Visual Tracking

Introduction to Computer Vision
CSE 152
Lecture 13

Announcements

• Homework 3 is due May 18, 11:59 PM
• Reading:
  – Chapter 11: Tracking

Main tracking notions

• State: usually a finite number of parameters (a vector) that characterizes the “state” (e.g., location, size, pose, deformation) of the object being tracked.
• Dynamics: How does the state change over time? How is that changed constrained?
• Representation: How do you represent the object being tracked?
• Prediction: Given the state at time \( t-1 \), what is an estimate of the state at time \( t \)?
• Correction: Given the predicted state at time \( t \) and a measurement at time \( t \), update the state.
• Initialization: What is the state at time \( t = 0 \)?

What is the state?

• 2D image location \( \Phi=(u,v) \)
• Image location + scale \( \Phi=(u,v,s) \)
• Image location + scale + orientation \( \Phi=(u,v,s,\theta) \)
• Affine transformation
• 3D pose
• 3D pose plus internal shape parameters (some may be discrete)
  – e.g., for a face, 3D pose + facial expression using FACS + eye state (open/closed)
• Collections of control points specifying a spline
• Above, but for multiple objects (e.g., tracking a formation of airplanes)
• Augment above with temporal derivatives \( (\dot{\phi}, \dot{\phi}) \)

State Examples

– Object is ball, state is 3D position + velocity, measurements are derived from stereo pairs
– Object is person, state is body configuration, measurements are derived from video frames
– What is state here?
Example: Blob Tracker

- From input image $I(u,v)$ at time $t$, create a binary image by applying a function $f(I(u,v))$
- Clean up binary image using morphological operators
- Perform connected component exploration to find “blobs” (i.e., connected regions)
- Compute their moments (mean and covariance of region coordinates) and use as state
- Using state estimate from time $t-1$ and perform “data association” to identify state at time $t$

Blob Tracking in IR Images

- Threshold about body temperature
- Connected component analysis
- Position, scale, orientation of regions
- Temporal coherence

Tracking: Probabilistic framework

- Very general model
  - Assume there are moving objects that have an underlying state $X$
  - There are observations (measurements) $Y$, some of which are functions of this state
- Over time
  - The state changes: $X_{t-1}$, $X_t$, $X_{t+1}$
  - There are new observations: $Y_{t-1}$, $Y_t$, $Y_{t+1}$

Tracking State

- Instead of “knowing state” at each instant, we treat the state as random variables $X_t$ characterized by a pdf $P(X_t)$ or perhaps conditioned on other random variables, e.g., $P(X_t | X_{t-1})$
- The observation (measurement) $Y_t$ is a random variable conditioned on the state $P(Y_t | X_t)$
- Generally, we don’t observe the state — it’s hidden

Three main steps

- **Prediction**: we have seen $y_0, \ldots, y_{t-1}$ — what state does this set of measurements predict for the $t$th frame? to solve this problem, we need to obtain a representation of $P(X_t | Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1})$.
- **Data association**: Some of the measurements obtained from the $i$th frame may tell us about the object’s state. Typically, we use $P(X_t | Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1})$ to identify these measurements.
- **Correction**: now that we have $y_t$ — the relevant measurements — we need to compute a representation of $P(X_t | Y_0 = y_0, \ldots, Y_t = y_t)$.

Simplifying Assumptions

- Only the immediate past matters: formally, we require
  \[ P(X_t | X_0, \ldots, X_{t-1}) = P(X_t | X_{t-1}) \]
- Measurements depend only on the current state: we assume that $Y_t$ is conditionally independent of all other measurements given $X_t$. This means that
  \[ P(Y_t | Y_0, , \ldots, Y_{t-1}, X_t) = P(Y_t | X_t) \]
Tracking as induction

- Assume data association is done
  - Sometimes challenging in cluttered scenes. See work by Christopher Rasmussen on Joint Probabilistic Data Association Filters (JPDAF).
- Do correction for frame $i = 0$
- Assume we have corrected estimate for frame $i$
  - We can predict the estimate for frame $i + 1$, correction for frame $i + 1$

Induction step: State Prediction

Given $P(X_{i-1} | y_0, \ldots, y_{i-1})$.

\[
\text{Prediction}
\]

Prediction involves representing

\[
P(X_i | y_0, \ldots, y_{i-1})
\]

Our independence assumptions make it possible to write

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

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

Base case

$P(y | x)$ is our observation model. For example, $P(y | x)$ might be a Gaussian with mean $x$.

