Human Computer Interaction

4. Fundamental of Vision-based Interaction (A)

National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

Goals

- Learn the basic feature extraction from vision tech.
- Efficient (and simple ?) approaches for real-time user interfaces.

Outline

- Feature extraction
  - Color matching
  - Grouping or clustering
  - Silhouette & foreground
- Motion tracking
  - Filtering and prediction
- 3D position estimation
  - Two views
  - mirrored views

Vision-based Interface

- Pointing
- gesture recognition
- Eye tracking
- … … …

Hand Gesture Recognition, Caltech

Interactive Wall, CSAIL, MIT

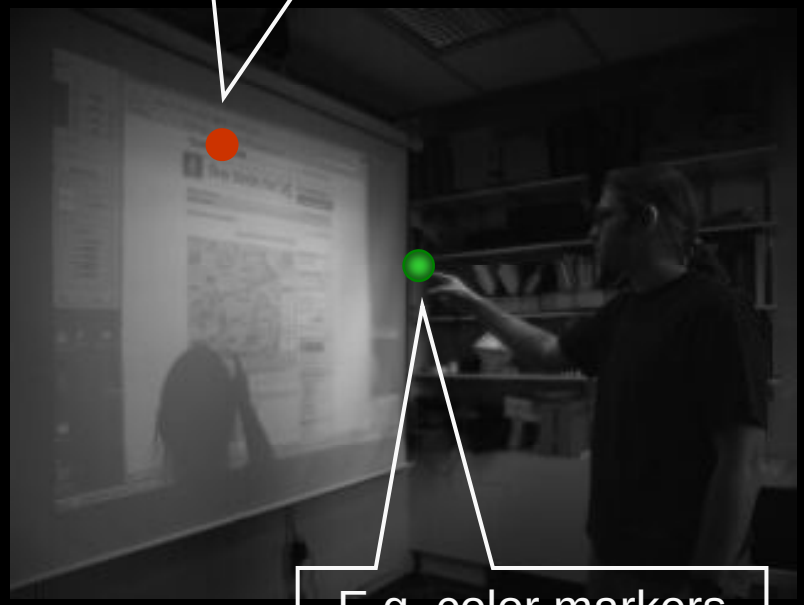
Vision-based Interface (cont.)

- A lot of delicate and advanced vision tech. for various targets.
- Our goals
  - Not focusing on developing brand-new algorithms.
  - Utilizing efficient and simple algorithms for friendly interfaces and attractive applications.

How about a simpler environment or condition?

Make the thing easier !!!

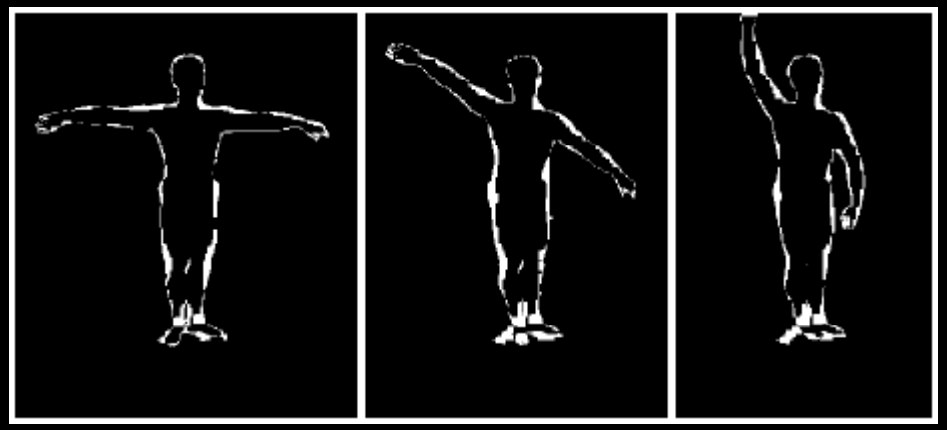
Vision-based Interface (cont.)

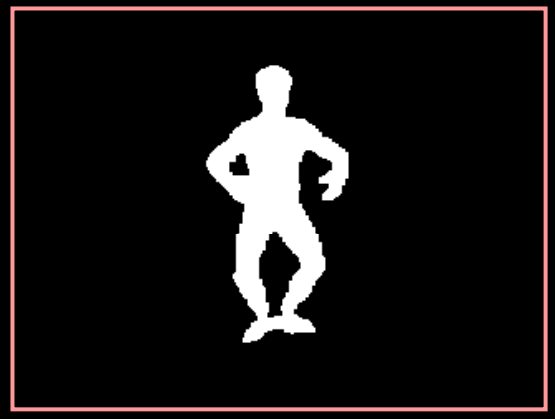


Blue screen, Star War III

conspicuous color features

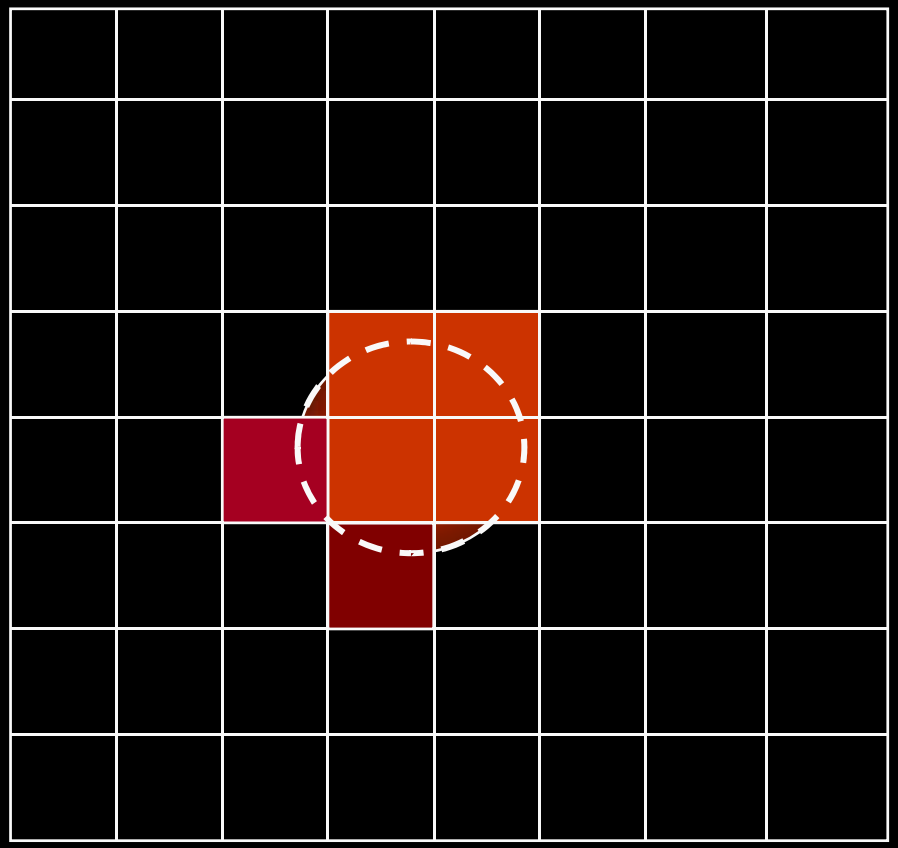
Vision-based Interface (cont.)





