Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Lane Boundary Detection and Tracking using NNF and HMM Approaches

M. Boumediene, A. Ouamri, and N. Dahnoun

Abstract—**In this communication we present a new algorithm of lane detection and tracking. In the detection step, from the first frame of a video sequence, a linear-parabolic model is used to smooth the estimated trajectories, obtained by using the NNF approach. In the step of tracking, assuming a small change in the model, we use the HMM to update each parameter of the model. The results obtained are satisfactory.**

I. INTRODUCTION

Lane detection and tracking is the problem of locating road lane boundaries without an a priori knowledge of the road geometry. Lane tracking techniques and a vision-based lane boundary location system can assist in a number of "driver assistant'' applications, such as intelligent vehicles, and automatic navigation systems. Basically, there are two classes of approaches used in lane detection: the feature-based technique and the model-based technique [1, 2, 3, 4, and 5]. In the feature based technique the lanes in the road images are detected by traditional image segmentation where the low-level features are combined [6, 7, and 8]. Accordingly, in this technique the studied road is assumed having well-painted lines or strong lane edges, otherwise it will fail. Moreover, this technique may suffer from occlusion or noise.

On the other hand, in the model-based technique, one uses only a few parameters to represent the lanes . This technique is based on the assumption that the shapes lane can be represented by straight lines or parabolic curves [3-5].

Hence, the lane detection problem is considered as a problem of the model' parameters estimation.

Several algorithms were developed to estimate the parameters of a lane model to achieve the lane detection. These algorithms are based on various techniques such as: edge detection [6, 7], Hough transform [8] and networks of neurons [9], etc.

In this paper we propose a new algorithm for lane detection / tracking. The problem is reduced to the estimation of a target trajectory. Basically our method consists in two steps:

1) Lane detection: this first step consists of detecting lane boundaries from the first frame of the video sequence, using a tracking algorithm.
2) Lane Tracking: this step consists of updating the detection in the previous frame to the subsequent one, by using the Hidden Markov Model (HMM).

In section 2 we present the new algorithm, based on a tracking algorithm inspired from radar tracking, to achieve the lane detection. In section 3, we present the lane tracking step by using the HMM. This paper concludes in section 4.

Manuscript received January 15, 2007.
A. Ouamri is with the Signal and Image Laboratory, USTO-MB BP :1505 Oran el'Mnouar 31000, Algeria (corresponding author: phone: 00 213 41 56 03 57 00; fax: 00 213 41 56 03 57 00; e-mail: ouamri@univ-usto.dz)
M. Boumediene is with the Signal and Image Laboratory, USTO-MB BP :1505 Oran el'Mnouar 31000, Algeria
(e-mail:mjboumediene@yahoo.fr_)
N. Dahnoun, is with Bristol University, Electrical and Electronic Engineering Merchant Venturers Building
Woodland road Bristol BS81UB, UK
(e-mail: Naim.Dahnoun@Bristol.ac.uk)

II. LANE DETECTION

A. Formulation of lane detection problem

In this section we propose a new formulation of the lane detection. The problem of lane detection is formulated as follows: we consider each lane boundary as being the trajectory of a moving object or target, which it will be necessary to estimate using a tracking algorithm based on the Kalman Filter (KF) [10] [11]. In our case, the fictive target will be a pixel or a set of pixels which will move throughout the lane boundary (Fig.1).

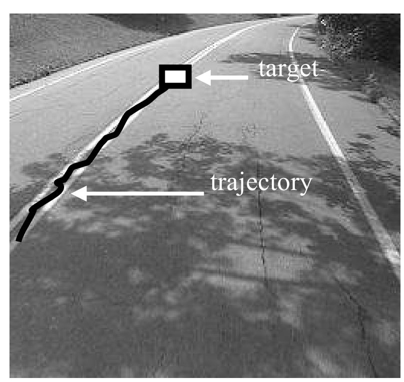


Fig. 1. Principle of the lane detection

In our case, the right and the left lanes to be detected are well separated, that is means that each lane can be considered as separate from the other. Hence, the tracking problem of the targets corresponding to each lane is reduced to two single targets tracking in clutter.

