A Multiple-Lane Vehicle Tracking Method for Forward Collision Warning System Applications

Yuan-Fu Li, Chia-Chi Tsai, Yi-Ting Lai, and Jiun-In Guo

Department of Electronics Engineering, National Chiao Tung University, HsinChu, 300 Taiwan, R.O.C.
E-mail: apple.35932003@gmail.com

Abstract—This paper proposes a vehicle tracking method for the Forward Collision Warning System (FCWS). The proposed method applies vehicle detection and feature tracking to each frame in the videos to enhance the detection accuracy and increase the stability of vehicle localization. The proposed method can be applied to three-lane FCWS which includes left, main, and right lanes. The performance of the proposed three-lane FCWS can achieve 720x480 video@183 fps in average when realized in PC with Intel Core i7 processor with average detection accuracy of 94.05% at daytime and 86.90% at night. The proposed method is also implemented on Freescale i.MX6 embedded platform with a USB webcam to capture the video. Under the D1 (720x480) resolution, the performance of the proposed 3-lane FCWS can achieve 29 fps.

I. INTRODUCTION

In recent years, a plenty of manufacturers have developed some useful ADAS functions, which plays important roles in the safety driving for vehicles. However, most of the ADAS algorithms are realized in PC instead of demonstrating the performance realized in embedded systems. Therefore, we propose an efficient FCWS, which is able to be implemented on embedded systems with a single camera, in order to meet the demands from manufacturers. In real driving environments, there are lots of problems, which may cause ADAS functions to fail, such as bad weather, changes of light and shadow and different scenes. As the test result shown in [1], only one simple feature, such as color information, is not enough for vehicle localization.

To solve the problem, feature tracking, which utilizes color and gradient direction information extracted from the detected car in previous frame to find position of cars in current frame, is adopted when the vehicles are undetected in current frame. With the aforementioned solutions, we are able to apply the proposed FCWS function in real world driving environments with low miss rate and low error rate.

The rest of this paper is organized as follows. Section II introduces related works of FCWS. Section III presents the proposed FCWS algorithm. Section IV shows the experimental results of proposed system. The performance and comparison of the proposed design to others is also included. The conclusion and future works are given in Section V.

II. RELATED WORKS

This section introduces some previous works about FCWS [1-9]. We summarize these designs in different aspects as below.

A. Existing FCWS Design

Huei-Yung et al. [1] proposed a vision based FCWS design which has three main steps: identify the frontal vehicle, estimate the distance to the vehicle and generate a warning signal if the range is smaller than a safety threshold. They use the gradient image captured from a single camera as the input of their system.

B. Vehicle Detection

Zehang Sun et al. [2] claimed two steps for detecting vehicles, Hypothesis Generation (HG) and Hypothesis Verification (HV), as shown in Fig. 1. Generally speaking, the computing time in HV is more than the one in HG, so eliminating most of free-driving space in HG by apparent vehicle features is needed. There are some famous methods of HG such as shadow feature at daytime [3] and tail-light feature at night [4]. SVM with HOG features [5] and Adaboost with Haar-like features [6] are famous methods of HV.

Fig. 1. Two-step vehicle detection [2]

C. Vehicle Tracking

Xianbin Cao et al. [7] predicted the position of the target by Kalman filter, and employed histogram matching algorithm near the predicted position to get the measurements. In histogram matching algorithm, they chose H-S histogram and Battacharyya distance to calculate similarity between the template and candidates.

Qiu Tu et al. [8] applied scale-invariant feature transform (SIFT) algorithm to match vehicles. SIFT can extract distinctive features from image to match different views, colors and shapes. However, using SIFT feature cannot meet real-time requirement. Consequently, Liu Yang et al. [9] used speeded-up robust feature (SURF), which has much less execution time with slightly worse performance than SIFT, to match vehicles.

