state

Tracking as Induction

- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$
- Given corrected estimate for frame *t*:
  - Predict for frame *t*+1
  - Correct for frame *t*+1



Summary: Prediction and Correction

- Prediction:

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

- Correction:

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

The Kalman Filter

- Method for tracking linear dynamical models with Gaussian noise

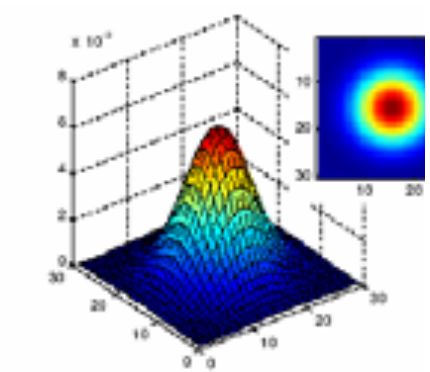
- The predicted/corrected state distributions are Gaussian
  - You only need to maintain the mean and covariance.
  - The calculations are easy (all the integrals can be done in closed form).

Notation Reminder

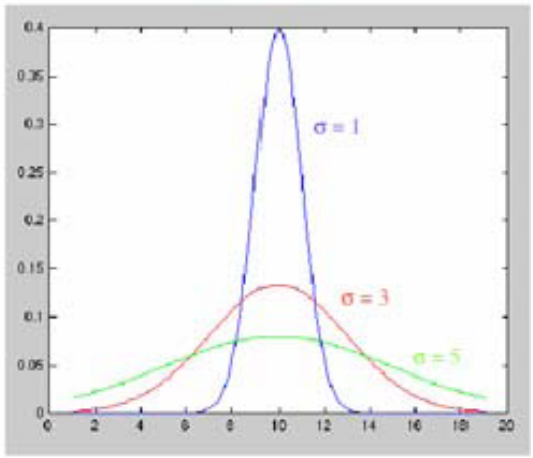
$$\mathbf{x} \sim N(\mu, \Sigma)$$

- Random variable with Gaussian probability distribution that has the mean vector μ and covariance matrix Σ.
- x and μ are *d*-dimensional, Σ is *d* x *d*.

***d*=2**



***d*=1**



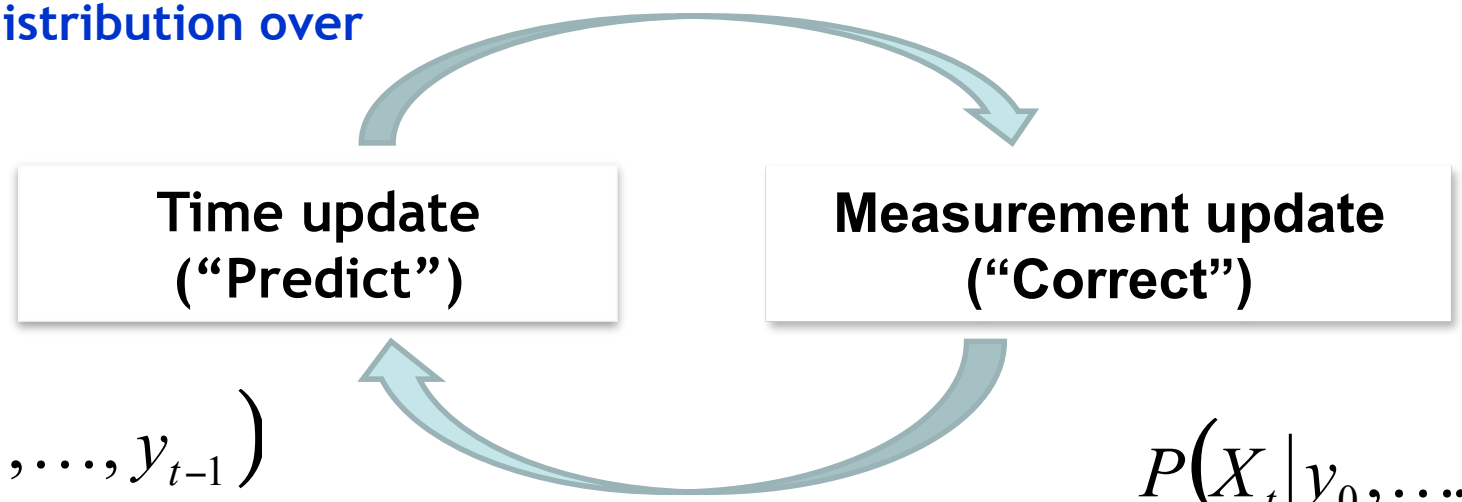
If x is 1D, we just have one Σ parameter: the variance σ^2

The Kalman Filter

Know corrected state from previous time step, and all measurements up to the current one
→ Predict distribution over next state.

