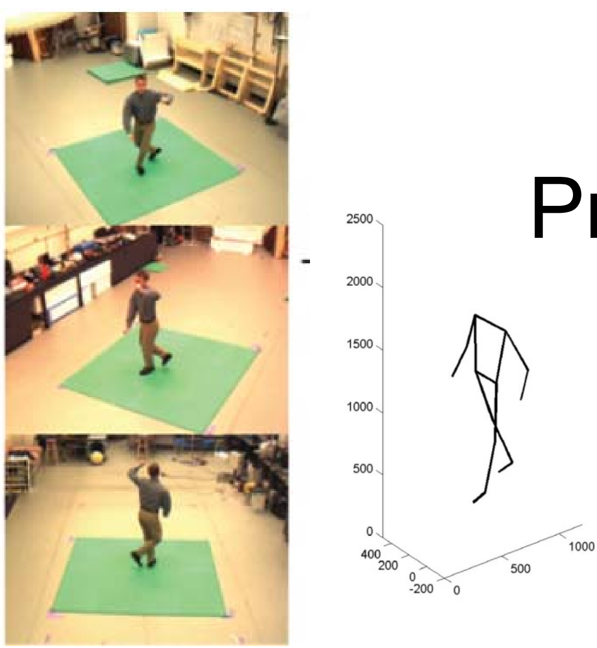
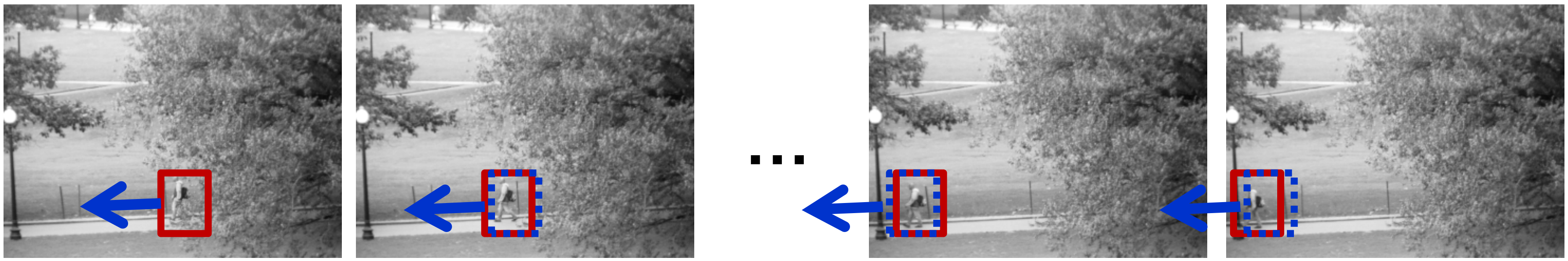
Lecture 21: Motion and tracking



Thursday, Nov 29

Prof. Kristen Grauman

Detection vs. tracking



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

Tracking with dynamics

- Have a model of expected motion
- Given that, predict where objects will occur in next frame, even before seeing the image
- Intent:
  - do less work looking for the object, restrict search
  - improved estimates since measurement noise tempered by trajectory smoothness

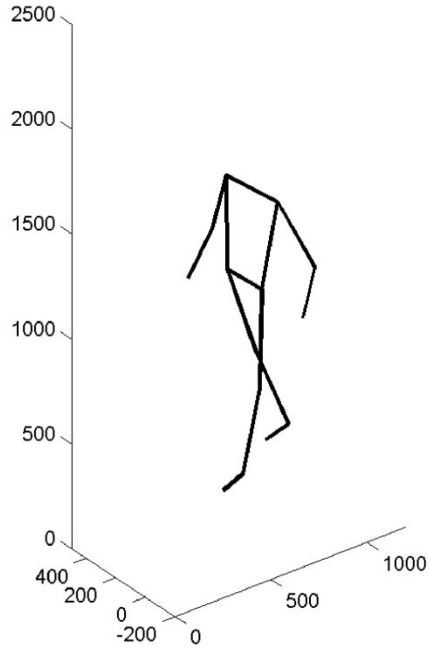
Tracking as inference: Bayes Filters

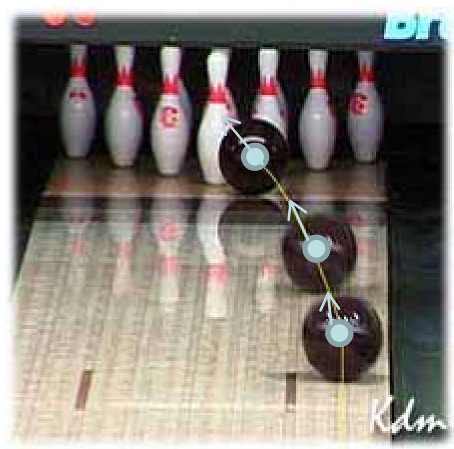
- Hidden state $\boldsymbol{x}_t$
  - The unknown true parameters
  - E.g., actual position of the person we are tracking

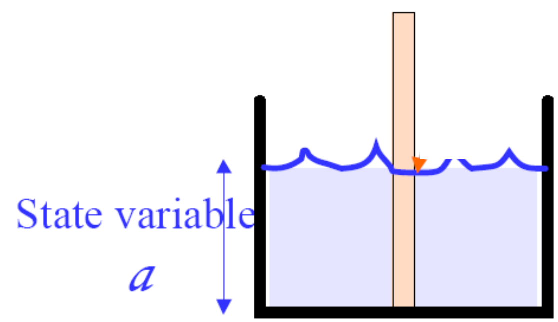
- Measurement $\boldsymbol{y}_t$
  - Our noisy observation of the state
  - E.g., detected blob's centroid

- Can we calculate $\boldsymbol{p(x_t \mid y_1, y_2, \ldots, y_t)}$?
  - Want to recover the state from the observed measurements

States and observations





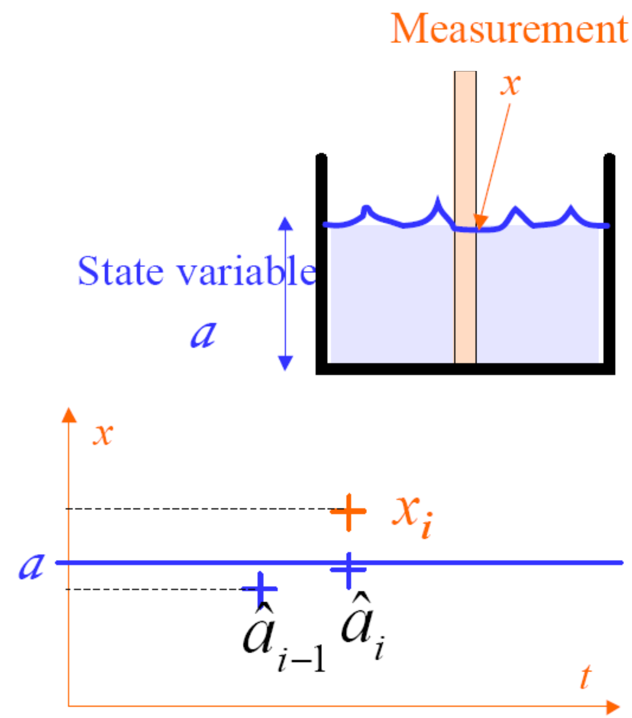


Hidden state is the list of parameters of interest

Measurement is what we get to directly observe (in the images)

Recursive estimation

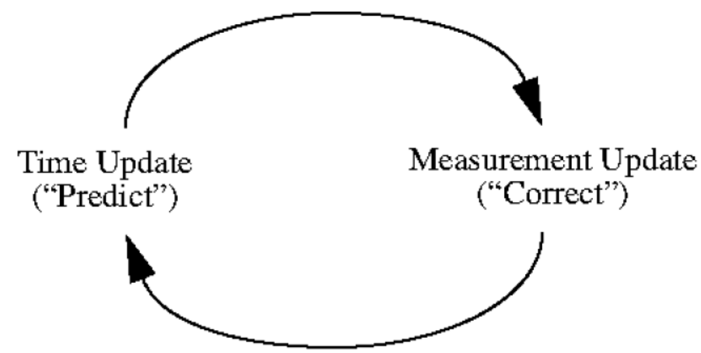
- Unlike a *batch* fitting process, decompose estimation problem into
  - Part that depends on new observation
  - Part that can be computed from previous history
- For tracking, essential given typical goal of real-time processing.



Example from last time: running average

Tracking as inference

- Recursive process:
  - Assume we have initial prior that *predicts* state in absence of any evidence: $P(\mathbf{X}_0)$
  - At the first frame, *correct* this given the value of $\mathbf{Y}_0=\mathbf{y}_0$
  - Given corrected estimate for frame *t*
    - Predict for frame *t*+1
    - Correct for frame *t*+1



Tracking as inference

- Prediction:
  - Given the measurements we have seen up to this point, what state should we predict?

$$P(\boldsymbol{X}_i|\boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1}).$$

- Correction:
  - Now given the current measurement, what state should we predict?

$$P(\boldsymbol{X}_i|\boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$$

Independence assumptions

- Only immediate past state influences current state

$$P(\boldsymbol{X}_i|\boldsymbol{X}_1,\ldots,\boldsymbol{X}_{i-1}) = P(\boldsymbol{X}_i|\boldsymbol{X}_{i-1})$$

- Measurements at time t only depend on the current state

$$P(\boldsymbol{Y}_i,\boldsymbol{Y}_j,\ldots\boldsymbol{Y}_k|\boldsymbol{X}_i) = P(\boldsymbol{Y}_i|\boldsymbol{X}_i)P(\boldsymbol{Y}_j,\ldots,\boldsymbol{Y}_k|\boldsymbol{X}_i)$$

Tracking as inference

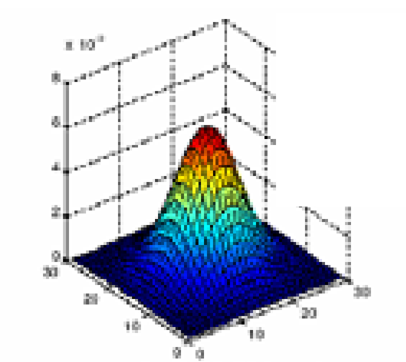
- Goal is then to
  - choose good model for the prediction and correction distributions
  - use the updates to compute best estimate of state
    - Prior to seeing measurement
    - After seeing the measurement

Gaussian distributions, notation

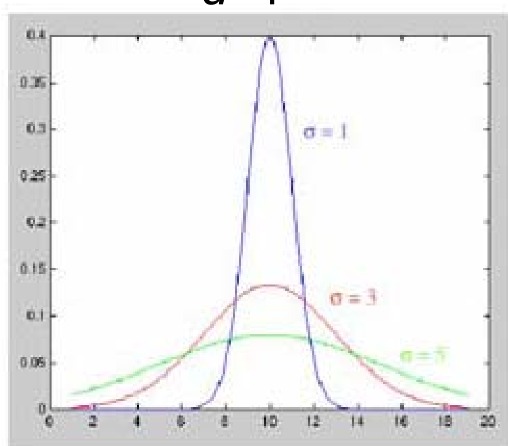
$$\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- 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



Linear dynamic model

- Describe the *a priori* knowledge about
  - System dynamics model: represents evolution of state over time, with noise

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

n x 1 n x n n x 1

  - Measurement model: at every time step we get a noisy measurement of the state

$$\mathbf{y}_t \sim N(\mathbf{M}\mathbf{x}_t; \mathbf{\Sigma}_m)$$

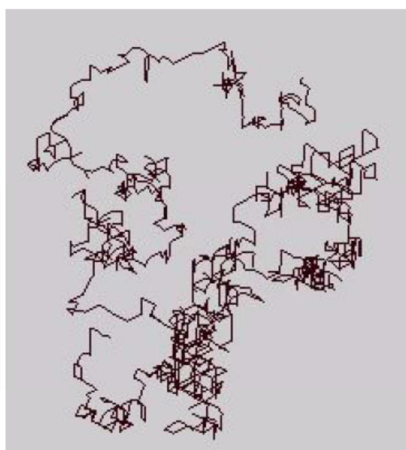
m x 1 m x n n x 1

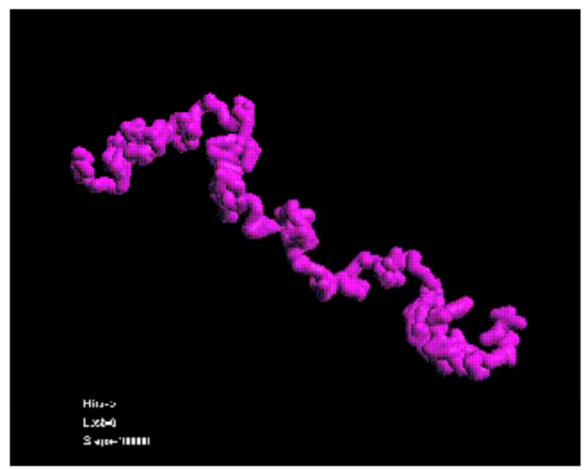
Example: randomly drifting points

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

$$\mathbf{y}_t \sim N(\mathbf{M}\mathbf{x}_t;\boldsymbol{\Sigma}_m)$$

- Consider a stationary object, with state as position
- State evolution is described by identity matrix **D**=$\mathbf{I}$
- Position is constant, only motion due to random noise term.





cic.nist.gov/lipman/sciviz/images/random3.gif
http://www.grunch.net/synergetics/images/ranc
3.jpg

Example: constant velocity

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

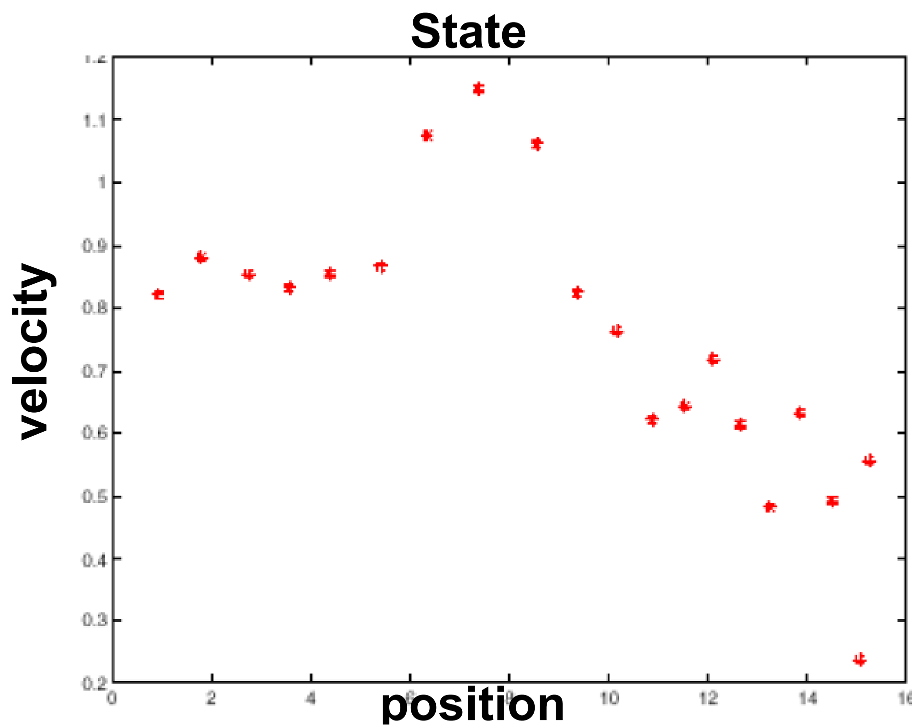
$$\mathbf{y}_t \sim N(\mathbf{M}\mathbf{x}_t; \mathbf{\Sigma}_m)$$

- State vector **x** is 1d position and velocity.
- Measurement **y** is position only.

