Real-Time Front Vehicle Detection Algorithm for an Asynchronous Binocular System*

CHUNG-CHENG CHIU, MENG-LIANG CHUNG AND WEN-CHUNG CHEN
Department of Electrical and Electronic Engineering
Chung-Cheng Institute of Technology
National Defense University
Taoyuan, 335 Taiwan

This paper describes a multi-resolution stereovision system for detecting the front-vehicle in advanced safety vehicles (ASVs). The two asynchronous CMOS cameras in the proposed system are mounted on a platform that can be easily clamped to the rear-view mirror of a vehicle for detecting vehicles ahead. The asynchronous binocular platform provides a small low-cost obstacle detection system for practical ASVs that is easy to set up. The system uses a stereovision vehicle detection algorithm for real-time matching because the exposure times of the CMOS cameras are not synchronous. The algorithm uses a line segment matching module to match the extreme points of the horizontal and vertical edge segments at different resolutions to decrease the search area and computing complexity. As the distance of each matched segment can be calculated from the disparity value, each vehicle can be detected by clustering the segments that have similar distances in a searching and distance estimation module. The system was evaluated using static and dynamic analyses. Experimental results show that the proposed system can robustly and accurately detect the front-vehicles in real time under different illumination and road conditions.

Keywords: intelligent transportation system, driving safety, front-vehicle detection, binocular system, stereovision

1. INTRODUCTION

Passive safety systems such as air bags, seat belts, and anti-lock braking systems are widely used in vehicles. Combining these with active (pre-crash) safety sensors would increase passenger safety by helping to avoid collisions. According to the National Highway Traffic Safety Administration [1], forward collision warning systems (FCWS) can effectively reduce rear-end collisions by 21%. FCWS are on-board electronic systems that monitor the roadway in front of a vehicle and warn the driver when a potential collision risk exists.

The devices currently available for detecting obstacles rely on vision sensors, laser radar, and millimeter-wave radar. Although millimeter-wave radar [2-5] has a good range, its angular resolution is low. Millimeter-wave radar systems have been used on commercial vehicles [6], but they are not suitable for distinguishing vehicles in complex backgrounds, such as urban areas or bumpy roads, and are still very expensive compared with other sensors. Other collision avoidance systems [7-9] have used laser radar as the sensor.

Received March 31, 2009; accepted September 30, 2009.
Communicated by Chih-Yung Chang, Chien-Chung Shen, Xuemin (Sherman) Shen, and Yu-Chee Tseng.
* This work was also partially supported by National Science Council of Taiwan under the Grant No. NSC 96-2221-E-606-026.
Laser radar costs less than millimeter-wave radar but does have the same limitations as the other radar systems that are unable to classify obstacles or recognize lane markings.

Vision-based techniques have become a very active research topic in the field of intelligent vehicles because of the large sensing area, and the potential for detecting and recognizing obstacles and lane markings. Some detection systems use a CCD camera combined with an active sensor, such as millimeter-wave radar [10-12], far infrared [13], or laser scanner [14], to increase the detection performance.

Vision sensors can be divided into mono- and stereovision systems. A number of reports have presented monovision algorithms for recognizing leading vehicles [15-19]. However, vibrations due to the road surface or the vehicle suspension system cause serious distortion in the distance measurements because these approaches rely on a monocular system mounted inside the passenger compartment. Furthermore, distance measurement is very sensitive to the depression angle and altitude parameters of a monocular system.

Stereovision systems are becoming increasingly popular for advanced driver assistance systems and advanced safety vehicles, because they provide distance measurement as well as vehicle detection. The underlying concept of stereovision systems is that fusing the images recorded by two or more cameras and exploiting the difference between them provides an indication of the distance [20].

Stereovision systems involve two processes: the fusion of features observed by two cameras and the reconstruction of their three-dimensional (3D) information. The well-known epipolar constraint [21] is generally used to describe the geometric relationships between stereo images. Labayrade et al. [22] described how a fast disparity space image [23] was computed, and then the similarity in each row of an image pair was used in a vision sensor system [24] that was part of TerraMax; TerraMax was one of five vehicles to complete the 2005 Defense Advanced Research Projects Agency Grand Challenge course, and the only one to use vision as its primary detection sensor during the race, demonstrating the practical applicability of a stereovision system.

All these previous stereovision algorithms were based on a synchronous stereo camera system. The price, size, and complicated setup of synchronous stereo camera systems make them unsuitable for real-world applications. Therefore, this study describes a small low-cost binocular platform that is easy to set up and overcomes the drawbacks of synchronous stereo camera systems. The proposed binocular platform does not meet the epipolar constraints [21] because the exposure times of the CMOS cameras on the binocular platform are not synchronous. The exposure time difference between the left and right cameras in the asynchronous stereo camera system causes two problems: a relative distance error between the ego vehicle and the front vehicle, and an increase in the computational load.