Receive measurement

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



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

Mean and std. dev. of predicted state:

μ_t^-, σ_t^-

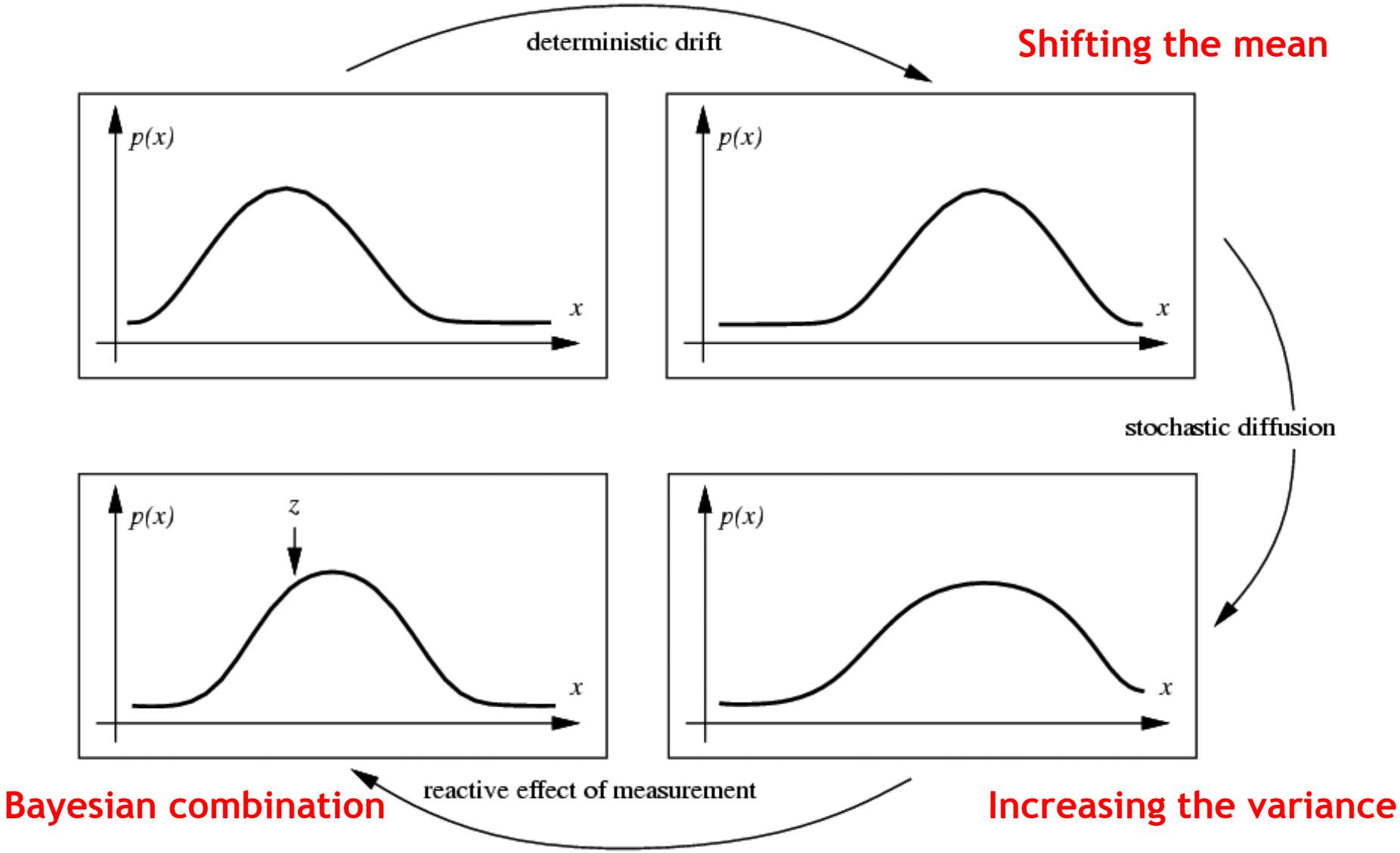
Time advances: t++

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

Mean and std. dev. of corrected state:

μ_t^+, σ_t^+

Propagation of Gaussian densities



Summary: Kalman Filter

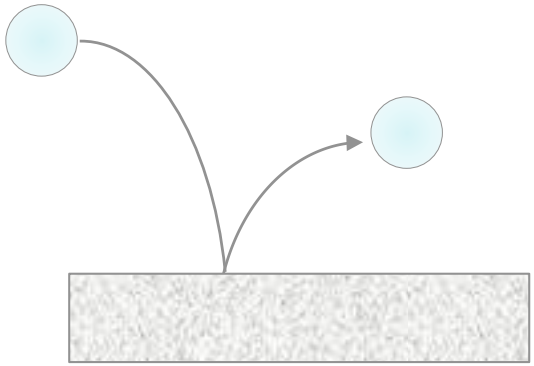
- Pros:
  - Gaussian densities everywhere
  - Simple updates, compact and efficient
  - Very established method, very well understood

- Cons:
  - Unimodal distribution, only single hypothesis
  - Restricted class of motions defined by linear model

Why Is This A Restriction?

