A Real-time Visual-based Front-mounted Vehicle Collision Warning System

Chiung-Yao Fang, Jui-Hung Liang, Chaio-Shan Lo, and Sei-Wang Chen
Department of Computer Science and Information Engineering
National Taiwan Normal University
Taipei, Taiwan
violet@csie.ntnu.edu.tw; infinte@hotmail.com; milkchickenpanda@gmail.com; schen@csie.ntnu.edu.tw

Abstract—This paper proposes a real-time collision warning system for the front of a vehicle, which contains three stages: lane marking detection, vehicle detection, and vehicle distance estimation. Sobel edge detection and Hough transform techniques are used in the lane marking detection stage to extract lane marking information. In the vehicle detection stage, two very different situations are considered: daytime and nighttime. In the daytime, two kinds of features, vehicle shadows and horizontal edges, are extracted to detect the locations of vehicles. These two features can respectively be obtained by Otsu’s method and a horizontal edge detection method. For the nighttime or in days of poor visibility, vehicle tail light features are used to detect the location of vehicles. These features can be obtained from the Cr component of the YCrCb color model and the hue component of the Hue, Saturation and Intensity (HSI) color model respectively. In the vehicle distance estimation stage, the system estimates the distance between the host vehicle and the front vehicles using exponential functions. Some warning messages will be output to the drivers if necessary.

In this study, a recorder is set on the front windscreen to obtain the input sequences. The experimental results show that the proposed method has great stability and usability. We intend for the proposed method to be embedded into driving assistance systems and installed in vehicles in the future.

Keywords—distance estimation, lane marking detection, vehicle collision warning system, vehicle detection.

I. INTRODUCTION

Drivers are easily affected by the unexpected behavior of neighboring vehicles, which sometimes causes traffic accidents. Therefore, many safety driving assistance systems (DAS) have been studied by researchers and developed by car companies to avoid these kinds of accidents. The DAS include blind spot detection systems [1], lane departure warning systems [2], collision avoidance systems [3, 4], and dangerous driving event detection systems [5]. This paper proposes a vision-based front-mounted vehicle collision warning system to detect the distances between the host vehicle and its neighbors, and warn the driver of the host vehicle if necessary.

The authors believe vision-based systems are superior to laser systems, radar systems, and supersonic systems for detecting the front neighbors in the right lane, left lane, and in the same lane as the host vehicle. Therefore, in this study a recorder is set on the front windscreen to obtain the input sequences. Figure 1 shows some input frames obtained by the recorder.

In Figure 1, we observe that the content of input frames varies dependent on the time of day and the weather conditions. In Figures 1(a) and (d), examples of a sunny day are shown. Figure 1(b) is a rainy day, Figure 1(a) is daytime, Figure 1(c) nighttime, and Figure 1(e) and (f) show tunnels and under bridges respectively. Many problems need to be considered and solved when the vision-based warning system is developed. For example, the windscreen wipers will trigger disturbances in the content of the input frames when taken during a rainy day; the darkness may greatly affect the lane detection results for a frame taken at night; and both the light change in the tunnel and the shadow under the bridges (shown in Figures 1(e) and (f)) confuse most vision-based systems. Our proposed system is shown to solve these problems and obtain desirable experimental results.

II. SYSTEM FLOWCHART

A flowchart of the front-mounted vehicle collision warning system is shown in Figure 2. This system contains three stages: lane marking detection, vehicle detection, and vehicle distance prediction. Detecting the lane markings to define the vehicle search area can reduce the computation time of the system.
Thus, in some situations, where the lane markings are clear (such as on sunny days and in the daytime) the system first detects the lane markings. In this stage, Sobel edge detection and Hough transform [6] techniques are used to extract the lane markings.

![Flowchart of the front-mounted vehicle collision warning system.](Image)

Figure 2. Flowchart of the front-mounted vehicle collision warning system.

In the second stage, the system detects the forward vehicles in the right lane, left lane, and in the same lane as the host vehicle. Two very different situations are considered: daytime (the tail lights of the vehicles are turned off) and nighttime (the tail lights of the vehicles are turned on). During the day, shadows and horizontal edges of vehicles are extracted as the features with which to verify the locations of the vehicles. These two features can be obtained respectively by Otsu’s method [7] and a horizontal edge detection method. In the nighttime or on rainy days, tail light features of vehicles are used to verify the location of the vehicles. These features can be obtained from the Cr component of the YCrCb color model and the hue component of the HSI color model respectively.

