COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Tracking

EECS 598-08 Fall 2014
Foundations of Computer Vision

Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

Readings: FP 11; SZ 8
Date: 10/20/14

From dense motion to tracking

F. Pernici: http://www.youtube.com/watch?v=yTvEzWg1cw0

From dense motion to tracking

Von Hardenburg ACM PUI 2001

From tracking to …

The VOICE of Mind's Eye

Video On an Index Card Engine

Demo Video for SUNY Buffalo's Mind's Eye Effort

PI: Jason Corso jcorso@buffalo.edu

Funded under contract W911NF-10-2-0062

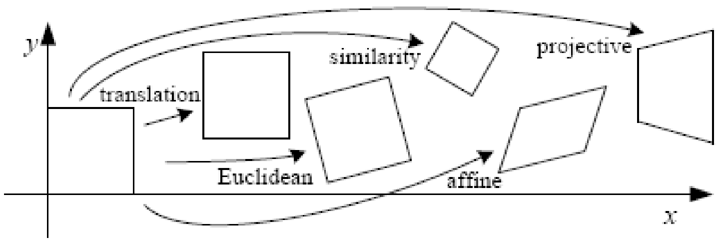
Plan

- Parametric Motion Estimation
  – From optical flow per pixel/block toward a motion model of certain regions or patches in the video.

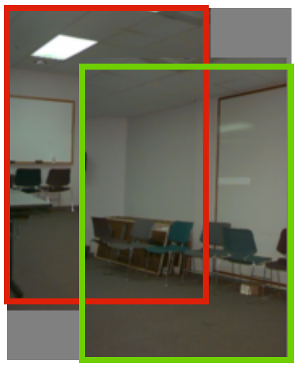
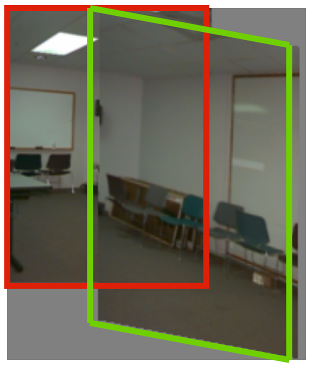
- Tracking and Filtering

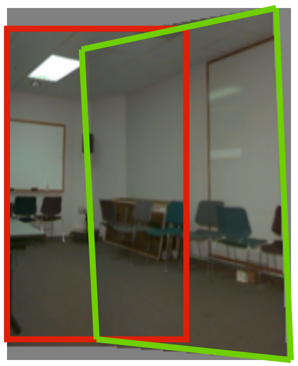
- Motion Segmentation
  – Interesting topic but not exactly tracking…

Motion models



Translation | **Affine** | **Perspective** | **3D rotation**



2 unknowns | **6 unknowns** | **8 unknowns** | **3 unknowns**

Source: Szeliski

Example: Affine Motion

$$u(x,y) = a_1 + a_2 x + a_3 y$$
$$v(x,y) = a_4 + a_5 x + a_6 y$$

- Substituting into the B.C. Equation:

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

Each pixel provides 1 linear constraint in 6 *global* unknowns

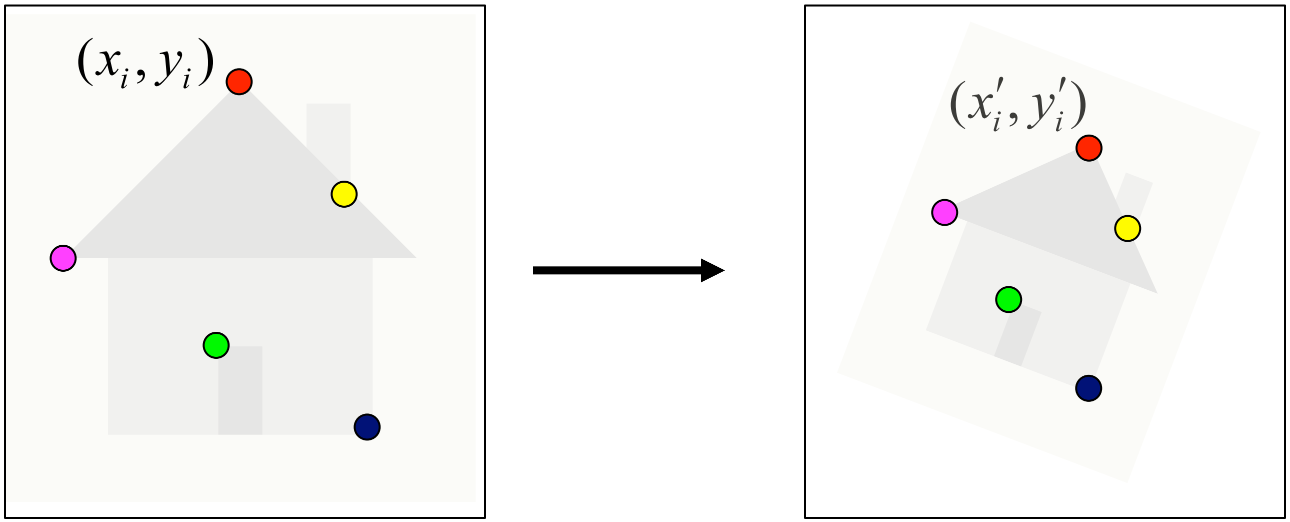
Least Squares Minimization (over all pixels):

$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

Source: Szeliski

Tracking with a motion model

- Two views presumed in temporal sequence…**track** or analyze **spatio-temporal gradient**

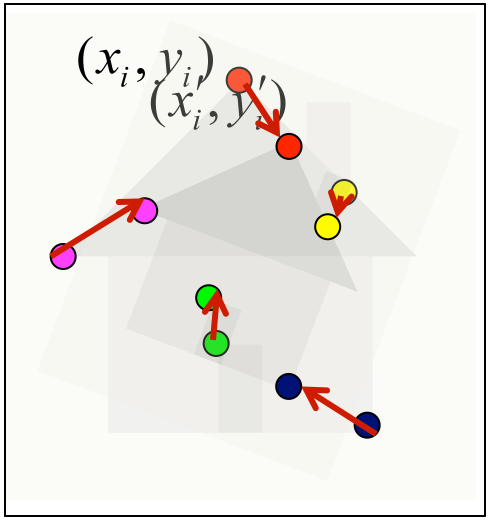


- ***Sparse or dense in first frame***
- ***Search in second frame***
- ***Motion models expressed in terms of position change***

Source: Szeliski

Tracking with a motion model

- Two views presumed in temporal sequence…**track** or analyze **spatio-temporal gradient**

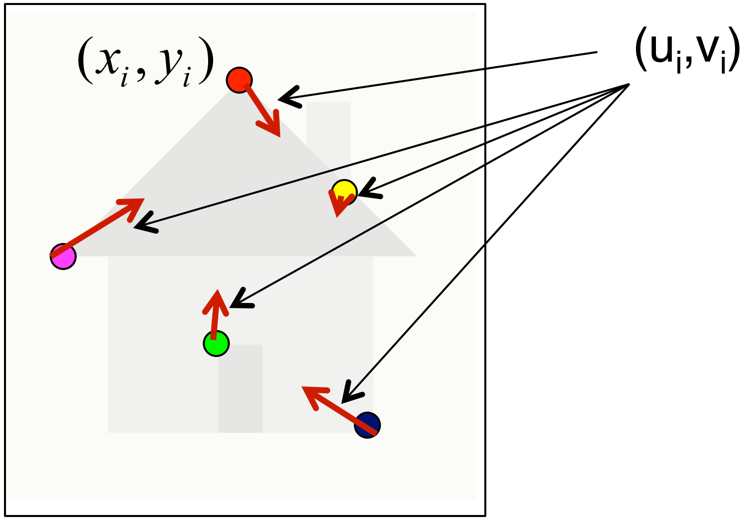


- ***Sparse or dense in first frame***
- ***Search in second frame***
- ***Motion models expressed in terms of position change***

Source: Szeliski

Tracking with a motion model

- Two views presumed in temporal sequence…**track** or analyze **spatio-temporal gradient**

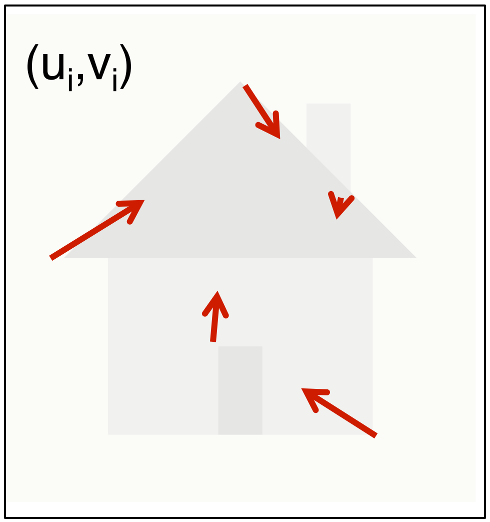


- ***Sparse or dense in first frame***
- ***Search in second frame***
- ***Motion models expressed in terms of position change***

Source: Szeliski

Tracking with a motion model

- Two views presumed in temporal sequence…**track** or analyze **spatio-temporal gradient**

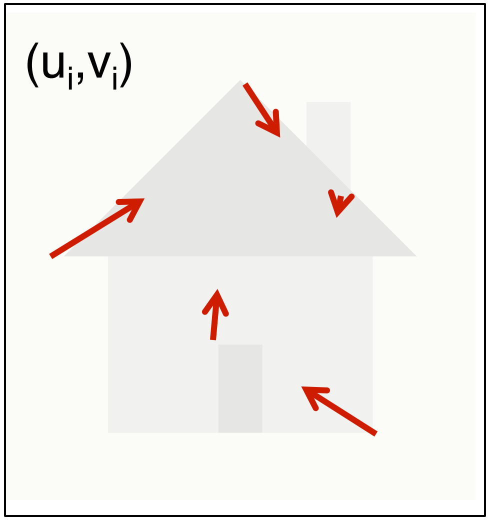


- ***Sparse or dense in first frame***
- ***Search in second frame***
- ***Motion models expressed in terms of position change***

Source: Szeliski

Tracking with a motion model

- Two views presumed in temporal sequence…**track** or analyze **spatio-temporal gradient**



Affine motion model:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} a_2 & a_3 \\ a_5 & a_6 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} a_1 \\ a_4 \end{bmatrix}$$

$$u(x, y) = a_1 + a_2 x + a_3 y$$
$$v(x, y) = a_4 + a_5 x + a_6 y$$

- ***Sparse or dense in first frame***
- ***Search in second frame***
- ***Motion models expressed in terms of position change***

Source: Szeliski

Other 2D Motion Models

Quadratic – instantaneous approximation to planar motion

$$u = q_1 + q_2 x + q_3 y + q_7 x^2 + q_8 xy$$
$$v = q_4 + q_5 x + q_6 y + q_7 xy + q_8 y^2$$

Projective – exact planar motion

$$x' = \frac{h_1 + h_2 x + h_3 y}{h_7 + h_8 x + h_9 y}$$
$$y' = \frac{h_4 + h_5 x + h_6 y}{h_7 + h_8 x + h_9 y}$$

and

$$u = x' - x, \quad v = y' - y$$

3D Motion Models

Instantaneous camera motion:

Global parameters: $\boxed{\Omega_X, \Omega_Y, \Omega_Z, T_X, T_Y, T_Z}$

Local Parameter: $\boxed{Z(x, y)}$

$$u = -xy\Omega_X + (1 + x^2)\Omega_Y - y\Omega_Z + (T_X - T_Z x)/Z$$
$$v = -(1 + y^2)\Omega_X + xy\Omega_Y - x\Omega_Z + (T_Y - T_Z x)/Z$$

Homography+Epipole

Global parameters: $\boxed{h_1, \ldots, h_9, t_1, t_2, t_3}$

Local Parameter: $\boxed{\gamma(x, y)}$

$$x' = \frac{h_1 x + h_2 y + h_3 + \gamma t_1}{h_7 x + h_8 y + h_9 + \gamma t_3}$$
$$y' = \frac{h_4 x + h_5 y + h_6 + \gamma t_1}{h_7 x + h_8 y + h_9 + \gamma t_3}$$
$$\text{and:} \quad u = x' - x, \quad v = y' - y$$

Residual Planar Parallax Motion

Global parameters: $\boxed{t_1, t_2, t_3}$

Local Parameter: $\boxed{\gamma(x, y)}$

$$u = x^w - x = \frac{\gamma}{1 + \gamma t_3}(t_3 x - t_1)$$
$$v = y^w - x = \frac{\gamma}{1 + \gamma t_3}(t_3 y - t_2)$$

Source: Szeliski

Discrete Search vs. Gradient Based

- Consider image I translated by u_0, v_0

$$I_0(x, y) = I(x, y)$$
$$I_1(x + u_0, y + v_0) = I(x, y) + \eta_1(x, y)$$

$$E(u, v) = \sum_{x,y} (I(x, y) - I_1(x + u, y + v))^2$$
$$= \sum_{x,y} (I(x, y) - I(x - u_0 + u, y - v_0 + v) - \eta_1(x, y))^2$$

- The discrete search method simply searches for the best estimate.
- The gradient method linearizes the intensity function and solves for the estimate

Source: Szeliski

Correlation and SSD

- For larger displacements, do template matching
  - Define a small area around a pixel as the template
  - Match the template against each pixel within a search area in next image.
  - Use a match measure such as correlation, normalized correlation, or sum-of-squares difference
  - Choose the maximum (or minimum) as the match
  - Sub-pixel estimate (Lucas-Kanade)

Source: Szeliski

Shi-Tomasi feature tracker

1. Find good features (min eigenvalue of 2×2 Hessian)
2. Use Lucas-Kanade to track with pure translation
3. Fit affine motion model (registration) with first feature patch
4. Terminate tracks whose dissimilarity gets too large
5. Start new tracks when needed

Source: Szeliski

Tracking results

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).

