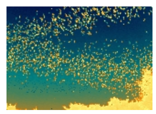
Tracking

Tuesday, Nov 25

Kristen Grauman
UT-Austin



Announcements

- My Wed office hours 1-2 pm
  – (and Thurs 2-3 pm)
- Pset 4 out today, due Thurs. Dec 4
  – Auto extension to Tues. Dec 9

Pset 4 overview

Part A: 100 pts

Track a corner through the video with feature-based matching

Part B: 25 pts

Generalize to multiple tracks, allow new tracks to form as new vehicles enter the frame.

E.C.: bg sub, Kalman filtering

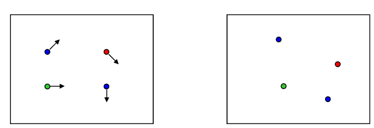
Outline

- Last time: Motion
  – Motion field and parallax
  – Optical flow, brightness constancy
  – Aperture problem
- Today:
  – Using optical flow (dense motion estimates) to recognize activities
  – Tracking
    - Tracking as inference
    - Linear models of dynamics
    - Kalman filters

Motion estimation techniques

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small

Direct methods: Estimating optical flow



$I(x,y,t-1)$ $I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field between them.
- Key assumptions
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Small motion:** points do not move very far
  - **Spatial coherence:** points move like their neighbors

Solving the aperture problem (grayscale image)

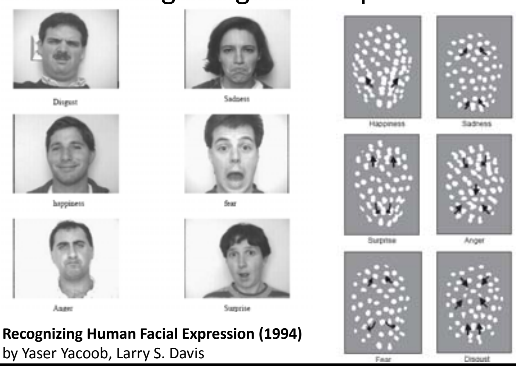
- How to get more equations for a pixel?
- **Spatial coherence constraint:** pretend the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

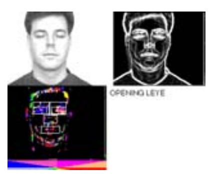
$$\underset{25\text{x}2}{A} \;\; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Using optical flow: recognizing facial expressions



Recognizing Human Facial Expression (1994)
by Yaser Yacoob, Larry S. Davis

Using optical flow: recognizing facial expressions



Using optical flow: action recognition at a distance

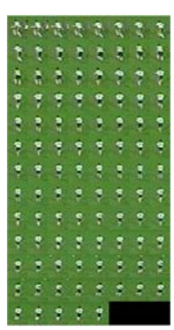
- Features = optical flow within a region of interest
- Classifier = nearest neighbors

The 30-Pixel Man

Challenge: low-res data, not going to be able to track each limb.

[Efros, Berg, Mori, & Malik 2003]
http://graphics.cs.cmu.edu/people/efros/research/action/

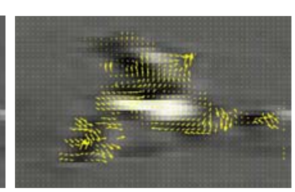
Using optical flow: action recognition at a distance



Correlation-based tracking
Extract person-centered frame window

[Efros, Berg, Mori, & Malik 2003]
http://graphics.cs.cmu.edu/people/efros/research/action/

Using optical flow: action recognition at a distance



Extract optical flow to describe the region's motion.

[Efros, Berg, Mori, & Malik 2003]
http://graphics.cs.cmu.edu/people/efros/research/action/

Using optical flow: action recognition at a distance

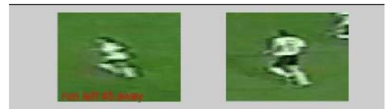
Input Sequence

Matched Frames

Use **nearest neighbor** classifier to name the actions occurring in new video frames.