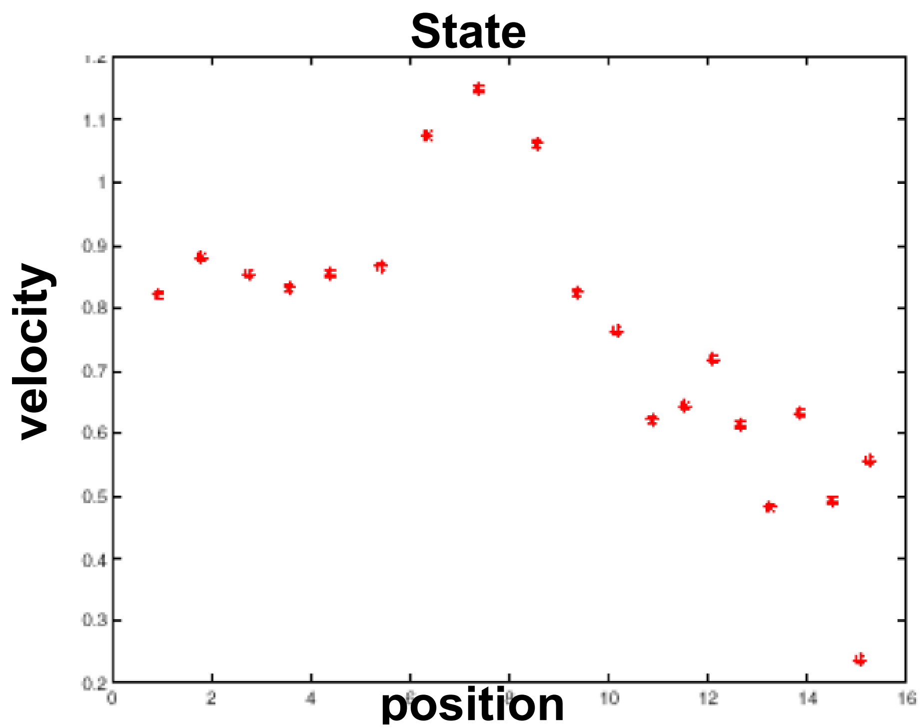
$$\left.\begin{aligned} p_t &= p_{t-1} + (\Delta t)v_{t-1} + \xi \\ v_t &= v_{t-1} + \zeta \end{aligned}\right\} \mathbf{x}_t = \begin{bmatrix} p \\ v \end{bmatrix}_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}_{t-1} + noise$$

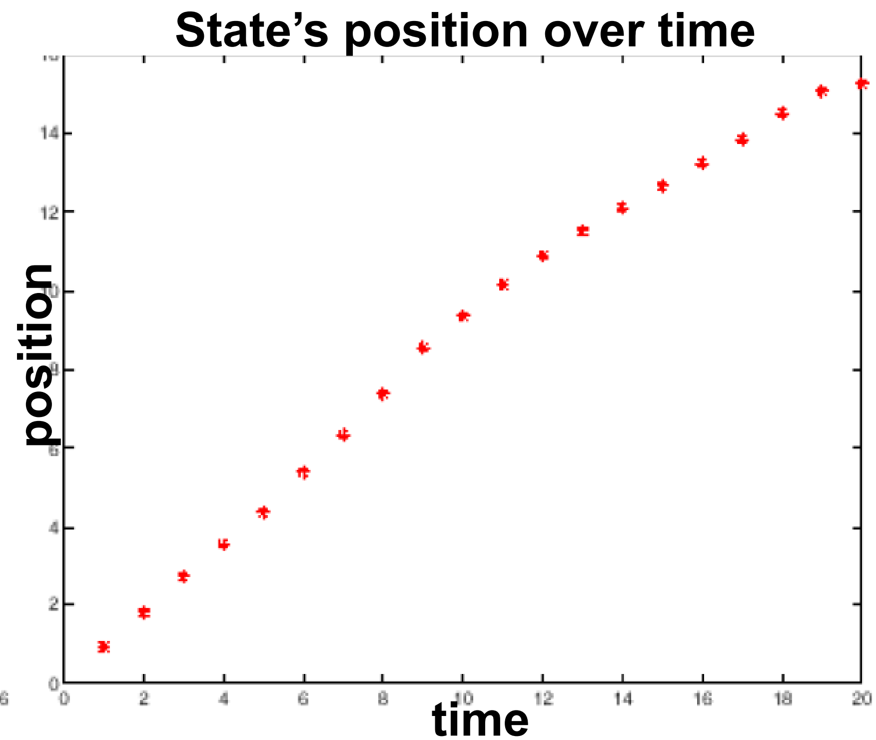
$$\mathbf{y}_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}_t + \xi = p_t + \xi$$

$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad \overbrace{\mathbf{M} = \begin{bmatrix} 1 & 0 \end{bmatrix}}$$

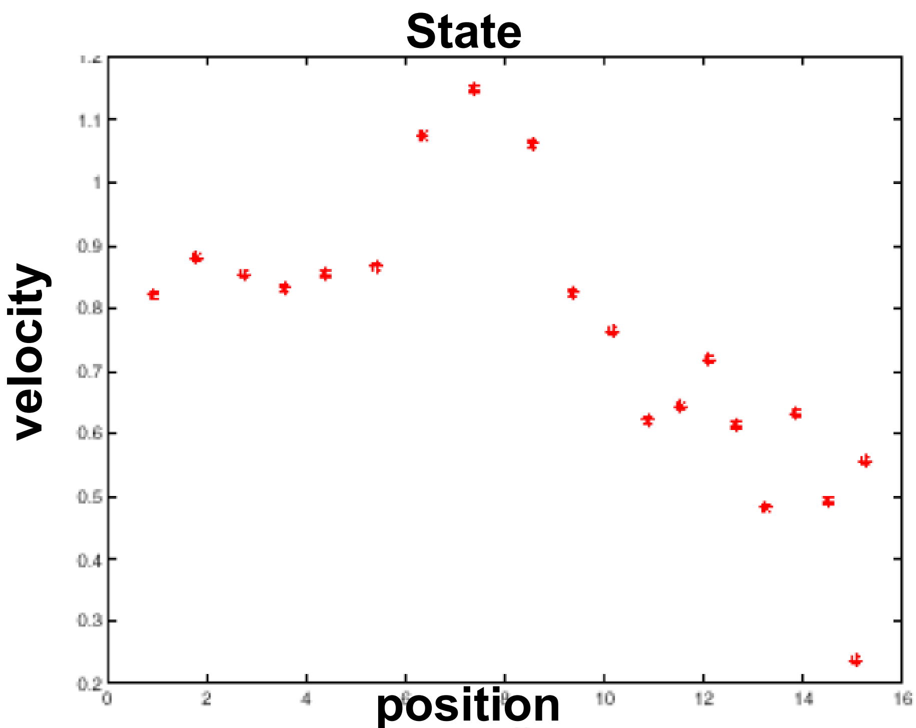


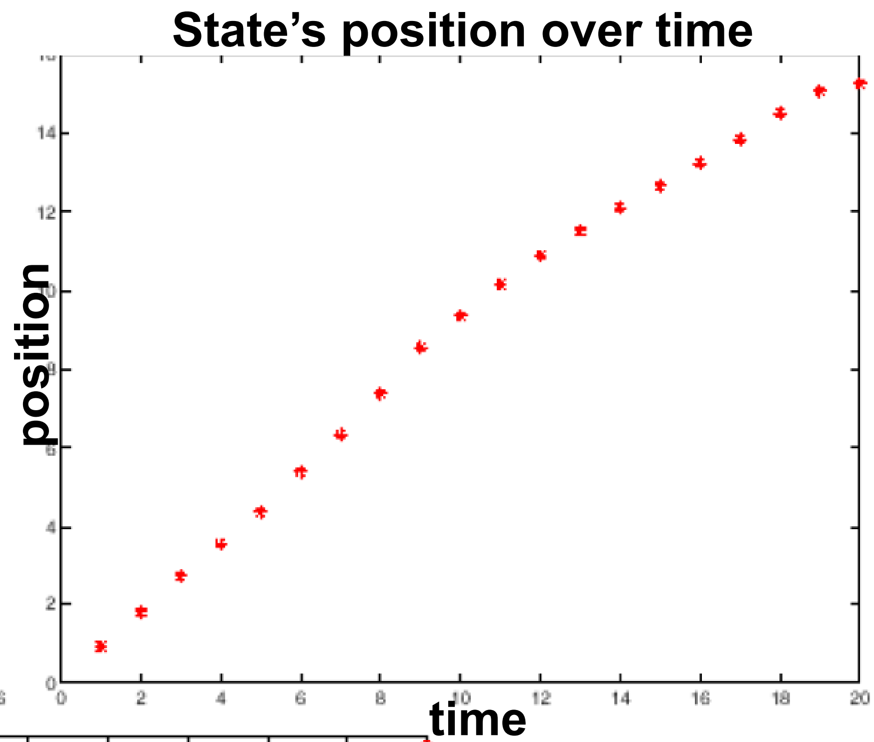
Figures
from F&P



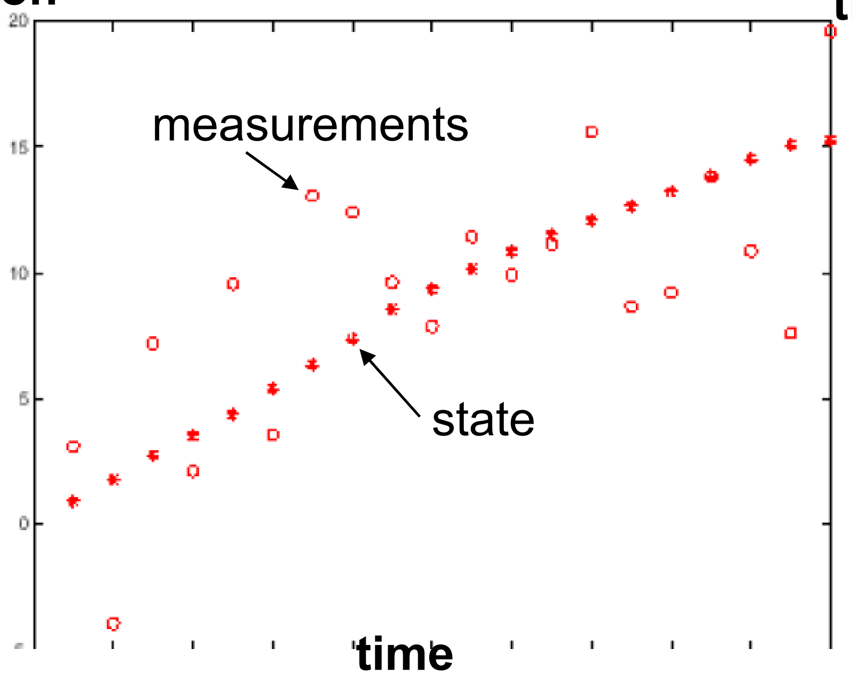


Figures
from F&P





Constant Velocity Model



Figures from F&P

Example: constant acceleration

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

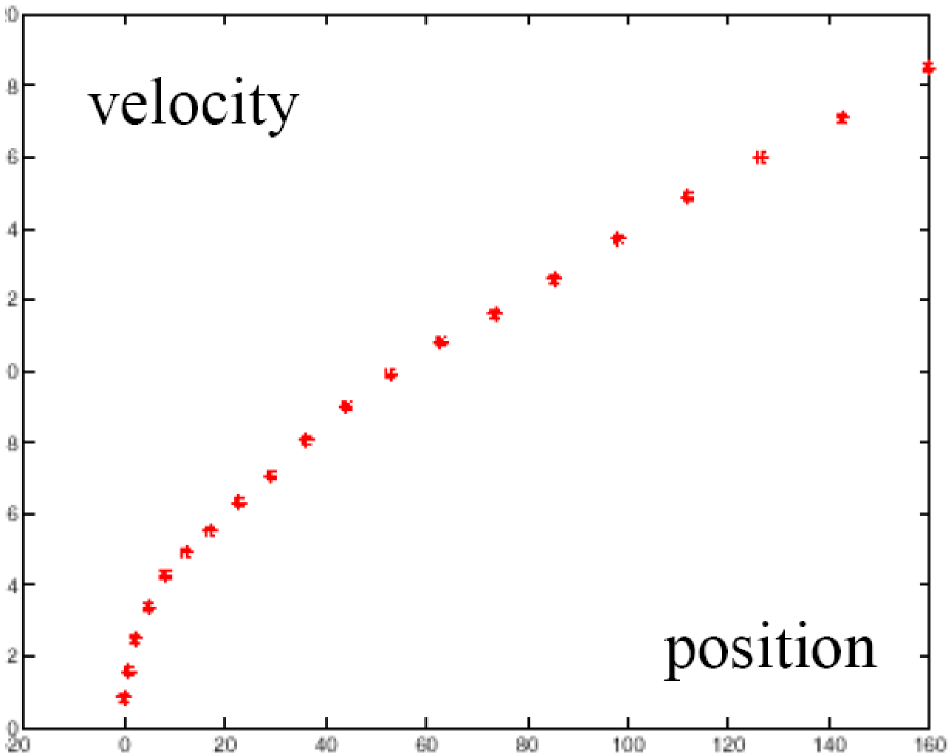
$$\mathbf{y}_t \sim N(\mathbf{M}\mathbf{x}_t; \mathbf{\Sigma}_m)$$

- State is 1d position, velocity, and acceleration
- Measurement is position only.

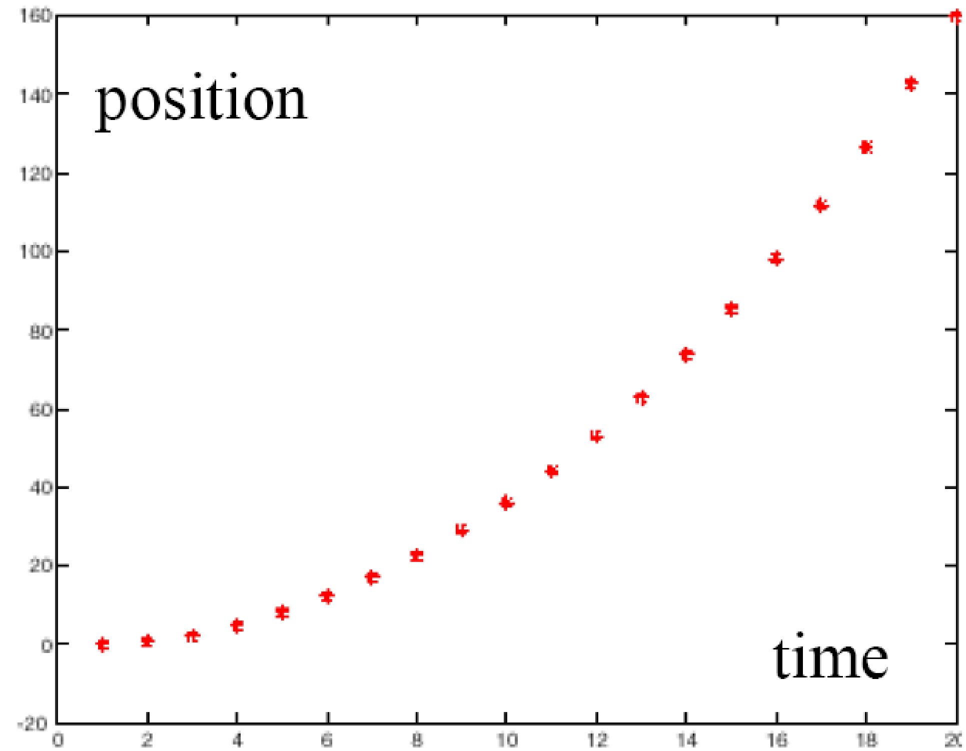
$$\left.\begin{aligned} p_t &= p_{t-1} + (\Delta t)v_{t-1} + \xi \\ v_t &= v_{t-1} + (\Delta t)a_{t-1} + \zeta \\ a_t &= a_{t-1} + \varepsilon \end{aligned}\right\} \quad \mathbf{x}_t = \begin{bmatrix} p \\ v \\ a \end{bmatrix}_t = \underbrace{\begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} p \\ v \\ a \end{bmatrix}_{t-1}}_{\mathbf{x}_{t-1}} + noise$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