Source: Szeliski

Tracking - dissimilarity

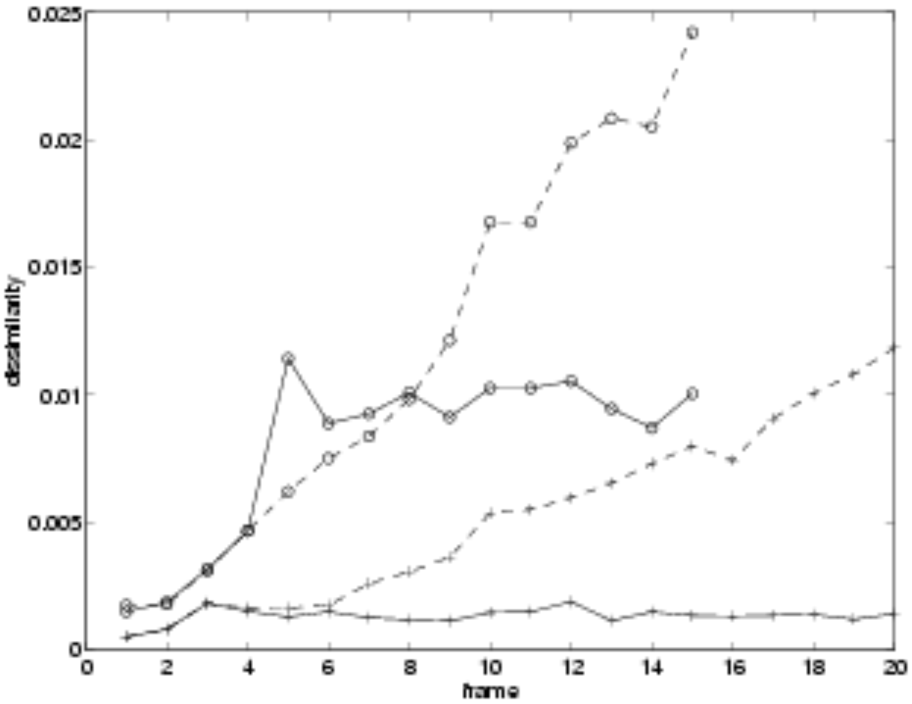


Figure 3: Pure translation (dashed) and affine motion (solid) dissimilarity measures for the window sequence of figure 1 (plusses) and 4 (circles).

Source: Szeliski

Tracking results

Figure 13: Labels of some of the features in figure 11.

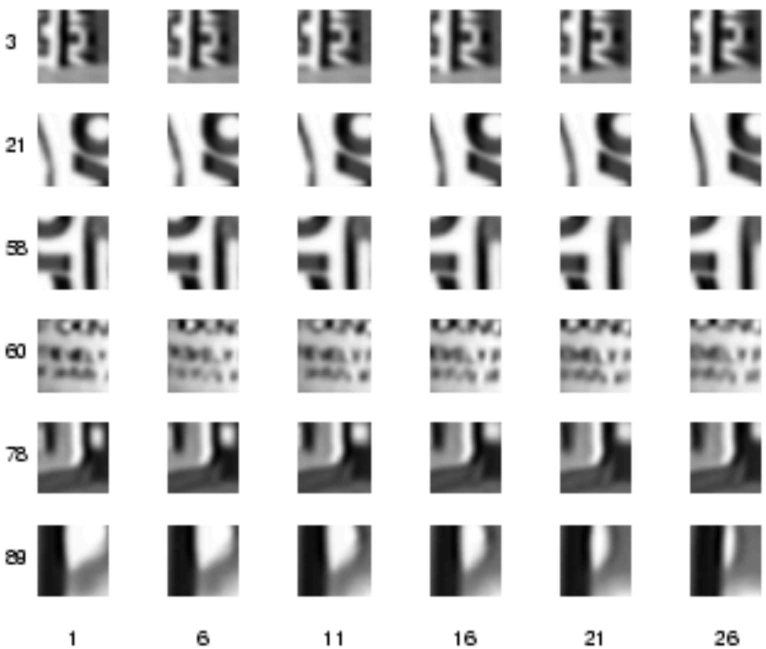


Figure 14: Six sample features through six sample frames.

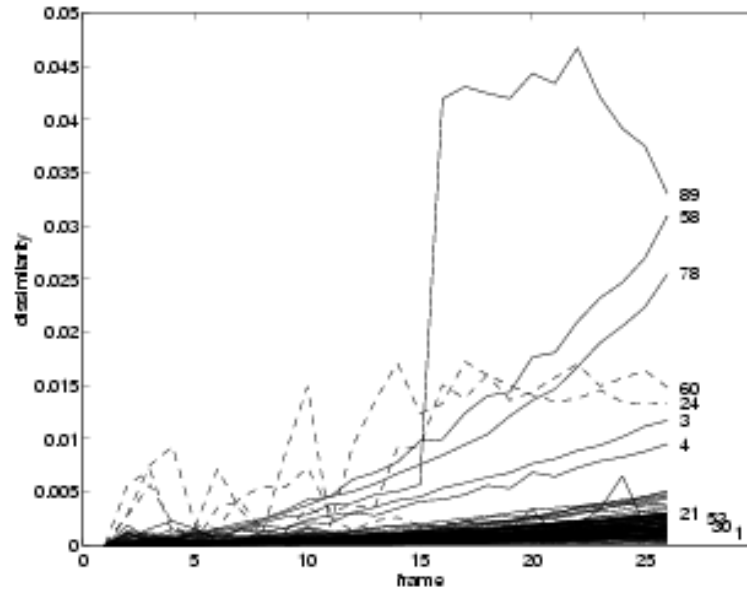


Figure 15: Affine motion dissimilarity for the features in figure 11. Notice the good discrimination between good and bad features. Dashed plots indicate aliasing (see text).

Features 24 and 60 deserve a special discussion, and

Source: Szeliski

Shi-Tomasi Feature Extraction Example

Source: https://www.youtube.com/watch?v=jKW2y5NhO_k

Points As Constraints in Tracking

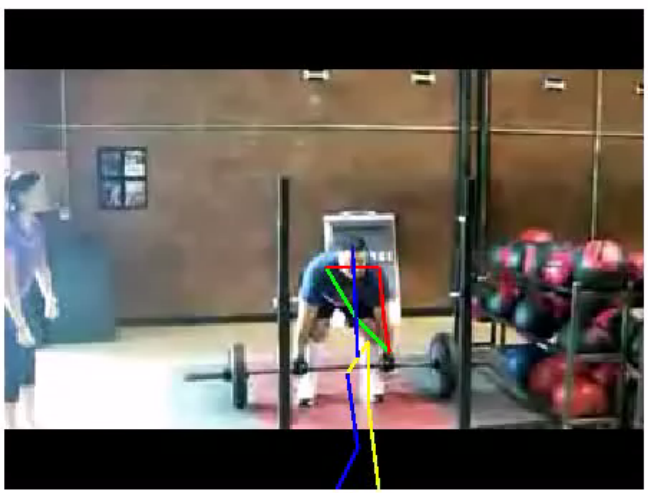
Filtering for Tracking

Tracking scenarios

- Follow a point
- Follow a template
- Follow a changing template
- Follow all the elements of a moving person, fit a model to it.

Source: Darrell

It can get very challenging…



Method: Yang and Ramanan. “Articulated Pose Estimation with Flexible Mixtures-of-Parts.” CVPR 2011.

Things to consider in tracking

What are the dynamics of the thing being tracked?

How is it observed?

Source: Darrell

Three main issues in tracking

- **Prediction:** we have seen $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{i-1}$ — what state does this set of measurements predict for the i'th frame? to solve this problem, we need to obtain a representation of $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$.

- **Data association:** Some of the measurements obtained from the i-th frame may tell us about the object's state. Typically, we use $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$ to identify these measurements.

- **Correction:** now that we have $\boldsymbol{y}_i$ — the relevant measurements — we need to compute a representation of $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$.

Source: Darrell

Simplifying Assumptions

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

$$P(\boldsymbol{X}_i|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i-1}) = P(\boldsymbol{X}_i|\boldsymbol{X}_{i-1})$$

  This assumption hugely simplifies the design of algorithms, as we shall see; furthermore, it isn't terribly restrictive if we're clever about interpreting $\boldsymbol{X}_i$ as we shall show in the next section.

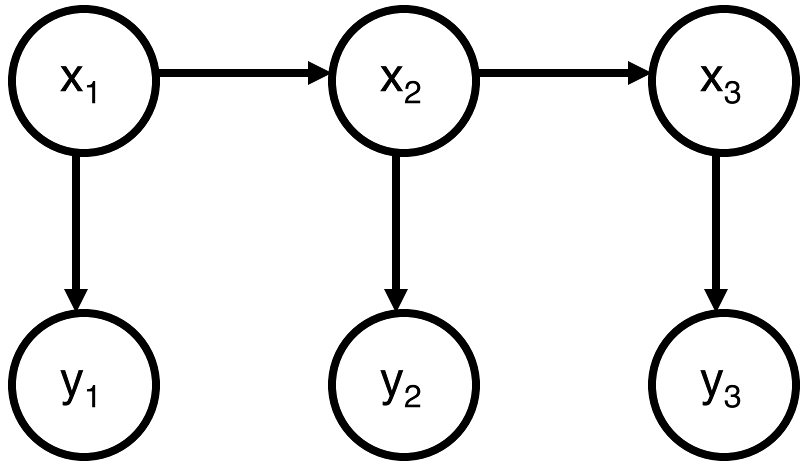
- **Measurements depend only on the current state:** we assume that $\boldsymbol{Y}_i$ is conditionally independent of all other measurements given $\boldsymbol{X}_i$. This means that

$$P(\boldsymbol{Y}_i, \boldsymbol{Y}_j, \ldots \boldsymbol{Y}_k|\boldsymbol{X}_i) = P(\boldsymbol{Y}_i|\boldsymbol{X}_i)P(\boldsymbol{Y}_j, \ldots, \boldsymbol{Y}_k|\boldsymbol{X}_i)$$

  Again, this isn't a particularly restrictive or controversial assumption, but it yields important simplifications.

Source: Darrell

Kalman filter graphical model and corresponding factorized joint probability



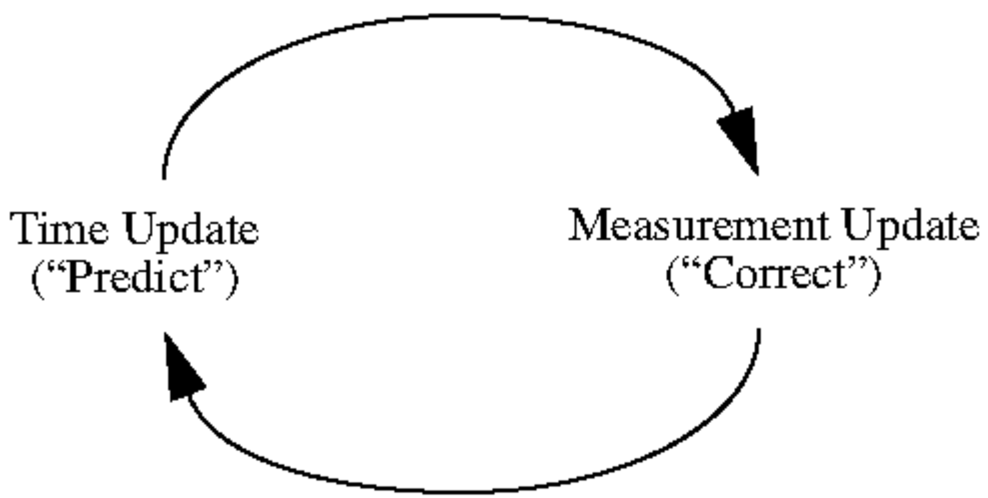
$$P(x_1, x_2, x_3, y_1, y_2, y_3) =$$

$$P(x_1)P(y_1 \mid x_1)P(x_2 \mid x_1)P(y_2 \mid x_2)P(x_3 \mid x_2)P(y_3 \mid x_3)$$

Source: Darrell

Tracking as induction

- Make a measurement starting in the 0^{th} frame
- Then: assume you have an estimate at the ith frame, after the measurement step.
- Show that you can do prediction for the i+1th frame, and measurement for the i+1th frame.



Source: Darrell

Base case

Firstly, we assume that we have $P(\boldsymbol{X}_0)$

$$P(\boldsymbol{X}_0|\boldsymbol{Y}_0 = \boldsymbol{y}_0) = \frac{P(\boldsymbol{y}_0|\boldsymbol{X}_0)P(\boldsymbol{X}_0)}{P(\boldsymbol{y}_0)}$$

$$\propto P(\boldsymbol{y}_0|\boldsymbol{X}_0)P(\boldsymbol{X}_0)$$

Source: Darrell

Prediction step

Prediction

Prediction involves representing

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

given

$$P(\boldsymbol{X}_{i-1}|\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{i-1}).$$

Our independence assumptions make it possible to write

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

Source: Darrell

Update step

Correction

Correction involves obtaining a representation of

$$P(\boldsymbol{X}_i|\boldsymbol{y}_0, \ldots, \boldsymbol{y}_i)$$

given

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

Our independence assumptions make it possible to write

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

Source: Darrell

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

Source: Darrell

Recall the three main issues in tracking

- **Prediction:** we have seen $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{i-1}$ — what state does this set of measurements predict for the i'th frame? to solve this problem, we need to obtain a representation of $P(\boldsymbol{X}_i|\boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$.

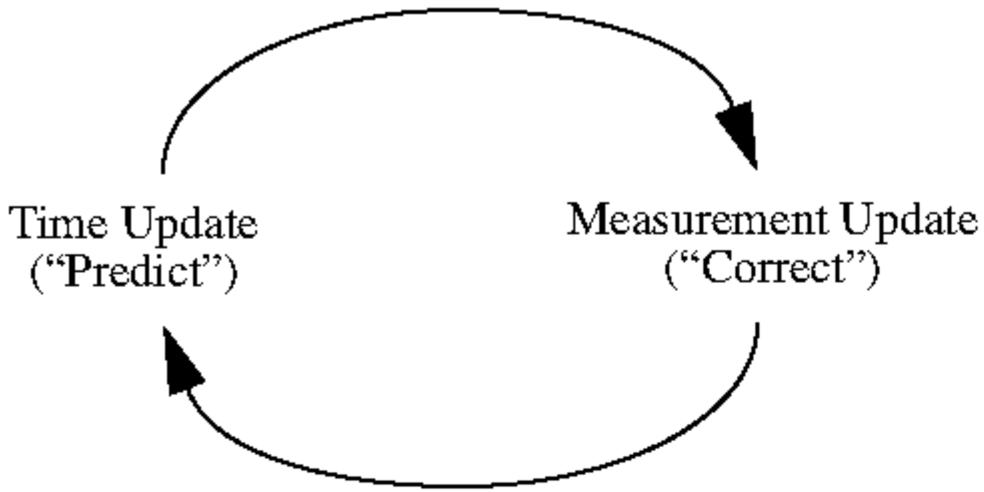
- **Data association:** Some of the measurements obtained from the i-th frame may tell us about the object's state. Typically, we use $P(\boldsymbol{X}_i|\boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$ to identify these measurements.

- **Correction:** now that we have $\boldsymbol{y}_i$ — the relevant measurements — we need to compute a representation of $P(\boldsymbol{X}_i|\boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$.