![Flowchart of the lane marking detection stage.](Image)

Figure 3. Flowchart of the lane marking detection stage.

Finally, the system predicts the distance between the host vehicle and the forward vehicles using exponential functions in the vehicle distance estimation stage. Some warning messages will be output to the drivers if necessary. For example, if the estimated distance is less than a standard safety distance, or if the estimated distance decreases too quickly.

III. LANE MARKING DETECTION STAGE

Figure 3 shows a flowchart of the lane marking detection stage. The system first translates the input frames into gray images before detecting edges with the Sobel edge detection algorithm. The Hough transform technique is then used to detect the lines which consist of the edges. Finally, the candidates for left and right lane markings are extracted and verified from these lines.

Since the lane markings always appear on the lower part of the input frames, the Sobel edge operator is applied only to the bottom quarter of the input frames. This not only avoids the effect of noise, but also reduces the computation time. Figure 4(a) shows an input frame and Figure 4(b) shows its corresponding edge detection result. Notice that the edges detected by the Sobel edge detection algorithm are clear and sufficiently strong for the following stages.

Initially, after applying the Hough transform technique on the detected edges, the lines whose slopes are in the ranges [-0.9, -0.5] (for the right lane marking) and [0.5, 0.7] (for the left lane marking) are regarded as candidates for lane markings. Moreover, the intersection point (vanish point) of these two lane markings should fall in some range, here the range is set between 0.4 to 0.6 of the image width, because the vehicle always drives between two lane markings. Based on these criteria, the system selects the lines which pass through the largest number of edge pixels to be the left and right lane markings respectively. The yellow lines shown in Figure 4(c) indicate the line detection result by the Hough transform method. Figure 4(d) shows the lane marking detection result. The yellow and green lines indicate the left and right lane markings respectively.

![A lane marking detection example.](Image)

Figure 4. A lane marking detection example. (a) An input frame. (b) Its corresponding edge detection result. (c) The line detection result. (d) The lane marking detection result.

The system will then monitor these two lane markings in following frames. Since the lane markings do not greatly change in two successive frames, the system checks two criteria to keep the stability of the detection result. Letting the slope of the same lane marking at frames t-1 and t be $m_{t-1}$ and $m_t$, respectively, the first criterion is

$$|m_t - m_{t-1}| < \max[0.2m_{t-1}, 0.1]$$

(1)

Secondly, the end-points of the same lane marking in two successive frames should be close. Each lane marking contains two end-points, one is the intersection point of the left and right lane markings, and the other is the intersection point of the lane marking and the bottom horizontal boundary line of the input frame. These two criteria help the system to verify the lane
markings correctly. However, if no lane marking candidates satisfy these two criteria at frame $t$, then the system stops the monitoring process and tries to re-detect the lane markings.

Figure 5 shows two examples of lane marking verification. In Figure 5(a), the vehicle is changing its lane. The system did not detect the lane markings correctly, and thus keeps the previous detection result until the lane markings are correctly detected again (two seconds later). In Figure 5(b), the lane markings are detected incorrectly at first, but the system automatically corrects the lane markings by re-detecting them.

In Figure 6, some input frames and their corresponding results of lane marking detection are shown. We observe that the detection results are correct in the daytime (Figure 6(a)) and in the evening (Figure 6(b)), even when the lane marking consists of dashed lines in Figure 6(b). Figures 6(c) and (d) show the detection results when a bridge appears in front of the vehicle. In these varying light situations, the detection results are both correct. Figure 6(e) shows a tunnel example. It shows that if the lane markings are solid double lines, the system selects one of the lines to be the lane marking. Figure 6(f) shows an example of lane detection in the rain. In this case, the left lane marking is a solid yellow line, and a windscreen wiper causes heavy noise. However, the system detects the lane markings successfully.

IV. VEHICLE DETECTION STAGE

As mentioned above, in the vehicle detection stage, two very different kinds of situations, day and night, are considered. The input frames taken during the day are when the tail lights of the vehicles are turned off (see Figure 6), while at night, the tail lights of the vehicles must also be accounted for (see Figure 1 (c)). The system extracts different features to detect the vehicles in these two situations.

A. Daytime

Figure 7 shows the flowchart for vehicle detection during the day. Two features, vehicle shadows and horizontal edges are extracted to detect the locations of vehicles. These two features can be obtained respectively by Otsu’s method [Ots79] and a horizontal edge detection method. These two features are then combined to verify the vehicles.