State's position and velocity



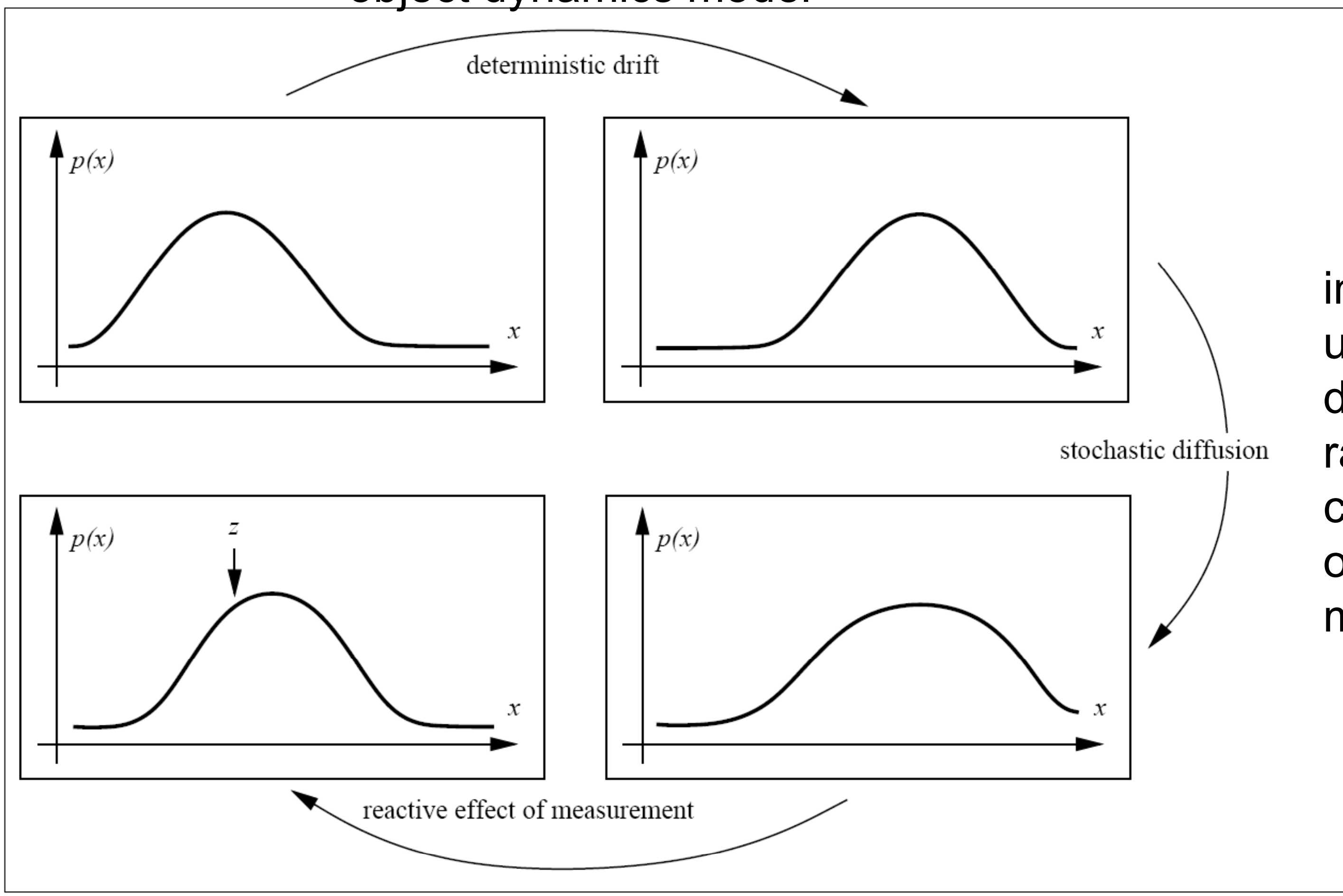
State's position over time



Constant
Acceleration
Model

Kalman filter as density propagation

Distribution shifts due to
object dynamics model

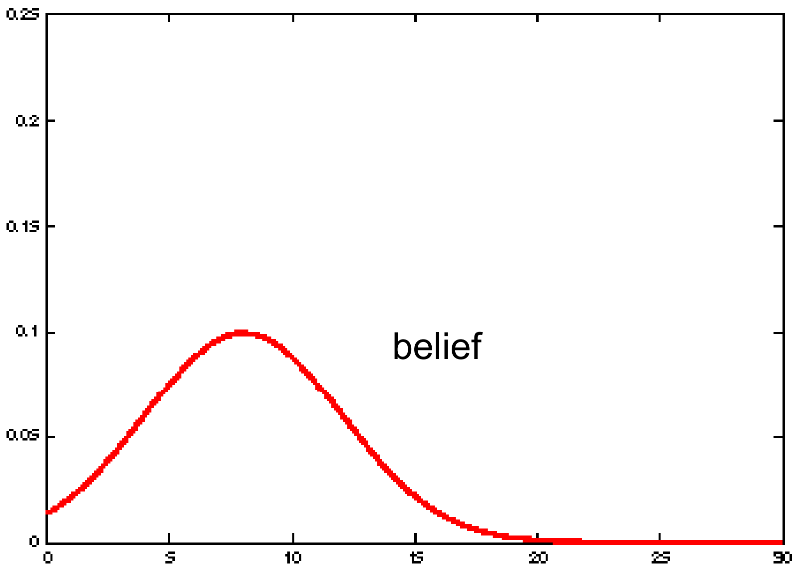


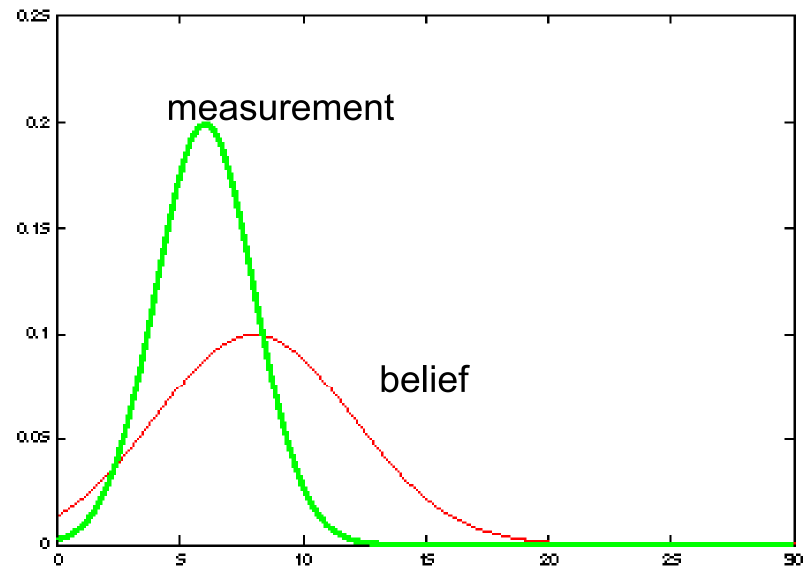
increased uncertainty due to random component of dynamics model

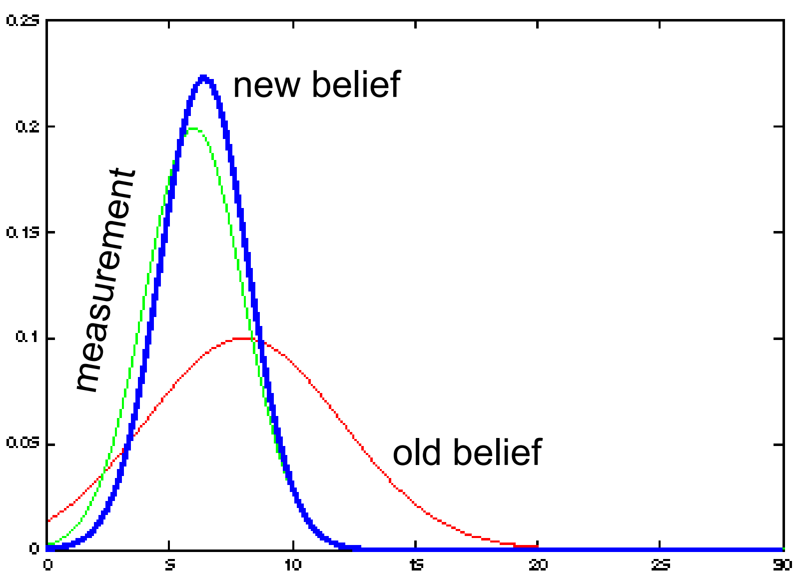
peak towards observation

Figure from Isard & Blake 1998

Kalman filter as density propagation





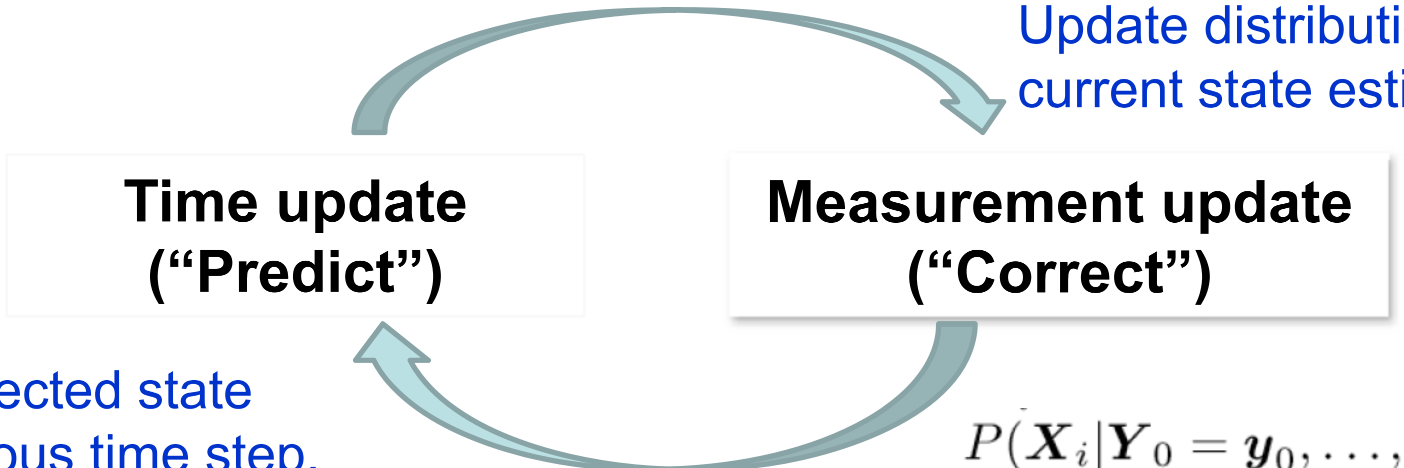


Slide by S. Thrun and J. Kosecka, Stanford

Kalman filtering

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$$

Know prediction of state, receive current measurement → Update distribution over current state estimate



$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$$

Know corrected state from previous time step, and all measurements up to the current one → Update distribution over predicted state

TIME ADVANCES
i++

Kalman filtering

- Linear models + Gaussian distributions work well (read, simplify computation)
- Gaussians also represented compactly

prediction $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1}).$

correction $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$

Kalman filter for 1d state

Dynamic model

$$x_i \sim N(dx_{i-1}, \sigma_d^2)$$

$$y_i \sim N(mx_i, \sigma_m^2)$$

Want to represent and update

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1}).$$

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$$

Notation shorthand

mean of $P(X_i|y_0, \ldots, y_{i-1})$ as $\overline{X}_i^-$ ← Predicted mean

mean of $P(X_i|y_0, \ldots, y_i)$ as $\overline{X}_i^+$ ← Corrected mean

the standard deviation of $P(X_i|y_0, \ldots, y_{i-1})$ as σ_i^-

of $P(X_i|y_0, \ldots, y_i)$ as σ_i^+

Kalman filtering

$$P(\boldsymbol{X}_i|\boldsymbol{Y}_0=\boldsymbol{y}_0,\ldots,\boldsymbol{Y}_{i-1}=\boldsymbol{y}_{i-1})$$

$$\overline{X}_i^-, \sigma_i^-$$

Know prediction of state, receive current measurement → Update distribution over current state estimate

$$P(\boldsymbol{X}_i|\boldsymbol{Y}_0=\boldsymbol{y}_0,\ldots,\boldsymbol{Y}_i=\boldsymbol{y}_i)$$

Know corrected state from previous time step, and all measurements up to the current one → Update distribution over predicted state

TIME ADVANCES

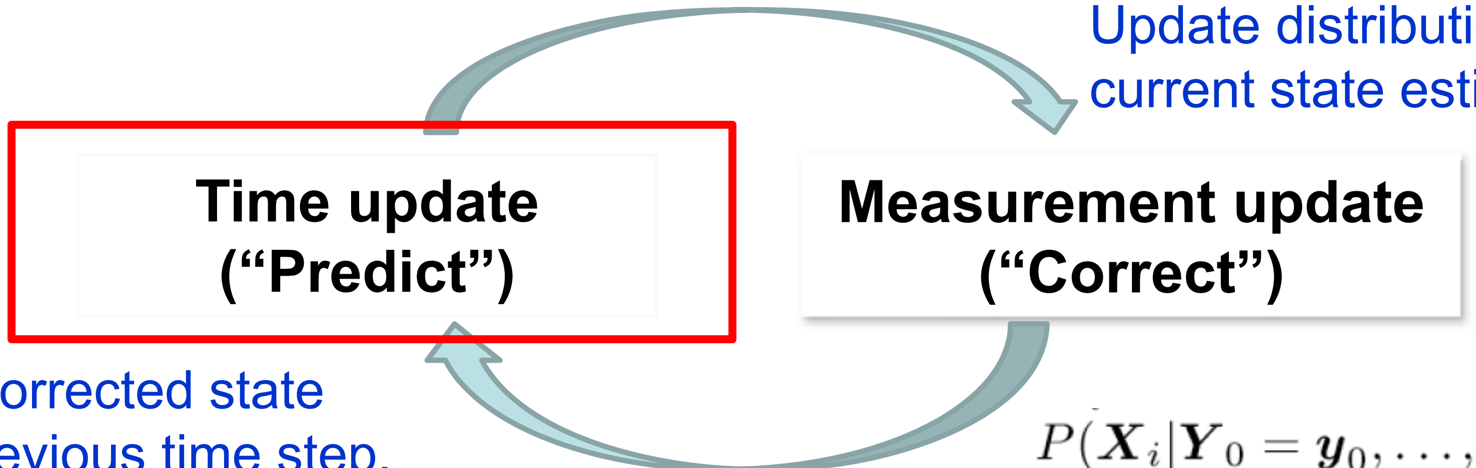
$$\overline{X}_{i-1}^+, \sigma_{i-1}^+$$

Kalman filtering

$$\overline{X}_i^-, \sigma_i^-$$

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$$

Know prediction of state, receive current measurement → Update distribution over current state estimate



$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$$

Know corrected state from previous time step, and all measurements up to the current one → Update distribution over predicted state

TIME ADVANCES

$$\overline{X}_{i-1}^+, \sigma_{i-1}^+$$

Kalman filter for 1d state: prediction

- Linear dynamic model defines expected state evolution, with noise:

$$x_i \sim N(dx_{i-1}, \sigma_d^2)$$

- Want to estimate distribution for next predicted state:

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1}) = N(\overline{X}_i^-, (\sigma_i^-)^2)$$

  – Update the mean:

$$\overline{X}_i^- = d\overline{X}_{i-1}^+$$

Predicted mean depends on state transition value (constant *d*), and mean of previous state.

  – Update the variance:

$$(\sigma_i^-)^2 = \sigma_d^2 + (d\sigma_{i-1}^+)^2$$

Variance depends on uncertainty at previous state, and noise of system's model of state evolution.

Kalman filtering

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$$

$$\overline{X}_i^-, \sigma_i^-$$

Know prediction of state, receive current measurement → Update distribution over current state estimate

Time update ("Predict")

Measurement update ("Correct")

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$$

Know corrected state from previous time step, and all measurements up to the current one → Update distribution over predicted state

TIME ADVANCES

$$\overline{X}_{i-1}^+, \sigma_{i-1}^+$$

Kalman filter for 1d state: correction

- Linear model of dynamics reflects how state is mapped to measurements:

$$y_i \sim N(m x_i, \sigma_m^2)$$

- Know predicted state distribution:

$$P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1}) = N(\overline{X}_i^-, (\sigma_i^-)^2)$$

- Want to correct distribution over current state given new measurement $\boldsymbol{y}_i$:
  - Update mean

$$\overline{X}_i^+ = \frac{\overline{X}_i^- \sigma_m^2 + m y_i (\sigma_i^-)^2}{\sigma_m^2 + m^2 (\sigma_i^-)^2}$$

  - Update variance

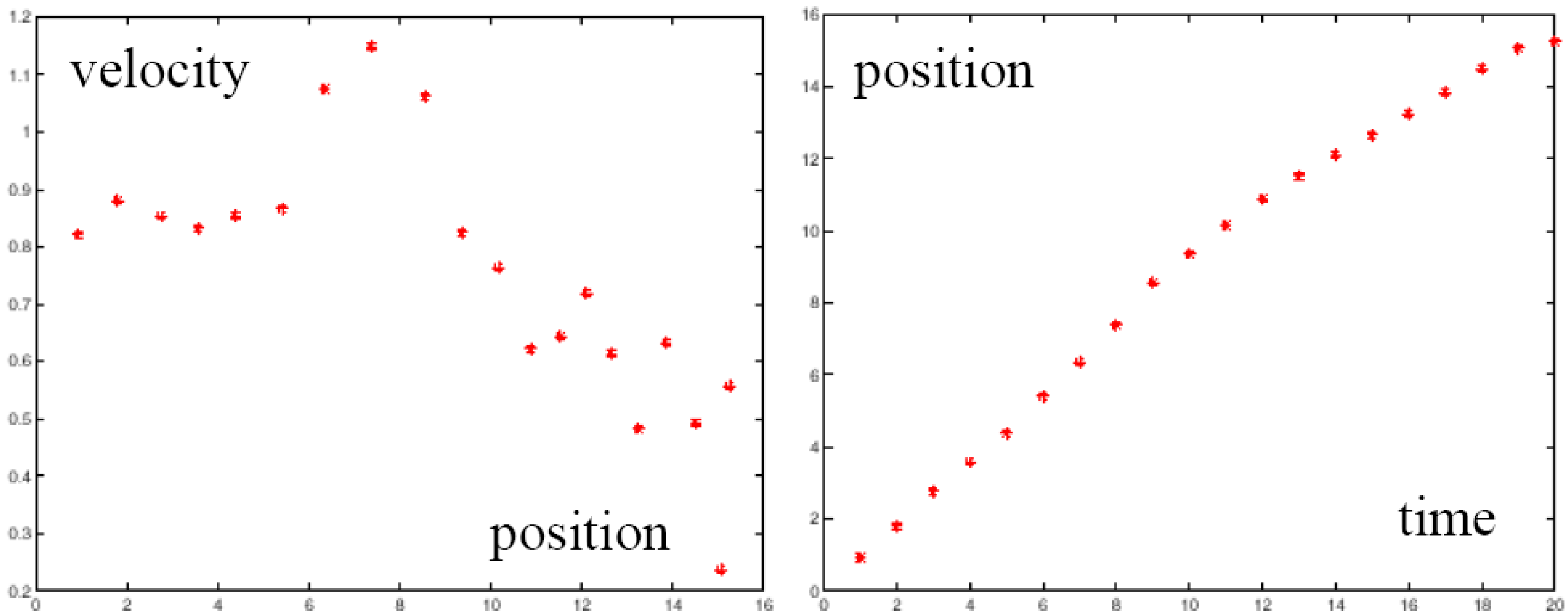
$$(\sigma_i^+)^2 = \frac{\sigma_m^2 (\sigma_i^-)^2}{\sigma_m^2 + m^2 (\sigma_i^-)^2}$$

Corrected state estimate incorporates current measurement, predicted state, meas. model, and their uncertainties.
Small measurement noise→ rely on?
Large measurement noise→ rely on?

