

Multiple Condensation filters for road detection and tracking

Li Bai · Yan Wang · Michael Fairhurst

Received: 4 August 2006 / Accepted: 17 August 2008 / Published online: 3 March 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract In this paper, we present a novel and robust road tracking system for vision-based personal navigation. Novelty of the work includes the use of multiple Condensation filters to track the road of arbitrary shape and automatic switching between trackers according to road conditions. The approach allows the road to be represented as a simple hyperbola. It also supports the representation of the road as a sequence of connected arcs/segments so that information from a digital map can be integrated into tracking. The parameters of the hyperbola road model are estimated using multiple vanishing points located in image strips. The road tracking method is robust in dealing with complex road shapes, background clutters, shadows, and road markings. Experiments using real videos demonstrate the robustness of our approach.

Keywords Road tracking · Vanishing point · Computer vision · Multiple Condensation tracker

1 Originality and Contribution

Novelty of the work includes (1) a robust curve fitting approach based on multiple vanishing point detection and (2) the use of multiple Condensation filters to track the road

of arbitrary shape and automatic switching between trackers according to road conditions. A significant contribution of this work lies in estimating the road model using high level and high confidence road features, and the exploitation of knowledge about the road geometry under perspective projection. Despite some potential drawbacks, when the ground plane or the parallel road boundary assumption is violated, our method provides a powerful tool for use in real-time road tracking.

2 Introduction

Video based systems have been used for road surveillance, e.g., recognition of vehicles and for autonomous vehicles and driver assistance, e.g., recognition of roads and obstacles [1–3]. In this paper, we describe a new method for road tracking—a multiple tracker framework, based on the Condensation method [4]. The work is part of the effort in developing a new generation of navigation systems using video inputs.

Many vision-based road detection systems instantiate road models from features extracted from the image. These features extraction methods are often based on corners, edges, ridges, colours and textures of the image [5–14]. For example, Rasmussen et al. [12] use information about texture orientations at pixels, obtained using Gabor filters, to detect the road. High level features such as road junctions and statistical models have also been used to assist road detection [15–17].

Once road features are detected, a model of the road can be built for vision-based road tracking. Some used straight lines to model the road [18] while others employed more complex models such as cubic B-Splines [19, 20], clothoid [21–23], parabola [24], and hyperbola [25–27].

L. Bai (✉) · Y. Wang
School of Computer Science and IT,
University of Nottingham, Nottingham, UK
e-mail: bai@cs.nott.ac.uk

Y. Wang
e-mail: yqw@cs.nott.ac.uk

M. Fairhurst
Electronics Department, University of Kent, Kent, UK
e-mail: m.c.fairhurst@kent.ac.uk

For example, Southall and Taylor [21] describe a collision warning system in which the road is modelled as a clothoid, approximated by a polynomial, in order to deal with road curvature and changes of curvature. However, Cramer et al. pointed out that a clothoid model is unreliable when the road curvature varies significantly [28, 29]. Instead, they model the road as connected arc segments with the aid of digital map data.

Our methodology for road detection and tracking takes into account the fact that a complex road shape can be decomposed into a sequence of simple shapes at road transition points (where there is a significant change of road curvature). Rather than labouring a single tracker to do all the work, we switch trackers at road transitions. The method is more tuned to road conditions and is more robust. It also supports the representation of the road as a sequence of connected arcs/segments so that information from a digital map can be integrated into tracking.

The remainder of this paper is organised as follows: Sect. 2 describes a method for estimating the road model. This includes methods for eliminate spurious line segments in images and multiple vanishing point detection and fitting, in order to produce an accurate estimate of a hyperbolic road model. Section 3 integrates the estimated hyperbolic road model with a single Condensation tracker first, then introduces a road transition model in order to handle the change of road curvature effectively, and finally describes the operations of road tracking using multiple Condensation trackers. Section 4 shows experimental results using real road videos. Section 5 draws conclusion to the paper.

3 Estimating the road model

The method for estimating the road model works as follows. First, we divide an image into horizontal strips and detect road boundary features. Second, we cluster road boundary features into left and right boundary groups. Third, we detect multiple vanishing points and the horizon necessary for estimating the parameters of the hyperbolic road model. The method was first introduced in [30] and is described here for completeness.

3.1 Clustering line segments into road boundary groups

We divide an image up into a number of horizontal strips containing only straight road boundaries and detect line segments in each strip using the Canny operator and the Hough transform. Some of the lines found may not belong to road boundaries so a procedure to eliminate these spurious line segments is necessary. We use the circular road model proposed in [17] to generate road samples in order to estimate whether a line segment is part of a road boundary.

The model is represented as $(h, C_0, l_0, \varphi, \theta, W)$: h height of the camera; C_0 road curvature; l_0 lateral offset of the camera from the lane axis; φ pitch angle of the camera relative to the ground plane; θ yaw angle of the camera relative to the lane axis; W lane width. Suppose (u, v) represents the image coordinate. Given the intrinsic and extrinsic parameters of the camera, the position and pose of the camera relative to the road, and the road parameters (width, curvature, etc.), the projections of the road boundary points in the image can be calculated. This involves estimating three attribute values for each road boundary point P_i the image coordinates u_i , and v_i , and gradient g_i of the tangent to the road boundary at (u_i, v_i) . As the image is divided into horizontal strips, we know for each strip the v_i coordinate, so only u_i and g_i need to be estimated. u_i can be determined by

$$u_i = e_u \left[\frac{e_v h C_0}{2(v_i - v_0 + e_v \varphi)} + \frac{\gamma W/2 - l_0}{e_v h} (v_i - v_c + e_v \varphi) + \theta \right] + u_c \quad (1)$$

where $\gamma = -1$ for the left boundary point, $\gamma = 1$ for the right boundary point; f is camera focal length; d_u, d_v width and height of the pixel; $e_u = f/d_u$ horizontal focal length in pixels; $e_v = f/d_v$, vertical focal length in pixels; (u_c, v_c) the centre of the image.

The proof of (1) is as follows: road boundaries can be approximated by a pair of parabolas on the ground plane expressed in camera coordinates, as depicted in Fig. 1, with the same C_0 and φ :

The pair of parabolas can be formulated as:

$$x = -\frac{1}{2} C_0 y^2 - y \tan \varphi + x_0 + \frac{\gamma W}{2}. \quad (2)$$

Assuming a 3D point P is projected onto an image pixel (u_i, v_i) under perspective projection and the image centre $(u_0, v_0) = (0, 0)$, the following equations hold:

$$\frac{v_i}{e_v} = \tan \left(\arctan \frac{z_0}{y} - \alpha \right), \frac{u_i}{e_u} = x / \left(\sqrt{y^2 + z_0^2} \cos \left(\arctan \frac{v_i}{e_v} \right) \right), \quad (3)$$

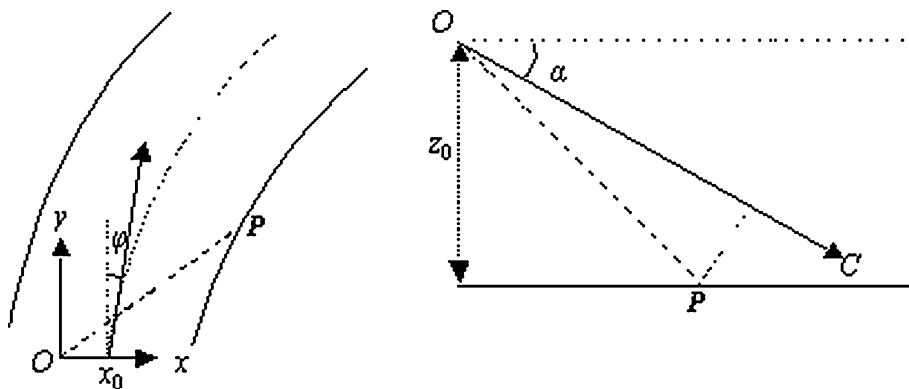
Substitute x, y in (2) with those derived from (3) and approximate $\tan \alpha$ and $\tan \varphi$ by α and φ , respectively, we obtain (1).

