Tracking

• Establish where an object is, other aspects of state, using time sequence
  • Biggest problem -- Data Association
• Key ideas
  • Tracking by detection
  • Tracking through flow
Track by detection (simple form)

- **Assume**
  - a very reliable detector (e.g. faces; back of heads)
  - detections that are well spaced in images (or have distinctive properties)
    - e.g. news anchors; heads in public

- **Link detects across time**
  - only one - easy
  - multiple - weighted bipartite matching
Matching

- Established problem
  - Use Hungarian algorithm
  - or nearest neighbours

![Diagram of matching process]
Point tracks reveal curious phenomena in public spaces

Yan+Forsyth, 04
Tracks

- Some detections might fail
- Build "tracks"
  - detect in each frame
  - link detects to tracks using matching algorithm
    - measurements with no track? create new track
    - tracks with no measurement? wait, then reap
  - (perhaps) join tracks over time with global considerations
- What happens if the objects move?
Example: SFM

- We need to fill in a data matrix
- Strategy
  - find points in one frame
  - link each to corresponding point in next frame; etc.
- Cues for linking
  - patches
    - “look the same”
    - “don’t move much”
Matching

- Patch is at \( u, t \); moves to \( u+h, t+1 \); \( h \) is small
- Error is sum of squared differences
  \[ E(h) = \sum_{u \in P_t} [I(u, t) - I(u+h, t+1)]^2 \]
  - This is minimized when
    \[ \nabla_h E(h) = 0. \]
- Substitute
  \[ I(u+h, t+1) \approx I(u, t) + h^T \nabla I \]
- Get
  \[ \left[ \sum_{u \in P_t} (\nabla I)(\nabla I)^T \right] h = \sum_{u \in P_t} [I(u, t) - I(u, t+1)] \nabla I \]
Matching

- We can tell if the match is good by looking at

\[ \sum_{u \in P_t} (\nabla I)(\nabla I)^T \]

- which will be poorly conditioned if matching is poor
  - eg featureless region
  - eg flow region
Matching

- Match must work from $i$ to $i+1$
  - Method is OK so far for this
  - what about 1 to 100?
- Second test; compare with first frame, by minimizing, testing

\[ E(\mathcal{M}, c) = \sum_{u \in \mathcal{P}_1} [I(u, 1) - I(\mathcal{M}u + c, t)]^2. \]
Image frame, from a sequence (Shi Tomasi 94)

Strongly textured points (Shi Tomasi 94)
Point patches in tracks (Shi Tomasi 94)
Dissimilarity (Shi Tomasi 94)
Efros et al, 03
Efros et al, 03
What if the pixels get mixed up?

- Describe with histograms
- Match with procedure called “mean shift” (chapter)
Track by flow (simple form)

- **Assume**
  - appearance unknown (but domain, parametric flow model known)
  - optic flow assumptions, as before
- **Initialize**
  - mark out domain
- **Track**
  - choose flow model parameters that align domain in pic n with n+1 best
  - push domain through flow model

\[ \sum_{x \in D_t} [I(x, t) - I(x + V(x, \theta), t)]^2 \]
Figure from Ju, Black and Yacoob, “Cardboard people”
Figure from Ju, Black and Yacoob, “Cardboard people”
\[ \mathbf{a} = (0, 1, 0, 0, 0, 1, 0, 0) \quad \mathbf{a} = (0, 1, 0, 0, 0, -1, 0, 0). \]

\[(u(x), v(x)^T) = (a_0 + a_1x + a_2y + a_6x^2 + a_7xy, a_3 + a_4x + a_5y + a_6xy + a_7y^2)\]

Figure from Ju, Black and Yacoob, “Cardboard people”
Figure from Ju, Black and Yacoob, “Cardboard people”
Dangers

- Loss of track
  - small errors accumulate in model of appearance
  - DRIFT
- Appearance often isn’t constant
When are large motions “easy”?

- When they’re “predictable”
  - e.g. ballistic motion
  - e.g. constant velocity

- Need a theory
Tracking - more formal view

• Very general model:
  • We assume there are moving objects, which have an underlying state $X$
  • There are observations $Y$, some of which are functions of this state
  • There is a clock
    • at each tick, the state changes
    • at each tick, we get a new observation

• Examples
  • object is ball, state is 3D position+velocity, observations are stereo pairs
  • object is person, state is body configuration, observations are frames, clock is in camera (30 fps)
• Given
  • $P(X_{i-1}|Y_0, ..., Y_{i-1})$
    • “Prior”
• We should like to know
  • $P(X_i|Y_0, ..., Y_{i-1})$
    • “Predictive distribution”
  • $P(X_i|Y_0, ..., Y_i)$
    • “Posterior”
The three main issues in tracking

- **Prediction**: we have seen $y_0, \ldots, y_{i-1}$ — what state does this set of measurements predict for the $i$’th frame? to solve this problem, we need to obtain a representation of $P(X_i|Y_0 = y_0, \ldots, Y_{i-1} = y_{i-1})$.

- **Data association**: Some of the measurements obtained from the $i$-th frame may tell us about the object’s state. Typically, we use $P(X_i|Y_0 = y_0, \ldots, Y_{i-1} = y_{i-1})$ to identify these measurements.

