“Stochastic Road Shape Estimation,” B. Southall & C. Taylor

Review by:

Christopher Rasmussen

University of Delaware

Announcements

- Readings for next Tuesday: Chapter 14-14.4, 22-22.5 in Forsyth & Ponce

Main Contributions

- Robust estimation of road shape 80 meters ahead on highways, plus car bearing, position within lane
  - Recovers from mistracking
- Handles variety of lane types in different lighting conditions
- Integrates camera with non-visual modalities

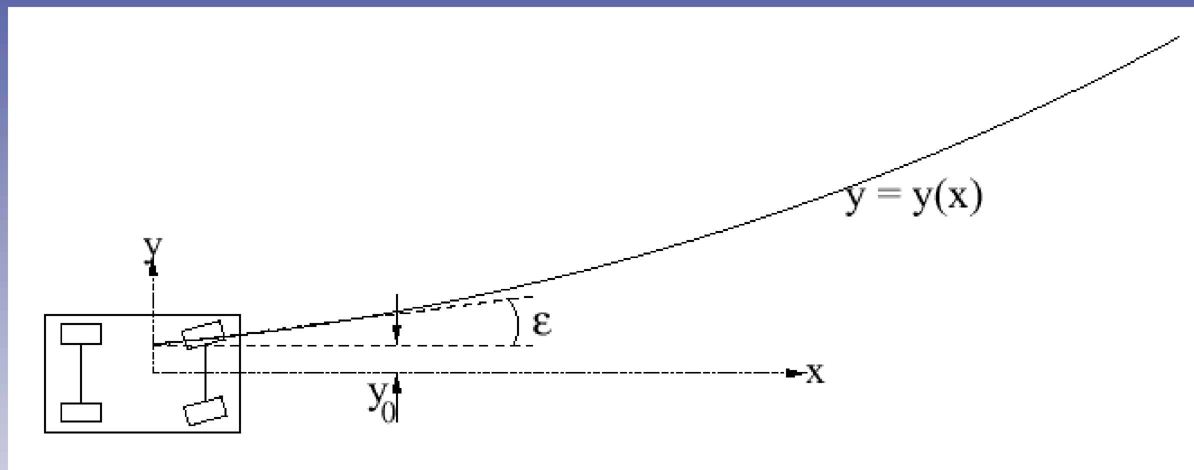
Primary Techniques

- Condensation algorithm (particle filtering) for lane line tracking
- Specialized image processing to detect lane lines despite significant changes in illumination conditions

Assumptions

- Internal camera calibration available
- Needs to initialize camera pitch, height on lane of known width
- Flat road
- Accelerometers provide velocity, yaw rates
- Scanning radar detects on-road obstacles

Lane, Vehicle State



$$\mathbf{s}(t) = [y_0(t), \tan\epsilon(t), C_0(t), C_1(t), W(t), \theta(t)]^T, \quad (1)$$

lateral offset within lane

tan of bearing wrt lane

lane curvature

rate of change of curvature

lane width

camera pitch wrt road plane

Road Shape Function

- Cubic polynomial

$$y(x) = y_0 + \tan(\epsilon)x + \frac{C_0}{2}x^2 + \frac{C_1}{6}x^3, \qquad (2)$$

Dynamical Model

noise

yaw change

$$\mathbf{s}(t+\Delta t) = A(\Delta x)\mathbf{s}(t) + \boxed{\begin{bmatrix} 0 \\ -\dot{\psi}\Delta t \\ 0 \\ 0 \\ 0 \\ \Delta\theta \end{bmatrix}} + \mathbf{w}(t), \qquad (4)$$

pitch change

control/correction term

where

$$A(\Delta x) = \begin{bmatrix} 1 & \Delta x & \frac{\Delta x^2}{2} & \frac{\Delta x^3}{6} & 0 & 0 \\ 0 & 1 & \Delta x & \frac{\Delta x^2}{2} & 0 & 0 \\ 0 & 0 & 1 & \Delta x & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (5)$$