We propose a real-time front vehicle detection algorithm to overcome the problems of computational load and image matching to provide better performance with an asynchronous stereo camera system. The organization of the remainder of this paper is as follows. Section 2 contains a system overview. Section 3 describes the stereovision vehicle detection algorithm. Experimental results are given in section 4, and the conclusions are presented in section 5.
2. SYSTEM OVERVIEW

Previous work [25, 26] on stereovision systems used a synchronous image capture system to obtain stereo pair images from two synchronous cameras. As the synchronous stereo pair images satisfy the properties of epipolar geometry, corresponding points in the stereo pair images can be matched in the corresponding horizontal scan lines. As the cost and size of a synchronous binocular system are prohibitive for most applications, this study proposes a binocular vision system using two low-cost compact CMOS cameras that can capture left and right image pairs at a rate of 30 pairs/s to create stereo pair images. The CMOS camera and the camera platform are shown in Figs. 1 (a) and (b), respectively. In this study, the focal length, $f$, of the CMOS camera was $12 \times 10^{-3}$m and the baseline, $B$, of the binocular platform was $228 \times 10^{-3}$m. Fig. 1 (c) shows the platform installed on the front of the rear-view mirror.

![Binocular vision system](image)

Fig. 1. Binocular vision system.

![Vehicle detection algorithm flowchart](image)

Fig. 2. The flowchart of vehicle detection algorithm.

The flowchart of the vehicle detection algorithm is shown in Fig. 2. The proposed algorithm includes four modules: pre-processing, edge detection, line segment matching, and vehicle search and distance estimation.
3. STEREOVISION FRONT-VEHICLE DETECTION ALGORITHM

The front-vehicle detection algorithm comprises four modules: pre-processing, edge detection, line segment matching, and vehicle searching and distance estimation. The pre-processing module performs downsampling and low-pass filtering processes. The input image pair is downsampled by a factor of $2^n$ to produce multi-resolution images and reduce the processing time. The original image size in our system is $320 \times 240$ pixels, and the stereo image pair is denoted as \{\(L(x, y)\), \(R(x, y)\)\} where \(L(x, y)\) and \(R(x, y)\) are the \(i\)th left and right images, respectively. The image is downscaled by a factor of 2, \(n = 1\), in the \(x\) and \(y\) directions. After downsampling, a 3 × 3 mean value filter is used for anti-aliasing to reduce the noise in the low-resolution images. After filtering, the outputs of image pairs are denoted as \{\(L D_i(x, y)\), \(R D_i(x, y)\)\}. At this point, a coarse-to-fine detection algorithm can be implemented depending on the different resolution images.

The horizontal or vertical edge detection modules include Sobel edge detection [27] and connected-component labeling [28]. After horizontal edge detection and connected component labeling, the bounding boxes of the horizontal lines are represented as \{\(L D_i h B^k\), \(R D_i h B^l\)\} where \(m_h\) and \(n_h\) are the numbers of bounding boxes in the left and right images, \{\(L D_i(x, y)\), \(R D_i(x, y)\)\}, respectively. After the vertical edge detection and connected component labeling, the bounding boxes of the vertical lines are represented as \{\(L D_i v B^p\), \(R D_i v B^q\)\} where \(m_v\) and \(n_v\) are the numbers of bounding boxes in the left and right images, \{\(L D_i(x, y)\), \(R D_i(x, y)\)\}, respectively.

As most of the horizontal and vertical line segments will appear in the left and right images simultaneously, the horizontal and vertical line segments can be matched in each stereo pair image. Therefore, the line segment matching module provides a horizontal/vertical line segment matching algorithm to match the line segments in stereo pairs. The algorithm uses the characteristics of stereo vision and block matching to achieve a coarse-to-fine line segment matching algorithm for different resolutions.

The horizontal line segment matching algorithm operates in two phases. The first phase uses the characteristics of stereo vision to find the candidates for matching horizontal line segments between the bounding boxes, \(L D_i h B^k\), \(R D_i h B^l\). Assuming that the origin is located at the upper left corner of an image, then the matching conditions of the first phase are:

**Condition 1:** Delete all noise where the width of the bounding box is smaller than 2 pixels.
**Condition 2:** The \(x\)-coordinate of the rightmost point of \(L D_i h B^k\) is greater than the \(x\)-coordinate of the rightmost point of \(R D_i h B^l\).
**Condition 3:** The \(x\)-coordinate of the leftmost point of the \(L D_i h B^k\) is located between the \(x\)-coordinates of the leftmost and rightmost points of the \(R D_i h B^l\).
**Condition 4:** \(\left| \frac{\Delta y_L - \Delta y_R}{\Delta y_L} \right| \leq \delta_h\), where \(\Delta y_L\) is the height of the left side of \(L D_i h B^k\) and \(\Delta y_R\) is the height of the right side of the \(R D_i h B^l\).
**Condition 5:** \(\left| \frac{L D_i h B^k(x) - R D_i h B^l(x)}{L D_i h B^k(x)} \right| \leq \epsilon_h\), where the \(L D_i h B^k(x)\) and \(R D_i h B^l(x)\) are...
the y-coordinates of the top left corners of $LD_{i_hBk}$ and $RD_{i_hBj}$, respectively. Here, $\delta_b$ and $\epsilon_b$ are set to 0.1.