By consider the problem of tracking a single target in clutter, the target dynamics are modelled in discrete time in the standard manner [10,11], using a state-space model of the form:

$$\mathbf{x}(k+1)=\mathbf{F}(k)\mathbf{x}(k)+\mathbf{G}(k)\mathbf{u}(k)+\mathbf{v}(k) \quad (1)$$

Where $\mathbf{x}(k)$ is the target state vector that includes the quantities to estimate, such as the position of the pixel and its speed of displacement in the image, u(k) is an input, $F(k)$ is the matrix transition, and v(k) is the noise vector k = 0, 1, . . . , assumed to be a zero-mean, white Gaussian process with covariance $\mathbf{E}\left[\mathbf{v}(k)\mathbf{v}^T(l)\right]=\mathbf{Q}(k)\delta(k,l)$, where $E\{[*]\}$ denotes the mathematical expectation.

The measurement of the target kinematics equation is

$$\mathbf{z}(k)=\mathbf{H}(k)\mathbf{x}(\mathbf{k})+\mathbf{w}(k) \quad (2)$$

where w(k) is the measurement noise, assumed to be a zero-mean, white Gaussian process with covariance $\mathbf{E}\left[\mathbf{w}(k)\mathbf{w}^T(l)\right]=\mathbf{R}(k)\delta(k,l)$, and $\mathbf{H}(k)$ is the matrix measurement. This matrix relates the state to the measurements. In practice $\mathbf{H}(k)$ might change at each time step or measurement, but here we assume it is constant. The process noise and measurement noise sequences are assumed to be uncorrelated with known covariances, $\mathbf{E}\left[\mathbf{w}(k)\mathbf{v}^T(l)\right]=0; \quad \forall\ k,l$. The matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ are positive definite matrices.

The matrices F, G, H, Q, and R are assumed to be known with appropriate dimensions.

The proposed algorithm is inspired from the Radar tracking algorithms [10, 11]. There are four processing stages in the algorithm:

--The measurements' generation by edge pixel extraction.

--Trajectory or track initiation, using the Hough transform.

--Tracking algorithm, based on the Nearest Neighbour Filter NNF approach to estimate the trajectory.

--Lane modelling: The estimated trajectory in the previous step is modelled by a linear-parabolic model.

B. *Generation of the measurements*

As the pixels representing the white marking of the road have a high intensity, the measurement generation may be achieved by the edge pixel extraction which is performed by Canny edge detection. The Canny filter is employed to obtain edge map (Fig. 2).

The detected edge points are used to estimate the trajectories of the moving targets by using the Kalman filter. They represent the measurement of the target position. From Fig.2 we see that the detected edge points can be classified in two groups: the first one represents the points originated from the targets or the lanes, and the second one represents the noise or the clutter.

It is well known that most of the target tracking algorithms need track initiation.

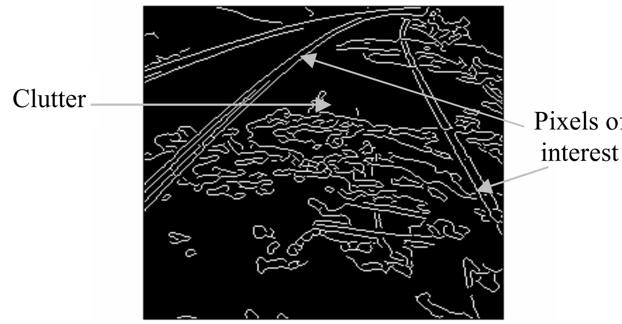


Fig. 2. Measure's generation: Edge Pixel Extraction

C. *Trajectory (Track) initiation*

The trajectory initiation is a problem similar to the tracks' initiation in the case of Radar tracking. Before using the Kalman filtering in the NNF, one must initialize the tracking algorithm from initial measurements. This represents an important step since a bad initialization leads to a misdetection of the lane. If additional information is used in the track initiation, efficient detection can be achieved.

