CS 2770: Computer Vision

Motion:
Tracking, Pose and Actions

Prof. Adriana Kovashka
University of Pittsburgh
April 4, 2017
In this lecture

• Tracking how an object moves
• Estimating human pose
• Recognizing human actions
Motion: Why is it useful?
Motion: Why is it useful?

- Even "impoverished" motion data can evoke a strong percept

Tracking: some applications

Body pose tracking, activity recognition

Censusing a bat population

Video-based interfaces

Medical apps

Surveillance
Tracking examples

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

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

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

Body: https://www.youtube.com/watch?v=Ahy0Gh69-M

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

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

- Video interface: use feature tracking as mouse replacement
- Specialized software for communication, games

James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/
Things that make visual tracking difficult

• Erratic movements, moving very quickly
• Occlusions, leaving and coming back
• Surrounding similar-looking objects

Adapted from Amin Sadeghi
Strategies for tracking

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

• Tracking w/ dynamics: Using model of expected motion, *predict* object location in next frame
  – Restrict search for the object
  – Measurement noise is reduced by trajectory smoothness
  – Robustness to missing or weak observations
  – Assumptions: Camera is not moving instantly to new viewpoint, objects do not disappear/reappear in different places in the scene
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 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 the object’s motion pattern.
Tracking: prediction + correction

Belief

Measurement

Corrected prediction
Tracking: prediction + correction

belief: prediction

measurement

corrected prediction

old belief

Time t

Time t+1
General model for tracking

- **State $X$:** The actual state of the moving object that we want to estimate but cannot observe
  - E.g. position, velocity

- **Observations $Y$:** Our actual measurement or observation of state $X$, which can be very noisy

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

- Our goal is to recover the most likely state $X_t$ given:
  - All observations so far, i.e. $y_1, y_2, ..., y_t$
  - Knowledge about dynamics of state transitions

Adapted from Amin Sadeghi and Kristen Grauman
Steps of tracking

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

\[
P(X_t | Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1})
\]
Steps of tracking

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

\[
P(X_t | Y_0 = y_0, \ldots , Y_{t-1} = y_{t-1})
\]

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

\[
P(X_t | Y_0 = y_0, \ldots , Y_{t-1} = y_{t-1}, Y_t = y_t)
\]
Problem statement

• We have models for

  Likelihood of next state given current state (dynamics model):
  \[ P(X_t | X_{t-1}) \]

  Likelihood of observation given the state (observation or measurement model):
  \[ P(Y_t | X_t) \]

• We want to recover, for each t:
  \[ P(X_t | y_0, \ldots, y_t) \]
The Kalman filter

- Linear dynamics model: state undergoes linear transformation plus Gaussian noise

- Observation model: measurement is linearly transformed state plus Gaussian noise

- The predicted/corrected state distributions are Gaussian
  - You only need to maintain the mean and covariance
  - The calculations are easy
Notation: Normal distribution

\[ x \sim N(\mu, \Sigma) \]

- Random variable with Gaussian probability distribution that has the mean vector \( \mu \) and covariance matrix \( \Sigma \).
- \( x \) and \( \mu \) are \( d \)-dimensional, \( \Sigma \) is \( d \times d \).

If \( x \) is 1-d, we just have one \( \Sigma \) parameter \( \rightarrow \) the variance: \( \sigma^2 \)
Dynamics and observation models

- **Dynamics** model (represents evolution of state over time)

$$P(X_t \mid X_{t-1}) = N(\mu_d + DX_{t-1}, \Sigma_d)$$

- **Observation or measurement** model (at every time step we get a noisy measurement of the state)

$$P(y_t \mid X_t) = N(\mu_m + MX_t, \Sigma_m)$$

Adapted from Kristen Grauman, Simon Prince
Example: Constant velocity (1D points)

- 1 d position
- 1 d position
- time
- measurements
- states

Kristen Grauman
Example: Constant velocity (1D points)

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

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

$$v_t = v_{t-1} + \xi$$

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

- Measurement is position only

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

Prediction:

\[ P(X_t \mid y_0, \ldots, y_{t-1}) = \int P(X_t \mid X_{t-1})P(X_{t-1} \mid y_0, \ldots, y_{t-1})dX_{t-1} \]