Constant velocity model

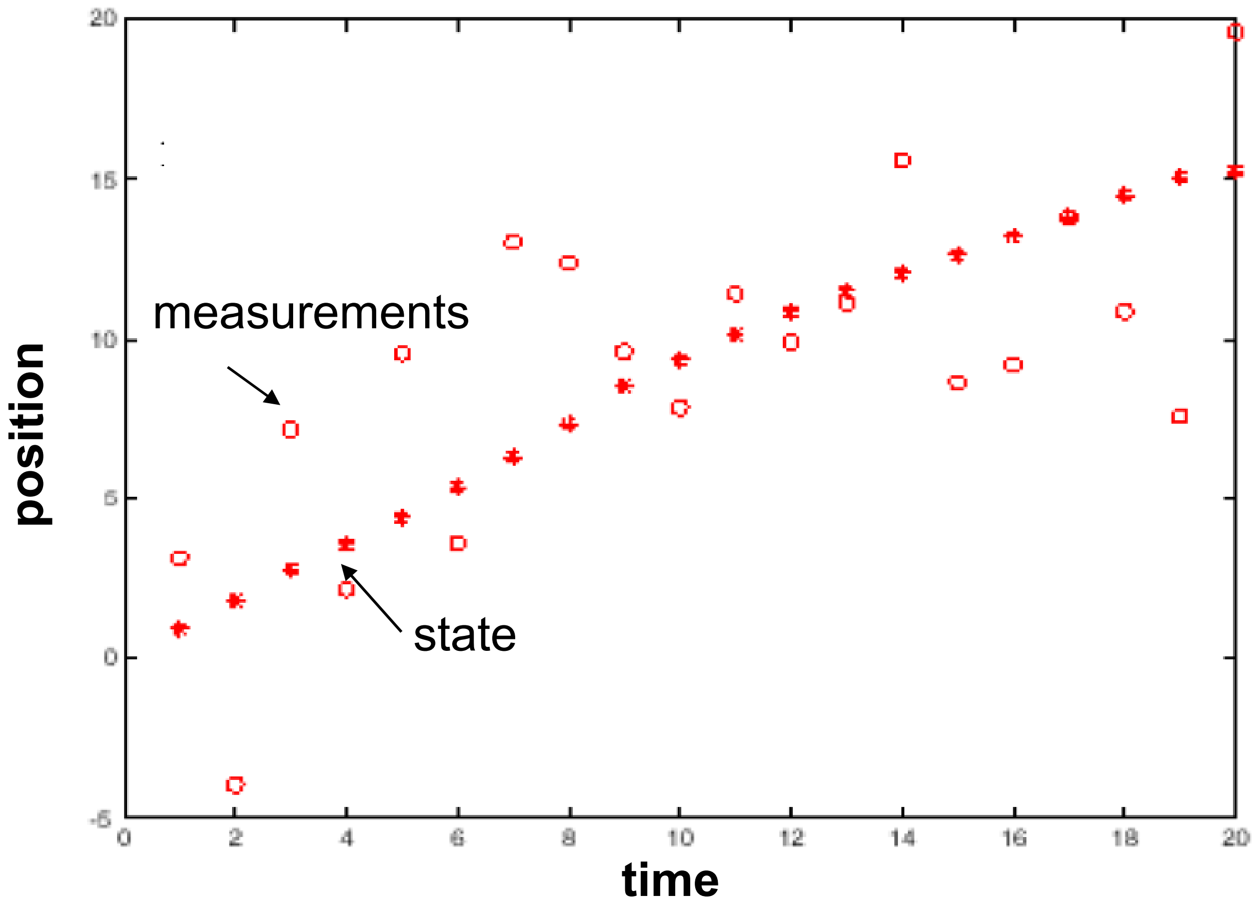
Recall this example:

State



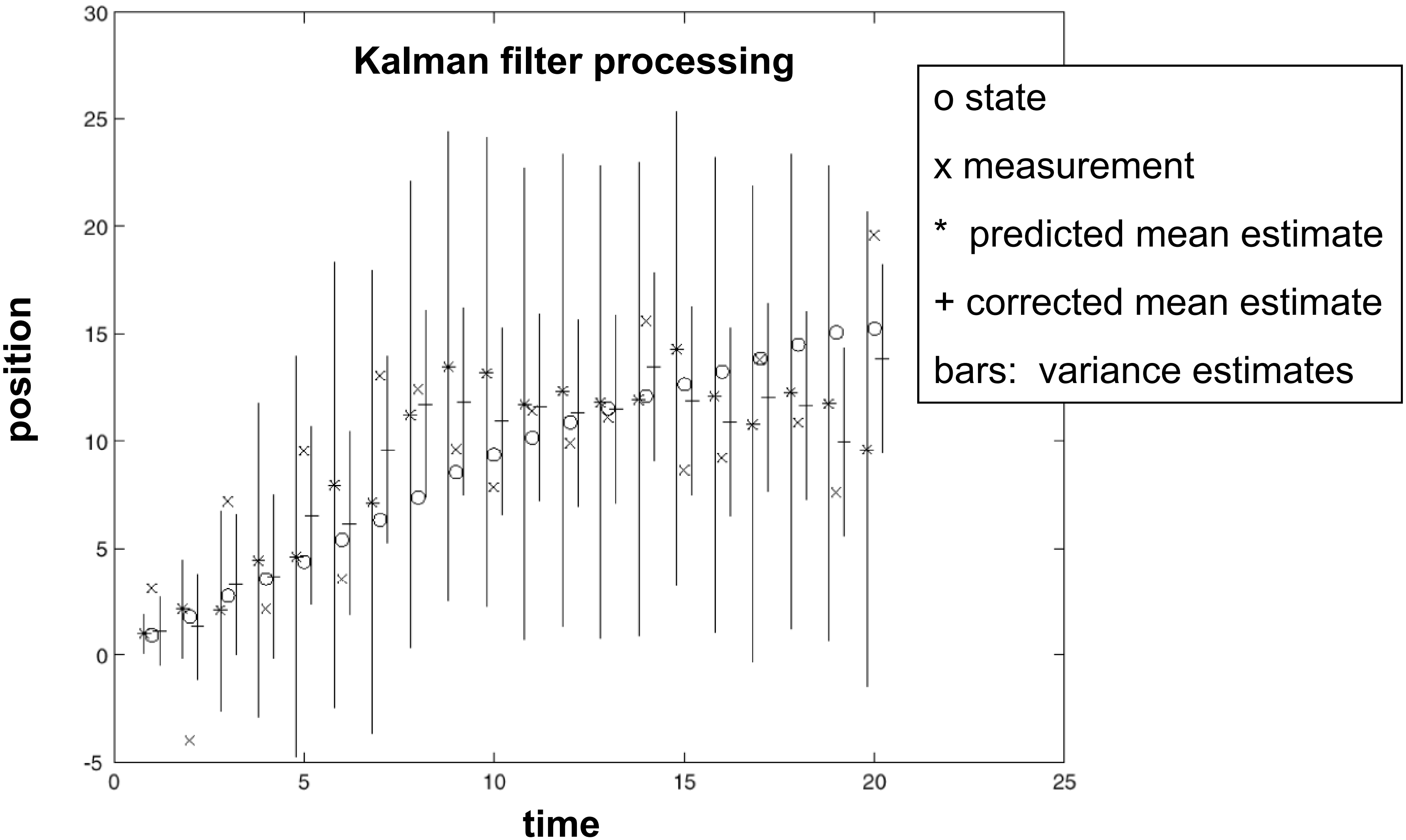
State is 2d: position + velocity
Measurement is 1d: position

Constant velocity model

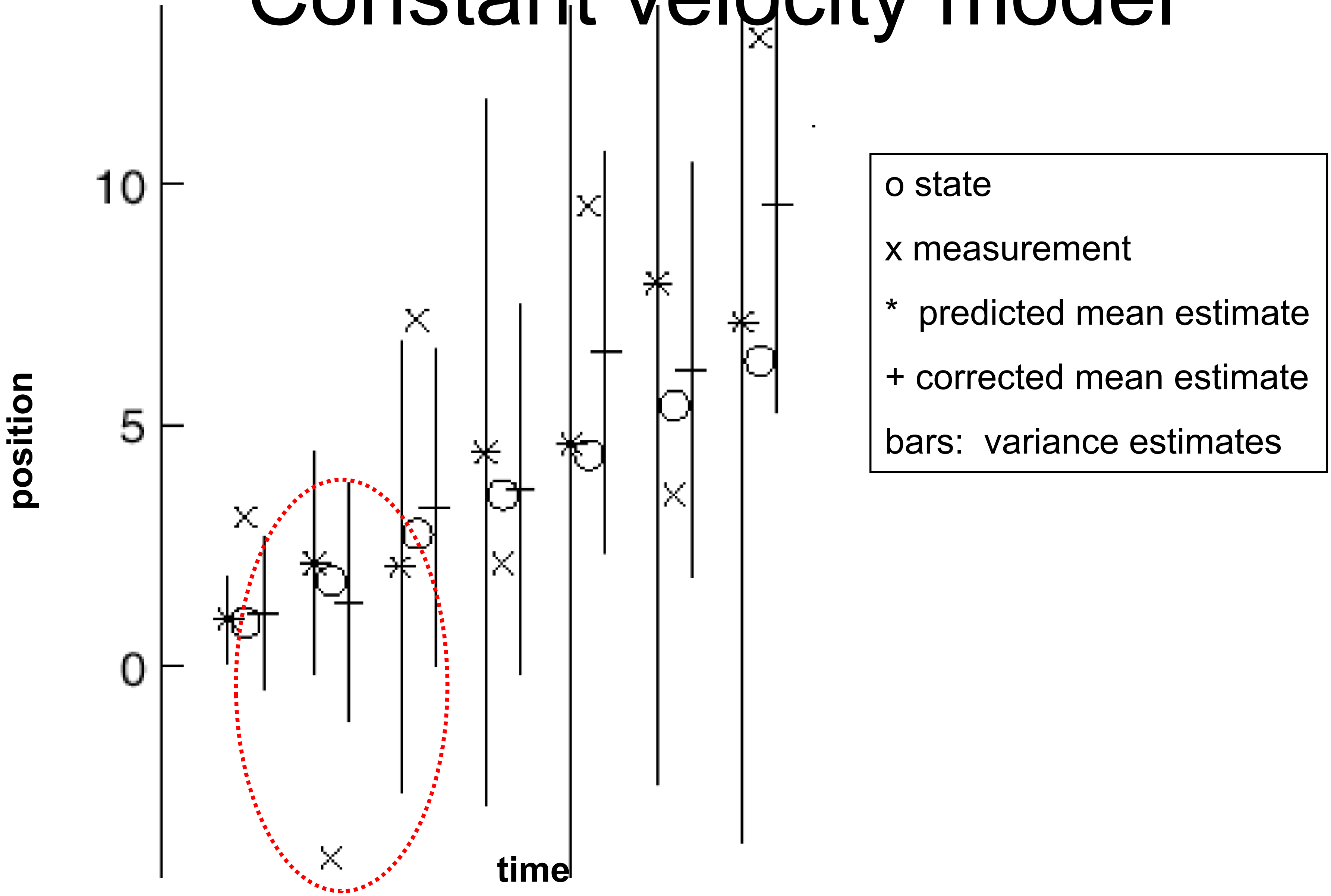


State is 2d: position + velocity
Measurement is 1d: position

Constant velocity model



Constant velocity model



N-d Kalman filtering

- This generalizes to state vectors of any dimension
- Update rules in FP Alg 17.2

Data association

- We've assumed entire measurement (**y**) was cue of interest for the state
- But, there are typically uninformative measurements too–clutter.
- **Data association**: task of determining which measurements go with which tracks.

http://www.dkimages.com/discover/previews/1002/50215713.JPG

Data association (single object in clutter)

- Global nearest neighbor
  - Choose to pay attention to the measurement with the highest probability given the predicted state
  - Can lead to tracking non-existent object
- Probabilistic approach
  - Weight the measurements by probability given predicted state

- http://www.cs.bu.edu/~betke/research/bats/

INFRARED THERMAL VIDEO ANALYSIS OF BATS

Home | Videos | Images | Publications | Investigator Intranet

Detection, Tracking, and Censusing

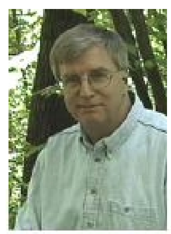
Censusing natural populations of bats is important for understanding the ecological and economic impact of these animals on terrestrial ecosystems. Colonies of Brazilian free-tailed bats (*Tadarida brasiliensis*) are of particular interest because they represent some of the largest aggregations of mammals known to mankind. It is challenging to census these bats accurately, since they emerge in large numbers at night from their day-time roosting sites. We have used infrared thermal cameras to record Brazilian free-tailed bats in California, Massachusetts, New Mexico, and Texas. We have developed an automated image analysis system that detects, tracks, and counts the emerging bats.

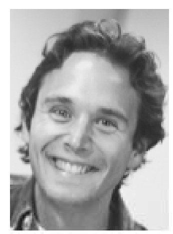
Research Team

Faculty


Margrit Betke


Cutler Cleveland


Thomas Kunz


Stan Sclaroff

- Thomas G. Hallam, University of Tennessee
- Nicholas C. Makris, Massachusetts Institute of Technology
- Gary F. McCracken, University of Tennessee
- John K. Westbrook, US Department of Agriculture

Students and Postdocs

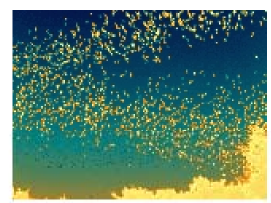
L. Allen, A. Bagchi, S. Crampton, D.E. Hirsh, J. Horn, N.I. Hristov, E. Immermann, E.Y. Lee, M. Procopio, J. Reichard, S. Tang

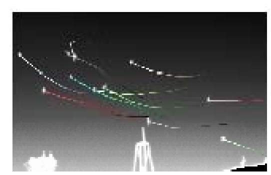
News

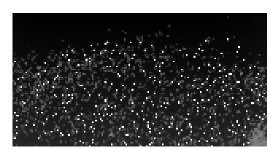
October 2007
- EcoTracker 2.1 posted under Investigator Intranet

July 2007
- Redesigned Website posted
- EcoTracker 2.0 posted under Investigator Intranet
- Video of EcoTracker in use

June 2007
CVPR Paper

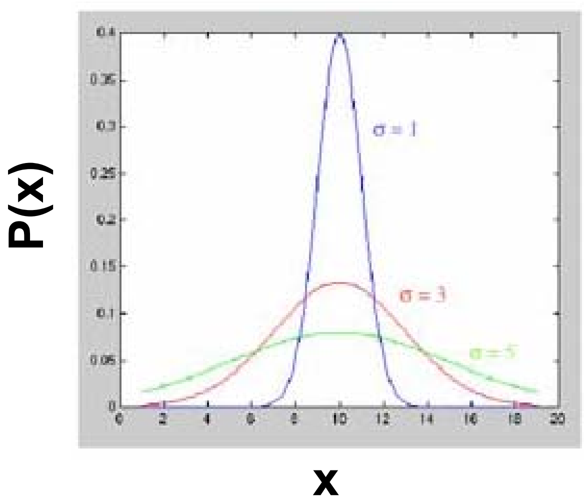






Kalman filter limitations

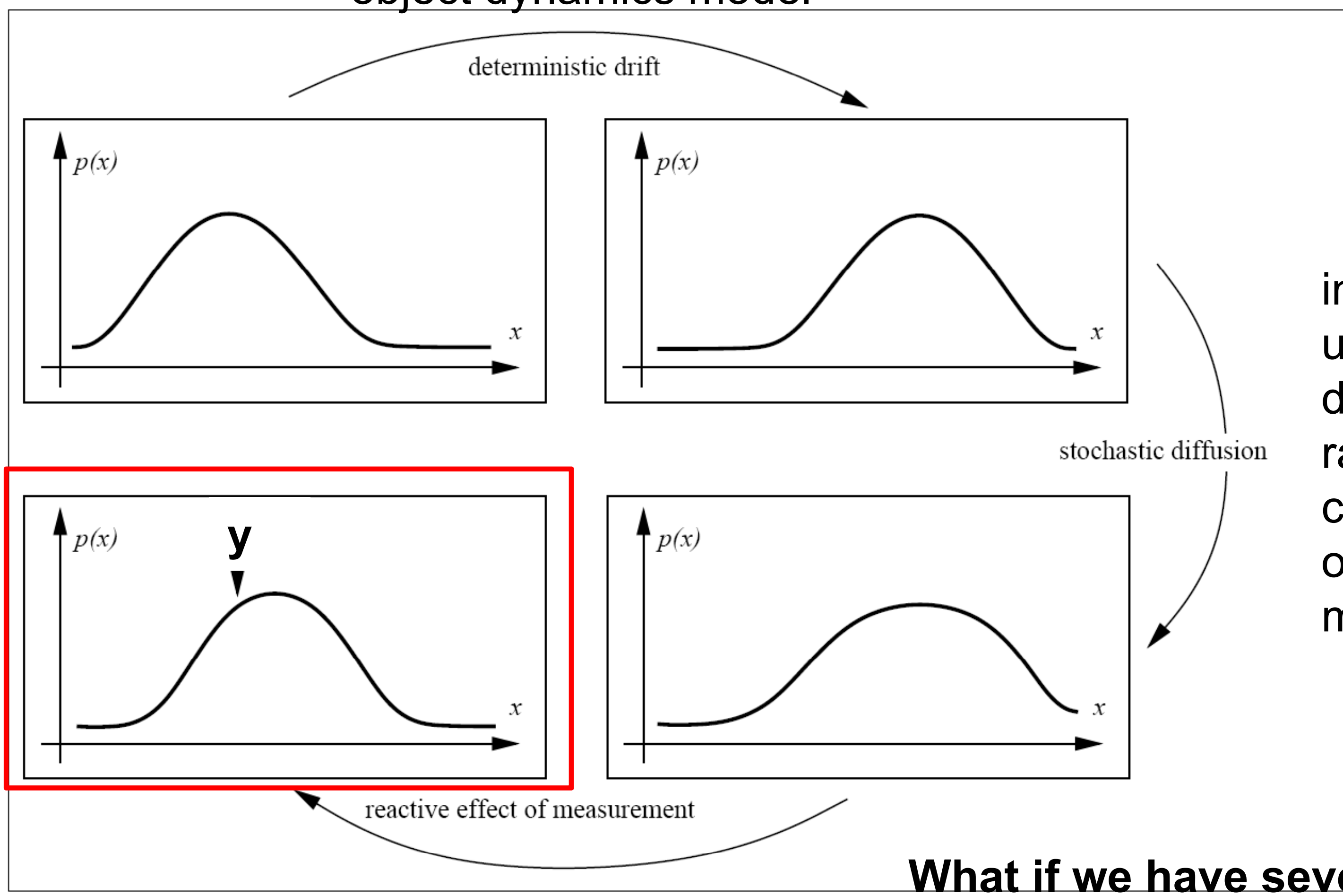
- Gaussian densities, linear dynamic model:
  - + Simple updates, compact and efficient
  - – But, unimodal distribution, only single hypothesis
  - – Restricted class of motions defined by linear model