Differentiating u_i in (1) in terms of v_i gives the gradient:

$$g_i = e_u \left[-\frac{e_v h C_0}{2(v_i - v_c + e_v \varphi)^2} + \frac{1}{e_v h} \left(-l_0 + \frac{\gamma W}{2} \right) \right]. \quad (4)$$

Each of the other parameters in the circular road model $(h, C_0, l_0, \varphi, \theta, W)$ is uniformly sampled from a certain range (e.g., $0.8 \text{ m} \leq h \leq 1.2 \text{ m}$, $-0.01 \leq C_0 \leq 0.01$, $-5^\circ \leq \varphi \leq 5^\circ$, $0 \leq \theta \leq 5^\circ$, $3.5 \text{ m} \leq W \leq 4.5 \text{ m}$ and $-W/2 \leq l_0 \leq W/2$).

Fig. 1 Road boundaries are approximated by a pair of parabolas on the ground plane. The camera’s attitude is also shown. O is the camera’s optical centre. OC is the optical axis. x_0 is the offset of the camera from the centre of the road. z_0 is the height of the camera over the ground plane. α and φ are the inclination and yaw angle, respectively



The circular road model is then used to generate road samples. The mean \bar{u}_i and variance $\sigma^2(u_i)$ of u_i for each v_i are calculated over the road samples as well as the average \bar{g}_i and variance $\sigma^2(g_i)$ of g_i according to (5). Confidence intervals for u_i and g_i can be then defined as $[u_i - \sigma(u_i), u_i + \sigma(u_i)]$ and $[g_i - \sigma(g_i), g_i + \sigma(g_i)]$, respectively, for the boundary points corresponding to v_i . Line segments in the image can be clustered into left and right road boundaries according to which confidence interval the end points of a line segment fall into.

3.2 Detecting multiple VPs

Using a single vanishing point (VPs) [31–33] to estimate the road boundary is error prone because boundaries of a complex road do not intersect at a point—they vanish at a line, called the vanishing line or horizon. In order to detect the horizon we need to detect multiple VPs.

Since there are still spurious line segments in each road boundary group, not all combinations of two line segments, one from each road boundary, are valid pairs for VP calculation. We therefore detect a VP in each image strip using the least median squares (LMedS) method [34]. LMedS randomly samples a number of points from a data set and uses these points to estimate model parameters. The sampling process is repeated for at least m times and m is related to the percentage of outliers in the data set. By selecting the estimation having the least median of square errors, LMedS is able to exclude outliers up to 50%. We also use the covariance matrix about valid pairs of line segments from training road samples to eliminate unlikely combinations of left and right road boundary segments.

Suppose (u, v) represents image coordinates. For the i th image strip, let $\mathbf{x}_i = (u_{l,i}, u_{r,i}, g_{l,i}, g_{r,i})^T$, where $u_{l,i}$ and $u_{r,i}$ are the u coordinates of the two intersection points of the horizontal centre line of the i th image strip with a left and a right road boundary, respectively, and $g_{l,i}$ and $g_{r,i}$ are the gradients at the two intersection points, respectively. The distribution of these pairs of intersection points can be

characterized by mean \bar{x}_i and covariance matrix $C\mathbf{x}_i$ which are calculated from training road samples. The multiple VP detection method can be stated as follows:

3.3 Fitting multiple VPs to estimate the horizon

To locate the horizon (or the vanishing line), we use the M-Estimator [35] to fit a line to the VPs detected. Suppose we have a set of VP candidates, $\{vp_i := (u_{VP_i}, v_{VP_i}) | i = 0, \dots, N\}$, each is for one image strip. The M-Estimator for estimating the v coordinate v_H of the horizon reduces the effect of spurious VPs by minimising a so-called ρ function of residuals:

$$\rho(r_i) = \frac{r_i^2}{\sigma^2 + r_i^2} = \frac{(v_H - v_{VP_i})^2}{\sigma^2 + (v_H - v_{VP_i})^2} \tag{5}$$

where r_i is the i th residual, $\sigma = 1.4826 * \text{median}(\|r_i\|)$. The M-Estimator aims to find a solution to (6):

$$\sum_i \frac{d\rho(r_i)}{dv} = \sum_i \frac{d\rho}{dr_i} \frac{dr_i}{dv} = \sum_i \left(\frac{1}{r_i} \frac{d\rho}{dr_i} \right) r_i \frac{dr_i}{dv} = \sum_i w(r_i) r_i \frac{dr_i}{dv} = 0 \tag{6}$$

where $w(r_i)$ is a weight function with the form:

$$w(r_i) = \frac{2\sigma^2}{[\sigma^2 + (v_H - v_{VP_i})^2]^2} \tag{7}$$

At each iteration the residuals are re-weighted and the solution is re-computed according to (8):

$$v_{H,t+1} = \left(\frac{\sum_i 2\sigma_t^2 v_{VP_i} / (\sigma_t^2 + (v_{H,t} - v_{VP_i})^2)^2}{\sum_i 2\sigma_t^2 / (\sigma_t^2 + (v_{H,t} - v_{VP_i})^2)^2} \right) \tag{8}$$

Once multiple VPs have been detected, a line fitting these points will be the horizon.

3.4 Estimating road parameters

We model the road boundaries in the image plane as a pair of hyperbolas:

$$u - u_H = a(v - v_H) + \frac{b}{v - v_H} \tag{9}$$

where (u, v) is the coordinate of points on the left or right road boundary; a is the inverse tangent of an asymptote of the hyperbola; (u_H, v_H) is the other asymptote on the horizon where the two asymptotes intersect; v_H is the horizon which has been determined in Sect. 2.3; b is linearly related to the road curvature C_0 in the 3D world. A more general form of the road shape can be represented by the polynomial [36, 37]: $u = \sum_{i=0}^n a_i v^{1-i}$, where the coordinate origin is a point on the horizon. The hyperbola is a special case, when $n = 2$. From (1) and (9) we have

$$v_H = v_c - e_v \varphi \tag{10}$$

$$a = \frac{e_u(\gamma W/2 - l_0)}{e_v h} \tag{11}$$

$$b = (e_u e_v h C_0)/2 \tag{12}$$

$$u_H = u_c + e_u \theta \tag{13}$$

The diagram of the hyperbolic road model is shown in Fig. 2.