In the first phase, each bounding box in the left image, $LD_{i_hBk}$, is taken as the matching reference. Therefore, each bounding box in the left image will have many matching candidates, $CRD_{i_hBk}$, where $u$ is the number of matched bounding boxes in the right image, $RD_{i_hBj}$, after the first phase. If a matching reference in $LD_{i_hBk}$ does not match any candidate from $RD_{i_hBj}$, then the bounding box in $LD_{i_hBk}$ is deleted.

In general, there may be more than one matching candidate for each $LD_{i_hBk}$. Therefore, the second phase uses a matching algorithm to find the best matched bounding box for each matching set, $(LD_{i_hBk}, CRD_{i_hBk})$ where $k$ is the number of bounding boxes in the left image, and $u$ is the number of matched bounding boxes in the right image. The matching algorithm for each matching set is as follows:

**Step 1:** For the low-resolution image $LD_{i}(x, y)$, obtain the left- and rightmost points of a connected component in $LD_{i_hBk}$, and use these as reference points.

**Step 2:** For the low-resolution image $RD_{i}(x, y)$, obtain the left- and rightmost points from each matching candidate in $CRD_{i_hBk}$, and use these as matching points.

**Step 3:** Consider each reference point as the central point of a $3 \times 3$-pixel block. The search window size is $15 \times 9$ pixels centered on the matching point. Use the full-search block-matching algorithm for each reference point in the left image to find the best-matching candidate that has the least total matching difference in the right image.

**Step 4:** Upsample the coordinates of the reference points of the left image and the matching points of the best-matching candidate of the right image to the original images $\{L_i(x, y), R_i(x, y)\}$. The block size is set to $5 \times 5$ pixels centered on the upsampled reference points and the search window size is $7 \times 7$ pixels centered on the upsampled matching points of the best-matching candidate. For the reference points in the left image, use the full-search block-matching algorithm to refine the matching point that has the least total matching difference in the right image.

**Step 5:** Carry out steps 1-4 in succession until all of the matching set is processed.

The vertical line segment matching algorithm also contains two phases. The first phase uses the characteristics of stereo vision to find candidates for the matching vertical line segments between the bounding boxes, $\{LD_{i_vBp}, RD_{i_vBq}\}$. The matching conditions of the first phase are:

**Condition 1:** Delete all noise where the height of the bounding box is smaller than 2 pixels.

**Condition 2:** The x-coordinate of the leftmost point of $LD_{i_vBp}$ is greater than the x-coordinate of the leftmost point of $RD_{i_vBq}$.

**Condition 3:** The x-coordinate of the rightmost point of $LD_{i_vBp}$ is greater than the x-coordinate of the rightmost point of $RD_{i_vBq}$.

**Condition 4:** $\frac{|\Delta x_L - \Delta x_R|}{\Delta x_L} \leq \delta_v$, where $\Delta x_L$ is the width of the top of $LD_{i_vBp}$ and $\Delta x_R$ is the width of the top of $RD_{i_vBq}$. 
Condition 5: \[ \frac{LD_i vB^p(y) - RD_i vB^q(y)}{LD_i vB^p(y)} \leq \varepsilon_v \], where \( LD_i vB^p(y) \) and \( RD_i vB^q(y) \) are the y-coordinates of the top left corners of \( LD_i vB^p \) and \( RD_i vB^q \), respectively. Here, \( \delta_v \) and \( \varepsilon_v \) are set to 0.1.

In the first phase, each bounding box in the left image, \( LD_i vB^p \), is taken as the matching reference. Therefore, each bounding box in the left image will have many matching candidates, \( CRD_i vB^w \), where \( w \) is the number of matched bounding boxes in the right image, \( RD_i vB^q \), after the first phase. If a matching reference in \( LD_i vB^p \) cannot find any matching candidate from the \( RD_i vB^q \), then the bounding box in \( LD_i vB^p \) is deleted.

The second phase uses a matching algorithm to find the best-matched bounding box for each matching set, \((LD_i vB^p, CRD_i vB^w)\), where \( p \) is the number of bounding boxes in the left image, and \( w \) is the number of the matched bounding boxes in the right image. The matching algorithm for each matching set is as follows:

**Step 1:** For the low-resolution image \( LD(x, y) \), obtain the top- and bottommost points of a connected component in \( LD_i vB^p \), and use these as reference points.