- Many interesting cases don't have linear dynamics
  - E.g. pedestrians walking

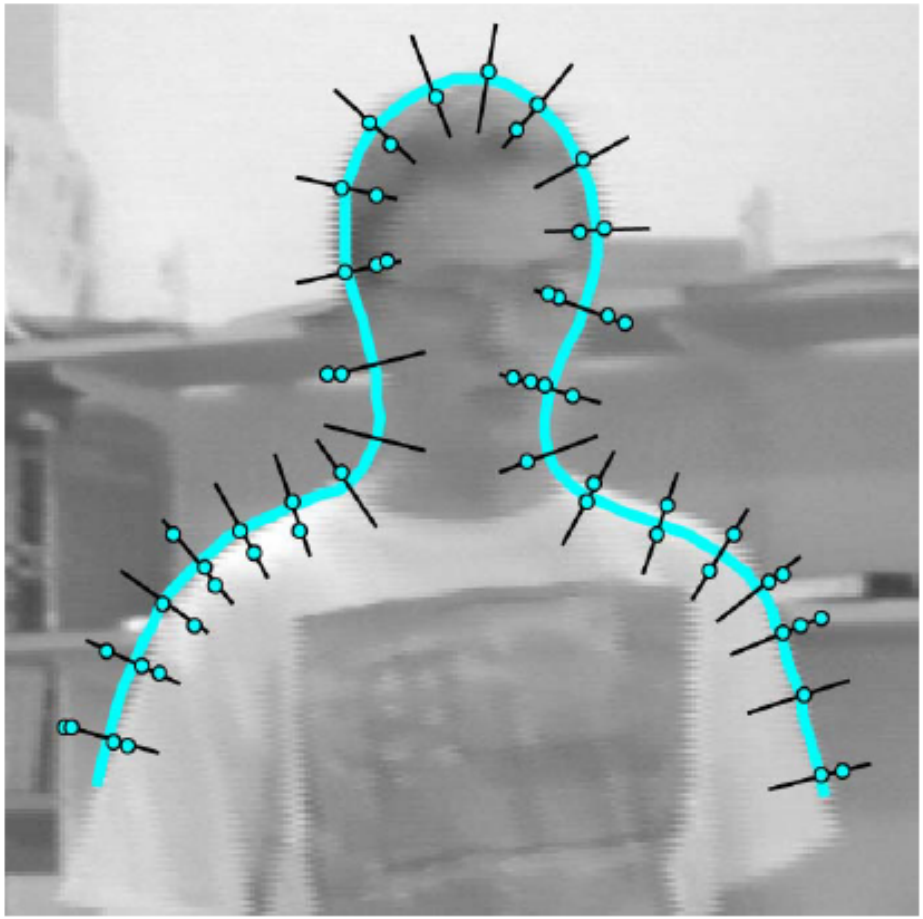
  - E.g. a ball bouncing



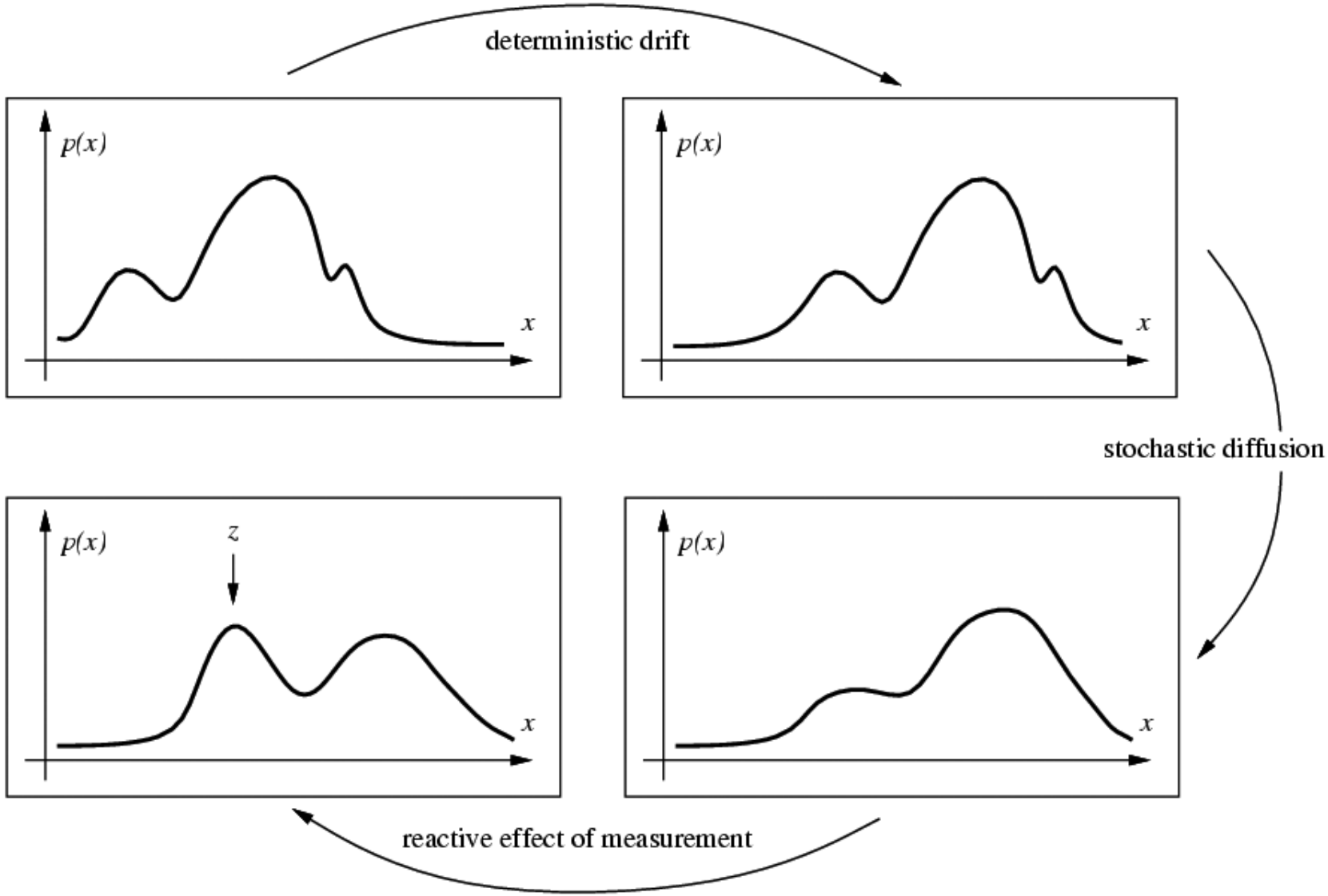
When Is A Single Hypothesis Too Limiting?

