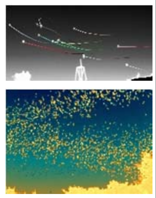
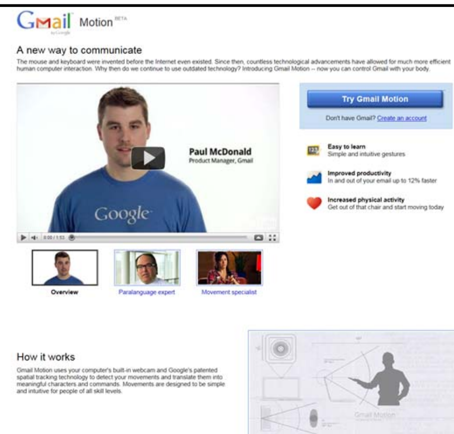
Tracking

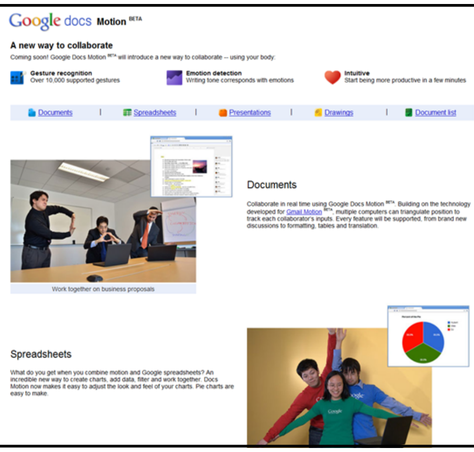
Wednesday, April 27

Kristen Grauman

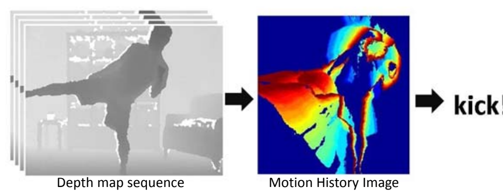
UT-Austin







Pset 5



Nearest neighbor action classification with Motion History Images + Hu moments

Kristen Grauman

Normalized Euclidean distance

$$D(h_1, h_2) = \sqrt{\sum_{i=1}^{d} \frac{(h_1(i) - h_2(i))^2}{\sigma_i^2}}$$

Normalize according to variance in each dimension

What does this do for our distance computation?

Kristen Grauman

Leave-one-out cross validation

- Cycle through data points, treating each one as the "test" case in turn, and training with the remaining labeled examples.
- Report results over all such test cases

Outline

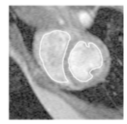
- Today: Tracking
  - Tracking as inference
  - Linear models of dynamics
  - Kalman filters
  - General challenges in tracking

Tracking: some applications

Body pose tracking, activity recognition

Censusing a bat population

Video-based interfaces



Medical apps

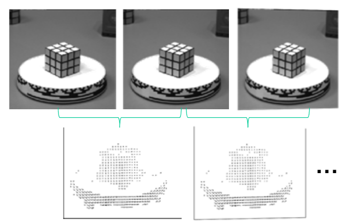
Surveillance

Kristen Grauman

Why is tracking challenging?

Optical flow for tracking?

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

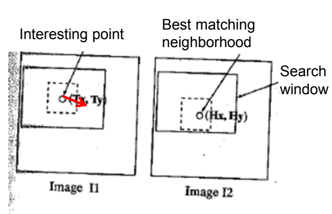


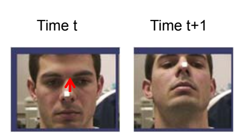
But flow only reliable for small motions, and we may have occlusions, textureless regions that yield bad estimates anyway…

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.

Kristen Grauman

Example: A Camera Mouse

Video interface: use feature tracking as mouse replacement

- User clicks on the feature to be tracked
- Take the 15x15 pixel square of the feature
- In the next image do a search to find the 15x15 region with the highest correlation
- Move the mouse pointer accordingly
- Repeat in the background every 1/30th of a second

James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

Kristen Grauman

Example: A Camera Mouse

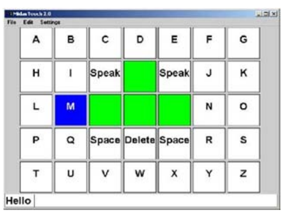
Specialized software for communication, games

James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

Kristen Grauman

A Camera Mouse

Specialized software for communication, games

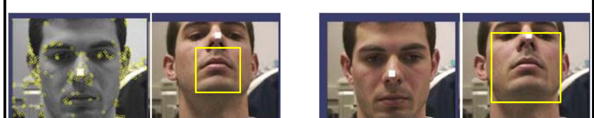


James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

Kristen Grauman

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, taking into account the expected *dynamics* of the object**

Kristen Grauman

Detection vs. tracking

t=1 t=2 … t=20 t=21

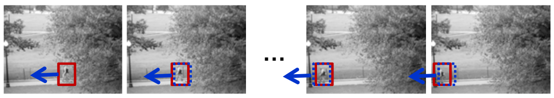
Kristen Grauman

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

Kristen Grauman

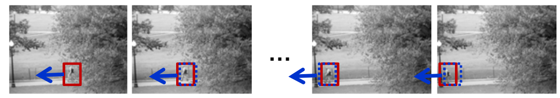
Detection vs. tracking



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.