Correction:

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

Adapted from Amin Sadeghi
Simplifying assumptions

• Only the immediate past matters

\[ P(X_t|X_0, \ldots, X_{t-1}) \]

dynamics model
Simplifying assumptions

- Only the immediate past matters

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

- Measurements depend only on the current state

\[ P(Y_t | X_0, Y_0 \ldots, X_{t-1}, Y_{t-1}, X_t) \]
Prediction

- Prediction involves representing \( P(X_t, y_0, \ldots, y_{t-1}) \) given \( P(X_{t-1}, y_0, \ldots, y_{t-1}) \)

\[
P(X_t | y_0, \ldots, y_{t-1}) = \int P(X_t, X_{t-1} | y_0, \ldots, y_{t-1}) dX_{t-1}
\]

Law of total probability

\[
\Pr(A) = \sum_n \Pr(A \cap B_n)
\]
Prediction

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

\[
P(X_t | y_0, \ldots, y_{t-1}) = \int P(X_t, X_{t-1} | y_0, \ldots, y_{t-1}) dX_{t-1}
\]

\[
= \int P(X_t | X_{t-1}, y_0, \ldots, y_{t-1}) P(X_{t-1} | y_0, \ldots, y_{t-1}) dX_{t-1}
\]

Conditioning on \( X_{t-1} \)

\[ P(A, B) = P(A | B)P(B) \]
Prediction

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

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

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

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

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

Independence assumption (only immediate past state matters)

Adapted from Amin Sadeghi
Prediction

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

\[
P(X_t | y_0, \ldots, y_{t-1}) = \int P(X_t, X_{t-1} | y_0, \ldots, y_{t-1}) dX_{t-1}
\]

\[
= \int P(X_t | X_{t-1}, y_0, \ldots, y_{t-1}) P(X_{t-1} | y_0, \ldots, y_{t-1}) dX_{t-1}
\]

\[
= \int P(X_t | X_{t-1}) P(X_{t-1} | y_0, \ldots, y_{t-1}) dX_{t-1}
\]

- dynamics model
- corrected estimate from previous step

Amin Sadeghi
Correction

• Correction involves computing given predicted value

\[ P(X_t|y_0,\ldots,y_t) \]

\[ P(X_t|y_0,\ldots,y_{t-1}) \]
Correction

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

\[
P(X_t | y_0, \ldots, y_{t}) = \frac{P(y_t | X_t, y_0, \ldots, y_{t-1}) P(X_t | y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})}
\]

Bayes’ Rule

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]
Correction

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

\[
P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t, y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})} P(X_t | y_0, \ldots, y_{t-1})
\]

\[
= \frac{P(y_t | X_t) P(X_t | y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})}
\]

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

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

\[
P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t, y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})} \cdot P(X_t | y_0, \ldots, y_{t-1})
\]

\[
= \frac{P(y_t | X_t) \cdot P(X_t | y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})}
\]

\[
= \frac{P(y_t | X_t) \cdot P(X_t | y_0, \ldots, y_{t-1})}{\int P(y_t | X_t) \cdot P(X_t | y_0, \ldots, y_{t-1}) dX_t}
\]

Law of total probability

Adapted from Amin Sadeghi
Correction

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

\[
P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t, y_0, \ldots, y_{t-1}) P(X_t | y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})}
\]

\[
= \frac{P(y_t | X_t) P(X_t | y_0, \ldots, y_{t-1})}{P(y_t | y_0, \ldots, y_{t-1})}
\]

\[
= \int P(y_t | X_t) P(X_t | y_0, \ldots, y_{t-1}) dX_t
\]
Summary: Prediction and correction

- **Prediction:**
  
  Know corrected state from previous time step, and all measurements up to (excluding) the current one →
  
  Predict distribution over next state

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

- **Correction:**
  
  Know prediction of state, and next measurement →
  
  Update distribution over current state

$$P(X_t \mid y_0, \ldots, y_t)$$
Example w/ constant velocity

Kalman filter processing

- state
- measurement
- predicted mean estimate
- corrected mean estimate
- bars: variance estimates before and after measurements

