Stochastic Road Shape Estimation

B. Southall & C. J. Taylor

Overview

- A single on board color camera
- Estimate position of car wrt the lane
- Estimate the width and curvature of the lane ahead at distances of up to 80 meters
- Deal with a range of lighting conditions
- System runs at 10.5 fps

Primary Techniques

- Specialized image processing to detect lane lines despite significant changes in illumination conditions

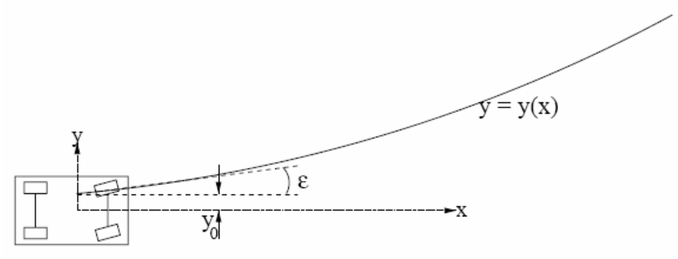
- Particle filtering for lane line tracking

  --Condensation algorithm (**Con**ditional **Dens**ity Propag**ation**)

Assumptions

- Internal camera calibration is 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 States



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

$y_0(t)$: lateral offset

ε: bearing of the vehicle wrt the centre-line of the lane

C_0, C_1: the curvature and rate of change of curvature of the lane ahead of the vehicle

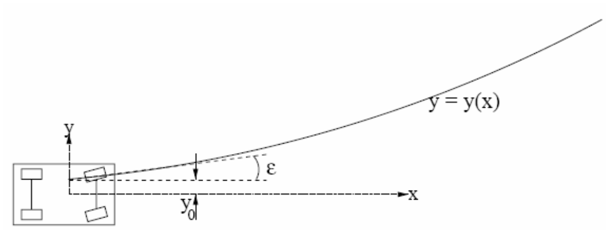
W : Width of the lane

θ : the pitch of the camera to the road surface

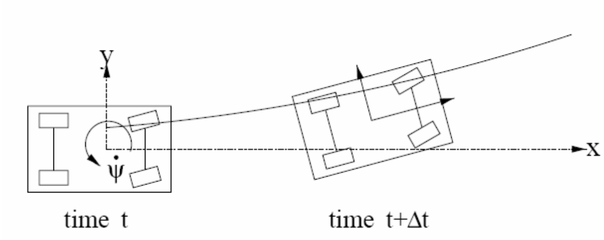
Road Shape Function

Cubic polynomial

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



State Evolution Model



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

Yaw change

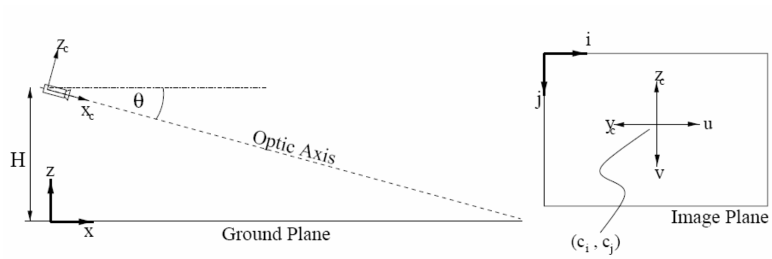
Control/ correction term

Noise

Pitch change

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

Nonlinear Imaging Model

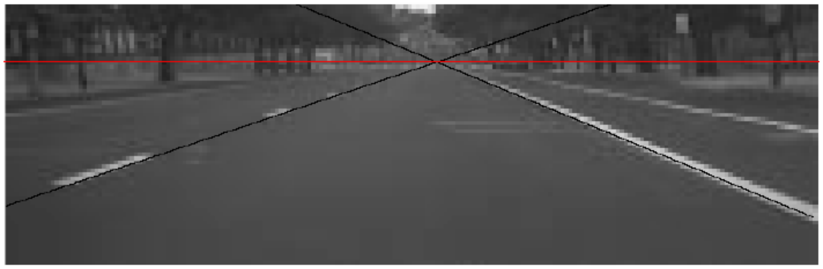


$$u = \frac{-y}{x\cos\theta + H\sin\theta}, v = \frac{H\cos\theta - x\sin\theta}{x\cos\theta + H\sin\theta},$$

Where H is the camera height

Pitch, Height Estimation

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



$v_h = -\tan\theta$

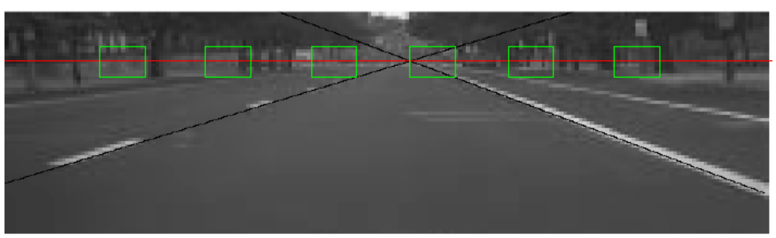
v_h : image height of horizon line

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

m is gradient of the left and right hand lane marking lines in the image

Measuring Pitch Change

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



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

d_j : vertical disparity

f_j : effective focal length of the camera in the j axis of the image

Lane Marking Extraction

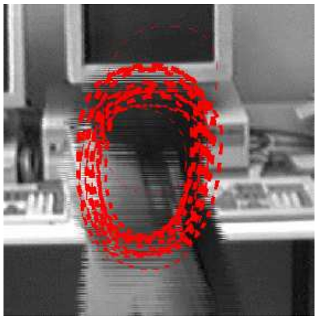
- Extract lane markings from the red channel (good contrast properties for both white and yellow image markers)
- A filter matched to the expected profile of a lane marker(Cross-correlation with triangular profile); threshold for candidates
- Also need exceed gray level threshold set dynamically depending on overall image brightness- helps with shadows and different light range
- Still have problems with false positives

