

Visual Tracking

Introduction to Computer Vision
CSE 152
Lecture 13

CSE 152, Spring 2017

Introduction to Computer Vision

Announcements

- Homework 4 is due May 24, 11:59 PM
- Reading:
 - Chapter 11: Tracking

CSE 152, Spring 2017

Introduction to Computer Vision

Visual Tracking



Main Challenges

1. 3D pose variation
2. Target occlusion
3. Illumination variation
4. Camera jitter
5. Expression variation etc.

[Ho, Lee, Kriegman]

CSE 152, Spring 2017

Introduction to Computer Vision

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 $t = 0$?

CSE 152, Spring 2017

Introduction to Computer Vision

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 $(\phi, \dot{\phi})$

CSE 152, Spring 2017

Introduction to Computer Vision

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?



CSE 152, Spring 2017

Introduction to Computer Vision

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

CSE 152, Spring 2017

Introduction to Computer Vision

Blob Tracking in IR Images



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

CSE 152, Spring 2017

Introduction to Computer Vision

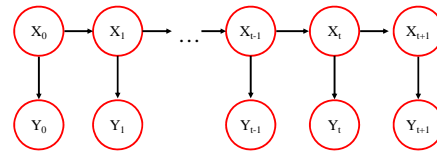
Tracking: Probabilistic framework

- 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}

CSE 152, Spring 2017

Introduction to Computer Vision

Tracking State



- Instead of “knowing state” at each instant, we treat the state as random variables X , characterized by a pdf $P(X_t)$ or perhaps conditioned on other random variables, e.g., $P(X_t | X_{t-1})$ etc.
- 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

CSE 152, Spring 2017

Introduction to Computer Vision

Three main steps



- **Prediction:** we have seen y_0, \dots, y_{i-1} — what state does this set of measurements predict for the i 'th frame? to solve this problem, we need to obtain a representation of $P(X_i | Y_0 = y_0, \dots, Y_{i-1} = y_{i-1})$.
- **Data association:** Some of the measurements obtained from the i -th frame may tell us about the object's state. Typically, we use $P(X_i | Y_0 = y_0, \dots, Y_{i-1} = y_{i-1})$ to identify these measurements.
- **Correction:** now that we have y_i — the relevant measurements — we need to compute a representation of $P(X_i | Y_0 = y_0, \dots, Y_i = y_i)$.

We can try to express these conditional distributions parametrically, sample the distribution, or estimate the mode.

CSE 152, Spring 2017

Introduction to Computer Vision

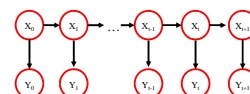
Simplifying Assumptions

- **Only the immediate past matters:** formally, we require

$$P(X_i | X_1, \dots, X_{i-1}) = P(X_i | X_{i-1})$$

- **Measurements depend only on the current state:** we assume that Y_i is conditionally independent of all other measurements given X_i . This means that

$$P(Y_i | Y_1, \dots, Y_{i-1}, X_i) = P(Y_i | X_i)$$



CSE 152, Spring 2017

Introduction to Computer Vision

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 prediction the estimate for frame $i + 1$, correction for frame $i + 1$

CSE 152, Spring 2017

Introduction to Computer Vision

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 $P(\mathbf{X}_0)$ ← Prior distribution of initial state

And, we make a measurement \mathbf{y}_0

$$\begin{aligned} P(\mathbf{X}_0 | \mathbf{Y}_0 = \mathbf{y}_0) &= \frac{P(\mathbf{y}_0 | \mathbf{X}_0)P(\mathbf{X}_0)}{P(\mathbf{y}_0)} \\ &= \frac{P(\mathbf{y}_0 | \mathbf{X}_0)P(\mathbf{X}_0)}{\int P(\mathbf{y}_0 | \mathbf{X}_0)P(\mathbf{X}_0)d\mathbf{X}_0} \\ &\propto P(\mathbf{y}_0 | \mathbf{X}_0)P(\mathbf{X}_0) \end{aligned}$$

CSE 152, Spring 2017

to Computer Vision

Induction step: State Prediction

Given $P(\mathbf{X}_{i-1} | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})$.