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

Kalman filter as density propagation

Distribution shifts due to object dynamics model



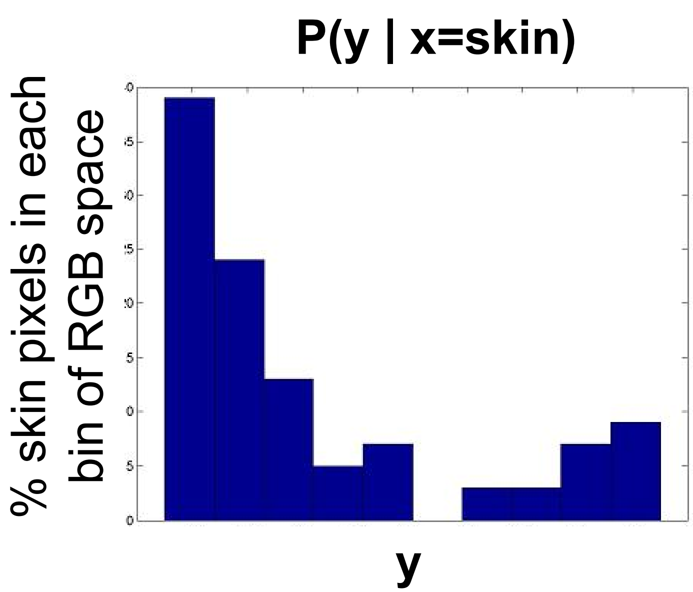
increased uncertainty due to random component of dynamics model

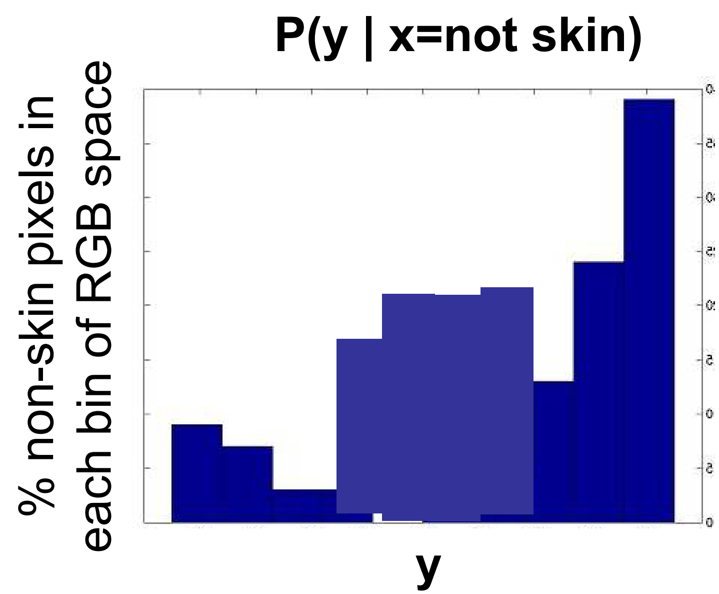
peak towards observation

What if we have several competing observations, say due to clutter?

Figure from Isard & Blake 1998

Recall conditional densities from skin detection example

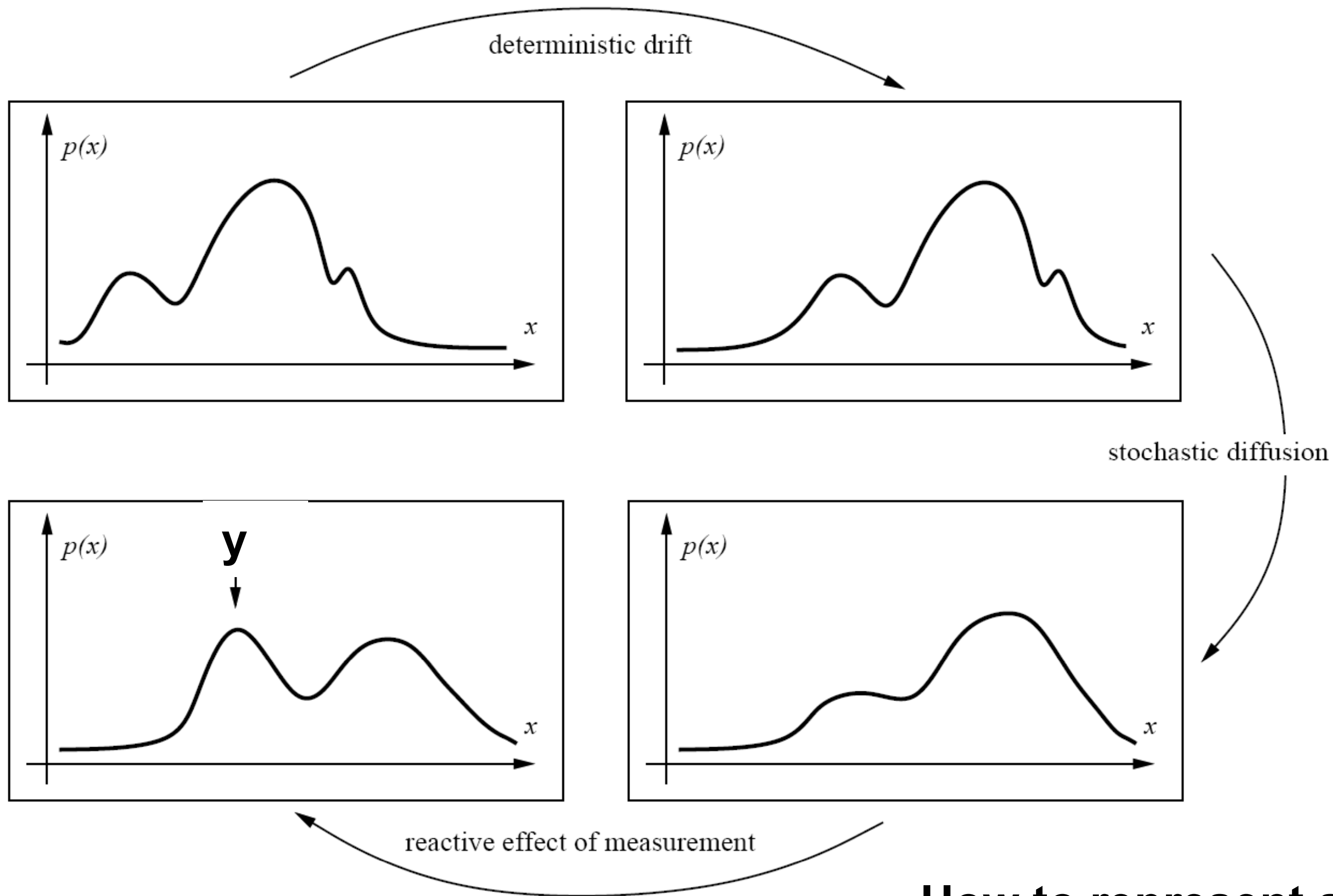




Measurement is feature y = [R G B]

Bayes' rule: P(skin | y) α P(y | skin) P(skin)

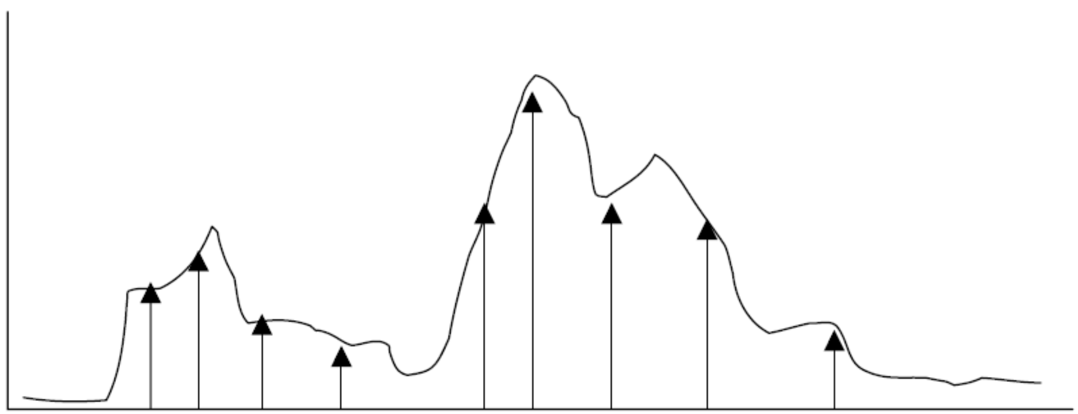
Density propagation with non-Gaussian densities



How to represent and update these distributions?

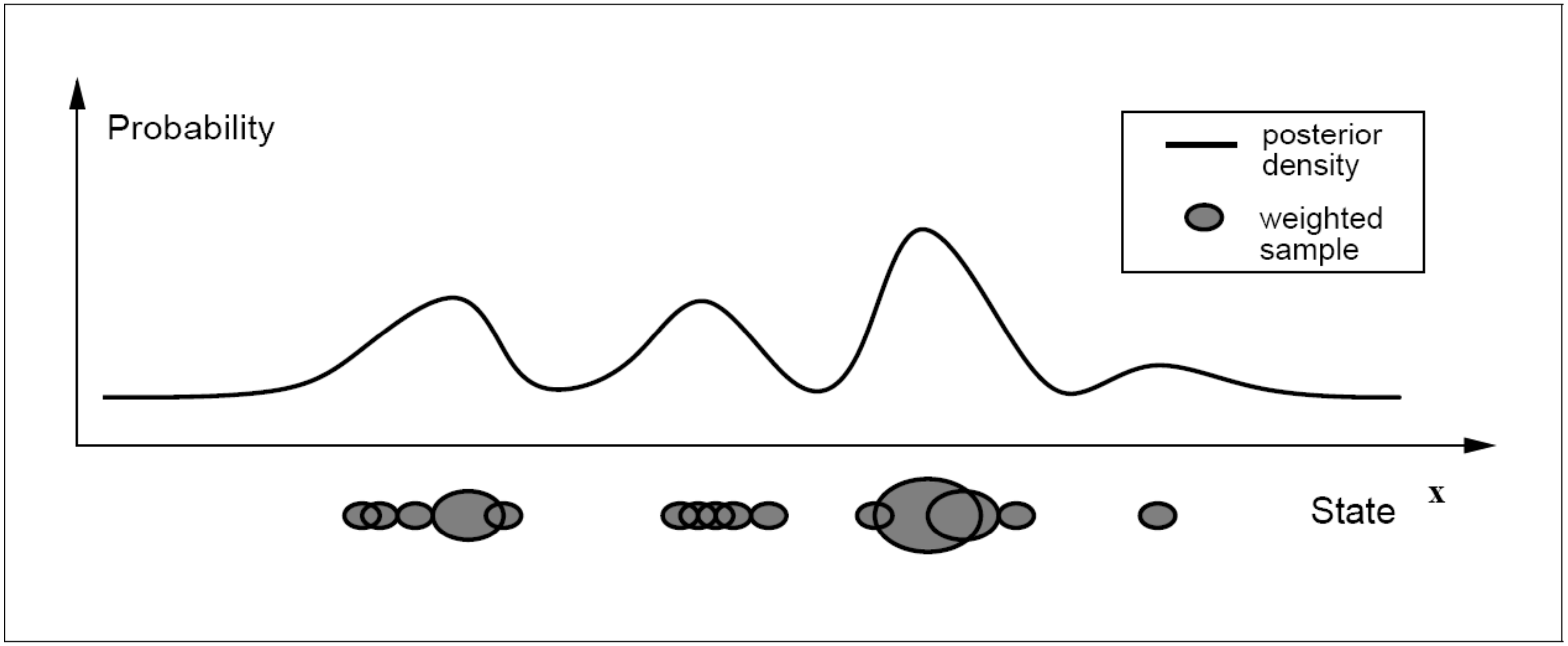
Figure from Isard & Blake 1998

Non-parametric representations for non-Gaussian densities



Can represent distribution with set of weighted samples ("particles")

Factored sampling (single frame)



Represent the posterior p(**x**|**y**) non-parametrically:

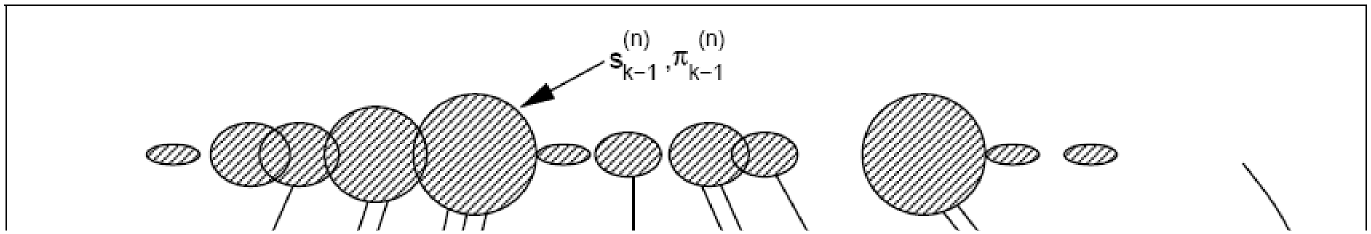
- Sample points randomly from prior density for the state, p(**x**).
- Weight the samples according to p(**y**|**x**).

Figure from Isard & Blake 1998

Particle filtering

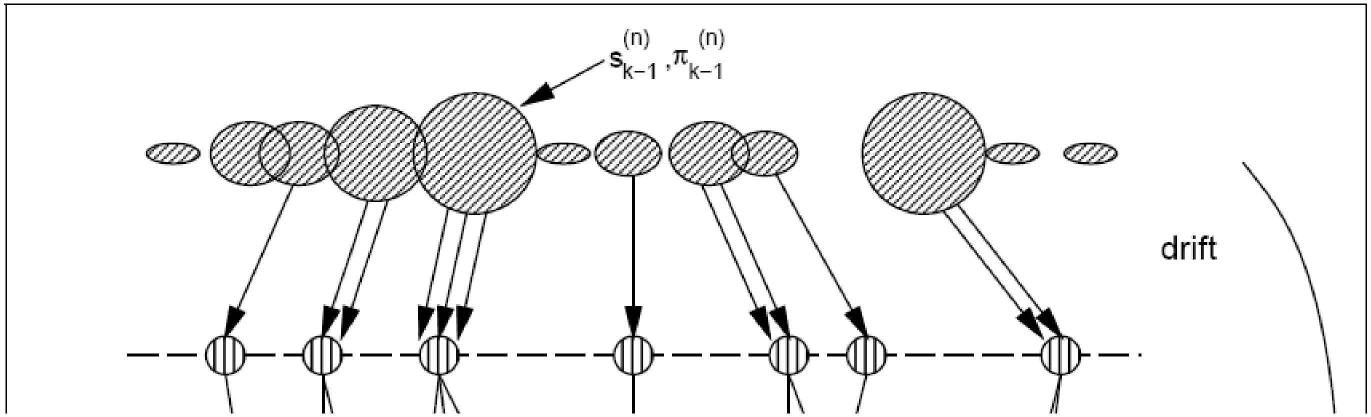
- Extend idea of sampling to propagate densities *over time* (i.e., across frames in a video sequence).
- At each time step, represent posterior $p(x_t|y_t)$ with weighted sample set
- Previous time step's sample set $p(x_t|y_{t-1})$ is passed to next time step as the effective prior
- (a.k.a. survival of the fittest, sequential Monte Carlo filtering, Condensation [Isard & Blake 96])