Measurement Model

- How to predict image coordinates of lane lines from road shape function (2), which is defined in the ground plane?
- Some trigonometry + applying perspective projection yields

$$u = \frac{-y}{x\cos\theta + H\sin\theta}, v = \frac{H\cos\theta - x\sin\theta}{x\cos\theta + H\sin\theta}, \quad (6)$$

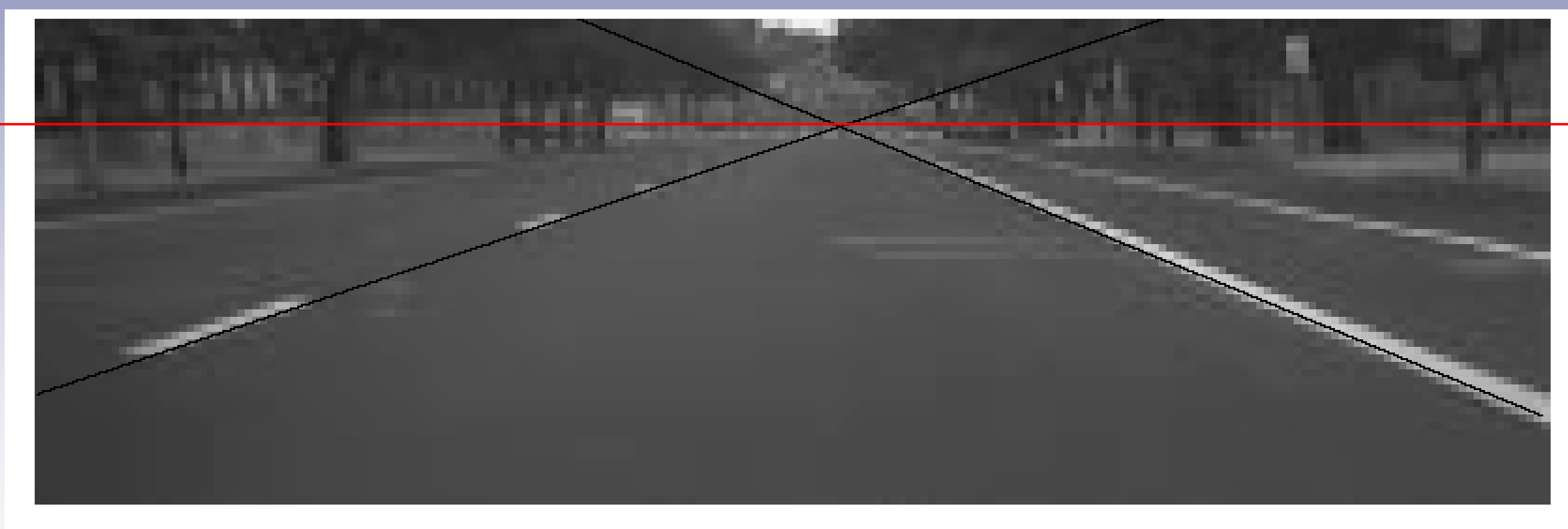
  where *H* is the camera height
- This is nonlinear

Handling Nonlinear Models

- Many system & measurement models can't be represented by matrix multiplications (e.g., sine function for periodic motion)
- Kalman filtering with nonlinearities
  - Extended Kalman filter
    - Linearize nonlinear function with 1st-order Taylor series approximation at each time step
  - Unscented Kalman filter
    - Approximate distribution rather than nonlinearity
    - More efficient and accurate to 2nd-order
    - See http://cslu.ece.ogi.edu/nsel/research/ukf.html

Pitch, Height Estimation

- Users indicates edges of known-width lane to find vanishing point & hence horizon line