![Flowchart showing vehicle detection during the day.](image)

In the shadow feature extraction stage, the system first transfers the input color frames into gray ones before applying Otsu’s method. Given a frame whose size is $M \times N$ pixels, where each pixel has a gray level $L$. The following Otsu’s
method algorithm can calculate a threshold to extract the shadow features.

1. First, construct the normalized histogram of the input image. Let \( n_i \) indicate the number of pixels whose gray level is \( i \). The normalized histogram value \( p_i \) can be calculated by

\[
p_i = \frac{n_i}{MN}, \quad \text{where} \quad i = 0, 1, 2, ..., L-1.
\]  

(2)

These values can be regarded as the occurrence probabilities of gray levels in the frame.

2. Compute the sum of the occurrence probabilities of gray levels in the range \([0, k]\), the cumulative probability as

\[
P(k) = \sum_{i=0}^{k} p_i, \quad \text{where} \quad k = 0, 1, 2, ..., L-1.
\]

(3)

3. Compute the mean of the occurrence probabilities of gray levels in the range \([0, k]\), the cumulative mean, as

\[
m(k) = \sum_{i=0}^{k} ip_i, \quad \text{where} \quad k = 0, 1, 2, ..., L-1.
\]

(4)

4. Compute the mean of the occurrence probabilities of gray levels in the range \([0, L-1]\), the global cumulative mean \( m_{\sigma} \), as

\[
m_{\sigma} = \sum_{i=0}^{L-1} ip_i.
\]

(5)

5. Compute \( \sigma^2_{\sigma}(k) \).

\[
\sigma^2_{\sigma}(k) = \frac{[m_{\sigma}P(k) - m(k)]^2}{P(k)[1-P(k)]}, \quad \text{where} \quad k = 0, 1, 2, ..., L-1.
\]

(6)

6. Find the threshold \( k^* \) which maximizes \( \sigma^2_{\sigma}(k) \).

\[
k^* = \arg \max_{i} \sigma^2_{\sigma}(k), \quad \text{where} \quad k = 0, 1, 2, ..., L-1.
\]

(7)

Figure 8 shows an example of Otsu’s method.

Figure 8 shows an example of Otsu’s method. Figure 8(a) is the gray image, and Figure 8(b) shows its corresponding normalized histogram. The red line indicates the position of the threshold. Figure 8(c) shows the thresholding results. We can observe that the shadows of the three front vehicles are all detected correctly but not completely. Thus, a dilation operator of the morphology will be applied to extract complete shadow features.

From Figure 8, we observe that many non-shadow pixels are also detected by Otsu’s method. Thus, the system should define the search region of shadow features to avoid the effect of other noises. The regions are automatically and dynamically defined by the lane markings extracted in Section 3. The system extends the search region to the left and right lanes of the host vehicle by a mathematical formula (see Figure 9(a)). Figure 9(b) shows the search region of the shadow feature in different frames. The red regions indicate the areas where new features may appear; the system applies the feature detection methods on these regions. The green regions indicate the areas where the system detects the feature tracking techniques where no new features appear. Notice that the search regions define the areas for detecting not only the shadow features, but also the edge features.

![Figure 9. To extend the search region of shadow feature.](image)

Figure 10 shows an example of shadow feature extraction. Figure 10(a) is the input frame; Figure 10(b) shows the thresholding result after applying Otsu’s method; Figure 10(c) represents the result of dilation; and Figure 10(d) shows the extracted shadow features. These features are enclosed by the blue boxes. We see there is some noise (the two boxes near the bottom of the frame) reflected on the windscreen, which is also detected and will be removed by the feature combination step.

Horizontal edges are extracted in the edge feature extraction, since most vehicles contain this kind of edge. Figure 11 illustrates an example of edge feature extraction. Figure 11(a) is the input frame, and Figure 11(b) shows the horizontal edges detected by a 1 by 3 mask. We see that many horizontal edges that do not belong to the vehicles are also detected. Thus, the system binarizes the input frame (Figure 11(c)) and subtracts this binarization image from Figure 11(b) to obtain Figure 11(d). We then filter the noise that does not fall in the search region shown in Figure 9(b), obtaining Figure 11(e). Now the system can extract the horizontal edges whose width is between 30% to 100% of the lane width, shown in Figure 11(f). Figure 11(g) shows the extracted edge features, which are enclosed by the four pink boxes.
Finally, Figure 11(h) shows the feature combination result of combining the shadow features (Figure 10(d)) and edge features (Figure 11(g)). The vehicles exist at the locations containing the two different kinds of features; these are enclosed by the red boxes.