(Ignore data association for now)

Source: Darrell

The Kalman Filter



[figure from http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html]

Source: Darrell

The Kalman Filter in 1D

- Dynamic Model

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

$$y_i \sim N(m_i x_i, \sigma_{m_i}^2)$$

- Notation

mean of $P(X_i|y_0, \ldots, y_{i-1})$ as $\overline{X}_i^-$ ← Predicted mean

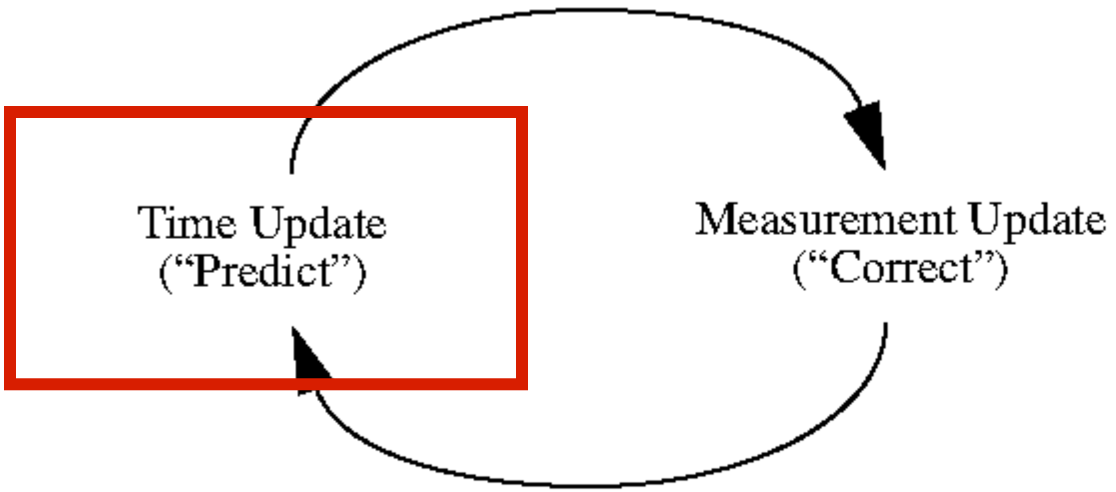
mean of $P(X_i|y_0, \ldots, y_i)$ as $\overline{X}_i^+$ ← Corrected mean

the standard deviation of $P(X_i|y_0, \ldots, y_{i-1})$ as σ_i^-

of $P(X_i|y_0, \ldots, y_i)$ as σ_i^+

Source: Darrell

The Kalman Filter



Source: Darrell

Prediction for 1D Kalman filter

- The new state is obtained by
  - multiplying old state by known constant
  - adding zero-mean noise

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

- Therefore, predicted mean for new state is
  - constant times mean for old state
- Old variance is normal random variable
  - variance is multiplied by square of constant
  - and variance of noise is added.

$$\overline{X}_i^- = d_i \overline{X}_{i-1}^+ \qquad (\sigma_i^-)^2 = \sigma_{d_i}^2 + (d_i \sigma_{i-1}^+)^2$$

Source: Darrell

Dynamic Model:

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

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

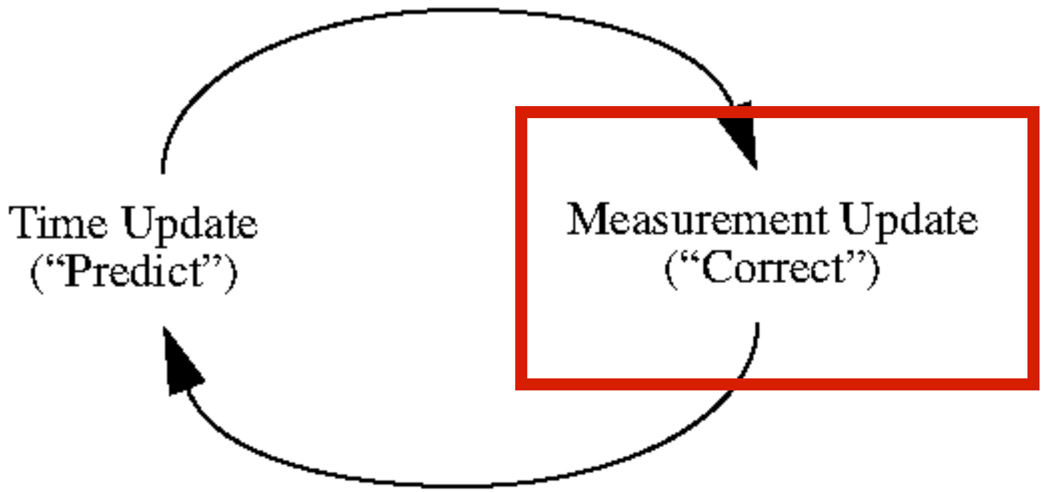
Start Assumptions: $\overline{x}_0^-$ and σ_0^- are known
Update Equations: Prediction

$$\overline{x}_i^- = d_i \overline{x}_{i-1}^+$$

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

Source: Darrell

The Kalman Filter



Source: Darrell

Measurement update for 1D Kalman filter

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

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

Notice:

– if measurement noise is small,
we rely mainly on the measurement,

– if it's large, mainly on the prediction

– σ does not depend on y

Source: Darrell

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: $\overline{x}_0^-$ and σ_0^- are known
Update Equations: Prediction

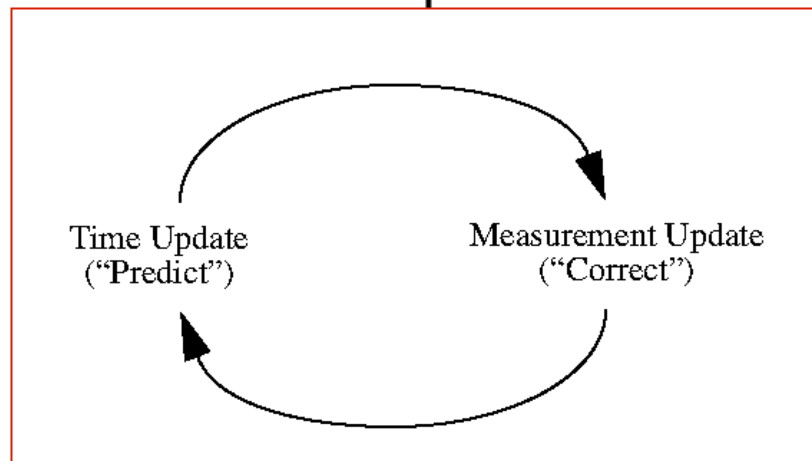
$$\overline{x}_i^- = d_i \overline{x}_{i-1}^+$$

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

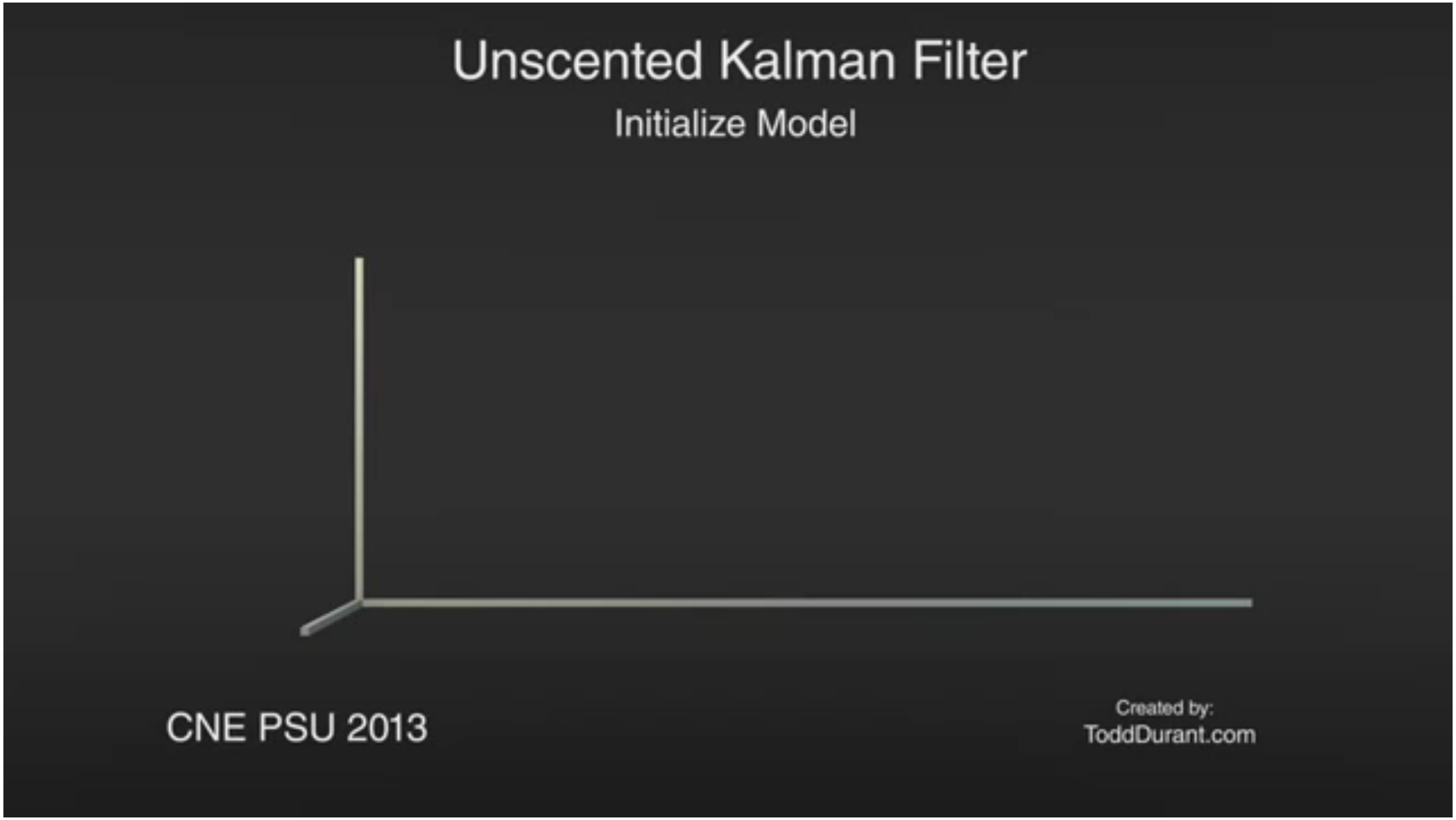
Update Equations: Correction

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

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



Source: Darrell



Source: https://www.youtube.com/watch?v=uiegHFqmkyE

KF Example Application

Source: https://www.youtube.com/watch?v=Y3TzhXYF0Lg

Kalman filter for computing an on-line average

- What Kalman filter parameters and initial conditions should we pick so that the optimal estimate for x at each iteration is just the average of all the observations seen so far?

Source: Darrell

odel:

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

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

ptions: $\overline{x}_0^-$ and σ_0^- are known

ations: Prediction

$$\overline{x}_i^- = d_i \overline{x}_{i-1}^+$$

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

ations: Correction

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

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

Source: Darrell

Kalman filter model

$$d_i = 1,\ m_i = 1,\ \sigma_{d_i} = 0,\ \sigma_{m_i} = 1$$

Initial conditions

$$\overline{x}_0^- = 0 \qquad \sigma_0^- = \infty$$

Iteration	0	1	2
$\overline{x}_i^-$	0	y_0	$\frac{y_0 + y_1}{2}$
$\overline{x}_i^+$	y_0	$\frac{y_0 + y_1}{2}$	$\frac{y_0 + y_1 + y_2}{3}$
σ_i^-	∞	1	$\frac{1}{\sqrt{2}}$
σ_i^+	1	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{3}}$

What happens if the x dynamics are given a non-zero variance?

Source: Darrell

: Model:

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

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

sumptions: $\overline{x}_0^-$ and σ_0^- are known

Equations: Prediction

$$\overline{x}_i^- = d_i \overline{x}_{i-1}^+$$

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

Equations: Correction

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

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

Source: Darrell

Kalman filter model

$$d_i = 1,\ m_i = 1,\ \sigma_{d_i} = 1,\ \sigma_{m_i} = 1$$

Initial conditions

$$\overline{x}_0^- = 0 \qquad \sigma_0^- = \infty$$

Iteration	0	1	2
$\overline{x}_i^-$	0	y_0	$\frac{y_0 + 2y_1}{3}$
$\overline{x}_i^+$	y_0	$\frac{y_0 + 2y_1}{3}$	$\frac{y_0 + 2y_1 + 5y_2}{8}$
σ_i^-	∞	$\sqrt{2}$	$\sqrt{\frac{5}{3}}$
σ_i^+	1	$\sqrt{\frac{2}{3}}$	$\sqrt{\frac{5}{8}}$

Linear dynamic models

- A linear dynamic model has the form

$$\mathbf{x}_i = N\left(\mathbf{D}_{i-1}\mathbf{x}_{i-1};\Sigma_{d_i}\right)$$

$$\mathbf{y}_i = N\left(\mathbf{M}_i\mathbf{x}_i;\Sigma_{m_i}\right)$$

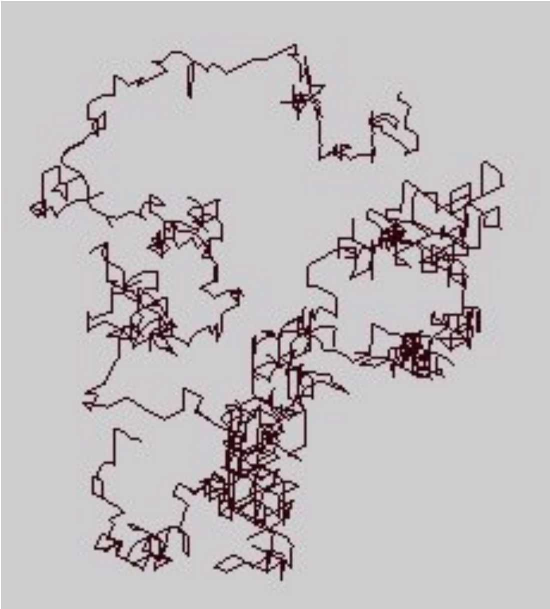
- This is much, much more general than it looks, and extremely powerful

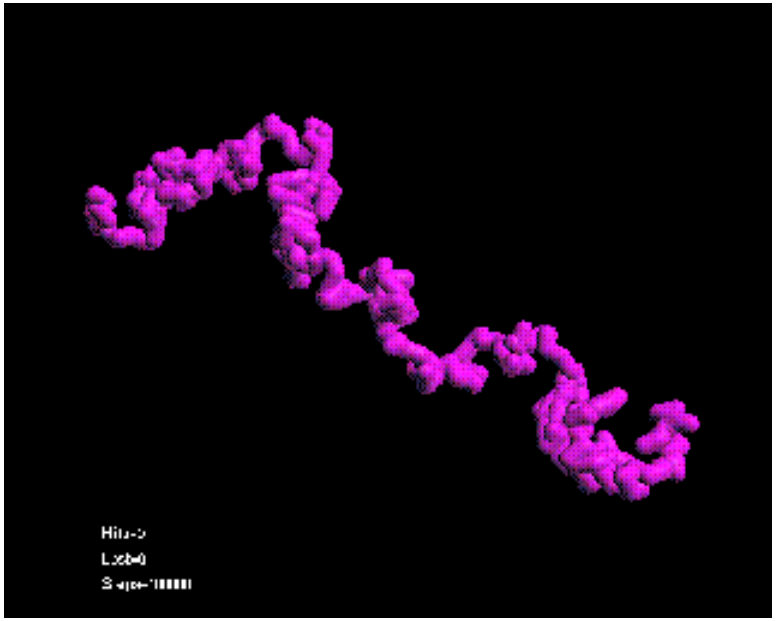
Examples of linear state space models

$$\mathbf{x}_i = N\left(\mathbf{D}_{i-1}\mathbf{x}_{i-1};\Sigma_{d_i}\right)$$
$$\mathbf{y}_i = N\left(\mathbf{M}_i\mathbf{x}_i;\Sigma_{m_i}\right)$$