Multi-Model Likelihoods

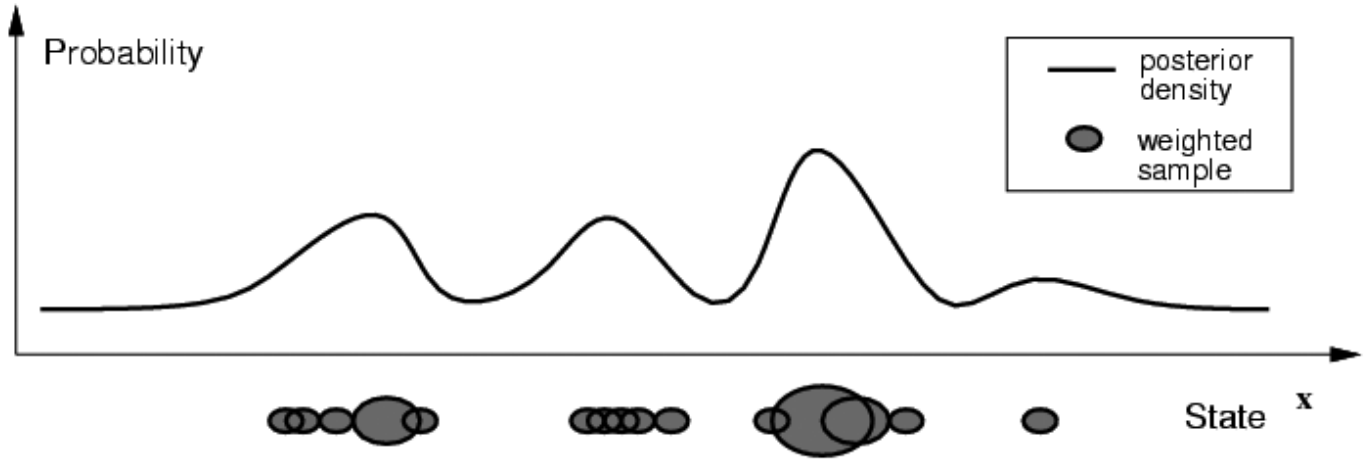
Measurement clutter in natural images causes likelihood functions to have multiple local maxima



Propagation of General Densities



Factored Sampling



- Idea: Represent state distribution non-parametrically
  - Prediction: Sample points from prior density for the state, $P(X)$
  - Correction: Weight the samples according to $P(Y|X)$

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

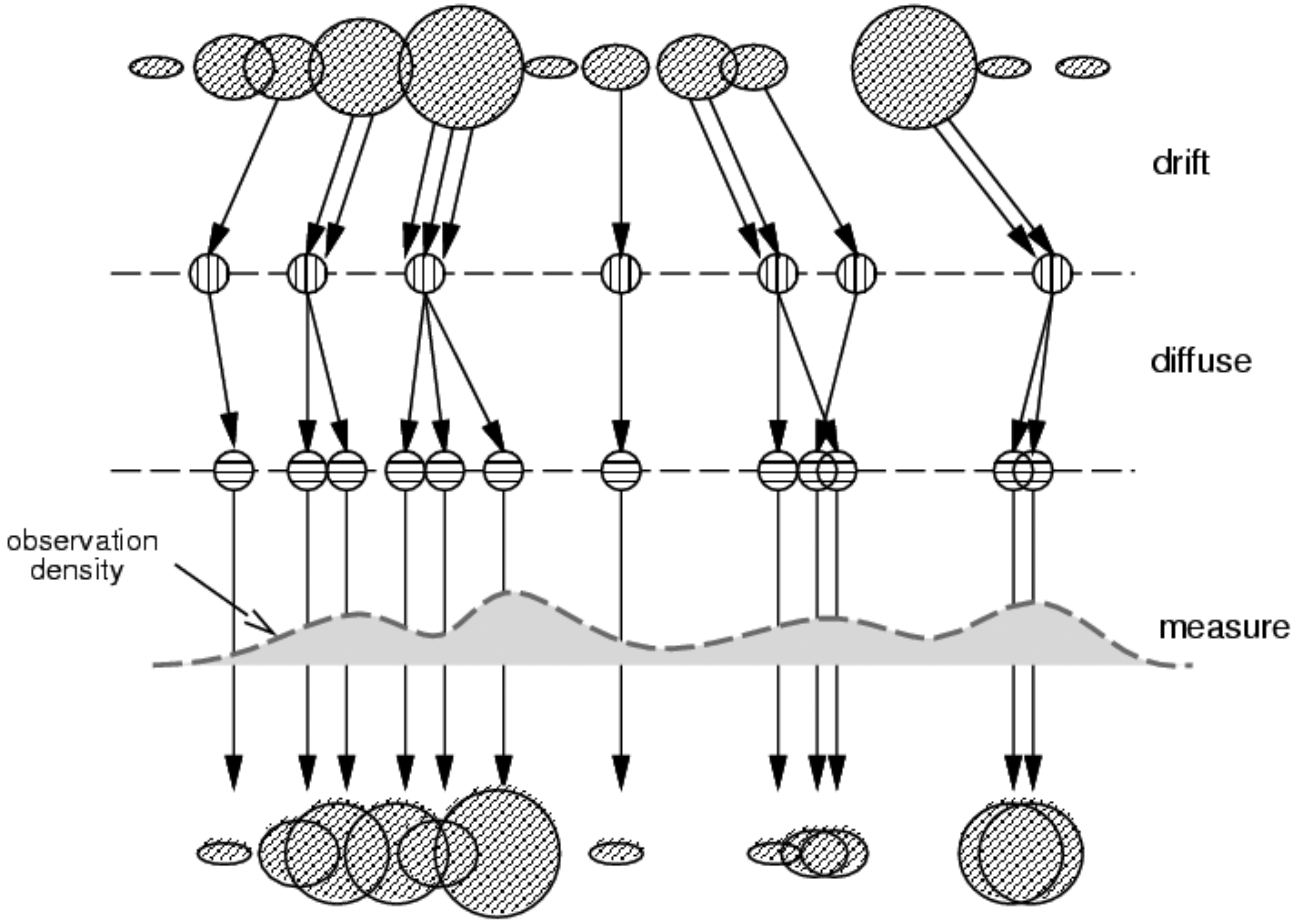
M. Isard and A. Blake,
CONDENSATION -- conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998

