PHT와 최소자승법을 이용한 효율적인 실시간 점선차선 추적

(An Efficient Method for Real-Time Broken Lane Tracking Using PHT and Least-Square Method)

쉬 수 단 [†]　　이 창 우 [††]

(Sudan Xu)　　(Chang Woo Lee)

요 약 차선검출시스템은 지능형 차량 시스템의 중요한 요소이다. 차선검출 시, 주변 환경과 날씨의 변화 때문에 차선검출은 다양한 어려움에 직면하게 된다. 본 논문에서는 차선검출 및 추적을 위해 다양한 환경에서도 안정적으로 동작하는 간단하면서 효율적인 방법을 제안한다. 제안된 방법에서는 차선을 추적하고 차선의 기울기를 수정하기 위해 확률적 허프 변환(Probabilistic Hough Transform, PHT)과 최소자승법(Least-square method, LSM)를 이용한다. 일반적으로 차량의 내부에 설치된 카메라로부터 획득된 영상은 영상의 하단부분에서 차선이 비교적 뚜렷이 나타나고, 주변의 간섭을 적게 받는다는 가정 하에 제안된 방법에서는 차선검출 및 추적의 효율성을 증대시키기 위해 영상의 하단부분에 관심의 대상이 되는 두 개의 영역을 설정한다. 제안된 방법의 효율성을 입증하기 위해 정지영상과 비디오 영상을 사용하여 실험 하였고, 실험결과 제안된 방법이 강건하고, 신뢰성있는 결과를 얻었음을 보였다.

키워드 : 점선차선 검출, ROI, 최소자승법, PHT

Abstract A lane detection system is one of the major components of intelligent vehicle systems. Difficulties in lane detection mainly come from not only various weather conditions but also a variety of special environment. This paper describes a simple and stable method for the broken lane tracking in various environments. Probabilistic Hough Transform (PHT) and the Least-square method (LSM) are used to track and correct the lane orientation. For the efficiency of the proposed method, two regions of interest (ROIs) are placed in the lower part of each image, where lane marking areas usually appear with less intervention in our system view. By testing in both a set of static images and video sequences, the experiments showed that the proposed approach yielded robust and reliable results.

Key words : Broken lane detection and tracking, ROI, LSM, PHT

1. 서 론

Lane detection and tracking offer safety assurance to intelligent vehicles. The aim of lane detection and tracking is to give reliable information about lanes for vehicle navigation.

Various algorithms in lane detection have been researched for many years. Hough Transform was used in lane detection for an automatic robot in the 2004IGVC [1]. Based on the segmentation or classification of road and non-road areas, lane detection is usually executed. Support Vector Machine (SVM) and other trained classifiers are used for classification [2] and all the trained classifiers were compared in [3]. On the other hand, B-snake [4] aiming at detecting curved lanes drives splines under dynamics. Probabilistic methods to estimating spline parameters was tested with algorithms [5] and some other literatures using a hypothesis pair of lane model for lane modeling [6]. Recently, a real-time lane detection algorithm which can detect curve lanes using the Random Sample Consensus Algorithm (RANSAC) was proposed in [3]. LSM was once used in [7], which use the inter-frame information and prior knowledge to detect lanes; however, how to separate the background is not clearly mentioned.

Considering the high consumption of training methods, RANSAC or spline parameter estimation which are in the lack of adaptability to all existing methods and systems, our method concentrates on vehicles passing by in the broken-shape lanes. In this paper, we propose an efficient method combi-

· 이 논문은 제34회 추계학술대회에서 'PHT와 최소자승법을 이용한 효율적인 실시간 점선차선 추적'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 군산대학교 컴퓨터정보공학과
xusudan1201@163.com
†† 정 회 원 : 군산대학교 컴퓨터정보공학과 교수
leecw@kunsan.ac.kr
(Corresponding author임)

논문접수 : 2008년 1월 8일
심사완료 : 2008년 5월 28일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제6호(2008.8)

ning PHT [8] with LSM [9] to detect broken lanes which frequently appear on roads, as well as solid-typed lanes.

