Tracking

Many slides adapted from Kristen Grauman, Deva Ramanan
Feature tracking

- So far, we have only considered optical flow estimation in a pair of images
- If we have more than two images, we can compute the optical flow from each frame to the next
- Given a point in the first image, we can in principle reconstruct its path by simply “following the arrows”
Tracking challenges

- Ambiguity of optical flow
  - Find good features to track

- Large motions
  - Discrete search instead of Lucas-Kanade

- Changes in shape, orientation, color
  - Allow some matching flexibility

- Occlusions, disocclusions
  - Need mechanism for deleting, adding new features

- Drift – errors may accumulate over time
  - Need to know when to terminate a track
Handling large displacements

- Define a small area around a pixel as the template
- Match the template against each pixel within a search area in next image – just like stereo matching!
- Use a match measure such as SSD or correlation
- After finding the best discrete location, can use Lucas-Kanade to get sub-pixel estimate
Tracking over many frames

- Select features in first frame
- For each frame:
  - Update positions of tracked features
    - Discrete search or Lucas-Kanade
  - Terminate inconsistent tracks
    - Compute similarity with corresponding feature in the previous frame or in the first frame where it’s visible
  - Start new tracks if needed
Shi-Tomasi feature tracker

- Find good features using eigenvalues of second-moment matrix
  - Key idea: “good” features to track are the ones that can be tracked reliably

- From frame to frame, track with Lucas-Kanade and a pure *translation* model
  - More robust for small displacements, can be estimated from smaller neighborhoods

- Check consistency of tracks by *affine* registration to the first observed instance of the feature
  - Affine model is more accurate for larger displacements
  - Comparing to the first frame helps to minimize drift

Tracking example

Figure 1: Three frame details from Woody Allen’s *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

Tracking with dynamics

• Key idea: Given a model of expected motion, predict where objects will occur in next frame, even before seeing the image
  • Restrict search for the object
  • Improved estimates since measurement noise is reduced by trajectory smoothness
General model for tracking

• The moving object of interest is characterized by an underlying state $X$
• State $X$ gives rise to measurements or observations $Y$
• At each time $t$, the state changes to $X_t$ and we get a new observation $Y_t$
Steps of tracking

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

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

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

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

• Tracking can be seen as the process of propagating the posterior distribution of state given measurements across time
Simplifying assumptions

• Only the immediate past matters

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

dynamics model
Simplifying assumptions

- Only the immediate past matters

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

**dynamics model**

- Measurements depend only on the current state

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

**observation model**
Simplifying assumptions

• Only the immediate past matters

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

dynamics model

• Measurements depend only on the current state

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

observation model

\[ X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_t \]

\[ Y_1 \rightarrow Y_2 \rightarrow \cdots \rightarrow Y_t \]
**Tracking as induction**

- **Base case:**
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$

- **Given corrected estimate for frame $t$:**
  - Predict for frame $t+1$
  - Correct for frame $t+1$
Tracking as induction

- **Base case:**
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0 = y_0$

\[
P(X_0 \mid Y_0 = y_0) = \frac{P(y_0 \mid X_0)P(X_0)}{P(y_0)} \propto P(y_0 \mid X_0)P(X_0)
\]
Induction step: Prediction

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

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

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

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

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

Independence assumption
Induction step: Correction

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

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

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

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

Independence assumption
(observation $y_t$ depends only on state $X_t$)
Induction step: Correction

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

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

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

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

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

Conditioning on $X_t$
Summary: Prediction and correction

• Prediction:

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

- dynamics model
- corrected estimate from previous step
Summary: Prediction and correction

• Prediction:

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

- dynamics model
- corrected estimate from previous step

• Correction:

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

- observation model
- predicted estimate
Linear Dynamic Models

- Dynamics model: state undergoes linear transformation plus Gaussian noise

\[ X_t \sim N(D_t x_{t-1}, \Sigma_{d_t}) \]

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

\[ Y_t \sim N(M_t x_t, \Sigma_{m_t}) \]
Example: Constant velocity (1D)

- State vector is position and velocity

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

\[
v_t = v_{t-1} + \xi
\]

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

- Measurement is position only

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

(greek letters denote noise terms)
Example: Constant acceleration (1D)

