image alignment, feature tracking, and the Kalman filter
image alignment applications

- local alignment:
  - tracking
  - stereo

- global alignment:
  - camera jitter elimination
  - image enhancement
  - panoramic mosaicing
video stabilization

external videos
image enhancement
image stitching
panorama mosaicing
Photo Tourism
Exploring photo collections in 3D

Noah Snavely    Steven M. Seitz   Richard Szeliski
University of Washington    Microsoft Research

SIGGRAPH 2006
Video Matching
correspondence approaches

✦ optical flow
✦ correlation
✦ correlation + optical flow
✦ any of the above, iterated (e.g., Lucas-Kanade)
✦ any of the above, coarse-to-fine
correspondence approaches

✦ optical flow
✦ correlation
✦ correlation + optical flow
✦ any of the above, iterated (e.g., Lucas-Kanade)
✦ any of the above, coarse-to-fine
optical flow for image registration

- compute local matches
- least-squares fit to motion model
- problem: outliers
outlier rejection

- robust estimation: tolerant to outliers
- in general, methods based on absolute value more robust than square:

\[
\text{minimize } \sum |x_i - f|, \text{ not } \sum (x_i - f)^2
\]

- also RANdom SAmple Consensus (RANSAC)
correspondence approaches

- optical flow
- correlation
- correlation + optical flow
- any of the above, iterated (e.g., Lucas-Kanade)
- any of the above, coarse-to-fine
correlation / search methods

- assume translation only
- given images $I_1$ and $I_2$
- for each translation $(t_x, t_y)$ compute

$$c(I_1, I_2, t) = \sum_i \sum_j \psi(I_1(i, j), I_2(i - t_x, j - t_y))$$

- select translation that maximizes $c$
- depending on window size, local or global

[Tomasi and Kanade]
sum of squared differences

✦ intuitive measure (we’ve seen before):

\[ \psi(u, v) = -(u - v)^2 \]

✦ negative sign so that higher values mean greater similarity

✦ disadvantage: ?
cross-correlation

- statistical definition of correlation:

\[ \psi(u, v) = uv \]

- disadvantage: ?
normalized cross-correlation

\[ \psi(w, w') = \frac{(w - \bar{w}) \cdot (w' - \bar{w}')}{|w - \bar{w}||w' - \bar{w}'|} \]

[Refs: Forsyth & Ponce]
local vs. global

- correlation with local windows not too expensive
- high cost if window size = entire image
- but computation looks like convolution
  - FFT to the rescue!
correlation

\[ c = \sum_i \sum_j I_1(i, j) I_2(i - \Delta x, j - \Delta y) \]

\[ \mathcal{F}(c) = \mathcal{F}(I_1) \times \mathcal{F}_{\text{translated}}(I_2) \]
correspondence approaches

- optical flow
- correlation
- correlation + optical flow
- any of the above, iterated (e.g., Lucas-Kanade)
- any of the above, coarse-to-fine
correlation + optical flow

- use e.g. phase correlation to find average translation (may be large)
- use optical flow to find local motions
correspondence approaches

✦ optical flow
✦ correlation
✦ correlation + optical flow
✦ any of the above, iterated (e.g., Lucas-Kanade)
✦ any of the above, coarse-to-fine
correspondence approaches

- optical flow
- correlation
- correlation + optical flow
- any of the above, iterated (e.g., Lucas-Kanade)
- any of the above, coarse-to-fine
image pyramids

Idea: Represent NxN image as a “pyramid” of 1x1, 2x2, 4x4,..., $2^k x 2^k$ images (assuming N=$2^k$)

level $k$ (= 1 pixel)

level $k-1$

level $k-2$

... level 0 (= original image)
coarse-to-fine

- compute optical flow / correlation at coarsest level
- use as starting position to estimate smaller displacements at next finer levels

In the end:
- resolve large motion at coarse levels
- lock onto detail at finer levels
object tracking

HMDs

[Welch & Bishop]

[ Birchfield]
object tracking

- local region
- take advantage of many frames
  - prediction, uncertainty estimation
  - noise filtering
  - handle short occlusions
The Kalman Filter

- assume results of experiment (i.e., optical flow) are noisy measurements of system state
- model how system evolves
- optimal combination of system model and observations
- prediction / correction framework

Acknowledgment: much of the following material is based on the SIGGRAPH 2001 course by Greg Welch and Gary Bishop (UNC).
simple example

- measurement of a single point $z_1$
- variance $\sigma_1^2$ (uncertainty $\sigma_1$)
- best estimate of true position $\hat{x}_1 = z_1$
- uncertainty in best estimate $\hat{\sigma}_1^2 = \sigma_1^2$
simple example

- second measurement $z_2$ with variance $\sigma_2^2$
- best estimate of true position?