III. PROPOSED METHOD
The proposed system is based on our previous work [10]. We optimize the method [10] by adding a feature-based tracking method to increase its detection accuracy, especially in the inclement weather conditions. The proposed FCWS algorithm is shown in Fig. 2. The new proposed algorithm is a feature tracking method which is marked as a gray rectangle in Fig. 2.

The FCWS initialization, which includes global parameter setting and vehicle model constraint, makes system suitable for various scenes. After initialization of FCWS, vehicle detection and vehicle tracking are adopted to find nearest vehicles in the region of three lanes. Finally, we estimate the values of distance between these vehicles and camera. According to the estimated distance, we can warn the driver to keep a safe distance from the forward vehicle. Distance estimation, vehicle detection and feature tracking will be described in the following subsections, respectively.

A. Distance Estimation

This system utilizes the position-based distance estimation equation from [11] which is shown in Eq. (1) and the sketch map is shown in Fig. 3.

\[ D = \frac{F_c \cdot H_c}{y_b - y_k} \]  

(1)

B. Vehicle Detection

As the vehicle detection procedure described in [10], the system extract shadow feature at daytime and tail-light feature at night. After vehicle detection, the detected vehicle is represented by a rectangular window and estimated the distance from the camera. Finally, the system stores the information of features which includes x-y coordinate of left-up point, width, height and estimated distance of detection window in the detection buffer.

C. Feature Extraction and Comparison

First, feature extraction is the most important part of tracking system. The structure of the proposed features includes the information of the descriptor window: left-top x-y position of the window, width & height of the window, Hue histogram of the window and gradient direction histogram of each keypoint. The feature extraction steps are described as below:

1. Move up the position of vehicle window by \( \frac{\text{window_height}}{8} \) as the required vehicle window and record the top-left x-y position, width of window and height of window.
2. Use the color image. Transform the pixels in center region of the vehicle window (see Fig. 4) from RGB color space to Hue color space, and accumulate the Hue histogram according to the value.
3. Evenly set 16 keypoints and record the associated relative positions in the vehicle window (see Fig. 5).
4. For each keypoints in the vehicle window, we expand an 8*8 descriptor window (see Fig. 6) and use the gray image to calculate gradient magnitude \( M \) and gradient direction \( \theta \) for each pixel among the descriptor window.

The system adopts the following three equations Eqs. (2), (3), and (4) and the relative position information to calculate the gradient magnitude of each pixel. In Fig. 7, the origin is one of the position of keypoints. In Fig. 8, U and D means upper and lower gray value, L and R means left and right gray value.

\[ M = \sum_{i} w_i m_i \]  

(2)

\[ m_i = (D_i - U_i)^2 + (R_i - L_i)^2 \]  

(3)

\[ w_i = e^{-\left(x_i^2 + y_i^2\right)/2} \]  

(4)
The system utilizes the method (see Eq. (5) and Fig. 9) to quantize the gradient direction into 8 directions with much less computation.

\[
\begin{align*}
\alpha &= R - L \\
\beta &= D - U
\end{align*}
\]  
(5)

Fig. 9. Gradient direction quantization

5. Accumulate the gradient direction histogram according to the quantized gradient direction and gradient magnitude value of each pixel in the descriptor window.

After feature extraction, the proposed feature comparison method is shown below.

1. Calculate the sum of absolute difference (SAD) value between two hue histograms of main colors \(D_{\text{hue}}\). Main colors are shown in Fig. 10. The colors that over the red line are called main colors.

2. Calculate the SAD value of gradient direction histogram of each keypoint between two descriptors \(D_{\text{dir}}\).

3. Sum up and multiply \(D_{\text{hue}}\) and \(D_{\text{dir}}\) by weights through (6) to get the difference value of two descriptors \(D_{\text{d}}\).

\[
D_{\text{d}} = \alpha D_{\text{hue}} + \beta D_{\text{dir}}, \quad \alpha = 0.3, \beta = 0.7
\]  
(6)

D. Vehicle Tracking Procedure

Start from setting single-lane vehicle tracking. According to the existence of detection feature and track feature, there are four situations which is explained below:

1. Detection feature exists but track feature does not exist
   Extract vehicle feature from detection buffer and take this feature as main track feature.