Prediction

Prediction involves representing

$$P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})$$

Our independence assumptions make it possible to write

$$\begin{aligned} P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_{i-1}) &= \int P(\mathbf{X}_i, \mathbf{X}_{i-1} | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})d\mathbf{X}_{i-1} \\ &= \int P(\mathbf{X}_i | \mathbf{X}_{i-1}, \mathbf{y}_0, \dots, \mathbf{y}_{i-1})P(\mathbf{X}_{i-1} | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})d\mathbf{X}_{i-1} \\ &= \int P(\mathbf{X}_i | \mathbf{X}_{i-1})P(\mathbf{X}_{i-1} | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})d\mathbf{X}_{i-1} \end{aligned}$$

CSE 152, Spring 2017

Introduction to Computer Vision

Induction step: State Correction

In prediction, we estimated the state \mathbf{X}_i given the measurements up to $i-1$.
Now we get the measure at time i called \mathbf{y}_i .

Correction

Correction involves obtaining a representation of

$$P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_i)$$

Our independence assumptions make it possible to write

$$\begin{aligned} P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_i) &= \frac{P(\mathbf{X}_i, \mathbf{y}_0, \dots, \mathbf{y}_i)}{P(\mathbf{y}_0, \dots, \mathbf{y}_i)} \\ &= \frac{P(\mathbf{y}_i | \mathbf{X}_i, \mathbf{y}_0, \dots, \mathbf{y}_{i-1})P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})P(\mathbf{y}_0, \dots, \mathbf{y}_{i-1})}{P(\mathbf{y}_0, \dots, \mathbf{y}_i)} \\ &= P(\mathbf{y}_i | \mathbf{X}_i)P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_{i-1}) \frac{P(\mathbf{y}_0, \dots, \mathbf{y}_{i-1})}{P(\mathbf{y}_0, \dots, \mathbf{y}_i)} \\ &= \frac{P(\mathbf{y}_i | \mathbf{X}_i)P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})}{\int P(\mathbf{y}_i | \mathbf{X}_i)P(\mathbf{X}_i | \mathbf{y}_0, \dots, \mathbf{y}_{i-1})d\mathbf{X}_i} \end{aligned}$$

CSE 152, Spring 2017

Introduction to Computer Vision

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.

CSE 152, Spring 2017

Introduction to Computer Vision

Linear dynamic models

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

$$\begin{aligned} \mathbf{x}_i &= N(\mathbf{D}_{i-1}\mathbf{x}_{i-1}; \Sigma_{d_i}) \\ \mathbf{y}_i &= N(\mathbf{M}_i\mathbf{x}_i; \Sigma_{m_i}) \end{aligned}$$

CSE 152, Spring 2017

Introduction to Computer Vision

Examples

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

CSE 152, Spring 2017

Introduction to Computer Vision

Points moving with constant velocity

- We have

$$\begin{aligned} u_i &= u_{i-1} + \Delta t v_{i-1} + \varepsilon_i && \text{Position} \\ v_i &= v_{i-1} + \zeta_i && \text{Velocity} \end{aligned}$$

– (the Greek letters denote noise terms)

- Stack (u, v) into a single state vector

$$\begin{pmatrix} u \\ v \end{pmatrix}_i = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_{i-1} + \text{noise}$$

which is the form we had above

CSE 152, Spring 2017

Introduction to Computer Vision

Points moving with constant acceleration

- We have

$$\begin{aligned} u_i &= u_{i-1} + \Delta t v_{i-1} + \varepsilon_i \\ v_i &= v_{i-1} + \Delta t a_{i-1} + \zeta_i \\ a_i &= a_{i-1} + \xi_i \end{aligned}$$

– (the Greek letters denote noise terms)

- Stack (u, v) into a single state vector

$$\begin{pmatrix} u \\ v \\ a \end{pmatrix}_i = \begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ a \end{pmatrix}_{i-1} + \text{noise}$$

which is the form we had above

CSE 152, Spring 2017

Introduction to Computer Vision

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

CSE 152, Spring 2017

Introduction to Computer Vision

The Kalman Filter in 1D

- Dynamic Model

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

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

- Notation

mean of $P(X_i | y_0, \dots, y_{i-1})$ as \bar{X}_i^- — Predicted mean
 Corrected mean — mean of $P(X_i | y_0, \dots, y_i)$ as \bar{X}_i^+
 the standard deviation of $P(X_i | y_0, \dots, y_{i-1})$ as σ_i^-
 of $P(X_i | y_0, \dots, y_i)$ as σ_i^+