$z_1$  $z_2$
simple example

- second measurement $z_2$ with variance $\sigma_2^2$
- best estimate of true position: weighted average

\[
\hat{x}_2 = \frac{\frac{1}{\sigma_1^2} z_1 + \frac{1}{\sigma_2^2} z_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}
\]

\[
\hat{x}_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} (z_2 - \hat{x}_1)
\]

\[
\hat{\sigma}_2^2 = \frac{1}{\frac{1}{\hat{\sigma}_1^2} + \frac{1}{\sigma_2^2}}
\]
online weighted average

- combine successive measurements into constantly-improving estimate
- uncertainty decreases over time
- only need to keep current measurement, last estimate of state and uncertainty
terminology

- in this example, position is state (in general, any vector)
- state can be assumed to evolve over time according to a system model or process model (in this example, “nothing changes”)
- measurements (possibly incomplete, possibly noisy) according to a measurement model
- best estimate of $\hat{x}$ with covariance P
linear models

* “standard” Kalman filtering everything must be linear

* system model:

\[ x_k = \Phi_{k-1} x_{k-1} + \epsilon_{k-1} \]

- state transition matrix
- additive noise with covariance \( Q \)
linear models

- measurement model:

\[ z_k = H_k x_k + \mu_k \]

- H is the “measurement matrix”
- the vector \( \mu \) is “measurement noise”, assumed to have covariance \( R \)
position + velocity model

- add velocity to the “state vector”:

\[
\begin{align*}
    z_k &= H_k x_k + \mu_k \\
    x_k &= \Phi_{k-1} x_{k-1} + \epsilon_{k-1} \\
    \Phi_k &= \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix} \\
    H &= \begin{bmatrix} 1 & 0 \end{bmatrix}
\end{align*}
\]
**prediction/correction**

- **predict new state**

\[ x'_k = \Phi_{k-1} \hat{x}_{k-1} \]

\[ P'_k = \Phi_{k-1} P_{k-1} \Phi^T_{k-1} + Q_{k-1} \]

- **correct to take new measurements into account**

\[ \hat{x}_k = x'_k + K_z (z_k - H_k x'_k) \]

\[ P_k = (I - K_k H_k) P'_k \]
Kalman gain

- weighting of process model vs. measurements:

\[ K_k = P_k' H_k^T (H_k P_k' H_k^T + R_k)^{-1} \]

- compare to what we saw before:

\[ \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \]
results: position-only model

moving

still

[Welch & Bishop]
results: position-velocity model

moving

still

[Welch & Bishop]
extension: multiple models

- simultaneously run many KFs with different system models
- estimate probability each KF is correct
- final estimate: weighted average
result: multiple models
result: multiple models

[Welch & Bishop] Tuesday, April 19, 2011
extension: SCAAT

- H can be different at different time steps
  - different sensors, types of measurements
  - sometimes measure only part of state
- Single Constraint At A Time (SCAAT)
  - incorporate results from one sensor at once
  - alternative: wait until you have measurements from enough sensors to know complete state (MCAAT)
  - MCAAT more complex, but sometimes necessary for initialization
UNC HiBall
UNC HiBall

- 6 cameras, looking at LEDs on ceiling
- LEDs flash over time
HiBall state model has nonlinear degrees of freedom (rotations)

Extended Kalman Filter allows nonlinearities by:
- using generalized functions instead of matrices
- linearizing functions to project forward
- like first-order Taylor series expansion
- only have to evaluate Jacobians (partial derivatives), not invert process/measurement functions
other extensions

✦ on-line noise estimation
✦ using known system input (e.g., actuators)
✦ using information from both past and future
✦ non-Gaussian noise and particle filtering