2. Detection feature does not exist but track feature exists
   Utilize the track feature extracted from the first case to search nearby region of vehicle position (see Fig. 11) with different size of search windows (see Fig. 12). The window with dotted line is the location of vehicle window in last frame.

3. Detection feature and track feature both exist
   If the detection window match one of these two conditions, the system will choose detection feature directly. The conditions are the estimated distance of detection window is less than 20, lower y-position of detection window is larger than lower y-position of previous vehicle window.
   For the other cases, the system will compare the track feature extracted from the previous frame \(T_{\text{fp}}\) and detection feature of the current frame \(D_{\text{fc}}\) with track feature extracted from first case \(T_{\text{ff}}\). If the difference between \(T_{\text{fp}}\) and \(T_{\text{ff}}\) is smaller than the difference between \(D_{\text{fc}}\) and \(T_{\text{ff}}\), we will choose \(T_{\text{ff}}\) in current frame. Otherwise, we will choose \(D_{\text{fc}}\) instead.

4. Detection feature and track feature both do not exist
   In this case, we will consider that there is no vehicle in the ROI of the current frame.

Before applying the single-lane vehicle tracking to the three-lane, the number of detection buffer is expanded to three and a new ROI is set up (see Fig. 13).
specification and performance of PC and i.MX6 is shown in Table 1 and Table 2, respectively.

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>Operating System</th>
<th>Input</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.MX6</td>
<td>Quad-core ARM® Cortex®-A9</td>
<td>Linux (64 bit)</td>
<td>Logitech HD Pro C920 webcam</td>
<td>29 fps</td>
</tr>
</tbody>
</table>

Table 2: The specification and performance of i.MX6

Experimental results are shown including accuracy rate, detection rate, miss rate, and error rate, which are defined in Table 4 and (7). Video test results are shown in Table 5 and Table 6, which include daytime (case 1), daytime with non-ideal shadow (case 2), daytime with rain (case 3), night (case 4), night (case 5), and the comparison between existing works and proposed system is shown in Table 7. Compare to [10] (see Table 5 and Table 6), the proposed method improves the detection rate and miss rate obviously.

<table>
<thead>
<tr>
<th></th>
<th>Detection rate (%)</th>
<th>Accuracy rate (%)</th>
<th>Miss rate (%)</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>93.53%</td>
<td>93.20%</td>
<td>6.46%</td>
<td>7.38%</td>
</tr>
<tr>
<td>Case 2</td>
<td>92.18%</td>
<td>91.41%</td>
<td>7.82%</td>
<td>8.59%</td>
</tr>
<tr>
<td>Case 3</td>
<td>80.19%</td>
<td>74.92%</td>
<td>29.11%</td>
<td>60.94%</td>
</tr>
<tr>
<td>Case 4</td>
<td>80.19%</td>
<td>74.92%</td>
<td>29.11%</td>
<td>60.94%</td>
</tr>
<tr>
<td>Case 5</td>
<td>80.19%</td>
<td>74.92%</td>
<td>29.11%</td>
<td>60.94%</td>
</tr>
</tbody>
</table>

Table 5: Average test results of multiple-lane FCWS in [10]

Table 6: Average test results of the proposed multiple-lane FCWS

V. CONCLUSIONS

Compared to the FCWS without feature tracking, the proposed system makes a great progress of increasing detection rate in inclement weathers like raining day, which may has many interference such as windshield wiper and raindrops. Performance of the proposed system can reach 183fps@720x480 at PC and 29fps@720x480 at i.MX6 embedded system. The proposed FCWS design can offer drivers a safer driving environment.

Table 7: Comparison among other works and the proposed system

REFERENCES