MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Estimation of lane state from car-mounted camera using multiple-model particle filter based on voting result for one-dimensional parameter space

Eisuke Adachi and Hiroaki Inayoshi and Takio Kurita
National Institute of Advanced Industrial Science and Technology
Umezono 1–1–1 Tsukuba Central 2, Tsukuba-shi, Ibaraki-ken, 305–8568 Japan
{e-adachi,h.inayoshi,takio-kurita}@aist.go.jp

Abstract

This papaer proposes a method of estimating the state of a driving lane by using a multiple-model particle filter for switching the type of multiple-lane model. This model represents whether there is a lane to the left or right of the running lane or there are lanes on both sides of the running lane. Using the context that an image is captured by car-mounted camera, we simplified the lane detection problem to reduce the lane state estimates. Assuming that the lane boundaries are straight lines, we can represent the lane model as a one-dimensional parameter space.

1 Introduction

Traffic informationization is hoped to enable automatic driving and safe driving supports for cars. Some studies have developed means of recognizing the situation outside the car from dynamic scenes captured with a car-mounted camera. Lane detection, the process of locating lanes in an image captured with a camera on a vehicle is a basic process for recognizing traffic conditions or the state of a vehicle relative to the lanes. Many lane detection methods have been proposed [5, 1, 6, 4, 3, 8, 2]. There are two basic approaches to detect lanes: region based and edge based. In the edge based method, lane detection is done by fitting lines to edges or lane markers calculated from an image. Li et al. [5] used a quadratic function to detect curved lanes. If the model is complex, the computation time is involved and the lanes may not be detected consistently. In many cases, the shape of lane boundaries is a straight line in the near field of the front or rear of the vehicle. Park et al. [6] proposed a lane curve detection method such that the first near field lane has to be detected as a straight line.

Many lane detection methods simplify the lane detection problem by using the context that an image is captured by a car-mounted camera. Concretely for example, one method uses the restrictions that road is a plane and the projection from the road plane to the image plane does not change [5]. Moreover, Apostoloff [1] applied a particle filter to a lane tracking framework integrating multiple cues, colors, edges etc. By using the tracking framework, the search region of the lane parameters is restricted to a region predicted by the history of the past estimates.

For further simplicity, in assumptions stated above, we assume that the lane direction on a road plane is parallel to the car's running direction. Furthermore, we handle multiple lanes with a multiple lane model, unlike the conventional methods that handle only the running lane or detect each lane boundary individually. For tracking, we adopted a multiple-model particle filter. To describe multiple lanes, the most plausible model among a set of models is selected in each frame by estimating the model parameters, lane position, and width. This method assumes that the shape of the road is a plane and the lane boundary is almost a straight line in an image sequence. The multiple lane model is thus able to be constructed with a one-dimensional probability density function.

2 Multiple-model particle filter

Let us consider the problem to estimate a system state vector $\mathbf{y}_k$ from measurements $\mathbf{Z}_k = \{\mathbf{z}_i, i = 1, ..., k\}$ up to time k. If the system state vector is described as $\mathbf{y}_k = [\mathbf{x}_k^T, r_k]^T$, where $\mathbf{x}_k$ are continuous-valued states and r_k is a discrete-valued state, this problem can be solved recursively using the following equations.

Prediction:

$$p(\mathbf{x}_k, r_k = j|\mathbf{Z}_{k-1}) = \sum_i \pi_{ij} \times \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k = j) p(\mathbf{x}_{k-1}, r_{k-1} = i|\mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} \quad (1)$$

Update:

$$p(\mathbf{x}_k, r_k = j|\mathbf{Z}_k) = \eta_k \times p(\mathbf{z}_k|\mathbf{x}_k, r_k = j) p(\mathbf{x}_k, r_k = j|\mathbf{Z}_{k-1}) \quad (2)$$

In (1), the $p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k)$ and $\pi_{k-1,k}$ express the motion model and state transition probabilities, respectively. Equation (2) is a Bayesian rule which calculates a posterior probability density function (pdf) $p(\mathbf{x}_k, r_k|\mathbf{Z}_k)$ from a likelihood function $p(\mathbf{z}_k|\mathbf{x}_k, r_k)$ and a prior pdf $p(\mathbf{x}_k, r_k|\mathbf{Z}_{k-1})$. The multiple-model (MM) particle filter is a sequential Monte Carlo approximation of the conceptual solution given by (1) and (2) [7].