Silhouette images, from J. Carranza et al., “Free-Viewpoint Video of Human Actors”

Color Features



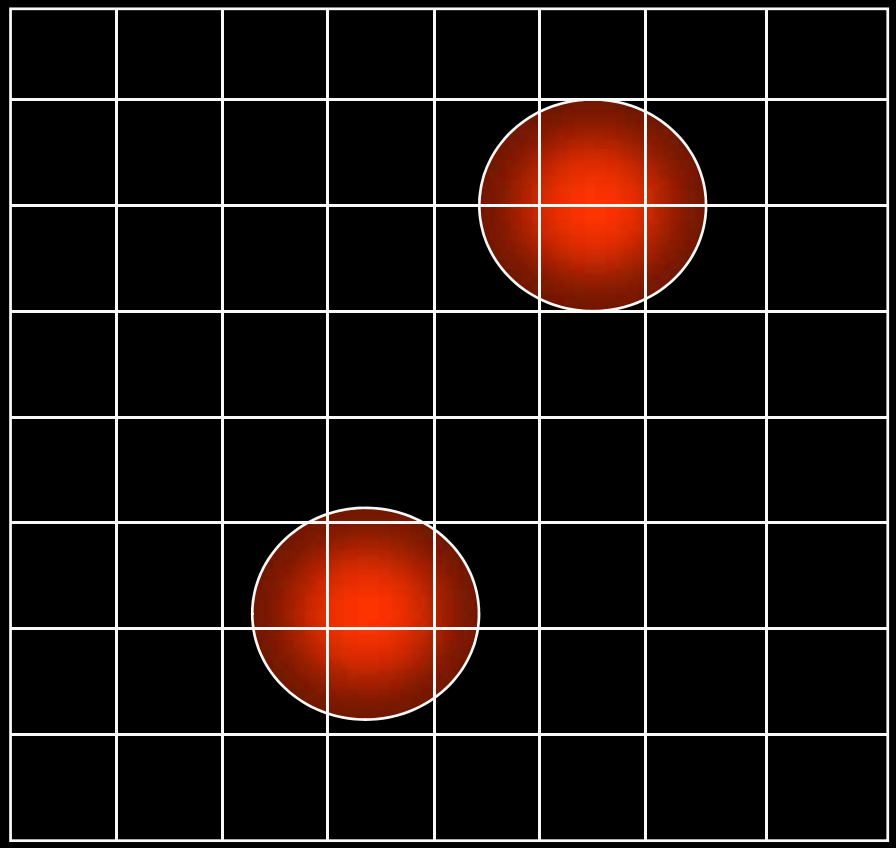
- RGB image array
- Digitization Effects
- Noise, etc.

- How to assign your target colors?
  - predefined in your system ?
  - Initialization stages ?

- How to estimate the target position ?

A constrained condition will make your computation easier and more reliable.

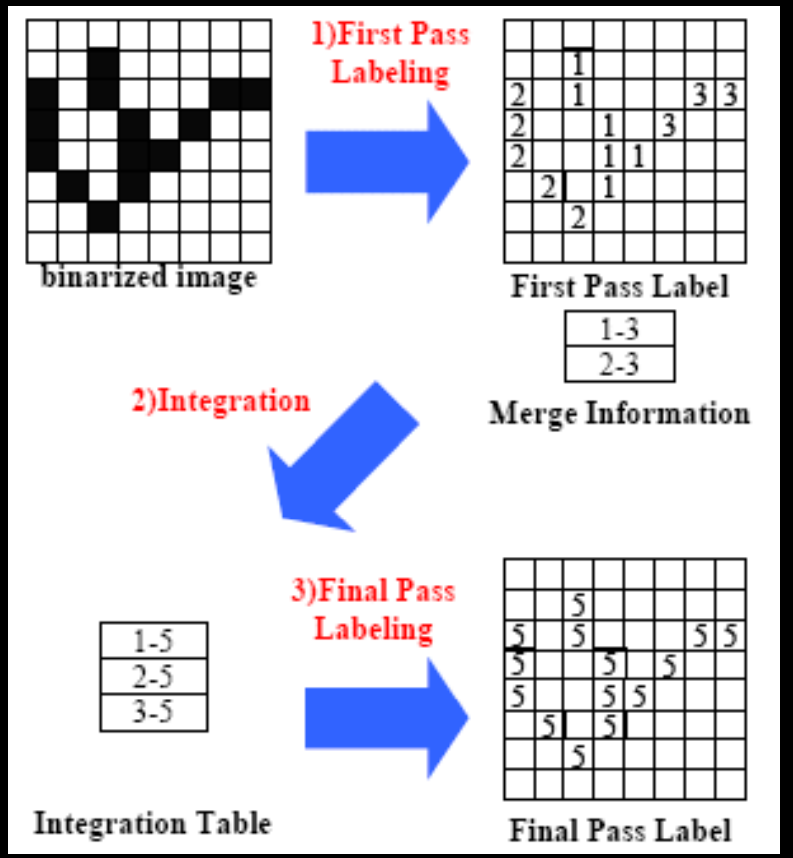
Color Features (cont.)



- How about two (or more) targets ?
  - Number of objects (with or without)
  - Connected component labeling
  - Noise effects

Connected Component Labeling

- 4-neighbor or 8 neighbor.
- Give a new labeling number to those without labeled neighbors.
- Merge the label numbers (equivalent classes).
- You can easily find classic algorithms.



http://www.design-reuse.com/news_img/20060216_ipflex2.gif

Clustering

- *K-means* algorithms are popularly used.

$$E = \sum_{j=1}^{K} \sum_{i=1}^{n} \left\| x_i^{(j)} - m_j \right\|^2$$

Make initial guesses for the means m_1, m_2, ..., m_k

Until there are no changes in any mean

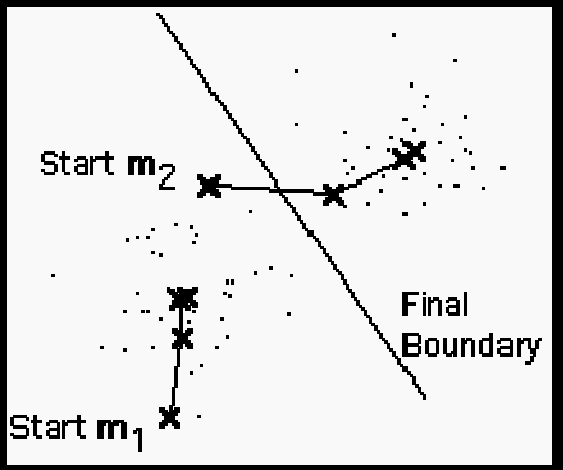
    Use the estimated means to classify the samples into clusters