Particle Filtering

- (Also known as Sequential Monte Carlo Methods)
- We want to use sampling to propagate densities over time (i.e., across frames in a video sequence).
- At each time step, represent posterior $P(X_t|Y_t)$ with weighted sample set.
- Previous time step's sample set $P(X_{t-1}|Y_{t-1})$ is passed to next time step as the effective prior.

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, 199

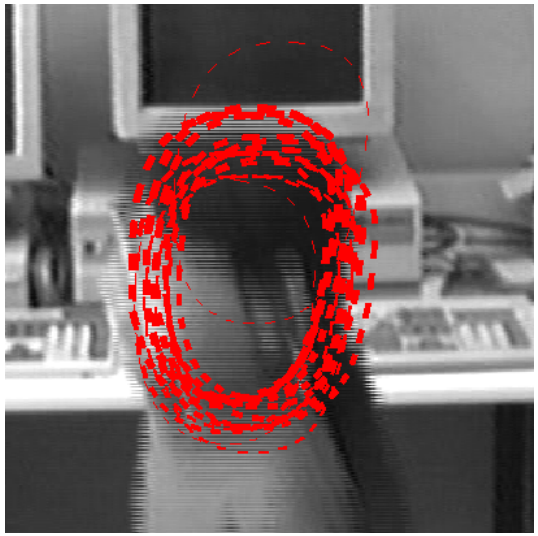
Particle Filter

Tracking in Clutter

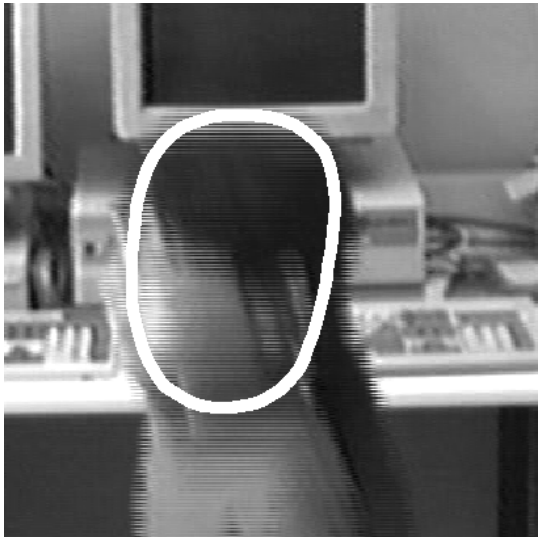
Obtaining a State Estimate

- Note that there's no explicit state estimate maintained—just a "cloud" of particles
- Can obtain an estimate at a particular time by querying the current particle set
- Some approaches
  - "Mean" particle
    - Weighted sum of particles
    - Confidence: inverse variance
  - Really want a mode finder—mean of tallest peak

Condensation: Estimating Target State



State samples
(thickness proportional to weight)



Mean of weighted state samples

From *Isard & Blake,1998*

Summary: Particle Filtering

- Pros:
  - Able to represent arbitrary densities
  - Converging to true posterior even for non-Gaussian and nonlinear system
  - Efficient: particles tend to focus on regions with high probability
  - Works with many different state spaces
    - E.g. articulated tracking in complicated joint angle spaces
  - Many extensions available

Summary: Particle Filtering

- Cons / Caveats:
  - #Particles is important performance factor
    - Want as few particles as possible for efficiency.
    - But need to cover state space sufficiently well.
  - Worst-case complexity grows exponentially in the dimensions
  - Multimodal densities possible, but still single object
    - Interactions between multiple objects require special treatment.
    - Not handled well in the particle filtering framework (state space explosion).