Kristen Grauman
Example w/ constant velocity

Kalman filter processing

state
measurement
predicted mean estimate
corrected mean estimate
bars: variance estimates before and after measurements
Example w/ constant velocity

Kalman filter processing

- $o$ state
- $x$ measurement
  - * predicted mean estimate
  - + corrected mean estimate

Bars: variance estimates before and after measurements

Kristen Grauman

Time $t$

Time $t+1$
Example w/ constant velocity
Example w/ constant velocity

Ground Truth  Observation  Correction
In this lecture

• Tracking how an object moves
• Estimating human pose
• Recognizing human actions
What is an action/activity?

Action: a transition from one state to another

- What is the name of the action?
- Who is the actor?
- How is the state of the actor changing?
- What (if anything) is being acted on?
- How is that thing changing?
- What is the purpose of the action (if any)?

Adapted from Derek Hoiem
How can we identify actions?

Motion

Pose

Held Objects

Nearby Objects

Derek Hoiem
Real-Time Human Pose Recognition in Parts from Single Depth Images

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, Andrew Blake

Best paper award at CVPR 2011
The mission

- Recognize large variety of human poses, all shapes & sizes
- Limited compute budget
  - super-real time on Xbox 360 to allow games to run concurrently

Adapted from Jamie Shotton
The approach: body part recognition

right hand

neck

right elbow

left shoulder
Body part recognition

- No temporal information
  - frame-by-frame

- Local pose estimate of parts
  - each pixel & each body joint treated independently

- Very fast
  - simple depth image features
  - decision forest classifier

Adapted from Jamie Shotton
The Kinect pose estimation pipeline

capture depth image & remove bg

infer body parts per pixel

cluster pixels to hypothesize body joint positions

fit model & track skeleton
Classifying pixels

- Compute $P(c_i | w_i)$
  - 31 body parts considered
  - body part $c_i$
  - image window $w_i$

- Discriminative approach
  - learn classifier $P(c_i | w_i)$ from training data

Adapted from Jamie Shotton
Fast depth image features

- Depth comparisons
  - Very fast to compute

\[ f_\theta(I, x) = d_I(x) - d_I(x + \Delta) \]

Adapted from Jamie Shotton
To classify pixel $x$, start here

To classify pixel $x$, start here

$f_{\Theta}(I, x; \Delta_1) > t_1$

$f_{\Theta}(I, x; \Delta_2) > t_2$

Adapted from Jamie Shotton
Decision forest classifier

- Trained on different random subset of images
  - “bagging” helps avoid over-fitting
- Average tree posteriors
  \[ P(c|I, x) = \frac{1}{T} \sum_{t=1}^{T} P_t(c|I, x) \]
Number and depth of trees

Adapted from Jamie Shotton
input depth

inferred body parts

front view  side view  top view

inferred joint positions (modes found using mean shift)

no tracking or smoothing

Jamie Shotton
Representing Actions

• Via the pose of persons in the video, how it changes
Representing Actions

• Via tracked points

Adapted from Derek Hoiem

Matikainen et al. 2009
Representing Actions

- Via spatio-temporal interest points (corners in space+time)
Learning realistic human actions from movies, Laptev et al. 2008

“Talk on phone”

“Get out of car”
Approach

• Space-time interest point detectors
• Descriptors
  – HOG, HOF
• Pyramid histograms (3x3x2)
• SVMs with Chi-Squared Kernel