3 Proposed method

3.1 Multiple lane model

The lane model is constructed by straight lines. They are equivalent to lane boundaries (for example, white lines) shown in Fig. 1. The set of lane boundaries is expressed as

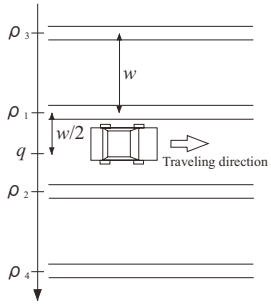

Figure 1. Coordinate system of road plane.

$$[\rho_1, \rho_2, \rho_3, \rho_4]^T = [1,1,1,1]^T q + [-1,1,-3,3]^T w/2, \quad (3)$$

where the ρ axis is perpendicular to the vehicle traveling direction and is on the road plane. The variables q and w respectively indicate the lane's position and width. We prepared four types of lane model according to whether there is a lane beside the running lane as follows.

1) There is only the running lane.

2) There is a lane at the left side of the running lane.

3) There is a lane at the right side of the running lane.

4) There are lanes at the left and right side of the running lane.

This lane models are described in Fig. 2.

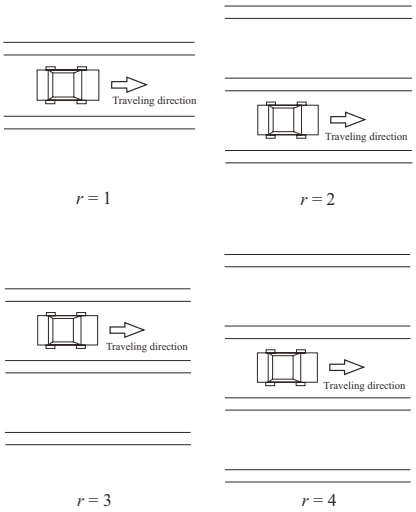

Figure 2. Model of two or more lanes.

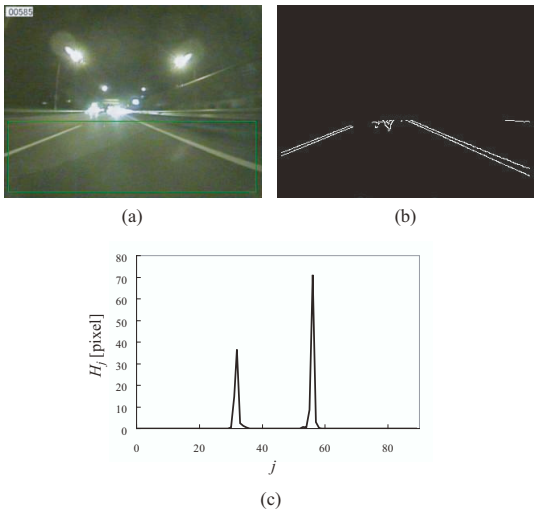

Figure 3. (a) Image captured with car-mounted camera, (b) Edge image, (c) Histogram of edges.

3.2 Likelihood

Figure 3 (b) is the edge image extracted from the boxed region of the image shown in Fig. 3 (a). The edge histogram $H_j(j = 1,2,...,M)$ obtained by voting edge points from the edge image to the axis of the lane boundary position (ρ) is shown in Fig. 3 (c).