Particle filtering: Condensation



Start with weighted samples from previous time step

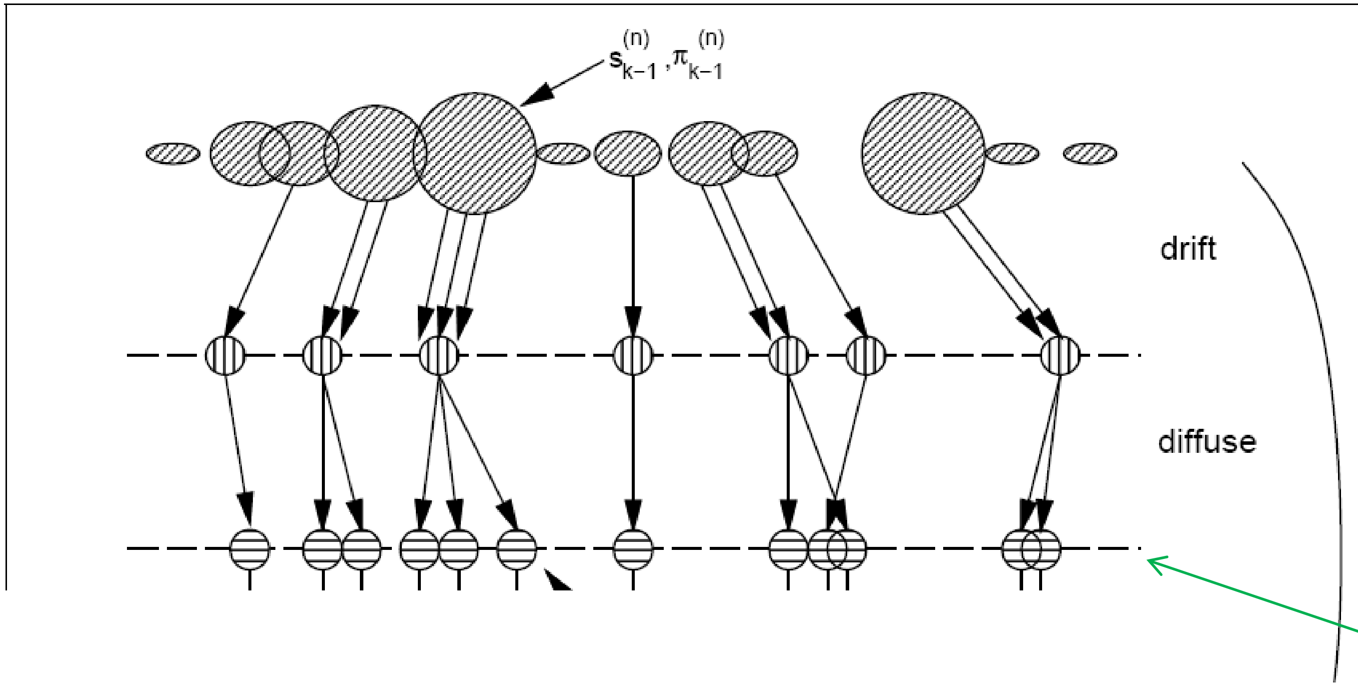
Particle filtering: Condensation



Start with weighted samples from previous time step

Sample and shift according to dynamics model

Particle filtering: Condensation

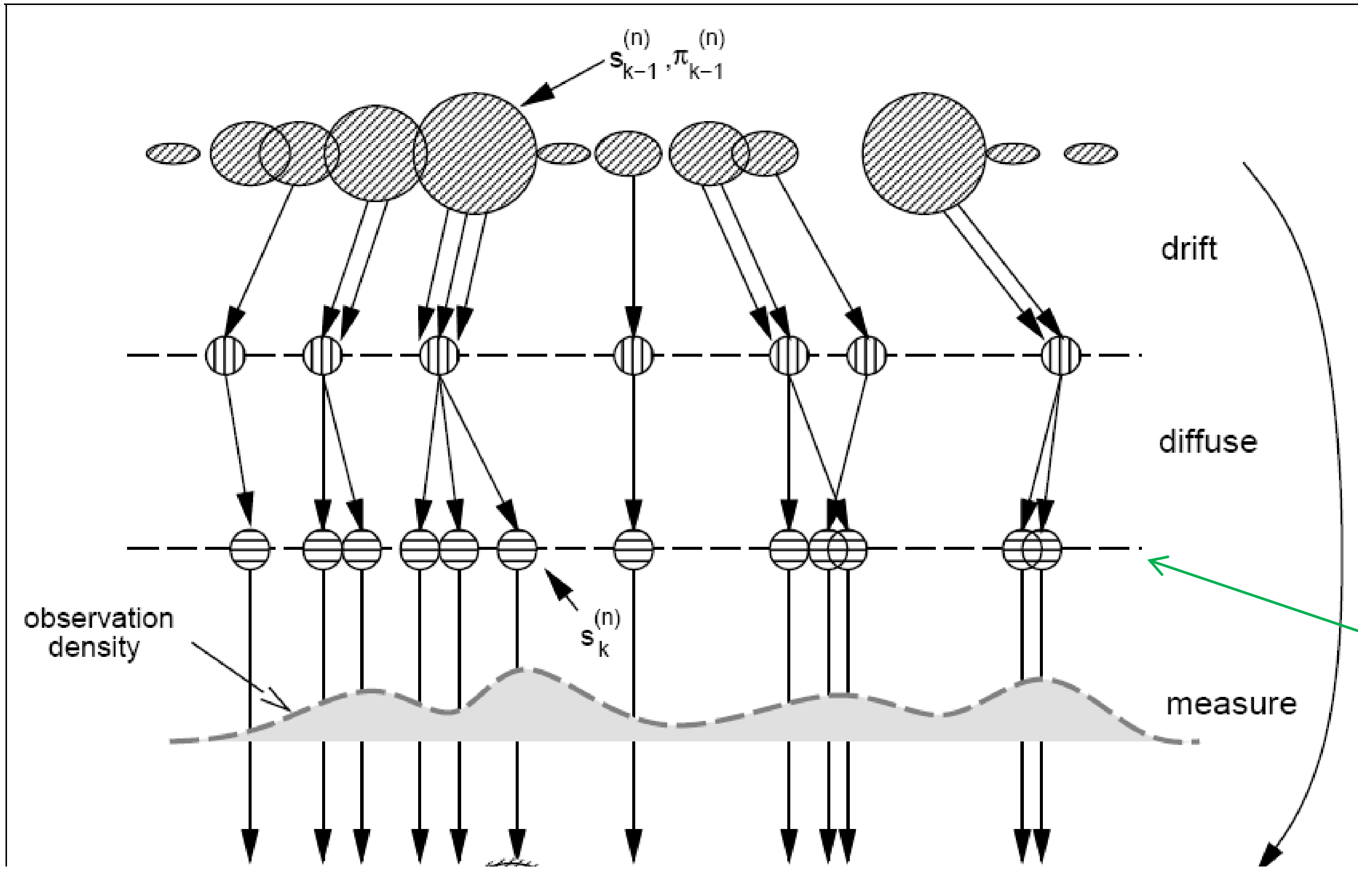


Start with weighted samples from previous time step

Sample and shift according to dynamics model

Spread due to randomness; this is effective prior density $p(x_t|y_{t-1})$

Particle filtering: Condensation



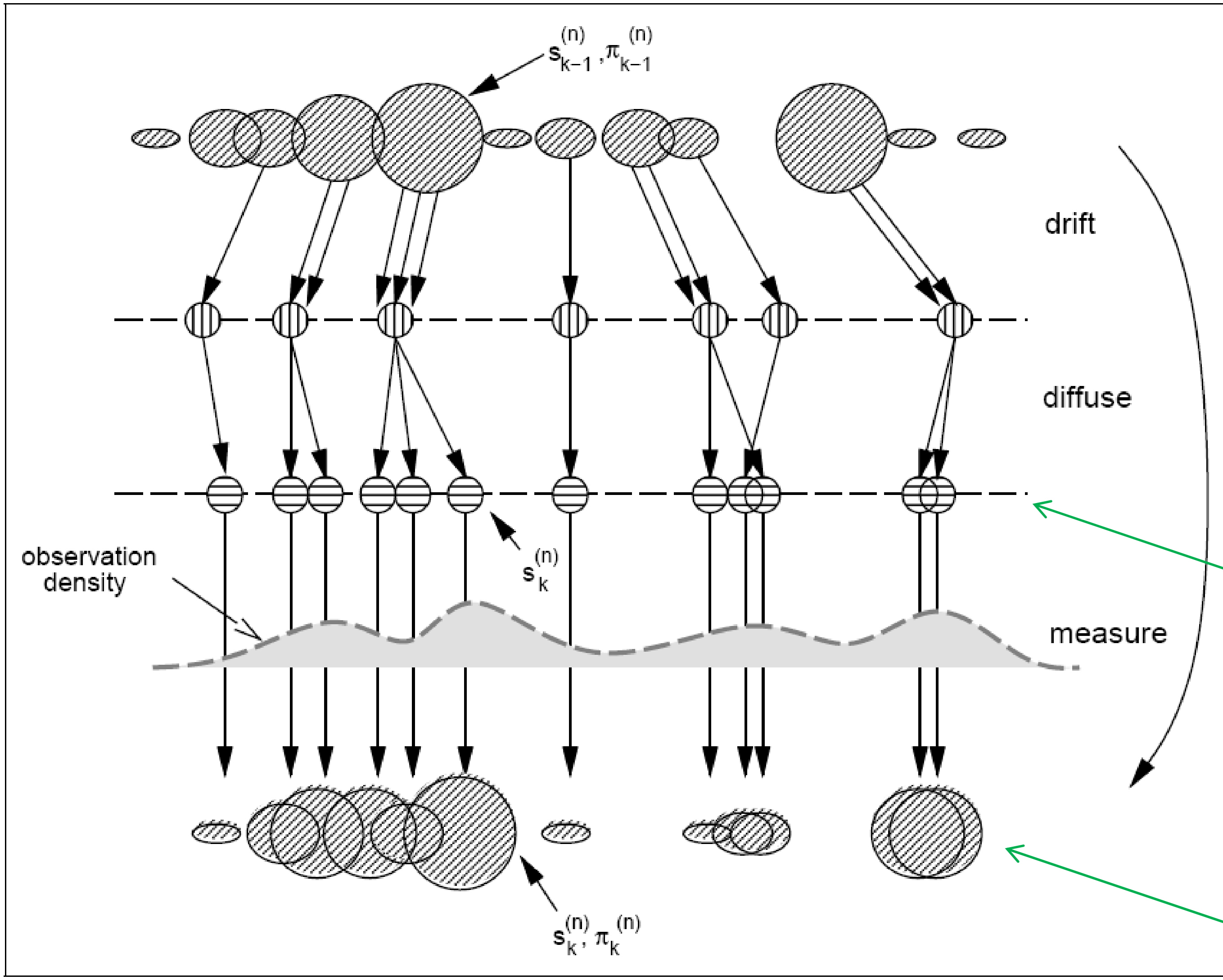
Start with weighted samples from previous time step

Sample and shift according to dynamics model

Spread due to randomness; this is effective prior density $p(x_t|y_{t-1})$

Weight the samples according to observation density

Particle filtering: Condensation



Start with weighted samples from previous time step

Sample and shift according to dynamics model

Spread due to randomness; this is effective prior density $p(x_t|y_{t-1})$

Weight the samples according to observation density

Arrive at current estimate for posterior $p(x_t|y_t)$

Figure from Isard & Blake 1998

Particle filtering: Condensation

The green spheres correspond to the members of the sample set, where the size of the sphere is an indication of the sample weight. The red line is the measurement density function.
http://www.robots.ox.ac.uk/~misard/condensation.html

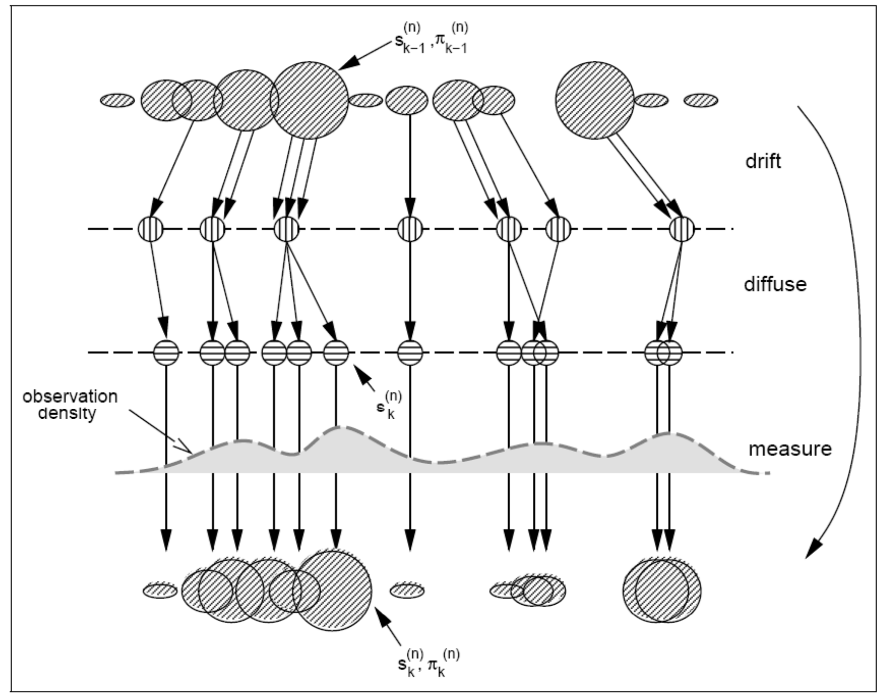
Particle filtering: what we need

Initialize according to prior on state p($\mathbf{x}_0$)

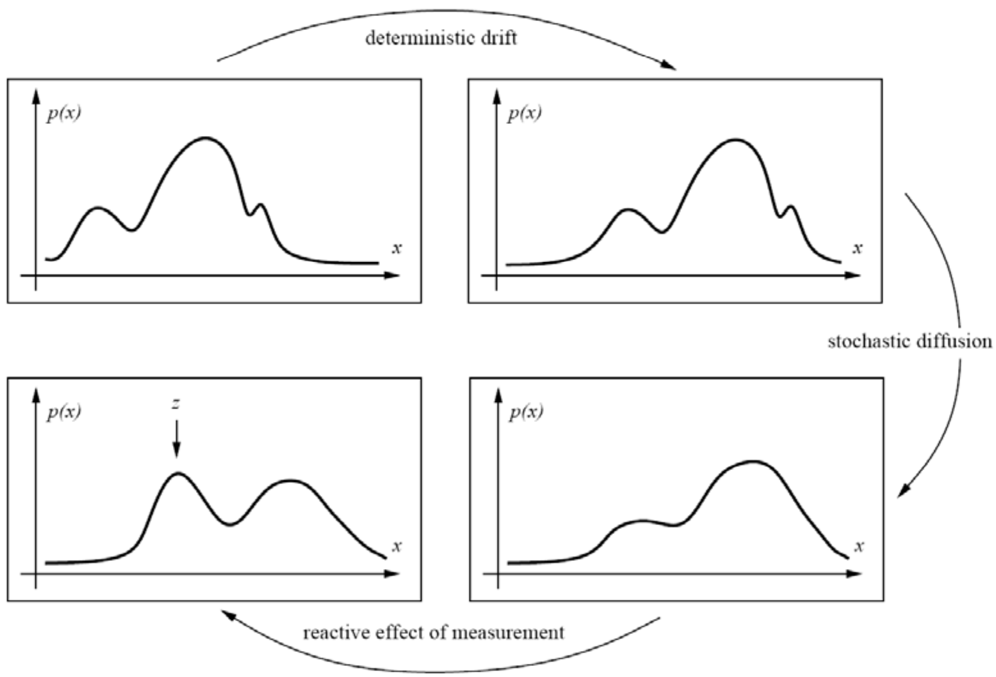
Conditional density p($\mathbf{y}|\mathbf{x}$) is defined

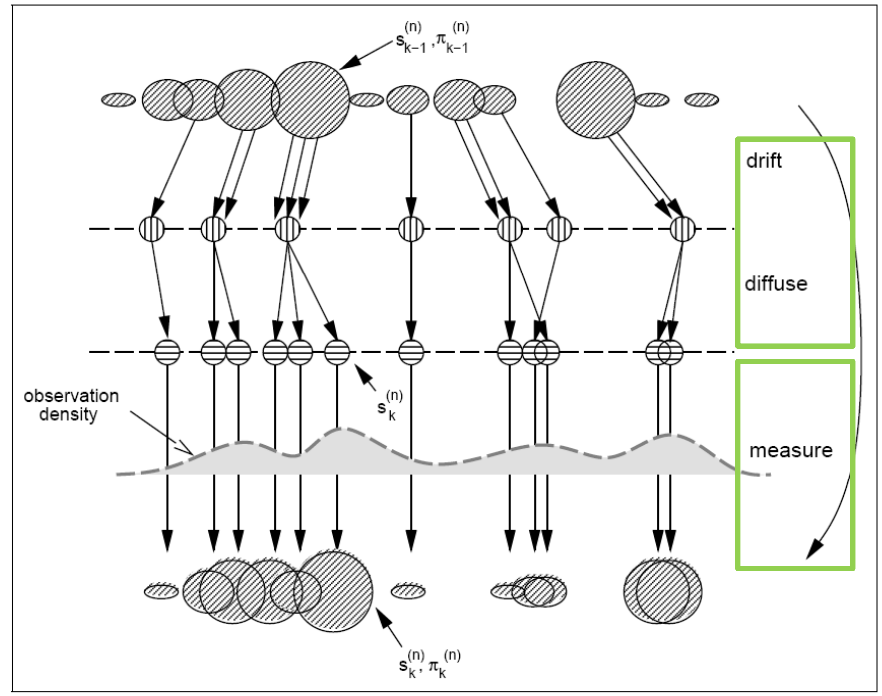
- e.g., render model according to state **x**, then compare actual image and that rendering

Object dynamics p($\mathbf{x}_t|\mathbf{x}_{t-1}$)



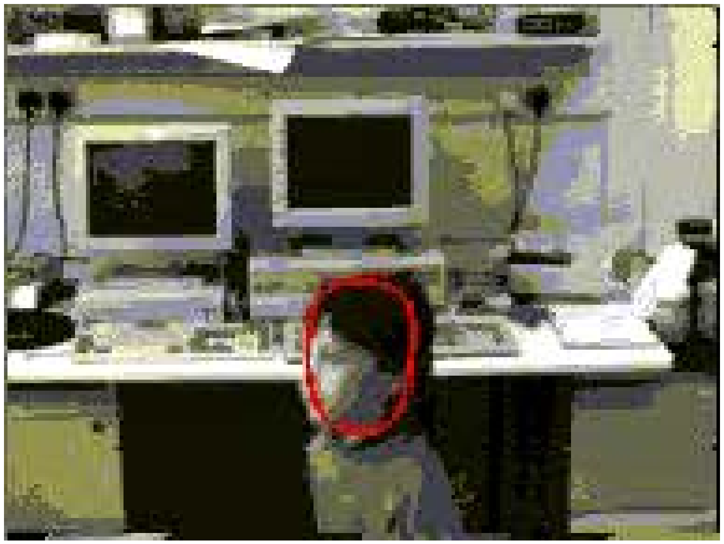
Particle filtering

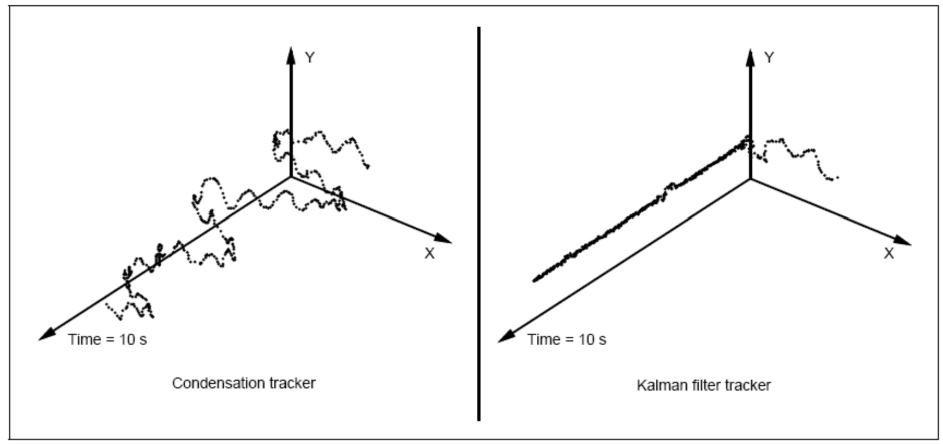




This matches our general picture of density propagation, and the prediction-correction cycle of tracking with dynamics.