2. Overview of Our Method

A brief overview of the proposed method is depicted in Fig. 1: under the simple assumption that the intensity of the lane marking is brighter than the backgrounds [10], HIS color segmentation method is initially carried out to separate the lane marking area from other environments. In videos, the average threshold of the first few frames is used to improve the efficiency and stability of the binarization as described in section 2.1. Based on the binarization results, the PHT is then carried out to model the lane markings, which is illustrated in section 2.2. After two ROIs are set, a LSM is applied in the ROIs to correct the orientation of lanes and to fit exact lanes as described in section 2.3.

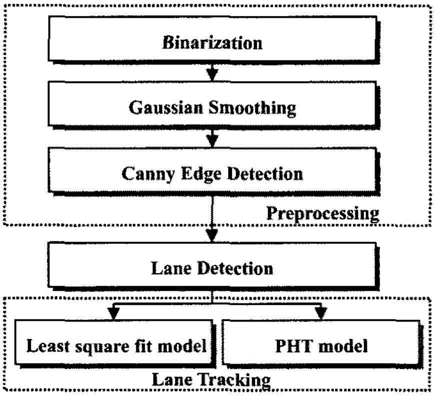


Fig. 1 Overview of the proposed method

2.1 Preprocessing

Our paper introduced a simple threshold searching method from [10]. Based on the empirical evidence that the intensities of two regions- (lanes or not) seldom change, the k-means clustering algorithm is applied to simplify the binarization with lower time consumption.

In video sequences, the set line used for the selection of threshold values will sometimes be on nothing (see Fig. 2). So we calculate the average threshold from the first several frames as the threshold in

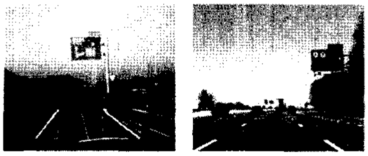


Fig. 2 Blue lines set on the 20th row from the bottom

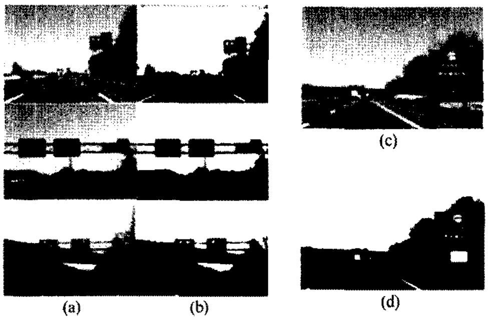


Fig. 3 Immediate results of the segmentation: (a) and (c) are inputs, (b) and (d) are the binary images after applying a threshold value

the latter frames. To extract lanes, we only consider the lower part of inputs so that the vehicles nearby can be removed as shown in Fig. 3. After the Gaussian smoothing, Canny edge detector is employed to extract edge pixels in the ROIs, in which the results are the input of the lane modeling.

2.2 Lane Modeling

By analyzing the binary images, it's easy to find out that the white pixels on the road are lanes while others are not; whereas, there are still other white pixels which are determined as non-lanes easily by human eyes but hard for computers.

With those observations, we find that the road area is usually in the lower part of image. ROIs are set at the bottom of each image and the lane markings only in the ROI are considered to be modeled as lanes. Therefore, the lane model is defined as a line model satisfying the following conditions:

1. Lanes are white pixels in the ROIs of the bipartite image.
2. Lanes have special slopes different from other objects.

To model the lanes, we employ the PHT only in the ROIs. The difference between the standard Hough Transform and the PHT is that the PHT is sufficient to compute the Hough Transform of only a proportion α ($0\% < \alpha < 100\%$) of the pixels in the image. These pixels are randomly chosen from a uniform probability density function defined over the image. This is equivalent to compute the Hough Transform of a sub-sampled version of the image.