In order to determine the first measurements which will allow thereafter the initialization of the Kalman filter, one could use the Hough transform to detect the two straight lines representing the two boundaries of the road.

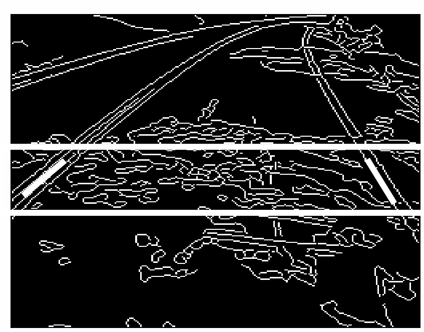


Fig. 3. Initialization of the Kalman Filter by the Hough Transform

The Hough transform is a well known method for initiating multiple target tracks. In our algorithm, this transform is applied only to one small section (about 20-30 lines) to of the filtered image where the likelihood of the presence of the road edges is the highest (Fig.3).

Since only a small section is used, the Hough transform computation load is not prohibitive. This load can be reduced by considering four to five sub-sections; each defined by 20-30 lines and 40-60 columns. This allows a Hough transform parallel implementation leading to a real time implementation.

Notice that some methods are based on the Hough transform by using the whole image to achieve the lane detection this very time consuming.

D. Tracking Algorithm

At each step time (iteration), several measurements of the target state are available. In a clutter environment, incorrect measurements may be received. Hence there is an uncertainly, related to the origin of measurements. This uncertainly poses the problem of which measurement, should be used to update the state of a given target.

The simplest method for target tracking in clutter is the Nearest Neighbour Filter (NNF), which utilizes the closest measurement to the predicted target measurement. The NNF assumes, at any time, that the Nearest Neighbour (NN) measurement is target-originated and uses it in a Standard Kalman Filter (SKF) to update the target state estimate. The performance of the NNF has been completely analyzed in [11].

The NNF (Nearest Neighbour Filter) is widely used for its computational simplicity.

The most commonly used validation gate in Radar tracking is an ellipsoid defined from the innovation and a gate size parameter. The validation gate is a region around the predicted measurement and is used to select the candidate measurements for association. A measurement falling inside the validation gate is referred to as a validated measurement, and is a candidate for the use in a tracking filter. In the NNF, the measurement closest to the predicted measurement is used for track update and the others are discarded.

In our case the gate is defined by a set of n_p pixels around the predicted position. Only those observations that are within the gate are considered for track updating.

Assuming the target moving with constant speed, hence the two-dimensional state vector is defined by position and speed. The tracking is performed with a Kalman filter of the second order (KF2) defined by:

$$F(k) = F_2 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad G(k) = G_2 = \begin{bmatrix} \frac{T^2}{2} & T \end{bmatrix}^T \text{ and}$$

$$Q(k) = Q_2 = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 \\ \frac{1}{2}T^3 & T^2 \end{bmatrix} \sigma_v^2 \qquad (3)$$

where $q = \sigma_v{}^2$ and T are, respectively, the measurement noise power and the sampling period.

The Kalman filter is initialized with the two points method by using the results provided by the Hough transform in the previous section.

The proposed algorithm has been tested on some sequences of images grabbed by an on-board camera at different locations and at different times. Fig. **4** shows some of our experimental results of lane boundary detection where estimated lane boundaries using the NN filter are superimposed onto the original images. The images presented in this communication are downloaded from http://vasc.ri.cmu.edu//idb/html/road/may30_90/index.html.

E. Lane model:

Lane model plays an important role in lane detection. We choose a linear-parabolic model as in [12] to represent lane boundary. This model is described by the function *f (x)*:

$$f(x) = \begin{cases} a + bx, & x > x_m \\ c + dx + ex^2, & x \le x_m \end{cases} \qquad (4)$$

Fig. 4. Estimated trajectories with NN filter.

where x_m represents the border between near and far fields. The function *f* verify the continuity and the differentiability conditions:

$$f\left(x_m^+\right) = f\left(x_m^-\right)$$
$$f'\left(x_m^+\right) = f'\left(x_m^-\right). \qquad (5)$$

The estimated trajectories obtained by the NNF in the previous step are modelled following the linear-parabolic model (4). Fig. **5** shows some results obtained by using linear-parabolic model, superimposed onto the original images.