Firstly, we assume that we have $P(X_0)$

And, we make a measurement $y_0$

\[
P(X_0 | y_0) = \frac{P(y_0 | X_0)P(X_0)}{P(y_0)}
\]

\[
= \int P(y_0 | X_0, X_0)P(X_0)dX_0
\]

\[
\propto P(y_0 | X_0)P(X_0)
\]

How is this formulation used

1. It’s ignored. At each time instant, the state is estimated (perhaps a maximum likelihood estimate or something non-probabilistic).
2. The conditional distributions are represented by some convenient parametric form (e.g., Gaussian).
3. The PDFs are represented non-parametrically, and sampling techniques are used.

Induction step: State Correction

In prediction, we estimated the state $X_i$ given the measurements up to $i-1$. Now we get the measure at time $i$ called $y_i$

\[
\text{Correction}
\]

Correction involves obtaining a representation of

\[
P(X_i | y_0, \ldots, y_i)
\]

Our independence assumptions make it possible to write

\[
P(X_i | y_0, \ldots, y_i) = \frac{P(X_i, y_i | y_0, \ldots, y_{i-1})}{P(y_0, \ldots, y_i)}
\]

\[
= \frac{P(y_i | X_i)P(X_i | y_0, \ldots, y_{i-1})}{P(y_0, \ldots, y_i)}
\]

\[
\propto P(y_i | X_i)P(X_i | y_0, \ldots, y_{i-1})
\]

Linear dynamic models

- Use notation $\sim$ to mean “has the pdf of,” $N(a, B)$ is a normal distribution with mean $a$ and covariance $B$.
- A linear dynamic model has the form

\[
x_i \sim N(D_i+1 x_{i-1}, \Sigma_i)
\]

\[
y_i \sim N(Mx_i, \Sigma_m)
\]
Examples

- Points moving with constant velocity
- Points moving with constant acceleration
- Periodic motion
- Etc.

Points moving with constant velocity

- We have
  \[ u_i = u_{i-1} + \Delta v_{i-1} + \epsilon_i \]
  Position
  \[ v_i = v_{i-1} + \xi_i \]
  Velocity
  – (the Greek letters denote noise terms)
- Stack \((u, v)\) into a single state vector
  \[
  \begin{bmatrix}
  u \\
  v \\
  a
  \end{bmatrix}
  = \begin{bmatrix}
  1 \\
  0 \\
  0
  \end{bmatrix}
  \begin{bmatrix}
  u_i \\
  v_i \\
  a_i
  \end{bmatrix}
  + \text{noise}
  \]
  which is the form we had above

Points moving with constant acceleration

- We have
  \[ u_i = u_{i-1} + \Delta v_{i-1} + \epsilon_i \]
  \[ v_i = v_{i-1} + \Delta a_{i-1} + \xi_i \]
  \[ a_i = a_{i-1} + \epsilon_i \]
  – (the Greek letters denote noise terms)
- Stack \((u, v)\) into a single state vector
  \[
  \begin{bmatrix}
  u \\
  v \\
  a
  \end{bmatrix}
  = \begin{bmatrix}
  1 \\
  0 \\
  0
  \end{bmatrix}
  \begin{bmatrix}
  u_i \\
  v_i \\
  a_i
  \end{bmatrix}
  + \text{noise}
  \]
  which is the form we had above

The Kalman Filter

- Key ideas:
  – Linear models interact uniquely well with Gaussian noise
    - Make the prior Gaussian, everything else Gaussian
      and the calculations are easy
    – Gaussians are really easy to represent
      - mean vector
      - covariance matrix

The Kalman Filter in 1D

- Dynamic Model
  \[ x_{i|i} \sim N(m_{x_{i|i}}, \sigma_{x_{i|i}}^2) \]
  \[ y_{i} \sim N(m_{y_{i}}, \sigma_{y_{i}}^2) \]
- Notation
  - mean of \( P(X_i|x_{i-1}, \ldots, y_{i-1}) \) as \( \overline{X}_i \)
  - Predicted mean
  - mean of \( P(X_i|x_{i}, \ldots, y_{i}) \) as \( \overline{X}_{i|i} \)
  - Corrected mean
  - the standard deviation of \( P(X_i|x_{i}, \ldots, y_{i}) \) as \( \sigma_i \)
  - Predicted variance

Prediction for 1-D Kalman filter

- The new state is obtained by
  – multiplying old state by known constant
  – adding zero-mean noise
- Therefore, predicted mean for new state is
  – constant times mean of old state
- Predicted variance is
  – sum of constant^2 times old state variance and noise variance

