Computer Vision II - Lecture 9

Beyond Kalman Filters

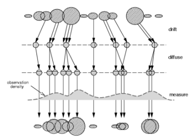
22.05.2014

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Course Outline

- Single-Object Tracking
  - Background modeling
  - Template based tracking
  - Color based tracking
  - Contour based tracking
  - Tracking by online classification
  - Tracking-by-detection
- Bayesian Filtering
  - Kalman filters
  - Particle filters
  - Case studies
- Multi-Object Tracking
- Articulated Tracking



Today: Beyond Gaussian Error Models

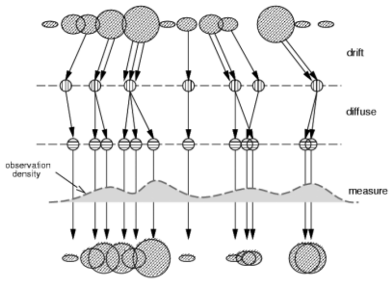


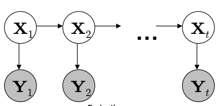
Figure from Isard & Blake

Topics of This Lecture

- Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions
- Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling
- Case study
  - Detector Confidence Particle Filter
  - Role of the different elements

Recap: Tracking as Inference

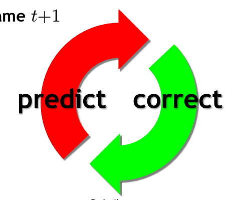
- Inference problem
  - The hidden state consists of the true parameters we care about, denoted $\mathbf{X}$.
  - The measurement is our noisy observation that results from the underlying state, denoted $\mathbf{Y}$.
  - At each time step, state changes (from $\mathbf{X}_{t-1}$ to $\mathbf{X}_t$) and we get a new observation $\mathbf{Y}_t$.
- Our goal: recover most likely state $\mathbf{X}_t$ given
  - All observations seen so far.
  - Knowledge about dynamics of state transitions.



Slide credit: Kristen Grauman

Recap: Tracking as Induction

- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(\mathbf{X}_0)$
  - At the first frame, *correct* this given the value of $\mathbf{Y}_0 = \mathbf{y}_0$
- Given corrected estimate for frame t:
  - Predict for frame $t+1$
  - Correct for frame $t+1$



Slide credit: Svetlana Lazebnik

Recap: Prediction and Correction

- **Prediction:**

$$P(X_t \mid y_0,\ldots,y_{t-1}) = \int \underbrace{P(X_t \mid X_{t-1})}_{\text{Dynamics model}} \underbrace{P(X_{t-1} \mid y_0,\ldots,y_{t-1})}_{\text{Corrected estimate from previous step}} dX_{t-1}$$

- **Correction:**

$$P(X_t \mid y_0,\ldots,y_t) = \frac{\overbrace{P(y_t \mid X_t)}^{\text{Observation model}} \overbrace{P(X_t \mid y_0,\ldots,y_{t-1})}^{\text{Predicted estimate}}}{\int P(y_t \mid X_t) P(X_t \mid y_0,\ldots,y_{t-1}) dX_t}$$

Slide credit: Svetlana Lazebnik B. Leibe 7

Computer Vision II, Summer'14

Recap: Linear Dynamic Models

- Dynamics model
  - State undergoes linear tranformation D_t plus Gaussian noise

$$\boldsymbol{x}_t \sim N\left(\boldsymbol{D}_t \boldsymbol{x}_{t-1}, \Sigma_{d_t}\right)$$

- Observation model
  - Measurement is linearly transformed state plus Gaussian noise

$$\boldsymbol{y}_t \sim N\left(\boldsymbol{M}_t \boldsymbol{x}_t, \Sigma_{m_t}\right)$$

Slide credit: S. Lazebnik, K. Grauman B. Leibe 8

Computer Vision II, Summer'14

Recap: Constant Velocity Model (1D)