The observations' sequence consists of M samples (1,…, M) of the model determined previously. Each sample defines a pixel coordinate, in the image, which will be associated to a normal line. The length of this normal line is $2N+1$ pixels indexed from $-N$ to N, position 0 on normal line $\emptyset$ corresponds to the pixel described by sample $\emptyset$. In this manner, the 2D representation of a contour will become a 1D representation where each contour on normal line $\emptyset$ will be identified by its position $\lambda \in [-N, N]$.

Fig. 5. The models superimposed on the original images.

In addition to the observations sequence, the HMM requires a hidden states sequence $S = \{S_{-N}, S_{-N+1}, \ldots, S_N\}$ describing all possible positions that a contour can take on a normal line. Thus the HMM is specified by:

--A set of state S, where each state has a probability $p(s_\phi) = 1/(2N+1), \phi \in [-N, N]$.

--A probability of transition between the states $p(s_\phi \mid s_{\phi-1}) = c \exp(-(s_\phi - s_{\phi-1})^2 / \sigma_s^2)$.

--An observation model $p(O_\phi \mid s_\phi = \lambda)$ describing the probability that a state $s_\phi = \lambda$ can generate the observation O_ϕ.

Before finding the optimal contour, we must generate the sequence of M observations. Each observation ($O_\emptyset$) on each normal line $\emptyset$ can be noised by the presence of many edges ($O_\emptyset = (c_1, c_2, \ldots, c_j)$). Of the J edges, at most one is the true contour. Hence we can define J+1 hypothesis [12]:

$$\begin{aligned} H_0 &= \{c_j = F : j = 1, \ldots, J\}, \\ H_j &= \{c_j = T, c_k = F : k = 1, \ldots, J; k \neq j\}. \end{aligned} \tag{6}$$

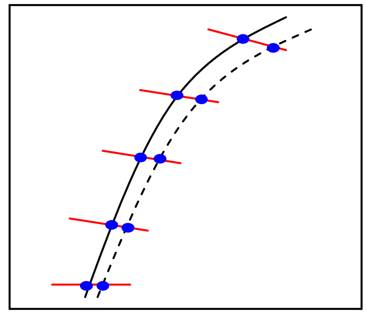


Fig. 6. The tracking with HMM, the continuous line represents contour at t-1 and the dashed line represents contour at t.

Where $c_j = T$ means the J-th edge is associated with the true contour, $c_j = F$ otherwise. Hypothesis H_0 therefore means that none of the edges is associated with the true contour.

The probability $p(O_\phi \mid s_\phi = \lambda)$, that a state $s_\phi = \lambda$ can generate the observation O_ϕ is approximated by:

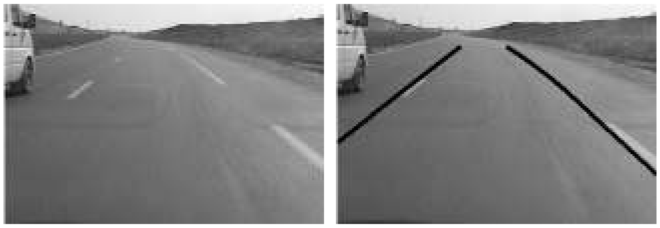
$$p(O_\phi \mid s_\phi = \lambda) \alpha 1 + (1/\sqrt{2\pi}\, \sigma_c q \gamma) \sum_{m=1}^{J} \exp(-(c_m - s_\phi)^2 / 2\sigma_c^2) \tag{7}$$

Where γ is the density of Poisson noise process along each normal line, σ_c is the standard deviation of true target measurement and q is the prior probability of hypothesis H_0. The results obtained in the tracking step are shown in fig. **7** and fig. 8. We see that the model is well superimposed on the real lane in the original frame.