[Efros, Berg, Mori, & Malik 2003]
http://graphics.cs.cmu.edu/people/efros/research/action/

Using optical flow: action recognition at a distance



Input Sequence

Matched NN Frame

Use **nearest neighbor** classifier to name the actions occurring in new video frames.

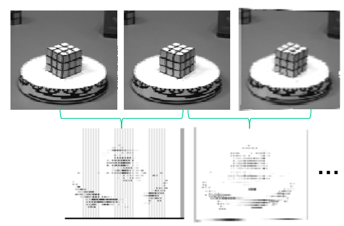
[Efros, Berg, Mori, & Malik 2003]
http://graphics.cs.cmu.edu/people/efros/research/action/

Do as I do: motion retargeting

- Include constraint for similarity within sequence as well as across sequences

Optical flow for tracking?

If we have more than just a pair of frames, we could compute flow from one to the next:

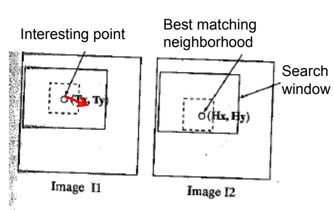


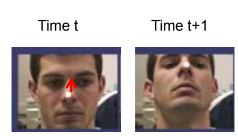
But flow only reliable for small motions, and we may have occlusions, textureless regions that yield bad estimates anyway…

Motion estimation techniques

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small
- **Feature-based methods**
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

Feature-based matching for motion





Search window is centered at the point where we last saw the feature, in image I1.

Best match = position where we have the highest normalized cross-correlation value.

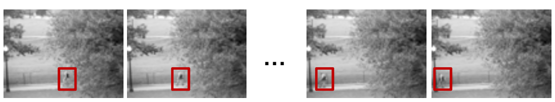
Feature-based matching for motion

- For a discrete matching search, what are the tradeoffs of the chosen **search window** size?

- Which patches to track?
  - Select interest points – e.g. corners
- Where should the search window be placed?
  - Near match at previous frame
  - **More generally, according to expected *dynamics* of the object**

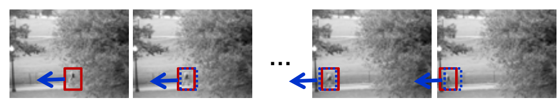
Detection vs. tracking

Detection vs. tracking



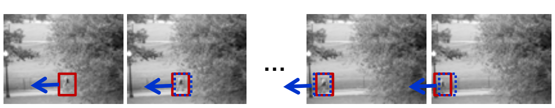
Detection: We detect the object independently in each frame and can record its position over time, e.g., based on blob's centroid or detection window coordinates

Detection vs. tracking



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

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.

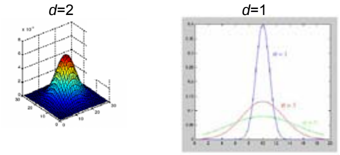
Tracking with dynamics

- Use model of expected motion to *predict* where objects will occur in next frame, even before seeing the image.
- **Intent**:
  – Do less work looking for the object, restrict the search.
  – Get improved estimates since measurement noise is tempered by smoothness, dynamics priors.
- **Assumption**: continuous motion patterns:
  – Camera is not moving instantly to new viewpoint
  – Objects do not disappear and reappear in different places in the scene
  – Gradual change in pose between camera and scene

Notation reminder

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

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



If x is 1-d, we just have one $\boldsymbol{\Sigma}$ parameter - → the variance: σ^2

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*.

State vs. observation

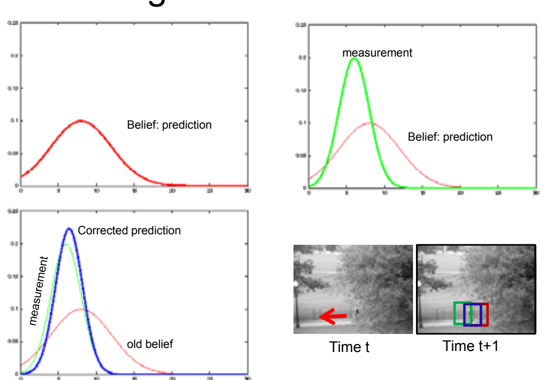
Hidden state : parameters of interest

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.