- State vector: position p and velocity v

$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} \qquad \begin{aligned} p_t &= p_{t-1} + (\Delta t) v_{t-1} + \varepsilon \\ v_t &= v_{t-1} + \xi \end{aligned}$$

(greek letters denote noise terms)

$$x_t = D_t x_{t-1} + noise = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + noise$$

- Measurement is position only

$$y_t = M x_t + noise = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix} + noise$$

Slide credit: S. Lazebnik, K. Grauman B. Leibe 9

Computer Vision II, Summer'14

Recap: Constant Acceleration Model (1D)

- State vector: position p, velocity v, and acceleration a.

$$x_t = \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} \qquad \begin{aligned} p_t &= p_{t-1} + (\Delta t) v_{t-1} + \varepsilon \\ v_t &= v_{t-1} + (\Delta t) a_{t-1} + \xi \\ a_t &= a_{t-1} + \zeta \end{aligned}$$

(greek letters denote noise terms)

$$x_t = D_t x_{t-1} + noise = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \\ a_{t-1} \end{bmatrix} + noise$$

- Measurement is position only

$$y_t = M x_t + noise = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} + noise$$

Slide credit: S. Lazebnik, K. Grauman B. Leibe 10

Computer Vision II, Summer'14

Recap: General Motion Models

- Assuming we have differential equations for the motion
  - E.g. for (undampened) periodic motion of a pendulum

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

- Substitute variables to transform this into linear system

$$p_1 = p \qquad p_2 = \frac{dp}{dt} \qquad p_3 = \frac{d^2 p}{dt^2}$$

- Then we have

$$x_t = \begin{bmatrix} p_{1,t} \\ p_{2,t} \\ p_{3,t} \end{bmatrix} \qquad \begin{aligned} p_{1,t} &= p_{1,t-1} + (\Delta t) p_{2,t-1} + \varepsilon \\ p_{2,t} &= p_{2,t-1} + (\Delta t) p_{3,t-1} + \xi \\ p_{3,t} &= -p_{1,t-1} + \zeta \end{aligned} \qquad D_t = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ -1 & 0 & 0 \end{bmatrix}$$

B. Leibe 11

Computer Vision II, Summer'14

Recap: The Kalman Filter

Know corrected state from previous time step, and all measurements up to the current one → Predict distribution over next state.

Receive measurement

Know prediction of state, and next measurement → Update distribution over current state.

Time update ("Predict")

Measurement update ("Correct")

$P(X_t \mid y_0,\ldots,y_{t-1})$

$P(X_t \mid y_0,\ldots,y_t)$

Mean and std. dev. of predicted state: μ_t^-, σ_t^-

Time advances: t++

Mean and std. dev. of corrected state: μ_t^+, σ_t^+

Slide credit: Kristen Grauman B. Leibe 12

Computer Vision II, Summer'14

Recap: General Kalman Filter (>1dim)

- What if state vectors have more than one dimension?

PREDICT

$$x_t^- = D_t x_{t-1}^+$$

$$\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_{d_t}$$

CORRECT

$$K_t = \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t}\right)^{-1}$$

"Kalman gain" "residual"

$$x_t^+ = x_t^- + K_t\left(y_t - M_t x_t^-\right)$$

$$\Sigma_t^+ = \left(I - K_t M_t\right)\Sigma_t^-$$

More weight on residual when measurement error covariance approaches 0.

Less weight on residual as a priori estimate error covariance approaches 0.

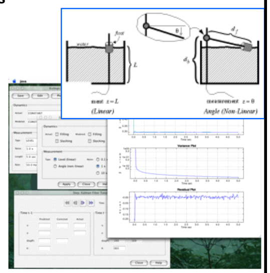
for derivations, see F&P Chapter 17.3

Slide credit: Kristen Grauman

Resources: Kalman Filter Web Site

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

- Electronic and printed references
  - Book lists and recommendations
  - Research papers
  - Links to other sites
  - Some software
- News
- Java-Based KF Learning Tool
  - On-line 1D simulation
  - Linear and non-linear
  - Variable dynamics



