Condensation Tracking of Ball

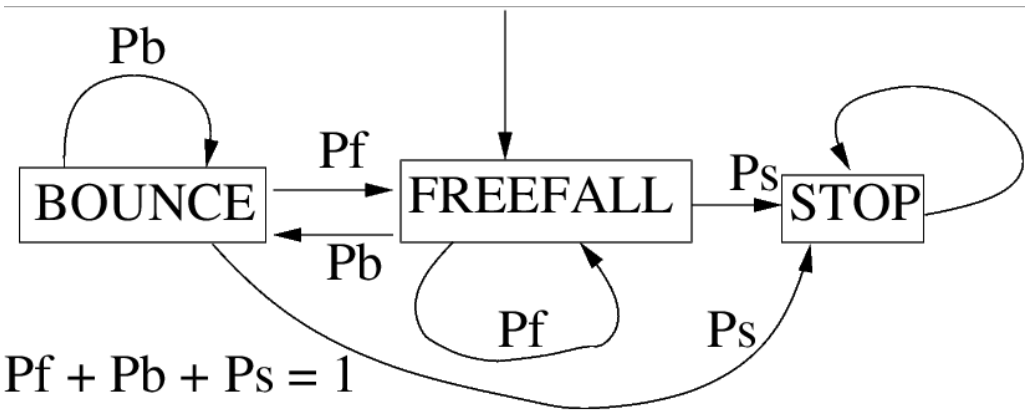
Robert B. Fisher
School of Informatics
University of Edinburgh

CONDENSATION TRACKING OF BOUNCING BALL

1) Select (N=100 samples) of a ball motion vector by probability of vector

2) Use estimated covariance P() to create state samples $\vec{s}_{t-1}$

3) Situation switching model. $P_b = 0.3, P_s = 0.05$



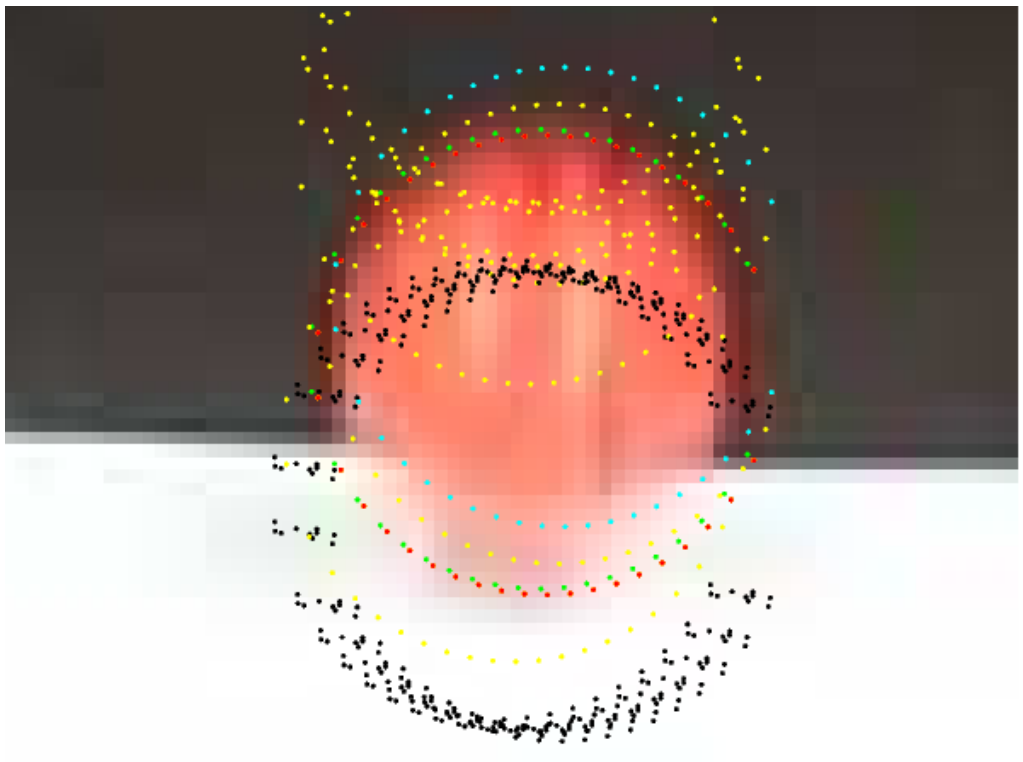
If in STOP situation: zero vertical speed