CSE 152, Spring 2017

Introduction to Computer Vision

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 constant² 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

CSE 152, Spring 2017

Introduction to Computer Vision

Dynamic Model:

$$x_i \sim N(d_i x_{i-1}, \sigma_d)$$

$$y_i \sim N(m_i x_i, \sigma_{m_i})$$

Start Assumptions: \bar{x}_0 and σ_0^2 are known
Update Equations: Prediction

$$\bar{x}_i = d_i \bar{x}_{i-1}$$

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

Update Equations: Correction

$$x_i^+ = \left(\frac{\bar{x}_i \sigma_{m_i}^2 + m_i y_i (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right)$$

$$\sigma_i^+ = \sqrt{\frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{(\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2)}}$$

CSE 152, Spring 2017 Introduction to Computer Vision

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

$$x_i^+ = \left(\frac{\bar{x}_i \sigma_{m_i}^2 + m_i y_i (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right)$$

$$\sigma_i^+ = \sqrt{\frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{(\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2)}}$$

CSE 152, Spring 2017 Introduction to Computer Vision

Multivariate Kalman Filter

Dynamic Model:

$$x_i \sim N(\mathcal{D}_i x_{i-1}, \Sigma_d)$$

$$y_i \sim N(\mathcal{M}_i x_i, \Sigma_{m_i})$$

Start Assumptions: \bar{x}_0 and Σ_0 are known
Update Equations: Prediction

$$\bar{x}_i = \mathcal{D}_i \bar{x}_{i-1}$$

$$\Sigma_i^- = \Sigma_d + \mathcal{D}_i \Sigma_{i-1}^- \mathcal{D}_i$$

Update Equations: Correction

$$K_i = \Sigma_i^- \mathcal{M}_i^T (\mathcal{M}_i \Sigma_i^- \mathcal{M}_i^T + \Sigma_{m_i})^{-1}$$

$$\bar{x}_i^+ = \bar{x}_i + K_i (y_i - \mathcal{M}_i \bar{x}_i)$$

$$\Sigma_i^+ = [I - K_i \mathcal{M}_i] \Sigma_i^-$$

CSE 152, Spring 2017 Introduction to Computer Vision

Another Approach: Measurement Generation

Sample from $p(\mathbf{X})$ Evaluate $p(\mathbf{I} | \mathbf{X})$ at samples

Keep high-scoring samples Ascend gradient & pick exemplars

CSE 152, Spring 2017 Introduction to Computer Vision

Tracking Modalities

(Define the features (observations, measurements) Y_i)

- Color
 - Histogram [Birchfield 1998; Bradski 1998]
 - Volume [Wren *et al.*, 1995; Bregler, 1997; Darrell, 1998]
- Shape
 - Deformable curve [Kass *et al.* 1988]
 - Template [Blake *et al.* 1993; Birchfield 1998]
 - Example-based [Cootes *et al.*, 1993; Baumberg & Hogg, 1994]
- Appearance
 - Correlation [Lucas & Kanade, 1981; Shi & Tomasi, 1994]
 - Photometric variation [Hager & Belhumeur, 1998]
 - Outliers [Black *et al.*, 1998; Hager & Belhumeur, 1998]
 - Nonrigidity [Black *et al.*, 1998; Sclaroff & Isidoro, 1998]
- Motion
 - Background model [Wren *et al.*, 1995; Rosales & Sclaroff, 1999; Stauffer & Grimson, 1999]
 - Optical flow [Cutler & Turk]
 - Egomotion [Sawhney & Ayer, 1996; Irani & Anandan, 1998]
- Stereo
 - Blob correlation [Azarbayejani & Pentland, 1996]
 - Disparity map [Kanade *et al.*, 1996; Konolige, 1997; Darrell *et al.*, 1998]

CSE 152, Spring 2017 Introduction to Computer Vision

Color Blob tracking

- Color-based tracker gets lost on white knight: Same Color

CSE 152, Spring 2017 Introduction to Computer Vision

Snakes: Active Contours

- Contour C : continuous curve on smooth surface in \mathcal{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