Condensation-based results



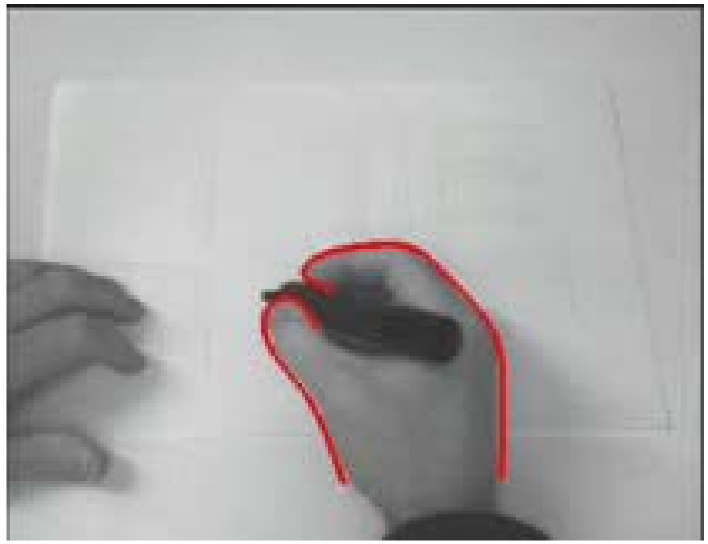


Monitor is a distractor, multiple hypotheses necessary.

Kalman filter fails once it starts tracking the monitor.

http://www.robots.ox.ac.uk/~vdg/dynamics.html
Visual Dynamics Group, Dept. Engineering Science, University of Oxford 1998

Condensation-based results



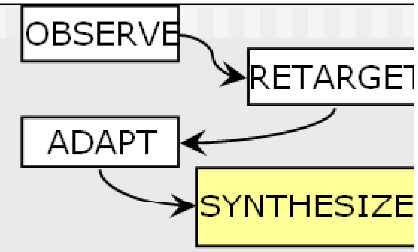
Switching between multiple motion models.

http://www.robots.ox.ac.uk/~vdg/dynamics.html
Visual Dynamics Group, Dept. Engineering Science, University of Oxford
1998

Issues

- Initialization
  – Often done manually
- Data association, multiple tracked objects
  – Occlusions
- Deformable and articulated objects
- Constructing accurate models of dynamics

Next, a brief look at an example-based technique for estimating pose and representing human motion dynamics…

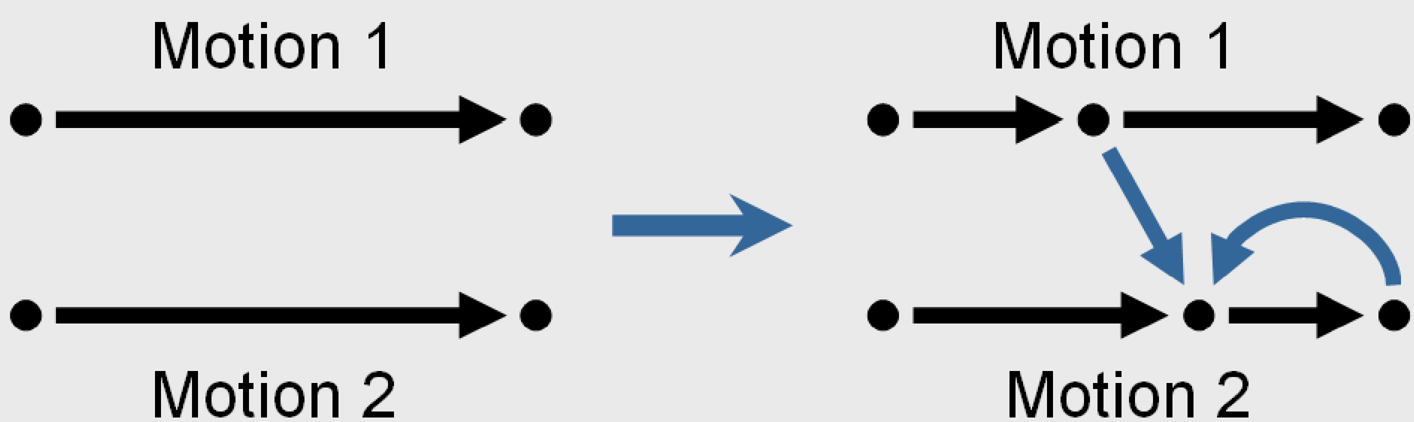


Motion Graphs

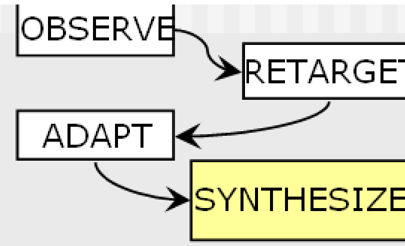
Kovar, Gleicher, Pighin ’02

Start with a database of motions, each with type and constraint information.

Goal: add transitions at opportune points.

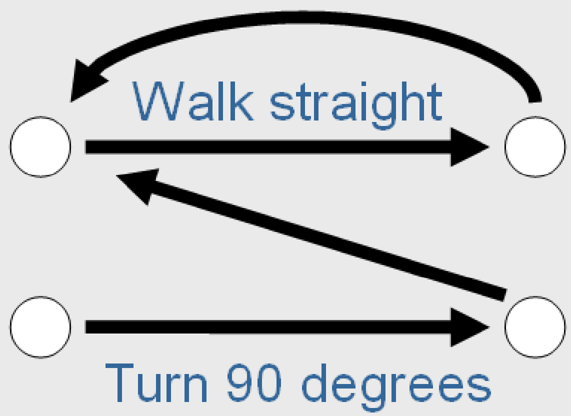


http://www.cs.wisc.edu/graphics/Talks/Gleicher/2002/AnimByExample_files/frame.htm



Motion Graphs

Idea: automatically add transitions within a motion database



Edge = clip

Node = choice point

Walk = motion

Quality: restrict transitions

Control: build walks that meet constraints

Motion capture (Mocap)

Collect pose data with active sensing – special markers, cameras.

http://www.cs.wisc.edu/graphics/Talks/Gleicher/2002/AnimByExample_files/frame.htm

Motion graphs

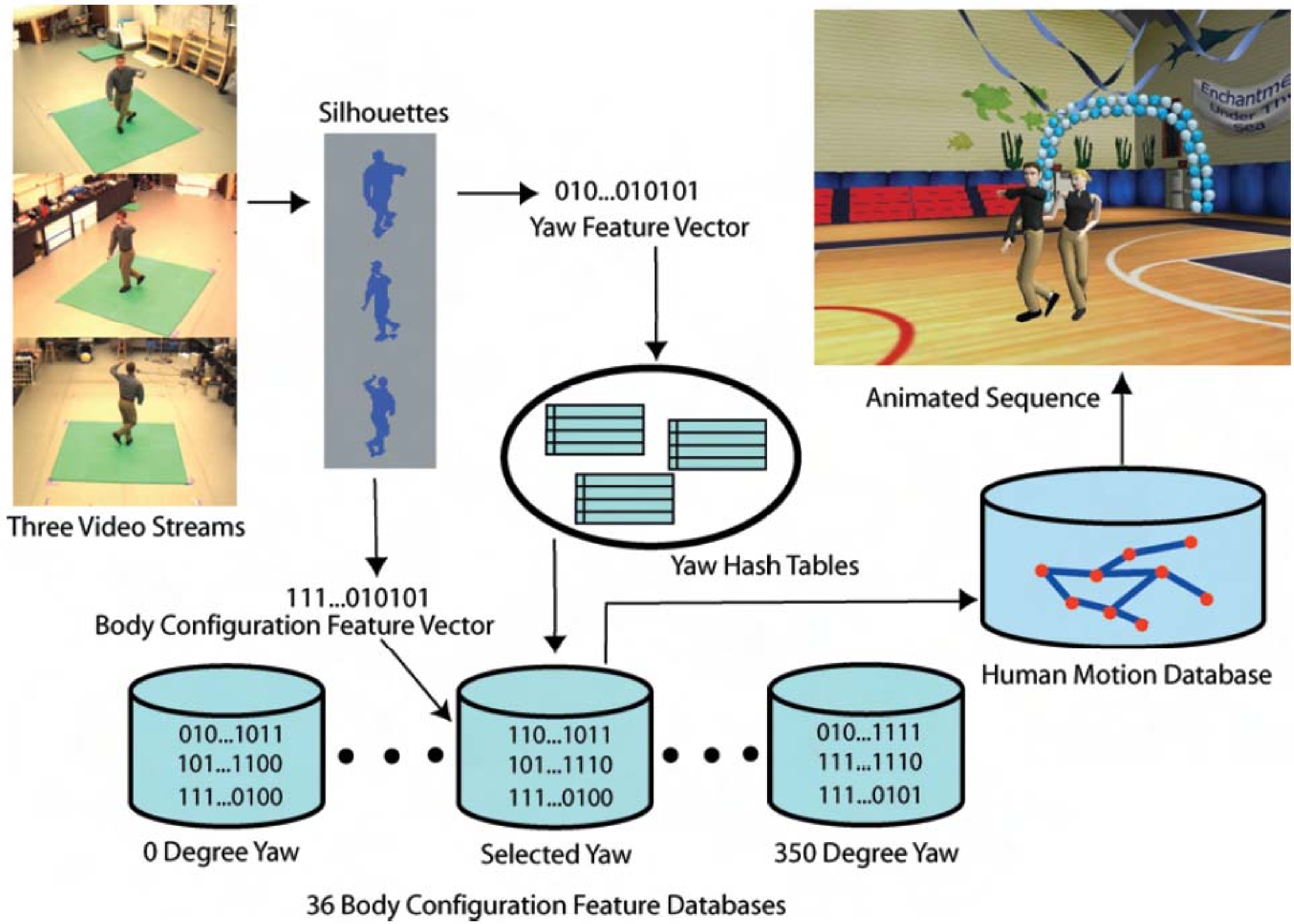
- Graphics application:
  - Any walk on the graph is a valid motion
  - Can synthesize new animation:
    - Select motion clips from the graph
    - Reassemble them to form new motion
  - Maintain realism of motions because clips retain subtle details of real motion.
- Vision application:
  - Non-parametric representation of human motion dynamics

Example-based pose estimation and animation

- Build a two-character motion graph from examples of people dancing with mocap
- Populate database with synthetically generated silhouettes in poses defined by mocap (behavior specific dynamics)
- Use discriminative silhouette features to identify similar examples in database
- Retrieve the pose stored for those similar examples to estimate user's pose
- Animate user and hypothetical partner

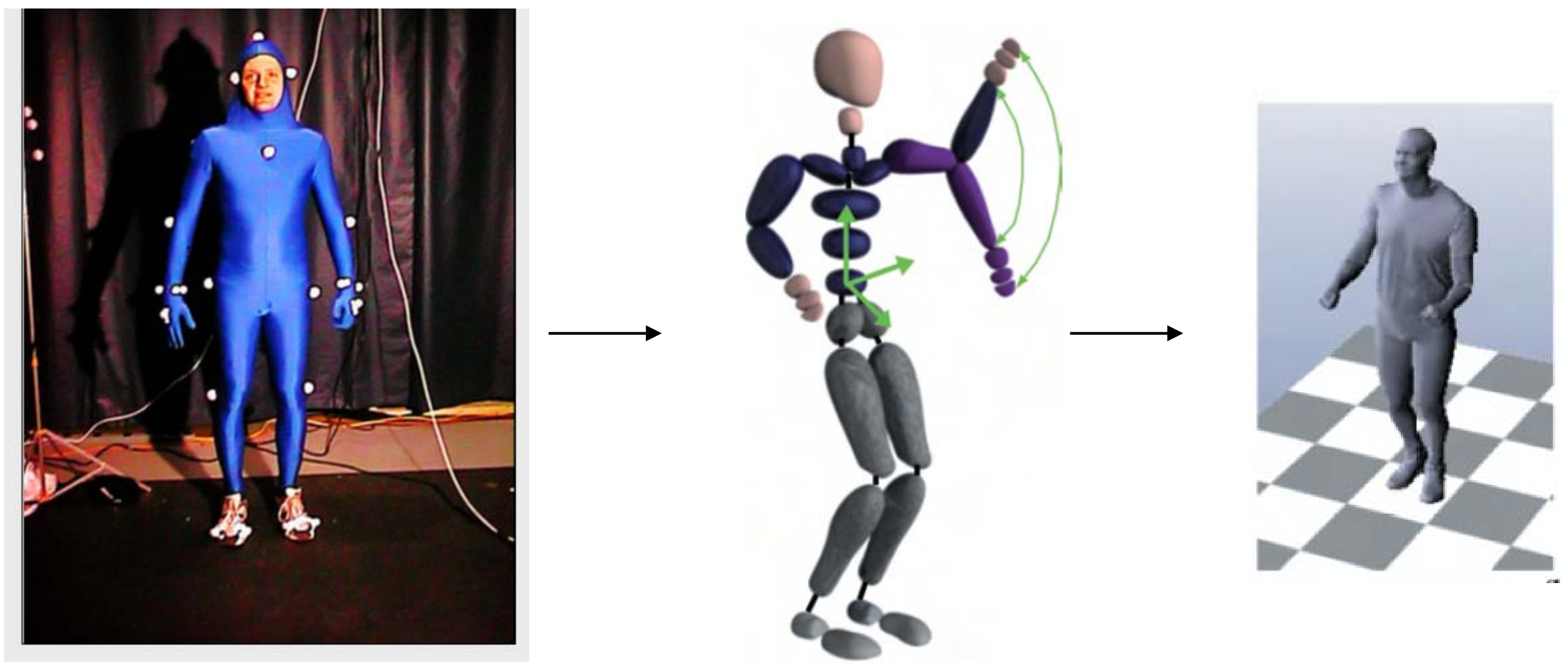
Ren, Shakhnarovich, Hodgins, Pfister, and Viola, 2005.

Overview



Ren, Shakhnarovich, Hodgins, Pfister, and Viola, 2005.