Tracking as inference: intuition



Standard 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 i 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

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

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

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

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

Tracking as inference

Recursive process:

- **Base case**: we have an initial prior P($\mathbf{X}_0$) on the state in absence of any evidence, which we can *correct* based on the first measurement $\mathbf{Y}_0=\mathbf{y}_0$.
- **Given corrected estimate** for frame ***t***:
  1) Predict for frame *t*+1
  2) Correct for frame *t*+1

Questions

- How to represent the known dynamics that govern the changes in the states?
- How to represent relationship between state and measurements, plus our uncertainty in the measurements?
- How to compute each cycle of updates?

Representation: We'll consider the class of *linear* dynamic models, with associated Gaussian pdfs.

Updates: via the Kalman filter.

Linear dynamic model

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

$$\underset{n \times 1}{\mathbf{x}_t} \sim N(\underset{n \times n}{\mathbf{D}}\,\underset{n \times 1}{\mathbf{x}_{t-1}};\Sigma_d)$$

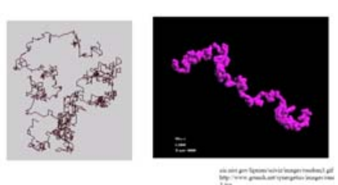
  - Measurement model: at every time step we get a noisy measurement of the state.

$$\underset{m \times 1}{\mathbf{y}_t} \sim N(\underset{m \times n}{\mathbf{M}}\,\underset{n \times 1}{\mathbf{x}_t};\Sigma_m)$$

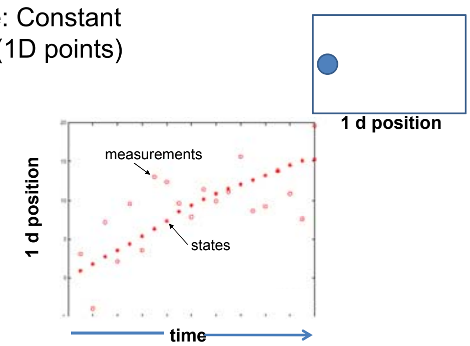
Example: randomly drifting points

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

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



Example: Constant velocity (1D points)



Example: Constant velocity (1D points)

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

- State vector: position *p* and velocity *v*

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

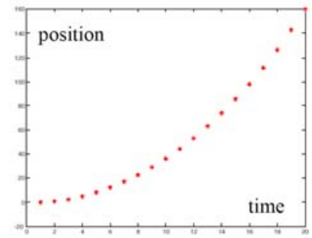
(greek letters denote noise terms)

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

- Measurement is position only

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

Example: Constant acceleration (1D points)



Example: Constant acceleration (1D points)

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

- State vector: position *p*, velocity *v*, and acceleration *a*.

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

(greek letters denote noise terms)

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

- Measurement is position only

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

Questions

- How to represent the known dynamics that govern the changes in the states?
- How to represent relationship between state and measurements, plus our uncertainty in the measurements?
- How to compute each cycle of updates?

Representation: We'll consider the class of *linear* dynamic models, with associated Gaussian pdfs.

Updates: via the Kalman filter.

The Kalman filter

- Method for tracking linear dynamical models in Gaussian noise
- The predicted/corrected state distributions are Gaussian
  - Only need to maintain the mean and covariance
  - The calculations are easy (all the integrals can be done in closed form)

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.