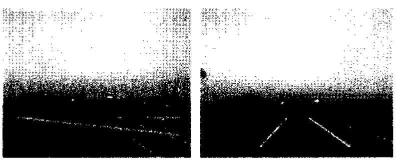
To model the defined lanes, three threshold values are determined to recognize the lanes. In the ROIs of the binary image, we only consider blobs that the number of pixels is more than 30 pixels. Each blob should be distant from each other more than 2 pixels. Using the PHT, we select lines as lanes that their accumulator counting is more than 30.

With the thresholds properly defined, lanes in the near distances of the driving vehicle are easily modeled. However, there are still some other incorrect results in videos which are caused by the white colored objects such as vehicles, the telegraph poles, or baluster along the roadside.

It's obvious that the lanes have a special slope coefficient different from the other lines. Therefore, to hold the direction of lanes, the slope coefficient is

```
for ( all points in ROI )
{
 if (distance between points < 2 pixels)
    perform PHT
  if ((found lines length > 30 pixels) &&
     (Accumulator > 30))
    Add to a set of Lane candidates
}
for ( all the lane candidates )
{
 if (slope coefficient of line is in the range
     between [-∞, -2.0/3] and [2.0/3, +∞])
    return Lane
}
```

Fig. 4 Pseudo codes for detecting lanes using the PHT



(a) (b)

Fig. 5 Result comparison with and without restriction

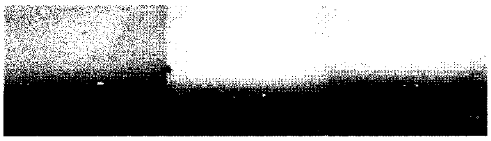
restricted in two intervals: $[-\infty, -2.0/3]$ and $[2.0/3, +\infty]$, and the PHT as showed in Fig. 4. It is easily found out that the slope constraint is useful to filter out the incorrect pseudo-lanes as shown in Fig. 5 in which both the frames are from the 19th frame of the same video sequence. The green lines in Fig. 5 are the detected lanes.

2.3 Lane Tracking

Solid-typed lanes are tracked in the ROI easily, while continuous tracking of broken lanes is difficult. By analyzing the characteristic of the broken lanes, we introduce three potential situations declared below and shown in Fig. 6.

1. Only small parts of defined lane exist in ROI (lost track);
2. No defined lanes exist in ROI (lost track);
3. Integrated fragments of defined lane exist in ROI (tracked).

The model of lane is directly instantiated by the PHT, and the model parameters are stored under the third situation as shown in Fig. 6(c), while a LSM is used to set up new model under the first situation as shown in Fig. 6(a). In the second situation shown in Fig. 6(b), the lane model of previous frames is used and stored for the second situation. The flow chart is showed in Fig. 7 below.



(a) (b) (c)

Fig. 6 Three potential situations: (a) (b) no lane markings in the near set line, (c) There is a lane

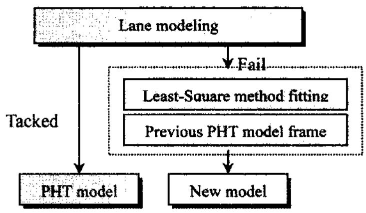


Fig. 7 Flow of the lane modeling method

LSM is used to find out the best model which fits the real model best. Here the slope coefficient and b in the former frame modeled by PHT are used as the estimated parameter. The estimated model can be written as Group Equation (1).

As in the group equation (1), there are (n-2) redundant points. From equation (1-1), we want to get a best line whose sum distances from all the redundant points to the line are the minimum (described in equation (1-2)). The sum distances are expressed by equation (1-3). In order to get the minimum of the sum distances, we take the local derivative of the sum distance function and set it to 0 as in equation (1-4). In group equation (1), Σd stands for the sum distances from redundant points to the line; n is used to present the number of points; (n-2) is the number of redundant points.