**Step 2:** For the low-resolution image \( RD(x, y) \), obtain the top- and bottommost points from each matching candidate in \( CRD_i vB^w \), and use these as matching points.

**Step 3:** Consider each reference point as the central point of a 3 \( \times \) 3-pixel block. The search window size is 9 \( \times \) 15 pixels centered on the matching point. Use the full-search block-matching algorithm for each reference point in the left image to find the best-matching candidate that has the least total matching difference in the right image.

**Step 4:** Upsample the coordinates of the reference points of the left image and the matching points of the best-matching candidate of the right image to the original images \{\( L(x, y) \), \( R(x, y) \)\}. The block size is set to 5 \( \times \) 5 pixels centered on the upsampled reference points, and the search window size is 7 \( \times \) 7 pixels centered on the upsampled matching points of the best-matching candidate. For the reference points of the left image, use the full-search block-matching algorithm to refine the matching point that has the least matching difference in the right image.

**Step 5:** Carry out steps 1-4 in succession until all of the matching set is processed.

After the horizontal and vertical line segment matching processes, the horizontal and vertical line segments are matched for the stereo pair images. The differences can be calculated from the matched points of the matched line segments. Therefore, the distance of each extreme point of a line segment can be calculated from the difference value. The distance of the leftmost point of a horizontal line segment is computed as follows:

\[ D_{h_{Left}} = \frac{B \times f}{[L.hl^k(x) - R.hl^k(x)]} \],

where \([L.hl^k(x) - R.hl^k(x)]\) is the difference of the leftmost point of the horizontal line segment, \( B \) is the baseline of the left and right cameras, and \( f \) is the focal length of the left and right cameras.
The distance of the rightmost point of the horizontal line segment is computed as follows:

\[ D_{h_{\text{Right}}} = \frac{B \times f}{[L_{h_{x}}(x) - R_{h_{x}}(x)]} \]  

(2)

where \([L_{h_{x}}(x) - R_{h_{x}}(x)]\) is the difference of the rightmost point of the horizontal line segment.

The distances of the top- and bottommost points of a vertical line segment are computed as follows:

\[ D_{v_{\text{Top}}} = \frac{B \times f}{[L_{v_{x}}(x) - R_{v_{x}}(x)]} \quad \text{and} \]

\[ D_{v_{\text{Bottom}}} = \frac{B \times f}{[L_{v_{x}}(x) - R_{v_{x}}(x)]} \]  

(3) \hspace{1cm} (4)

where \([L_{v_{x}}(x) - R_{v_{x}}(x)]\) and \([L_{v_{x}}(x) - R_{v_{x}}(x)]\) are the differences of the top- and bottommost points, respectively.

In the vehicle search and distance estimation module, the front vehicles and their distances will be found and estimated using the average distances of the horizontal and vertical line segments based on the right image. The average distances of the horizontal and vertical line segments, respectively, are defined as:

\[ D_{h_{\text{avg}}} = \frac{D_{h_{\text{Left}}} + D_{h_{\text{Right}}}}{2} \]  

(5) \hspace{1cm} D_{v_{\text{avg}}} = \frac{D_{v_{\text{Top}}} + D_{v_{\text{Bottom}}}}{2} \]  

The details of this module are described in the following steps.

**Step 1:** Delete the horizontal lines not located on a vertical plane that is parallel to the image plane in the real world.

If \(\left|\frac{D_{h_{\text{Left}}} - D_{h_{\text{Right}}}}{D_{h_{\text{avg}}}}\right| \geq \tau\), delete the record of the horizontal line segment.

**Step 2:** Delete the vertical lines not located on a vertical plane that is parallel to the image plane in the real world.

If \(\left|\frac{D_{v_{\text{Top}}} - D_{v_{\text{Bottom}}}}{D_{v_{\text{avg}}}}\right| \geq \tau\), delete the record of the vertical line segment.

**Step 3:** Working from the top to the bottom row on the right image, calculate the average distance of the first horizontal line segment to select the horizontal line segments that have similar average distances and overlap in the vertical direction within the image.

Assume \(D_{h_{\text{avg}}} \) and \(D_{h_{\text{avg}}} \) denote the average distances of the first horizontal
line segment and the matching horizontal line segment, respectively. Assume \( l_1(x) \) and \( r_1(x) \) are the \( x \)-coordinates of the left- and rightmost points of the first line segment, respectively, and \( l'(x) \) and \( r'(x) \) are the \( x \)-coordinates of the left- and rightmost points of the matching horizontal line segment, respectively.

If \( \{(D_{h_{\text{avg1}}} - D_{h'_{\text{avg1}}} \leq \frac{1}{10} \times D_{h_{\text{avg1}}}] \land [(l' \leq r_1 \leq r') \lor (l' \leq l_1 \leq r' \land r' \leq r)\} \), the matching horizontal line segment is clustered with the first horizontal line segment.