If in BOUNCE situation: $v_{row} = -0.7 * v_{row}$
Also don't know when bounce was so
add some random vertical motion

Then use Kalman filter

4) Estimate hypothesis goodness by
$1/ \mid\mid \mathrm{H}\vec{x}_t - \vec{z}_t \mid\mid^2$
Normalize to estimate hypothesis probability

EXAMPLE OF SAMPLING EFFECTS



Red:final estimate Green:data
Yellow:BOUNCE Blue:STOP Black:FALL

CONDENSATION TRACKING CORE CODE

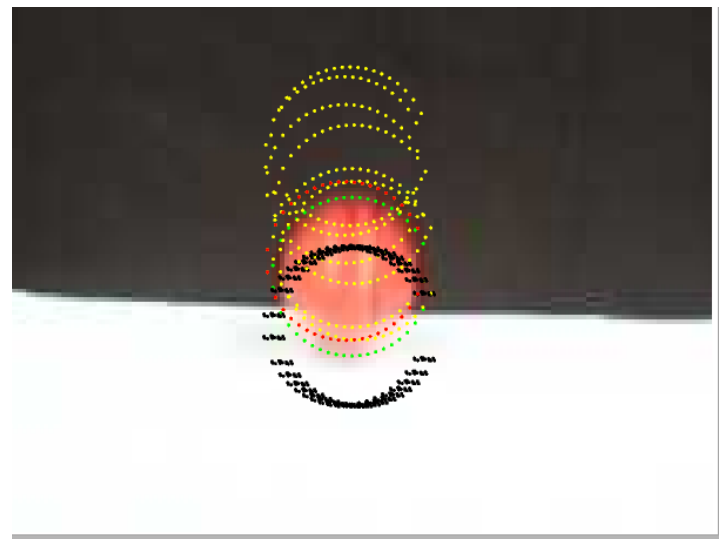
```
ident: an array of IDCOUNT sample ids. Each id
appears with the same probability as in H_(t-1)
P(): estimated state covariance
x(): state vectors

% generate NCON new samples
for newsample = 1 : NCON

  oldsample = ident(ceil(IDCOUNT*rand(1)));
  xc(:) = x(oldsample,time-1,:); % get old state
```

```
% generate a new SAMPLE at this state
xc=xc+5*sqrt(P(oldsample,time-1,:,:))*randn(4);
if tracksituation(oldsample,time-1)==1 %if stop
  A,B = ...    % replace A,B for stop model
  xc(4) = 0;   % zero vertical velocity
  tracksituation(newsample,time)=1;
else
  r=rand(1);% random number for sit. selection

  if r < pstop   % gone to stop situation
    A,B = ...    % replace A,B for state model
    xc(4) = 0; % zero vertical velocity
    tracksituation(newsample,time)=1;
```

```
  elseif r < (pbounce + pstop) % bounce sit.
    % add random vertical motion due to
    % imprecision about time of bounce
    xc(2) = xc(2) + 3*abs(xc(4))*(rand(1)-0.5);
    % invert velocity with some loss
    xc(4) = -loss*xc(4);
    tracksituation(newsample,time)=2;

  else % normal motion
    tracksituation(newsample,time)=3;

% update new hypotheses via Kalman filter
x(newsample,time,:) = f(xc)
P(newsample,time,:,:) = ...
```

```
  % weight hypothesis by distance from data
  dvec = [cc(time),cr(time)]
     - [x(newsample,time,1),x(newsample,time,2)];
  weights(newsample,time) = 1/(dvec*dvec');

% rescale new hypothesis weights to give sum=1
totalw=sum(weights(:,time)');
weights(:,time)=weights(:,time)/totalw;

% select top hypothesis to draw
subset=weights(:,time);
top = find(subset == max(subset));
```

KALMAN FILTER FAILURES FIXED



N=20 HYPS Y:BOUNCE B:FALL R:FINAL G:DATA

RESULTS URL

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

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

What We Have Learned

1. State model for bouncing ball
2. Details of how to implement Condensation model