B. Nighttime

Vehicle tail light features are extracted and verified to locate the vehicles at night. These features can be obtained from the Cr component of the YCrCb color model and the hue component of the HSI color model respectively.

Figure 12 illustrates two examples of tail light feature extraction. Figure 12(a) shows the input frame. Figures 12(b) and (c) show the Cr component of the YCrCb color model and the hue component of the HSI color model respectively. Figure 12(d) represents the binarized intersection result of these two components. Two criteria are then used to extract the vehicle tail light features. First, the pair of tail lights belonging to a single vehicle should have almost the same height. Second, the distance between one pair of tail lights should be less than the lane width. Notice that the lane width calculated from the lane markings projected on the input frames is relative to the y position (that is, vertical height) of the features. Moreover, if the lane markings cannot be detected, the system will estimate them automatically.

Figure 12. Two examples of vehicle tail light feature extraction.

Figure 13 shows three examples of vehicle tail light feature detection. Figure 13(a) shows the situation where the lane markings are detected successfully. The vehicle tail light feature is enclosed by a yellow box. Figure 13(b) shows the situation when the lane markings cannot be detected. The vehicle tail light features are enclosed by blue boxes.

V. DISTANCE ESTIMATION STAGE

To estimate the distance to the forward vehicle, we first obtain the mapping of the real distances (m) and vertical height (h) (counting from the bottom of the frame by a pixel unit) of lane marking pixels manually, and then attempt to find a
suitable estimation function. In Figure 14, the red and green lines are the mapping results obtained from different input sequences. Figures 14(a) and (b) show the mapping results from the input frames whose ratio of width to height is 16:9 and 4:3 respectively. We observe these mapping results are similar to exponential functions. Thus, this study uses exponential functions to estimate the distance between the host vehicle and its neighbors.

The exponential function is defined as,

\[ d = \alpha \exp(\beta y) \]  \tag{8} \]

If the input frame has a width to height ratio of 16:9, we set \( \alpha = 0.4 \) and \( \beta = 0.053 \). Otherwise, if the ratio is 4:3, we set \( \alpha = 0.5 \) and \( \beta = 0.085 \). These two exponential functions are shown in Figures 14(a) and (b) respectively as the blue lines.

Figure 14. The mapping of the real distance (m) and vertical height (h) of lane marking pixels.

Figure 15 shows some examples of distance estimation of the vehicles in front. In Figure 15(a) the vehicles in front are shown in red if the estimated distance between them and the host vehicle is within 10 meters. In comparison, the vehicles in front are represented by green and blue if the distance between them and the host vehicle is within 10 to 20, or more than 20 meters respectively. Figure 15(a) shows a daytime example containing three input frames and the corresponding distance estimation results. These input frames, whose ratio of width to height is 4:3, are obtained from an Interceptor F-1 recorder set in a Nissan Bluebird vehicle. In this example, the system first detects three vehicles ahead, one in the left lane and two in the right lane. We can see that the color representing the vehicle closest to the host vehicle turns from red (5.40 meters) to green (11.61 meters) in 3 seconds. Therefore, the forward vehicle is driving faster than the host vehicle.

In Figures 15(b), (c), and (d) the input frames, whose ratio of width to height is 16:9, are obtained from a Guardian ADR36 recorder set in a Ford Festiva vehicle. Figure 15(b) shows a daytime example containing three input frames and their corresponding distance estimation results. In this example, the lane markings have been detected and the extracted features are the vehicle shadows and the horizontal edges. Figure 15(c) shows an example in the evening where it is also raining. A truck is driving in the right lane. In this example, the lane markings have also been detected but the extracted features are the tail lights. We can see the truck is detected successfully. Figure 15(d) shows an example at night where only the tail lights can be detected in the frames. In this example, the lane markings cannot be detected, and accordingly, the system estimates them. We observe the vehicle in the right lane is driving faster than the host vehicle. Notice that in this example, the estimated distance between the vehicle in the left lane (the blue one) and the host vehicle is inaccurate since the estimation of the lane markings may not be correct.

Figure 15. Some examples of forward vehicle distance estimation. (a, b) An example in the daytime, (c) an example in the evening with rain, and (d) an example at night.