Recap so far

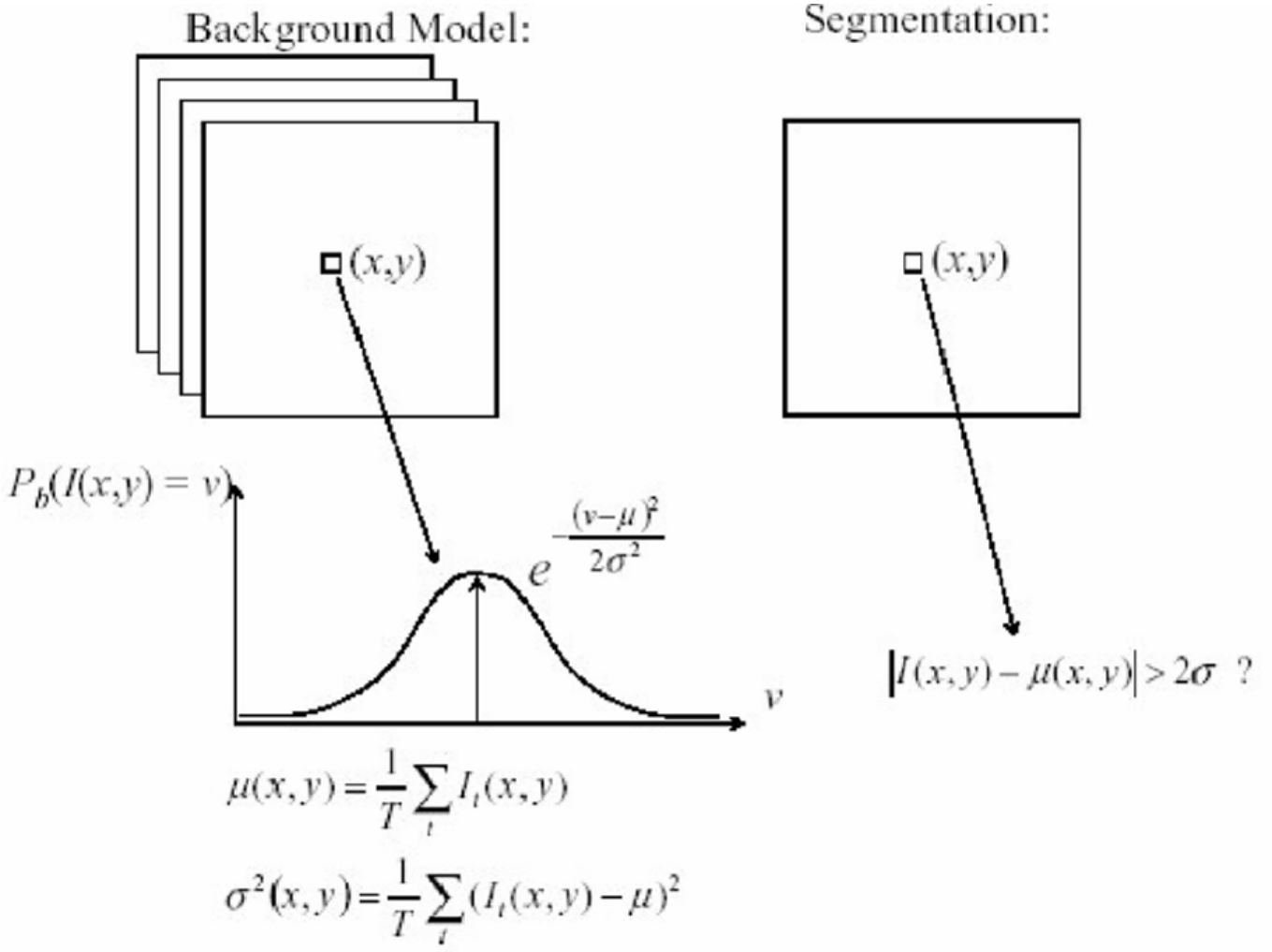
- Recap: Kalman Filter
- Particle Filters
  - Propagation of general densities
  - Factored sampling
- Obtaining the Observations
  - Background modelling
- Issues in Tracking
- Examples: Articulated Tracking

Obtaining the Observations

- Up to now, I've only talked about the tracking part.
- How do we get the object measurements?

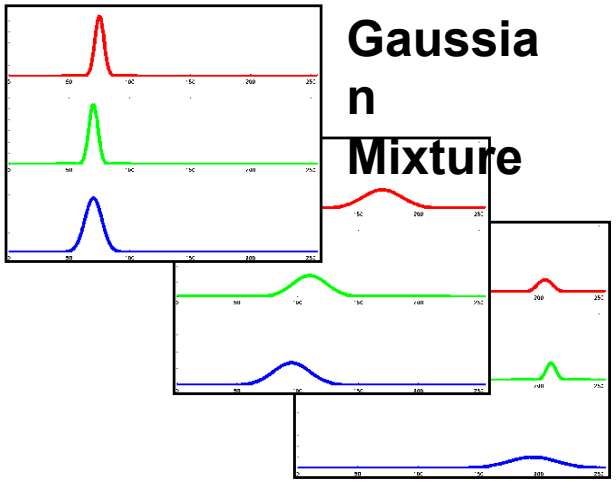
- Some possible approaches
  - Background modelling
  - Tracking lines & contours
  - Tracking-by-detection

Background Model



Background Color Model

- Gaussian mixture for each pixel
  - Model “common“ appearance variation for each pixel separately.
  - Update the mixtures over time
    - Adapt to lighting changes, etc.
  - Easiest when an empty scene is first seen for some frames.
  - But can also be applied for online learning.