CSE 152, Spring 2017

Introduction to Computer Vision

Snakes: Basic Approach

- Parameterize a closed contour
- $\mathbf{r}(s) = \mathbf{q}'\mathbf{B}(s)$ or $\mathbf{r}(s) = \mathbf{U}(s)\mathbf{Q}$
- Given a predicted state \mathbf{q} , search radially for edges
- Solve a least squares problem for new state

$$\mathbf{Q} = (q_0^x \quad q_n^x \quad q_0^y \quad q_n^y)$$

$$\mathbf{U}(s) = \begin{pmatrix} \mathbf{B}(s)^T & 0 \\ 0 & \mathbf{B}(s)^T \end{pmatrix}$$



CSE 152, Spring 2017

Introduction to Computer Vision

Tracker Composition: Only Shape (Snakes)

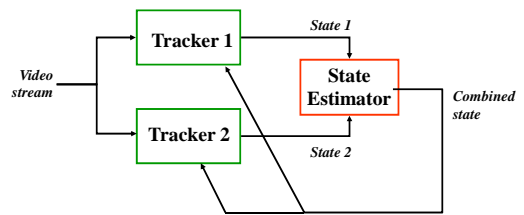


- Geometry-based tracker gets lost on black pawn: Same shape

CSE 152, Spring 2017

Introduction to Computer Vision

Tracker Composition



CSE 152, Spring 2017

Introduction to Computer Vision

Tracker Composition: Color and Shape

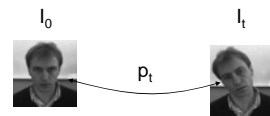


- Combining Trackers => Robustness

CSE 152, Spring 2017

Introduction to Computer Vision

Visual Tracking using regions



Variability model: $I_t = g(I_0, p_t)$

Incremental Estimation: From I_0 , I_{t+1} and p_t compute Δp_{t+1}

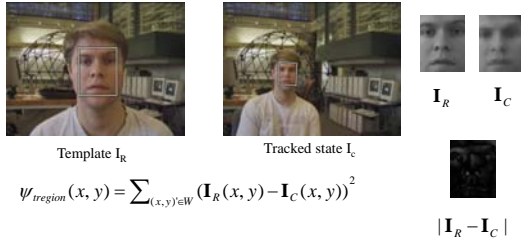
$$\| I_0 - g(I_{t+1}, p_{t+1}) \|^2 \implies \min$$

CSE 152, Spring 2017

Introduction to Computer Vision

Tracking using Textured Regions

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

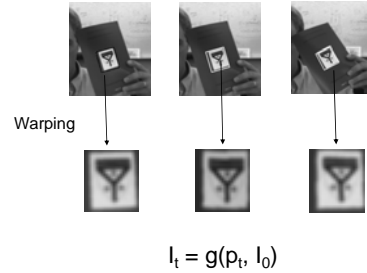


CSE 152, Spring 2017

Introduction to Computer Vision

Template tracking: Planar Case

Planar Object \Rightarrow Affine motion model: $u_i = A u_i + d$

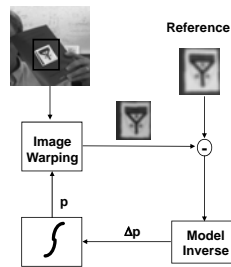


CSE 152, Spring 2017

Introduction to Computer Vision

Hager/Toyama: Tracking Cycle

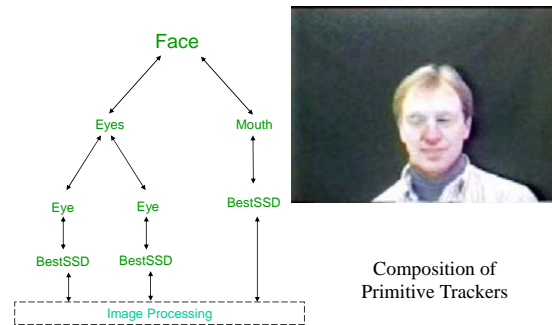
- Prediction
 - Prior states predict new appearance
- Image warping
 - Generate a “normalized view”
- Model inverse
 - Compute error from nominal
- State integration
 - Apply correction to state



CSE 152, Spring 2017

Introduction to Computer Vision

XVision: A tracking System



CSE 152, Spring 2017

Introduction to Computer Vision

Next Lectures

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

CSE 152, Spring 2017

Introduction to Computer Vision