We prepared the lane model as a probability distribution of edges on the ρ axis. When n edge points $\mathbf{z} = [\rho_1, \rho_2, ..., \rho_n]^T$ are observed in an image where the edge position is measured on the ρ axis, it is considered that the edge points $\mathbf{z}$ were independently generated at each edge point with a probability distribution function as follows.

$$p(\rho_j|r,q,w) = \alpha \left(\sum_{l=1}^{4} c_{rl} \bar{H}_j(\rho_l(q,w)) + \beta N_j \right) \quad (4)$$

In (4), α is a coefficient to normalize $p(\rho_j|r,q,w) = 1$. $\bar{H}_j(\rho)$ is a histogram where one lane boundary is at ρ in an image. N_j is also the histogram in which all of pixels in the target region in the box in Fig. 3 (a) are voted. N_j generates noise edges that are not for the lane boundary. $\mathbf{c}_r = [c_{r1}, c_{r2}, c_{r3}, c_{r4}]^T$ are coefficients that represent whether there are lane boundaries in each model, as follows.

$$\begin{cases} \mathbf{c}_1 = [1,1,0,0]^T \\ \mathbf{c}_2 = [1,1,1,0]^T \\ \mathbf{c}_3 = [1,1,0,1]^T \\ \mathbf{c}_4 = [1,1,1,1]^T \end{cases} \quad (5)$$

The likelihood of measurements $\mathbf{z}$ is calculated with $p(\rho_j|r,q,w)$ and H_j as follows.

$$p(\mathbf{z}|r,q,w) = \prod_{j=1}^{M} p(\rho_j|r,q,w)^{H_j} \quad (6)$$

3.3 Motion model

In (1), the following equations are used as the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k)$.

$$w_k = w_{k-1} + \delta_w \quad (7)$$

$$q'_k = q_{k-1} + \delta_q \quad (8)$$

$$q_k = \begin{cases} q'_k + w_k & (q'_k < -w_k/2) \\ q'_k - w_k & (q'_k \geq w_k/2) \\ q'_k & \textit{elsewhere} \end{cases} \quad (9)$$

In these equations, δ_w and δ_q are randomly generated with a zero-mean Gaussian distribution function with respective variances σ_w^2 and σ_q^2. The space of q is closed such that $-\frac{w}{2} \leq q < \frac{w}{2}$. The type of model r is switched according to the probabilities $\pi_{k-1,k}$ shown in Table 1.

Table 1. Transition probabilities of lane model.

	$r_k = 1$	$r_k = 2$	$r_k = 3$	$r_k = 4$
$r_{k-1} = 1$	6/8	1/8	1/8	0
$r_{k-1} = 2$	1/12	9/12	1/12	1/12
$r_{k-1} = 3$	1/12	1/12	9/12	1/12
$r_{k-1} = 4$	0	1/8	1/8	6/8

4 Experiment

4.1 Test sequence

The test image sequences were taken with cameras installed on the roof of a vehicle. Five cameras were installed. Four were synchronized for forward and backward stereo. They each had 360×240 [pixel]. The remaining camera was forward monocular, but it produced high-resolution images (720×480[pixel]). The vehicle run on both ordinary and express roads for about thirty hours in total, capturing image sequences at 30 [frame/sec]. The captured scenes included scenes taken during daytime and nighttime.

For this experiment, we used the scenes captured by the right camera of the forward stereo pair. The scenes were taken in daytime on an express road.

4.2 Experimental results

4.2.1 Examples of lane estimation

Figure 4 is an example of lane estimation. Figure 4 (a) shows the image of the left edge of the white line, and (b) shows the voting result for edge points on the lane position axis. Figure 4 (c) shows the original image on which the estimated lane is drawn. In this figure, red thin lines show peaks of the edge histogram H_j, green thick lines are the boundaries of the running lane, and yellow thick lines are lane boundaries beside the running lane.