**Step 4:** Delete the records of the first horizontal line segment and the clustered horizontal line segments.

**Step 5:** If all the horizontal line segments have been clustered, go to step 6. Otherwise, go to step 3.

**Step 6:** Every cluster of horizontal line segments is a detected obstacle. Compute the average value of the average distances of the horizontal line segments for each cluster as the distance to the detected obstacle.

**Step 7:** Working from the leftmost to the rightmost column of the image, calculate the average distance of the first vertical line segment to select the vertical line segments that have similar average distances and overlap in the horizontal direction within the image.

Assume \( D_{v_{\text{avg1}}} \) and \( D_{v'_{\text{avg1}}} \) denote the average distances of the first vertical line segment and the matching vertical line segment, respectively. Assume \( t_1(y) \) and \( b_1(y) \) are the \( y \)-coordinates of the top- and bottommost points of the first vertical line segment, respectively, and \( t'(y) \) and \( b'(y) \) are the \( y \)-coordinates of the top- and bottommost points of the matching vertical line segment, respectively.

If \( \{(D_{v_{\text{avg1}}} - D_{v'_{\text{avg1}}} \leq \frac{1}{10} \times D_{v_{\text{avg1}}}] \land [(t' \leq t_1 \leq b') \lor (t' \leq b_1 \leq b') \lor (t_1 \leq t' \land b' \leq b_1)\} \), the matching vertical line segment is clustered with the first vertical line segment.

**Step 8:** Delete the records of the first vertical line segment and the clustered vertical line segments.

**Step 9:** If all the vertical line segments have been clustered, go to step 10. Otherwise, go to step 7.

**Step 10:** Every cluster of vertical line segments is a detected obstacle. Compute the average value of the average distances of the vertical line segments for each cluster as the distance of the detected obstacle.

**Step 11:** Check the detected obstacles. If any obstacle detected by the vertical line segments overlaps the obstacles detected by the horizontal line segments, delete the obstacle detected by the vertical line segments.

According to the experimental results shown in Figs. 7-9, the distance error caused by the image resolution is approximately \( \pm 0.1 \) meter per meter. Therefore, the threshold value, \( \tau \), is set to 0.2. In steps 3 and 7, the threshold values are defined as 0.1 \( D_{h_{\text{avg1}}} \) and 0.1 \( D_{v_{\text{avg1}}} \) for the same reason.
4. EXPERIMENTAL RESULTS

Fig. 3 shows an example that illustrates the differences between synchronous and asynchronous cameras. As the exposure times of the low-cost CMOS cameras are asynchronous, the 3D variations ($\Delta x$, $\Delta y$, and $\Delta z$) must be considered in the matching processes for the right and left images. Assuming that the left image is captured earlier than the right image, then $\Delta z$ is the relative distance error between the ego vehicle and the front vehicle in the right image.

In addition, the asynchronous exposure time combined with vibrations of the ego vehicle will cause $\Delta x$ and $\Delta y$ variations. Therefore, the proposed vehicle detection algorithm focuses on the matching problems caused by the $\Delta x$ and $\Delta y$ variations.

The left and right images in Fig. 4 are captured at exactly the same time by two synchronous cameras. Fig. 4 (a) is an oscilloscope image of the two video signals, and Figs. 4 (b) and (c) are the left and right images captured by the synchronous cameras, respectively. As the red lines in Figs. 4 (b) and (c) cover the same row, we can obviously match the left and right images in the same row.

The video signals and output images of the asynchronous binocular system of the proposed system are shown in Fig. 5. As the low-cost CMOS cameras cannot capture images at exactly the same time, vibrations of the cameras and vehicle will influence the
apparent positions of the front vehicles in the captured images. Therefore, we cannot use existing algorithms developed for synchronous cameras to match the corresponding points in the stereo pair images captured with asynchronous cameras.

In the following experiments, 320 × 240-pixel grayscale test images were captured by two low-cost asynchronous CMOS cameras at a frame rate of 30 fps. The system used an Intel Pentium IV CPU with a clock speed of 2.6 GHz, 512 MB of RAM, and software developed using Visual C++ 2005. The proposed algorithm was tested on numerous image sequences captured under different illumination and road conditions. The time required to process each stereo image pair was between $25 \times 10^{-3}$ and $30 \times 10^{-3}$s. The proposed stereovision system worked smoothly on urban roads and freeways in real time.

Static and dynamic analyses were conducted to analyze the characteristics and performance of the asynchronous binocular platform and the proposed vehicle detection algorithm. Only the right-hand images are shown.

**Static analyses**

For static analyses, the proposed stereovision system was fixed on a platform where the roll angle, pitch angle, and height could be adjusted to simulate the variations in setup and vehicle configurations. The adjustment ranges were set as follows: detection distance, 4-50 m; height, 105-135 cm; pitch angle, ±5°; and roll angle, ±10°.