- Drifting points
  - assume that the new position of the point is the old one, plus noise

D = Identity





cic.nist.gov/lipman/sciviz/images/random3.gif
http://www.grunch.net/synergetics/images/random3.jpg

Source: Darrell

Constant velocity

$$\mathbf{x}_i = N\left(\mathbf{D}_{i-1}\mathbf{x}_{i-1};\Sigma_{d_i}\right)$$

$$\mathbf{y}_i = N\left(\mathbf{M}_i\mathbf{x}_i;\Sigma_{m_i}\right)$$

- We have

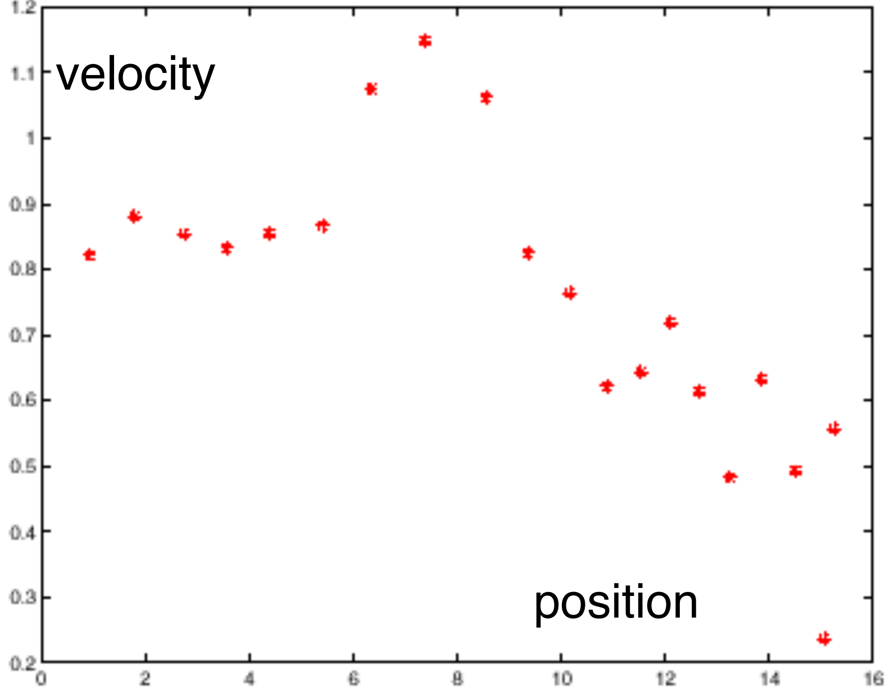
$$u_i = u_{i-1} + \Delta t v_{i-1} + \varepsilon_i$$

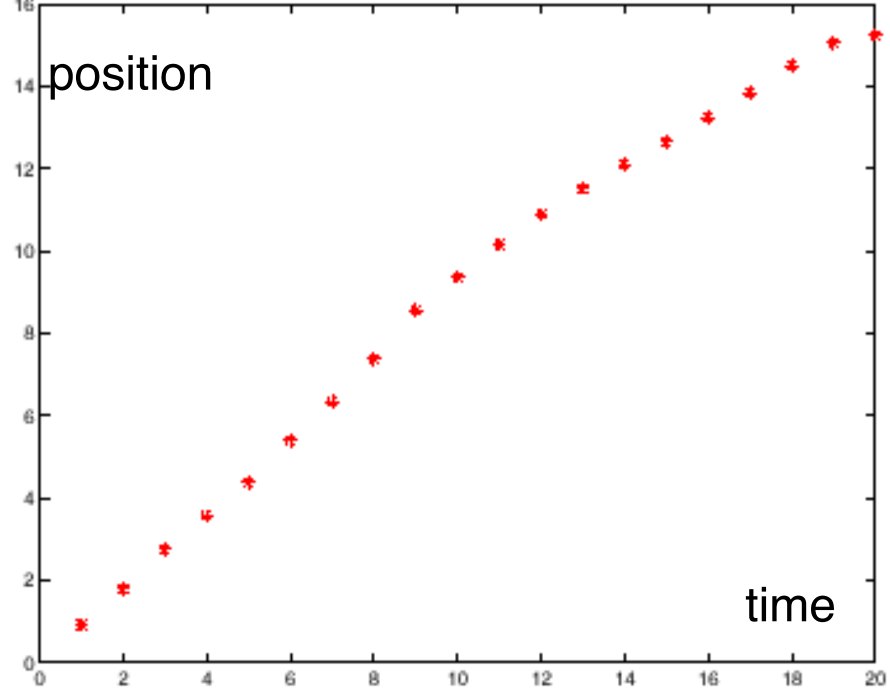
$$v_i = v_{i-1} + \varsigma_i$$

  - (the Greek letters denote noise terms)
- Stack (u, v) into a single state vector

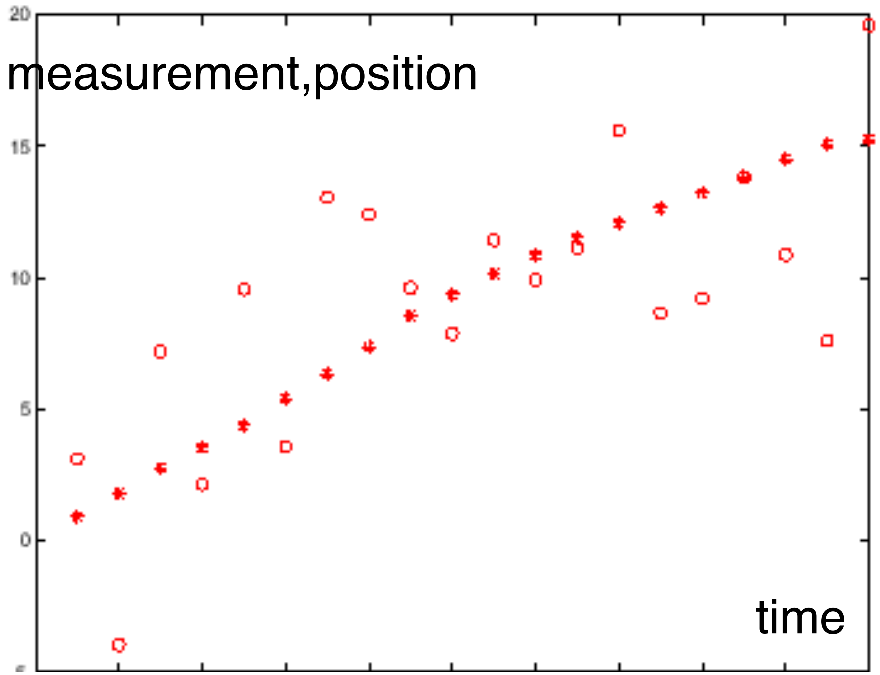
$$\underbrace{\begin{pmatrix} u \\ v \end{pmatrix}_i}_{\mathbf{x}_i} = \underbrace{\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}}_{\mathbf{D}_{i-1}} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}_{i-1}}_{\mathbf{x}_{i-1}} + \text{noise}$$

  - which is the form we had above





Constant Velocity Model



Source: Darrell

Constant acceleration

$$\mathbf{x}_i = N\left(\mathbf{D}_{i-1}\mathbf{x}_{i-1};\Sigma_{d_i}\right)$$
$$\mathbf{y}_i = N\left(\mathbf{M}_i\mathbf{x}_i;\Sigma_{m_i}\right)$$

- We have

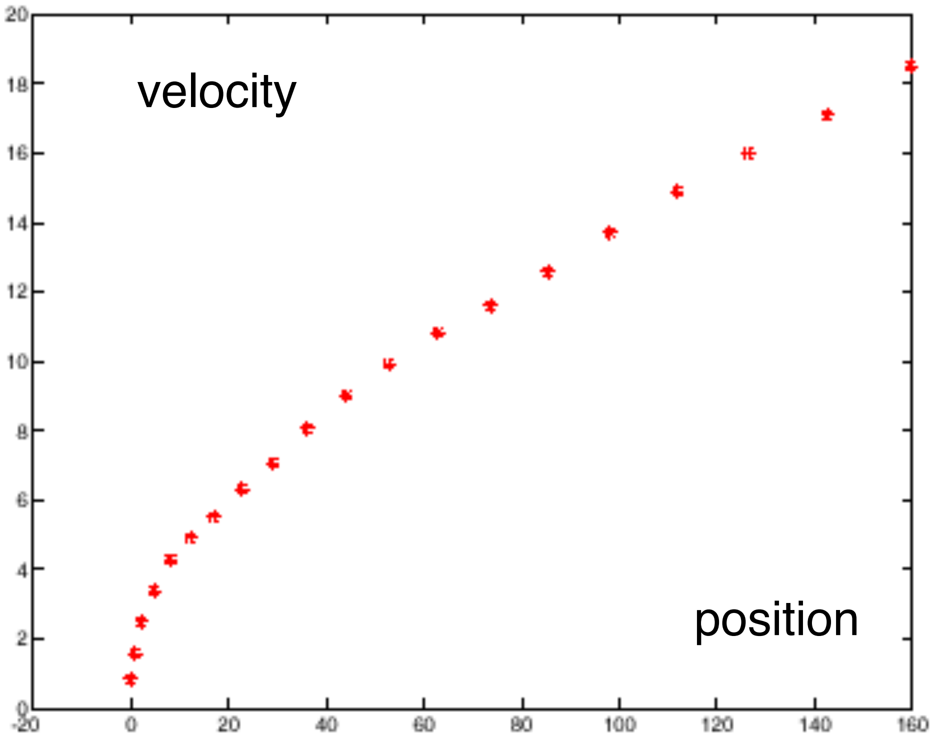
$$u_i = u_{i-1} + \Delta t v_{i-1} + \varepsilon_i$$
$$v_i = v_{i-1} + \Delta t a_{i-1} + \varsigma_i$$
$$a_i = a_{i-1} + \xi_i$$

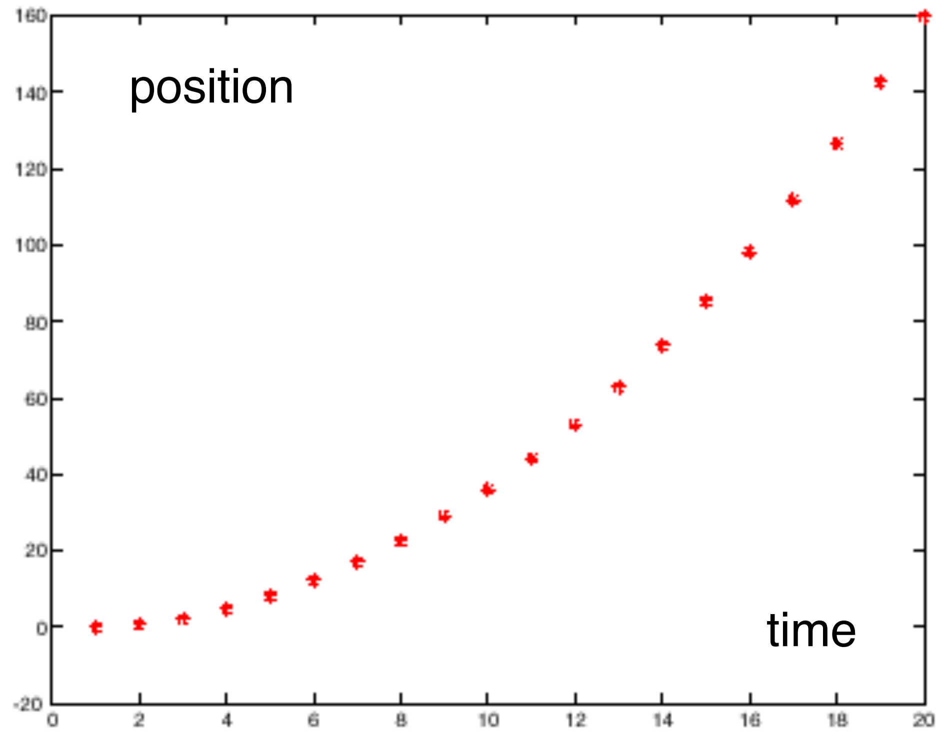
  - (the Greek letters denote noise terms)
- Stack (u, v) into a single state vector

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

  - which is the form we had above

Source: Darrell





Constant Acceleration Model

Source: Darrell

Periodic motion

$$\mathbf{x}_i = N\left(\mathbf{D}_{i-1}\mathbf{x}_{i-1};\Sigma_{d_i}\right)$$
$$\mathbf{y}_i = N\left(\mathbf{M}_i\mathbf{x}_i;\Sigma_{m_i}\right)$$

Assume we have a point, moving on a line with a periodic movement defined with a differential eq:

$$\frac{d^2p}{dt^2} = -p$$

can be defined as

$$\frac{d\boldsymbol{u}}{dt} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \boldsymbol{u} = \mathcal{S}\boldsymbol{u}$$

with state defined as stacked position and velocity u=(p, v)

Source: Darrell

Periodic motion

$$\mathbf{x}_i = N\left(\mathbf{D}_{i-1}\mathbf{x}_{i-1};\Sigma_{d_i}\right)$$

$$\mathbf{y}_i = N\left(\mathbf{M}_i\mathbf{x}_i;\Sigma_{m_i}\right)$$

$$\frac{d\boldsymbol{u}}{dt} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \boldsymbol{u} = \mathcal{S}\boldsymbol{u}$$

Take discrete approximation….(e.g., forward Euler integration with Δt stepsize.)

$$\begin{aligned} \underbrace{\boldsymbol{u}_i}_{\mathbf{x}_i} &= \boldsymbol{u}_{i-1} + \Delta t \frac{d\boldsymbol{u}}{dt} \\ &= \boldsymbol{u}_{i-1} + \Delta t \mathcal{S} \boldsymbol{u}_{i-1} \\ &= \underbrace{\begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix}}_{\mathbf{D}_{i-1}} \underbrace{\boldsymbol{u}_{i-1}}_{\mathbf{x}_{i-1}} \end{aligned}$$

Source: Darrell

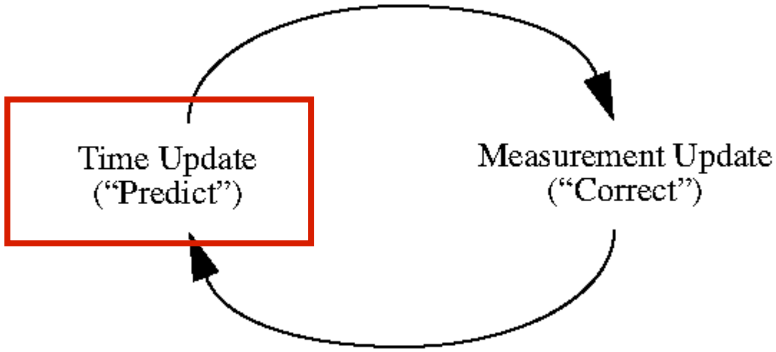
n-D

Generalization to n-D is straightforward but more complex.