We now estimate u_H , a and b to determine the road shape using the VPs. Differentiating (9) with respect to v gives:

$$du/dv = a - b/(v - v_H)^2 \tag{14}$$

Suppose that a road boundary point $P_i (u_{P_i}, v_{P_i})$ has the same v image coordinate as the centre line of the i th image strip, i.e., $v_{P_i} = v_i$. Linking the i th VP VP_i to P_i gives the gradient to P_i :

$$(du/dv)_i = (u_{VP_i} - u_{P_i}) / (v_H - v_{P_i}) = a - b / (v_{P_i} - v_H)^2 \tag{15}$$

$$u_{P_i} = u_H + a(v_{P_i} - v_H) + b(v_{P_i} - v_H)^{-1} \tag{16}$$

Replacing u_{P_i} in (15) by (16), we have the following linear equations:

$$\mathbf{AX} = \mathbf{B} \tag{17}$$

where $\mathbf{A} = \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix} = \begin{bmatrix} v_{P_1} - v_H & 2 \\ \vdots & \vdots \\ v_{P_n} - v_H & 2 \end{bmatrix}$ $\mathbf{X} = \begin{bmatrix} u_H \\ b \end{bmatrix}$ and

$$\mathbf{B} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} u_{P_1}(v_{P_1} - v_H) \\ \vdots \\ u_{P_n}(v_{P_n} - v_H) \end{bmatrix}.$$

Note that a has been cancelled out and (17) involves only two unknowns b and u_H , which can be easily solved using the least squares method. However, detected VPs may still contain outliers or non-Gaussian noise, so a robust fitting method is needed. Thus a weight matrix is multiplied to both sides of the linear system:

$$\mathbf{WAX} = \mathbf{WB}$$

where $\mathbf{W} = \text{diag}[w_1 \cdots w_n]$.

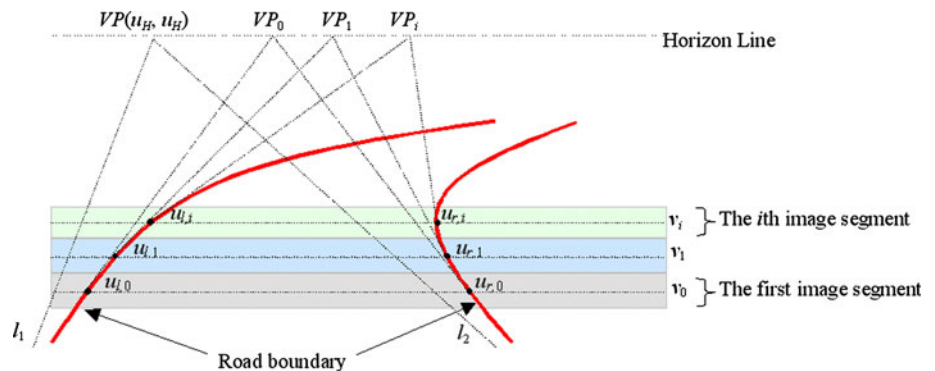
An iterative robust fitting algorithm is performed until the solution converges or maximal iteration has been reached, according to the following procedure:

1. Initialize X_0 with $W = I$, and set $j = 1$;
2. For all $i (1 \leq i \leq n)$, compute $w_{i,j} = \varphi' \left(\left(\frac{a_i^T X_{j-1} - b_i}{\sigma} \right)^2 \right)$;
3. Solve the linear system $W_j \mathbf{A} X_j = W_j \mathbf{B}$;
4. If $\|X_j - X_{j-1}\| > \varepsilon$ and $j < j_{\max}$, set $j = j + 1$, and go to 2, else $X = X_j$

where Φ is the error probability density function (pdf) of the detected VPs.

The final step is to estimate a_l and a_r . A search is carried out in the space containing all possible values of a_l and a_r . a_l varies from 0 to 3, and a_r from -3 to 0 with steps of 0.1. The procedure produces a set of hypothesised hyperbolas evenly distributed in the image. We then find the best hyperbola by measuring the spatial proximity of estimated model to edge features in the edge map (the measurement model is discussed in Sect. 3). The procedure can also be replaced by a Hough transform— a_l and a_r can be voted by all edge points below the horizon, but this may not be efficient as noise is present in the image.

Fig. 2 The diagram of the hyperbola road model. The left road boundary has two asymptotes: l_1 and the horizon. The right road boundary has two asymptotes: l_2 and the horizon. l_1 and l_2 vanish at VP (u_H, v_H)



3.5 Road detection results

We test our road detection methods using real road videos. We demonstrate the results using three representative videos taken from motorways, suburban roads and rural roads respectively. We randomly pick hundreds of frames from each video and compare the detected road boundaries in them with real road boundaries displayed in the video frames. Detection rate is calculated based on how close the detected boundaries are to the real road boundaries. We show the test results in Table 1.

Figure 3 demonstrates the robustness of our approach under adverse conditions such as occlusion by other vehicles, road markings and shadows. Since at least some vanishing points can be correctly detected, and they aggregate around the horizon, a robust estimator is able to eliminate outliers.

We discuss the effect of the image strip height on vanishing point detection. The higher the image strip, the more line features can be detected. However, too high an image strip will degrade the accuracy of vanishing point detection, especially when the road boundary has a significant curvature. This proves the need for dividing the image into horizontal strips. On the other hand, if the height of the image segment is too small, line features of

the road segment may be too short to be detected. Basically, two conditions have to be satisfied: 1. the height of an image strip has to be small enough such that road boundaries contained in the strip can be approximated by straight lines. 2. the detected line segments are long enough to be distinguished from image noise. A compromise thus has to be made. Figure 4 shows the results of VP detection by varying the height of an image strip. It is evident that the positions of the VPs detected vary as the height varies.

4 Tracking the road

First, we show how to integrate the hyperbolic road model with a Condensation tracker. Second, we develop a road transition model to handle large road curvature. Third, we integrate the road model and road transition model with multiple Condensation trackers to track the road in real-time for vision-based navigation applications.

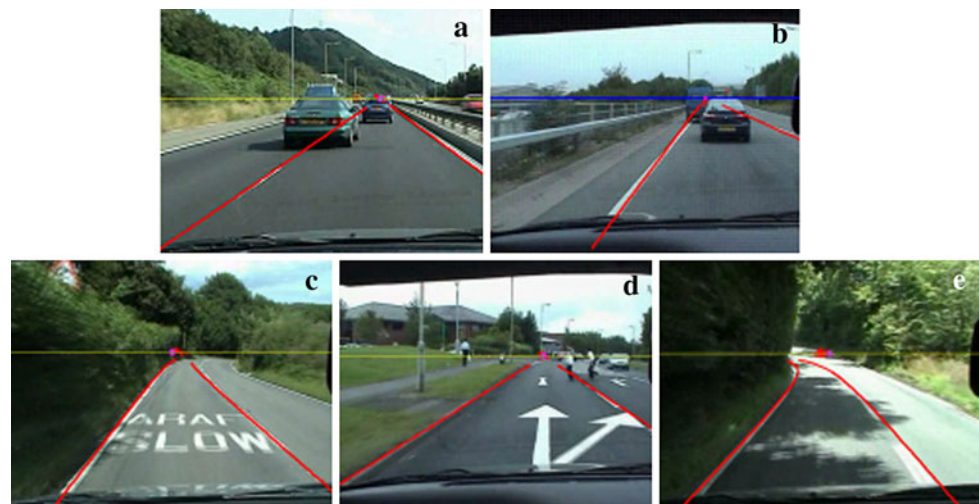
4.1 Road tracking with a single Condensation tracker

The main feature of Condensation is Bayesian filtering, which recursively estimate a posterior density of the object’s state using the Bayesian inference:

Table 1 Detection rates for three road video sequences

Video sequences	Number of frames	% of frames with shadows and road markings	% of frames with occlusions	% of road without boundary markings	Detection rate (%)
Motorway	200	3.5	5	0	89.5
Suburban	300	6	2	3	93.0
Countryside	200	54	0	6	94.5

Fig. 3 a, b Partial occlusion by cars. c, d Signs on the road. e Shadows on the road