The static vehicle detection results achieved using the proposed stereovision system are shown in Fig. 6. Here the distances to the static vehicle were varied from 4-50m to verify the accuracy of the vehicle detection algorithm for static analyses. Fig. 6 shows that the proposed vehicle detection algorithm has satisfactory accuracy.

Fig. 7 shows the experimental results for the distance error and the platform height. The distance error is the difference between the actual distance and the detected distance, and the distance accuracy rate is the percentage of the distance error over the actual distance. The experimental results in Fig. 7 show that the platform height does not influence the distance detection, and the distance accuracy rate is greater than 92% for different platform heights.

The platform height was fixed to 120 cm for the pitch and roll angle experiments because the height of the rearview mirror of the experimental vehicle was 120 cm. Figure 8 shows the relationship between the distance error and the platform roll angle. The distance accuracy rate was greater than 90% and the distance error was approximately ±0.1 meter per meter when the platform roll angles were adjusted in the range ±10°.
REAL-TIME FRONT VEHICLE DETECTION ALGORITHM

Fig. 6. Distance measurements for a static vehicle.

<table>
<thead>
<tr>
<th>Distance Error (m)</th>
<th>Platform Height (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>105 cm</td>
</tr>
<tr>
<td>Accuracy Rate (%)</td>
<td>105 cm</td>
</tr>
<tr>
<td>Real Distance (m)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.10 / 97.5</td>
</tr>
<tr>
<td>5</td>
<td>0.11 / 93.8</td>
</tr>
<tr>
<td>6</td>
<td>0.20 / 93.8</td>
</tr>
<tr>
<td>7</td>
<td>0.02 / 94.8</td>
</tr>
<tr>
<td>8</td>
<td>0.10 / 95.7</td>
</tr>
<tr>
<td>9</td>
<td>0.10 / 94.3</td>
</tr>
<tr>
<td>10</td>
<td>0.20 / 97.9</td>
</tr>
<tr>
<td>15</td>
<td>0.10 / 99.4</td>
</tr>
<tr>
<td>20</td>
<td>0.30 / 98.0</td>
</tr>
<tr>
<td>25</td>
<td>1.80 / 92.5</td>
</tr>
<tr>
<td>30</td>
<td>1.30 / 99.2</td>
</tr>
<tr>
<td>40</td>
<td>1.10 / 94.0</td>
</tr>
<tr>
<td>50</td>
<td>2.10 / 93.5</td>
</tr>
</tbody>
</table>

Fig. 7. Distance error as a function of platform height.

Fig. 9 shows the relationship between the distance error and the platform pitch angle. The distance accuracy rate was greater than 90.4% and the distance error was approximately ± 0.1 meter per meter for platform pitch angles in the range ± 5°.
### Platform Roll Angle (°) - Distance Error (m) - Accuracy Rate (%)

<table>
<thead>
<tr>
<th>Real Distance (m)</th>
<th>Platform Roll Angle (°)</th>
<th>Right 8°</th>
<th>Right 4°</th>
<th>Right 2°</th>
<th>Left 4°</th>
<th>Left 8°</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td>0.10 / 97.5</td>
<td>0.08 / 98.0</td>
<td>0.18 / 95.5</td>
<td>0.15 / 96.3</td>
<td>0.08 / 98.0</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.14 / 97.2</td>
<td>0.40 / 92.0</td>
<td>0.41 / 91.8</td>
<td>0.38 / 92.4</td>
<td>0.06 / 98.8</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.08 / 98.6</td>
<td>0.32 / 94.7</td>
<td>0.18 / 97.0</td>
<td>0.27 / 95.5</td>
<td>0.14 / 97.7</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0.15 / 97.8</td>
<td>0.53 / 92.4</td>
<td>0.50 / 92.9</td>
<td>0.36 / 94.9</td>
<td>0.02 / 99.7</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.05 / 99.4</td>
<td>0.22 / 97.3</td>
<td>0.40 / 95.0</td>
<td>0.28 / 96.5</td>
<td>0.04 / 99.5</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.10 / 98.9</td>
<td>0.09 / 99.0</td>
<td>0.77 / 91.4</td>
<td>0.50 / 94.4</td>
<td>0.05 / 99.4</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.30 / 97.0</td>
<td>0.29 / 97.1</td>
<td>0.45 / 95.5</td>
<td>0.90 / 91.0</td>
<td>0.70 / 93.0</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>1.20 / 92.0</td>
<td>0.70 / 95.3</td>
<td>0.57 / 96.2</td>
<td>1.06 / 92.9</td>
<td>0.60 / 96.0</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.80 / 96.0</td>
<td>1.70 / 91.5</td>
<td>1.40 / 93.0</td>
<td>1.90 / 90.5</td>
<td>0.70 / 96.5</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>0.90 / 96.4</td>
<td>0.12 / 99.5</td>
<td>2.30 / 90.8</td>
<td>2.15 / 91.4</td>
<td>0.50 / 98.0</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.20 / 99.3</td>
<td>0.20 / 93.3</td>
<td>0.67 / 97.8</td>
<td>2.55 / 91.5</td>
<td>1.90 / 93.7</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>2.50 / 93.8</td>
<td>0.30 / 99.3</td>
<td>2.70 / 93.3</td>
<td>2.40 / 93.5</td>
<td>1.50 / 96.3</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>1.80 / 96.4</td>
<td>3.20 / 93.6</td>
<td>3.10 / 93.1</td>
<td>2.70 / 94.6</td>
<td>2.10 / 95.8</td>
</tr>
</tbody>
</table>