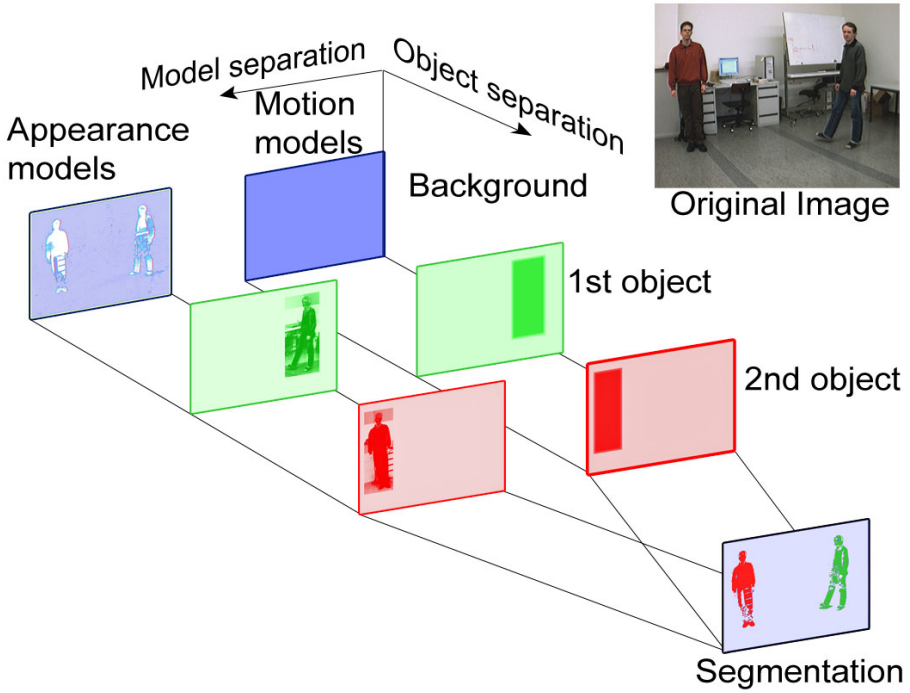


- De-facto standard for many tracking applications
  - With fixed-camera scenarios
  - With limited background motion

C. Stauffer, E. Grimson, Learning Patterns of Activity Using Real-Time Tracking, *IEEE Trans. PAMI*, 22(8):747-757, 2000.

Background Modeling for Tracking

- Segmentation based tracking
- Object separation
  - Specialized models for each foreground object and for the background
- Model separation
  - Appearance
  - Motion (Tracking)