- **Correction**: now that we have $y_i$ — the relevant measurements — we need to compute a representation of $P(X_i|Y_0 = y_0, \ldots, Y_i = y_i)$. 
Key assumptions:

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

\[
P(X_i | X_1, \ldots, X_{i-1}) = P(X_i | X_{i-1})
\]

This assumption hugely simplifies the design of algorithms, as we shall see; furthermore, it isn’t terribly restrictive if we’re clever about interpreting \( X_i \) as we shall show in the next section.

- **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

\[
P(Y_i, Y_j, \ldots Y_k | X_i) = P(Y_i | X_i)P(Y_j, \ldots, Y_k | X_i)
\]

Again, this isn’t a particularly restrictive or controversial assumption, but it yields important simplifications.
Tracking as Induction - base case

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

$$P(X_0 | Y_0 = y_0) = \frac{P(y_0 | X_0)P(X_0)}{P(y_0)}$$

$$= \frac{P(y_0 | X_0)P(X_0)}{\int P(y_0 | X_0)P(X_0)\,dX_0}$$

$$\propto P(y_0 | X_0)P(X_0)$$
Given

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

**Prediction**

Prediction involves representing

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

Our independence assumptions make it possible to write

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

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

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

**Correction**

Correction involves obtaining a representation of

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

Our independence assumptions make it possible to write

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

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

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

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

\[ x_i \sim N(D_i x_{i-1}; \Sigma_{d_i}) \]

\[ y_i \sim N(M_i x_i; \Sigma_{m_i}) \]
Examples

- Drifting points
  - Observability
- Points moving with constant velocity
- Points moving with constant acceleration
- Periodic motion
- Etc.
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 --- once you know the mean and covariance, you’re done
The Kalman Filter in 1D

- Dynamic Model
  \[ x_i \sim N(d_i x_{i-1}, \sigma^2_{d_i}) \]
  \[ y_i \sim N(m_i x_i, \sigma^2_{r_{ni}}) \]

- Notation
  - mean of \( P(X_i|y_0, \ldots, y_{i-1}) \) as \( \overline{X}_i \)
  - mean of \( P(X_i|y_0, \ldots, y_i) \) as \( \overline{X}_i^+ \)
  - the standard deviation of \( P(X_i|y_0, \ldots, y_{i-1}) \) as \( \sigma_i^- \)
  - of \( P(X_i|y_0, \ldots, y_i) \) as \( \sigma_i^+ \)
Dynamic Model:
\[ x_i \sim N(d_i x_{i-1}, \sigma_{d_i}) \]
\[ y_i \sim N(m_i x_i, \sigma_{m_i}) \]

Start Assumptions: \( \bar{x}_0 \) and \( \sigma_0^- \) are known

Update Equations: Prediction

\[ \bar{x}_i^- = d_i \bar{x}_{i-1}^+ \]
\[ \sigma_i^- = \sqrt{\sigma_{d_i}^2 + (d_i \sigma_{i-1}^+)^2} \]

Update Equations: Correction

\[ x_i^+ = \left( \frac{\bar{x}_i^- \sigma_{m_i}^2 + m_i y_i (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i (\sigma_i^-)^2} \right) \]
\[ \sigma_i^+ = \sqrt{\left( \frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right)} \]
Dynamic Model:

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

\[ y_i \sim N(M_i x_i, \Sigma_{m_i}) \]

Start Assumptions: \( \bar{x}_0 \) and \( \Sigma_0 \) are known

Update Equations: Prediction

\[ \bar{x}_i^- = D_i \bar{x}_{i-1}^+ \]

\[ \Sigma_i^- = \Sigma_{d_i} + D_i \sigma_{i-1}^+ D_i \]

Update Equations: Correction

\[ K_i = \Sigma_i^- M_i^T [M_i \Sigma_i^- M_i^T + \Sigma_{m_i}]^{-1} \]

\[ \bar{x}_i^+ = \bar{x}_i^- + K_i [y_i - M_i \bar{x}_i^-] \]

\[ \Sigma_i^+ = [I - K_i M_i] \Sigma_i^- \]
Smoothing

- Idea
  - We don’t have the best estimate of state - what about the future?
  - Run two filters, one moving forward, the other backward
  - Now combine state estimates
Data Association

- **Nearest neighbours**
  - choose the measurement with highest probability given predicted state
  - popular, but can lead to catastrophe
- **Probabilistic Data Association**
  - combine measurements, weighting by probability given predicted state
  - gate using predicted state
Figure 4: Example tracks of corner features.  
Figure 5: Example groups of corner features.

from A Real-Time Computer Vision System for Measuring Traffic Parameters, Beymer, McClachlan, Coifman and Malik et al. p.500, in the fervent hope of receiving permission
Beyond the Kalman Filter

- Various phenomena lead to multiple modes
  - nonlinear dynamics
  - kinematic ambiguities
  - data association problems
- Kalman filters represent these poorly
  - alternatives
    - Mixture models
    - particle filters
Multiple Modes from Data Association

- Linear dynamics, Linear measurement, two measurements
  - Both Gaussian, one depends on state and other doesn’t
  - Not known which depends on state
- One hidden variable per frame
- Leads to $2^{\text{(number of frames)}}$ mixture of Gaussians