![Fig. 8. Distance error as a function of platform roll angle.](image)

### Platform Pitch Angle (°) - Distance Error (m) - Accuracy Rate (%)

<table>
<thead>
<tr>
<th>Real Distance (m)</th>
<th>Platform Pitch Angle (°)</th>
<th>Up 5°</th>
<th>Up 3°</th>
<th>Up 1°</th>
<th>Down 3°</th>
<th>Down 5°</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td>0.21 / 97.5</td>
<td>0.04 / 99.0</td>
<td>0.26 / 93.5</td>
<td>0.13 / 96.8</td>
<td>0.32 / 92.0</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.17 / 97.8</td>
<td>0.32 / 93.6</td>
<td>0.21 / 95.8</td>
<td>0.38 / 92.4</td>
<td>0.09 / 98.2</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.27 / 96.7</td>
<td>0.27 / 95.5</td>
<td>0.34 / 94.3</td>
<td>0.20 / 96.7</td>
<td>0.24 / 96.0</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0.38 / 99.7</td>
<td>0.08 / 98.9</td>
<td>0.41 / 94.1</td>
<td>0.21 / 97.0</td>
<td>0.09 / 98.7</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.35 / 98.8</td>
<td>0.07 / 99.1</td>
<td>0.35 / 95.6</td>
<td>0.22 / 97.3</td>
<td>0.38 / 95.3</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.16 / 98.9</td>
<td>0.19 / 97.9</td>
<td>0.31 / 96.6</td>
<td>0.04 / 99.6</td>
<td>0.15 / 98.3</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.07 / 98.0</td>
<td>0.30 / 97.0</td>
<td>0.46 / 95.4</td>
<td>0.40 / 96.0</td>
<td>0.05 / 99.5</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.40 / 99.3</td>
<td>0.70 / 95.3</td>
<td>1.35 / 91.0</td>
<td>1.30 / 91.3</td>
<td>0.30 / 98.0</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.70 / 98.5</td>
<td>0.20 / 99.0</td>
<td>1.75 / 91.3</td>
<td>1.80 / 91.0</td>
<td>0.70 / 96.5</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>0.20 / 92.8</td>
<td>0.60 / 97.6</td>
<td>0.73 / 97.1</td>
<td>0.10 / 99.6</td>
<td>0.10 / 99.6</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.90 / 95.7</td>
<td>1.90 / 93.7</td>
<td>2.10 / 93.0</td>
<td>0.80 / 97.3</td>
<td>0.20 / 99.3</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>1.20 / 97.3</td>
<td>0.50 / 98.8</td>
<td>1.50 / 96.3</td>
<td>1.80 / 95.5</td>
<td>1.20 / 97.0</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>2.10 / 95.8</td>
<td>3.20 / 93.6</td>
<td>3.50 / 93.0</td>
<td>0.60 / 98.8</td>
<td>3.20 / 93.6</td>
</tr>
</tbody>
</table>

![Fig. 9. Distance error as a function of platform pitch angle.](image)

We can determine the relationships between the distance error, roll angle, pitch angle, and height of the proposed stereovision system from Figs. 7-9. The detection ranges of the system were from 4-50 m and the distance accuracy rate was over 90%.

### Dynamic analyses

The test environments were divided into freeway and urban road segments for the dynamic analyses. Illumination conditions, including sun, clouds, and night, and various road environments, including bumps, curves, and steep hills, were included to analyze the performance of the proposed system. The detection rate is defined as the percentage of front vehicles detected correctly in the detection zone. If the detection system has wrong or missing detection in the detection zone, this is counted as false detection.

### Freeway experiments under various illumination conditions

(a) Sunny day

The ego vehicle maintained a speed of 90-100 km/h to test the proposed system on
the freeway. All of the front vehicles in the test images were selected randomly without prior planning. In Fig. 10, the ego and front vehicles are on a freeway exit ramp. The detection rate during the sunny test images exceeded 93.8%.

(b) Cloudy day

The horizontal and vertical edges of the obstacles on cloudy days may not be detected completely because the gray values of the background are close to those of the obstacles. The detection rate for the cloudy test images was approximately 88.6%. Fig. 11 shows some detection results on the freeway on a cloudy day.