Time update ("Predict")

Measurement update ("Correct")

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

Time advances: t++

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

Mean and std. dev. of predicted state:

μ_t^-, σ_t^-

Mean and std. dev. of corrected state:

μ_t^+, σ_t^+

Kalman filter for 1d state

Want to represent and update

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

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

1D Kalman filter: **Prediction**

- Have linear dynamic model defining predicted state evolution, with noise

$$X_t \sim N\left(dx_{t-1}, \sigma_d^2\right)$$

- Want to estimate predicted distribution for next state

$$P\left(X_t \middle| y_0, \ldots, y_{t-1}\right) = N\left(\mu_t^-, (\sigma_t^-)^2\right)$$

- Update the mean:

$$\mu_t^- = d\mu_{t-1}^+$$

- Update the variance:

$$(\sigma_t^-)^2 = \sigma_d^2 + (d\sigma_{t-1}^+)^2$$

1D Kalman filter: **Correction**

- Have linear model defining the mapping of state to measurements:

$$Y_t \sim N\left(mx_t, \sigma_m^2\right)$$

- Want to estimate corrected distribution given latest meas.: $P\left(X_t \middle| y_0, \ldots, y_t\right) = N\left(\mu_t^+, (\sigma_t^+)^2\right)$
- Update the mean:

$$\mu_t^+ = \frac{\mu_t^- \sigma_m^2 + m y_t (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

- Update the variance:

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

Prediction vs. correction

$$\mu_t^+ = \frac{\mu_t^- \sigma_m^2 + m y_t (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2} \qquad (\sigma_t^+)^2 = \frac{\sigma_m^2 (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

- What if there is no prediction uncertainty $(\sigma_t^- = 0)$?

$$\mu_t^+ = \mu_t^- \qquad (\sigma_t^+)^2 = 0$$

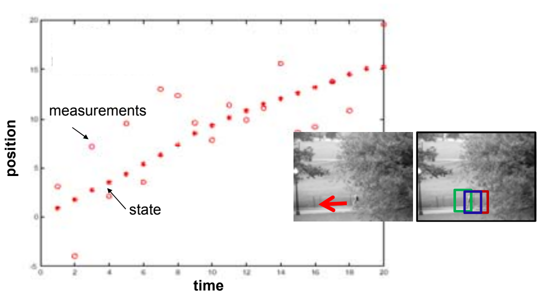
The measurement is ignored!

- What if there is no measurement uncertainty $(\sigma_m = 0)$?

$$\mu_t^+ = \frac{y_t}{m} \qquad (\sigma_t^+)^2 = 0$$

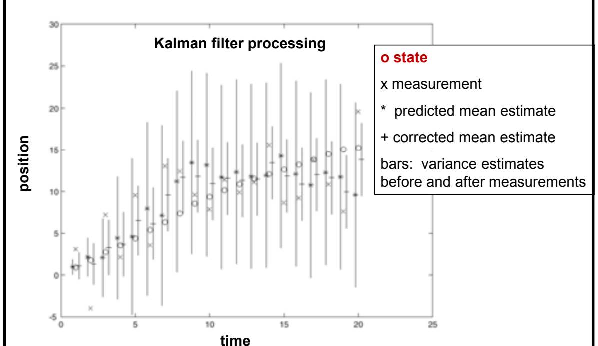
The prediction is ignored!

