Ball Tracking Example

Robert B. Fisher
School of Informatics
University of Edinburgh

BALL TRACKING WITH THE KALMAN FILTER

Ball physical model:

Position: $\vec{p}_t = (col_t, row_t)'$
Velocity: $\vec{v}_t = (velcol_t, velrow_t)'$
Position update: $\vec{p}_t = \vec{p}_{t-1} + \vec{v}_{t-1}\Delta t$
Velocity update: $\vec{v}_t = \vec{v}_{t-1} + \vec{a}_{t-1}\Delta t$
Acceleration (gravity down): $\vec{a}_t = (0, g)'$

State vector: $\vec{x}_t = (col_t, row_t, velcol_t, velrow_t)'$
Initial state vector: random

Ball physics update

Prediction: $\vec{y}_t = \mathbf{A}\vec{x}_{t-1} + \mathbf{B}\vec{u}_t$

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathrm{B}\vec{u}_t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ g\Delta t \end{bmatrix}$$

Use $\Delta t = 1$

Rest of model

Observation process:

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

Measurement noise:

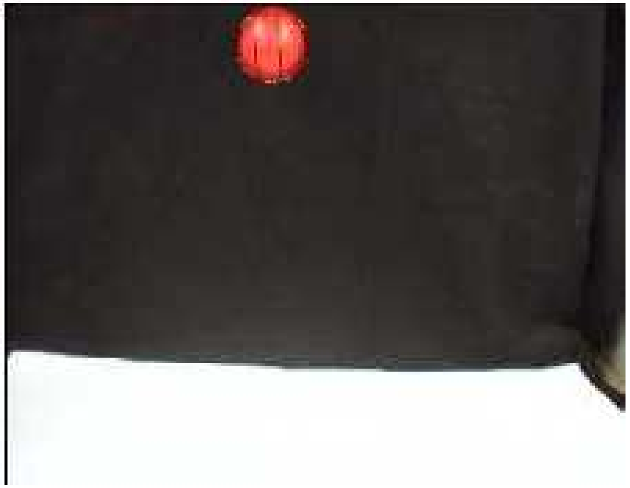
$$R = \begin{bmatrix} 0.285 & 0.005 \\ 0.005 & 0.046 \end{bmatrix}$$

System noise: $Q = 0.01 \times I$

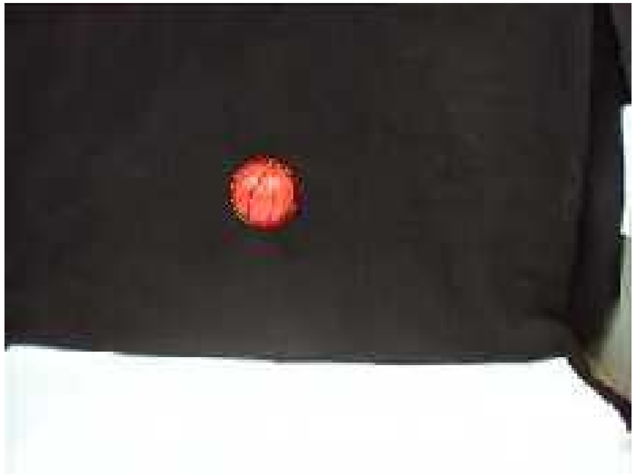
KALMAN FILTER SUCCESSES

SEE: `homepages.inf.ed.ac.uk/rbf/...`

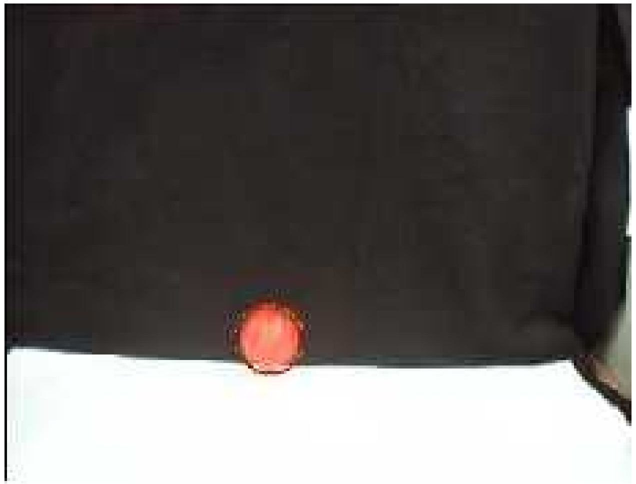
`...AVINVERTED/DEMOS/TRACK/demo.html` 8:



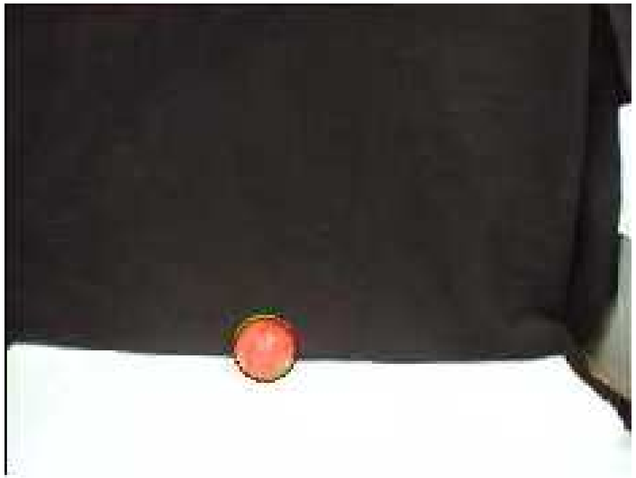
21:



35:

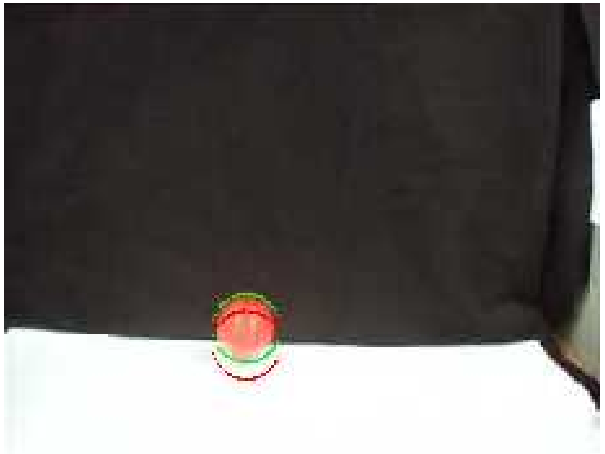


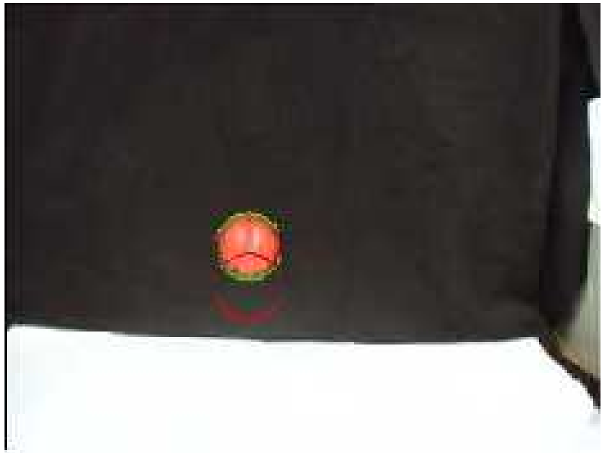
42:



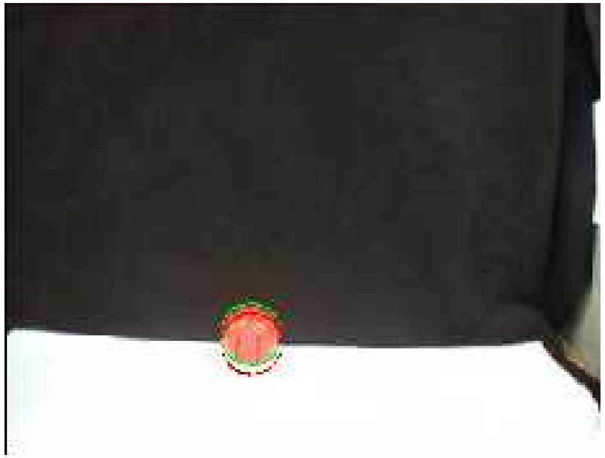
KALMAN FILTER FAILURES

14: BOUNCE OVERSHOOT 16: SLOW CATCH UP





59: GRAVITY PULLS DOWN AT REST



Ball tracking analysis

- KF smooths noisy observations (not so noisy here) to give better estimates
- Could also estimate ball radius
- Could also plot boundary of 95% likelihood of ball position - grows when fit is bad
- Dynamic model doesn't work at bounce & stop