Source: Darrell

n-D

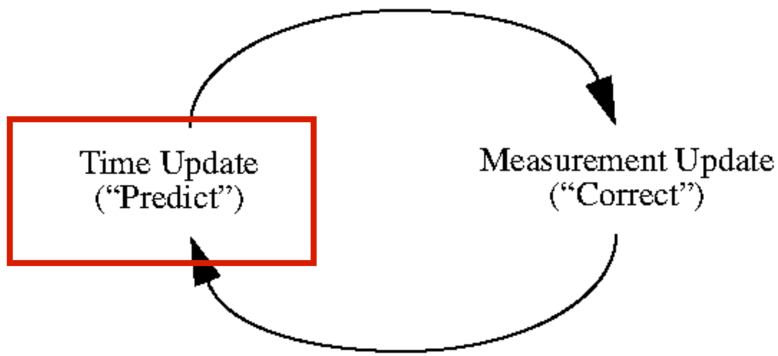
Generalization to n-D is straightforward but more complex.



Source: Darrell

n-D Prediction

Generalization to n-D is straightforward but more complex.



Prediction:

- Multiply estimate at prior time with forward model:

$$\overline{\boldsymbol{x}}_i^- = \mathcal{D}_i \overline{\boldsymbol{x}}_{i-1}^+$$

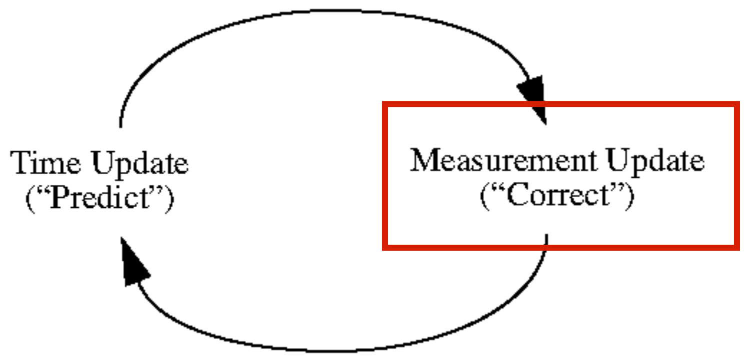
- Propagate covariance through model and add new noise:

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

Source: Darrell

n-D Correction

Generalization to n-D is straightforward but more complex.



Correction:

- Update *a priori* estimate with measurement to form *a posteriori*

n-D correction

Find linear filter on innovations

$$\overline{\boldsymbol{x}}_i^+ = \overline{\boldsymbol{x}}_i^- + \mathcal{K}_i \left[\boldsymbol{y}_i - \mathcal{M}_i \overline{\boldsymbol{x}}_i^- \right]$$

which minimizes *a posteriori* error covariance:

$$E\left[\left(x - \overline{x^+}\right)^T \left(x - \overline{x^+}\right)\right]$$

K is the *Kalman Gain* matrix. A solution is

$$\mathcal{K}_i = \Sigma_i^- \mathcal{M}_i^T \left[\mathcal{M}_i \Sigma_i^- \mathcal{M}_i^T + \Sigma_{m_i} \right]^{-1}$$

Source: Darrell

Kalman Gain Matrix

$$\overline{\boldsymbol{x}}_i^+ = \overline{\boldsymbol{x}}_i^- + \mathcal{K}_i \left[\boldsymbol{y}_i - \mathcal{M}_i \overline{\boldsymbol{x}}_i^- \right]$$

$$\boxed{\mathcal{K}_i = \Sigma_i^- \mathcal{M}_i^T \left[\mathcal{M}_i \Sigma_i^- \mathcal{M}_i^T + \Sigma_{m_i} \right]^{-1}}$$

As measurement becomes more reliable, K weights residual more heavily,

$$\lim_{\Sigma_m \to 0} K_i = M^{-1}$$

As prior covariance approaches 0, measurements are ignored:

$$\lim_{\Sigma_i^- \to 0} K_i = 0$$

Source: Darrell

Dynamic Model:

$$\boldsymbol{x}_i \sim N(\mathcal{D}_i \boldsymbol{x}_{i-1}, \Sigma_{d_i})$$

$$\boldsymbol{y}_i \sim N(\mathcal{M}_i \boldsymbol{x}_i, \Sigma_{m_i})$$

Start Assumptions: $\overline{\boldsymbol{x}}_0^-$ and Σ_0^- are known
Update Equations: Prediction

$$\overline{\boldsymbol{x}}_i^- = \mathcal{D}_i \overline{\boldsymbol{x}}_{i-1}^+$$

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

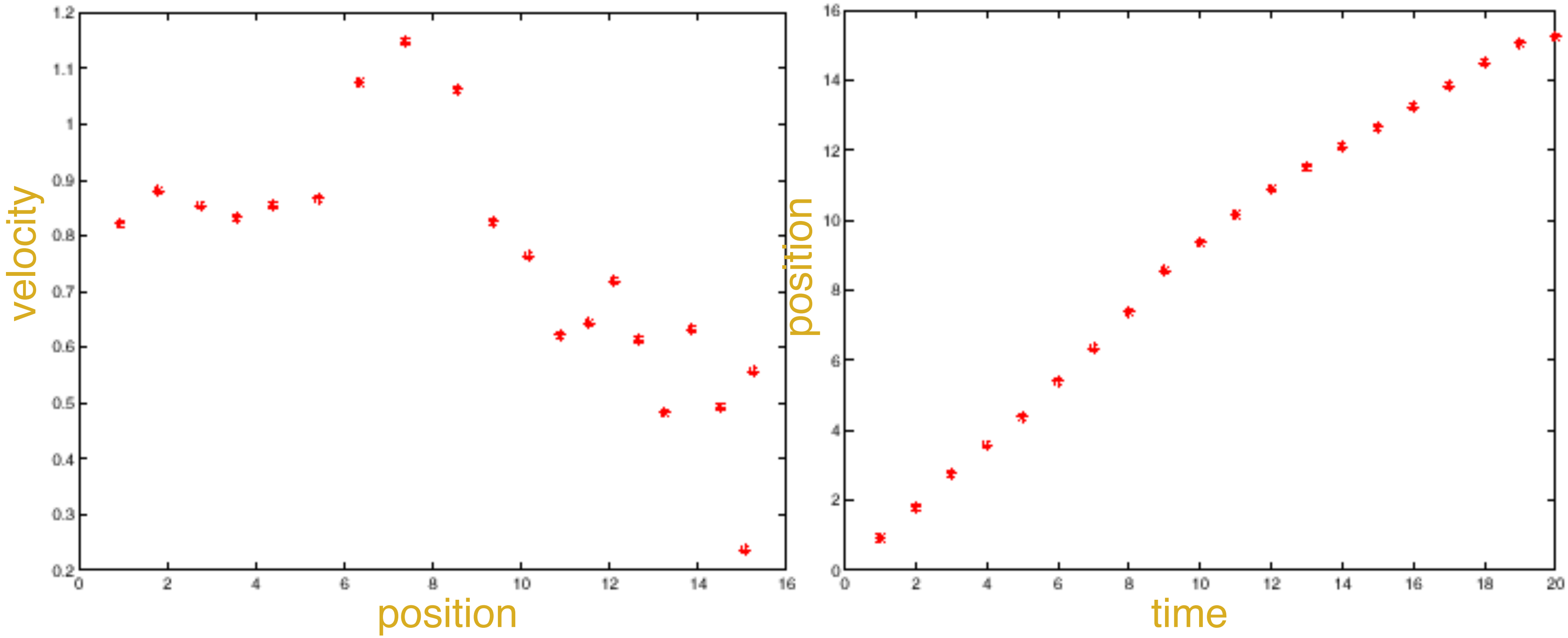
Update Equations: Correction

$$\mathcal{K}_i = \Sigma_i^- \mathcal{M}_i^T \left[\mathcal{M}_i \Sigma_i^- \mathcal{M}_i^T + \Sigma_{m_i} \right]^{-1}$$

$$\overline{\boldsymbol{x}}_i^+ = \overline{\boldsymbol{x}}_i^- + \mathcal{K}_i \left[\boldsymbol{y}_i - \mathcal{M}_i \overline{\boldsymbol{x}}_i^- \right]$$

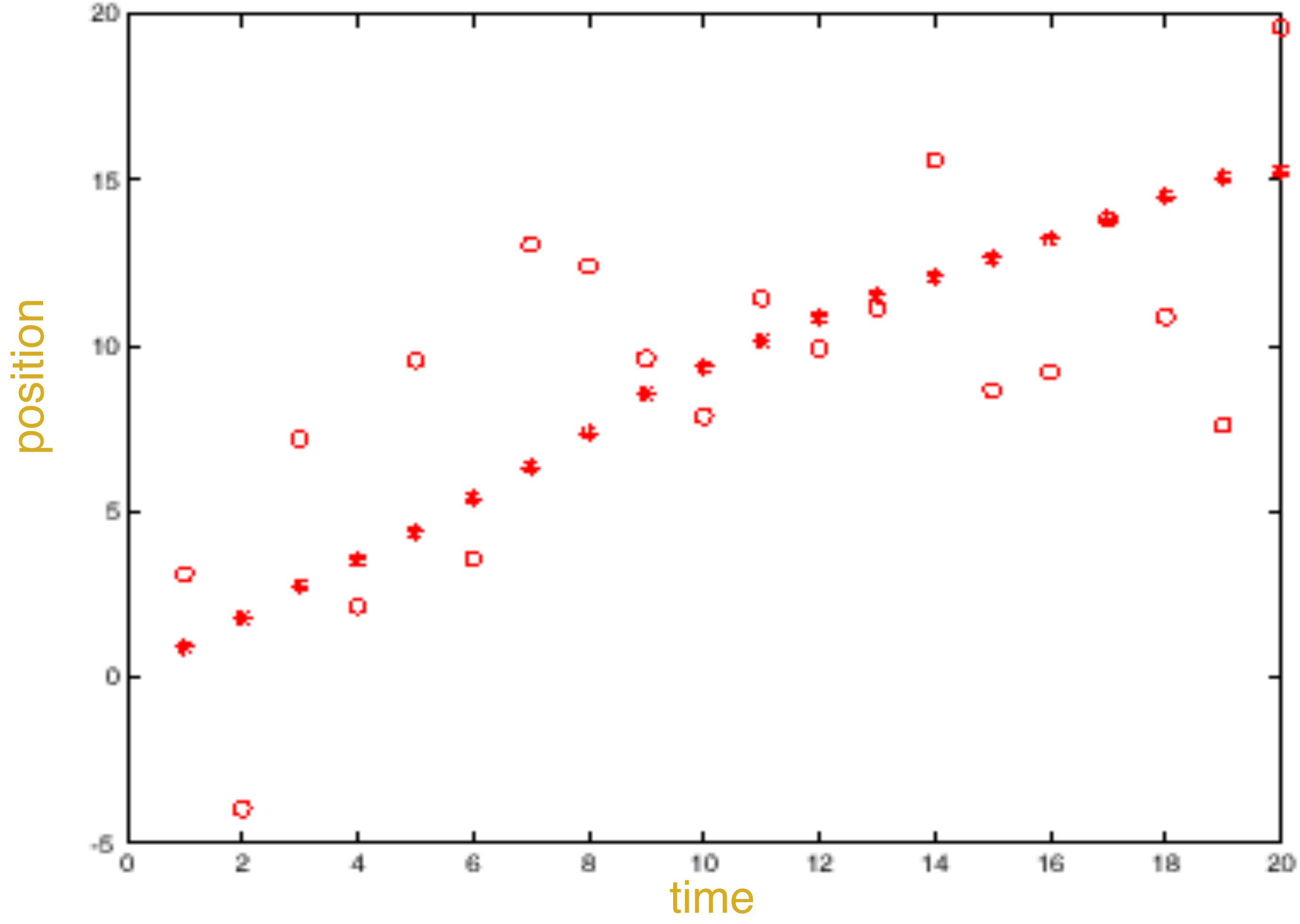
$$\Sigma_i^+ = \left[\boldsymbol{Id} - \mathcal{K}_i \mathcal{M}_i \right] \Sigma_i^-$$