Interest Points

Spatio-Temporal Binning

Derek Hoiem, figures from Ivan Laptev
## Results

<table>
<thead>
<tr>
<th>Task</th>
<th>HoG BoF</th>
<th>HoF BoF</th>
<th>Best channel</th>
<th>Best combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH multi-class</td>
<td>81.6%</td>
<td>89.7%</td>
<td>91.1% (hof h3x1 t3)</td>
<td>91.8% (hof 1 t2, hog 1 t3)</td>
</tr>
<tr>
<td>Action AnswerPhone</td>
<td>13.4%</td>
<td>24.6%</td>
<td>26.7% (hof h3x1 t3)</td>
<td>32.1% (hof o2x2 t1, hof h3x1 t3)</td>
</tr>
<tr>
<td>Action GetOutCar</td>
<td>21.9%</td>
<td>14.9%</td>
<td>22.5% (hof o2x2 1)</td>
<td>41.5% (hof o2x2 t1, hog h3x1 t1)</td>
</tr>
<tr>
<td>Action HandShake</td>
<td>18.6%</td>
<td>12.1%</td>
<td>23.7% (hog h3x1 1)</td>
<td>32.3% (hog h3x1 t1, hog o2x2 t3)</td>
</tr>
<tr>
<td>Action HugPerson</td>
<td>29.1%</td>
<td>17.4%</td>
<td>34.9% (hog h3x1 t2)</td>
<td>40.6% (hog 1 t2, hog o2x2 t2, hog h3x1 t2)</td>
</tr>
<tr>
<td>Action Kiss</td>
<td>52.0%</td>
<td>36.5%</td>
<td>52.0% (hog 1 1)</td>
<td>53.3% (hog 1 t1, hog 1 t1, hog o2x2 t1)</td>
</tr>
<tr>
<td>Action SitDown</td>
<td>29.1%</td>
<td>20.7%</td>
<td>37.8% (hog 1 t2)</td>
<td>38.6% (hog 1 t2, hog 1 t3)</td>
</tr>
<tr>
<td>Action SitUp</td>
<td>6.5%</td>
<td>5.7%</td>
<td>15.2% (hog h3x1 t2)</td>
<td>18.2% (hog o2x2 t1, hog o2x2 t2, hog h3x1 t2)</td>
</tr>
<tr>
<td>Action StandUp</td>
<td>45.4%</td>
<td>40.0%</td>
<td>45.4% (hog 1 1)</td>
<td>50.5% (hog 1 t1, hog 1 t2)</td>
</tr>
</tbody>
</table>

Derek Hoiem, figures from Ivan Laptev
Detecting Activities of Daily Living in First-person Camera Views

Hamed Pirsiavash, Deva Ramanan

CVPR 2012
Wearable ADL detection

It is easy to collect natural data
Challenges
What features to use?

Low level features

High level features

Space-time interest points
Laptev, IJCV’05

Human pose

Difficulties of pose:
• Detectors are not accurate enough
• Not useful in first person camera views

Adapted from Hamed Pirsiavash
Challenges
What features to use?

Low level features

High level features

Space-time interest points
Laptev, IJCV’05

Human pose

Object-centric features

Difficulties of pose:
• Detectors are not accurate enough
• Not useful in first person camera views
Appearance feature: bag of objects

Video clip → Bag of detected objects → SVM classifier

Objects: fridge, stove, TV
Temporal pyramid
Coarse to fine correspondence matching with a multi-layer pyramid

Inspired by “Spatial Pyramid” CVPR’06 and “Pyramid Match Kernels” ICCV’05

Video clip

SVM classifier

Temporal pyramid descriptor

Hamed Pirsiavash
Accuracy on 18 action categories

- Our model: 40.6%
- STIP baseline: 22.8%
Human-Object Interaction

Holistic image based classification

Integrated reasoning

• Human pose estimation
Human-Object Interaction

Holistic image based classification

Integrated reasoning
  • Human pose estimation
  • Object detection
Human-Object Interaction

Holistic image based classification

Integrated reasoning
  • Human pose estimation
  • Object detection
  • Action categorization

Activity: Tennis Forehand
Human pose estimation is challenging.

- Felzenszwalb & Huttenlocher, 2005
- Ren et al, 2005
- Ramanan, 2006
- Ferrari et al, 2008
- Yang & Mori, 2008
- Andriluka et al, 2009
- Eichner & Ferrari, 2009
Human pose estimation & Object detection

Facilitate

Given the object is detected.
Human pose estimation & Object detection

Object detection is challenging

Small, low-resolution, partially occluded

Image region similar to detection target
Human pose estimation & Object detection

Facilitate

Given the pose is estimated.
Human pose estimation & Object detection

Mutual Context

Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities, B. Yao and L. Fei-Fei, CVPR 2010
http://www-CS.stanford.edu/groups/vision/documents/YaoFei-Fei_CVPR2010b_full.pdf