    For i from 1 to k

        Replace m_i with the mean of all of the samples for cluster i

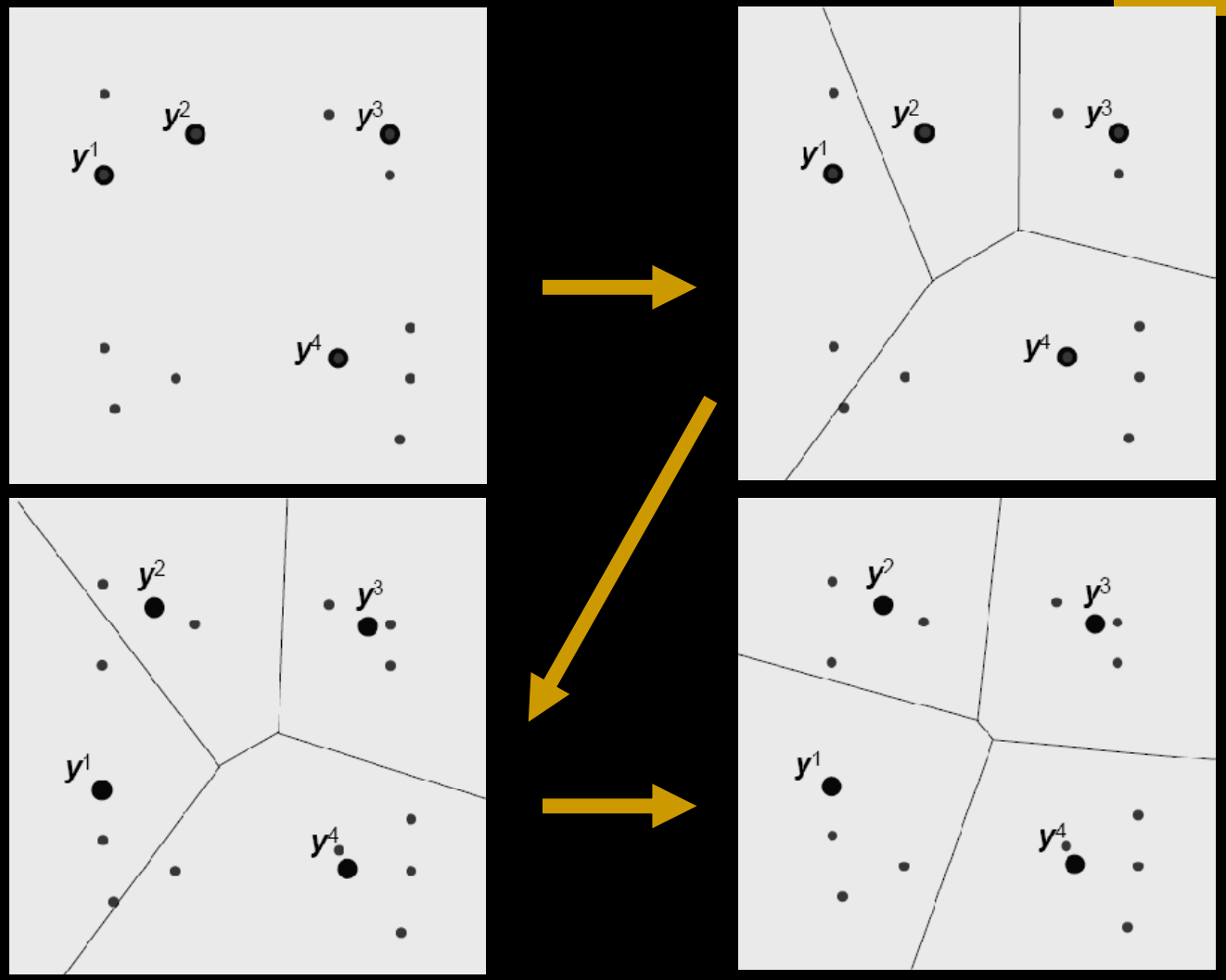
    end_for

end_until



http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/kmeans.html

Lloyd's algorithm



www.dice.ucl.ac.be/~verleyse/lectures/elec2870

Color Features

Eyetoy games, PS2

Feature extractions

- How about these applications?

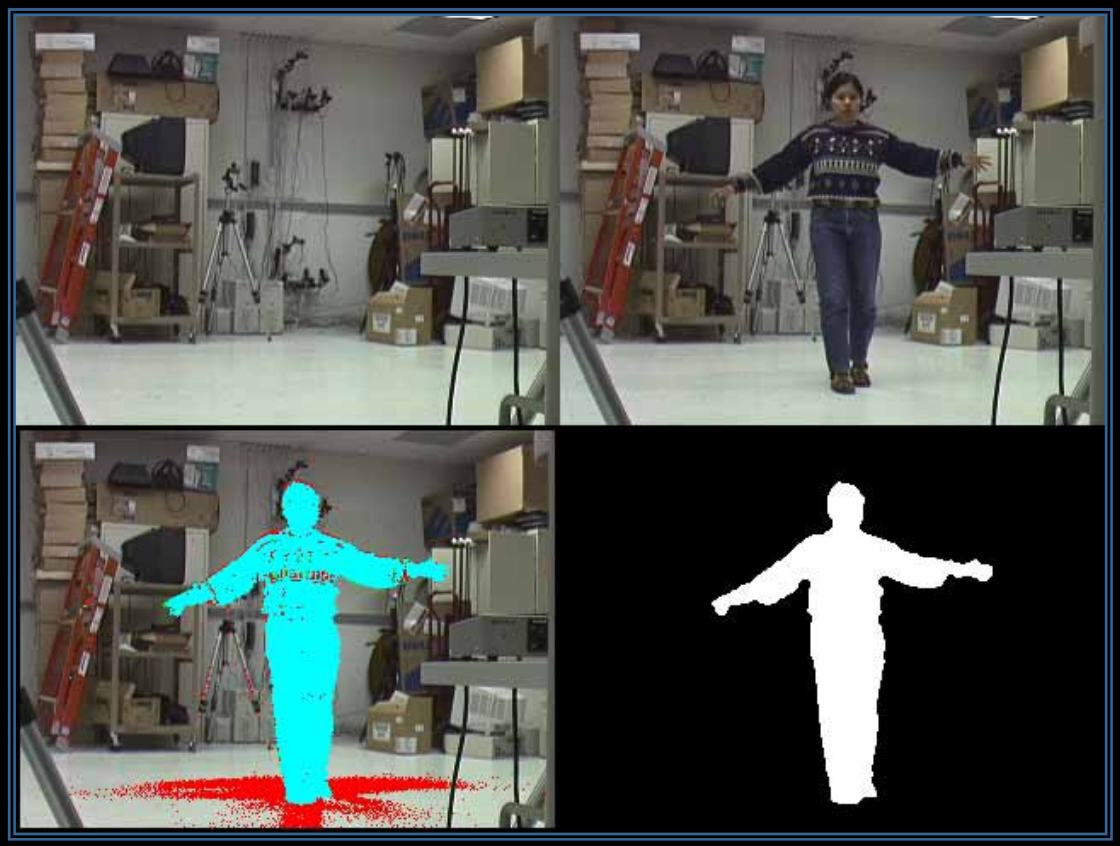
Eyetoy games, PS2

Foreground Extraction

- Under an environment with fixed lighting and static scenes, the detection of new objects can be more reliable.

Foreground Extraction (cont.)

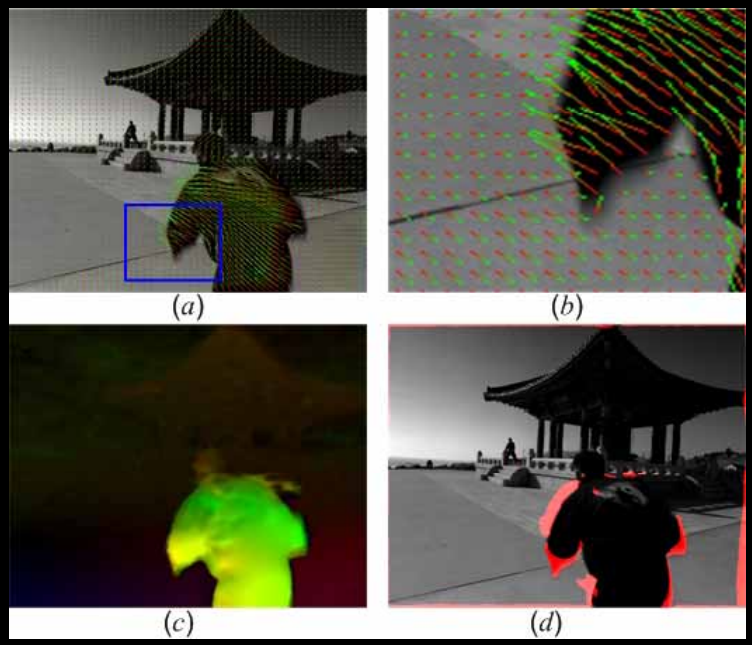
- Noise should be taken into account …
  - Noise suppression



T. Horprasert et al., Vision lab, UMD

Moving Object Extraction

- Thresholding on Intensity difference
- Optical flow



http://serdis.dis.ulpgc.es/~jsanchez/research/demos/opticalFlow

http://www.cs.ucf.edu/~jxiao/images/kungfu.jpg

Motion Tracking

- How to distinguish multiple objects or targets in a scene ?