Because:
  - Old state is normal random variable,
  - Multiplying normal random variable by constant implies
    - mean is multiplied by a constant
    - variance is multiplied by square of constant
  - Adding zero mean noise adds zero to the mean,
  - Adding random variables adds variance
Correction for 1D Kalman Filter

- Notice:
  - if measurement noise is small, then we rely mainly on the measurement
  - if measurement noise is large, then we rely mainly on the prediction

\[
x_t = \begin{pmatrix} x_t^1 - m_u^y \sigma_t^1 \sqrt{\sigma_{x_t}^2 + m_u^y \sigma_t^1} \\ \sigma_t^1 \end{pmatrix}
\]

Multivariate Kalman Filter

Another Approach: Measurement Generation

Sample from \( p(X) \)
Evaluate \( p(I, X) \) at samples

Keep high-scoring samples
Ascend gradient & pick exemplar

Tracking Modalities

- Color
  - Histogram [Birchfield 1998; Bradski 1998]
  - Volume [Wren et al., 1995; Hoger, 1997; Durall, 1998]
- Shape
  - Deformable curve [Kan & Milzer 1998]
  - Template [Blake et al. 1993; Birchfield 1998]
  - Example-based [Costa et al., 1995; Buenberg & Hagg, 1994]
- Appearance
  - Correlation [Lucas & Kanade, 1981; Shi & Tomasi, 1996]
  - Photometric variation [Hager & Belforte, 1990]
  - Outliers [Black et al., 1998; Hager & Belforte, 1990]
  - Nonrigidity [Black et al., 1998; Schclar & Isidori, 1998]
- Motion
  - Background model [Wren et al., 1995; Rosales & Schclar, 1999; Stauffer & Grimson, 1999]
  - Optical flow [Cutler & Turk]
  - Egomotion [Kovacic & Ayar, 1996; Ivan & Aman, 1998]
- Stereo
  - Blob correlation [Auboustayani & Pentland, 1996]
  - Disparity map [Kanade et al., 1996; Konole, 1997; Durrill et al., 1998]

Color Blob tracking

- Color-based tracker gets lost on white knight: Same Color
Snakes: Active Contours

- Contour $C$: continuous curve on smooth surface in $\mathbb{R}^3$
- Snake $S$: projection of $C$ to image
- Curve types
  - Edge between regions on surface with contrasting properties
  - Line that contrasts with surface properties on both side
  - Silhouette of surface against contrasting background
- General Algorithm:
  - Perform edge detection
  - Fit parametric or non-parametric curve to data

Snakes: Basic Approach

- Parameterize a closed contour
  \[ Q = (x_0^s, x_1^s, \ldots, x_n^s) \]
- \[ r(s) = q'B(s) \quad \text{or} \quad r(s) = U(s)Q \]
- Given a predicted state $q$, search radially for edges
- Solve a least squares problem for new state

Tracker Composition: Only Shape (Snakes)

- Geometry-based tracker gets lost on black pawn: Same shape

Tracker Composition

Tracker Composition: Color and Shape

- Combining Trackers $\Rightarrow$ Robustness

Visual Tracking using regions

- Variability model: $l_t = g(l_{t-1}, p_t)$
- Incremental Estimation: From $l_0$, $l_{t-1}$, and $p_t$ compute $\Delta p_{t+1}$
- \[ \| l_0 - g(l_{t+1}, p_{t+1}) \|^2 \Rightarrow \text{min} \]
Tracking using Textured Regions

- Mean intensity difference between \( I \) and affine warp of template image [Shi & Tomasi, 1994]

\[
\Psi_{\text{spatial}}(x, y) = \sum_{(x, y) \in B} \left( I(x, y) - I_t(x, y) \right)^2
\]

| Template \( I_t \) | Tracked state \( I_c \) |

Template tracking: Planar Case

Planar Object => Affine motion model: \( \mathbf{v}_t = A \mathbf{u}_0 + \mathbf{d} \)

\[
l_t = g(p_t, l_0)
\]

Hager/Toyama: Tracking Cycle

- Prediction
  - Prior states predict new appearance
- Image warping
  - Generate a “normalized view”
- Model inverse
  - Compute error from nominal
- State integration
  - Apply correction to state

XVision: A tracking System

Composition of Primitive Trackers

Next Lectures

- Recognition, detection, and classification
- Reading:
  - Chapter 15: Learning to Classify
  - Chapter 16: Classifying Images
  - Chapter 17: Detecting Objects in Images