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.

Kristen Grauman

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

Kristen Grauman

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.

Kristen Grauman

State vs. observation

Hidden state : parameters of interest

Measurement : what we get to directly observe

Kristen Grauman

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.

Kristen Grauman

Tracking as inference: intuition

Time t   Time t+1

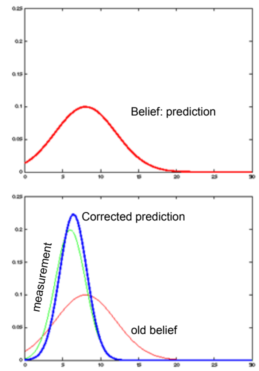
Belief

Measurement

Corrected prediction

Kristen Grauman

Tracking as inference: intuition



Time t   Time t+1

Kristen Grauman

Independence assumptions

- Only immediate past state influences current state

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

dynamics model

- Measurement at time t depends on 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

Kristen Grauman

Tracking as inference

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

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

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

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

Kristen Grauman

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.

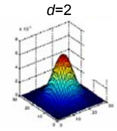
Kristen Grauman

Notation reminder

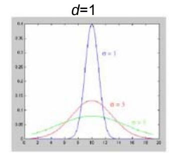
$$\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- 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*.

d=2



d=1



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

Kristen Grauman

Linear dynamic model

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

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

n x 1   n x n   n x 1

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

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

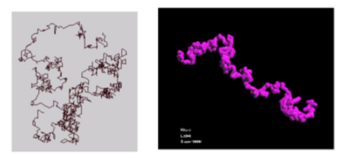
m x 1   m x n   n x 1

Kristen Grauman

Example: randomly drifting points

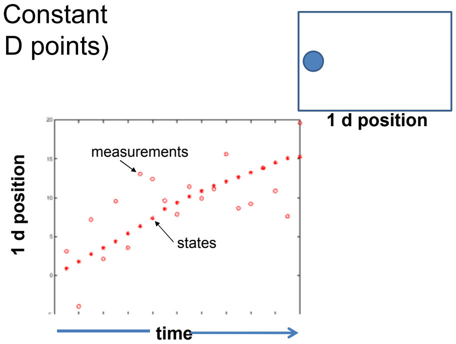
$$\mathbf{x}_t \sim N(\mathbf{D}\mathbf{x}_{t-1}; \boldsymbol{\Sigma}_d)$$

- 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**



Kristen Grauman

Example: Constant velocity (1D points)



Kristen Grauman

Example: Constant velocity (1D points)

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

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

$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} \qquad p_t =$$

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

- Measurement is position only

$$y_t = M x_t + noise =$$

Kristen Grauman

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.

Kristen Grauman

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

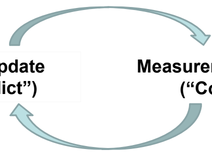
Kristen Grauman

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")

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

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

Mean and std. dev. of predicted state:

μ_t^-, σ_t^-

Time advances: t++

Mean and std. dev. of corrected state:

μ_t^+, σ_t^+

Kristen Grauman

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
$$P\left(X_t | y_0, \ldots, y_{t-1}\right) = N\left(\mu_t^-, (\sigma_t^-)^2\right)$$
- Update the mean:
$$\mu_t^- = d\mu_{t-1}^+$$
- Update the variance:
$$(\sigma_t^-)^2 = \sigma_d^2 + (d\sigma_{t-1}^+)^2$$

Lana Lazebnik

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.: $P\left(X_t | y_0, \ldots, y_t\right) = N\left(\mu_t^+, (\sigma_t^+)^2\right)$
- 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:
$$(\sigma_t^+)^2 = \frac{\sigma_m^2 (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2}$$

Lana Lazebnik

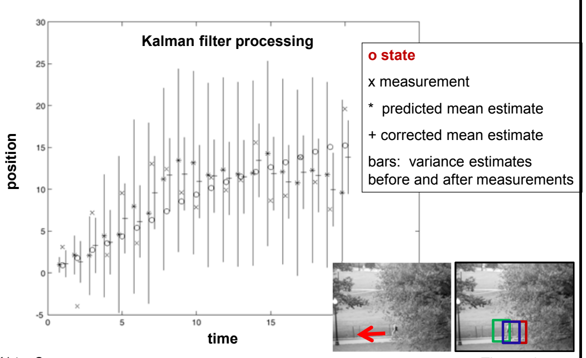
Prediction vs. correction

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

- 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!