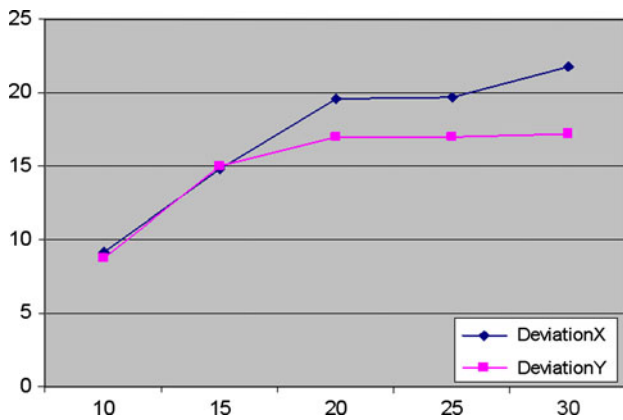


Fig. 4 Deviation of u (DeviationX) and v (DeviationY) of detected vanishing point candidates. The horizontal axis is the size (height) of the image segment in pixels. The vertical axis is deviation in pixels

$$p(\mathbf{x}_t|\mathbf{z}_t) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{t-1}) \tag{18}$$

where $p(\mathbf{z}_t|\mathbf{x}_t)$ is the *likelihood* and $p(\mathbf{x}_t|\mathbf{z}_{t-1})$ is the prior density derived from the previous posterior density $p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})$ and a *dynamical model* or a motion prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$

$$p(\mathbf{x}_t|\mathbf{z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})d\mathbf{x}_{t-1} \tag{19}$$

Bayesian filtering is also the root of Kalman filtering, but the Condensation tracker differs from the Kalman filter in that it can estimate object distributions which are not Gaussian. It does this by generating a set of particles at time t , $\{\mathbf{x}_{n,t}, w_{n,t}\}, n = 1, \dots, N$, where $\mathbf{x}_{n,t}$ represents the n th observation of the object state at time t , $w_{n,t}$ is the probability for $\mathbf{x}_{n,t}$ to be the true state and N is the number of particles. These particles together approximate the probability density of the object’s state. Condensation uses a sampling scheme called Sampling Importance Resampling (SIR), which recursively draws samples from an importance proposal density function $q(\mathbf{x})$ and weights them using equation (20) to arrive at $\tilde{p}(\mathbf{x}_t|\mathbf{z}_t)$, the approximation of the true posterior density estimation $p(\mathbf{x}_t|\mathbf{z}_t)$.

$$w_{n,t} \propto \frac{p(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1}, \mathbf{z}_t)}{q(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1}, \mathbf{z}_t)} = \frac{p(\mathbf{z}_t|\mathbf{x}_{n,t})p(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1})}{q(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1}, \mathbf{z}_t)} \tag{20}$$

Initially, Condensation tracker assigns each possible state the same probability. At each time step, a new set of particles is sampled from those with higher probabilities in the previous step and propagated according to the dynamics model. This implies that it actually takes the dynamics model or motion prior as the proposal density: $q(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1}, \mathbf{z}_t) = p(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1})$. Subsequently, according to (20), $w_{n,t} \propto p(\mathbf{z}_t|\mathbf{x}_{n,t})$. This means that the probabilities are updated based on likelihood, which can be represented by some measurement related to the presence of image

features in the vicinity of the predicted particles. Finally, the true object state can be estimated as the weighted moments of particles at each time step t . In summary, to use a Condensation tracker for a specific application, we need to define an application specific dynamical and likelihood models.

4.2 The dynamical model for road tracking

The road model estimated in the previous section is integrated with the Condensation tracker described in [4] for road tracking. The state of the object to be tracked can be represented by a vector $\mathbf{x} = (a_l, a_r, b, u_H, v_H)^T$, whose elements are the parameters of the road shape model in Equation (9). The dynamics of the states have the following form:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}_t \tag{21}$$

where \mathbf{x}_t is the state vector at time t ; \mathbf{A} is the state transition matrix; \mathbf{B}_t is the stochastic part. We model the dynamics simply as a Gaussian random walk, so that $\mathbf{A} = \mathbf{I}$ and \mathbf{B}_t shall carry the stochastic variation $\Delta\mathbf{x}_{t-1,t}$ of \mathbf{x} from time $t - 1$ to time t . Now \mathbf{B}_t is derived as follows. Notice that each state variable in \mathbf{x} can be expressed using parameters $W, h, l_0, \theta, \varphi$ and C_0 via Equations (10)–(13). It is not difficult to see that some state variables are implicitly correlated to each other (such as a_r and a_l) via (10)–(13), so are their variations from $t - 1$ to t . It is thus desirable to incorporate such correlation information into \mathbf{B}_t . Let $\mathbf{y} = (W l_0 \theta \varphi C_0)^T$ and assume h is a constant. Parameters in \mathbf{y} can be regarded as independent to each other. Indeed equations (10)–(13) specify a non-linear transformation from \mathbf{y} to \mathbf{x} : $\mathbf{x} = \Phi(\mathbf{y})$. We shall derive $\Delta\mathbf{x}_{t-1,t}$ from the variation $\Delta\mathbf{y}_{t-1,t}$ of \mathbf{y} from time $t - 1$ to t . $\Delta\mathbf{x}_{t-1,t}$ has the following relation with $\Delta\mathbf{y}_{t-1,t}$ by linearizing Φ :

$$\Delta\mathbf{x}_{t-1,t} = \mathbf{x}_t - \mathbf{x}_{t-1} \approx \mathbf{H}_t(\mathbf{y}_t - \mathbf{y}_{t-1}) = \mathbf{H}_t\Delta\mathbf{y}_{t-1,t} \tag{22}$$

where

$$\mathbf{H}_t := \left. \frac{\partial(a_l a_r b u_H v_H)}{\partial(W l_0 \theta \varphi C_0)} \right|_t = \begin{bmatrix} \frac{-e_u}{2e_v h} & \frac{-e_u}{e_v h} & 0 & 0 & 0 \\ \frac{2e_u}{2e_v h} & \frac{-e_u}{e_v h} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{e_u e_v h}{2} \\ 0 & 0 & e_u & 0 & 0 \\ 0 & 0 & 0 & -e_v & 0 \end{bmatrix} \tag{23}$$

The variation $\Delta\mathbf{y}_{t, t+1}$ is modelled as a set of independently distributed normal variables:

1. $\Delta W \sim N(0, \sigma_W^2), \sigma_W = 0.1 \text{ m}$,
2. $\Delta C_0 \sim N(0, \sigma_{C_0}^2), \sigma_{C_0} = 10^{-3} \text{ m}^{-1}$,
3. $\Delta \varphi \sim N(0, \sigma_\varphi^2), \sigma_\varphi = 10^{-3} \text{ rad}$,
4. $\Delta \theta \sim N(0, \sigma_\theta^2), \sigma_\theta = 10^{-3} \text{ rad}$,
5. $\Delta l_0 \sim N(0, \sigma_{l_0}^2), \sigma_{l_0} = 0.1 \text{ m}$.