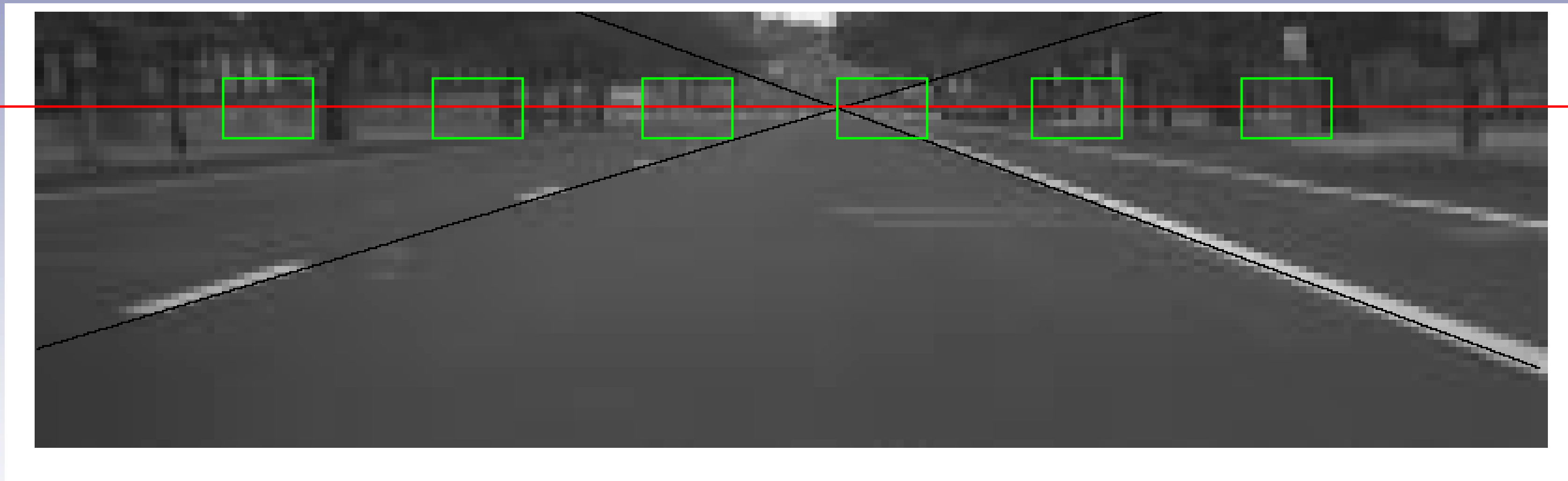
$$v_h = -\tan\theta$$

$$H = \frac{W\cos\theta}{m_r - m_l}$$

image height of horizon line
(limit of v term in (6) as x $\rightarrow \infty$)

Measuring Pitch Change

- SSD comparison of locations above and below horizon between successive frames to estimate vertical shift d_j



- Function:

$$\Delta\theta = \frac{d_j}{f_j(1+\tan^2\theta)}$$

camera focal length (vertical)

Finding Lane Markings

- Cross-correlation with triangular profile (e.g., kernel for line/roof edge detection) in red channel; threshold for candidates
- Must also exceed gray level threshold set dynamically depending on overall image brightness—helps with shadows
- Still have problems with false positives

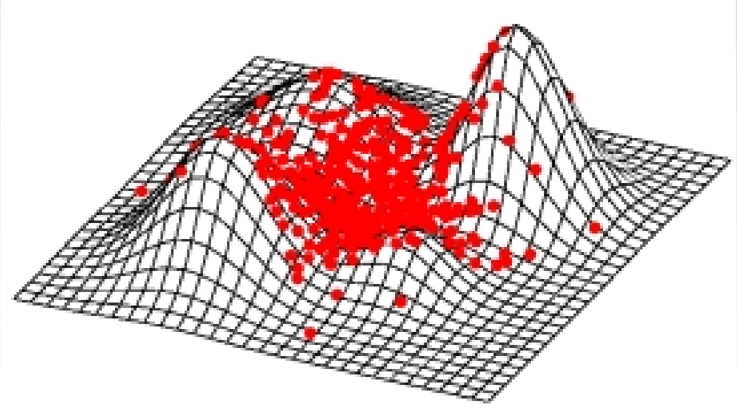
Figure 5. Feature extraction examples (extracted features marked with black dots)