VI. EXPERIMENTAL RESULTS

In the experiments for lane marking detection, 39 test sequences were collected over 108 minutes, including 1 rain sequence, 3 tunnel sequences, and 35 sequences obtained from early morning till sunset. The test sequences were obtained from a Guardian ADR36 and an Interceptor F-1 whose width to height ratios are 16:9 and 4:3 respectively. The frame rates are both 20 frame/sec.

Table 1 shows the experimental results for the lane marking detection. The 39 testing sequences include 11 sequences
obtained from the Interceptor F-1 recorder and 28 sequences obtained from the Guardian ADR36 recorder. The total number of testing frames is 118439. The term “actually correct” describes the situation where the detected lane markings exactly overlap the real lane markings seen in the frames (see Figure 16(a)). The term “partially correct” indicates that the maximum distance between the detected lane markings and the real lane markings in the frames was only a few pixels (see Figure 16(b)). The term “incorrect” indicates that the detected lane markings and the real lane markings were far from each other (see Figure 16(c)).

From Table I, the incorrect proportions are zero for the raining and tunnel testing sequences since the host vehicle did not change its lane. Thus, we conclude that incorrect lane detection always occurs when the host vehicle is performing a lane change. Moreover, the incorrect proportions are similar irrespective of the recorder used.

![Figure 16. Some experimental results of lane marking detection. (a) An actually correct example, (b) a partially correct example, and (c) an incorrect example.](image)

Table II shows a summary of the lane marking detection results. The total number of testing frames is 118439. The actually correct proportions of left and right lane markings are 92.2% and 88% respectively. This difference can be explained due to the right lane markings containing more dashed lines than left lane markings. The partially correct proportions of left and right lane markings are 5.3% and 8.9% respectively. The incorrect proportions of left and right lane markings are 2.5% and 3.1% respectively.

Table III shows the experimental results for vehicle detection during the day. The vehicles are separated into three classes: appearing in the left lane, middle lane, and right lane. Totally 427 vehicles appear in the test sequences. The term “number of frames” indicates the number of the frames in which some vehicles appear in the sequences, and the term “number of detected frames” indicates the number of frames in which vehicles are detected successfully. From Table III, we know that no vehicle was driving in the left lane during the raining sequence. The precision of vehicle detection in the middle lane is higher than the other lanes, except in the tunnel scenario. We believe the light in the tunnel to affect the precision of vehicle detection.

Table IV gives a summary of the vehicle detection results from the daytime samples. From Table IV, the precision of vehicles in the left, middle, and right lanes are 86.8%, 88.5%, and 86.6% respectively. Thus the average precision is 87.3% and the recall is 98%.

### Table I. The Experimental Results for the Lane Marking Detection.

<table>
<thead>
<tr>
<th>Testing sequences (the type of recorder)</th>
<th>Left lane marking</th>
<th>Right lane marking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actually correct</td>
<td>Partially correct</td>
</tr>
<tr>
<td>11 sunny sequences (Interceptor F-1)</td>
<td>40740</td>
<td>1352</td>
</tr>
<tr>
<td>No. of frames</td>
<td>94.4%</td>
<td>3.1%</td>
</tr>
<tr>
<td>Proportion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 raining sequence (Guardian ADR36)</td>
<td>4693</td>
<td>195</td>
</tr>
<tr>
<td>No. of frames</td>
<td>96%</td>
<td>4%</td>
</tr>
<tr>
<td>Proportion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 tunnel sequences (Guardian ADR36)</td>
<td>7860</td>
<td>240</td>
</tr>
<tr>
<td>No. of frames</td>
<td>97%</td>
<td>3%</td>
</tr>
<tr>
<td>Proportion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 other sequences (Guardian ADR36)</td>
<td>55923</td>
<td>4476</td>
</tr>
<tr>
<td>No. of frames</td>
<td>89.7%</td>
<td>7.2%</td>
</tr>
<tr>
<td>Proportion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table II. A Summary of Lane Marking Detection Results.

<table>
<thead>
<tr>
<th>No. of frames</th>
<th>Actually correct</th>
<th>Partially correct</th>
<th>Incorrect</th>
<th>Actually correct</th>
<th>Partially correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>109216</td>
<td>92.2%</td>
<td>5.3%</td>
<td>2.5%</td>
<td>88%</td>
<td>8.9%</td>
<td>3.1%</td>
</tr>
<tr>
<td>Proportion</td>
<td>97.5%</td>
<td>2.5%</td>
<td>96.9%</td>
<td>3.1%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table III gives a summary of the vehicle detection results from the daytime samples. From Table IV, the precision of vehicles in the left, middle, and right lanes are 86.8%, 88.5%, and 86.6% respectively. Thus the average precision is 87.3% and the recall is 98%.
TABLE III. THE EXPERIMENTAL RESULTS OF VEHICLE DETECTION DURING THE DAY.