Feature Extraction Example

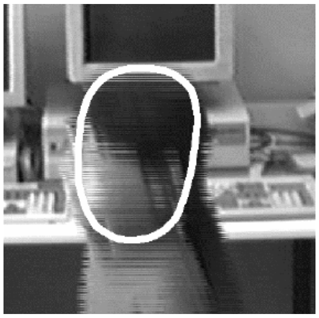
Introduction of CONDENSATION Algorithm

- CONDENSATION is a particle filter which was initially developed to address the problem of tracking curves in clutter.
- Stochastic approximation of state posterior with a set of N weighted particles (S, π) , where s is a possible state vector, and π is its weight that reflects the plausibility of s as a representation of the true state of the system.
- No assumptions on the distributions involved, so can represent arbitrary, multi-modal distributions. It's more useful when tracking in presence of clutter than Kalman filter (uni-model)

Condensation Example



State samples



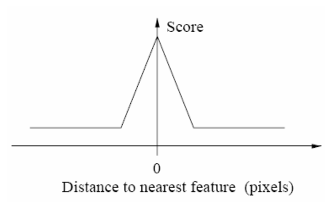
Mean of weighted state samples

From Isard and Blake 1998

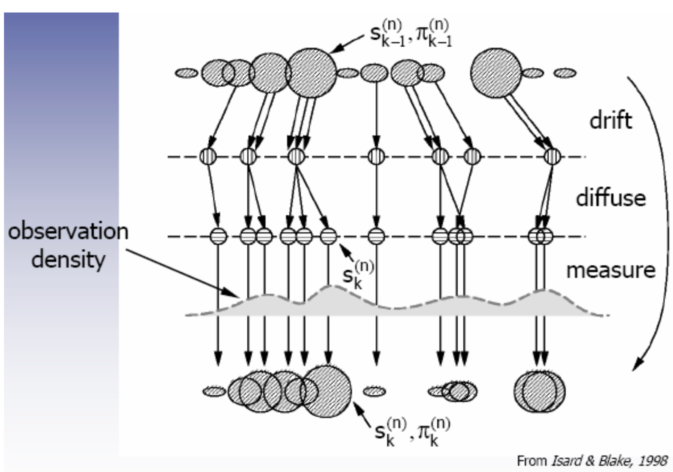
Estimation by Particle Filtering

Repeat N times for each images

- Randomly select a sample **s** from the particle set, based upon corresponding weight π
- Predict the motion of the sample in the state space using a stochastic state evolution model
- Measure the plausibility of the evolved sample by comparing its position to that of observations made from the image. Generate a new weight



Condensation: conditional density propagation



Performance Enhancements

(increase efficiency of particle filter)

- Partitioned sampling

$$s_1 = [y_0, \tan\varepsilon, W, \theta]^T \quad s_2 = [C_0, C_1]^T$$

S1 describes the width of the road ahead of the vehicle and the camera's position relative to the centre of the road (straight road' properties) S2 describes the curvature properties.

- Importance sampling

Introduce new samples into the particle set to make them concentrated around "more likely" parts of the state space.

Experiment Result

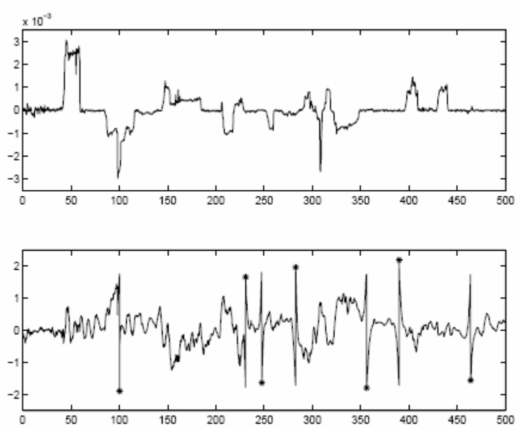


Figure 7. Estimates vs time (s). Top: Road curvature C_0 (m^{-1}). Bottom: Offset y_0 (m)(lane changes are marked with '*')).

Achievements and Problems

- Runs at 10.5 fps
- No ground truth data for lane tracking systems, so quantitative measures of algorithm performance are not given
- Brightness sunlight and specularities from wet roads are a problem
- Curve estimation lags because dynamical model does not predict non-random changes of curvature.

Reference

- M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 1998
- J. P. MacCormick and M. Isard. Partitioned sampling, articulated objects and interface – quality hand tracking. In *Proc. ECCV 2000*. Springer Verlag, 2000

Questions?