(c) Night

The detection rate of the proposed system dropped to 78.1% at night. The major causes of false detection were light sources, such as headlights and tail lights that appeared suddenly. These spots of light make vehicle outlines unclear and disrupt the edge segments. Fig. 12 shows some detection results on the freeway at night.

Fig. 10. Detection results on a freeway on a sunny day.

Fig. 11. Detection results on the freeway on a cloudy day.

Fig. 12. Detection results on the freeway at night.
Experiments on urban roads under various road conditions

(a) Uphill or downhill

The ego vehicle maintained a speed of 50-70 km/h to test the proposed system on urban roads. The images in Figs. 13 and 14 were captured on uphill and downhill sections of road, respectively, where the road slope was between $-5^\circ$ and $+15^\circ$. The system still detected the front vehicle accurately and the detection rate of the proposed system was 93.9%.

(b) Curved road

The image sequences in Fig. 15 were captured on a curved road where the ego vehicle maintained a speed of 35 km/h as it followed the vehicle ahead. The front vehicle was detected easily, and the detection rate of the system was as high as 89%.
(c) Bumpy road

Fig. 16 shows the detection results on a bumpy road. The road was being resurfaced and the asphalt in the right lane had been removed. The vehicles on the left side of Fig. 19 could not be detected because of the fragmented line segment and non-overlap area between the left and right images. However, the lead vehicle was still detected stably on the bumpy road, and the detection rate of the proposed system was 91.2% under these conditions.

<table>
<thead>
<tr>
<th>Table 1. Detection rate of the proposed system on the freeway.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Illumination Conditions</strong></td>
</tr>
<tr>
<td>Number of frames</td>
</tr>
<tr>
<td>False detections</td>
</tr>
<tr>
<td>Detections</td>
</tr>
<tr>
<td>Detection rate (%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Detection rate of the proposed system on urban roads.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Road Conditions</strong></td>
</tr>
<tr>
<td>Number of frames</td>
</tr>
<tr>
<td>False detections</td>
</tr>
<tr>
<td>Detections</td>
</tr>
<tr>
<td>Detection rate (%)</td>
</tr>
</tbody>
</table>

Tables 1 and 2 summarize the detection rates using the proposed stereovision system on the freeway and on urban roads, respectively.

5. CONCLUSIONS

This paper described a multi-resolution stereovision system for detecting the front-vehicle under various road and illumination conditions. An asynchronous binocular platform and a real-time front-vehicle detection algorithm were proposed as the heart of the system to detect the distances to leading vehicles. The asynchronous binocular platform provides a small low-cost obstacle detection system that is easy to set up and is practical.
for real-world applications. The vehicle detection algorithm was used to overcome the matching problem in real time caused by the asynchronous exposure times of the CMOS cameras. Static and dynamic analyses were conducted to analyze the characteristics and performance of the proposed system. The accuracy of the vehicle detection algorithm was tested statically at distances in the range 4-50m, and the roll angle, pitch angle, and height of the binocular system were adjusted to simulate various setups and vehicle variations. The proposed system was evaluated dynamically under different illumination and road conditions. The system overcomes various issues related to the complexity of the road conditions and asynchronous image capture problems. The experimental results obtained with various test images showed that the proposed system can successfully detect the front vehicles and estimate their distances under various illumination and road conditions.

REFERENCES

22. R. Labayrade, D. Aubert, and J. P. Tarel, “Real time obstacle detection on non flat road geometry through V-disparity representation,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2002, pp. 646-651.
2375.

Chung-Cheng Chiu (鍾正) received the B.S. degree and M.S. degree in Electrical Engineering from the Chung Cheng Institute of Technology, Taiwan in 1990 and 1993 respectively, and the Ph.D. degree in department of Electrical and Control Engineering at National Chiao Tung University, Hsinchu, Taiwan in 2004. Since 1993, he is a lecturer of the department of Electrical Engineering at the Chung Cheng Institute of Technology. In 2005, he became an Associate Professor. In 2003, he received Dragon Golden Paper Award sponsored by the Acer Foundation and the Silver Award of Technology Innovation Competition sponsored by the AdvanTech. His research interests include image processing, image compression, document segmentation, computer vision and applications on intelligent transportation system.

Meng-Liang Chung (鍾孟良) received the B.S. degree in Naval Engineering from National Taiwan Ocean University, Keelung, Taiwan, in 2005, and M.S. degree in Electrical Engineering from Chung Cheng Institute of Technology, National Defense University, Taoyuan, Taiwan, in 2007. He is interested in image processing and applications on intelligent transportation system, especially on object segmentation and matching.

Wen-Chung Chen (陳文忠) received the M.S. degree in Electrical Engineering from Chung Cheng Institute of Technology, National Defense University, Taoyuan, Taiwan, in 2008. He is specialized in the applications of embedded systems. His research interests include the applications on intelligent transportation system and surveillance system, especially on system integration and design.