Tracking as Estimation

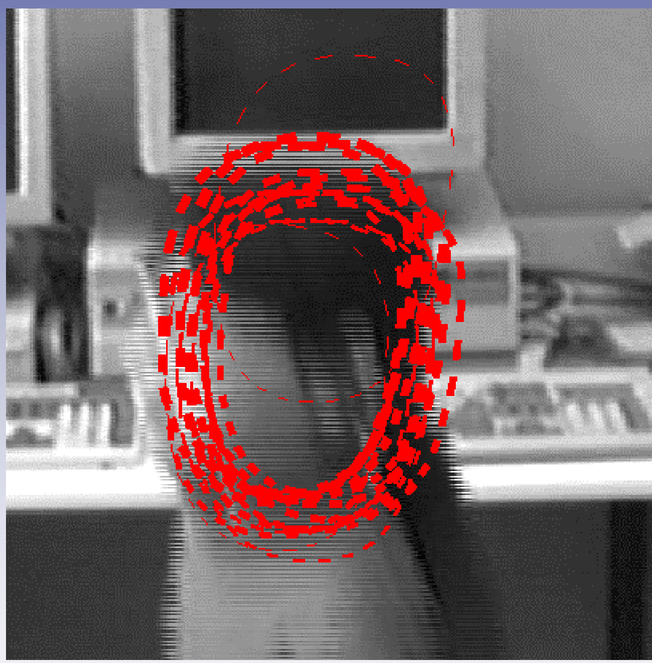
- Image likelihood $p\,(\mathbf{I} \mid \mathbf{X})$ compares image to expectation based on state
- State prior $p\,(\mathbf{X})$ summarizes domain knowledge, past estimates
- Bayesian approach: State posterior $p\,(\mathbf{X} \mid \mathbf{I}) \propto p\,(\mathbf{I} \mid \mathbf{X})\, p\,(\mathbf{X})$
- *Maximum a posteriori* (MAP) estimate: argmax of this expression—i.e., the most probable state
- Maximum likelihood (ML) estimate: state which maximizes image likelihood (i.e., all states equally likely *a priori*)

Estimation Using Condensation

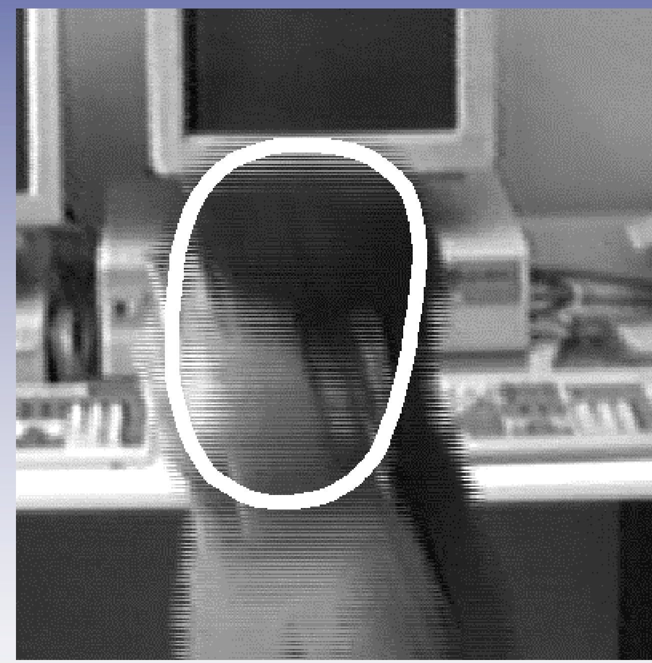
- Condensation: A particle filter developed for person tracking (Isard & Blake, 1996)
- Idea: Stochastic approximation of state posterior with a set of N weighted *particles* ($\mathbf{s}$, π), where $\mathbf{s}$ is a possible state and π is its weight
- Simulation instead of analytic solution—underlying probability distribution may take any form
- State estimate
  - Mean approach
    - Average particle
    - Confidence: inverse variance
  - Really want a mode finder