Slide adapted from Greg Welch

Remarks

- Try it!
  - Not too hard to understand or program
- Start simple
  - Experiment in 1D
  - Make your own filter in Matlab, etc.
- Note: the Kalman filter "wants to work"
  - Debugging can be difficult
  - Errors can go un-noticed

Slide adapted from Greg Welch

Topics of This Lecture

- Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions
- Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling
- Case study
  - Detector Confidence Particle Filter
  - Role of the different elements

Extension: Extended Kalman Filter (EKF)

- Basic idea
  - State transition and observation model don't need to be linear functions of the state, but just need to be differentiable.

$$x_t = f\left(x_{t-1}, u_t\right) + \varepsilon$$

$$y_t = h\left(x_t\right) + \xi$$

  - The EKF essentially linearizes the nonlinearity around the current estimate by a Taylor expansion.
- Properties
  - Unlike the linear KF, the EKF is in general *not* an optimal estimator.
    - If the initial estimate is wrong, the filter may quickly diverge.
  - Still, it's the de-facto standard in many applications
    - Including navigation systems and GPS

Kalman Filter - Other Extensions

- Unscented Kalman Filter (UKF)
  - Further development of EKF
  - Probability density is approximated by nonlinear transform of a random variable.
  - More accurate results than the EKF's Taylor expansion approx.
- Ensemble Kalman Filter (EnKF)
  - Represents the distribution of the system state using a collection (an *ensemble*) of state vectors.
  - Replace covariance matrix by *sample covariance* from ensemble.
  - Still basic assumption that all prob. distributions involved are Gaussian.
  - EnKFs are especially suitable for problems with a large number of variables.

Even More Extensions

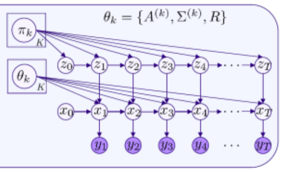
Switching linear dynamical system (SLDS):

$z_t \sim \pi_{z_{t-1}}$

$x_t = A^{(z_t)} x_{t-1} + e_t(z_t)$

$y_t = C x_t + w_t$

$e_t \sim \mathcal{N}(0, \Sigma^{(z_t)}) \quad w_t \sim \mathcal{N}(0, R)$



- Switching Linear Dynamic System (SLDS)
  - Use a set of k dynamic models $A^{(1)}, \ldots, A^{(k)}$, each of which describes a different dynamic behavior.
  - Hidden variable z_t determines which model is active at time t.
  - A switching process can change z_t according to distribution $\pi_{z_{t-1}}$.

Figure source: Erik Sudderth

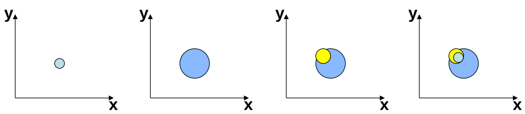
Topics of This Lecture

- Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions
- Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling
- Case study
  - Detector Confidence Particle Filter
  - Role of the different elements

Today: only main ideas

Formal introduction next Tuesday

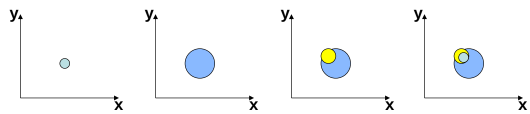
When Is A Single Hypothesis Too Limiting?



Slide credit: Kristen Grauman

Figure from Thrun & Kosecka

When Is A Single Hypothesis Too Limiting?



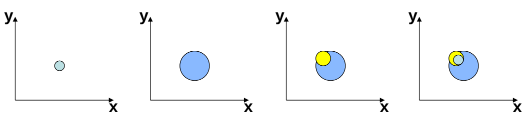
- Consider this example: say we are tracking the face on the right using a skin color blob to get our measurement.

Video from Jojic & Frey

Slide credit: Kristen Grauman

Figure from Thrun & Kosecka

When Is A Single Hypothesis Too Limiting?