Recall: constant velocity example

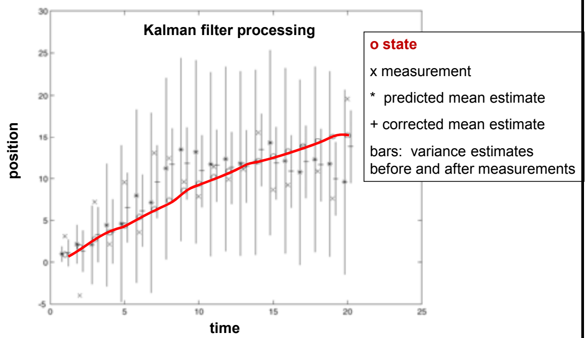


State is 2d: position + velocity
Measurement is 1d: position

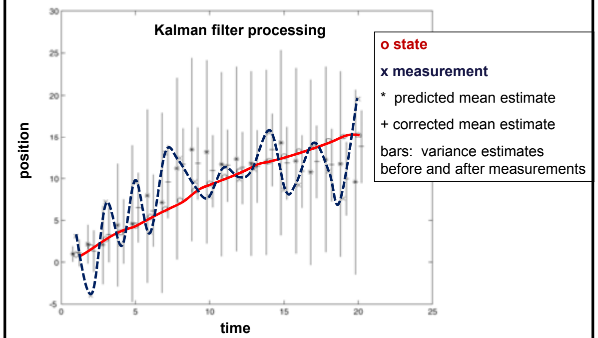
Constant velocity model



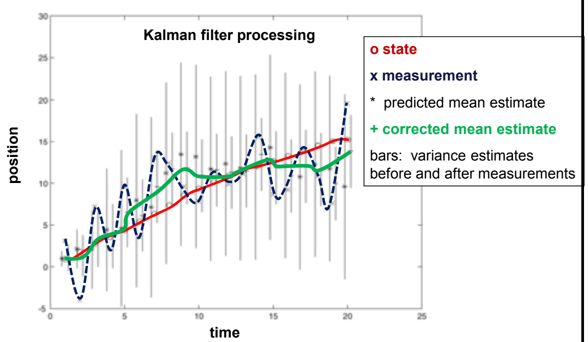
Constant velocity model



Constant velocity model



Constant velocity model



Kalman filter: General case (> 1dim)

What if state vectors have more than one dimension?

PREDICT

$$x_t^- = D_t x_{t-1}^+$$

$$\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_{d_t}$$

CORRECT

$$K_t = \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t} \right)^{-1}$$

$$x_t^+ = x_t^- + K_t \boxed{\left(y_t - M_t x_t^- \right)}$$

$$\Sigma_t^+ = \left(I - K_t M_t \right) \Sigma_t^-$$

More weight on residual when measurement error covariance approaches 0.

Less weight on residual as a priori estimate error covariance approaches 0.

Tracking: issues

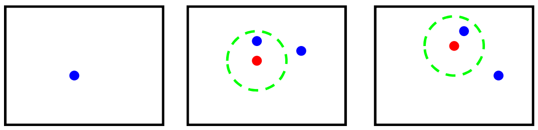
- Initialization
  - Often done manually
  - Background subtraction, detection can also be used
- Data association, multiple tracked objects
  - Occlusions

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.

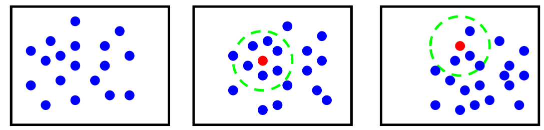
Data association

- Simple strategy: only pay attention to the measurement that is “closest” to the prediction



Data association

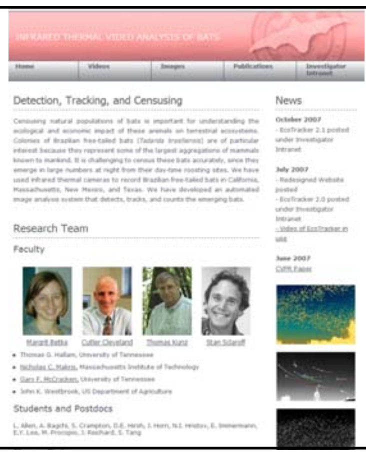
- Simple strategy: only pay attention to the measurement that is “closest” to the prediction



Doesn’t always work…
Alternative: keep track of **multiple hypotheses** at once.

Source: Lana Lazebnik

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



Tracking: issues

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

Drift

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

Source: Lana Lazebnik

Summary

- Using optical flow to recognize activities
  - Low-level feature captures motion patterns in a region of interest
- Tracking as inference
  - Goal: estimate posterior of object position given measurement
- Linear models of dynamics
  - Represent state evolution and measurement models
- Kalman filters
  - Recursive prediction/correction updates to refine measurement