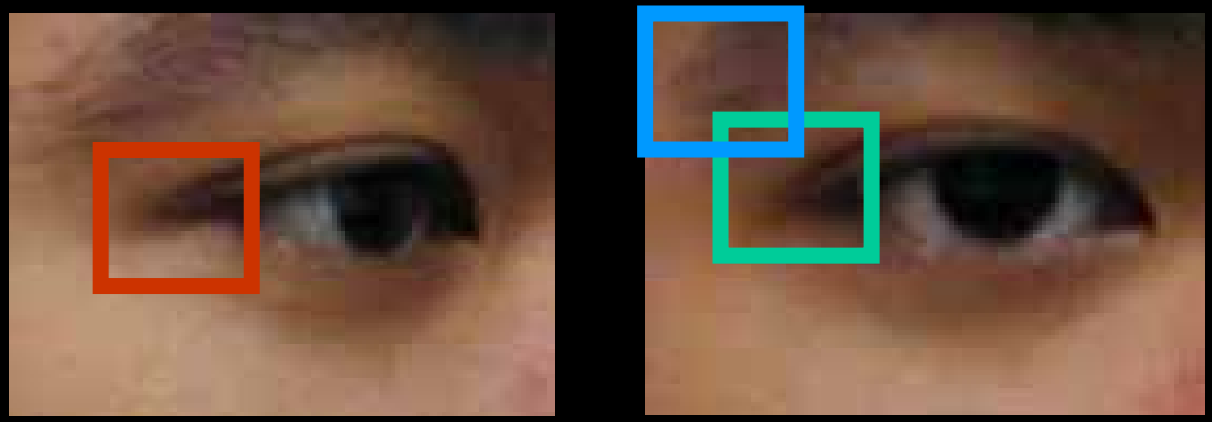
http://www.site.uottawa.ca/~laganier/surveillance/

- The problem of occlusion ?

Feature Matching (Correlation)

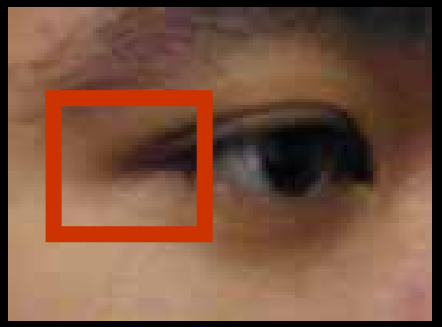
- Find corresponding points in image video sequence
  - one simple technique: find two patches with minimal summed squared error.

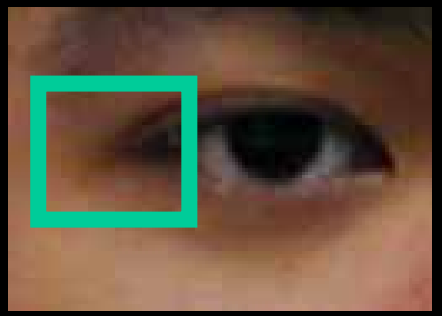
$$E_{x,y}(u,v) = \sum_{k=x-w}^{x+w} \sum_{l=y-w}^{y+w} [I_1(k+u, l+v) - I_0(k,l)]^2$$



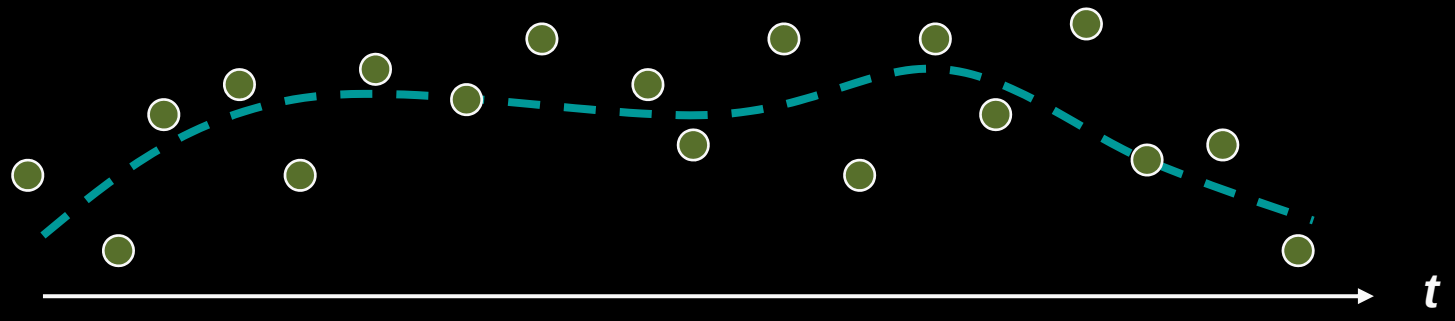
Feature Matching (cont.)

- Block matching can be unreliable due to:
  - Noise
  - Occlusion
  - Projection of different view directions
  - Reflection of specular lights, etc.
  - Deformation of objects
  - ……
- Other approaches: optic flow, etc.



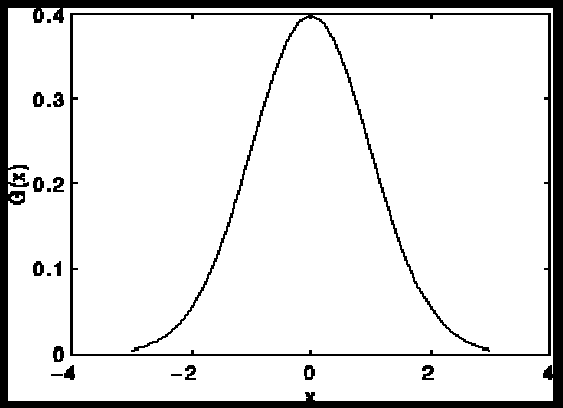


Filtering



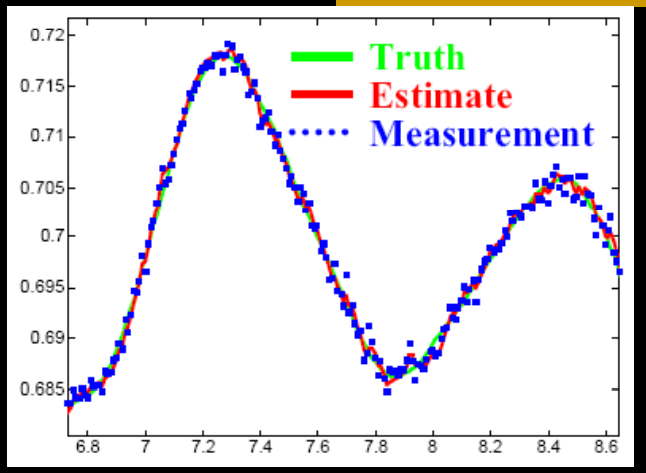
- Simply filtering by Gaussian filters.