III. CONCLUSION

In this communication we proposed new lane detection and tracking algorithm using a linear-parabolic model based on the NNF approach. During the initialization phase, we model each lane boundary as the trajectory of a moving target. In the lane tracking step, the parameters of the model are updated by the Hidden Markov Model (HMM). The computational complexity of the proposed method can be reduced by using others edge detection algorithms that Canny operator.

The complete system was analysed and simulated matlab software and the results seem to be very satisfactory. The next step is to implement this system on a real-time DSP processor such as the TMS320DM64x [13].



885

1068

1112

Fig. 7 **.** Frames of video sequence 1 and corresponding results.

REFERENCES

[1] Andrew, S. Lai, H. Nelson, C. Yung, Lane detection by orientation and length discrimination, IEEE Transactions On Systems, Man and Cybernetics, Part B 30 (4) (2000) 539–548.

[2] S.G. Jeong, C.S. Kim, K.S. Yoon, J.N. Lee, J.I. Bae, M.H. Lee, Realtime lane detection for autonomous navigation, IEEE Proceedings Intelligent Transportation Systems 2001 ,508–513.

[3] D. Grimmer, S. Lakshmanan, A deformable template approach to detecting straight edges in radar images, IEEE Transactions on Pattern

[4]] D. Jung Kang, J. Won Choi, I.S. Kweon, Finding and tracking road lanes using line-snakes, Proceedings of Conference on Intelligent Vehicle, 1996, pp. 189–194, Japan.

[5] Y. Wang, E. K. Teoh, et D. Shen, “Lane detection and tracking using B Snake,” Image and Vision Computing, vol.22, no.4, pp.269-280, Apr.2004

[6] A. Broggi, Robust real-time lane and road detection in critical shadow conditions, Proceedings IEEE International Symposium on Computer Vision, Coral Gables, Florida, Nov. 19–21 ,1995.

[7] S.G. Jeong, C.S. Kim, K.S. Yoon, J.N. Lee, J.I. Bae, M.H. Lee, Realtime lane detection for autonomous navigation, IEEE Proceedings Intelligent Transportation Systems 2001 ,508–513.

[8] Y. Xuan, B. Serge, B. Michel, Road tracking lane segmentation and obstacle recognition by mathematical morphology, Proceedings of the Intelligent Vehicles ’92 , 166–170.

[9] C. Rosito Jung, C. R. Kelber, A Robust Linear-Parabolic Model for Lane Following, Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI’04), 2004

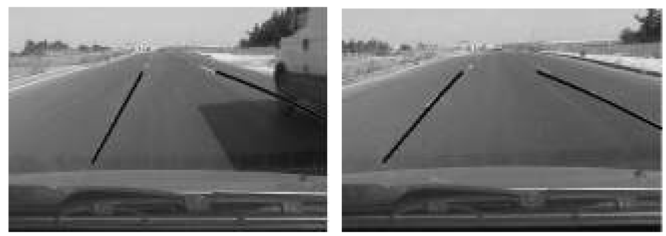
[10] M. Keche, A. Ouamri, I. Harrisson, M.S. Woolfson, " Performance Evaluation of an Algorithm for Multiple Target Angle Tracking", IEE Proceedings review, Radar, Sonar and Navigation; vol 144, N° 5, October 1997.

[11] X. R. Li and Y. Bar-Shalom, “Tracking in Clutter with Nearest Neighbor Filters: Analysis and Performance,” IEEE Trans. Aerospace and Electronic Systems, vol.AES-32, pp. 995—1010, July 1996.

[12] Y. Y Chen, Y Rui, and T. S.Huang, "JPDAF Based HMM for Real-TimeContourTracking “,IEEE 2001.

[13] N. Dahnoun: Digital Signal Processing implementation Using the TMS320C6000 Platform, Prentice Hall, 2000.

440 490



520 570

Fig. 8 **.** Frames of video sequence 2.