Group Equation (1)

$$y_n = ax_n + b \tag{1-1}$$

$$\sum_{i=0}^{n-2} d_i = \sum_{i=0}^{n-2} \frac{|ax_i + by_i + c|}{\sqrt{a^2+b^2}} = \sum_{i=0}^{n-2} \frac{|ax_i + b - y_i|}{\sqrt{1+b^2}} = \min \tag{1-2}$$

$$\sum d = \min \xrightarrow{transform} \sum (y_i - ax_n - b)^2 = \min \tag{1-3}$$

$$\xrightarrow{locaderevive} \frac{\partial \sum d}{\partial a} = \frac{\partial \sum d}{\partial b} = 0 \tag{1-4}$$

The equation group (2) for straight lines is simplified by deriving from group equation (1), and $\bar{x}$, $\bar{y}$ stand for the average values of all redundant points, lxx is the variance of x value and lxy is the covariance of x and y value, n stands for the number of points.

Group Equation (2)

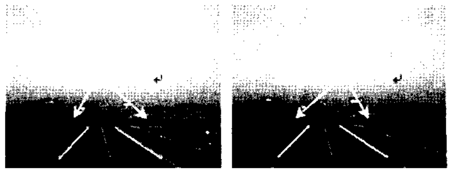
$$a = \bar{y} - b\bar{x} \qquad b = \frac{l_{xy}}{l_{xx}} \tag{2-1}$$

$$\bar{x} = \frac{\sum_{i=0}^{n-2} x_i}{n-2} \qquad \bar{y} = \frac{\sum_{i=0}^{n-2} y_i}{n-2} \tag{2-2}$$

$$l_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}) \quad l_{xx} = \sum (x_i - \bar{x})^2 \tag{2-3}$$

In the situations referred in 2.3, two areas are set according to the tracking results as shown in Fig. 8, and the LSM is only executed in the two ROIs. Finally the defined lane model is updated in real time. If no lane model is found in ROIs, the previous successive model is used, while if there are redundant white pixels appearing in ROIs, LSM is used to correct the orientation and fit a better model. Therefore, the lane model is updated time by time while the ROI is moving on. In the proposed method, lanes with little curves are approximately expressed by straight lines.

With the combination of PHT and LSM, we perform a computing loop that a fitting model and a checking model are performed in turn to model and track the broken lanes in video sequences.



(a) (b)

Fig. 8 Tracking results in two situations: (a) the lane model directly expressed by PHT tracking, (b) the lane model is computed by Least Square Fit

3. Experiment results

To evaluate our proposed approach, we collected 10 real-world test video sequences with 20 to 24 fps on 320×240 image resolution from a camera mounted inside vehicle. Also, using the Google engine, we downloaded 20 images with 640×480 image resolution, which are divided into 3 groups of pictures according to road condition, weather condition and complex backgrounds. The test platform is windows XP professional running on a desktop PC with Pentium IV 3 GHz CPU and 1 GB of memory.

As shown in Fig. 9, the broken lane markings under different situations are detected, and are not affected by the vehicles as shown in the first row that is a complex situation while the third row is foggy situation.

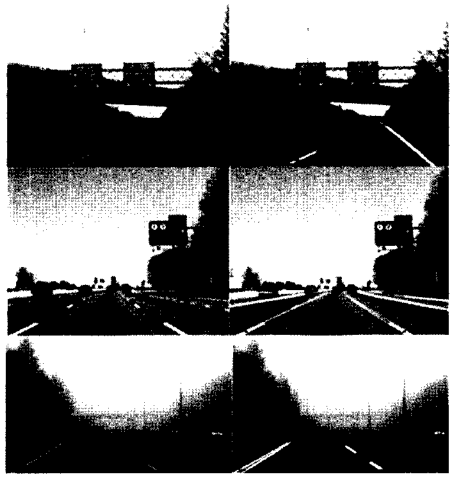
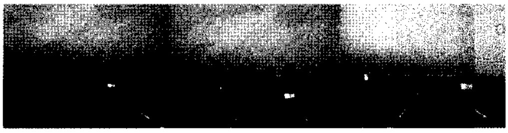


Fig. 9 Lane detection results in static images; left column images are original ones, and right column images represent the detected lane detection

(a)



(b)

Fig. 10 Results of lane tracking in video sequences: (a) straight lane (b) curved lane

As shown in Fig. 10, the tracking results are shown in both curved lane markings and straight lanes in videos, in which slightly curved lanes are modeled out by the tangent of the lane. In both experiments, only the inner parts of lanes are expressed by straight green lines.