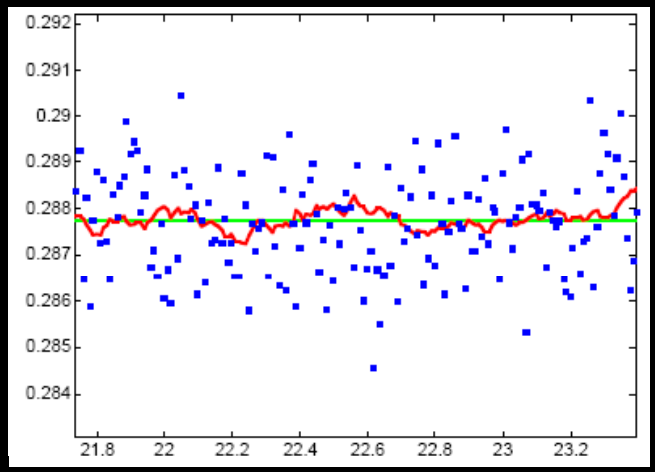
$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



Kalman Filtering

- Just some applied math.
- A linear system: f(a+b) = f(a) + f(b).
- Noisy data in→ hopefully less noisy out.
- But delay is the price for filtering…





Ref:

- G. Welch, G. Bishop, SIGGRAPH 2001 course notes "An Introduction to Kalman Filter"
- http://www.cs.unc.edu/~welch/kalman/
- S.M. Bozic, Digital and Kalman Filtering, Edward Arnold

Kalman Filtering (cont.)

- What is it used for ?
  - Tracking missiles
  - Tracking heads/hands/drumsticks/…
  - Extracting lip motion from video
  - Fitting Bezier patches to point data
  - Lots of computer vision applications
  - Economics
  - Navigation
  - ……..

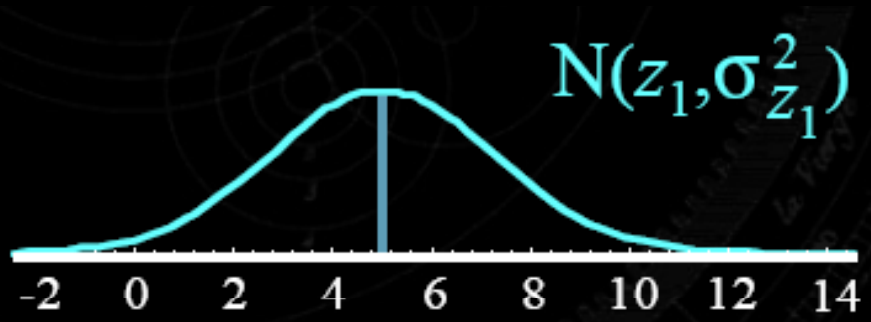
Measurement & State

- Measurement 1

$$z_1, \sigma_{z_1}^2$$

- State

$$\hat{x}_1 = z_1$$
$$\sigma_{x_1}^2 = \sigma_{z_1}^2$$

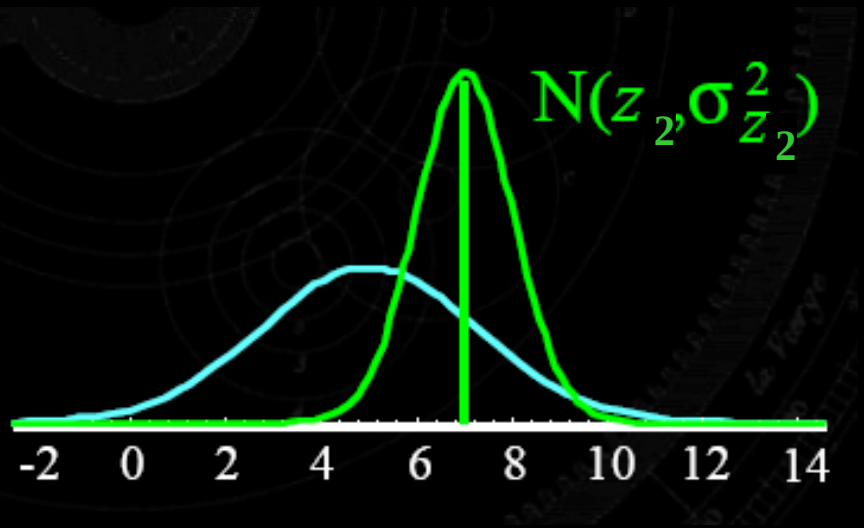


- Measurement 2

$$z_2, \sigma_{z_2}^2$$

- State ?

$$\hat{x}_2 = ?$$
$$\sigma_{x_2}^2 = ?$$



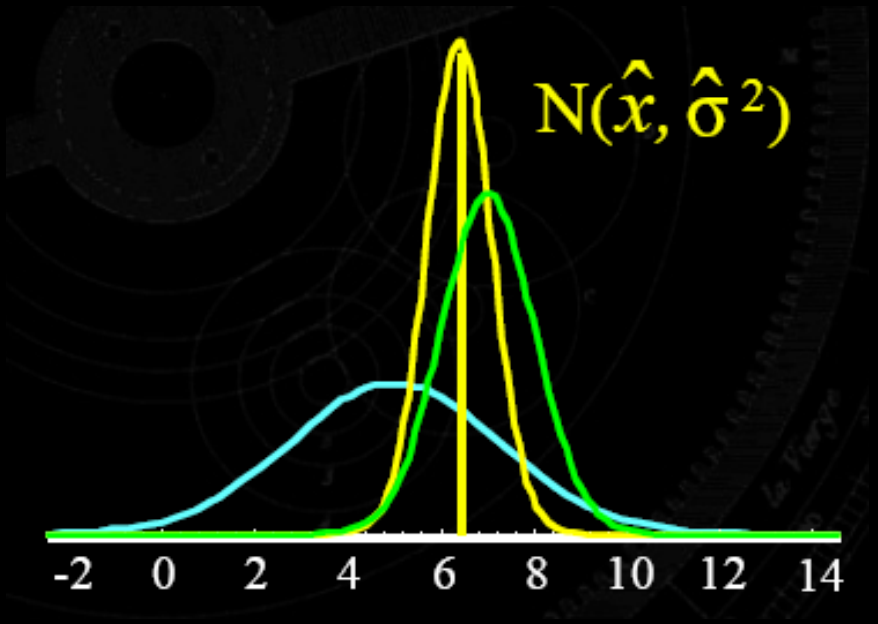
Combine the measurements

- The 2nd estimation (state) from measurement 1 + 2:

$$\hat{x}_2 = \hat{x}_1 + K_2\left(z_2 - \hat{x}_1\right)$$

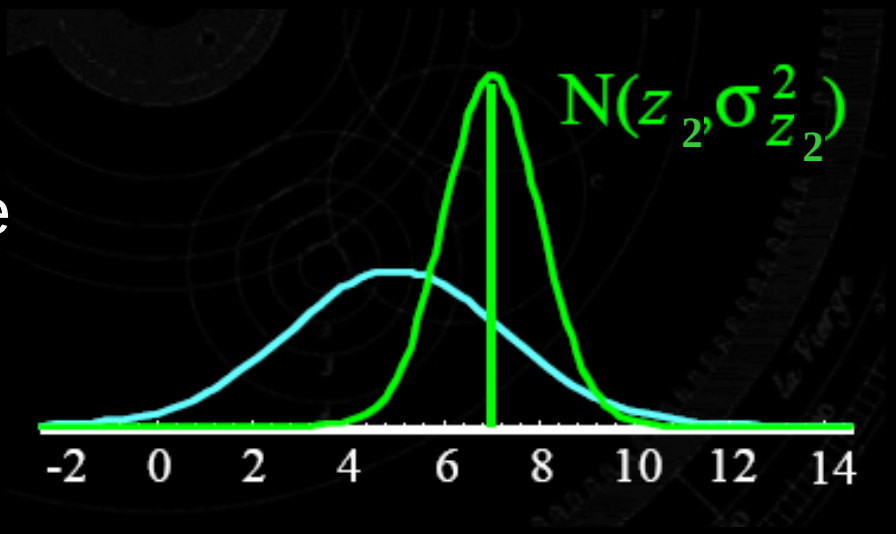
$$K_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_{z_2}^2}$$

$$\frac{1}{\sigma_2^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_{z_2}^2}$$



Suppose There're Motions ...