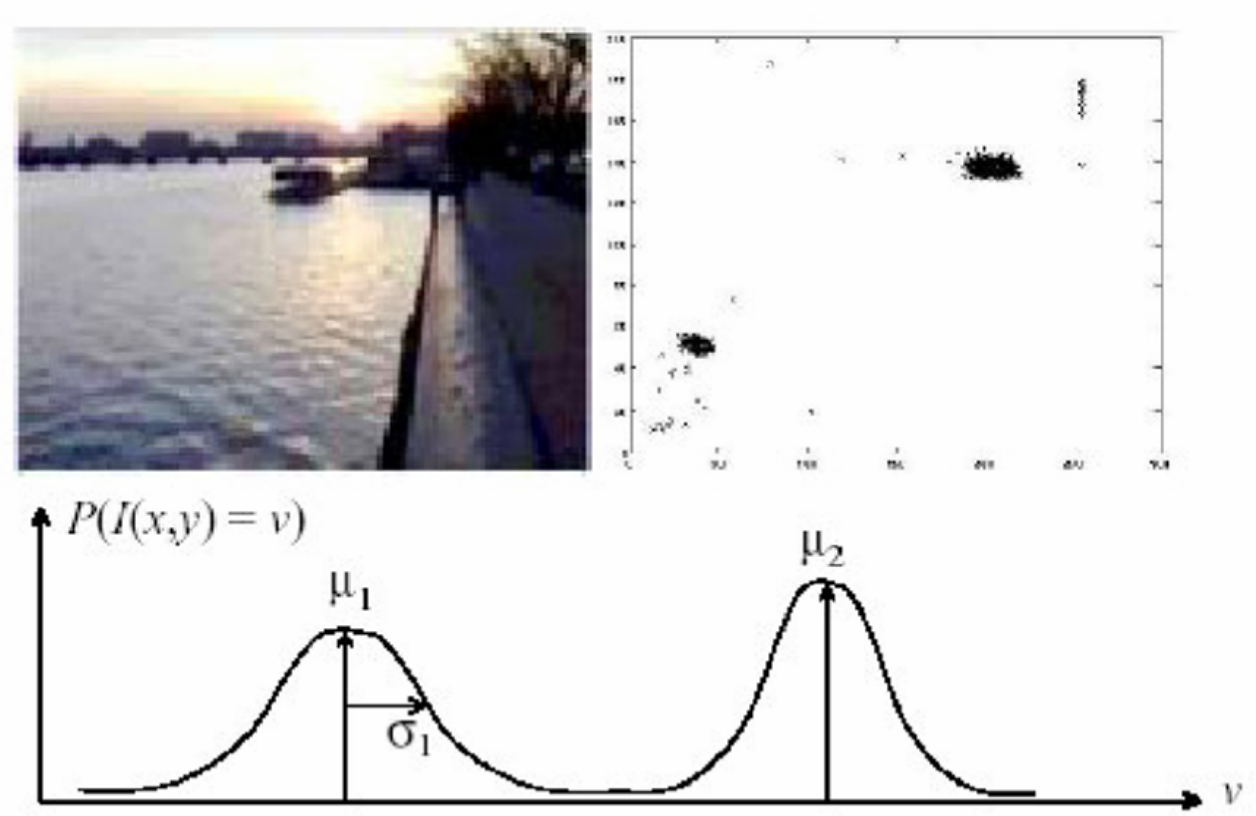


- Bayesian per-pixel classification

$$P_{posterior}(object|pixel) \propto P_{prior}(object)P(pixel|object)$$

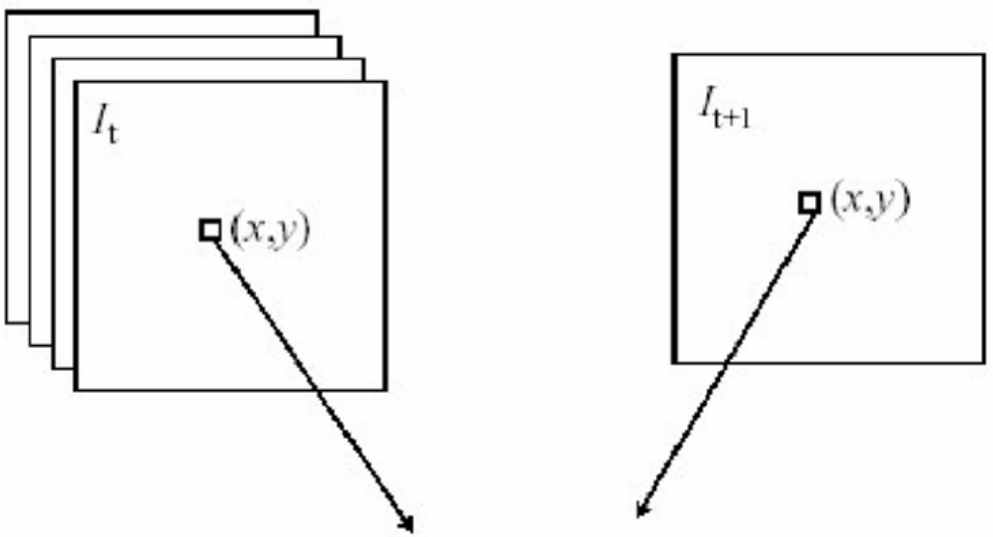
Slide adapted from Daniel Roth

Mixture Models



$$P(I(x,y)=v)=\sum_i w_i e^{-\frac{(v-\mu_i)^2}{2\sigma_i^2}}$$

Linear Prediction and Adaptation



model at time t+1 = (1-α) (model at time t) + α (new data)

α = "learning rate"

Gaussian model:

$$\rho = \alpha P_b(v_{t+1})$$

$$\mu_{t+1} = (1-\rho)\mu_t + \rho v_{t+1}$$

$$\sigma^2_{t+1} = (1-\rho)\sigma^2_t + \rho(v_{t+1} - \mu_{t+1})^2$$

Recap

- Kalman Filter
- Particle Filters
- Obtaining the Observations
  - Background modelling
- Issues in Tracking
- Articulated Tracking

Summary: Tracking Issues

- Initialization
  - Manual
  - Background subtraction
  - Detection

Summary: Tracking Issues

- Initialization
- Obtaining observation and dynamics model
  - Generative observation model: “render” the state on top of the image and compare
  - Discriminative observation model: classifier or detector score
  - Dynamics model: learn (very difficult) or specify using domain knowledge

Summary: Tracking Issues

- Initialization
- Obtaining observation and dynamics model
- 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

Summary: Tracking Issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Nonlinear dynamics
  - Sometimes needed to keep multiple trackers in parallel
  - E.g. to model different driver behavior
  - Or for abrupt direction changes

Summary: Tracking Issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Nonlinear dynamics
- Data association
  - What if we don't know which measurements to associate with which tracks?
  - Especially problematic in case of occlusions

Data Association

- So far, we've assumed the entire measurement to be relevant to determining the state.
- In reality, there may be uninformative measurements (clutter) or measurements may belong to different tracked objects.
- Data association: task of determining which measurements go with which tracks.

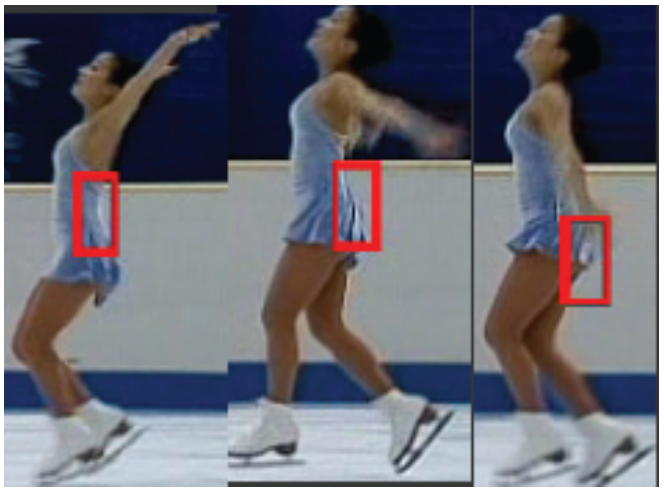
Data Association

- Simple strategy: only pay attention to the measurement that is “closest” to the prediction.
- More sophisticated strategy: keep track of multiple state/ observation hypotheses
  - Can be done with particle filtering
  - But leads to state space explosion (many particles needed)
- This is a general problem in computer vision, there is no easy solution.

Summary: Tracking Issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Nonlinear dynamics
- Data association
- Drift
  - Errors caused by dynamical model, observation model, and data association tend to accumulate over time

Drift



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

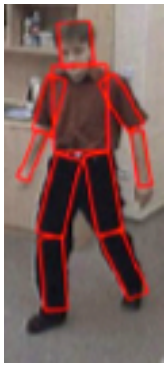
Slide credit: Svetlana Lazebnik

So far …

- Kalman Filter
- Particle Filters
- Obtaining the Observations
- Issues in Tracking
- Articulated Tracking (example applications …)

Articulated Tracking

- Goal
  - Recover a person's body articulation
  - Detailed parametrization in terms of joint locations or joint angles



- Two basic classes of approaches
  - Articulated tracking as high-dimensional inference
  - Part-based models