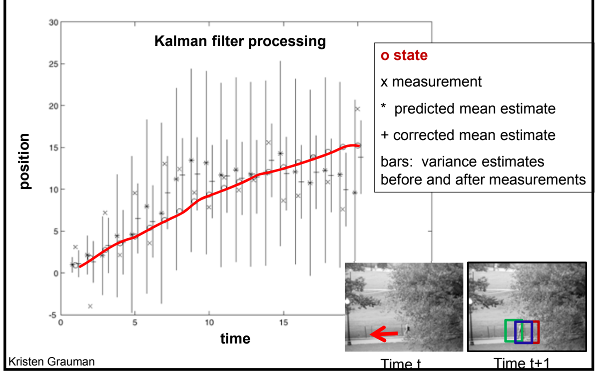
Lana Lazebnik

Constant velocity model



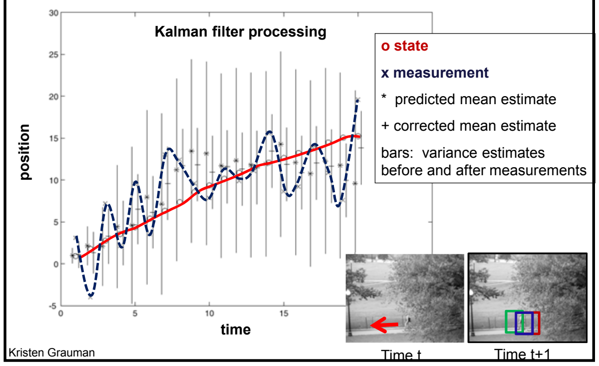
Kristen Grauman

Constant velocity model



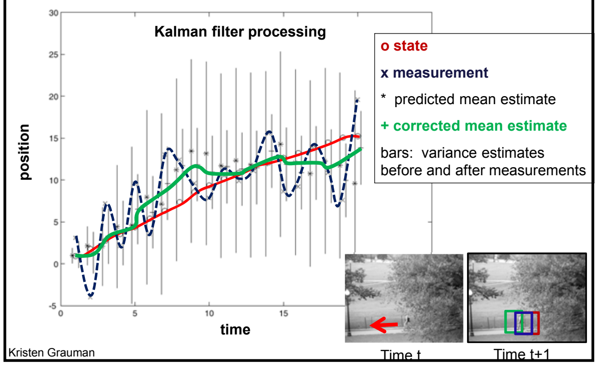
Kristen Grauman

Constant velocity model



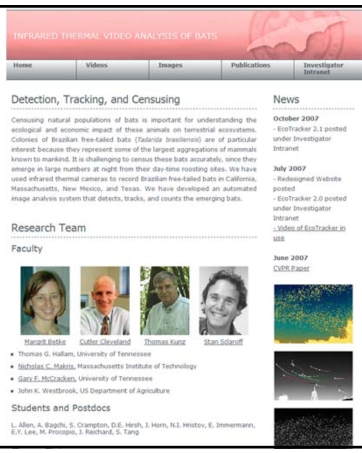
Kristen Grauman

Constant velocity model



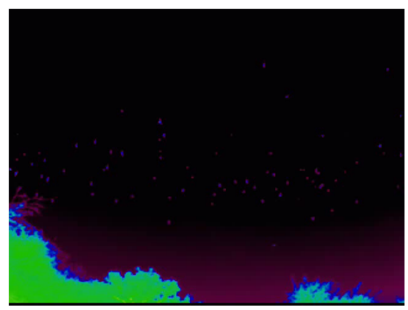
Kristen Grauman

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



Kristen Grauman

A bat census



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

Kristen Grauman

Video synopsis

- http://www.vision.huji.ac.il/video-synopsis/

Kristen Grauman

Tracking: issues

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

Tracking: issues

- **Initialization**
  - Often done manually
  - Background subtraction, detection can also be used
- **Data association**, multiple tracked objects
  - Occlusions, clutter
  - Which measurements go with which tracks?


Tracking: issues

- **Initialization**
  - Often done manually
  - Background subtraction, detection can also be used
- **Data association**, multiple tracked objects
  - Occlusions, clutter
- **Deformable** and articulated objects

Recall: tracking via deformable contours

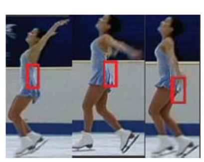
1. Use final contour/model extracted at frame *t* as an initial solution for frame *t+1*
2. Evolve initial contour to fit exact object boundary at frame *t+1*
3. Repeat, initializing with most recent frame.


Visual Dynamics Group, Dept. Engineering Science, University of Oxford.

Tracking: issues

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

Drift


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

Summary

- 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
- General tracking challenges