- Not all the difference is error
- Some may be motion
- Kalman filter can include a motion model
- Estimate velocity and position

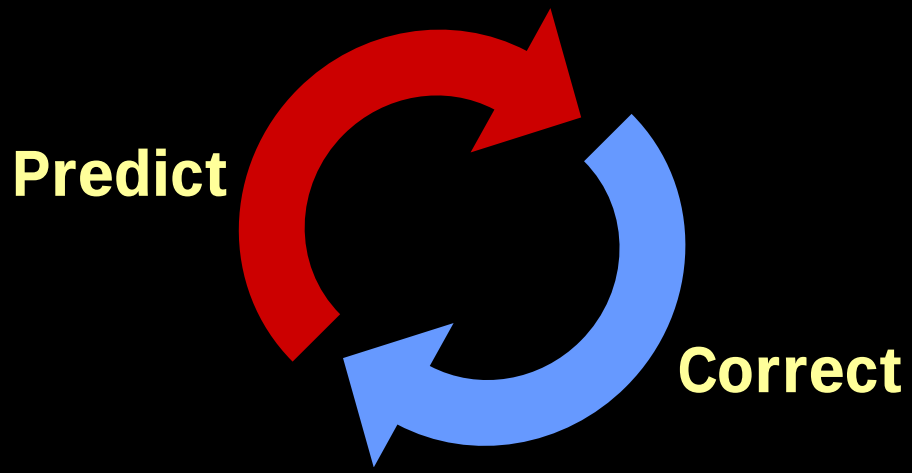


Process Model

- Describes how *state* changes over time
- The *state* for the first example was scalar
  - The *process model* was “nothing changes”
- A better model might be
  - State is a 2-vector [position, velocity]
  - $\text{position}_{n+1} = \text{position}_n + \text{velocity}_n * \text{time}$
  - $\text{velocity}_{n+1} = \text{velocity}_n$

From Prediction to Correction

- KF operates by
  - Predicting the new state and its uncertainty
  - Correcting with the new measurement



Kalman Filter

When the state equations are

$$x(n) = Ax(n-1) + w(n-1)$$
$$z(n) = Cx(n) + v(n)$$

Estimator:

$$\hat{x}(n) = A\hat{x}(n-1) + K(n)\left[z(n) - CA\hat{x}(k-1)\right]$$

Filter gain:

$$K(n) = P_p(n)C^T\left[CP_p(n)C^T + R(n)\right]^{-1}$$
$$\text{where } P_p(n) = AP(n-1)A^T + Q(n-1)$$

Error covariance matrix

$$P(n) = P_p(n) - K(n)C(n)P_p(n)$$

Example: 2D Pos. Only

- Apparatus: 2D Tablet

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} w_1(n-1) \\ w_2(n-1) \end{bmatrix}$$

x(n) A **x(n-1)** **w(n-1)**

$$\begin{bmatrix} z_1(n) \\ z_2(n) \end{bmatrix} = \begin{bmatrix} C_{11} & 0 \\ 0 & C_{22} \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} v_1(n) \\ v_2(n) \end{bmatrix}$$

z(n) C **x(n)** **v(n)**

X: internal state

(e.g. at screen coordinate)

A: state transition matrix

Z: measurement

(e.g. at tablet coordinate)

C: measurement matrix

W: noise

V: noise

Preparation

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

State transition

$$Q = E\{ww^T\} = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix}$$

Process Noise Covariance

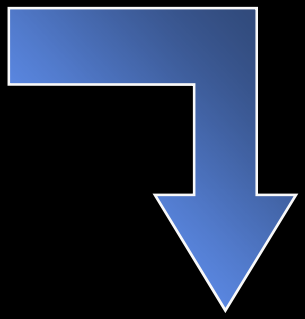
$$R = E\{vv^T\} = \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix}$$