Figure 5 shows other lane estimation results. Images (a) and (b) in this figure are examples of frames in which the estimate of the multiple-lane model was successful. Image (b) is a frame showing a changing lane sequence. Images (c) and (d) are examples of failure frames where the type of lane model could not be correctly estimated. In image (c),

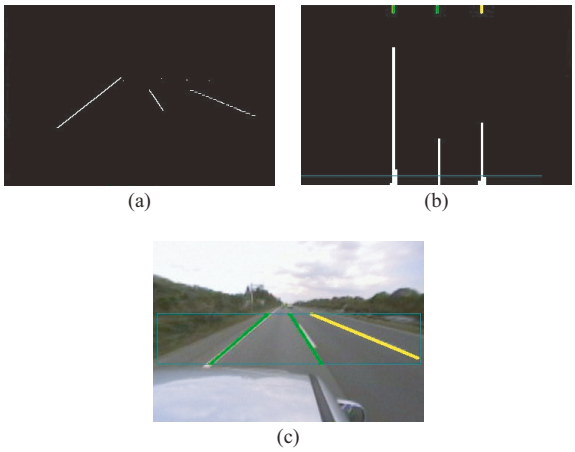

Figure 4. Example of lane estimation. (a) Image of left edge of the white line, (b) Voting result of edge points on the lane position axis, (c) Original image in which the estimated lane is drawn.

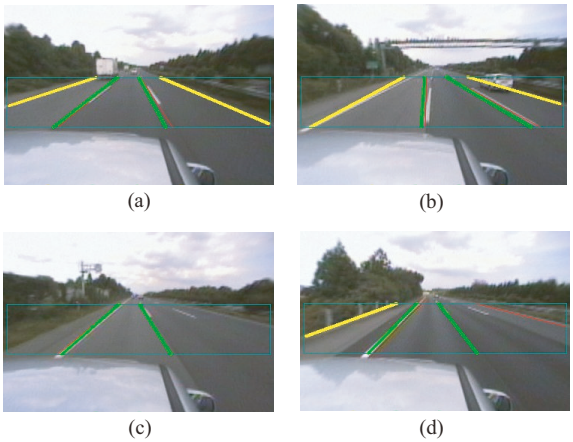

Figure 5. Examples of lane estimation.

this failure is supposed to have occurred because the white line is dotted and the length is insufficient to recognize a lane boundary. In image (d), a pair of edges is mistaken to be a single white line because there is a similar structure to white line on the road. In situations like (d), the estimation may require other cues besides the edge cue.

Figure 6 shows lane estimation results for complicated scenes. Figure 6 (a) shows a curved scene. In this image lane position is approximately estimated, although the lane's curvature can not be estimated. The other examples are results of complicated scenes with vehicles or traffic signs painted on the road. The lanes were well estimated even under these complex conditions.

4.2.2 Accuracy of lane position

Figure 7 shows the results of the estimated lane position q (solid line) compared with the true position (diamond mark). The true position was manually detected in the image and projected to the ρ axis. The four graphs in this figure are the results for lane change scenes. The estimated results are reasonable.

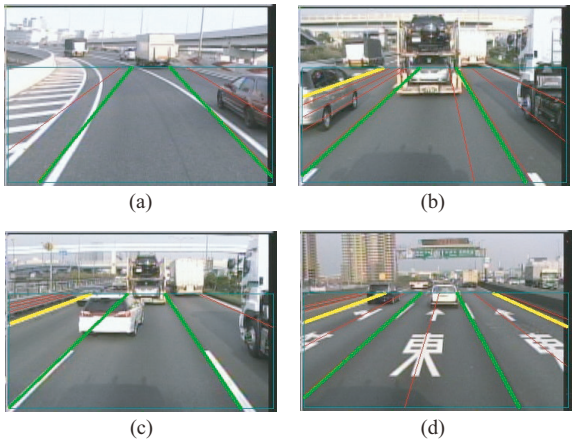

Figure 6. Examples of lane estimation in complicated scenes.

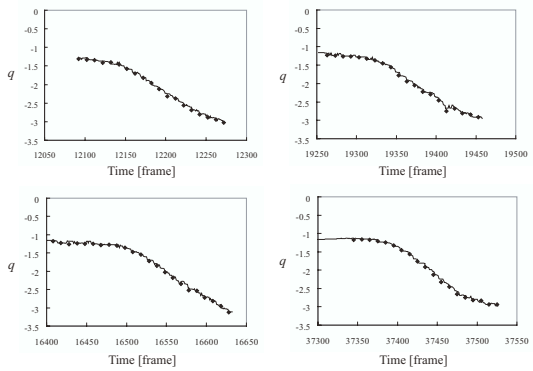

Figure 7. Results of estimated lane position q (solid line) compared with the true position (diamond mark) detected manually and projected to the ρ axis.