- State vector is position, velocity, and acceleration
  \[
  x_t = \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix}
  \]
  \[
  p_t = p_{t-1} + (\Delta t)v_{t-1} + \epsilon
  \]
  \[
  v_t = v_{t-1} + (\Delta t)a_{t-1} + \xi
  \]
  \[
  a_t = a_{t-1} + \zeta
  \]

- Measurement is position only
  \[
  y_t = Mx_t + noise = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} + noise
  \]
The Kalman filter

• Method for tracking linear dynamical models in Gaussian noise

• The predicted/corrected state distributions are Gaussian
  • You only need to maintain the mean and covariance
  • The calculations are easy (all the integrals can be done in closed form)
Propagation of Gaussian densities
The Kalman Filter: 1D state

Make measurement

Given corrected state from previous time step and all the measurements up to the current one, predict the distribution over the current step

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

Mean and std. dev. of predicted state:

\[
\mu_t^-, \sigma_t^-
\]

Predict

Time advances (from \(t-1\) to \(t\))

Correct

Given prediction of state and current measurement, update prediction of state

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

Mean and std. dev. of corrected state:

\[
\mu_t^+, \sigma_t^+
\]
1D Kalman filter: Prediction

- Linear dynamic model defines predicted state evolution, with noise
  \[ X_t \sim N(dx_{t-1}, \sigma_d^2) \]
- Want to estimate distribution for next predicted state
  \[
  P(X_t \mid y_0, \ldots, y_{t-1}) = \int P(X_t \mid X_{t-1})P(X_{t-1} \mid y_0, \ldots, y_{t-1})dX_{t-1}
  \]
1D Kalman filter: Prediction

- Linear dynamic model defines predicted state evolution, with noise
  \[ X_t \sim N(dx_{t-1}, \sigma_d^2) \]

- Want to estimate distribution for next predicted state
  \[ P(X_t | y_0, \ldots, y_{t-1}) = N(\mu_t^-, (\sigma_t^-)^2) \]
1D Kalman filter: Prediction

- Linear dynamic model defines predicted state evolution, with noise
  \[ X_t \sim N(dx_{t-1}, \sigma_d^2) \]
- Want to estimate distribution for next predicted state
  \[ P(X_t | y_0, \ldots, y_{t-1}) = N(\mu_t^-, (\sigma_t^-)^2) \]
- 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

- Mapping of state to measurements: \( Y_t \sim N(mx_t, \sigma_m^2) \)

- Predicted state: \( P(X_t | y_0, \ldots, y_{t-1}) = N(\mu_t^-, (\sigma_t^-)^2) \)

- Want to estimate corrected distribution

\[
P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})}{\int P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})dX_t}
\]
1D Kalman filter: Correction

- Mapping of state to measurements:  \( Y_t \sim N(mx_t, \sigma_m^2) \)
- Predicted state:  \( P(X_t|y_0,\ldots, y_{t-1}) = N(\mu^-, (\sigma^-)^2) \)
- Want to estimate corrected distribution
  \[
P(X_t|y_0,\ldots, y_t) = N(\mu^+, (\sigma^+)^2)
  \]
1D Kalman filter: Correction

- Mapping of state to measurements: $Y_t \sim N(mx_t, \sigma^2_m)$
- Predicted state: $P(X_t|y_0, \ldots, y_{t-1}) = N(\mu^-_t, (\sigma^-_t)^2)$
- Want to estimate corrected distribution
  $$P(X_t|y_0, \ldots, y_t) = N(\mu^+_t, (\sigma^+_t)^2)$$
- Update the mean:
  $$\mu^+_t = \frac{\mu^-_t \sigma^2_m + my_t (\sigma^-_t)^2}{\sigma^2_m + m^2 (\sigma^-_t)^2}$$
- Update the variance:
  $$(\sigma^+_t)^2 = \frac{\sigma^2_m (\sigma^-_t)^2}{\sigma^2_m + m^2 (\sigma^-_t)^2}$$
Prediction vs. correction

\[ \mu_t^+ = \frac{\mu_t^- \sigma_m^2 + my_t (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2} \quad (\sigma_t^+)^2 = \frac{\sigma_m^2 (\sigma_t^-)^2}{\sigma_m^2 + m^2 (\sigma_t^-)^2} \]

- What if there is no prediction uncertainty \((\sigma_t^- = 0)\)?
  \[ \mu_t^+ = \mu_t^- \quad (\sigma_t^+)^2 = 0 \]
  The measurement is ignored!