Source: Darrell



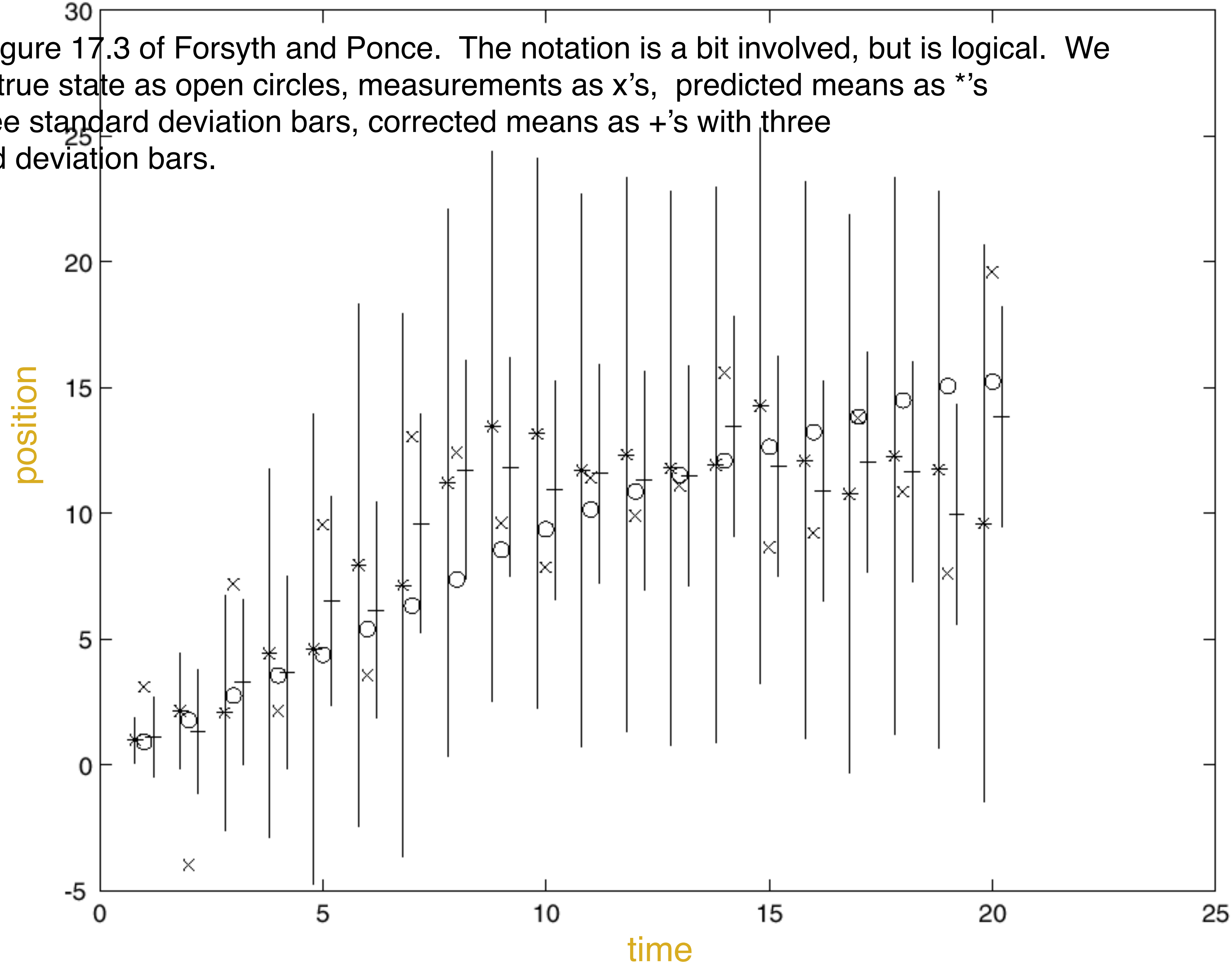
Constant Velocity Model

Source: Darrell

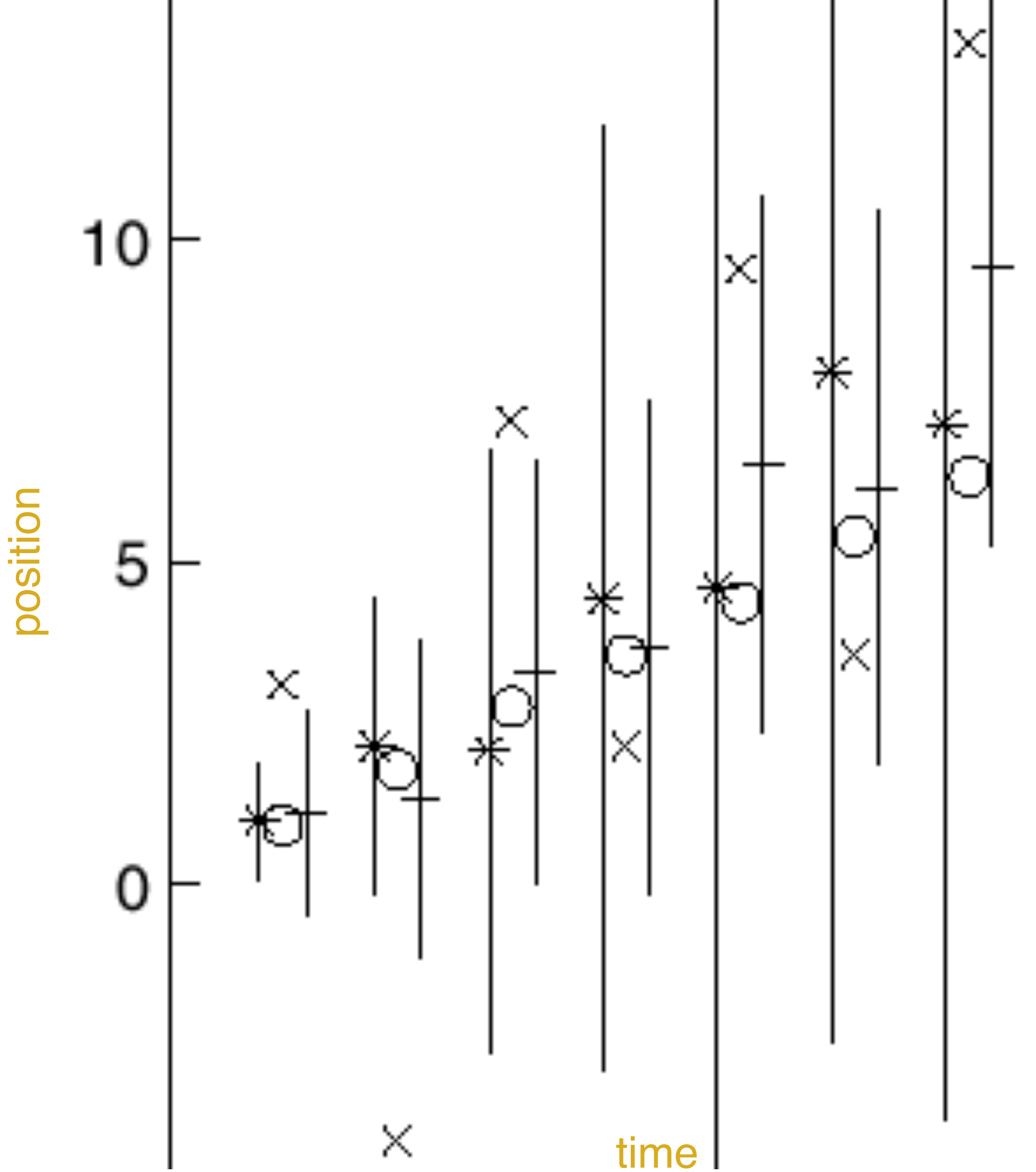


Source: Darrell

This is figure 17.3 of Forsyth and Ponce. The notation is a bit involved, but is logical. We plot the true state as open circles, measurements as x's, predicted means as *'s with three standard deviation bars, corrected means as +'s with three standard deviation bars.



The *-s give $\overline{\boldsymbol{x}}_i^-$, +-s give $\overline{\boldsymbol{x}}_i^+$, vertical bars are 3 standard deviation bars



The o-s give state, x-s measurement.

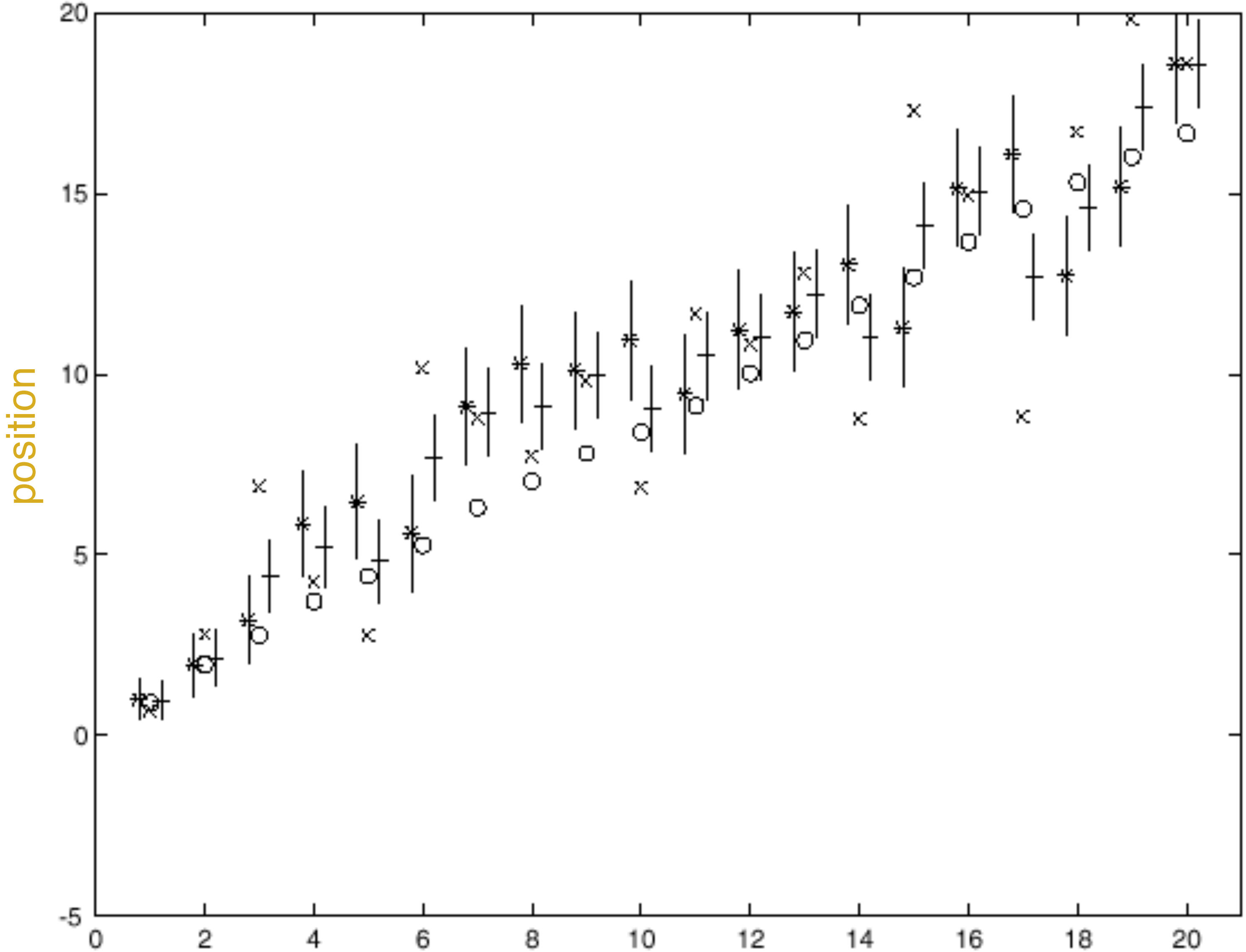
The $*$-s give $\overline{\boldsymbol{x}}_i^-$, $+$-s give $\overline{\boldsymbol{x}}_i^+$, vertical bars are 3 standard deviation bars

Smoothing

- Idea
  - We don't have the best estimate of state - what about the future?
  - Run two filters, one moving forward, the other backward in time.
  - Now combine state estimates
    - The crucial point here is that we can obtain a smoothed estimate by viewing the backward filter's prediction as yet another measurement for the forward filter

Source: Darrell

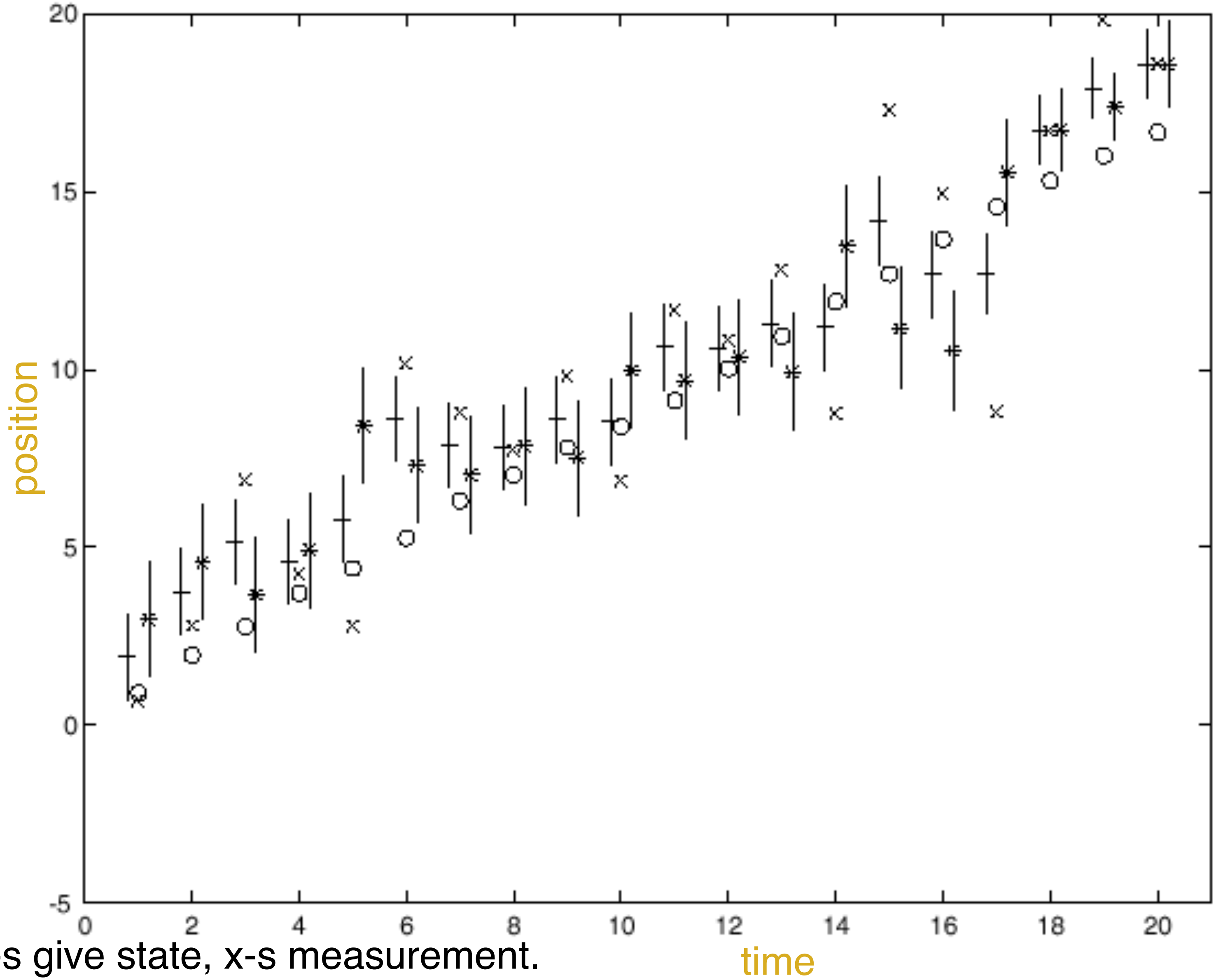
Forward estimates.



The o-s give state, x-s measurement.