Finally \mathbf{B}_t may be written as:

$$\mathbf{B}_t = \Delta \mathbf{x}_{t,t+1} = \mathbf{H}_t \boldsymbol{\Xi} \boldsymbol{\omega} \tag{24}$$

where $\boldsymbol{\Xi} := \text{diag}[\sigma_w \sigma_{l_0} \sigma_\theta \sigma_\phi \sigma_{C_0}]$ and $\boldsymbol{\omega}$ is a 5×1 vector with the i th element $\boldsymbol{\omega}^{(i)} \sim N(0,1)$. Note that \mathbf{H}_t has incorporated the correlations between variations of state variables. It is easy to show that $x_t \sim N(\mathbf{x}_{t-1}, \boldsymbol{\Sigma}_x)$ where

$$\boldsymbol{\Sigma}_x = \mathbf{H}_t^T \boldsymbol{\Xi}^T \boldsymbol{\Xi} \mathbf{H}_t \tag{25}$$

4.3 The likelihood model for road tracking

In [4], a general form of the likelihood measurement model is proposed

$$p(\mathbf{z}|\mathbf{x}) \propto \exp\left(-\frac{1}{2r} \int_0^L f(\mathbf{z}(s) - \mathbf{x}(s); \mu) ds\right) \tag{26}$$

where $f(v; \mu) = \min(v^2, \mu^2)$; r is a variance constant and $z(s)$ is the closest image feature to the predicted state $x(s)$. μ is the spatial scale constant defining the maximal scale of the search interval for image features. So the likelihood measures how close the predicted feature is to the actual feature found in the image. As shown in Fig. 5, a set of points are sampled from the estimated road boundary. We search for the nearest edge point \mathbf{Q}_i to each sampled point \mathbf{P}_i on a horizontal line with the sampled point at the centre and measure the distance between them. We also take into account the similarity between the normal \mathbf{n}_{P_i} at the sampled point \mathbf{P}_i with \mathbf{n}_{Q_i} at the edge point \mathbf{Q}_i . The likelihood model is thus

$$p(\mathbf{z}|\mathbf{x}) \propto \exp \sum_i \left(-|\mathbf{n}_{P_i} \times \mathbf{n}_{Q_i}| - \frac{\|\mathbf{P}_i - \mathbf{Q}_i\|^2}{2\sigma^2} \right) \tag{27}$$

If no edge point is found, $\|\mathbf{P}_i - \mathbf{Q}_i\|$ is set to σ (5 pixels for our case), the upper bound of μ in (26), and $|\mathbf{n}_{P_i} \times \mathbf{n}_{Q_i}|$ is set to 1.

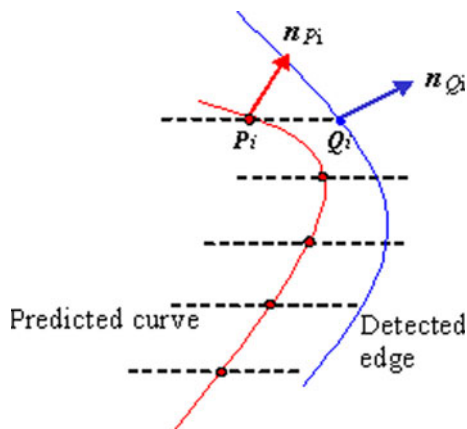


Fig. 5 The likelihood model uses edge cues near sampled points from the predicted road curve

4.4 The road transition model

The hyperbolic road model cannot handle well road transitions between road segments with different curvatures. This results in poor tracking performance. For example, in the diagram on the left in Fig. 6, the real road boundaries are curve AB connected to curve BC and curve $A'B'$ connected to curve $B'C'$ but the hyperbolic road model is shown as a pair of curves (dashed lines in the diagram). We therefore need to define a geometric model of road transition in order to handle the transition between road segments with different curvatures, and integrate this to the hyperbola model.

The projection of the road transition on the image is shown in the diagram on the right in Fig. 6. The road transition is modelled by two road segments connected at \mathbf{B} and \mathbf{B}' with G1 continuity (i.e., share the same tangent \mathbf{t}_B and $\mathbf{t}_{B'}$ at \mathbf{B} and \mathbf{B}' respectively). The projections of AB and $A'B'$ are approximated by a pair of hyperbolas in the lower part of the image (lower hyperbolas), defined in (9). The projections of BC and $B'C'$ are approximated by another pair of hyperbolas in the upper part of the image (upper hyperbolas). $P_{vc1}(u_{H1}, v_{H1})$ is the vanishing point of the asymptotes of the lower hyperbolas. \mathbf{P}_B and $\mathbf{P}_{B'}$ are projections of \mathbf{B} and \mathbf{B}' respectively. $P_{vc2}(u_{H2}, v_{H2})$ is the vanishing point of the asymptotes of the upper hyperbolas (projections of BC and $B'C'$). Thus the parameters for the lower and upper hyperbolas are $(a_{l1}, a_{r1}, b_1, u_{H1}, v_{H1})$ and $(a_{l2}, a_{r2}, b_2, u_{H2}, v_{H2})$, respectively. According to (9) and its derivative (14), the parameters of the lower and upper hyperbolas must satisfy the following relations due to G1 continuity at transition points:

$$a_{l2} = a_{l1} + \frac{b_2 - b_1}{(v_B - v_{H1})^2} \tag{28}$$

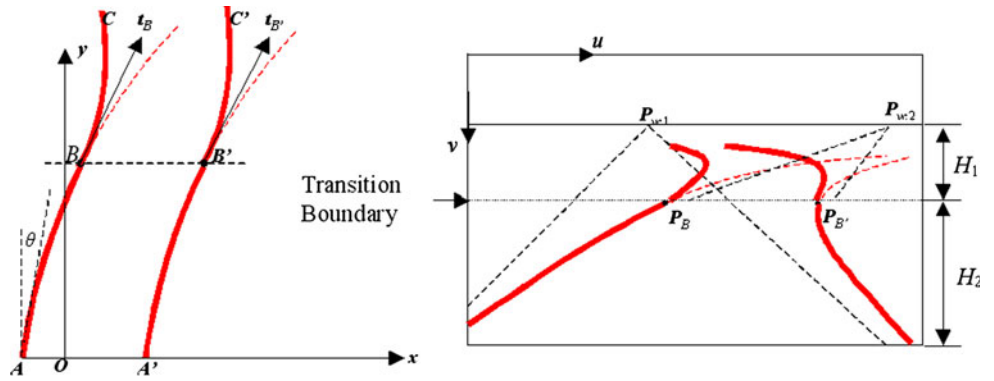
$$a_{r2} = a_{r1} + \frac{b_2 - b_1}{(v_B - v_{H1})^2} \tag{29}$$

$$v_{H1} = v_{H2} \tag{30}$$

$$u_{H2} = u_{H1} + 2 \frac{b_1 - b_2}{v_B - v_{H1}} \tag{31}$$

where v_B is the v coordinate of \mathbf{P}_B and $\mathbf{P}_{B'}$. Thus, in contrast to the road shape parameters defined in Sect. 3.2, which is (a_l, a_r, b, u_H, v_H) for single pair of hyperbolas, only two more free parameters b_2 and v_B are needed to model the road transition. Therefore, the road transition model is $(a_{l1}, a_{r1}, b_1, u_{H1}, v_{H1}, b_2, v_B)$. Indeed, v_B defines a transition boundary which split the road image to two strips: lower and upper with different heights (e.g., strips with height H_1 and H_2 in Fig. 6). We look for the first road segment in the lower strip and the second in the upper strip in the image. Alternatively, v_B can be replaced by a ratio $r = H_1/(H_1 + H_2)$, where

Fig. 6 *Left* a road transition model on the ground plane. *Right* image projection of the transition model



$H_1 + H_2 = \text{ImageHeight} - v_{H_1}$ and $0 \leq r \leq 1$, indicating the proportion of the upper image strip in the image. Finally the transitional model is defined as $(a_{11}, a_{r1}, b_1, u_{H_1}, v_{H_1}, b_2, r)$.