- What if there is no measurement uncertainty \((\sigma_m = 0)\)?
  \[ \mu_t^+ = \frac{y_t}{m} \quad (\sigma_t^+)^2 = 0 \]
  The prediction is ignored!
The Kalman filter: General case

**PREDICT**

\[
x_t^- = D_t x_{t-1}^+
\]

\[
\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_d_t
\]

**CORRECT**

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

\[
x_t^+ = x_t^- + K_t \left( y_t - M_t x_t^- \right)
\]

\[
\Sigma_t^+ = (I - K_t M_t) \Sigma_t^-
\]
Kalman filter pros and cons

• **Pros**
  • Simple updates, compact and efficient

• **Cons**
  • Unimodal distribution, only single hypothesis
  • Restricted class of motions defined by linear model
Propagation of general densities
Factored sampling

- Represent the state distribution non-parametrically
  - Prediction: Sample points from prior density for the state, $P(X)$
  - Correction: Weight the samples according to $P(Y|X)$

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

M. Isard and A. Blake, CONDENSATION -- conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998
Particle filtering

- We want to use sampling to propagate densities over time (i.e., across frames in a video sequence)
- At each time step, represent posterior $P(X_t|Y_t)$ with weighted sample set
- Previous time step’s sample set $P(X_{t-1}|Y_{t-1})$ is passed to next time step as the effective prior

M. Isard and A. Blake, CONDENSATION -- conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998
Particle filtering

Start with weighted samples from previous time step

Sample and shift according to dynamics model

Spread due to randomness; this is predicted density $P(X_t | Y_{t-1})$

Weight the samples according to observation density

Arrive at corrected density estimate $P(X_t | Y_t)$

M. Isard and A. Blake, CONDENSATION -- conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998
Particle filtering results

http://www.robots.ox.ac.uk/~misard/condensation.html
Tracking issues

• Initialization
  • Manual
  • Background subtraction
  • Detection
Tracking issues

• Initialization

• Obtaining observation and dynamics model
  • Generative observation model: “render” the state on top of the image and compare
  • Discriminative observation model: classifier or detector score
  • Dynamics model: learn (very difficult) or specify using domain knowledge
Tracking issues

• Initialization
• Obtaining observation and dynamics model
• Prediction vs. correction
  • If the dynamics model is too strong, will end up ignoring the data
  • If the observation model is too strong, tracking is reduced to repeated detection
Tracking issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Data association
  - What if we don’t know which measurements to associate with which tracks?
Tracking issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Data association
- Drift
  - Errors caused by dynamical model, observation model, and data association tend to accumulate over time
Drift

Data association

- So far, we’ve assumed the entire measurement to be relevant to determining the state.
- In reality, there may be uninformative measurements (clutter) or measurements may belong to different tracked objects.
- **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…
Data association

- Simple strategy: only pay attention to the measurement that is “closest” to the prediction
- More sophisticated strategy: keep track of multiple state/observation hypotheses
  - Can be done with particle filtering
- This is a general problem in computer vision, there is no easy solution
Recall: Generative part-based models

\[ P(\text{image} \mid \text{object}) = P(\text{appearance}, \text{shape} \mid \text{object}) \]
Recall: Generative part-based models

\[ P(\text{image} \mid \text{object}) = P(\text{appearance}, \text{shape} \mid \text{object}) = \max_h P(\text{appearance} \mid h, \text{object}) p(\text{shape} \mid h, \text{object}) p(h \mid \text{object}) \]

\( h \): assignment of features to parts
Tracking people by learning their appearance

• Person model = appearance + structure (+ dynamics)
• Structure and dynamics are generic, appearance is person-specific
• Trying to acquire an appearance model “on the fly” can lead to drift
• Instead, can use the whole sequence to initialize the appearance model and then keep it fixed while tracking
• Given strong structure and appearance models, tracking can essentially be done by repeated detection (with some smoothing)

Tracking people by learning their appearance

Pictorial structure model

Fischler and Elschlager (73), Felzenszwalb and Huttenlocher (00)

\[
\Pr(P_{\text{tor}}, P_{\text{arm}}, \ldots | \text{Im}) \propto \prod_{i,j} \Pr(P_i | P_j) \prod_i \Pr(\text{Im}(P_i))
\]

part geometry

part appearance
Bottom-up initialization: Clustering

Top-down initialization: Exploit “easy” poses

Example results