The $*$-s give $\overline{\boldsymbol{x}}_i^-$, $+$-s give $\overline{\boldsymbol{x}}_i^+$, vertical bars are 3 standard deviation bars

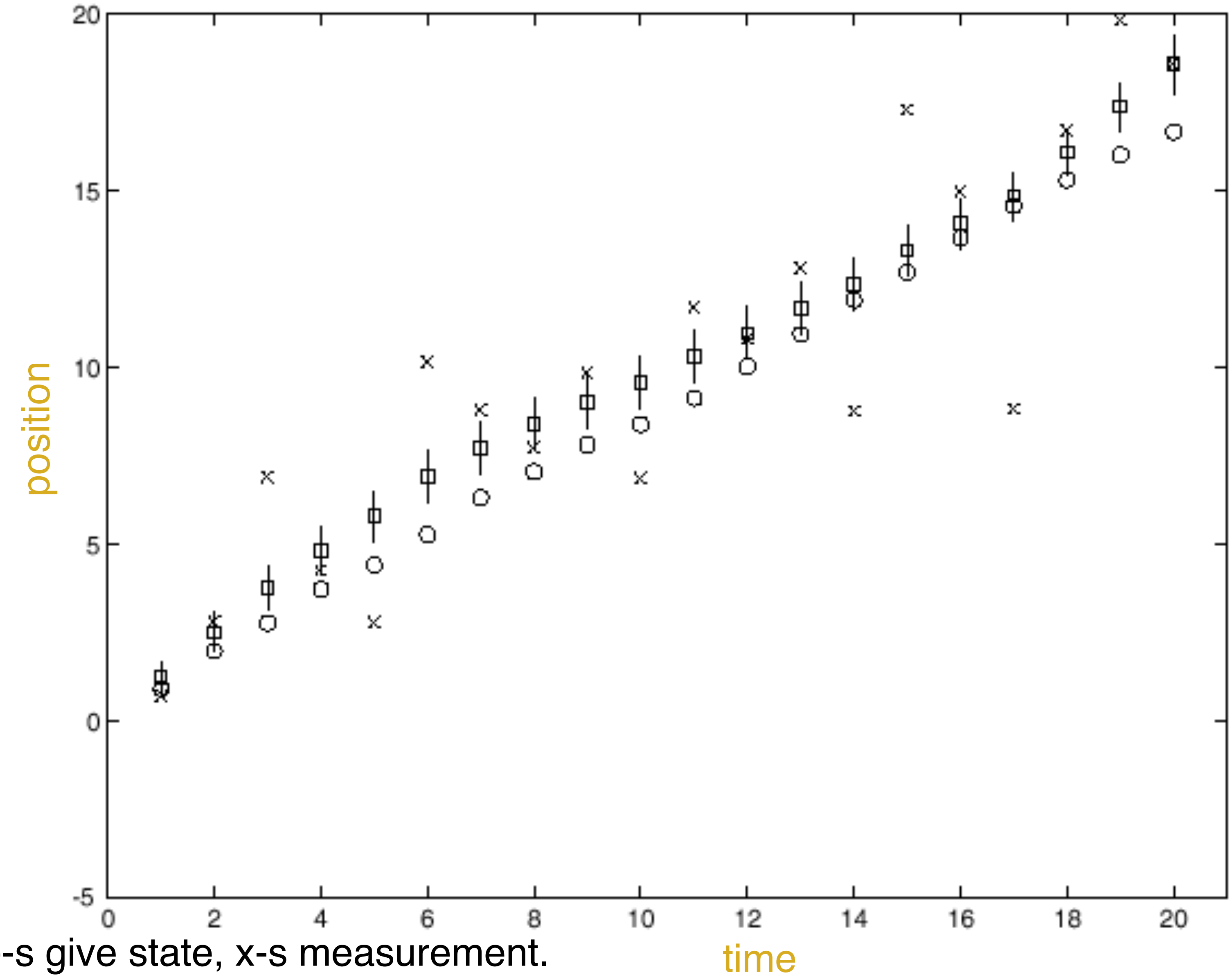
Backward estimates.



The o-s give state, x-s measurement.

The $*$-s give $\overline{\boldsymbol{x}}_i^-$, $+$-s give $\overline{\boldsymbol{x}}_i^+$, vertical bars are 3 standard deviation bars

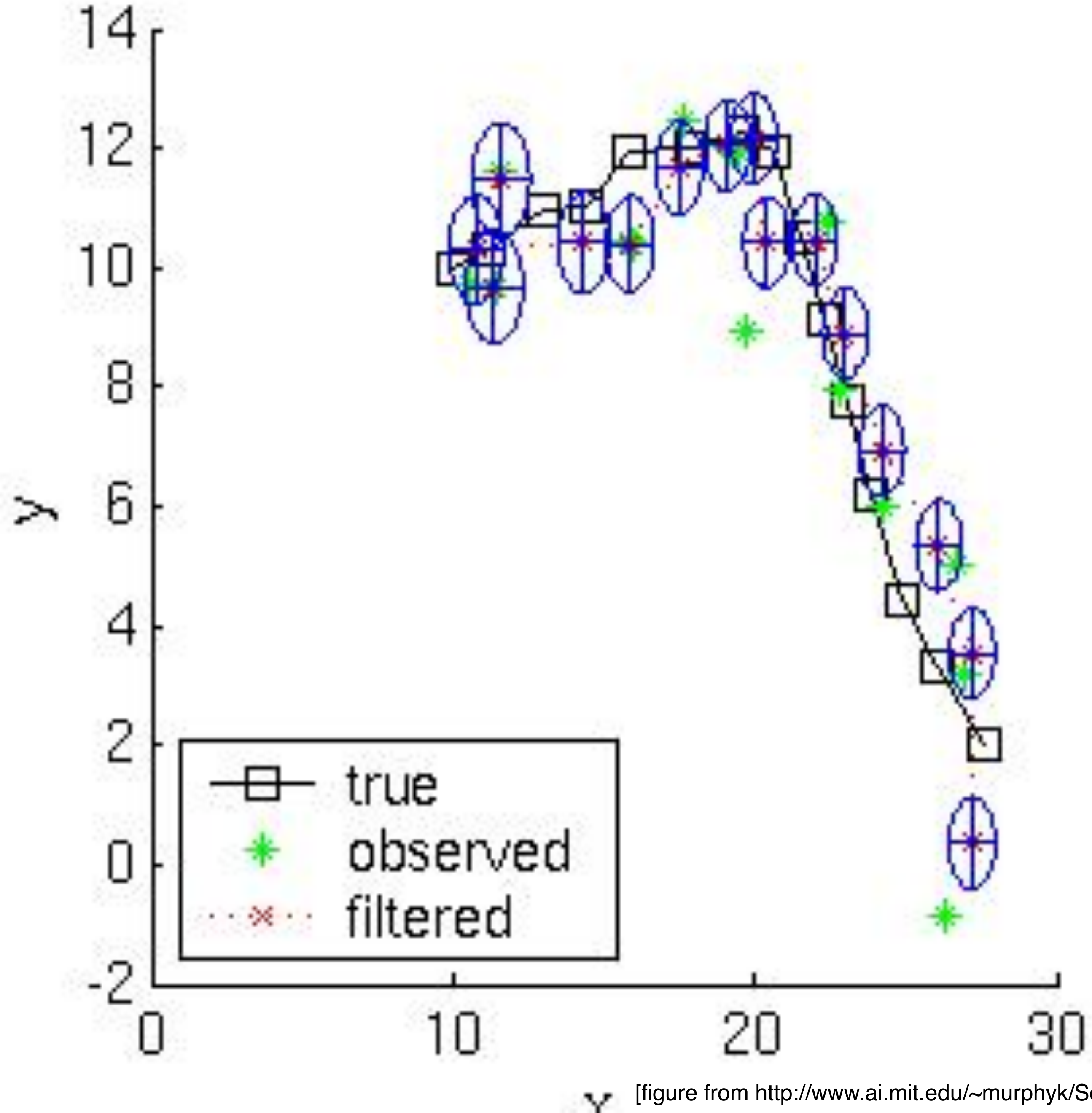
Combined forward-backward estimates.



The o-s give state, x-s measurement.

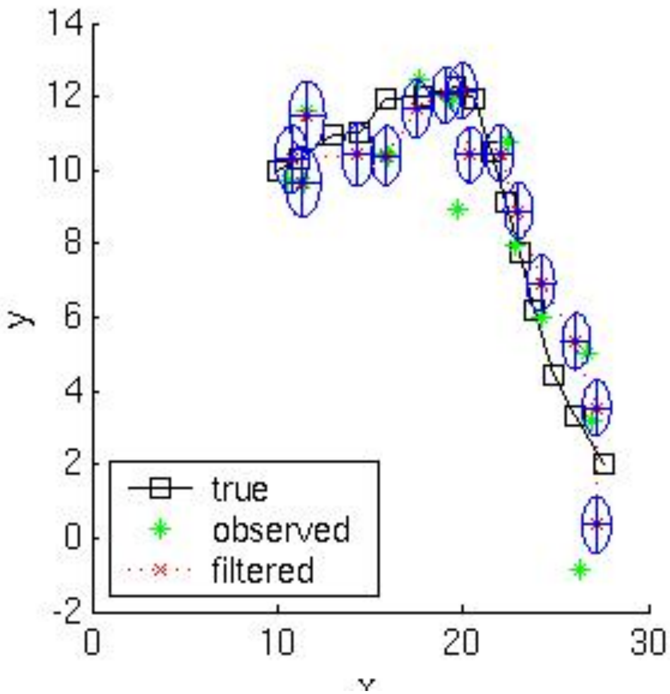
The *-s give $\overline{\boldsymbol{x}}_i^-$, +-s give $\overline{\boldsymbol{x}}_i^+$, vertical bars are 3 standard deviation bars

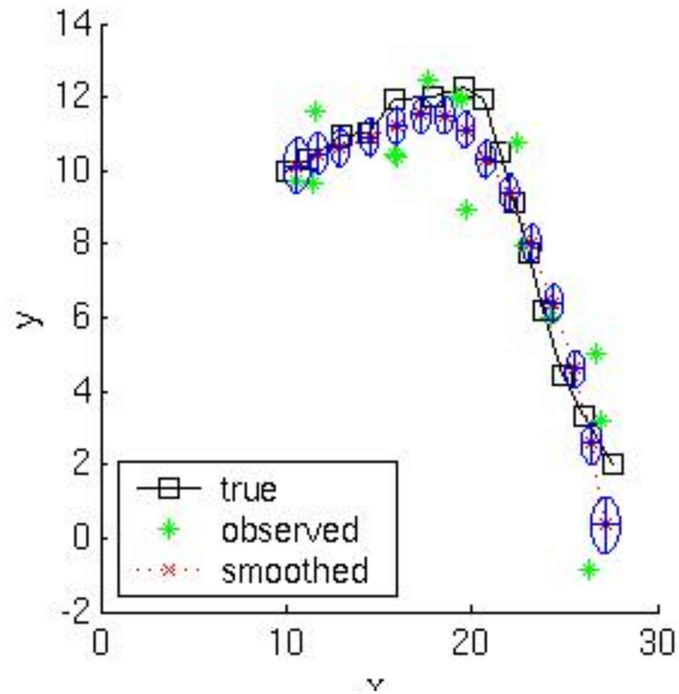
2-D constant velocity example from Kevin Murphy's Matlab toolbox



Source: Darrell

[figure from http://www.ai.mit.edu/~murphyk/Software/Kalman/kalman.html]





2-D constant velocity example from Kevin Murphy's Matlab toolbox

- MSE of filtered estimate is 4.9; of smoothed estimate. 3.2.
- Not only is the smoothed estimate better, but we know that it is better, as illustrated by the smaller uncertainty ellipses
- Note how the smoothed ellipses are larger at the ends, because these points have seen less data.
- Also, note how rapidly the filtered ellipses reach their steady-state ("Ricatti") values.

[figure from http://www.ai.mit.edu/~murphyk/Software/Kalman/kalman.html]

Source: Darrell

Linear Filtering Resources

- Kalman filter homepage

http://www.cs.unc.edu/~welch/kalman/

(kalman filter demo applet)

- Kevin Murphy's Matlab toolbox:

http://www.ai.mit.edu/~murphyk/Software/Kalman/kalman.html

Source: Darrell

Motion segmentation

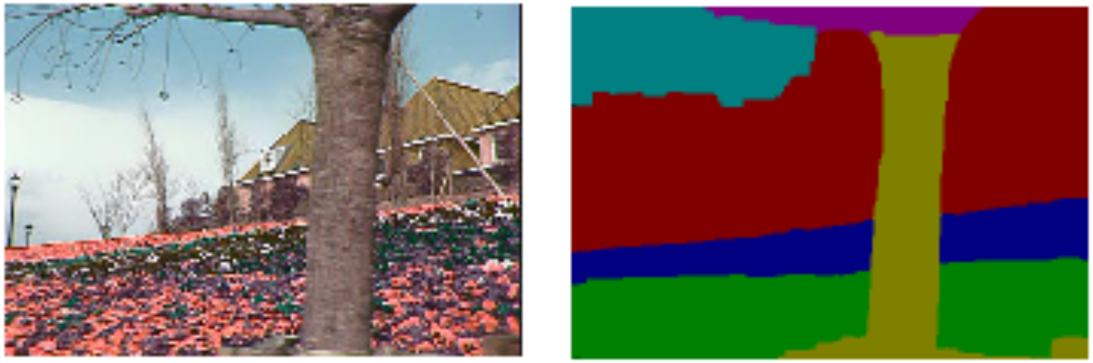
- How do we represent the motion in this scene?

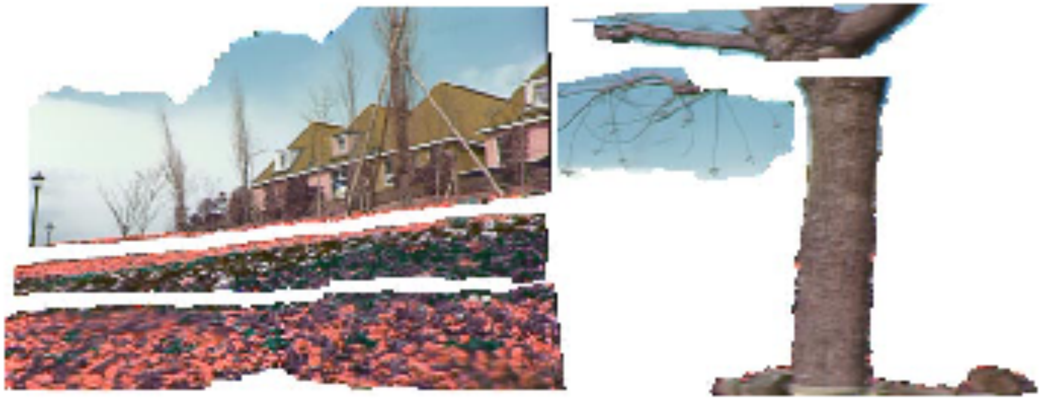
Source; Szeliski, Savarese

Motion segmentation

J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

- Break image sequence into “layers” each of which has a coherent (affine) motion