Condensation: Estimating Target State



State samples



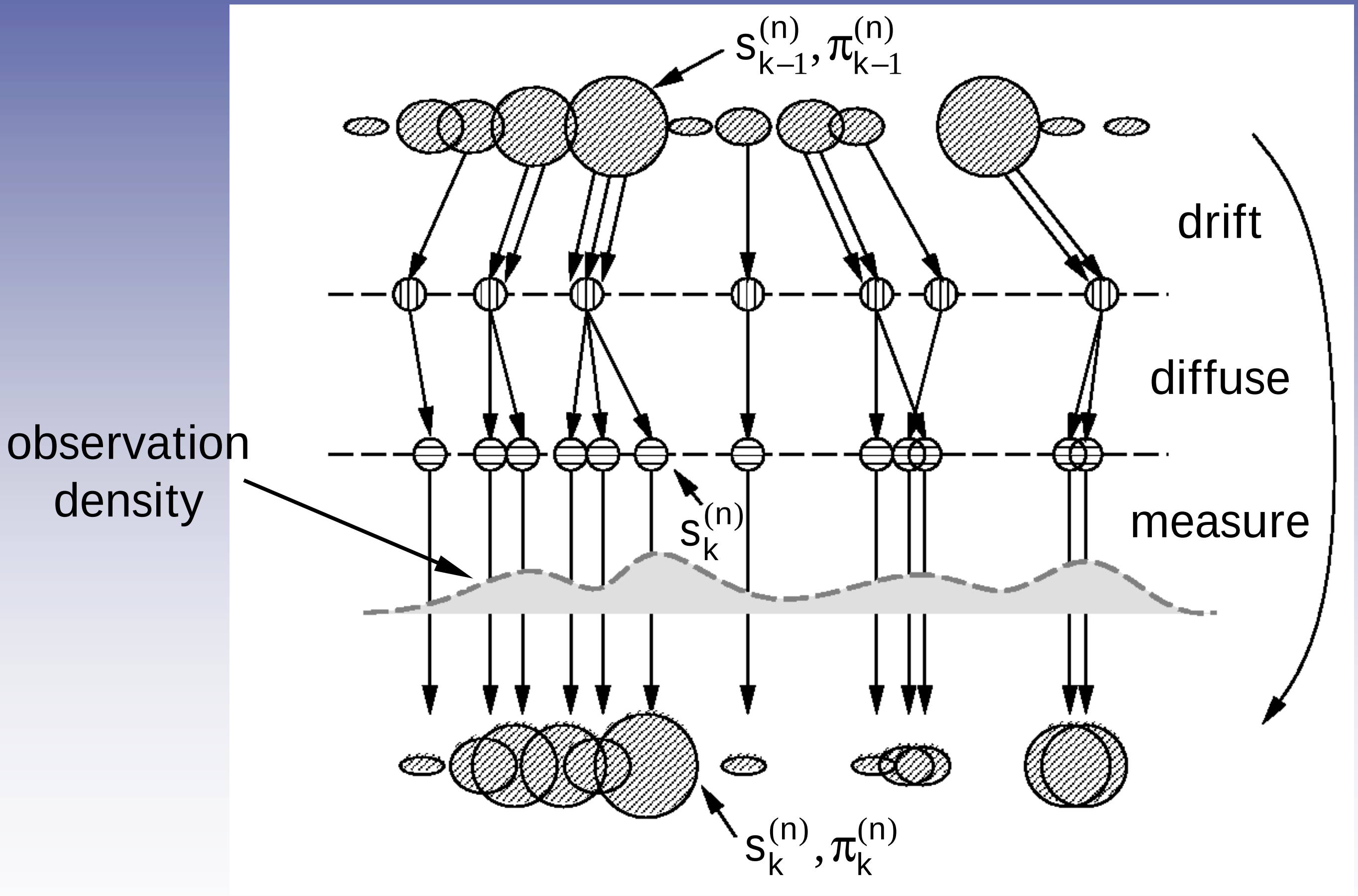
From *Isard & Blake, 1998*

Mean of weighted state samples

Updating the Particle Set

- (1) **Select**: Randomly select *N* particles based on weights; same particle may be picked multiple times (*factored sampling*)
- (2) **Predict**: Move particles according to deterministic dynamics (*drift*), then perturb individually (*diffuse*)
- (3) **Measure**: Get a likelihood for each new sample by making a prediction about the image's local appearance and comparing; then update weight on particle accordingly