4.5 Road tracking with multiple trackers

In this section, we will incorporate the road transition model in Sect. 3.4 into the hyperbolic model for tracking using multiple Condensation trackers. The new state vector has now become $(a_{11}, a_{r1}, b_1, u_{H_1}, v_{H_1}, b_2, r)$ for handling road transition. Higher dimensional feature space usually means we need to sample more particles in order to approximate the probability density function. We realised that this can be avoided if the state vector is partitioned into two parts and each part is handled by a different Condensation tracker. The state vector can be partitioned into $\mathbf{x}_c := (a_{11}, a_{r1}, b_1, u_{H_1}, v_{H_1})$, modelling the lower hyperbolas, and $\mathbf{x}_s := (b_2, r)$, modelling the upper hyperbolas. We assign a tracker C_1 to \mathbf{x}_c and C_2 to \mathbf{x}_s . Each tracker evolves its own state vector but each has the knowledge of the full state vector to be able to measure likelihoods for their own particles. We model the dynamics of \mathbf{x}_s as a Gaussian random walk: $(\mathbf{x}_s)_t \sim N((\mathbf{x}_s)_{t-1}, \Sigma_{x_s})$ where Σ_{x_s} is the covariance matrix of \mathbf{x}_s . In our experiment we set $\Sigma_{x_s} = \text{diag}(500^2, 0.1^2)$.

Road detection in Sect. 2 provides an automatic initialization of C_1 . C_2 is initialized after C_1 has been initialized. At the beginning we assume C_2 has the same curvature as C_1 , so b_2 is set to b . r is sampled uniformly from the interval $[0, 1]$. At each time step, for C_1 we draw N_1 particles from the posterior density approximation of C_2 at the previous time step:

$$\tilde{p}_{C_2}(\mathbf{x}_{t-1} | \mathbf{z}_{t-1}) = \{\mathbf{x}_{n,t-1}, w_{n,t-1}\}, n = 1, \dots, N_2$$

For each of these particles a new state is predicted using the dynamics of the state \mathbf{x}_c . Since C_1 maintains the full state vector, we can measure the likelihoods by sampling points on both estimated road segments. When tracking using C_2 , N_2 samples $\{\mathbf{x}_{n,t}\}$, $n = 1, \dots, N_2$ are drawn from the current posterior density approximation of the state of

C_1 : $\tilde{p}_{C_1}(\mathbf{x}_t | \mathbf{z}_t) = \{\mathbf{x}_{n,t}, w_{n,t}\}$, $n = 1, \dots, N_1$. The current posterior density estimation of C_1 is used as the importance proposal density function q_{C_2} :

$$q_{C_2}(\mathbf{x}_{n,t} | \mathbf{x}_{n,t-1}, \mathbf{z}_t) = p_{C_1}(\mathbf{x}_{n,t} | \mathbf{z}_t) \approx \tilde{p}_{C_1}(\mathbf{x}_{n,t} | \mathbf{z}_t) \tag{32}$$

From (20), the weighting scheme for C_2 has thus become

$$\begin{aligned} (w_{n,t})_{C_2} &\propto \frac{p_{C_2}(\mathbf{z}_t | \mathbf{x}_{n,t}) p_{C_2}(\mathbf{x}_{n,t} | \mathbf{x}_{n,t-1})}{p_{C_1}(\mathbf{x}_{n,t} | \mathbf{z}_t)} \\ &= \frac{p_{C_2}(\mathbf{z}_t | \mathbf{x}_{n,t}) p_{C_2}(\mathbf{x}_{n,t} | \mathbf{x}_{n,t-1})}{p_{C_1}(\mathbf{z}_t | \mathbf{x}_{n,t}) p_{C_1}(\mathbf{x}_{n,t} | \mathbf{x}_{n,t-1})} \\ &= \frac{p_{C_2}(\mathbf{z}_t | \mathbf{x}_{n,t}) f_N(\mathbf{x}_s)_{n,t}; (\mathbf{x}_s)_{n,t}, \Sigma_{x_s}}{(w_{n,t})_{C_1} f_N(\mathbf{x}_c)_{n,t}; (\mathbf{x}_c)_{n,t-1}, \Sigma_{x_c}} \end{aligned} \tag{33}$$

where Σ_{x_c} is (25), and $f_N(\alpha; \beta, \Sigma)$, $\alpha, \beta \in R^M$, is the evaluation function of a Gaussian density having the following form:

$$f_N(\alpha; \beta, \Sigma) := \frac{1}{(2\pi)^{M/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\alpha - \beta)^T \Sigma^{-1} (\alpha - \beta)\right\} \tag{34}$$

Note that in (35) C_2 weight updating takes into account C_1 weight updating, which means C_2 uses the knowledge about the state density estimates by C_1 . Note that N_1 and N_2 can be different and we assign more particles for the tracker with a state space of higher dimension.

Finally, the upper hyperbolas will become the lower hyperbolas if the proportion between the upper hyperbolas and the lower hyperbolas is high and a new pair of upper hyperbolas will be initialised in order to search for forthcoming transitions. This is indicated by r . In detail, for each particle in C_2 , if r is larger than a threshold r_{takeover} , we work out the upper hyperbola parameters $a_{12}, a_{r2}, u_{H2}, v_{H2}$ using equations (29)–(32) and update the lower hyperbolas as $\mathbf{x}_c = (a_{12}, a_{r2}, u_{H1}, v_{H1}, b_2)^T$; (2) r is uniformly sampled from the range $[0, r_{\text{init}}]$ to initialize tracking for the next possible road transition. In our experiment, we set $r_{\text{init}} = 0.3$ and $r_{\text{takeover}} = 0.6$.

5 Experiments

We have tested our road detection and tracking methods using real road videos with resolution 512×384 and frame rate 25 fps. We demonstrate the results here using three videos. The first video contains road occlusion. The second contains spurious road markings. The third has shadows and significant road curvature. We used 400 particles for tracker C_1 and 200 particles for C_2 . The tests were run on a PC with a 2.4 GHz processor. Table 2 shows a breakdown of the average processing time per frame.

Some tracking results are shown in Fig. 7. Images in the first row show a two-lane motorway. Highlighted hyperbolas represent the estimated road shape. Lower and upper hyperbolas are in different colours. At frame 80, the car is moving to the left lane, which triggers re-initialization of

the tracking. The lane changing action is near completion at frame 86, and the boundaries of the current lane are successfully re-locked. Images in the second row demonstrate the robustness of our method to background clutter such as painted white arrows, zebra crossing lines and partial occlusion by people crossing the road. Images in the last two rows show some rural roads with significant curvature and shadows of trees on them. Frames 180, 295 and 305 show transitions between road segments with different curvatures.

The estimated road curvatures (by C_1) for the three videos are plotted in Fig. 8. Positive values indicate right bent and negative values left bent. Figure 9 illustrates the lateral position of the vehicle relative to the centre axis of the current lane in the three videos. The two jumps in the graph for video 1 indicate vehicle lane switching actions.