Pose parameters



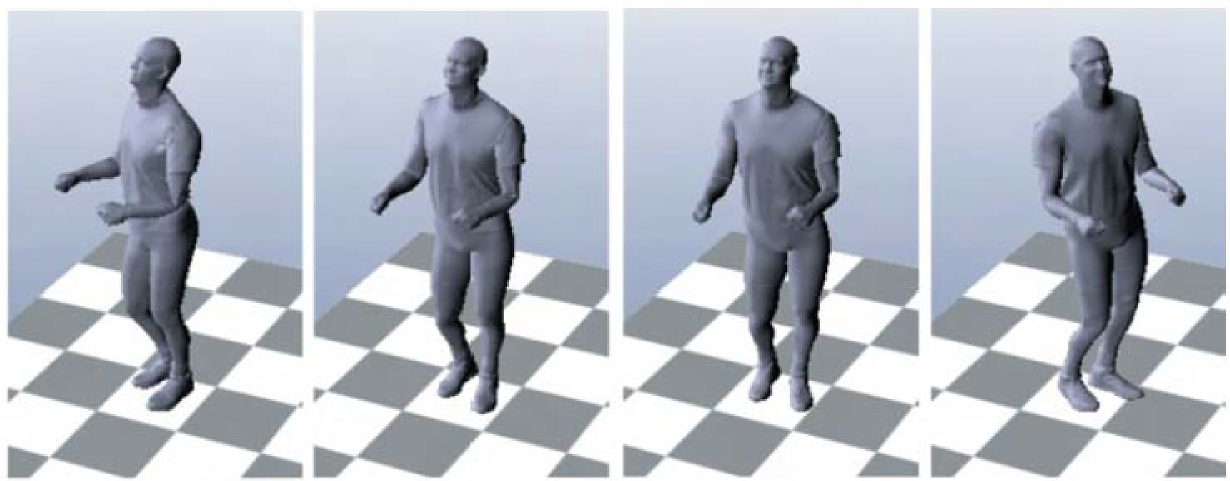
3d joint positions:

[x1 y1 z1, x2 y2, z2,…x20 y20 z20]

Rendering database examples

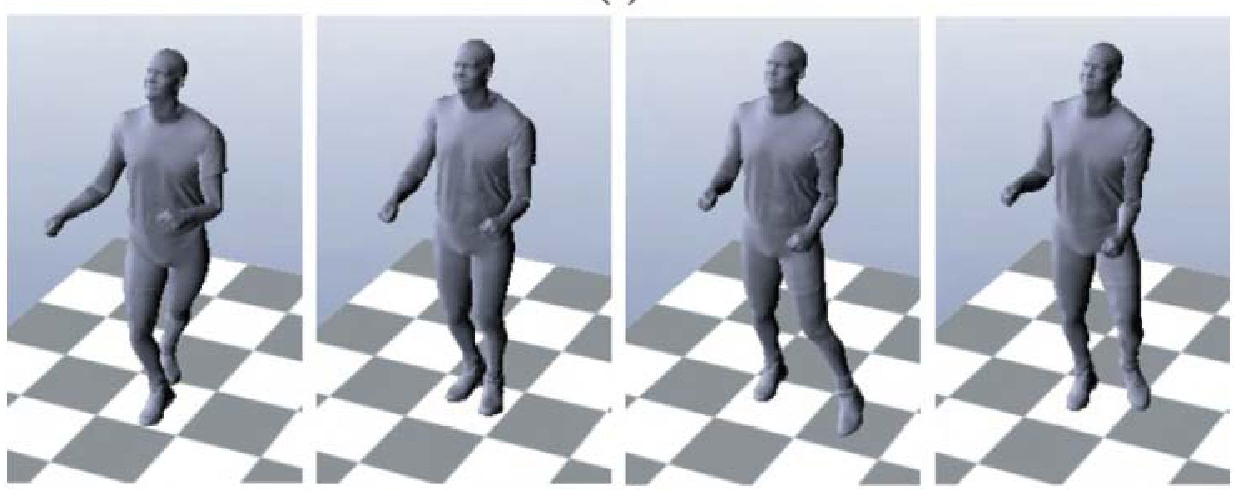
Learning Silhouette Features for Control of Human Motion • 1315

body
uration,
nt orientations



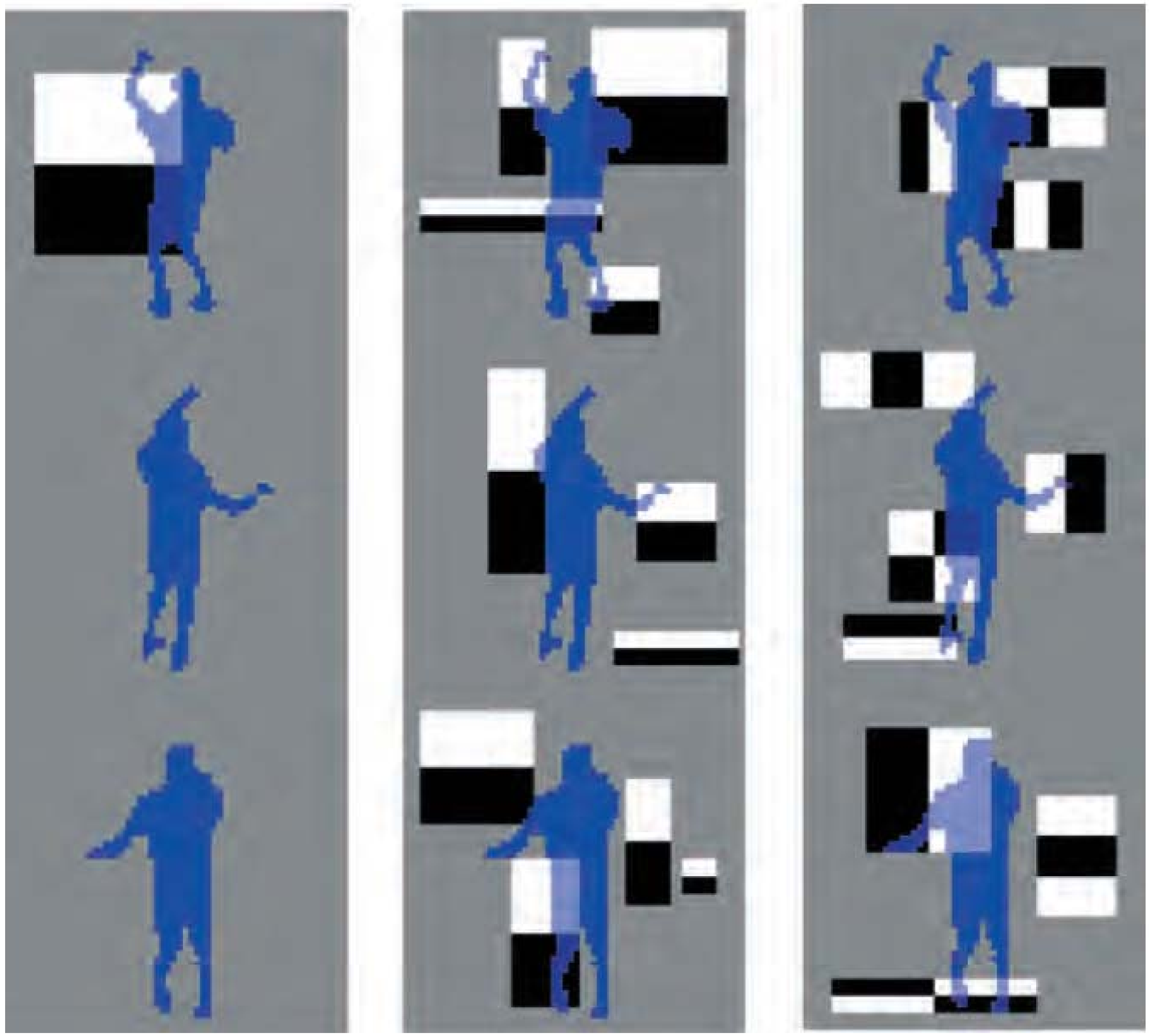
(a)

nt body
urations, same
tion

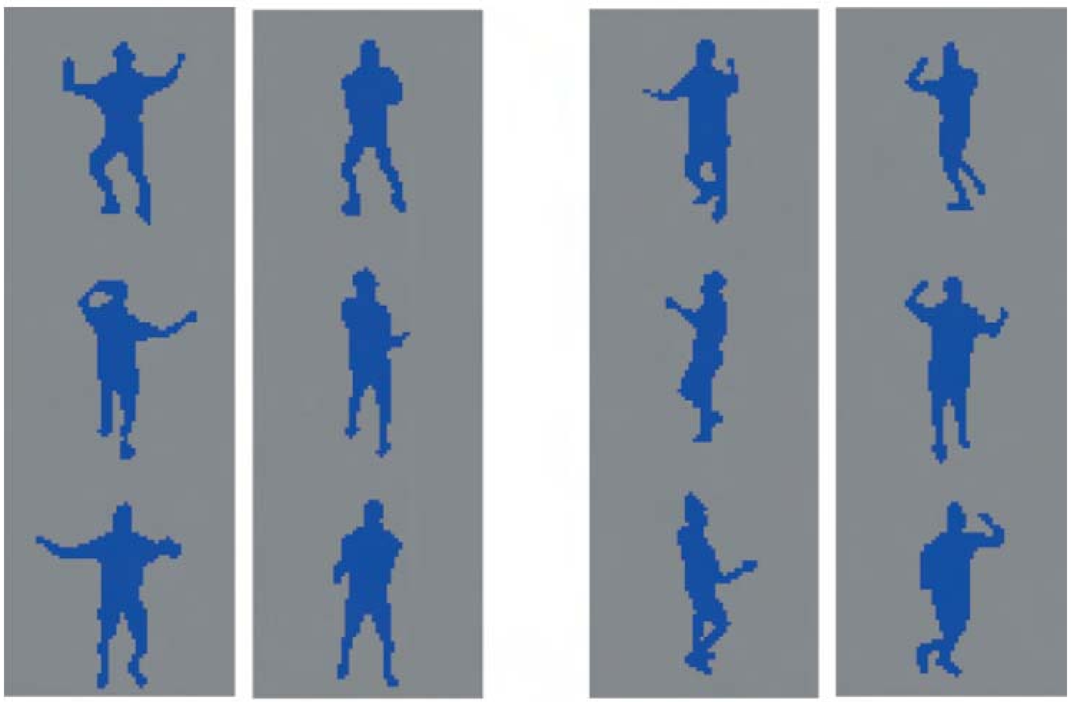


(b)

Possible silhouette features



Feature selection

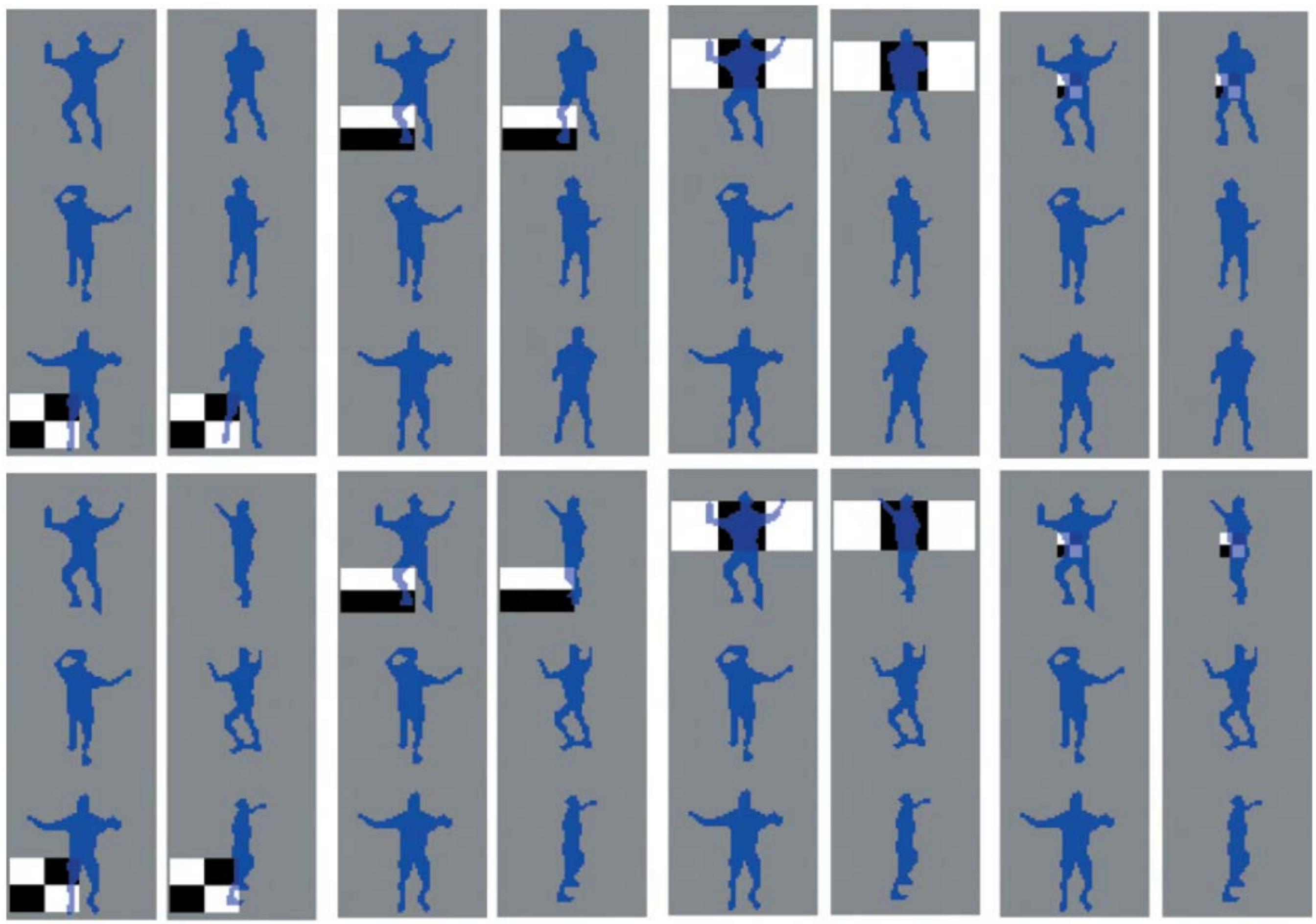


(a) Positive (b) Negative

- Want to find features that are discriminative for overall orientation, and specific body configuration
- Use boosting to choose features that separate “similar” and “dissimilar” pairs well

Feature selection

Some features selected with AdaBoost based on *paired* classification task

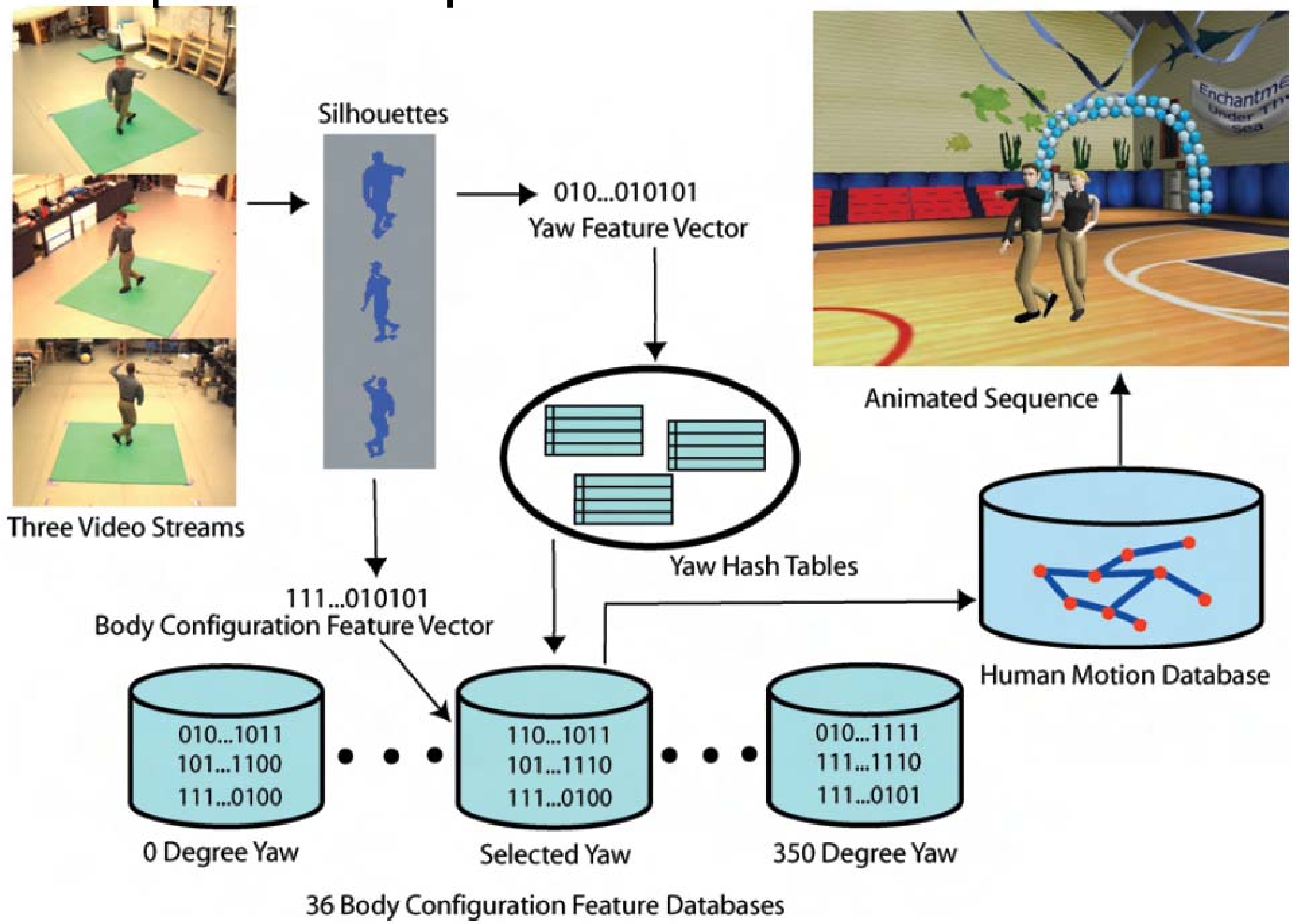


Feature 1 Feature 2 Feature 5 Feature 20

Two-character motion graph

- Dancing partners’ motions are highly correlated
- Extend motion graph to represent *partner’s* pose relative to user’s

Example-based pose estimation and animation



Ren, Shakhnarovich, Hodgins, Pfister, and Viola, 2005.

- http://graphics.cs.cmu.edu/projects/swing/

- Issues?

References

- Conditional density propagation for visual tracking (CONDENSATION), Isard and Blake, IJCV 1998.
- Lucas Kovar Michael Gleicher Frederic Pighin. Motion Graphs. *ACM Transactions on Graphics 21(3) (Proceedings of SIGGRAPH 2002).* July 2002.
- L. Ren, G. Shakhnarovich, J. Hodgins, H. Pfister, P. Viola, "Learning Silhouette Features for Control of Human Motion", ACM Transactions on Graphics,2005.