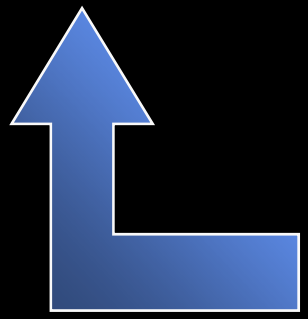
Measurement Noise Covariance

Prediction & Correction

$$\hat{x}_p = A\hat{x}(n-1)$$

$$P_p(n) = AP(n-1)A^T + Q(n-1)$$



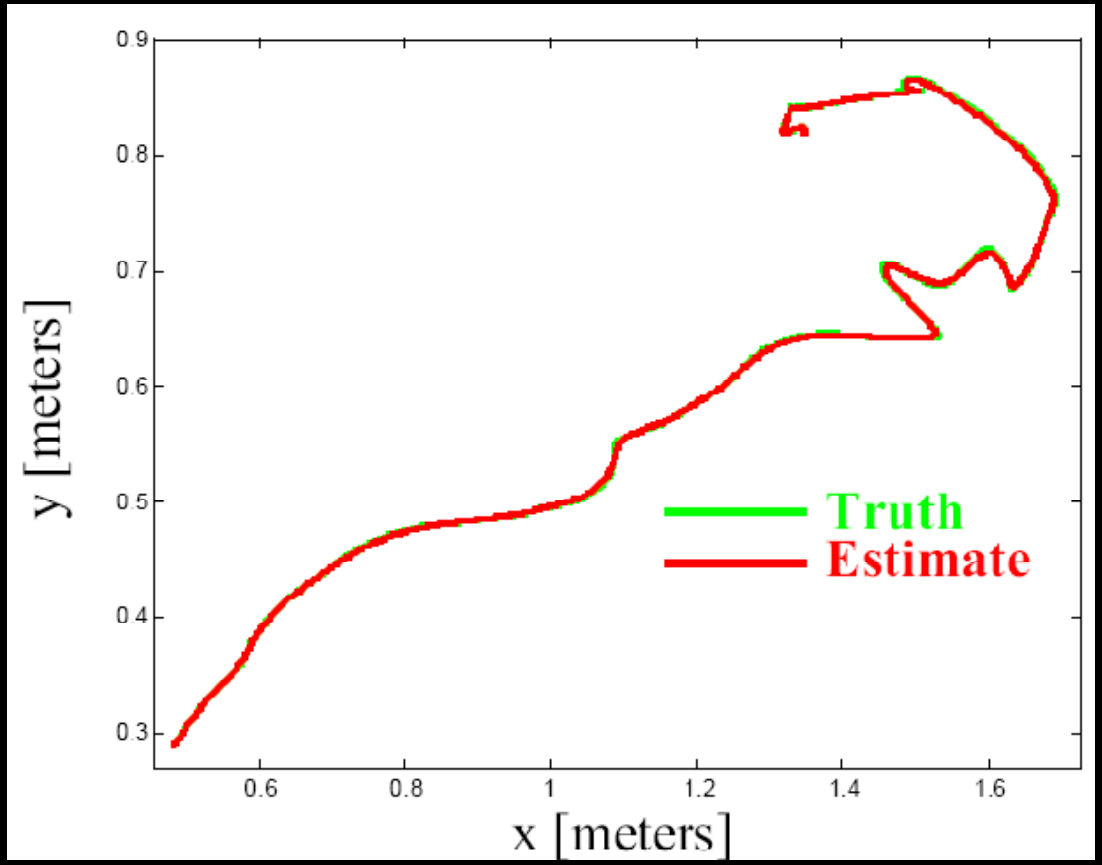


$$K(n) = P_p(n)C^T\left[CP_p(n)C^T + R(n)\right]^{-1}$$

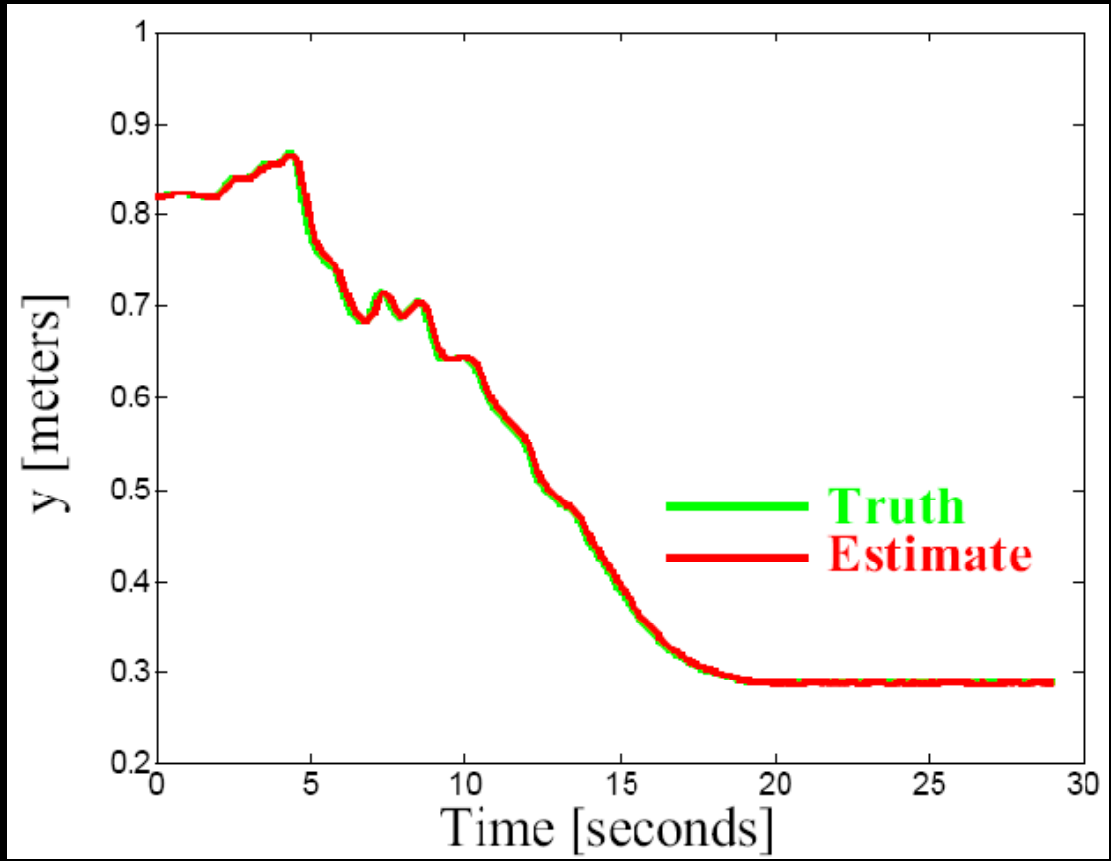
$$\hat{x}(n) = \hat{x}_p(n) + K(n)\left[z(n) - C\hat{x}_p(n)\right]$$

$$P(n) = P_p(n) - K(n)C(n)P_p(n)$$

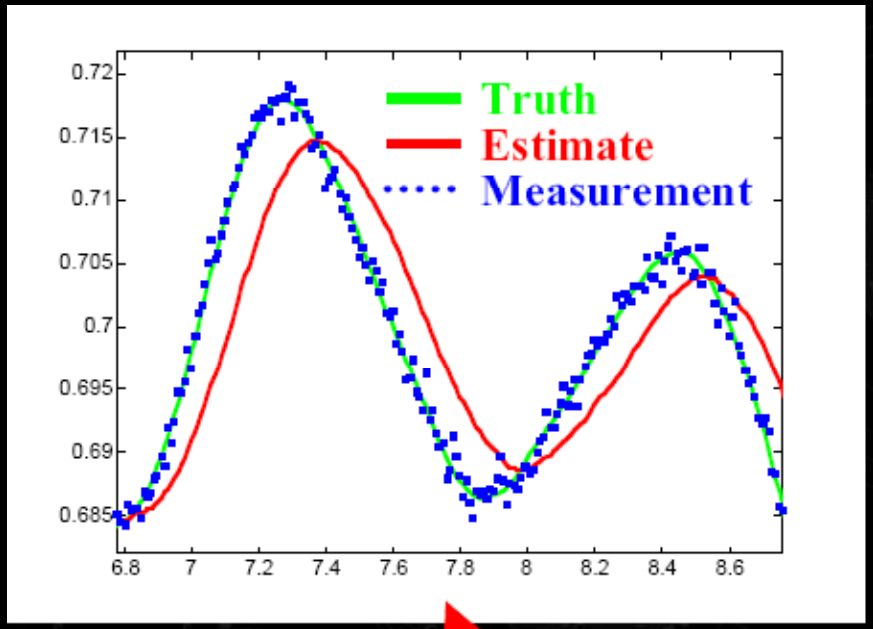
XY Track

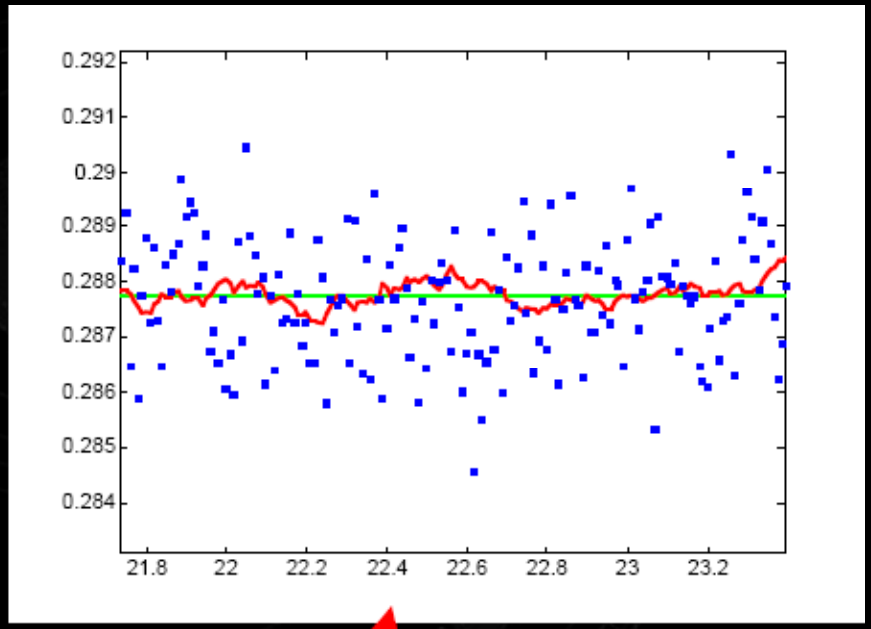

0.9
0.8
0.7
0.6
0.5
0.4
0.3
y [meters]
0.6
0.8
1
1.2
1.4
1.6
x [meters]
Truth
Estimate