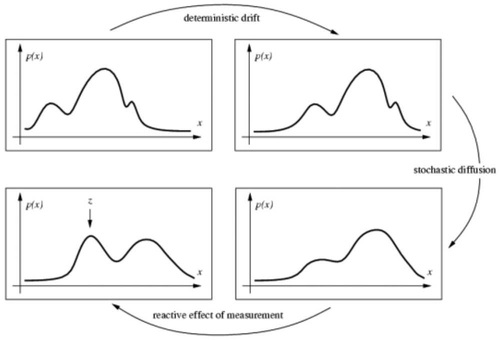
- Consider this example: say we are tracking the face on the right using a skin color blob to get our measurement.

Video from Jojic & Frey

Slide credit: Kristen Grauman

Figure from Thrun & Kosecka

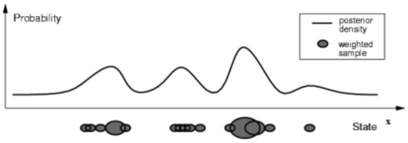
Propagation of General Densities



Slide credit: Svetlana Lazebnik

Figure from Isard & Blake

Factored Sampling



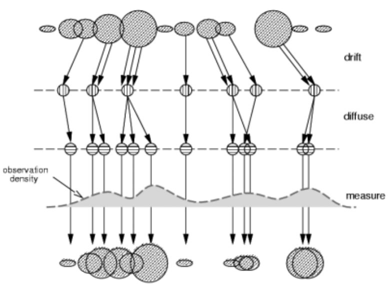
- Idea: Represent state distribution non-parametrically
  - Prediction: Sample points from prior density for the state, $P(X)$
  - Correction: Weight the samples according to $P(Y|X)$

$$P(X_t \mid y_0,\ldots,y_t) = \frac{P(y_t \mid X_t)P(X_t \mid y_0,\ldots,y_{t-1})}{\int P(y_t \mid X_t)P(X_t \mid y_0,\ldots,y_{t-1})dX_t}$$

Particle Filtering

- (Also known as Sequential Monte Carlo Methods)
- Idea
  - We want to use sampling to propagate densities over time (i.e., across frames in a video sequence).
  - At each time step, represent posterior $P(X_t | Y_t)$ with weighted sample set.
  - Previous time step's sample set $P(X_{t-1} | Y_{t-1})$ is passed to next time step as the effective prior.

Particle Filtering



Start with weighted samples from previous time step

Sample and shift according to dynamics model

Spread due to randomness; this is predicted density $P(X_t | Y_{t-1})$

Weight the samples according to observation density

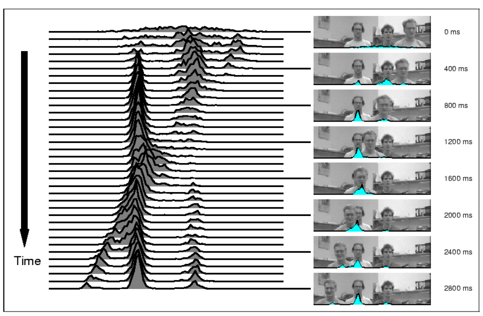
Arrive at corrected density estimate $P(X_t | Y_t)$

M. Isard and A. Blake, CONDENSATION -- conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998

Particle Filtering - Visualization

Code and video available from
http://www.robots.ox.ac.uk/~misard/condensation.html

Particle Filtering Results



http://www.robots.ox.ac.uk/~misard/condensation.html

Particle Filtering Results

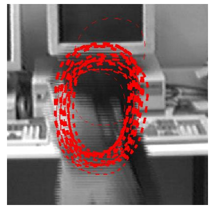
- Some more examples

http://www.robots.ox.ac.uk/~misard/condensation.html

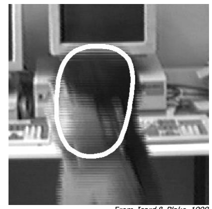
Obtaining a State Estimate

- Note that there's no explicit state estimate maintained, just a "cloud" of particles
- Can obtain an estimate at a particular time by querying the current particle set
- Some approaches
  - "Mean" particle
    - Weighted sum of particles
    - Confidence: inverse variance
  - Really want a mode finder—mean of tallest peak