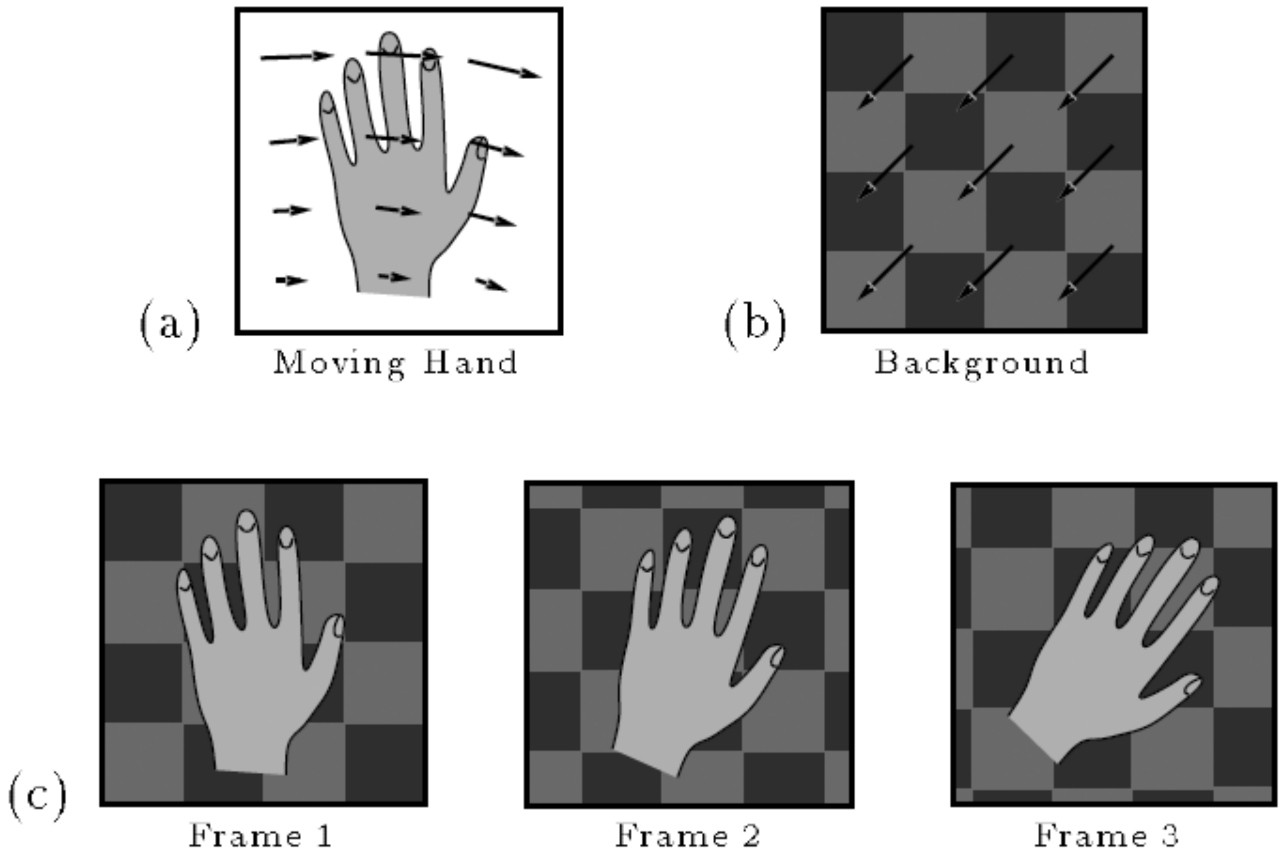




Source; Szeliski, Savarese

What are layers?

- Each layer is defined by an alpha mask and an affine motion model



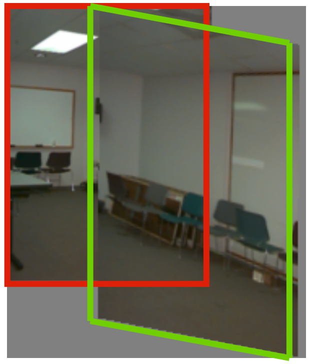
J. Wang and E. Adelson. [Layered Representation for Motion Analysis.](#) *CVPR 1993*.

Source; Szeliski, Savarese

Affine motion

$$u(x, y) = a_1 + a_2 x + a_3 y$$

$$v(x, y) = a_4 + a_5 x + a_6 y$$



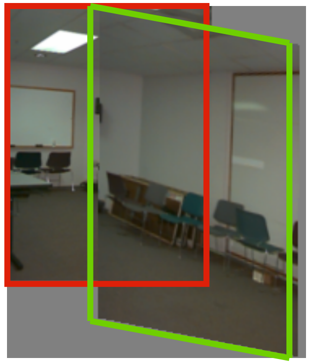
- Substituting into the brightness constancy equation:

$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0}$$

Source; Szeliski, Savarese

Affine motion

$$u(x, y) = a_1 + a_2 x + a_3 y$$
$$v(x, y) = a_4 + a_5 x + a_6 y$$



- Substituting into the brightness constancy equation:

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

- Each pixel provides 1 linear constraint in 6 unknowns
- If we have at least 6 pixels in a neighborhood, *a_1… a_6* can be found by least squares minimization:

$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

Source; Szeliski, Savarese

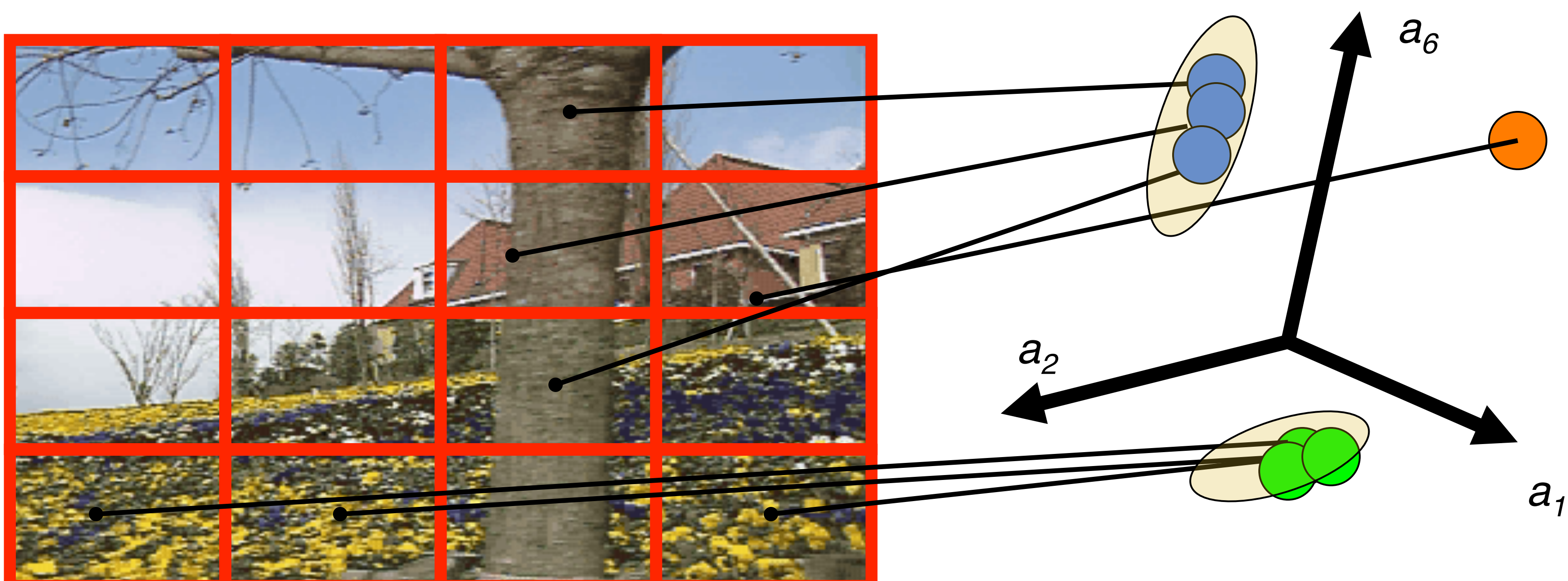
How do we estimate the layers?

- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error

2. Map into motion parameter space

3. Perform k-means clustering on affine motion parameters

   –Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene



Source: Szeliski, Savarese

How do we estimate the layers?

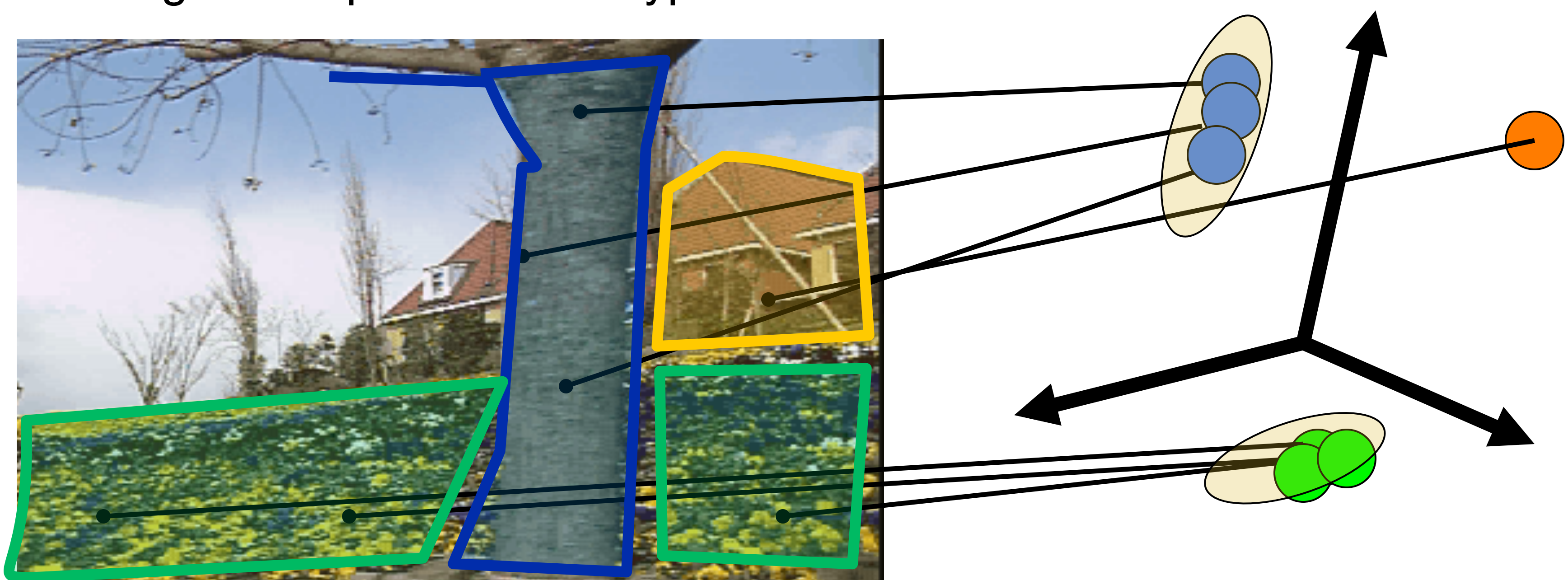
- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error

2. Map into motion parameter space

3. Perform k-means clustering on affine motion parameters

   –Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene

4. Assign each pixel to best hypothesis --- iterate



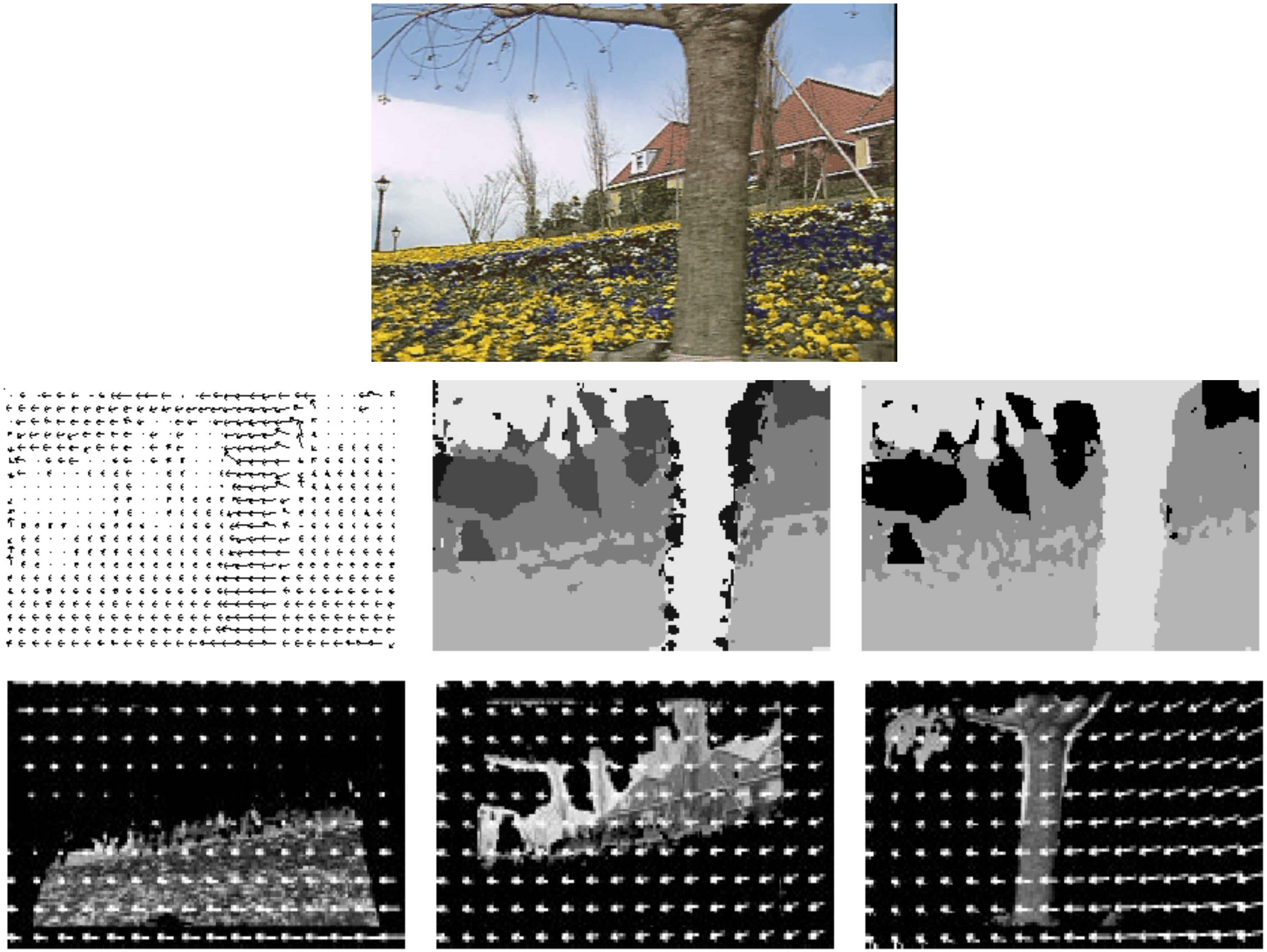
Source; Szeliski, Savarese

How do we estimate the layers?

1. Obtain a set of initial affine motion hypotheses
   - Divide the image into blocks and estimate affine motion parameters in each block by least squares
     - Eliminate hypotheses with high residual error
   - Map into motion parameter space
   - Perform k-means clustering on affine motion parameters
     - Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene
2. Iterate until convergence:
   - Assign each pixel to best hypothesis
     - Pixels with high residual error remain unassigned
   - Perform region filtering to enforce spatial constraints
   - Re-estimate affine motions in each region

Source; Szeliski, Savarese

Example result



J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

Source; Szeliski, Savarese

Motion and Tracking in Omnidirectional Video

Next Lecture: Structure from Motion

- Readings: FP 8; SZ 7