Y Track: Moving then Still

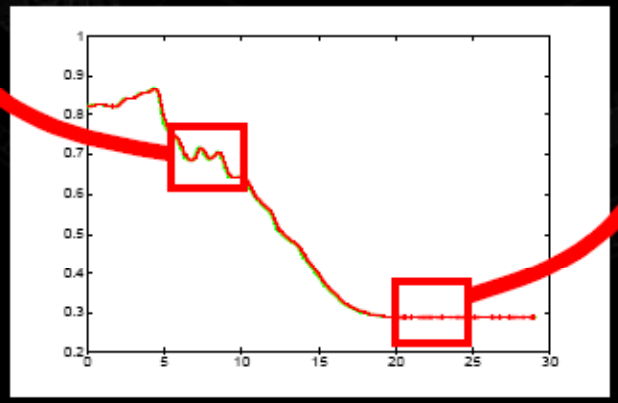


Motion-dependent Performance





significant *latency* when moving…



…relatively *smooth* when not

Example: 2D PV Model

- Position-velocity model

$$\begin{bmatrix} x_1(n) \\ x_2(n) \\ \dot{x}_1(n) \\ \dot{x}_2(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \\ \dot{x}_1(n-1) \\ \dot{x}_2(n-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_1(n-1) \\ u_2(n-1) \end{bmatrix}$$

$$\begin{bmatrix} z_1(n) \\ z_2(n) \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \\ \dot{x}_1(n) \\ \dot{x}_2(n) \end{bmatrix} + \begin{bmatrix} v_1(n) \\ v_2(n) \end{bmatrix}$$

u(n): change in velocity

v(n): measurement error

Preparation & Initialization

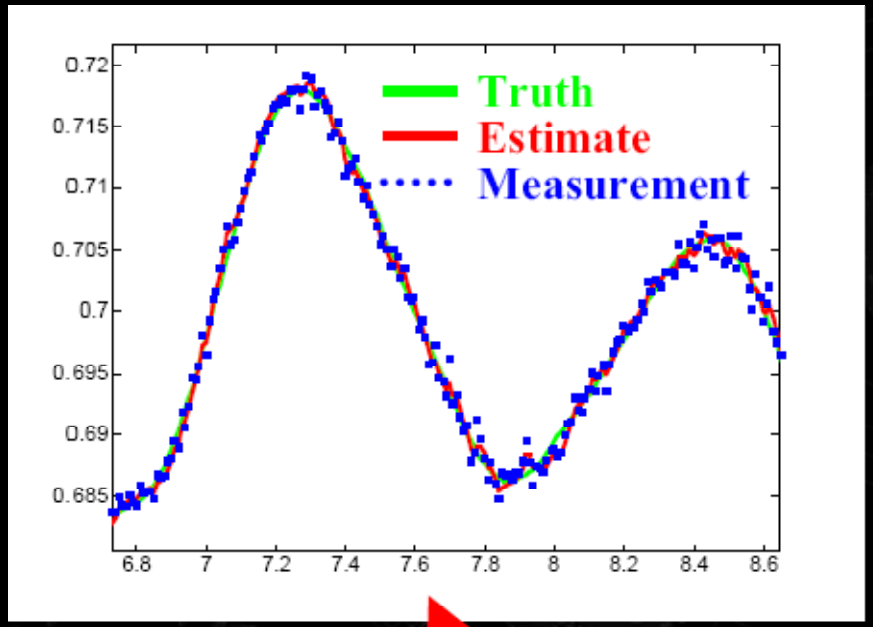
$$R(n) = E\left[v(n)v(n)^T\right] = \begin{bmatrix} \sigma_{v1}^2(n) & 0 \\ 0 & \sigma_{v2}^2(n) \end{bmatrix}$$

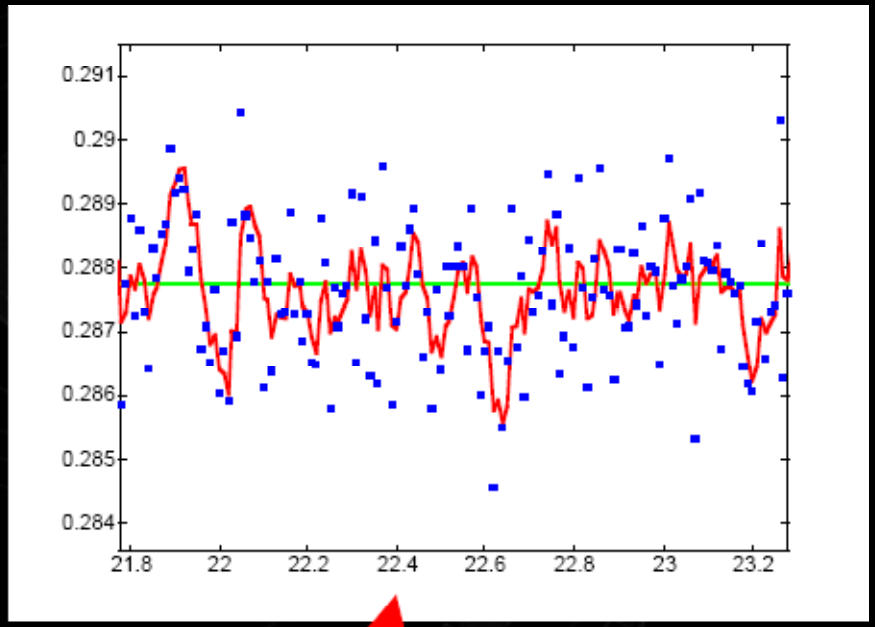
$$Q(n) = E\left[w(n)w(n)^T\right] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{u1}^2(n) & 0 \\ 0 & 0 & 0 & \sigma_{u2}^2(n) \end{bmatrix}$$

$$x(2) = \begin{bmatrix} z_1(2) \\ z_2(2) \\ \dfrac{z_1(2) - z_1(1)}{dt} \\ \dfrac{z_2(2) - z_2(1)}{dt} \end{bmatrix}$$

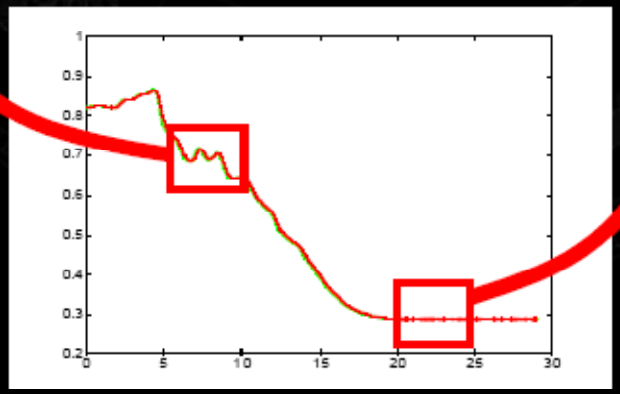
P the co-variance matrix of initial guess

Performance of PV Model





improved latency when moving…



…relatively *noisy* when not

Many Applications

- **Engineering**
  - Robotics, spacecraft, aircraft, automobiles
- **Computer**
  - Tracking, real-time graphics, computer vision
- **Economics**
  - Forecasting economic indicators
- .....