Condensation: Estimating Target State



State samples
(thickness proportional to weight)



From *Isard & Blake, 1998*

Mean of weighted
state samples

Summary: Particle Filtering

- **Pros:**
  - Able to represent arbitrary densities
  - Converging to true posterior even for non-Gaussian and nonlinear system
  - Efficient: particles tend to focus on regions with high probability
  - Works with many different state spaces
    - E.g. articulated tracking in complicated joint angle spaces
  - Many extensions available

Summary: Particle Filtering

- **Cons / Caveats:**
  - #Particles is important performance factor
    - Want as few particles as possible for efficiency.
    - But need to cover state space sufficiently well.
  - Worst-case complexity grows exponentially in the dimensions
  - Multimodal densities possible, but still single object
    - Interactions between multiple objects require special treatment.
    - Not handled well in the particle filtering framework (state space explosion).

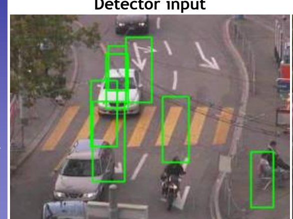
Topics of This Lecture

- Recap: Kalman Filter
  - Basic ideas
  - Limitations
  - Extensions
- Particle Filters
  - Basic ideas
  - Propagation of general densities
  - Factored sampling
- Case study
  - Detector Confidence Particle Filter
  - Role of the different elements

Challenge: Unreliable Object Detectors

- Example:
  - Low-res webcam footage (320×240), MPEG compressed

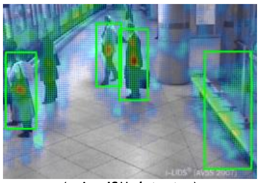
Detector input

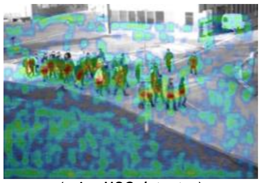


Tracker output

How to get from here... ? ...to here?

Tracking based on Detector Confidence

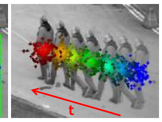

(using ISM detector)


(using HOG detector)

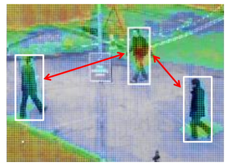
- Detector output is often not perfect
  - Missing detections and false positives
  - But continuous confidence still contains useful cues.
- Idea employed here:
  - Use continuous detector confidence to track persons over time.

Main Ideas

- Detector confidence particle filter
  - Initialize particle cloud on strong object detections.
  - Propagate particles using continuous detector confidence as observation model.

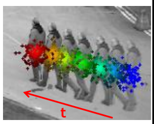


- Disambiguate between different persons
  - Train a person-specific classifier with online boosting.
  - Use classifier output to distinguish between nearby persons.



[Breitenstein, Reichlin, Leibe et al., ICCV'09]

Detector Confidence Particle Filter



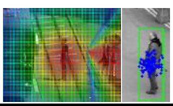
- State: $x = \{x, y, u, v\}$
- Motion model (constant velocity)

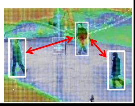
$$(x,y)_t = (x,y)_{t-1} + (u,v)_{t-1} \cdot \Delta t + \varepsilon_{(x,y)}$$
$$(u,v)_t = (u,v)_{t-1} + \varepsilon_{(u,v)}$$

- Observation model

$$w_{tr,p} = p(y_t | x_t^{(i)}) = \beta \cdot \mathcal{I}(tr) \cdot p_{\mathcal{N}}(p - d^*) + \gamma \cdot d_c(p) \cdot p_o(tr) + \eta \cdot c_{tr}(p)$$

Discrete detections | Detector confidence | Classifier confidence





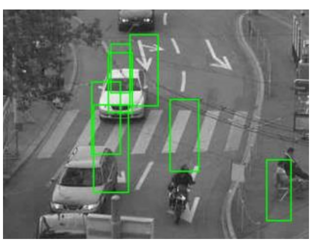
When Is Which Term Useful?