<table>
<thead>
<tr>
<th>Testing sequences</th>
<th>Vehicle in the left lane</th>
<th>Vehicle in the middle lane</th>
<th>Vehicle in the right lane</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 sunny sequences (Interceptor F-1)</td>
<td>No. of appearance frames 9951</td>
<td>No. of detected frames 8972</td>
<td>Precision 90.1%</td>
</tr>
<tr>
<td></td>
<td>3916</td>
<td>3665</td>
<td>93.6%</td>
</tr>
<tr>
<td></td>
<td>13794</td>
<td>11875</td>
<td>86.1%</td>
</tr>
<tr>
<td>1 raining sequence (Guardian ADR36)</td>
<td>No. of appearance frames 0</td>
<td>No. of detected frames 0</td>
<td>Precision 0%</td>
</tr>
<tr>
<td></td>
<td>303</td>
<td>287</td>
<td>94.7%</td>
</tr>
<tr>
<td></td>
<td>2017</td>
<td>1857</td>
<td>86.6%</td>
</tr>
<tr>
<td>3 tunnel sequences (Guardian ADR36)</td>
<td>No. of appearance frames 1257</td>
<td>No. of detected frames 1072</td>
<td>Precision 85.3%</td>
</tr>
<tr>
<td></td>
<td>5426</td>
<td>4426</td>
<td>81.6%</td>
</tr>
<tr>
<td></td>
<td>3565</td>
<td>2795</td>
<td>78.4%</td>
</tr>
<tr>
<td>24 other sequences (Guardian ADR36)</td>
<td>No. of appearance frames 23245</td>
<td>No. of detected frames 19858</td>
<td>Precision 85.4%</td>
</tr>
<tr>
<td></td>
<td>23395</td>
<td>24411</td>
<td>95.8%</td>
</tr>
<tr>
<td></td>
<td>23868</td>
<td>20926</td>
<td>87.7%</td>
</tr>
</tbody>
</table>

TABLE IV. A SUMMARY OF VEHICLE DETECTION RESULTS DURING THE DAY.

<table>
<thead>
<tr>
<th>Daytime</th>
<th>Vehicle in the left lane</th>
<th>Vehicle in the middle lane</th>
<th>Vehicle in the right lane</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of appearance frames</td>
<td>34453</td>
<td>37040</td>
<td>43244</td>
</tr>
<tr>
<td>No. of detected frames</td>
<td>29902</td>
<td>32789</td>
<td>37435</td>
</tr>
<tr>
<td>Precision</td>
<td>86.8%</td>
<td>88.5%</td>
<td>86.6%</td>
</tr>
<tr>
<td>Recall</td>
<td>98%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table V shows the experimental results of vehicle detection at night. In this situation, the lane markings cannot be observed. The total number of vehicles in the test sequences was 107. From Table V, the precision of vehicle detection is 87.3% and the recall is 76%. Compared with the detection results in daytime, the precision is similar but the recall is lower. This means that false negatives occur more often at night. In other words, many vehicles are not detected at night.

TABLE V. THE EXPERIMENTAL RESULTS OF VEHICLE DETECTION AT NIGHT.

<table>
<thead>
<tr>
<th>Nighttime</th>
<th>No. of appearance frames</th>
<th>20315</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of detected frames</td>
<td>17738</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>87.3%</td>
<td></td>
</tr>
<tr>
<td>Recall</td>
<td>76%</td>
<td></td>
</tr>
</tbody>
</table>

VII. CONCLUSIONS

The paper presents a real-time collision warning system to detect and track vehicles in front of a host vehicle. The system can estimate the distances between the host vehicle and the vehicles in front, outputting warning messages to the driver to avoid vehicle collisions. To develop this system, three stages (lane marking detection, vehicle detection, and vehicle distance estimation) were implemented. Extracting different kinds of vehicles, the system can work in two very different situations (day and night), which are distinguished by whether the tail lights of the forward vehicles are turned on or off. The experimental results show that the proposed method has great stability and usability. However, the distance estimation at night may not be accurate and may be improved further. We believe the proposed method can be embedded into driving assistance systems and installed in vehicles in the future.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 99-2221-E-003-019-MY2 and NSC-101-2221-E-003-026-. REFERENCES