Figure 10 demonstrates the tracking performance for the three videos. We measure tracking performance in terms of two criteria: (1) the tracking quality as the ratio of the number of detected road feature points to the number of the sampled positions along the estimated road boundaries, using the image measurement model described in Sect. 3. (2) The number of times the tracker has to be re-initialised due to low tracking quality. A summary is shown in

Table 2 Average processing time

Task	Time (ms)
Initialisation	580
Tracking	40

Fig. 7 First row motorway. Second row suburban road. Third and forth rows rural road



Fig. 8 Estimated road curvature in three videos (unit: m^{-1}). x Axis indicates the frame number. Positive curvature indicates right bend. *Negative* indicates left bend

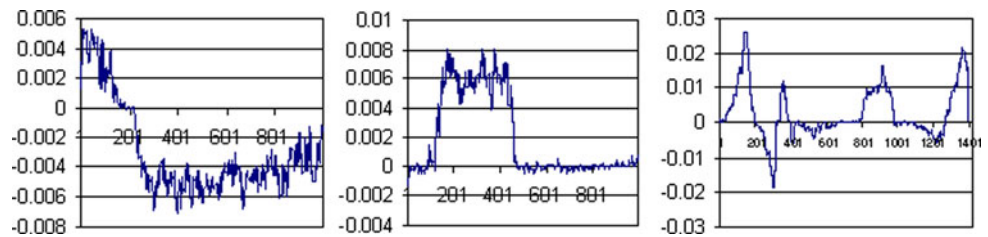


Fig. 9 Estimated lateral position of the vehicle relative to the current lane centre in three videos (unit: m). Two jumps in video 1 signal lane changing actions of the vehicle

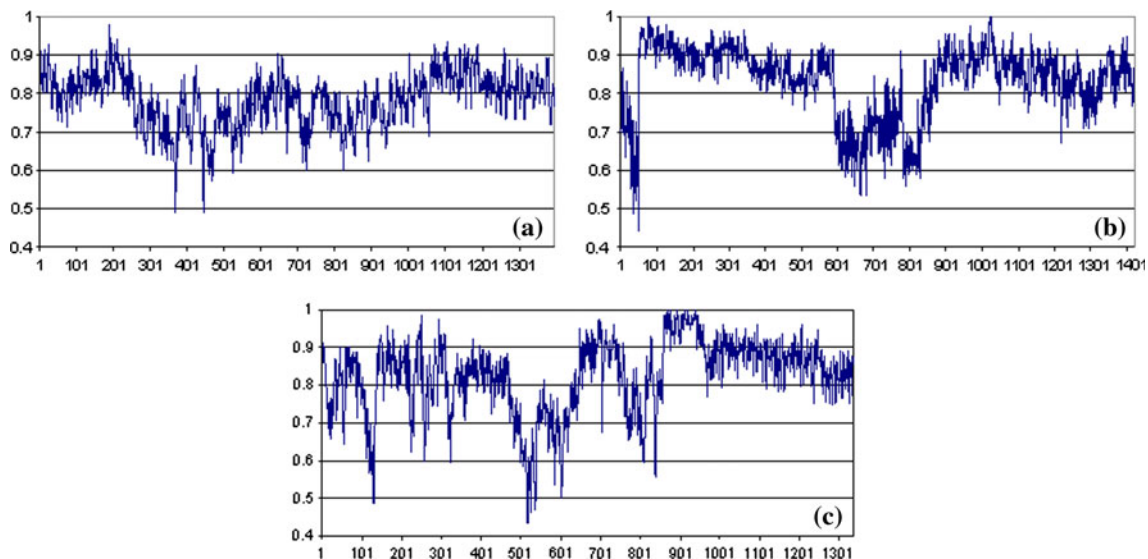
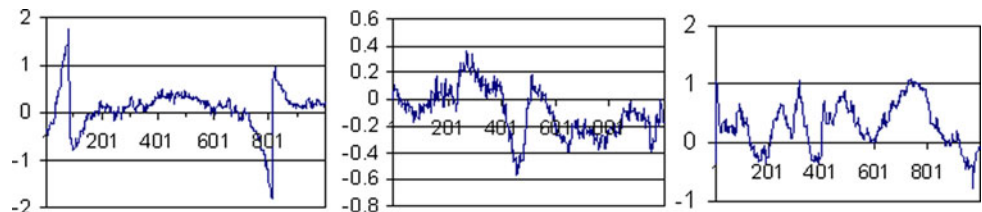


Fig. 10 Tracking quality **a** countryside; **b** motorway; **c** suburban

Table 3. We observed that the low quality of tracking is mainly due to low contrast of road boundaries against the background or significant occlusions by other vehicles or no presence of parallel road boundaries.

We also compared the tracking result using a single pair of hyperbolas with that using the road transition model, as shown in Fig. 11. Although the single pair of hyperbolas can generally follow the road boundaries close to the camera, it cannot follow those further away with a different curvature. Figure 12 gives a comparison between tracking with and without tracker switching. Although both algorithms use the same amount of particles (600 for our experiment, which is a major factor in computation time), multiple trackers gave results which are visually more stable and accurate than a single tracker.

Table 3 Tracking performance for three video sequences

Video	Number of frames	Re-initialization
Countryside	1,398	2
Motorway	1,423	2
Suburban	1,348	10

6 Conclusion

Road detection and tracking in an outdoor environment is a non-trivial problem, due to varying curvatures of the roads and variations and noise in the image. We have developed a multiple Condensation tracker framework and integrated it with a hyperbola road model for road boundary detection

Fig. 11 Comparison of the tracking performance with and without road transition handling. *First row* tracking using a single pair of hyperbolas. *Second row* tracking using the transition model (two pairs of connected hyperbolas)

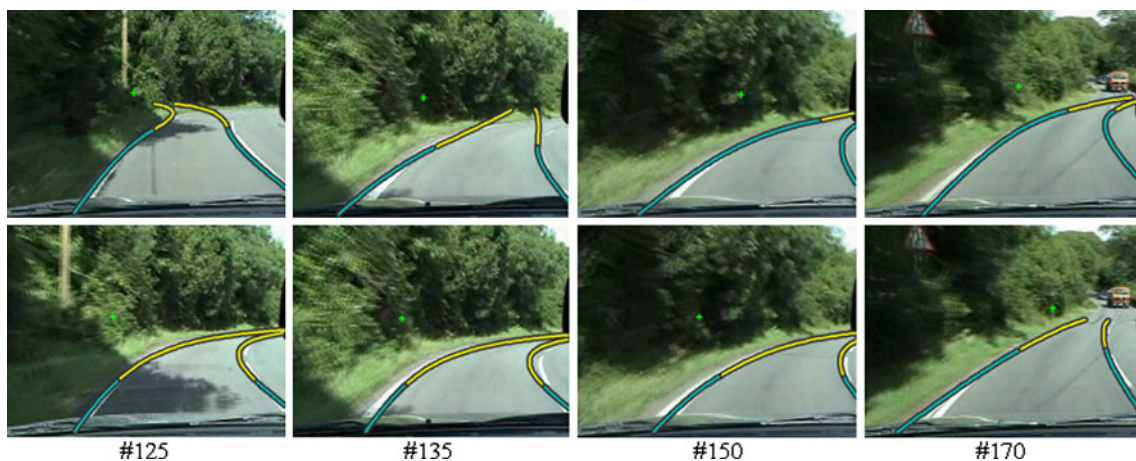
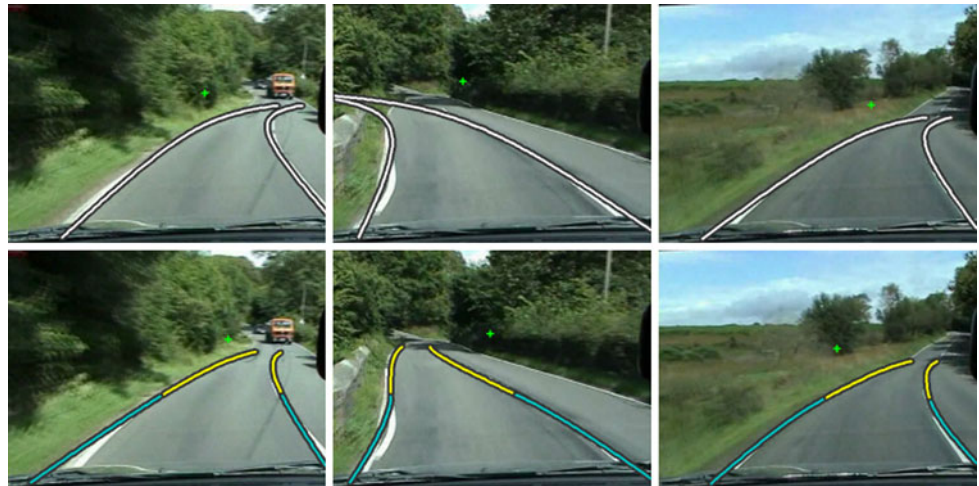