Discrete detections | Detector confidence | Classifier confidence

Each Observation Term Increases Robustness!

Observation Model Terms	MOTP	MOTA	FN	FP	ID Sw.
1: Det+DetConf+Class	70.0%	72.9%	26.8%	0.3%	0
2: Det+DetConf	64.0%	54.5%	28.2%	17.2%	5
3: Det+Class	65.0%	55.3%	31.3%	13.4%	0
4: Det	67.0%	40.9%	30.7%	28.0%	10

Detector only

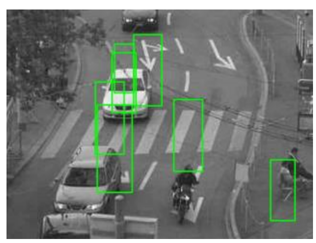


CLEAR MOT scores

Each Observation Term Increases Robustness!

Observation Model Terms	MOTP	MOTA	FN	FP	ID Sw.
1: Det+DetConf+Class	70.0%	72.9%	26.8%	0.3%	0
2: Det+DetConf	64.0%	54.5%	28.2%	17.2%	5
3: Det+Class	65.0%	55.3%	31.3%	13.4%	0
4: Det	67.0%	40.9%	30.7%	28.0%	10

Detector
+ Confidence

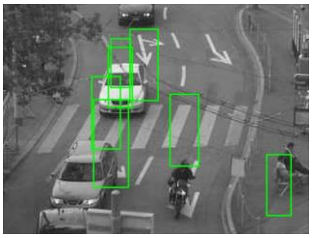


CLEAR MOT scores

Each Observation Term Increases Robustness!

Observation Model Terms	MOTP	MOTA	FN	FP	ID Sw.
1: Det+DetConf+Class	70.0%	72.9%	26.8%	0.3%	0
2: Det+DetConf	64.0%	54.5%	28.2%	17.2%	5
3: Det+Class	65.0%	55.3%	31.3%	13.4%	0
4: Det	67.0%	40.9%	30.7%	28.0%	10

Detector
+ Classifier

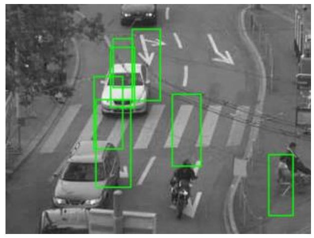


CLEAR MOT scores

Each Observation Term Increases Robustness!

Observation Model Terms	MOTP	MOTA	FN	FP	ID Sw.
1: Det+DetConf+Class	70.0%	72.9%	26.8%	0.3%	0
2: Det+DetConf	64.0%	54.5%	28.2%	17.2%	5
3: Det+Class	65.0%	55.3%	31.3%	13.4%	0
4: Det	67.0%	40.9%	30.7%	28.0%	10

Detector
+ Confidence
+ Classifier

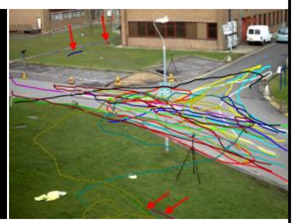


False negatives, false positives, and ID switches decrease!

CLEAR MOT scores

Qualitative Results

Remaining Issues



- Some false positive initializations at wrong scales…
  - Due to limited scale range of the person detector.
  - Due to boundary effects of the person detector.

References and Further Reading

- A good tutorial on Particle Filters
  - M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. In *IEEE Transactions on Signal Processing*, Vol. 50(2), pp. 174-188, 2002.
- The CONDENSATION paper
  - M. Isard and A. Blake, CONDENSATION - conditional density propagation for visual tracking, IJCV 29(1):5-28, 1998