The tracking accuracy as shown in Table 1 is the ratio of the correct tracking for the whole frames and static images. The reason why the accuracy of videos is higher than static pictures is that we determined more exact threshold value for the binarization during experiments on video data to avoid real-time tracking failures, for the background may sometimes change more significantly than the static ones. The analysis of the experimental results revealed that the incorrect tracking accuracy is due to the noise that is not totally eliminated in the preprocessing step. Comparing with [7], our method needs no structure knowledge of road; while comparing with [3], ours showed superiority in speed and a 13.4% of improvement on average accuracy. (They reported no speed progress when test on higher resolution videos)

Table 1 Tracking accuracy of videos and static images

	accuracy	speed (fps)
Static images	85%	/
Video	93.4%	20

Table 2 Comparisons with method [3] and [7]

	resolution	Speed	accuracy
Our method	320×240	20fps	93.4%
Method[3]	176×120	15fps-33fps	80%
Method[7]	256×256	25fps	/

4. Conclusion

In this paper, we have proposed a lane detection method by combining PHT with LSM to solve the special problems in vision-based lane detection: broken lanes are hard to be detected efficiently, which is usually ignored in other algorithms. In the experimental results, we have shown the simplicity and stability of the proposed lane tracking algorithm which broken lanes with slightly changed colors according to various road conditions have been detected and tracked. In addition, curved lanes were detected and tracked successfully in which they were expressed by the tangent line of the lane. We expect that the proposed method is applicable to many real-time applications, for example intelligent vehicle, navigation systems, and so on.

References

[1] Andrew Reed Bacha, "Line Detection and Lane Following for an Autonomous Mobile Robot," IGVC Intelligent Ground Vehicle Competition, 2004.

[2] Hao Zhang, Dibo Hou, and Zekui Zhou, "A Novel Lane Detection Algorithm Based on support Vector Symposium 2005, Hangzhou, China, August 22-26.

[3] ZuWhan Kim, "Robust Lane Detection and Tracking in Challenging Scenarios," 2007 IEEE Transactions on Intelligent Transportation Systems VOL.8 NO.4, December 2007.

[4] Yue Wang, Eam Khwang Teoh and Dinggang Shen, "Lane Detection Using B-Snake," 0-7695-0446-9 /99 $10.00 0 1999 IEEE, 1999.

[5] Yue Wang, Dinggang Shen and Eam Khwang Teoh, "Lane Detection Using Catmull-Rom Spline," IEEE International Conference on Intelligent Vehicles, 1998.

[6] Hong Wang and Qiang Chen, "Real-time Lane Detection in Various Conditions and Night cases," 2006 IEEE Intelligent Transportation Systems Conference Toronto, Canada, September 17-20, 2006.

[7] Gu Jieyu, Ye Xiuqing, Gu Weikang, "A Real-time Lane Detection in Structural Road," Chinese Journal of Sensors and Actuators, Vol.18, No.3, Sep.2005.

[8] N. Kiryati, Y. Eldar and A.M. Bruckstein, "A Probabilistic Hough Transform," Pattern Recognition Vol.24, pp. 303-316, 1991.

[9] Hervé Abdi, "The Method of Least Squares,: http://www.utd.edu/~herve

[10] T. Sun, S. Tsai and V. Chan, "HSI Color Model Based Lane-Marking Detection," IEEE Intelligent Transportation Systems Conference Toronto, Canada, September 17-20, 2006.