Fig. 12 Comparison of tracking performance with (*first row*) and without tracker switching (*second row*)

and tracking. Experimental results have shown that our method is robust for real world applications.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Badenas J, Bober M, Pla F (2001) Segmenting traffic scenes from grey level and motion information. *Pattern Anal Appl* 4(1)
2. Messelodi S, Modena MC, Zanin M (2005) A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Anal Appl* 8(1-2)
3. Bai, L., Wang, Y. and Tompkinson, W. (2004) Computer vision techniques for traffic flow computation, *Pattern Analysis and Applications*, vol 7(4)
4. Isard M. and Blake A. (1998) CONDENSATION: conditional density propagation for visual tracking. *Int J Comput Vis*
5. Simond N. (2006) Reconstruction of the road plane with an embedded stereo-rig in urban environment. In: *IEEE intelligent vehicle symposium*
6. McCall JC, Trivedi MM (2004) An integrated, robust approach to lane marking detection and lane tracking. In : *Proceedings of IEEE intelligent vehicles symposium*
7. Wilson MB, Dickson S (1999) Poppet: a robust road boundary detection and tracking algorithm. In: *British machine vision conference*
8. Smuda P, Schweiger R, Neumann H, Ritter W (2006) Multiple cue data fusion with particle filters for road course detection in vision systems. In: *IEEE intelligent vehicles symposium*
9. Fascioli A. (1997) GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection, <http://vislab.ce.unipr.it/GOLD/>
10. Crisman JD, Thorpe CE (1991) UNSCARF: a color vision system for the detection of unstructured roads. In: *Proceedings of IEEE international conference on robotics and automation*, Sacramento, California
11. He Y, Wang H, Zhang B (2004) Color-based road detection in urban traffic scenes. In: *IEEE transactions on intelligent transportation systems*, vol 5, No 4
12. Rasmussen C (2004) Texture-based vanishing point voting for road shape estimation. In: *British machine vision conference*
13. Tsai H, Rasmussen C, Chang T, Shneier M (2002) Road detection and tracking for autonomous mobile robots. In: *Proceedings of the SPIE 16th annual international symposium on aerospace/defense sensing, simulation, and controls*, Orlando, Florida, USA

14. Ke Q, Kanade T (2003) Transforming camera geometry to a virtual downward-looking camera: robust ego-motion estimation and ground-layer detection. In: IEEE international conference on computer vision and pattern recognition
15. Lützel M, Dickmanns ED (2000) EMS vision: recognition of intersections on unmarked road networks. In: Proceedings of the IEEE intelligent vehicles symposium, IEEE Computer Society, Washington, D.C., pp 302–307.
16. Rasmussen C. (2003) Road shape classification for detecting and negotiating intersections. In: IEEE intelligent vehicles symposium, IV-03, Columbus, OH
17. Chapuis R, Aufrere R, Chausse F (2002) Accurate road following and reconstruction by computer vision. In: IEEE transactions on intelligent transportation systems, vol 3, No. 4
18. Apostoloff N, Zelinsky A (2003) Robust vision based lane tracking using multiple cues and particle filtering. In: Proceedings of IEEE intelligent vehicles symposium, pp 558–563
19. Wang Y, Teoh EK, Shen D (2004) Lane detection and tracking using B-Snake. *Image Vis Comput* 22:269–280
20. Yagi Y, Brady M, Kawasaki Y, Yachida M (2000) Active contour road model for smart vehicle. In: Proceedings of international conference on pattern recognition, vol 3, pp 819–822
21. Southall B, Taylor CJ (2001) Stochastic road shape estimation, *ICCV*, vol 1, p 205
22. Park JW, Lee JW, Jhang KY (2003) A lane-curve detection based on an LCF. *Pattern Recogn Lett* 24:2301–2313
23. Eidehall A, Gustafsson F (2006) Obtaining reference road geometry parameters from recorded sensor data. In: IEEE intelligent vehicles symposium
24. McCall JC, Trivedi MM (2006) Video based lane estimation and tracking for driver assistance: survey, system and evaluation. *IEEE Trans Intel Transp Syst* 7(1):20–37
25. Guiducci A. (2000) Camera calibration for road application, In: *Computer vision and image understanding*, vol 79
26. Chen Q, Wang H (2006) A real-time lane detection algorithm based on a hyperbola-pair model. In: IEEE intelligent vehicles symposium
27. Wang Y, Bai L, Fairhurst M (2008) Robust road detection and tracking using condensation. *IEEE Trans Intel Transp Syst* 9(4):570–579
28. Weigel H, Cramer H, Wanielik G, Polychronopoulos A, Saroldi A (2006) Accurate road geometry estimation for a safe speed application. In: IEEE intelligent vehicles symposium
29. Cramer H, Scheunert U, Wanielik G (2004) A new approach for tracking lanes by fusing image measurements with map data. In: IEEE intelligent vehicles symposium
30. Li Bai, Wang Y (2007), Extended hyperbola road model for real-time road tracking. In: The 27th SGAI international conference on artificial intelligence
31. Cipolla R, Drummond T, Robertson D (1999) Camera calibration from vanishing points in images of architectural scenes, *BMVC*
32. Robertson D, Cipolla R (2004) An image-based system for urban navigation, *BMVC*
33. Cantoni V, Lombardi L, Porta M, Sicard N (2001) Vanishing point detection: representation analysis and new approaches. In: The 11th international conference on image analysis and processing
34. Least Median of Squares <http://www-sop.inria.fr/robotvis/personnel/zhang/Publis/Tutorial-Estim/node25.html>
35. M-Estimators <http://www-sop.inria.fr/robotvis/personnel/zhang/Publis/Tutorial-Estim/node24.html>
36. Tarel JP, Guichard F (2000) Combined dynamic tracking and recognition of curves with application to road detection. In:

- Proceedings IEEE international conference on image processing, vol I, pp 216–219
37. Guichard F, Tarel JP (1999) Curve finder combining perceptual grouping and a Kalman like fitting, *ICCV*

Author Biographies



Li Bai was a mathematician before she came to the Computer Science Department at the University of Nottingham, UK, in 1989. She completed a PhD in Computer Science at the University of Nottingham. Her research interests in pattern recognition and computer vision include face recognition, visual interfaces, video based navigation, and medical image analysis.



Yan Wang was born in China. He received a BSc from Northern Jiao Tong University in Beijing, China in 2001. He obtained a MSc degree in Information Technology from the University of Nottingham, UK in 2003. He has been doing a PhD degree in Computer Science in the University of Nottingham, UK, since 2004. His major research interests include pattern recognition, image processing and computer vision.



Michael Fairhurst is currently Head of the Department of Electronics at the University of Kent in the UK. His long term research interests focus on computational architectures and algorithms for image analysis and classification, with applications including handwritten text reading and document processing, medical image analysis and, especially, security and biometrics. He is a member of the Biometrics Working Group and the UK Government's Biometrics Assurance Group. He sits on numerous Conference and Workshop committees, and on the Editorial Boards of several international Journals. He has published more than 350 papers in the scientific literature, and is an elected Fellow of the International Association for Pattern Recognition (IAPR), an award made in recognition of his contribution to the field.