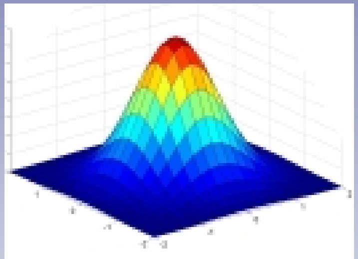
Condensation: Conditional density propagation



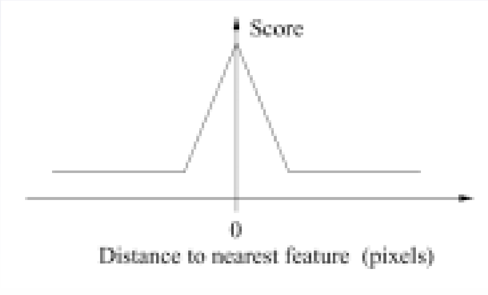
From *Isard & Blake, 1998*

Notes on Updating

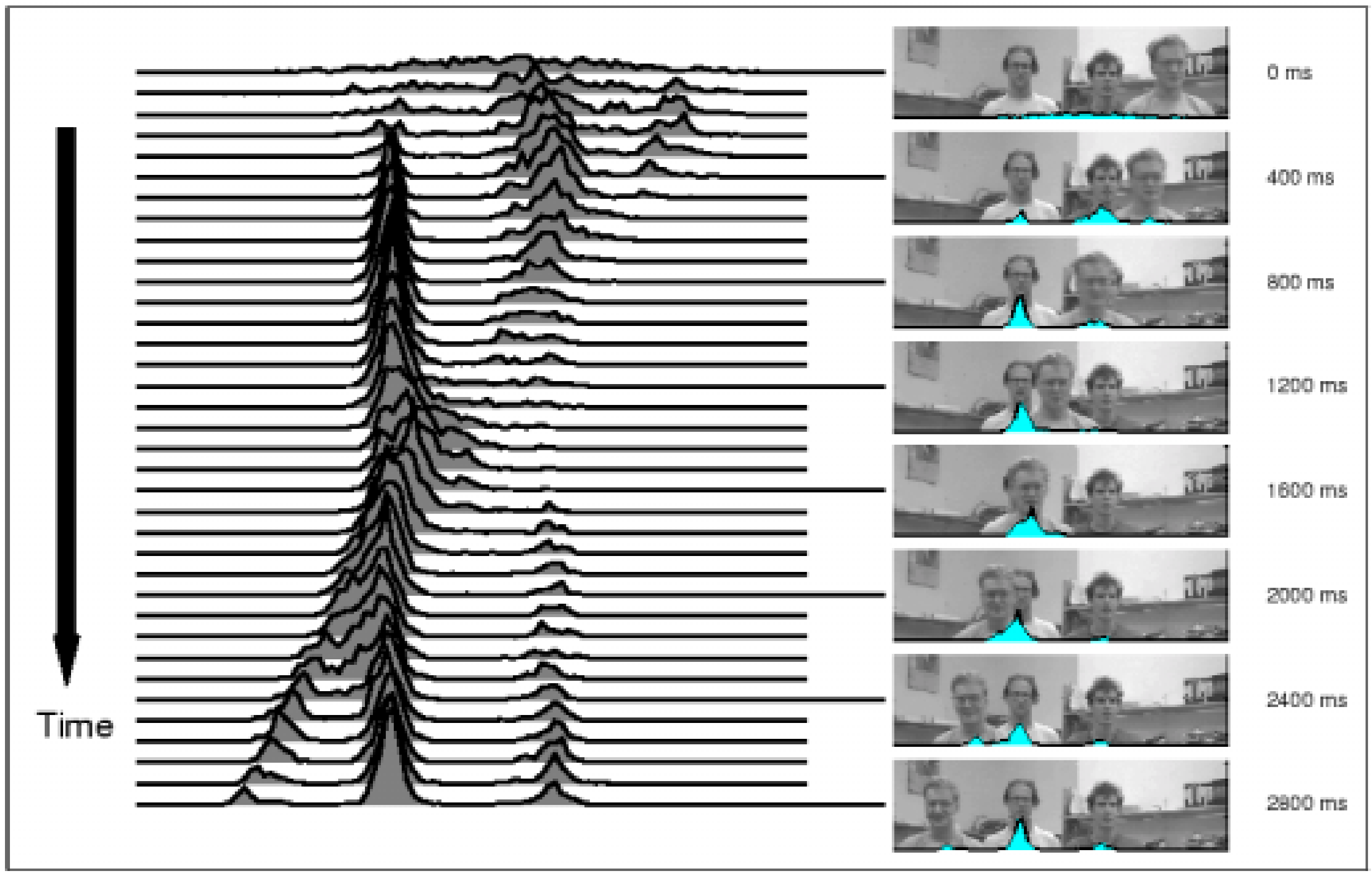
- Enforcing plausibility: Particles that represent impossible configurations are discarded
- Diffusion modeled with a Gaussian