4.2.3 Feasibility of particle filter

We tested the feasibility of using the particle filter. Figure 8 shows the estimated lane positions q for two cases: one case using the maximum likelihood method and grid search, and the other using the particle filter. The particle filtering gave stable estimates, but the non-particle filtering estimates were instable. Accordingly, we confirmed that the particle filter works well for lane tracking.

5 Conclusion

This paper proposed a method of estimating the state of a driving lane by using a multiple-model particle filter for switching the type of multiple-lane model. This model represents whether there is a lane to the left or right of the running lane or there are lanes on both sides of the running lane. Using the context that an image is captured by car-mounted camera, we simplified the lane detection problem to reduce the estimates of the lane state. Assuming the lane boundary is a straight line, the lane model can be represented as a one-dimensional parameter space. We used the left edge of the white line as a lane boundary. In the particle filter, the likelihood of the lane's model is calculated from the distribution of edges voted in the one-dimensional space.

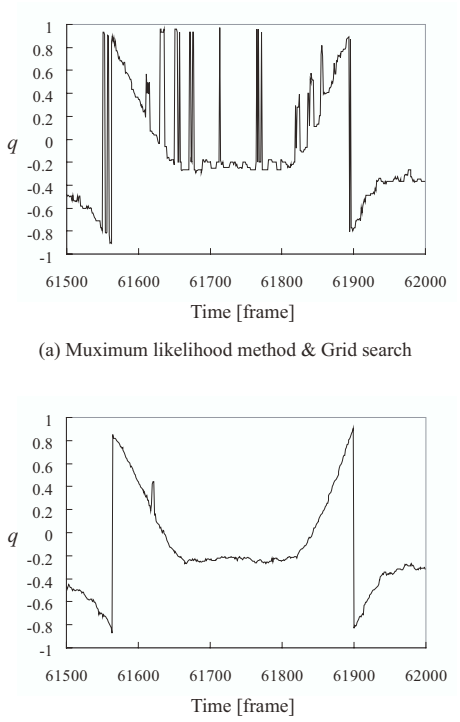

Figure 8. Results of estimated lane position.

We tested the method in an experiment using image sequences captured by a camera mounted on a car running on an express road in daytime. The experimental results showed that the method worked well in most situations but also revealed the limitations of using only edge cues. Where the white line is dotted line, the estimate of the type of lane model failed. Our future work will thus involve devising a way to handle dotted white lines.

References

[1] N. Apostoloff and A. Zelinsky. Vision in and out of vehicles: Integrated driver and road scene monitoring. *The International Journal of Robotics Research*, 23(4-5):513–538, April-May 2004.

[2] J. B. M. Donald. Application of the hough transform to lane detection and following on high speed roads. In *Proc. of the Irish Signals and Systems Conference*, 2001.

[3] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21:359–381, 2003.

[4] J. W. Lee, U. K. Yi, and K. R. Bark. A cumulative distribution function of edge direction for road-lane detection. *IEICE Trans. Inf. & syst.*, E84-D(9):1206–1216, September 2001.

[5] Q. Li, N. Zhang, and H. Cheng. Springrobot: A prototype autonomous vehicle and its algorithms for lane detection. *IEEE Trans. Transportation systems*, 5(4):300–308, December 2004.

[6] J. W. Park, J. W. Lee, and K. Y. Jhang. A lane-curve detection based on an lcf. *Pattern Recognition Letters*, 24:2301–2313, 2003.

[7] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter; Particle filters for tracking applications*. Artech house, 2004.

[8] T. M. van Veen and F. C. A. Groen. Discretization errors in the hough transform. *Pattern Recognition*, 14(1-6):137–145, 1981.