- Likelihood function: Convert "goodness of prediction" score to pseudo-probability
  - More markings closer to predicted markings → Higher likelihood



Condensation: State posterior



From *Isard & Blake, 1998*

Benefits of Particle Filtering

- Nonlinear dynamics, measurement model easily incorporated
- Helps deal with lots of false positives for lane markings—i.e., multi-modal posterior okay, whereas it contradicts Kalman filter assumptions

Estimation on Real Sequence

Extensions to Condensation

- Partitioned sampling (MacCormick & Isard, 2000)
  - Split state up into low- (straight line) and high-frequency (curvature) components and sample hierarchically for efficiency
- Importance sampling (Isard & Blake, 1998)
  - Give "hints" by introducing samples at more likely spots in state space
    - Hough transform to fit lines to lane markings (see Forsyth & Ponce, Chapter 16.1)
    - Accelerometer data to get instantaneous curvature C_0
- Initialization samples
  - Importance samples drawn from prior to allow auto-initialization and recovery

Connections

- MAV paper also estimates horizon line (using a Kalman filter)—but with a bit more variation!
- Car tracking paper by Dellaert et al. detects cars visually, tracks them with Kalman filter
- Condensation algorithm used by museum tour guide to track its position

Related Work

- Shape extraction
  - Edge-based [Dickmanns, 1997; Taylor *et al.*, 1996]
  - Texture curvature [Pomerleau, 1995]
- Region-based segmentation
  - Color [Crisman & Thorpe, 1991; Fernandez & Casals, 1997]
  - Texture [Zhang & Nagel, 1994]
  - Structure from Motion [Smith, 1996]
- Sign finding
  - Template-matching [Betke & Makris, 1995]
  - Color [Piccioli *et al.*, 1994; Lauzière *et al.*, 2001]

Results

- Runs at 10.5 fps on PIII 867 MHz
- Good details on numbers of samples N in partitioned particle filter, percentage of importance samples and initialization samples, etc.
- No ground truth—surely could use GPS/ differential GPS + map for some quantification
- Found that bright sunlight and specularities from wet roads are a problem
- Curve estimation lags because dynamical model (Eq. 4) "does not predict non-random changes of curvature"

Comments

- No comparison of performance with and without partitioned sampling, importance sampling, etc. For that matter, there’s no comparison to Kalman filtering
- Image processing for illumination invariance fairly ad-hoc—isn’t there a better way than just using the red channel?
- No formula given for dynamic calculation of gray level threshold

Applications/Improvements

- Obviously, autonomous driving for transportation and cargo
  - Driver assistance: Computer doesn't steer, but it can warn, etc. Or, a more advanced version of cruise control
- Put yaw rate, velocity into state, even if they are estimated non-visually
- Initialize pitch, height automatically—their procedure "requires the user to specify the lines"—that's not trying hard enough
- Mean particle not a robust state estimation technique—what if multiple lanes are visible? How about trying to find them all, or detecting whether there are none?

Questions?